



# Network Station Manager Version 2

## Installing Applications on Windows Terminal Server

Network Station Education  
IBM NCD  
June 1999



# Objectives

---



- A **very brief** and condensed summary of the multi-user Microsoft Windows NT server application installation issues
- **What's the problem?**
- **Tools and techniques to fix**
  - Change User Command
  - Home Directories
  - Home Drives
  - Logon Scripts
  - Application Installation Scripts

# Notes

---



This presentation is about installing applications on Windows Terminal Server.

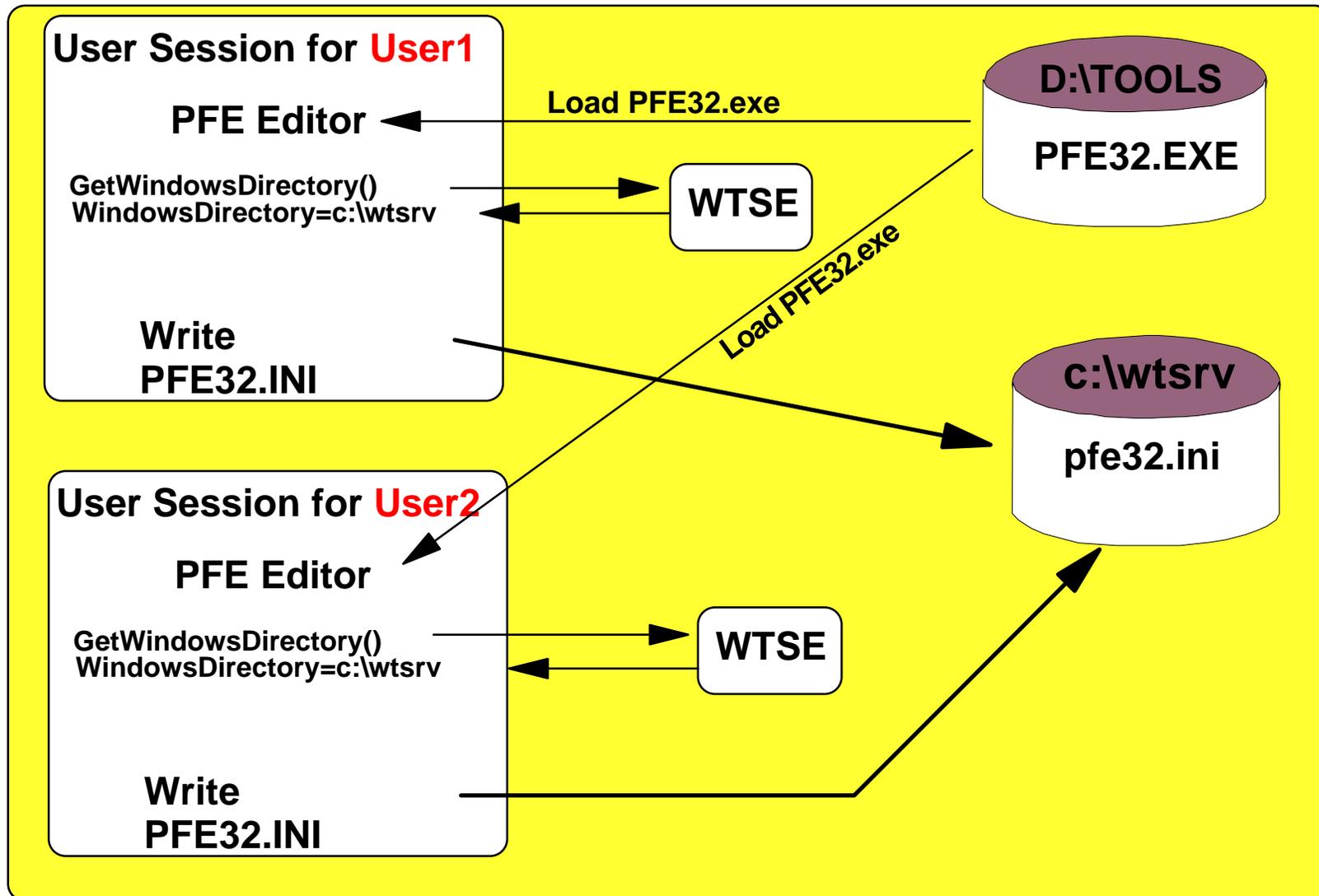
The objective is to give you a very, very brief and condensed summary of some of the installation issues that you run into when you're installing an application on a multi-user Windows NT Server system.

Basically, all we do here is describe what the problem is and identify some of the tools and techniques that are available to fix these problems.

I highly recommend that you read some of the other documentation that's available on this subject because there is no way we to do justice to this particular topic in just a few minutes.

In fact, I suggest that you read the complete chapter on this that's present in the redbook on the CD or access some of the other documentation that we reference at the end of this topic.

# What's The Problem? (Simplification)



# Notes

---



What is the problem? Actually, it is not easy to describe in a few words, but let me try to over simplify the problem by using this diagram.

Look at the top portion left hand portion of the diagram where I have User Session for User1, and assume that this user starts using an editor called PFE32.

The editor is loaded from the tools directory and after it loads, it issues a GetWindowsDirectory to the API in order to find out where the Windows directory is, because that is where it needs to write a PFE32.INI file.

The response it gets from the Windows Terminal Server Edition system is wtsrv as the location of the system file. And so it writes PFE32.INI there with the user's characteristics, such as the colors he wants with this editor and some of the settings for margins, and so on.

But then comes along User Session for User number 2, and user2 does the same thing and starts the PFE32 Editor. That's fine, he can start two versions of this editor, but when the GetWindowsDirectory is issued by the second copy of the editor, it gets the same answer, which is that the Windows directory is located at c:\wtsrv, and it therefore writes PFE32.INI at the same location, thereby erasing any of the characteristics that were stored by User1.

That, in a nutshell, is the nature of the problem that we're facing in terms of running multiple applications simultaneously by multiple users.

# Compatibility Issues

---



- **Compatibility Issues**

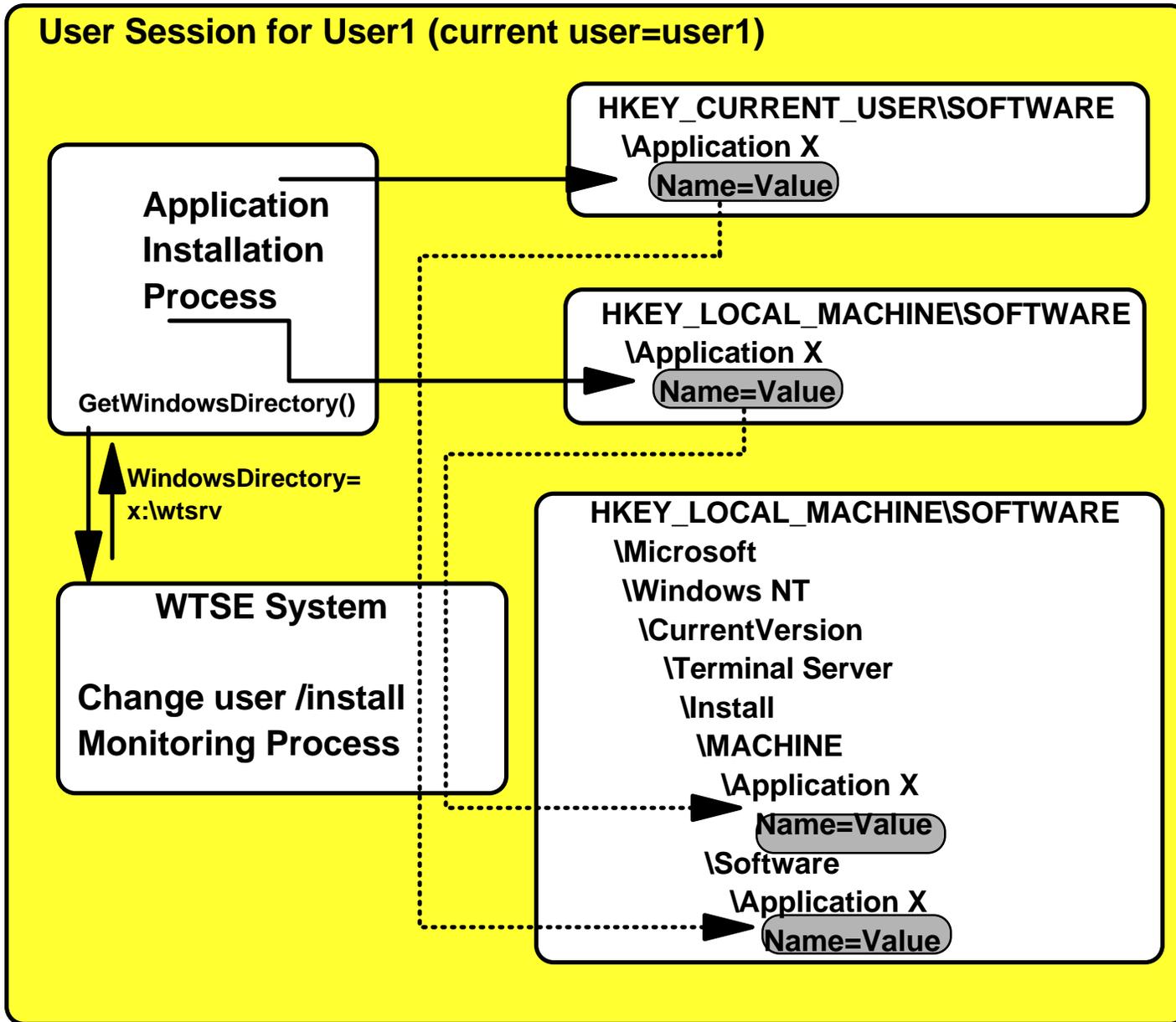
- Registry issues
- Files issues
- Object problems (Events, Devices, Semaphores)

- **Tools**

- The "change user" command
- Application Installation scripts
- User logon scripts
- Home drives and directories



# Change user /install - Saving Registry Keys



- **Command issued BEFORE launching an application installation process**

# Notes

---



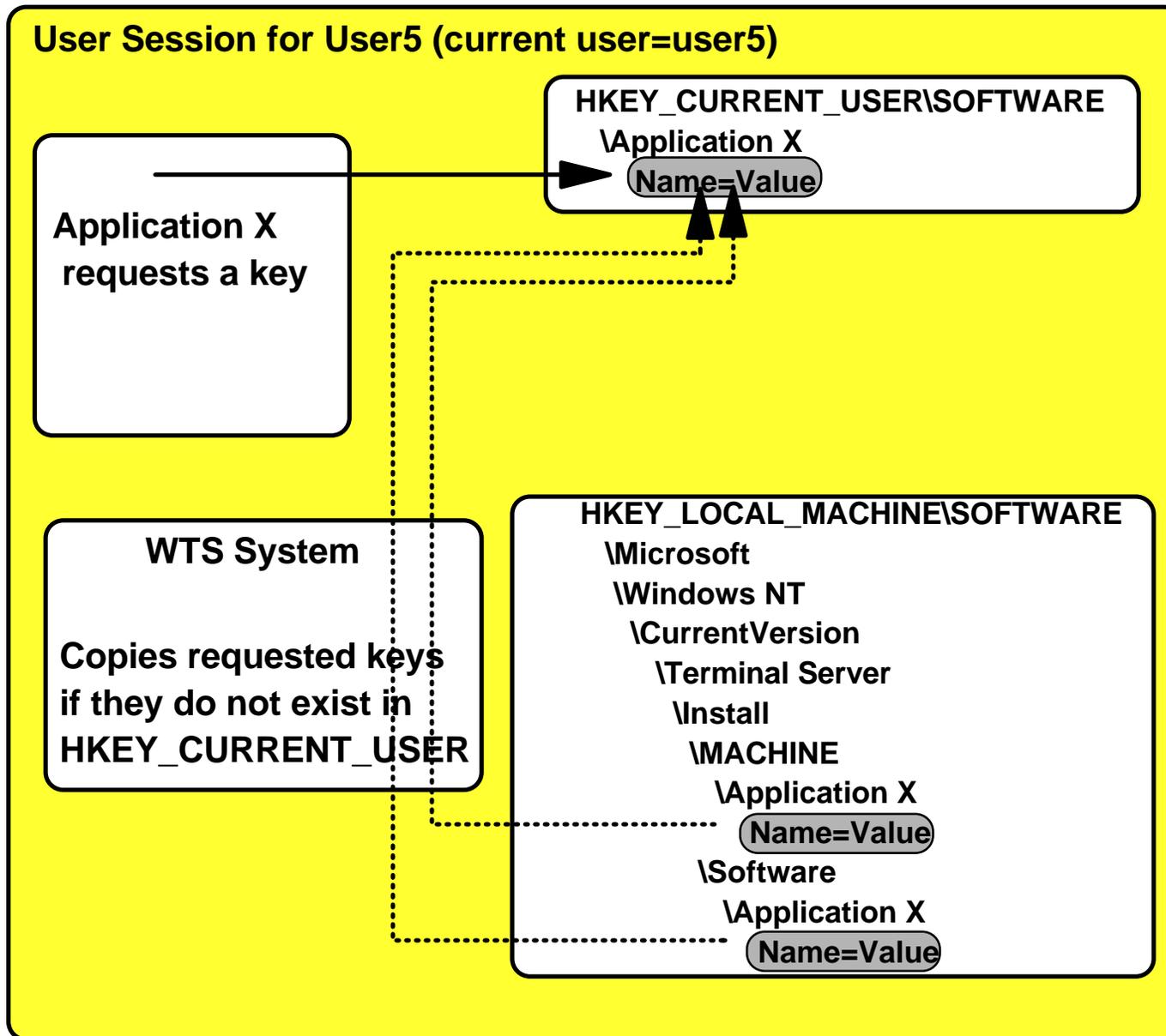
Let's take a look at the change user /install command. What does this do?

In the top left-hand corner of this diagram is an installation application process. When this installation process runs and creates keys in the Windows NT registry, whether it's in the current user or the HKEY\_LOCAL\_MACHINE portions of the registry, any keys and values that are written to these registries are automatically stored and saved in another location of the registry, identified here in the bottom right-hand corner.

In the case of Windows Terminal Server Edition, the key is under HKEY\_LOCAL\_MACHINE\ SOFTWARE \Microsoft\Windows NT and so on until you reach Terminal Server. Under Terminal Server is an identical copy of whatever gets stored by this application in this registry.

The change user /install command must be issued before launching the application process in order to alert the system that it must capture and save these keys as they are stored.

# Change User /execute - Restoring Keys



- **Command issued AFTER an application installation process completes**

# Notes

---



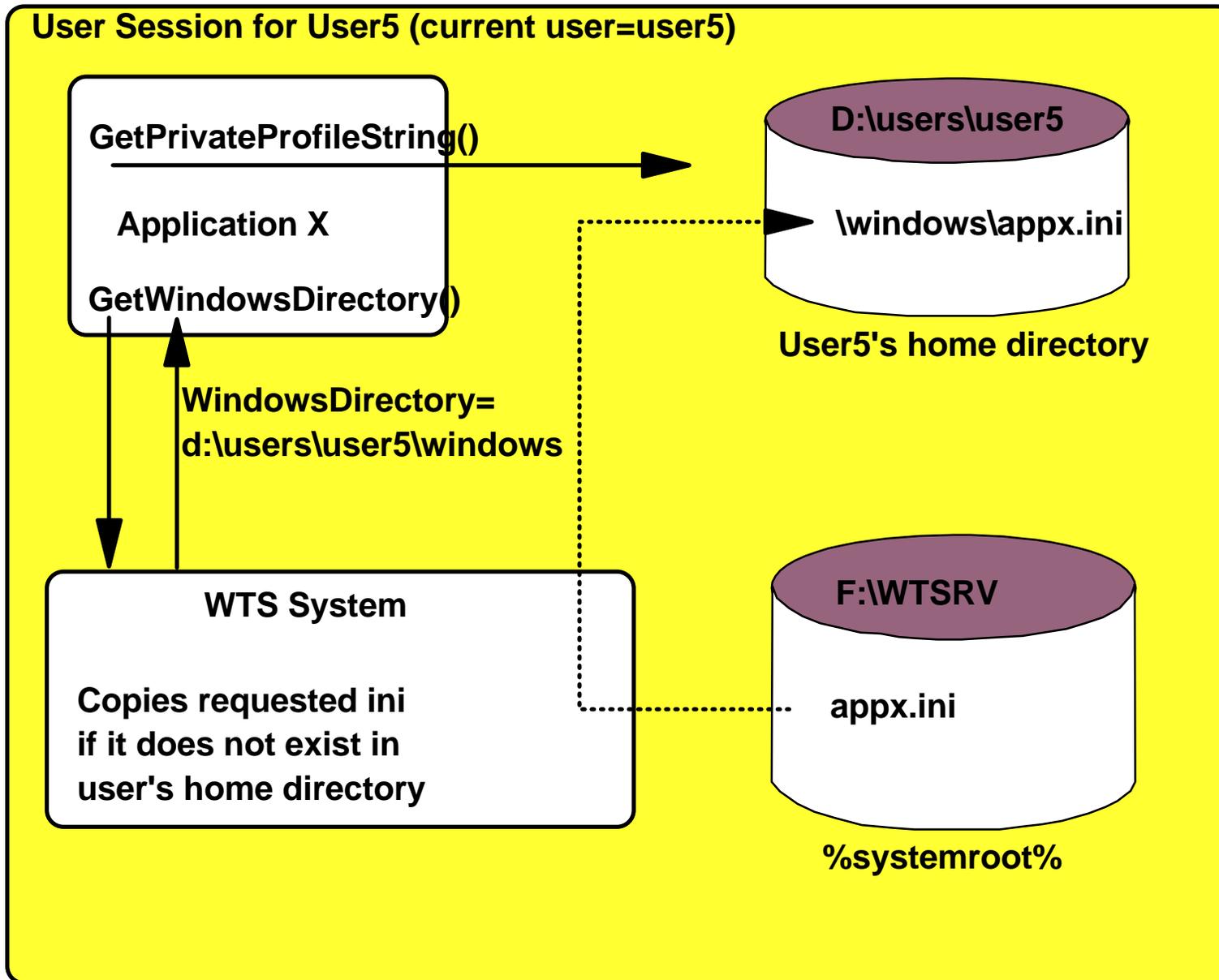
Once the installation process completes, the command `change user /execute` must be issued in order to revert back to a normal operational mode.

This is again an oversimplification of the process, but after you have installed the application and the first user logs in to use this application, that application requests a key from the `HKEY_CURRENT_USER` location.

When the system finds out that the key does not exist in that registry location, the system automatically copies the missing key from the registry location where it was stored during the installation to the current user software application, or wherever the location of the key is supposed to be.

Another aspect of the `change user /install` and `/execute` commands is, if you remember our first example, that when an application issues the `GetWindowsDirectory`, it is supposed to get back the location of the system root directory. However, after issuing the `change user /execute` command, what gets returned to the application is the specific Windows directory that belongs to each specific user.

# Change User /execute - Restoring INI



# Notes

---



In this example shown on this chart, application X issues a `GetPrivateProfileString` in order to retrieve an `.ini` file from the Windows directory.

If that file does not exist in the user's home directory (and note that this home directory is a Windows directory that has been created automatically by the Windows Terminal Server Edition system under the user's home directory) it is copied automatically by the system from the Windows directory, and from that point on, this `.ini` file becomes the user's private own `.ini` file for that particular application.

The next time that this user runs this application, it is going to request the `.ini` file from his own user's directory and therefore get his own user characteristics that were saved in that particular `.ini` file.

# What is a Home Directory?



**User Environment Profile**

User: Huebner (Edward Huebner)

OK  
Cancel  
Help

User Profiles

User Profile Path:

Terminal Server Profile Path:

Logon Script Name:

Home Directory

Local Path:

Connect  To

Terminal Server Home Directory

Local Path:

Connect  To

File Explorer Inset:

- Program Files
- Recycler
- Users
  - huebner
    - windows
    - system
  - newuser
  - userx1

- Home directory is a directory reserved for a specific user only
- It is created at the time that the user account is created
- WTSE also automatically creates a windows subdirectory in each user's home directory

# Notes

---



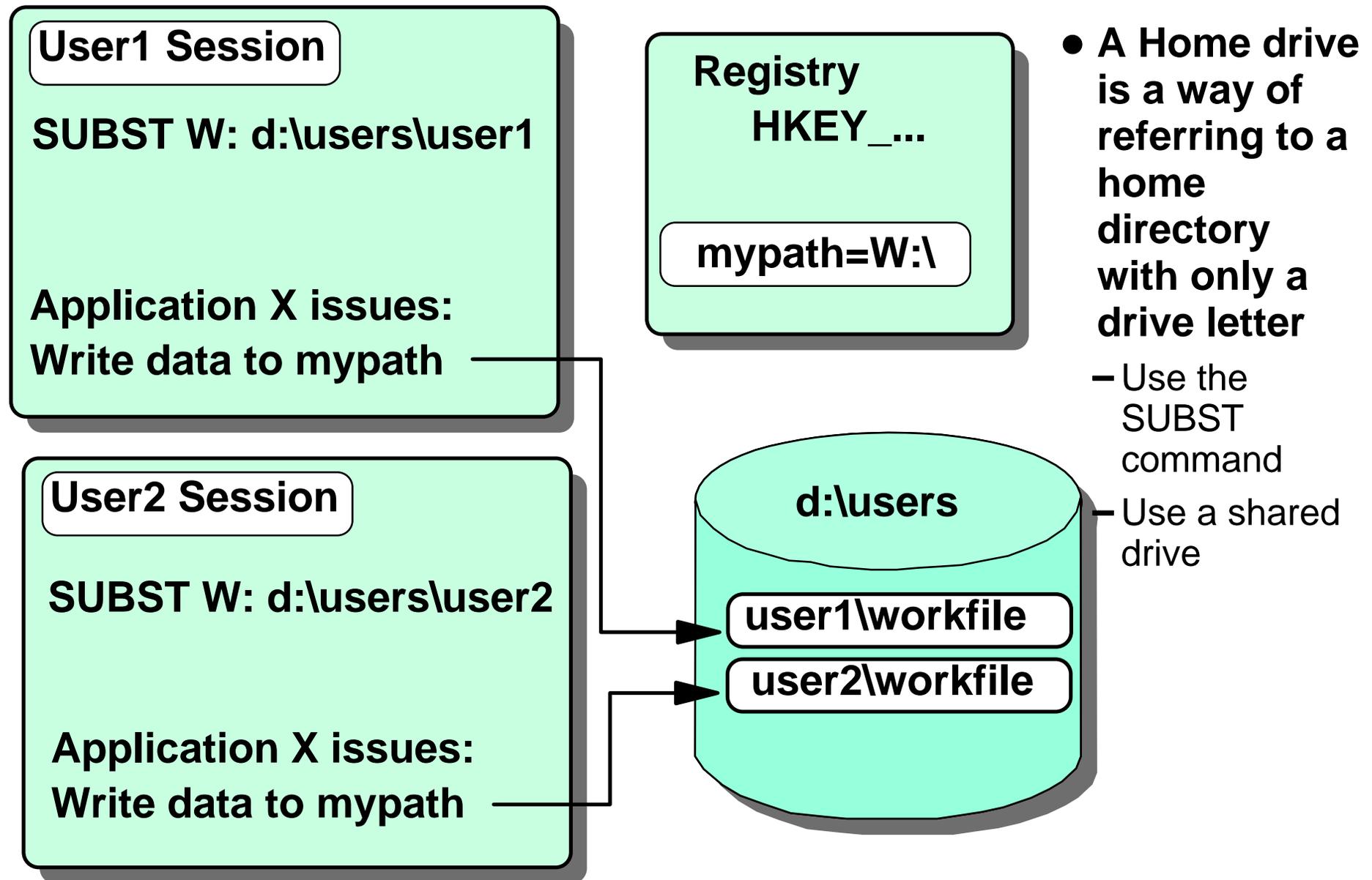
What's a home directory?

A home directory is an area on a disk reserved specifically for a particular user. It is only available to that user and it is created at the time that the user account is created on the Windows NT system.

When you do this on a non-multi-user Windows NT Server system, you just get a home directory created that is usually or typically under the name of the user name.

When this is done on a Windows Terminal Server Edition system, the system automatically adds a Windows subdirectory in each user's home directory, and this is where it'll store all of the .ini files automatically thereby providing for each user his own copies of a Windows directory.

# What is a Home Drive?



# Notes

---



What is a home drive then?

A home drive is a technique. It is, in fact, the same thing as a home directory except that instead of referring to the home directory with a path, such as an example we have here, `d:\users\user1` which would be user's 1 - User1's home directory, we refer to it with a single drive letter.

And there are a couple of techniques used in order to do this.

The first one is to use the `SUBST` command, which is illustrated here. By issuing, in User Session 1, the command `SUBST W` for `d:\users\user1`, in effect, I can refer to this path and directory with just the letter `W`.

I am therefore making the path generic so that if there is in the registry a key and a value that states, "`mypath=W:\`," as illustrated in this diagram, this means, for user1, that `mypath` is equal to `d:\users\user1`. But when User2 uses the same key in the registry, it points to `d:\users\user2`.

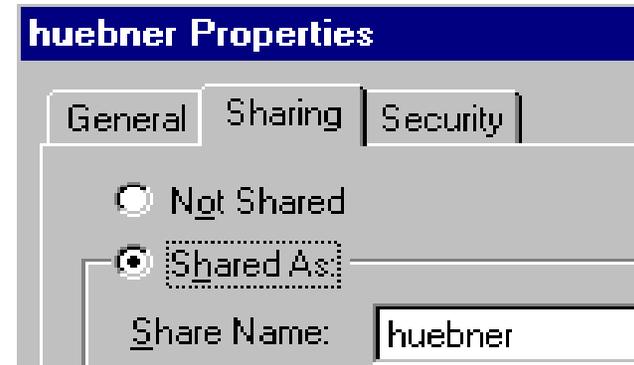
So in effect, a home drive is a way of specifying a generic drive that for individual users represents each individual user's own directory.

# Home Drive via Shared Drive



- **Create a shared drive**

- d:\users\huebner, on the server called "nstationserver" shared using the name "huebner"



- **Modify user account profile**

- Point to the shared drive using the Connect To Option



- **Equivalent to using the SUBST command**
- **Requires more work (must share drive for every user)**
- **Requires Server Options Pack in the case of WinFrame**
- **Do not use "Connect To" to create the Windows Subdirectory (Permissions not set correctly)**



# What is a Logon Script?



- A command file or batch file executed when the user logs on
- Can be used to map a drive letter to the user's home directory (home drive)
- Can be used to copy files from a template directory to the user's home directory
- Any other task required to initialize a user environment
- Can be triggered in three ways:
  - Command file placed in the user's STARTUP folder
  - Command file placed in the All Users STARTUP folder
  - Command file specified in the User Environment Profile
    - If no path, command file is in %systemroot%\system32\Rep\Import\Scripts

**User Environment Profile**

User: Huebner (Edward Huebner)

User Profiles

User Profile Path:

Terminal Server Profile Path:

Logon Script Name:

Home Directory

Local Path:

Connect To:  To



What is a logon script then?

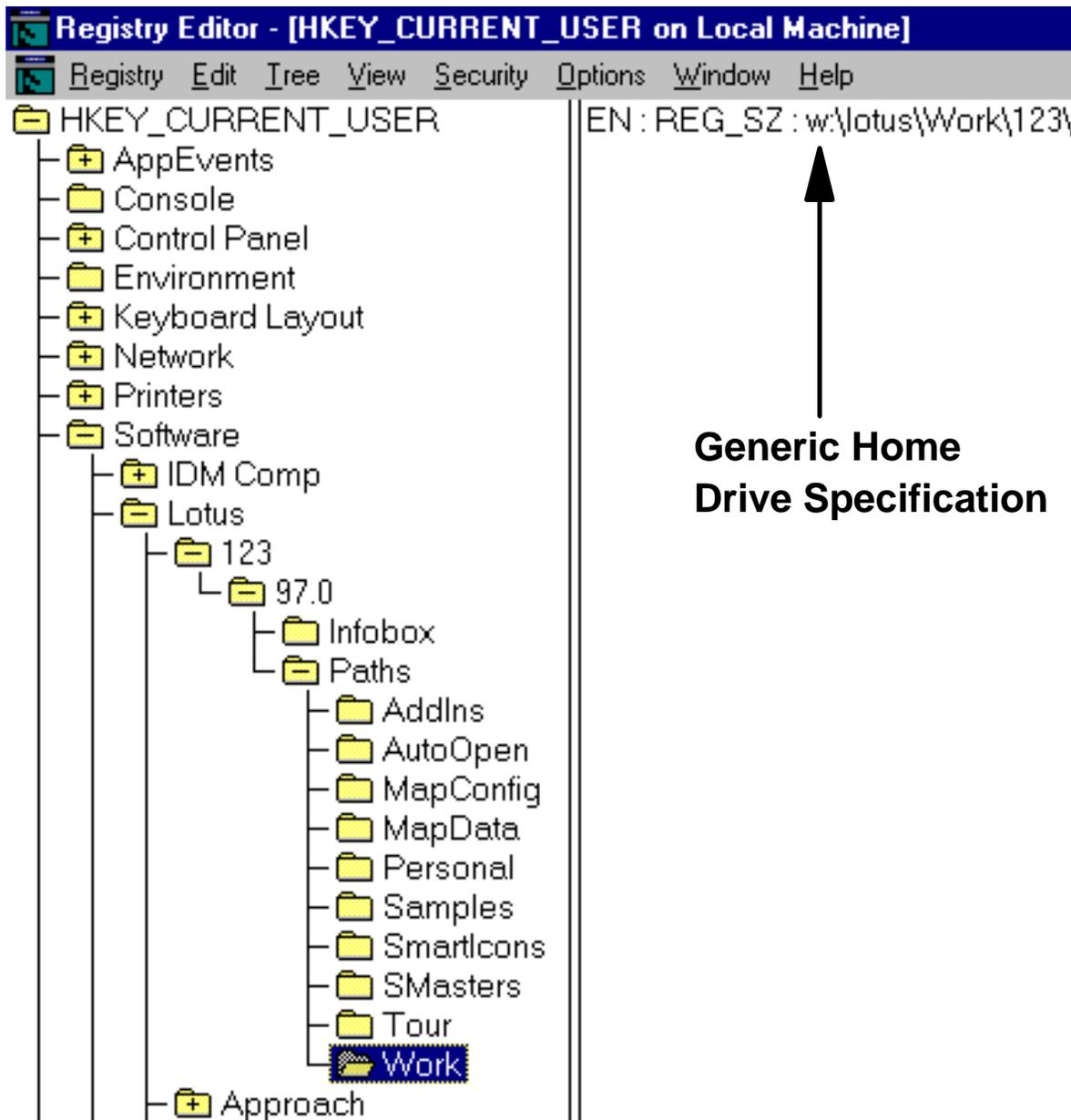
A logon script is a command or a batch file that is executed automatically when the user logs on. It can be used, for example, to map a drive letter like we just did when we used the SUBST command.

It can also be used to copy files from one directory to another when a user logs in. Typically, this might be because you have set up a template for a typical user that contains a lot of files that are applicable to each user, and you need to copy all these files over into his private home directory when the user logs on for the first time so that these files become his files from this point on. A logon script is the easiest way to accomplish this.

A logon script can be triggered in three ways.

- You can place the command file in the user's STARTUP folder
- Or you can put it in the All Users STARTUP folder, which then applies to any user
- Or you can specify the file in the User Account Environment Profile under the logon script name. If you do not use a path in front of the file name, the default path is %systemroot%\system32\Rep\ Import\Scripts as illustrated in this chart.

# Application Using Registry and User Files



- **Application creates registry entries with pointers to user files**
- **Use a generic drive during the install (if application allows it) or modify registry entry afterwards to use generic home drive**
  - Use a script supplied with WTSE
  - Create own script
- **Registry entry then becomes applicable to any user using a home drive of W:\**

# Notes

---



Here's an example of an application, in this case a Lotus 1-2-3 spreadsheet application, that uses registry entries to indicate the path to certain files that it needs to operate.

When this application was installed, it made an entry "w:\lotus\Work\123" in the registry under the Work subkey in the HKEY\_CURRENT\_USER directory. This was done because, during the Lotus application install process, when the question was asked of where are the user files to be located?, the answer provided was W. And we equated W, in this case, to the user's home directory so that from this point on then, any user using this registry entry will have these Lotus work files located in their own specific user's directory because W is a generic home drive specification.

Some applications, dependent on how the install process is done, require additional steps after installation to correct some settings. For example, using the lotus application as an example, if the install process had not asked the user to enter the path to the user file but instead chose to write its own path by choosing the default system path, then it would be written and recorded in the registry not with a generic home drive like W but pointing to a drive like C maybe, or some other drive. You would therefore need to modify this registry entry, after installation, to change it to W if you chose to use W as the generic home drive to use.

Some applications however do not require any corrective action because they "behave correctly" during the installation or because the installation process is designed to handle a multi-user environment.

The challenge is to be aware of which applications are well behaved and which ones are not.

# Application Notes Example



## Microsoft Internet Explorer 3.x

### Description of Issues

By default, Internet Explorer stores the user's history, cookies, and temporary Internet files in the %systemroot% directory. Since this directory is read-only for normal users, they are unable to run Internet Explorer. Also, all users would share the same set of data, which is undesirable.

### How to Solve

After installing Internet Explorer, run the %systemroot%\Application Compatibility Scripts\Install\MSIE30.cmd script.

This will change the directories used for the items listed above to W:\MSIE30, which is in the user's home directory.

### Customization

If the user's home directory is mapped to a drive letter other than W:, update the path in %systemroot%\Application Compatibility Scripts\Install\MSIE30.key.

### Limitations

None

# Notes

---



Many applications have scripts or application notes supplied in the Windows Terminal Server Edition system, on the CD, describing what each application requires in terms of steps to solve the compatibility issues.

This chart shows an example of an Application Notes for Microsoft Internet Explorer 3.x.

It contains a Description of the Issues for this particular application, how to solve them, how to customize, and whether there are any limitations to using this application in a multi-user environment.

Notice that in the "How to Solve" and in the "Customization" portion it refers to some scripts.

These are application installation scripts supplied with these applications that simply need to be executed following the application installation in order to correct some of these potential registry issues.

You can also customize these scripts before executing them to make them applicable to your own environment and change the defaults that they are using such as changing W as the home drive for Q or R for example.

# WTSE Supplied Application Scripts

---



Corel Office7  
Corel WordPerfect Suite 8  
Executive Software Diskeeper 2.0  
Lotus SmartSuite 97  
Microsoft BackOffice 2.5  
Microsoft BackOffice 4.0  
Microsoft Clipboard Viewer  
Microsoft Dr. Watson  
Microsoft Excel 97 (stand-alone installation)  
Microsoft Exchange 4.0 and 5.0 and higher  
Microsoft Exchange Server 5.5  
Microsoft FoxPro 2.6a  
Microsoft FrontPage 98  
Microsoft Internet Explorer 3.x  
Microsoft Internet Explorer 4.0  
Microsoft Network Monitor  
Microsoft ODBC

**Microsoft Office 4.3 and Office 4.3 Applications**  
**Microsoft Office 95 and Office 95 Applications**  
**Microsoft Office 97 and Office 97 Applications**  
**Microsoft Project 95**  
**Microsoft Project 98**  
**Microsoft SNA Server 3.0**  
**Microsoft SNA Server and Client 4.0**  
**Microsoft Word 97 (stand-alone installation)**  
**Netscape Communicator 4.x**  
**Netscape Navigator 3.x**  
**Novell IntraNetware Client for Windows NT 4.11a**  
**Seagate Backup Exec 7.0**  
**Sybase SQL Anywhere**

# Notes

---



As an example, the application scripts that were supplied with the initial version of Windows Terminal Server Edition are shown in this list.

Today these might be more available with the latest version of WTSE or on the Web so please verify these sources for the latest information.

Also, there are examples in some of the redbooks showing that if you set up your environment such that you use the techniques of the home drive and the logon script, in many cases you might not have to modify anything after installation.

# Where To Go For More Information

---



- **SG24-5221 - Chapter on Installing Applications in WTSE**
- **The terminal.doc file in the root directory of the WTSE CD**
  - This is probably the best overall current reference
- **The NCD Inc. WinCenter Connect System Administrator's Guide publication, Chapter 3, entitled Setting Up Multi-User Applications**
- **Application notes from NCD Inc. at their [www.ncd.com](http://www.ncd.com) web site**
- **Application notes and WinFrame Solutions Guide from Citrix Inc. available from the support area at their [www.citrix.com](http://www.citrix.com) web site**
- **The SG24-2127-01 redbook entitled IBM Network Station Guide for Windows NT**
- **Microsoft Windows NT White Paper entitled Guide To Microsoft Windows NT 4.0 Profiles and Policies at <http://www.microsoft.com/NTServer/Basics/TechPapers/default.asp> can help understand some of the issues**

# Notes

---



For additional information on this topic, please read the chapter on installing applications in the redbook SG24-5221 because it will give you a lot more detail than the short summary we just went through.

It will also direct you to read the terminal.doc file that's in the root directory of the Windows Terminal Server Edition CD. That is probably the best overall current reference on these compatibility issues and on the ways to solve them.

Other references are also listed here, all of which may be useful to provide you with a good understanding of the issues we have discussed.

Thank you.