

*Tun* PLUS

**Quick Start Guide**

**Version 7.30**



TUN PLUS copyright 1989-1995

Dear Reseller,

This demonstration package of TUN PLUS version 7.30 has been created especially for you to be able to experiment with TUN PLUS in your own environment. The enclosed product represents a complete operating version of TUN PLUS that will be valid for 2 weeks counting from the date of installation. Please use the enclosed serial number and activation key.

This package may be installed on multiple computers so that you can test TUN PLUS' complete network printer sharing, but because the TCP/IP kernel is "aware" of duplicate copies, you will periodically receive error messages. These messages will disappear when you install final versions of TUN PLUS with valid serial numbers.

This Quick Start manual is designed to help users to get going fast, and offers step-by-step instructions for startup and basic use. Try these instructions first - *they may be all you need.*

TUN, TUN EMUL, TUN TCP, TUN KERNEL copyright ESKER  
Windows, Windows for Workgroups, MS-DOS, XENIX copyright Microsoft Corporation  
UNIX is a registered trademark of AT&T Bell Laboratories  
SCO UNIX, SCO XENIX copyright The Santa Cruz Operation  
ODI copyright Novell  
vt52, vt320 copyright Digital Equipment Corporation  
Ethernet is a registered trademark of XEROX  
NFS copyright SUN Microsystems

## **TABLE OF CONTENTS**

### ***PREFACE***

**GUIDELINES FOR CHOOSING TUN TCP MODULES**

### ***TUN TCP USING TCP/IP FOR DOS***

**Installing TCP/IP for DOS**

**Configuring TSR-based TCP/IP**

**Sharing UNIX disks using NFS (TSR)**

**Sharing UNIX printers (rxprint TSR)**

**Sharing PC printers (vxprint TSR)**

### ***TUN TCP USING TCP/IP FOR WINDOWS***

**Installing TCP/IP for Windows**

**Configuring DLL-based TCP/IP**

**Sharing UNIX disks using NFS under Windows (DLL)**

**Sharing UNIX printers (LPR DLL)**

**Sharing PC printers (LPD DLL)**

### ***TUN TCP WINDOWS APPLICATIONS***

**Overview**

### ***TUN EMUL TERMINAL EMULATION***

**Installing TUN EMUL**

**Using TUN EMUL under Windows**

**Using TUN EMUL under DOS**

### ***APPENDICES***

**Installing the E-Mail server**

**TUNNET.DRV vs TUNTCP.DRV**

**Netware coexistence**

**Coexistence with Windows for Workgroups**

**General note on program syntax**

***Conclusion***

## PREFACE

TUN PLUS is a complete terminal emulation and TCP/IP network software package for PCs running DOS and Microsoft Windows. It is composed of two independent products, **TUN EMUL** and **TUN TCP**, each of which contains several different modules. The most important distinction to make, before you proceed with installation, lies within TUN TCP.

TUN PLUS includes TCP/IP protocol stacks as well as a complete set of complementary applications for establishing network connections between PCs and remote servers, such as UNIX and AS400. TUN PLUS delivers the following services:

- ◆ Multinet TCP/IP stacks for Ethernet/Token Ring, SLIP and PPP
- ◆ Simple **telnet** and **advanced** terminal emulation
- ◆ **FTP** file transfer
- ◆ **NFS** file sharing
- ◆ PC/UNIX **Printer sharing**
- ◆ PC **backup**

The TUN TCP software package is divided into three parts:

	<b>Package</b>	<b>Description</b>
1	<b>TCP/IP for DOS</b>	TCP/IP protocol stacks and resource sharing modules in the form of DOS TSR (Terminate and Stay Resident programs).  Includes DOS versions of file transfer, telnet, PC backup programs, remote shell, NFS and more.
2	<b>TCP/IP for Windows</b>	TCP/IP protocol stacks and resource sharing modules as VxD, 100% DLL and Windows executables.
3	<b>Windows Applications</b>	Telnet, remote shell, e-Mail, PC backup, FTP, etc. as Windows applications that run over BOTH TSR and DLL TCP/IP kernels. Installation is automatic following either TCP/IP stack.

Because of numerous conflicts, it is not possible to run the TSR *and* DLL TCP/IP kernels at the same time. Here are the different combinations possible with the TUN TCP modules:

1. TCP/IP for DOS alone (1)
2. TCP/IP for DOS and the MS Windows Applications (1+3)
3. TCP/IP for Windows and the MS Windows Applications (2+3)

The TUN TCP products you choose to install therefore vary depending on the configuration and intended use of your PC.

	<b>Advantages</b>	<b>Disadvantages</b>
<b>TCP/IP for DOS</b>	Provides transport services under both DOS and Windows	Uses between 60 Kb and 120 kb of conventional memory, depending on users' needs
<b>TCP/IP for Windows</b>	Uses much less conventional memory - around 10 Kb	Works <i>only</i> under Windows

---

## GUIDELINES FOR CHOOSING TUN TCP MODULES

---

1. Users without MS Windows only need to install package #1
2. Users with DOS *and* Windows most likely need packages #1 and #3
3. Users who run Windows exclusively can get the best performance by installing packages #2 and #3

**TUN TCP**  
**USING TCP/IP**  
**FOR DOS**

## Installing TCP/IP for DOS

This chapter describes how to install and use the Terminate-and-Stay-Resident (TSR) version of ESKER's TCP/IP stack, which of course may be used under both DOS and Windows.

### Navigating within TUN TCP menus

If you load a **mouse driver**, you may use a mouse to move through the menus and to select options.

Move through the fields using the mouse, **arrow keys**, or the <Tab> key, pressing the space bar to place an "X" where necessary.

### Installing the product

1. Remove previous versions of TUN from AUTOEXEC.BAT and CONFIG.SYS
2. Insert TUN TCP **TCP/IP for DOS disk 1** in a drive, and type: `INSTALL` ↵
4. Choose the floppy drive containing the program and the destination directory if desired. Press ↵ on OK to continue
5. Enter your serial number and activation key. Press ↵ on OK to begin installing the program
6. You will be prompted to insert **TCP/IP for DOS disk 1** and **disk 2**. After inserting them press ↵ on OK
7. Once the files have been installed into the appropriate directory, the program will be branded with your serial number and you will be notified that the installation is complete. Press ↵ on OK and you will return to DOS

**Note:** If you chose to install TUN TCP's Windows Applications, the installation program will start Windows, install the new programs and then create a program group called TUN TCP. These applications are described further in this guide.

This procedure will also install TUNTCP.DRV, a network device driver that interfaces with the **resident** TCP/IP kernel in order to provide true network configuration services for TCP/IP (NFS and printer sharing) under Windows.

## Configuring TSR-based TCP/IP

This section describes the basics for setting up your LAN card, TCP/IP kernel, and UNIX host in order to be able to use TUN TCP effectively:

1. From DOS type **TUNTCP** ↵
2. Select **TCP/IP** ⇒ **TCP/IP** ⇒ **Startup Parameters** ↵

**Note:** You may use the <tab>, <enter> or arrow keys to move the cursor around through the fields, pressing the space bar to place an "x" where appropriate, and scrolling through multiple options with the left/right arrow keys.

Navigation is also possible using a mouse, if you loaded the driver before entering the program.

3. Select one of the five interface types (Packet driver, NDIS, ODI, NDIS with Coexistence, Windows for Workgroups) that you want to use. We strongly recommend that you try **Packet drivers** first, as they are easiest to configure. More details are given on coexistence with other PC networks (Novell, Lan Manager and Windows for Workgroups) at the end of this guide.

You will need to enter the following information, depending on your Network card:

- ◆ Board Type (press space to toggle through selections)
- ◆ Interrupt Vector (IRQ) (enter the number)
- ◆ Base I/O Address (press space to toggle through selections)
- ◆ Base Memory Address (press space to toggle through selections)
- ◆ IP Address (i.e. 89.85.37.10)

- ◆ Local Hostname (i.e. Sarah\_PC)
- ◆ User Name (i.e. Sarah)
- ◆ Press F2 to accept settings. Press ← on OK.



3. Select **Tables** and press ←. Select **Host Table** by pressing ←. Go to a new line and enter the IP address of the server you wish to connect to, i.e.:

```
89.85.37.20      host_1
```

Press F2 to accept settings.

4. Press ESC ⇒ ESC ⇒ Q to exit TUNTCP

### Declare the user/workstation on the UNIX host

1. Login to UNIX host as root to set up the PC
2. Edit **/etc/hosts** and add the IP address and host name of the PC, i.e.:

```
89.85.37.10      sarah_pc
```

3. Ensure there is a login for the user, for example: sarah
4. Create a file **.rhosts** in the user's home directory that contains the host name of the PC (without IP address) as set up in the Host Name field in TUN TCP, for example:

```
sarah_pc
```

### Test the connection

1. Reboot your PC and run AUTOTCP.BAT.
2. Check that your Ethernet card is correctly recognized (not FF:FF:FF:FF or 00:00:00:00)
3. Use PING to test your TCP/IP connection to the server :

```
ping host_1 ←
```

The system should display **Host responding** if the connection is correct. Some types of network cards need to be initialized before they connect, and therefore require PING to be run twice to test connections.

4. If your host is not responding, check your cabling and make sure that TCP/IP is running correctly on the host (try a telnet from the host to itself).

## Sharing UNIX disks using NFS (TSR)

To configure TUN TCP to allow PCs to mount remote directories as virtual DOS drives:

1. Ensure TUN TCP is installed on the PC and NFS is installed on the UNIX host machine
2. Login to UNIX host as root to configure the NFS server
3. Edit **/etc/exports** and add line for remote file system, for example: `/u/sarah`. This will add the filesystem for future restarts of NFS
4. To update the current NFS process, from the UNIX prompt use the **exportfs** command, for example:

```
exportfs /u/sarah ↵
```

5. Type TUNTCP ↵ on the PC to configure the remote directories
6. Move cursor to **Network File System** and press ↵
7. Press ↵ on **NFS**
8. Press ↵ on **TSR Startup Parameters**
9. Press M for 1 virtual drive and press F2
10. Press ↵ on OK
11. Move cursor to **Setup** and press ↵ on **Filesystems**
12. Enter Filesystem information. An example may be:

```
Filesystem Name      SARAH
Drive                D
Host Name            host_1
Remote File System   /u/sarah
User Name            sarah
```

Press F2 to accept settings

15. Press ESC ESC Q to exit TUNTCP
16. Add the line `CALL AUTONFS` to `AUTOEXEC.BAT` after the line that has `CALL AUTOTCP` if you would like NFS to load automatically at startup.
17. Reboot your PC.

## To mount a remote directory from DOS

1. After loading AUTOTCP.BAT and AUTONFS.BAT, at the DOS prompt type :  

```
MOUNT SARAH
```
2. You will be prompted for the UNIX password
3. Enter the password
4. You will now be able to access the remote directory by typing D:

## To mount a remote directory from within WINDOWS

1. Select Admin from TUN TCP Windows
2. Click on NFS and select MOUNT
3. You will be prompted to enter File Resource
4. Enter File Resource (SARAH) and press TAB key
5. Enter local disk and press TAB key
6. Enter Password and click on Connect
7. On another window, you will see current connections

## Sharing UNIX printers (rxprint TSR)

To configure TUN TCP to allow PC applications to print to a printer attached to UNIX hosts:

1. Make sure that TUN TCP is installed
2. Login to the UNIX host as root and create a printer device for the printer you wish DOS applications to print to, i.e.: /dev/lp0. Use a dumb interface for this device.

Note: You may also choose to use any existing "lp" command.

3. From DOS type TUNTCP ↵
4. Move cursor to **Printer Sharing** and press ↵.
5. Press ↵ on **Printers**
6. Press ↵ on **Startup Parameters**
7. To make LPT2: print to a UNIX printer, use the <tab> key or press <enter> to move the cursor down to LPT2:, move the cursor to **Remote** and press the space bar to select it. An example of the settings may be:

```
Remote Host Name:  host_1
UNIX Command:     lp -dtun
Timeout:          20
```

Press F2 to accept settings. Press ↵ on OK

8. Press Q Q to exit TUNTCP
9. Edit AUTOEXEC.BAT and add CALL AUTOLPT after the line that has CALL AUTOTCP
10. Reboot your PC
11. If you now print to LPT2: from your DOS applications, the jobs will be printed on the configured printer attached to the UNIX host.

## Sharing PC printers (vxprint TSR)

To configure TUN TCP to allow UNIX applications to print to a printer attached to a DOS PC:

1. Make sure that TUN TCP is installed
2. Login to the UNIX host as root and **create** a printer device for the DOS printer you wish to print to, for example: device for sarah: /dev/null. Set the device to /dev/null and use a dumb interface.
3. Edit the interface file (created in step 2) for this device, for example:

On Xenix: /usr/spool/lp/interface/sarah

On UNIX: /usr/spool/lp/admins/lp/interfaces/sarah

Find the line that prints the file, eg: `cat "$file" 2>&1` and add `"| rsh pc_name lptx"` to the end, where `rsh` may be `RCMD` on some systems such as SCO UNIX, `pc_name` is the name of the PC and `LPT` is the port name on the PC for this printer. An example is:

```
cat "$file" 2>&1 | rsh sarah lpt1
```

4. From DOS type TUNTCP ↵
5. Move cursor to **Printer Sharing** and press ↵.
6. Select **Printers** ⇒ **Startup Parameters**
7. To set **LPT1**: so UNIX users can print to it, cursor to Public and press space to select it. Press ↵ on CR/LF Conversion. Press F2 to accept settings. Press ↵ on OK
8. Press Q Q to exit TUNTCP
9. Edit AUTOEXEC.BAT and add CALL AUTOLPT after the line that has CALL AUTOTCP
10. Reboot your PC
11. As a UNIX user, if you now print to the configured printer device, in this example sarah, it will print to the printer attached to the DOS PC.

Note: You may also test a public printer by issuing a simple "cat" command, such as shown below :

```
cat /etc/hosts | rcmd pc_name lpt1
```

**TUN TCP**

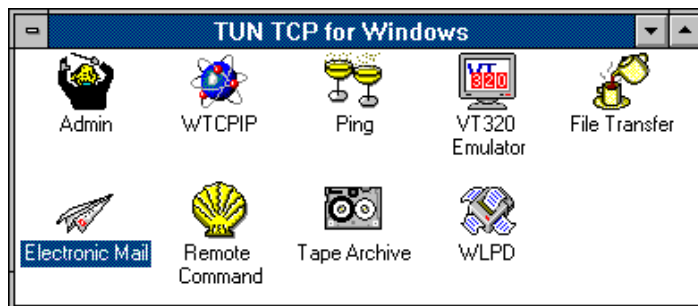
**USING TCP/IP  
FOR WINDOWS**

## Installing TCP/IP for Windows

1. Insert TUN TCP TCP/IP for Windows disk 1 disk into a floppy drive and type: WINSTALL ↵

Note: You may also run WINSTALL.EXE from within Windows using **File** ⇒ **Run**.

2. Enter your **serial number** and **activation key** ↵
3. Choose the drive and destination directory where you wish TUN TCP to be installed ↵
4. You will be prompted to select all or part of TUN TCP Windows applications package ↵
5. You will have a choice to continue or re-select the previous option ↵
6. If there is a network driver already installed, you will have a choice to replace it ↵
7. You will be prompted to insert the second disk (TUN TCP TCP/IP for Windows disk 2) ↵
8. The following group will be created:



## Configuring DLL-based TCP/IP

### Setting up TCP/IP

This section describes the basics for setting up your LAN card and a TCP/IP interface for a LAN connection under Windows.

1. Click on **Admin**, and then on the TCP/IP button
2. In the kernel configuration menu, click on **New...** to add an interface.

You will at least need to fill in the following elements:

- ◆ A name for the interface (for reference only)
- ◆ Protocol type (in this case choose Ethernet)
- ◆ IP Address (i.e. 89.85.37.10)
- ◆ Local Hostname (i.e. sarah\_pc)

3. Click on **OK** to save your changes to a file called WTUNTCP.INI
4. Click on **Options** ⇒ **Hosts** in order to update the **Host table**. Click on **Add** and enter the IP addresses and alias names of the PC and of other hosts on the network, for example:

```
89.85.37.10 sarah_pc
89.85.37.11 server_names
```

5. Click on OK to save your changes to the Host table.

### Setting up the LAN card

1. Click on the **Interface name** and then on the **Drivers** button to configure your network card.
2. Select one of the available interface types (Packet driver, NDIS, ODI, NDIS with coexistence, or Windows for Workgroups).
3. Fill in the name and other address parameters as necessary
4. Click on OK to save your changes. You will need to run these drivers before entering Windows (see next section).

## Declaring the user/workstation on the UNIX host

1. Login to UNIX host as root to set up the PC
2. Edit `/etc/hosts` and add the IP address and host name of the PC, i.e.:
 

```
89.85.37.10      sarah_pc
```
3. Make sure a login exists for the user whose name will be used for gaining access to the server (for **printer redirection** and other TUN programs such as **RSH** and **TAR**), for example: *sarah*
4. Create a file `.rhosts` in that user's home directory that contains the host name of the PC (without IP address) as set up in the Host Name field in TUN TCP, for example:

```
sarah_pc
```

## Starting TCP/IP and testing network connections

1. Quit Windows and reboot your PC.
2. Run AUTOTCP.BAT before re-entering Windows. This will load the appropriate packet drivers.
3. Make sure that your LAN card was recognized by the drivers (and not as FF:FF:FF:FF or 00:00:00:00)
4. The kernel will start automatically as soon as you run an application that requires TCP/IP. You may also load the kernel "manually" by selecting the interface you want in the Admin program, and then click on the down arrow ↓.
5. Run **WPING** (whose icon is in the TUN TCP group), select the server you would like to test and press **Start**. If the kernel is running correctly, the number of data packets sent should equal the number of packets received, and that value for "**percent received**" should be 100. If you do not get this result, it is most likely that the card is not loaded correctly or that there is a cabling problem.



## Sharing UNIX disks using NFS under Windows (DLL)

### Defining remote filesystems

1. Click on the **New** (or **Setup...**) button to begin configuration.
2. Here is an example of how you could fill in these fields:

Remote file system	
<b>Name</b>	<input style="width: 100%;" type="text" value="SARAH'S DISK"/>
<b>Remote host</b>	<input style="width: 90%;" type="text" value="risc"/> <input style="width: 10%; text-align: center;" type="button" value="↓"/>
<b>Remote path</b>	<input style="width: 90%;" type="text" value="/usr/sarah"/> <input style="width: 10%; text-align: center;" type="button" value="↓"/>
<b>User name</b>	<input style="width: 100%;" type="text" value="sarah"/>
<b>Default local drive</b>	<input style="width: 10%; text-align: center;" type="text" value="D"/> <input style="width: 10%; text-align: center;" type="button" value="↓"/>

3. To view available NFS filesystems on the host, click on the arrow next to the **Remote Path** field. This will send a request to the host, and will display the paths of the currently-exported filesystems on your host.
4. To save a configuration, click on **OK**, which also returns you to the previous menu.

### Mounting NFS filesystems

There are two ways to mount configured filesystems:

1. Using the Down arrow ↓
2. Using the Windows File Manager, **Drive** ⇒ **Connect network drive** (if TUN TCP is installed as the secondnet.driv, then click on **Other**).

Sharing UNIX printers (LPR DLL)
---------------------------------

**Setting up print redirection**

1. Click on **LPR** ⇒ **New** in the TUN TCP Supervisor control panel or use the Windows Print manager, **Printers** ⇒ **Network connections** to open the printer configuration menu.
2. Fill in the following fields:

<b>Name</b>	Invent a name for referring to your redirected printer within Windows applications
<b>Remote host</b>	Enter the name or IP address of the server that you wish to print to. Pressing the button to the right of the field opens the current Host table.
<b>Protocol</b>	Use these check-boxes to specify the protocol that will be used by the TCP connection to transfer print jobs to the remote machine. There are three options:
<b>rsh</b>	Runs on any UNIX host with TCP/IP, but requires prior configuration
<b>rexec</b>	Runs on any UNIX host with TCP/IP, does not require prior configuration, but asks for a password every time the user prints.
<b>lpd</b>	Not available on all UNIX hosts, but does not require prior configuration

When possible, it is best to use the LPD protocol (UNIX versions such as AIX, SunOs, Solaris, HP-UX). UNIX configuration for using RSH is described further in this section.

**Remote command** You only need to fill in this field if you are using RSH or REXEC. Enter the name of the UNIX print (or **cat**) command that will handle the LPT port redirection. The most frequently-used type of command is **lp**, but you may also use others (i.e. `cat > /tmp/file_name`).

**UNIX printer** Fill in this field with the name of a valid UNIX print queue when using **LPD** to print. The UNIX command **lpstat** will give you a list of the currently-available print queues.

**User name** Enter the name of the UNIX user whose access rights will be used during printing. This account name must have a **.rhosts** file with the name of the PC in it, as

**Local port** This field shows the name of the PC's parallel port that you wish to redirect: LPT1-LPT9 may be used.

**Priority** You may select **High**, **Med** or **Low** to affect the rate at which print jobs will be sent from the PC.

3. After you have filled in the relevant fields described above (for up to three different printers), click on **Connect**. LPT ports that have been redirected are listed in the lower portion of the LPR window. You may disconnect active printers with the **Disconnect** option.

### UNIX configuration for RSH printing

In order for a PC to use RSH as the protocol for print redirection, access rights need to be obtained on the UNIX host. For this, users must be declared in the **.rhosts** and **/etc/hosts** files. This procedure is described in the section **Declaring the user/workstation on the UNIX host** earlier in this guide.

Note: Remote host declaration is not necessary when using LPD or REXEC protocols. In that case, all you need to know is the name of the UNIX command that will be used for printing.

## Sharing PC printers (LPD DLL)

### Setting up printer sharing

1. Click on the **LPD** button in the TUN TCP Supervisor control panel (or by clicking directly on the **LPD** program icon in the TUN TCP group and using **Setup**).
2. Add printers and assign them to print queues, by clicking on the **New** button.
3. Fill in the following fields
 

<b>Public Printer</b>	This field contains a logical name for the public printer. The logical name will be used by the LPD or RSH client in order to print to the right printer.
<b>Print Man. queue</b>	Enter the real name of an existing Windows print queue in this field. WLPD really shares Windows print queues and not parallel ports.
<b>CR/LF Conversion</b>	If this option is used, all LF characters received from a remote machine will be converted into CR+LF.
<b>ISO8859 Conversion</b>	This field indicates whether or not WLPD should apply an ISO8859 character conversion filter.
<b>Timeout</b>	Expressed in seconds, the value given in the <b>Timeout</b> field determines the length of time after which VXPRINT will consider that a print job is finished. If WLPD has not received characters within the Timeout period, it will close its connection with the server.
4. After setting-up public printers, click on **OK** to save your changes.
5. Start WLPD clicking on the WLPD icon in the TUN TCP group.

### Using public printers with RSH

WLPD is an LPD/RSH server. For this reason, using a public printer from a UNIX machine requires access to an LPD or RSH client command.

### Using RSH from a UNIX host

Usually the RSH client command is called **rsh**, but on some systems it is called **rcmd** (SCO UNIX), or **remsh** (Motorola). This section describes the mechanism for sending print jobs to public printers..

To print a file from a UNIX machine on a PC printer without using a spooler, make sure LPD is running on the PC, then execute the following command on the UNIX host:

```
cat file | rsh pc_name printer1
```

In this command:

<b>cat file  </b>	types a file and "pipes" the characters into another UNIX command
<b>file</b>	name of a file (i.e. <b>/etc/passwd</b> ).
<b>rsh</b>	name of the RSH client command (possibly replaced by <b>rcmd</b> or <b>remsh</b> ).
<b>pc_name</b>	is the name or IP address of the PC with the public printer (running VXPRINT).
<b>printer1</b>	logical name of the public printer (assigned during setup).

This type of command may be included in a spooler **model**, as discussed in the next section.

### Remote query of public printers

In order to find out which parallel ports have been made public on a PC, execute the following command:

```
rsh pc_name lpts
```

- rsh** is the name of the RSH client command (possibly replaced by **rcmd** or **remsh**).
- pc\_name** is the name or IP address of the PC with the public printer (running WLPD).
- lpts** is a command recognized by WLPD that returns the list of available print queues.

The message displayed by this command might be:

```
printer1 : HP LaserJet 4/4M
printer2 : Postscript
```

## Creating a print queue on SCO UNIX

Here is an example of a printer model that has been tested on an SCO UNIX system:

```
# lp model for WLPD on TUN TCP
while true
do
    A=`echo "\007" | rcmd pc printer1`
    if echo $A | grep "001"
    then
        break
    fi
    sleep 5
    echo "Printer Problem PC ${A}" >/dev/console
done
(
copies=$4
echo "\014\c"
shift; shift; shift; shift; shift
files="$*"
i=1
while [ $i -le $copies ]
do
    for file in $files
    do
        cat $file
        echo "\014\c"
    done
    i=`expr $i + 1`
done
) | rcmd pc printer1
```

**Note:** In the **rcmd** command on the last line, **pc** refers to the name or IP address of the PC with WLPD, and **printer1** is the printer's logical name as defined during setup.

## Enabling the new print queue (SCO UNIX)

The easiest way to add a print queue using the above model to an SCO UNIX system is to use the commands given below rather than using **system administration** menus (such as **sysadmsh**):

```
/usr/lib/lpshut
/usr/lib/lpadmin -ptun -mmodel -v/dev/null
/usr/lib/lpsched
/usr/lib/accept tun
enable tun
```

In this example:

- tun** name of the print queue to create
- model** name of the file containing the model shown above
- /dev/null** device driver assigned to the printer port

## Using public printers with LPD

On UNIX systems, remote printing to LPD servers is a function in the print sub-system. Therefore, in order to use LPD, you must first create a print spooler on the UNIX machine.

The procedure for declaring printers on UNIX varies a widely from system to system, but this section gives a brief overview of some of the most popular.

Note: Not all UNIX systems include the LPD protocol (i.e. SCO, Motorola). In this case, use RSH for printing to public printers.

## LPD on SUN

On SUN servers (SunOs, Solaris 1.1) you may declare remote printers by inserting the following lines in the file **/etc/printcap**:

```
wlpd|wlpd line printer:\
    lp=:rm=pc_name:rp=printer1:sd=/tmp/wlpd:mx=0:
```

In this example:

- rm**            "**remote**" specifies the name of the PC with the public printer. The PC "pc\_name" must be in the **/etc/hosts** file on the UNIX machine.
- rp**            "**remote printer**" indicates the public printer's **logical name**
- sd**            "**spool directory**" indicates the name of the UNIX directory that will serve as a print queue. You will need to create this directory using a **mkdir** command.

After inserting the above lines in **/etc/printcap**, you may print to a public printer using the command:

```
lpr -Pwlpd /etc/passwd
```

- wlpd**           name of the print queue as described above

## LPD on IBM RS/6000 (AIX)

To declare a remote printer on an IBM RS/6000 (AIX) machine, insert the following lines in **/etc/qconfig**:

```
wlpd:
    device=wlpd
    up = TRUE
    host = pc_name
    s_statfilter = /usr/lpd/aixshort
    l_statfilter = /usr/lpd/aixlong
    rq = printer1
wlpd:
    backend = /usr/lpd/rembak
```

- host**           specifies the name of the PC with the public printer. The PC "pc\_name" must be entered in the **/etc/hosts** file on the UNIX machine.

**rq** "remote queue" indicates the **logical name** of the public printer (as configured on the PC)

After inserting the above lines in /etc/qconfig, you will be able to print using the command:

```
lp -Pwlpd /etc/passwd
```

**wlpd** is the name of the UNIX print queue as described above

### LPD on HP 9000 (HP-UX)

On HP 9000 (HP-UX) machines, it is best to create remote printers using the system administration program **sam**. Run **sam** and follow these steps:

1. Printers and Plotters ⇨
2. Printers and Plotters ⇨ Actions
3. Add Remote Printer/Plotter
4. Fill in the fields with the name of the PC and with the public printer and the logical name of the printer
5. After declaring a remote printer, you will be able to print to it using the command:

```
lp -dwlpd /etc/passwd
```

**wlpd** is the name of the UNIX print queue as described above

### Using public printers directly from PCs

With TUN TCP, PCs may print directly to public printers without going through a UNIX system. In other words, the print redirection client (LPR) may declare another PC running LPD as the remote host. To configure direct PC-to-PC printing using LPR, follow these steps:

1. Click on the LPR button in the TUN TCP Supervisor.
2. Fill in the fields just as you would for redirecting an LPT port to a UNIX host:

**Server** the name of the **PC** with the public printer

**Unix printer** Enter the **logical name** of the public printer in this field. You may use either RSH or LPD.

3. Click on **Connect**
4. Print to the redirected port as if it were a normal local LPT port.








**TUN TCP**

**WINDOWS APPLICATIONS**

## Overview

The TUN TCP Windows Applications are installed automatically at the end of either TCP/IP for DOS or TCP/IP for Windows.

	Standard protocol	Description
 File transfer	FTP	A graphical interface for easily transferring files between PCs and UNIX servers. Allows multiple sessions on different servers at the same time for transferring files between servers.
 tnvt320	Telnet	Terminal emulation in DEC vt320 mode. Allows multiple simultaneous sessions and customization of terminal display.
 Remote shell	RSH, REXEC	Remote command execution. May be used to execute any non-interactive UNIX command (lptstat, who, finger, ps -ef, etc.). REXEC mode prompts the user to enter a password. <i>RSH mode requires user/workstation declaration as described earlier.</i>  Includes customizable button bar, and may be run in "button mode" (WRSH -b) for greater convenience.
 Tape archive	TAR	PC backup onto UNIX disks and tape drives. Uses standard UNIX TAR format. Allows creation of backup and restore procedures and "point and click" file selection.
 E-mail	SMTP, POP3, UUENCODE	E-mail over TCP/IP, using a UNIX host as a central Post Office (mail server). See Appendices for instructions on installing the POP3 server on your UNIX host.  Provides binary file attachment using UUENCODE and automatic notification when new messages are received. Completely compatible with standard UNIX mail.

# **TUN EMUL TERMINAL EMULATION**

## Installing TUN EMUL

The installation procedure for TUN EMUL is started under DOS, and offers users the option to automatically configure the Windows portion:

1. Insert TUN EMUL Disk 1/2 disk in drive
2. A: ↵
3. INSTALL ↵
4. Choose the floppy drive containing the program and the destination directory if desired. Press ↵ on OK to continue
5. Enter your serial number and activation key. Press ↵ on OK to begin installing the program
6. You will be prompted to insert TUN EMUL Prog 1 disk. After inserting this press ↵ on OK
7. Once the files have been installed, the program will be branded with your serial number and you will be notified that DOS installation is complete
8. You will then be prompted to continue with TUN EMUL configuration under Windows. Select "Yes" in order to have the installation program install the proper icons and fonts under MS Windows.
9. When this is done, press ↵ on OK and you will return to a normal Windows session
10. Check AUTOEXEC.BAT and ensure C:\TUNEMUL is in the PATH statement

## Using TUN EMUL under Windows

### Opening a terminal emulation session

1. Start TUN EMUL by clicking on the Emulation icon
2. Select **File** ⇒ **New session**
3. Select the communication protocol you would like to use (RS-232, TCP/IP, etc.).
4. If you select TCP/IP, make sure that the kernel is installed and running (with a successful PING)
5. Enter the **Host name** and **terminal type** (and any other communications parameters if you are using serial lines or modems)
6. You should then receive a Login prompt from the host

## Customizing and saving your current session(s)

After you have opened one or more emulation sessions as described above, you can save the connection and host settings (in a .CFG file) by clicking on **File** ⇒ **Save configuration**.

After you have saved a terminal session configuration (.CFG file), you no longer need to start EMULWIN, select **File** ⇒ **Open configuration**, etc. All you need to do is create a Windows icon with the following command line:

```
C:\TUNEMUL\EMULWIN config_name.CFG
```

This will automatically open the desired session, without user intervention.

## Customizing the terminal display

You may customize the display, select character fonts, and then store all the current settings in **Display settings** files (.CTX). Select **Parameters** ⇒ **Display settings** in an open emulation session, and you will have access to all the different elements that may be customized.

Note: Best results are obtained by using **Dynamic Sizing** and the scaleable font delivered with TUN PLUS called **SystemPC**.

## Using TUN EMUL under DOS

1. Ensure you have a valid connection to your UNIX server with TCP/IP or RS-232
2. From DOS type TUNEMUL ↵
3. Cursor to Setup and press ↵. Select Configuration by pressing ↵. Select the default CONF1.CFG by pressing ↵



# **APPENDICES**

## Installing the E-Mail server

POP is the standard Post Office Protocol for UNIX systems. However, UNIX machines do not usually have the POP server necessary for the UNIX Mail interface provided in TUN TCP. In order to provide the mail service for all UNIX systems, TUN TCP includes a freeware version of this server from the University of California at Berkeley.

**Note:** If your UNIX server already has POP3 installed, you do not need to install this version of the program.

The POP3 server delivered with TUN TCP is in a file called POP3.TAR (in TAR format for UNIX). This file includes:

- ♦ the C source code for the POP server, modified by ESKER to be compatible with UNIX System V.
- ♦ a README file
- ♦ an installation program (by ESKER)
- ♦ several compiled version of the POP server (for SCO, AIX, HP, and SUN).

Follow these steps to install the POP server on a UNIX host:

1. Transfer the **POP3.TAR** file from the DOS directory containing TUN TCP into the /tmp directory on the UNIX machine (using the WFTP file transfer program in TUN TCP, for example).
2. Extract the files from POP3.TAR using the following command:

```
tar xvf pop3.tar
```

3. Run the installation program **pop3.install**, which will update the /etc/services and /etc/inetd.conf files on your system.

**Note:** If your UNIX system is not listed among the machines for which there is already a compiled version of POP3, the installation procedure contains an option to compile the source code for your particular machine.

## TUNNET.DRV vs TUNTCP.DRV

As we have discussed, TUN TCP provides TCP/IP network services through Windows with both versions of its TCP/IP.

Using a Network Device Driver (which is given as an option during installation), TUN TCP is able to mount NFS drives and change printer configurations directly from within Windows.

One very important to detail to know is that because TUN TCP offers TSR and DLL TCP/IP stacks, it has to provide DIFFERENT network device drivers, one specific to each type of kernel.

TCP/IP stack	Network Device Driver (in [boot] section of System.ini)	Other declarations
TUNTCPIP.DLL	network.drv=c:\wtuntcp\ tunnet.drv - or - secondnet.drv=c:\ wtuntcp\tunnet.drv	[386enh] device=c:\wtuntcp\ redir.386
Ethtcp.exe, Toktcp.exe	network.drv=c:\tuntcp\ tuntcp.drv - or - secondnet.drv=c:\tuntcp\ tuntcp.drv	none



## Netware coexistence

TCP/IP is a versatile protocol that is able to function in many different network environments, and over different topologies such as Ethernet and Token Ring. As the interconnection of different types of networks becomes more popular, it is increasingly necessary for PCs to be able to run several communications protocols at the same time.

The TUN TCP Supervisor program automates installation of TCP/IP with Netware, by creating appropriate AUTOTCP.BAT and NET.CFG files. These files, as well as any differences in configuring Netware 3.11 and 3.12, are described in more detail in this chapter.

### Using Novell's ODI drivers with Ethernet

The most common way to configure network cards in PCs with Novell Netware is using ODI drivers (Open Datalink Interface). Because ODI has become a firmly-established standard, most LAN cards on the market are delivered with them.

### Configuration files

The configuration files necessary for loading ODI and several network protocols at the same time are:

<b>NET.CFG</b>	Delivered by Netware, the card manufacturer, or created using the TUN TCP Supervisor
<b>CONFIG.SYS</b>	Necessary for loading TCP/IP parameters into memory
<b>AUTOTCP.BAT</b>	As configured in TUN TCP, this batch file may be used or may serve as an example of the programs that need to be loaded.

### NET.CFG (Ethernet)

Virtual LAN cards are configured through a text file called NET.CFG. The necessary additions in this example for an NE2000 card are given in bold characters:

```

LINK SUPPORT
    buffers 6 1600
    Max board 4
LINK DRIVER NE2000
    int 5
    port 300
    Frame ETHERNET_802.3
    Frame ETHERNET_II
    Protocol IPX 0 ETHERNET_802.3
    Protocol ODIPKT 8137 ETHERNET_II

```

The NET.CFG file prepares the network card to use the different frame types required by Netware (802.3) and TCP/IP (Ethernet II).

## CONFIG.SYS (Ethernet)

When using TUN TCP for Windows, there is **no need to change** anything in your current CONFIG.SYS file.

## ODI Startup files (Ethernet)

Dual protocol configurations using ODI require different resident programs to be loaded than single protocol configurations.

For example, an **AUTOTCP.BAT** (for packet drivers only) might execute the following commands:

```

ls1
ne2000
ipxodi
odipkt 1 98
winpkt 0x62
vlm (or netx)

```

This AUTOTCP.BAT file loads the following programs:

**ls1(.com)** Link Support Layer (also called the Link Services Layer); used for routing data packets between Network Interface Cards; also maintains LAN card, protocol, and packet buffer information. Furnished by Novell (in general, try to use the most recent version available).

**ne2000(.com)** The ODI driver for an NE2000 network card; also known as the MLID (Multiple Link Interface Driver); receives and copies packets from the Link Support Layer. **Replace this with the driver for your LAN card.** This driver is usually supplied by the board manufacturer.

As the ODI driver is loaded, a message indicating the presence of both protocols configured in the NET.CFG should be displayed. For example:

```

Int 5, Port 300, Mem CA000, Node Address C01C2665
Max Frame 1514 bytes, Line Speed 10 Mbps
Board 1, Frame ETHERNET_802.3
Board 2, Frame ETHERNET_II

```

**Note:** The messages given by the ODI driver are essential. Any difficulties encountered here indicate an incorrect net.cfg.

**odipkt 1** The packet driver for the TCP/IP kernel. The parameter "1" links the odipkt driver to the virtual board #1 (which is really the second virtual card defined in the NET.CFG; the first is #0.)

**ipxodi(.com)** Replaces ipx.com traditionally used for Novell-only networks. IPXODI does not need to be generated for a particular LAN card configuration, it addresses the virtual LAN card configured in the NET.CFG. Furnished by Novell (use versions more recent than 1.0).

**winpkt.com** A special "virtual" packet driver translator needed by Windows.

**vlm.com** The Netware workstation shell, for establishing connection with Netware file servers. Replaces the older NETX.COM. Furnished by Novell.

For more information on ODI installations, please consult the Novell Netware User Manual.

### Using ODI with Token Ring

The instructions and principles given in the previous section apply to Token Ring environments as well as Ethernet, with one or two minor differences.

Token Ring networks use Token-Ring 802.5 frame types (as opposed to 802.3 for Ethernet); TCP/IP over Token Ring uses Token-Ring\_Snap frame types.

### NET.CFG (Token Ring)

The NET.CFG file for a Token Ring installation needs to be modified in order to take into account the name of the Token Ring ODI driver:

```
LINK SUPPORT
    buffers 6 1600
    Max board 4
LINK DRIVER LANSUP
    Frame TOKEN-RING
    Frame TOKEN-RING_SNAP
    Protocol IPX E0 TOKEN-RING
    Protocol ODIPKT E0 TOKEN-RING_SNAP
```

### ODI STARTUP FILES (TOKEN RING)

The AUTOTCP.BAT file is similar to that given in the previous section; however, two lines need to be changed:

```
lsl
lansup (or ibmtoken.com)
ipxodi
odipkt 1 98
winpkt 0x62
```

#### **vlm or netx**

**lansup(.com)** ODI driver for an IBM Token Ring LAN card; used in conjunction with DXMA0MOD.SYS and DXMC0MOD.SYS (described below). Furnished by card manufacturer. In the NET.CFG, the line **LINK DRIVER LANSUP** refers to lansup.com.

## CONFIG.SYS(TOKEN RING)

As mentioned earlier, the TUN TCP Supervisor program creates and updates two system files: ESTCPIP.SYS, ESTCPIF.SYS. These files contain IP address information (including Hostname, Gateway Address, Name server, etc.).

In an ODI installation on Token Ring, the **CONFIG.SYS file** also needs to include special Token Ring drivers. In addition, we recommend using the DOS **stacks** command as shown below.

For example:

```
shell=c:\command.com /e:1024 /p
files=30
buffers=30
device=c:\xxx\DXMA0MOD.SYS
device=c:\xxx\DXMC0MOD.SYS
stacks=9,512
```

Note: The drivers DXMA0MOD.SYS and DXMC0MOD.SYS are furnished by the LAN board manufacturer, and must be present when using LANSUP.COM to access a Token Ring card.

You may also use the ODI driver IBMTOKEN.COM, which does **not** require the use of DXMA0MOD.SYS and DXMC0MOD.SYS to activate a Token Ring card.

## TUN TCP, NetWare & WFWG 3.11

Running TUN TCP, Novell Netware and Windows for Workgroups at the same time requires a configuration with three network protocols: TCP/IP, IPX/SPX and NDIS.

Before installing TUN TCP, please make sure that WFWG and Novell are installed and running correctly. If this is the case, then adding a third protocol is very simple. Usually it is best to install Netware first, and then install Windows for Workgroups - declaring Netware (IPX) as an already-existing protocol in the PC.

After installing TUN TCP, use the Supervisor to select a card configuration using **NDIS Coexistence** in order to not create an appropriate AUTOTCP.BAT file. Usually, the network programs are configured by WFWG to load automatically in the AUTOEXEC.BAT file.

## NET.CFG (Triple protocol)

If Netware and WFWG are properly running, add the two lines shown in bold type below to the existing NET.CFG file:

```
LINK SUPPORT
    buffers 6 1600
    Max board 4
LINK DRIVER SMC8000
    int 5
    port 300
    Frame ETHERNET_802.3
    Frame ETHERNET_II
    Protocol IPX 0 ETHERNET_802.3
    Protocol ODIPKT 8137 ETHERNET_II
    Frame Ethernet_802.2
    Frame Ethernet_SNAP
```

## Startup files (Triple protocol)

The order of program execution should then be as follows:

```
c:\windows\net start
cd \windows
lsl
smc8000(ODI driver for an SMC card)
c:\wtuntcp\drivers\odipkt 1 98
c:\wtuntcp\winpkt 0x62
c:\windows\odihlp.exe
ipxodi
```

vlm(or netx)

**Note:** Do not run a packet driver from the WTUNTCP\CLARKSON directory when using this type of configuration

## Using NDIS drivers (LAN Manager)

This section describes a configuration with TUN TCP and Microsoft LAN Manager using NDIS drivers.

The following lines should be in the CONFIG.SYS file

```
device=c:\lanman\protman.sys /i:c:\lanman
device=c:\lanman\elnkii.sys
device=c:\wtutcp\drivers\convert.dos /i:c:\lanman
```

Update the AUTOEXEC.BAT as follows:

```
...
REM == LANMAN == do not change these lines
PATH=C:\LANMAN.DOS\NETPROG;%PATH%
NET START WORKSTATION
LOAD NETBEUI
C:\WTUNTCP\WINPKT 0x62
REM == LANMAN == do not change these lines
...
```

Update C:\LANMAN\PROTOCOL.INI to include the lines shown in bold type:

```
;MACS
;3Com Etherlink II
[ETHERLINKII]
    drivename = ELNKII$
    interrupt  = 5
    ioaddress  = 0x300
[PROTMAN]
    DRIVENAME = PROTMAN$
[NETBEUI_XIF]
    Drivename = netbeui$
    SESSIONS  = 6
    NCBS      = 12
    BINDINGS  = ETHERLNKII
    LANABASE  = 0

[PKTDRV]
    Drivename = PKTDRV
    BINDINGS  = ETHERLINKII
    INTVEC    = 0x62
    CHAINVEC  = 0x65
```

## Coexistence with Windows for Workgroups

TUN TCP provides automatic installation and configuration for **Windows for Workgroups** (WFWG). This section describes the additions that are made to a standard TUN TCP setup in order to function correctly with WFWG.

1. The packet driver protocol is added to the end of the PROTOCOL.INI file (read by **NET START**):

```
[network.setup]
...

[protman]
...

[XIRCMAC]
...

[MS$NDISHLP]
...

[NETBEUI]
...
```

```
[PKTDRV]
DRIVERNAME=pktdrv$
BINDINGS=XIRCMAC
intvec=0x62
chainvec=0x65
```

2. A packet driver converter is added in the system.ini ( on a single line):

```
transport=ndishlp.sys,*netbeui,C:\WTUNTCP\DRIVERS\CONVERT.DOS
```

<p><b>Note:</b> When NET START is run, you should see a message on the screen saying that this driver has been loaded.</p>
--

As mentioned in the configuration section of this manual, it is highly advised to install and test Windows for Workgroups before installing TUN TCP for Windows. Most problems relating to proper LAN card configuration can be avoided if WFWG is setup correctly to start with.

## General note on program syntax

There are several other programs included in TUN PLUS that were not described in this Quick Start Guide. Most of these programs are either very standard (such as ftp) or very easy to understand (such as all the TUN TCP Windows Applications).

The DOS programs all include self-displaying command options if you execute them without any options. The most interesting examples are :

- ◆ ping.exe (for testing TCP/IP connections)
- ◆ ftp.exe (includes a help command once connection with a server has been established)
- ◆ rcopy.exe (DOS/UNIX file transfer from the c: prompt)
- ◆ rsh.exe (for executing non-interactive commands such as ls, finger and lpstat on a remote host)
- ◆ tnvt52.exe (basic terminal emulation)

Items of special interest in Windows Applications

- ◆ Multiple FTP sessions may be opened on different servers at the same time, allowing easy drag and drop file transfer between them
- ◆ TUN E-Mail interfaces with standard UNIX mail, and will notify users when new mail is received
- ◆ The Remote Shell program may be run with the option "-b" in order to display macro buttons that you may customize



# Conclusion

We hope that you have been able to get a good idea of the power that TUN PLUS provides, and how the product may be of use to you in creating network environments using TCP/IP.

Enjoy experimenting with these programs, and please do not hesitate to contact your distributor if you need more help or have any questions.

Note: When you decide to upgrade your demo product into a full (permanent) working version, you do not have to reinstall everything. When you receive your final serial number and activation key, all you will have to do is run a simple serialization program.