Win4Lin Administration Guide: Win4Lin 9x and WTS

Table of Contents

Chapter 1 − Introduction **1**
Introduction 1

Chapter 2 − Windows Session Administration **2**
Graphical Administration Tools 2
Manage User Sessions 3

Chap 3 − Windows Session Configuration **5**
Relocate Windows C:\ Drive 5
Attach Physical and Virtual DOS Drives 5
Customize Windows Drive Labels 7
Force Proper Windows Shutdown 7
Identify Idle User Sessions 7

Chap 4 − Printer Configuration **9**
Share Printer With Samba 9
Setup Multiple Printers in Win4Lin 9

Chap 5 − Run Windows Applications **11**
Launch Windows Application From Linux Command Line 11
Configure Windows to Launch Single Application 11
Launch Linux Commands From Windows 12

Chap 6 − WTS Specific Configuration **13**
Setup Dynamic Profile−Based Provisioning 13
Troubleshoot Profile−Based Provisioning 14
Tunnel WTS Client via SSH 15

Appendix A: Environment Variables **16**
How to Use Environment Variables 16
Win4Lin Environment Variables 16

Chapter 1 − Introduction

Introduction

The Win4Lin Administration Guide: Win4Lin 9x and WTS covers some common and some not−so common administrative tasks. The sources for the information are varied and may include release notes, old documentation, or personal experience. With few exceptions, information that is documented in the installation guides, troubleshooting guides, or existing manuals will not be duplicated in this Win4Lin Administration Guide.

Win4Lin 9x, formerly called Win4Lin 5, and Win4Lin Terminal Server 3.0 are based on the same core technology which means many of the features available to Win4Lin 9x users are also available to WTS users. The generic term Win4Lin will be used to apply to both Win4Lin 9x and WTS products. When a difference between the two products needs to be highlighted, the more descriptive product names will be used: Win4Lin 9x and WTS.

The information contained in this guide is not necessarily difficult. However, this guide assumes that you are familiar with Linux commands, Linux file system navigation, and basic Linux concepts.

Your feedback is welcome. Good luck administering your Win4Lin installation.

Chapter 2 − Windows Session Administration

Graphical Administration Tools

Win4Lin offers a graphical post−install administration/configuration utility that lets you configure both global and individual user settings. Win4Lin 9x users run the command `win4linadmin`. WTS users run the command `wtsadmin`. Both root and non−root users can run `win4linadmin` or `wtsadmin`. Configuration access is granted based on the user running the configuration utility.

Both `win4linadmin` and `wtsadmin` offer the same features at the time of this writing (Win4Lin 9x Rev. 5.1.20 and WTS 3.0.20).

Root Configuration

When you run `win4linadmin` or `wtsadmin` as root, the initial panel has the following options:

- Product Update/Licensing/Registration
- Reload Windows Media
- User Administration
- Advanced Configuration (runs winsetup)

An overview of each option follows.

Product Update/Licensing/Registration
: This option allows you to update or remove the current Win4Lin installation. This option will also allow you to enter a paid license or bump license.

Reload Windows Media
: Will reinstall the Windows setup files that you loaded in Stage 2 of Win4Lin installation. This option is only useful for troubleshooting purposes or if you will be changing the version of Windows.

User Administration
: Selecting this option will allow you to manage an individual user's configuration settings. Select the desired user and click on OK. A second Post−Install Administration/Configuration window will be open for the selected user. The options presented are the same as the configuration options available when a user runs `win4linadmin` or `wtsadmin`. The user configuration options will be covered in the User Configuration section.

Advanced Configuration (runs winsetup)
: Opens the winsetup configuration window. Winsetup is well documented in the existing user manuals and can be ran directly with the command `winsetup`. Changes made in winsetup as root will affect all users.

User Configuration

When you run `win4linadmin` or `wtsadmin` as a normal user, the initial panel has the following options:

- Change Network Style
- Backup Windows Installation's C:
- Restore Windows Installation's C: from Backup
- Reinstall Windows
- Remove Windows Installation
- Advanced Configuration (runs winsetup)

An overview of each options follows.

Change Network Style
: Switch between winsock1, winsock2, and VNET networking types. The current network type will be greyed out and will not be selectable.

Backup Windows Installation's C
: This will backup up the $HOME/win directory to $HOME/win−BACKUP. The backup is not compressed.

Restore Windows Installation's C: from Backup
: Restores the $HOME/win−BACKUP directory to $HOME/win.

Reinstall Windows
: Reinstalls the Personal Windows session. Has the same affect as removing $HOME/win and then running `installwindows`. You will have to reinstall all applications. Documents saved to My Documents are not affected. They are stored in $HOME/mydata.

Remove Windows Installation
: Removes $HOME/win. Has the same affect as running the command `unloadwindowsCD`. Does not remove the Windows media files.

Advanced Configuration (runs winsetup)
: Opens the winsetup configuration window. Winsetup is well documented in the existing user manuals and can be ran directly with the command `winsetup`. Changes made in winsetup as root will affect all users.

Manage User Sessions

You can view and kill running Windows sessions for a particular user, or for all users. Win4Lin 9x users use the command `win4lin−sessions`. WTS users use the command `wts−sessions`. The usage options for each command is identical with the exception of the command. For illustration, the `wts−sessions` will be used.

Run `wts−sessions` with no arguments to get the command usage.

Usage

```
wts-sessions -l [-u <user>] [-a]
wts-sessions -k <pid>
```

-l	list running sessions
-k <pid>	kill running session <pid> (use -l first)
-u <user>	list only sessions for <user>
-a	list sessions for all users (default if running as root)

Examples:
List all sessions running on the system (root) or for the current user:

```
wts-sessions -l
```

List all sessions running for the user guest:

```
wts-sessions -lu guest
```

Kill session pid 23506:

```
wts-sessions -k 23506
```

Chap 3 − Windows Session Configuration

Relocate Windows C:\ Drive

The Linux directory $HOME/win is the Windows C:\ drive. You can use `winsetup` to move the location of the Windows C:\ drive, aka $HOME/win. For example, to move /home/username/win to /usr/username/win follow the steps below, as the user:

1. `mv /home/username/win /usr/username/win`
2. `winsetup`
3. `chmod 700 /usr/username/win`
4. Select Personal session configuration: win −> Drives & Filesystems
5. Highlight "C: Linux $HOME/win" and select Properties...
6. Edit the directory to reflect /usr/mbadger/win
7. Select Ok −> Save −> Exit
8. Start Windows

Caution: Pay attention to file permissions and file ownership when relocating the win directory. The above procedure assumes the user has permission to write to /usr/username.

Attach Physical and Virtual DOS Drives

Attach Physical DOS Drives

Attaching a physical DOS Drive is covered in the User Manuals for Win4Lin 9x and WTS under the heading "Accessing the Windows Partition of a Dual Boot System." An overview of mapping native DOS partitions is reiterated because of its close relationship to Adding virtual DOS Drives.

Win4Lin will automatically query for native DOS partitions each time the computer is rebooted and will assign a token in the form of dosc, dosd, etc.... To map a native DOS or Windows partition as a drive in the Win4Lin Window session, you must first make the token available. As root:

1. Run `winsetup`
2. Select System Wide Administration −> View/Create/Modify/Device Definitions −> DOS Drive
3. Select the appropriate "Token Name" from the available options. Tokens in the form of "dosc" and "dosd" represent native, physical DOS drives.
4. Set the "Usable by" option to "All"
5. Select Save −> Close −> Close −> Exit

Next map the drive into the Windows session. As the user who runs Win4Lin:

1. Run `winsetup`

2. Select Personal session configuration:win −> Drives & Filesystems −> Add...
3. From the Drive properties box, click "Dos Drive"
4. In the box labeled "Select one of the following DOS drives," highlight the drive that matches the description of the native DOS drive (i.e. "Native DOS c:")
5. Note the Drive letter in the top left corner − this is the drive letter you use to access the drive in the Win4Lin Windows session
6. Select OK −> Save −> Exit

Now start the Window session. You should be able to read and write files to the native DOS or Windows partition.

Warning: If you have your native DOS or Windows partitions mounted in Linux, then you will not be able to map the drives in Win4Lin using the method. You instead need to map the Linux mount point of the DOS or Windows partition as a drive in the Win4Lin Windows session. See the users guide for more information.

Note: You may attempt to run applications from the native DOS partition, but you assume all risk in doing so. This is not Win4Lin's design or intent.

Attach Virtual DOS Drives

When no native DOS partitions exist, Win4lin allows the creation of a virtual DOS partition. A virtual DOS drive may allow you to install copy−protected software that is otherwise uninstallable via the shared Linux/Windows files system. In addition, some applications will not properly access files stored on the Linux file system, and require low level disk IO access to the file, which the virtual DOS drive provides.

Mapping virtual DOS drives is similar to mapping a physical DOS drive, with some exceptions. The virtual DOS drive is created as a single Linux file. The single Linux file is formatted as a DOS filesystem that you can use to install applications and files onto. Win4Lin creates the default token "homevdrv" for this purpose.

The following example uses the mkvdisk command to create a 500MB virtual disk image in the user's home directory. Winsetup maps the DOS image as a Drive in Windows. As a non−root user:

1. `/opt/win4lin/mkvdisk −s vdrive.dsk 500000`
2. Run `winsetup`
3. Select Personal session configuration: win −> Drives & Filesystems −> Add...
4. From the Drive properties box, click "Dos Drive"
5. In the box labeled "Select one of the following DOS drives," highlight "Virtual DOS volume $HOME/vdrive.dsk"
6. Note the Drive letter in the top left corner − this is the drive letter you use to access the drive in the Win4Lin Windows session
7. Select OK −> Save −> Exit

Now start the Window session. You should be able to read and write files to the native DOS or Windows partition.

Note: When using the mkvdisk command, the size of the virtual DOS disk must be entered in KB. The disk space will be allocated for vdrive.dsk and will not be available to the rest of the Linux system, regardless of how much data is stored on the virtual disk. In other words, this procedure creates a virtual drive that only a Win4Lin Windows session can read and write to.

Customize Windows Drive Labels

In Windows, the drive labels are the Linux directory paths by default. For example, the C:\ drive is labeled as $HOME/win (C:) in when viewing My Computer. To change the Windows drive label, create a directory named `.labeldosdrive` in the root of the drive you wish to change the label. Inside the `.labeldosdrive` directory touch a file with the desired drive name. The following example will change the Windows default C:\ drive label from `$HOME/win (C:)` to `Badger (C:)`. From a Linux terminal as the non−root user:

- `cd win`
- `mkdir .labledosdrive`
- `cd .labeldosdrive`
- `touch Badger`
- Restart Windows

You can not use the Windows tools to change the drive label. Windows 95 and some versions of Windows 98 may restrict drive names to 11 characters.

Force Proper Windows Shutdown

Ensure users properly shut down Windows by selecting Start −> Shutdown... from Windows instead of closing the X window that Windows is displayed. With the environment variable MERGE_WINDOW_CLOSE enabled, clicking on the "X" to close Windows will have no affect.

Close all open Windows sessions before enabling MERGE_WINDOW_CLOSE. As rot:

1. edit /etc/default/merge
2. Append `MERGE_WINDOW_CLOSE="off"`
3. Save change to /etc/default/merge
4. Restart the Win4Lin Service: `/etc/rc.d/init.d/Win4Lin restart`

Identify Idle User Sessions

Identify idle user sessions by enabling the User Input Timestamp functionality. System administrators can define a filename to be touched, or created, each time a Win4Lin user inputs a keystroke or inputs a mouse click in Windows. Define the environment variable MERGE_XCRT_EVENT_FILE to begin tracking idle sessions.

Close all open Windows sessions before enabling MERGE_XCRT_EVENT_FILE. As root:

1. edit /etc/default/merge
2. Append `MERGE_XCRT_EVENT_FILE=".timestamp"`
3. Save change to /etc/default/merge
4. Restart the Win4Lin Service: `/etc/rc.d/init.d/Win4Lin restart`

When a user inputs a keystroke or inputs a mouse click in Windows, the file $HOME/win/.timestamp will be touched. Each time $HOME/win/.timestamp is touched, the file's modification time is updated. The file's modification date is the "timestamp" as the .timestamp file itself is empty. You can then create a script to monitor and end idle Windows sessions.

You may also want to review the Manage User Sessions topic in the Basic Administration chapter.

Chap 4 − Printer Configuration

Share Printer With Samba

Printing is configured to pass the print job from Windows to Linux by default during the Win4Lin setup and is covered in respective Win4Lin users guide. Direct attaching a parallel port to the Windows session is an option for parallel port printers and is also in the users guide. If neither of these options provide a suitable solution, you can use Linux to create a network printer that can be accessed from the Win4Lin Windows session.

This information is intended as a starting point to sharing a printer via Samba. You are responsible for knowing how to install and configure Samba and for knowing how to configure Windows networking. If you are unfamiliar with VNET, consult the Win4Lin user manual for your product. Win4Lin Home users will not be able to apply this procedure.

Samba's Official Howto at http://www.samba.org has a sample `smb.conf` file for anonymously sharing a printer. The direct link is http://us1.samba.org/samba/docs/man/Samba−HOWTO−Collection/StandAloneServer.html#AnonPtrSvr.

Once you have Samba configured and your Windows network configured, follow your printer manufacturers installation instructions for installing a networked printer.

Setup Multiple Printers in Win4Lin

Win4Lin setups up the "default" token that Windows uses to direct print jobs to the default Linux printer. You may have multiple printers configured in Linux that you want to access in Windows. Or perhaps you want to use a printer other than the default Linux printer. This can be accomplished by creating a new printer token in winsetup and creating a new printer configuration in Windows that uses the new token. The following steps provide an overview of configuring an additional printer in Win4Lin:

1. Determine the correct print command
2. Create new printer token
    - Run `winsetup` as root
    - Select System Wide Administration −> View/Create/Modify Device Definitions
    - Select New from the Printer panel
    - Enter a descriptive token name, i.e. LaserJet, and click OK
    - Type a description for the new printer in the Description box
    - Select Usable by "All"
    - Enter the print command from Step 1 − the command must be prefaced with exec
    - Select Save −> Close −> Close −> Exit
3. Start Windows
4. Select Start −> Settings −> Printer −> Add Printer
5. Setup a Local printer using the Apple Laserwriter print driver

6. Select the new token name from the list of available port
7. Give the Printer a name in Windows

The new printer will be an option when you print from your Windows applications.

Chap 5 − Run Windows Applications

Launch Windows Application From Linux Command Line

Win4Lin can be configured to automatically launch a Windows application from the command line. The `win` command will accept a Windows program executable as an argument. For example, to automatically launch Notepad when Windows starts, use the command:

```
win notepad.exe
```

Windows will start and Notepad will open.

If the command's executable is your Windows path, you can just specify the executable. Otherwise, you need to specify the complete Windows command path.

This will still display the Windows desktop. See the next section for help in launching a Windows application in place of the Windows desktop.

Configure Windows to Launch Single Application

Appwrapper.exe allows a Win4lin user to launch a single application instead of the traditional Windows desktop. To launch a single application in place of the Windows desktop, edit the c:\windows\system.ini and change the shell= line in Windows c:\windows\system.ini must be modified. Remember to back up the c:\windows\system.ini before modifying it.

1. Open c:\windows\system.ini in a Windows text editor, such as Notepad
2. Change `shell=Explorer.exe` to `shell=j:\mwin95\appwrapper.exe command` where `command` is the path required to launch the Windows application
3. Save the change to c:\windows\system.ini
4. Restart Windows to activate the change
5. Close the program using the File −> Close or File −> Exit menu choices to close the program and the Windows session.

For example, if you changed System.ini to reflect `shell=j:\mwin95\appwrapper.exe c:\Program Files\Accessories\wordpad.exe`, Wordpad would would be displayed each time Windows starts.

Even though the normal Windows desktop is not displayed, a full Windows session is still launched. The Windows session will terminate when the user exits the application. Users are bound by their existing Netraverse and Microsoft licenses.

To change change back to the normal Windows desktop, edit $HOME/win/windows/system.ini in a Linux text editor. Change the shell= line to `shell=Explorer.exe`. Save the change and restart Windows.

Launch Linux Commands From Windows

The Windows command `uexec` can be used to launch a linux command from Windows. Uexec has the following usage:

```
uexec command [arguments]
```

For example, the following command will open a Gnome terminal from the Windows command line.

```
uexec gnome-terminal
```

Just like in Linux, `uexec` will wait for the command to exit. Placing ""at the end of the command will disable the wait. However, uexec does not show Linux stdout or stderr, and does not return an error if there was a problem running the command.

If there is a script or command that will be needed on a regular basis, a Windows shortcut can be created. To create a shortcut in Windows, follow the steps below:

1. Right−click on your desktop and select New −> Shortcut
2. In the command line, enter: `uexec command [arguments]`
3. Click Next
4. Type a name for the shortcut
5. Click Finish
6. Right−click on the new shortcut and select Properties
7. Under Run: Choose minimized
8. To change the icon, select Change Icon... Select the desired icon from the list
9. Click OK

Administration

Uexec can be enabled and disabled in /etc/default/merge. In Win4Lin 9x, `uexec` is enabled by default. In WTS, `uexec` is disabled by default.

The following are valid /etc/default/merge environment variables to enable or disable uexec:

```
MERGE_ALLOW_UEXEC=on
MERGE_ALLOW_UEXEC=off
```

After you edit /etc/default/merge, restart the Win4Lin service. As root, enter the command:

```
/etc/rc.d/init.d/Win4Lin restart
```

Chap 6 − WTS Specific Configuration

Setup Dynamic Profile−Based Provisioning

Starting with Win4Lin Terminal Server 3.0.17, users have the ability to provision dynamic Windows sessions from the WTS master account. This feature allows the same user to run multiple Windows sessions. There are two modes available, disposable and transient.

Disposable Session
: A disposable session is created on−the−fly and is removed when the Windows session is exited. A disposable session does not follow the master profile's omit, copy, preserve, and registry profile rules.

Transient Session
: A transient session follows the master profile's copy, omit, preserve, and registry rules. A transient session retains the target directory when the Windows session is exited.

Each Windows session launched with dynamic−profile−based provisioning consumes one Microsoft Windows license and one WTS seat.

For a complete usage list, run `dynwin` without arguments to display the dynwin commands online help. General usage is as follows:

```
dynwin master−home [target−config] [−a
win−args]
```

<master−home> home directory of master profile to inherit

<target−config> optional transient configuration name

−a <win−args> optional arguments to pass to 'win'

Provision Dynamic Disposable Windows

Dynamic Profile Provisioning will allow the same user account to run multiple instances of Windows with ease. The following example will provision a disposable Windows session from the master profile, test.

```
dynwin /home/test
```

This windows session will be provisioned from the master profile, test and will be be provisioned to the $HOME/.dynwin−00001. When the Windows session is terminated, the $HOME/.dynwin−00001 directory is removed. Provisioning additional disposable Windows sessions will cause the temporary directory to increment its file name by a count of 1, i.e. $HOME/.dynwin−00002, $HOME/.dynwin−00003....

Provision Dynamic Transient Windows

The following example will provision a transient Windows session from the master profile, test. The target directory is mike.

```
dynwin /home/test mike
```

This Windows session will be provisioned from the master profile, test, and will be provisioned to $HOME/dynwin−mike. The $HOME/dynwin−mike directory will will remain after the Windows session is terminated.

Provision Dynamic Windows with Additional Options

The dynwin command also accepts valid `win` arguments. For a list of the valid arguments, type `win --help` in a terminal. To create a disposable Windows session with a descriptive title, use the following command:

```
dynwin /home/test −a −title Disposable−Windows
```

This Windows session will be provisioned from the master profile, test, and will have the title "Disposable Windows" at the top of the Windows session window. When passing win arguments to dynwin, the arguments must be preceded by the `−a` as shown in the example.

Run `win --help` for a list of additional options.

Troubleshoot Profile−Based Provisioning

If you use profile−based provisioning in Win4Lin Terminal Server, you may experience problems running certain applications from the inheriting user accounts. The application in question will likely work correctly when run from the master account, but will not run correctly from the user accounts. This typically indicates a Linux permission problem or an access violation on the part of the program you are trying to run.

To debug file access violations in a profile−based installation, use the MERGE_DEBUG_PBP environment variable, which is only available in Win4Lin Terminal Server 3.0.17 or higher. MERGE_DEBUG_PBP will log files that pose potential permissions problems, and may need to be added to the `copy−files` rule. However, not all files reported will need to be placed in the `copy−files` rule. As administrator, you will need to interpret the data.

Profile−based provisioning setup is discussed in Chapter 3 of the Win4Lin Terminal Server Installation and Operations Guide.

MERGE_DEBUG_PBP_USAGE

To capture the files, MERGE_DEBUG_PBP will need to be added to the inheriting user's environment. The files will be sent to the standard output, where they can either be displayed real time or piped to a file.

To display the file access violations for an inheriting user account in real time, log in as that user and run this command:

```
MERGE_DEBUG_PBP=on win
```

A Windows session will be started for the user. Attempt to use the application that is having problems. A list of files causing access violations will be printed to the standard output. This method will print duplicate file names.

To get the same information without the duplicate entries, pipe the output through `uniq` with this command:

```
MERGE_DEBUG_PBP=on win |uniq >/tmp/pbp.out
```

This method will avoid duplicate entries, but the /tmp/pbp.out file will be empty until the Windows session is closed.

Tunnel WTS Client via SSH

You may wish to use WTS Client and run your Windows session over a secure connection. The following example, will establish an ssh connection to the WTS server. The client ports 48602, 48603, and 48604 are forwarded to the corresponding ports on the WTS server.

```
ssh −L 48602:localhost:48602 −L 48603:localhost:48603 −L
48604:localhost:48604 hostname
```

Substitute the appropriate WTS server address for hostname in the preceding command. For additional ssh options, consult `man ssh`.

Open WTS client and connect to localhost to start the Windows session.

Appendix A: Environment Variables

How to Use Environment Variables

Win4Lin uses environment variables to configure and customize Windows. Environment variables can be applied in the following situations:

- Per session − declare and export the variable before running a Windows session
- Globally − declare variable in `/etc/default/merge`

When making changes to the global configuration file /etc/default/merge, you must issue the command `/etc/rc.d/init.d/Win4Lin restart` for the changes to take affect. Make sure all Windows sessions are closed.

Win4Lin Environment Variables

The following list is not a cumulative list of all possible environment variables. Many of the default environment variables are defined in `/etc/default/merge`.

MERGE_VNET_IFNAME
: Configures the network device the Windows session will use in VNET mode. The default value is "auto" which uses the first device listed in `/sbin/ifconfig −a`. Accepts the the device names, as listed in `/sbin/ifconfig −a` as valid arguments. Does not affect winsock networking modes.

MERGE_CD_DEV_NAME
: Specifies the default CD device to use. The default value is /dev/cdrom.

MERGE_WINDOW_CLOSE
: Forces users to properly shutdown Windows. By default, users are allowed to close the Windows session by clicking on the "X" in the window manager window. Use MERGE_WINDOW_CLOSE=off to force users to properly shutdown Windows.

MERGE_XCRT_EVENT_FILE
: Enables a way to identify and track idle user sessions. Specify a file name as argument. See Chapter 3 for more information.

MERGE_DEBUG_PBP
: Creates a log file to troubleshoot profile−based provisioning in WTS. See Chapter 6 for more information.

MERGE_ALLOW_UEXEC
: Enables users to launch a Linux command from the Windows command line. Defaults to off in WTS and on in W4L. Accepts either "on" or "off" as valid arguments. See Chapter 3 for more information.

MERGE_WINDOWS_TITLEBAR
: Specifies the name of the title in the Windows window. Accepts a text string as its argument. Enclose the title within quotes if it contains spaces. MERGE_WINDOWS_TITLEBAR is the same as using the "−title" option to the win command.

MERGE_XVNC_FONTHACK

Disables "font hack when X server is Xvnc. Set to "off" to try and correct improperly displayed fonts in winsetup.

MERGE_AUDIO_PLUGIN

Manually configure the audio plugin Win4Lin uses. The default value is "auto." To specify a specific sounder server, use the following plugins:

Sound Server	Audio Plugin
aRts	"/opt/win4lin/libartsdplugin.so"
OSS	"/opt/win4lin/libossplugin.so"
Esound	"/opt/win4lin/libesdplugin.so"
Nnaudio − remote sound server	"/opt/win4lin/libnnaudioplugin.so"

MERGE_HELP_BROWSER

Specifies a web browser to use when selecting reading the online help. Commented out of /etc/default/merge by default. Accepts a valid command as an argument. For example, MERGE_HELP_BROWSER="firefox" will open the online help topics in Firefox.