

Trumpet Winsock

Version 1.0

By Peter R. Tattam

Managed by Trumpet Software International

Copyright (C) 1993,1994 by Peter R. Tattam

All Rights Reserved

Introduction

Thank you for using the Trumpet Winsock. It is through the kind support of many users out there that quality networking software has been available at affordable prices to the Internet community. The Trumpet Winsock is a Windows Sockets 1.1 compatible TCP/IP stack which provides a standard networking layer for many Windows(tm) networking applications to use, and has itself been a major vehicle in achieving widespread use of Windows Sockets 1.1. The product is a shareware item and as such, you are permitted to evaluate it for a period of 30 days. If you are satisfied with its usefulness, a registration form is provided which you can fill out and send to Trumpet Software International. A registration fee is requested to maintain the development and support of this software. Suitable arrangements have been made for site licenses, and details can be found in a later section.

Disclaimer & Copyright

These programs are Copyright (C) 1991-1994 by Peter R. Tattam,
All Rights Reserved.

They are provided as shareware with the following limitations:

These programs are shareware and are not to be resold or distributed for sale with other programs which are for sale. There is no warranty or claim of fitness or reliability. The programs are distributed AS IS, and as such neither the author, nor Trumpet Software International nor the University of Tasmania shall be held liable for any loss of data, down time, loss of revenue or any other direct or indirect damage or claims caused by these programs.

Instructions for Installing the Trumpet Winsock.

The Trumpet Winsock will only run on your PC under the following conditions. You must have either a packet driver available for use by network programs, or if you wish to use SLIP, a free comms port. Additionally, packet drivers can only be used reliably under enhanced mode using WINPKT. Standard mode can be used, but care must be taken to avoid system crashes. NDIS and ODI can be used via packet driver shims, but their use is not supported. PKTMUX may also be used instead of WINPKT, and must be version 1.2c or later, but again its use is not supported.

If you already have some kind of TCP/IP networking package installed, it

is most likely that the Trumpet Winsock will not run and you will have to massage your system configuration to install the Trumpet Winsock, possibly even to the extent of uninstalling that networking package. Alternatively, there may be a Winsock available for your package in which case the Trumpet Winsock will not be required.

Installing Trumpet Winsock over Packet Driver.

Firstly, if you don't know what a packet driver is, it is normally a small piece of software which sits in between your network card and your TCP program. This provides a standard interface which many programs can use in a similar manner to BIOS calls using software interrupts.

Why is it called a packet driver? This is because modern networks send information using packets of information rather than sending information one byte or character at a time. For example, Ethernet sends information in frames of up to 1514 bytes long. The reason for sending things in packets is that information can be transmitted much more efficiently in packets.

Central to the concept of the packet driver is the vector which is used to communicate with it. The 80x86 family of processors allows programs to communicate with the operating system through what is called a "software interrupt", which always has a number in the range 0 to 255. This is termed a "vector" and is the one of the key mechanisms to pass control to the MS-DOS operating system. Usually the vectors are in hexadecimal, making the range expressed as 0x00 to 0xFF. The 0x in front of the number means that we are using hexadecimal numbers instead of decimal numbers. They may also be expressed in the notation 00H to FFH, or \$00 to \$FF. If you are dealing with packet drivers, hexadecimal notation is much more common, but occasionally they are expressed in decimal. Examples of software interrupts in use on PC's are 0x10 for the video BIOS, or 0x21 for calls to DOS.

Packet drivers are only allowed to have a software interrupt vector in the range 0x60 to 0x7F. Normally, you will pick 0x60 as the default place to install your packet driver, but certain machine configurations may make that vector unavailable. Just choose one that is free - the packet driver should tell you if you can use it or not.

The Trumpet Winsock also uses a special virtual packet driver "wrapper" which enables your packet driver to function correctly in Windows. While the packet driver is an efficient way to communicate with your

network card, it will not work correctly from Windows without a little assistance. The program "WINPKT" was written by some clever people on the Internet to allow a packet driver to work correctly within Windows by making sure that packets get directed to the correct "virtual machine" under Windows enhanced mode. A "virtual machine" can be either the entire Windows session, or any dos session active within Windows. Refer to the Windows system documentation for more details.

In addition to this, you will need to have some understanding of IRQ vectors and I/O addresses that may be relevant to installing your network card.

Where do I obtain packet drivers from?

These days, packet drivers are usually provided with your network card, but a comprehensive collection of public domain packet driver can be obtained from a packet driver collection called the "Crynwr Packet Driver Collection." Information on where to get this packet driver collection from is provided as an appendix to this document.

Actually Installing the Winsock.

Before you do anything, copy the files **winsock.dll**, **tcpman.exe**, **winpkt.com**, **hosts**, **services** and **protocol** to a suitable directory.

eg. c:\trumpet

the essential files:

winsock.dll	the guts of the TCP/IP driver
tcpman.exe	interface program for managing the winsock
winpkt.com	virtual packet driver interface for windows
hosts	list of host names & aliases
services	list of Internet services
protocol	list of Internet protocols

Modify the path line in your autoexec.bat to contain a reference to that directory.

eg. path c:\dos;c:\windows;**c:\trumpet**

Make sure it is active by rebooting or executing autoexec.bat again.

The most basic setup of packet driver and WINPKT would look something like this example :

```
ne2000 0x60 2 0x300
WINPKT 0x60
```

The first line installs an NE2000 packet driver on vector 0x60 using IRQ 2 and I/O address 0x300

The second line installs the WINPKT virtual packet driver using the same vector that the ne2000 packet driver was installed on.

This is of course an example so your mileage will of course vary. Some example configurations are described later in this document. Choose the one which suits you the best and modify it to your requirements.

Now you are ready to start windows. Start it up!!

From windows, start up tcpman. From the file manager, go File/Run, and then "tcpman". If this fails, the path is probably not set up correctly, so fix it. Later on, you can install it as an icon to use it directly.

Assuming you are a first time user, a setup screen will appear giving you a number of options to fill in. You will need to fill in a few details to enable the TCP package to function. Fill in the following details. If you are unclear on any of them, try to seek some help from qualified Internet support staff - it will save you a lot of time.

IP address	your Internet IP address, "bootp", or "rarp". lower case please. If you use BOOTP, be sure to have a BOOTP service on the network or the winsock will not load.
Netmask	your Internet network mask. (eg. 255.255.0.0)
Default Gateway	your default Internet gateway. (IP address)
Name server	your name server IP address for DNS searches. You may provide more than one address by

	separating the addresses with spaces. (IP addresses only)
Time server	at present unused - future winsock API's may support this. (IP addresses only)
Domain suffix	a space separated list of domain suffixes to be used when resolving names in the DNS system.
Packet Vector	either leave this as 00 to search for the packet driver, or the vector that you installed the packet driver under. The number is required in hexadecimal without the leading "0x". In our example, you would provide "60". (numeric)
MTU	Maximum Transmission Unit. (numeric) For Ethernet, 1500 is the maximum, and is recommended.
TCP RWIN	TCP Receive Window (numeric) eg. (defaults to 4096 but can be larger)
TCP MSS	TCP Maximum Segment Size (numeric) (usually MTU - 40)

The rest of the details should be greyed out and you need not try to fill them in. The Internal SLIP check box should not be checked.

The first four parameters and the packet vector are required for successful functioning of the winsock, while the rest can be tailored to suit your needs.

When you are done, click on <OK> and if all goes well, the Trumpet Winsock will be initialised. You are now ready to start using the winsock.

What to do if something goes wrong

Firstly...

The Trumpet Winsock requires that you have the correct combination of tcpman.exe, winsock.dll and winpkt.com. When upgrading to a new release, replace each of these files to be sure that everything is up to date.

If you get the messages about not finding a packet driver or unable to load TCP, then check that the packet driver loaded properly, that WINPKT managed to find it, and that the correct vector was chosen from tcpman.

At the moment, only Ethernet and SLIP packet driver types are supported. Token ring is only available via the ibmtoken packet driver, and should work, but is untested by the author.

ODI can be used via the ODIPKT shim, and NDIS via the DIS_PKT shim. examples are provided later on. Also, examples are provided of installation using NetWare.

Possible causes for tcpman load errors specific to packet drivers.

unable to bind protocol 0806	another TCP stack is using the packet driver... remove it.
WINPKT or pktdrv not found	couldn't find the correct packet driver. Also check the vector number in TCPMAN.
unable to allocate network buffers	critical error... try to free up some special driver memory by removing windows device drivers.
network buffers low	not critical but inadvisable... see above

If WINPKT can't load (No packet driver found), check your packet driver vector number. Some drivers may choose a default vector which is not at 0x60. eg. ODIPKT default is 0x69

If you are using ODIPKT and you can't get any response, you probably accessing the wrong protocol. If you have the ARP trace on, you will possibly get "ARP timed out" messages as well. The first parameter of ODIPKT selects the correct protocol. Try adjusting this.

Anything else... contact me. I'll try to figure out what's wrong, but first browse the samples provided.

Installing Trumpet Winsock over Internal SLIP

SLIP is a simple protocol which allows an Async serial connection to send Internet Protocol (IP). You usually need to have access to a server which can understand SLIP. Usually, SLIP is accessed via a phone line, and with the advent of high speed modems, TCP/IP is a reality over a dial-up connection.

The Trumpet Winsock has facilities for managing a SLIP connection as well as the ability to use dialling scripts for logging in and out of your SLIP server.

Actually Installing the Winsock.

Before you do anything, copy the files winsock.dll, tcpman.exe, hosts, services and protocol to a suitable directory.

eg. c:\trumpet

the essential files:

winsock.dll	the guts of TCP/IP driver
tcpman.exe	interface program for setting up the winsock
hosts	list of host names
services	list of Internet services
protocol	list of Internet protocols

Modify the path line in your autoexec.bat to contain a reference to that directory.

eg. path c:\dos;c:\windows;c:\trumpet

Make sure it is active by rebooting or executing autoexec.bat again.

Now you are ready to start windows. Start it up!!

From windows, start up tcpman. From the file manager, go File/Run, and then "tcpman". If this fails, the path is probably not set up correctly, so fix it. Later on, you can install it as an icon to use it directly.

Assuming you are a first time user, a setup screen will appear giving you a number of options to fill in. You will need to fill in a few details to enable the TCP package to function. Fill in the following details. If you are unclear on any of them, try to seek some help from qualified Internet support staff - it will save you a lot of time.

Firstly, click on Internal SLIP. Some of the parameters will be greyed and others ungreyed.

IP address	your Internet IP address or "bootp". lower case only. Only use BOOTP if you are not intending to use a dial in script. If using a dialler script with the address extracted by the script, or BOOTP later, just leave it with default value of 0.0.0.0 Only use BOOTP if your server supports it, otherwise the winsock will delay for about 15 seconds and the message "Unable to load TCP" will come up.
Name server	your name server IP address for DNS searches. You may provide more than one address by separating the addresses with spaces. (IP addresses only).
Time server	at present unused - future winsock API's may support this. (IP addresses only).
Domain suffix	a space separated list of domain suffixes to be used when resolving names in the DNS system.
MTU	Maximum Transmission Unit. Related to TCP MSS... usually TCP MSS + 40. (Numeric)
TCP RWIN	TCP Receive Window. It is recommended that this value be roughly 3 to 4 times the value of TCP MSS. (Numeric)
TCP MSS	TCP Maximum Segment Size, It is recommended that this be a smallish value when using SLIP - say 512 bytes for SLIP and lower for CSLIP. CSLIP is able to compress data more efficiently when it is less than 255. (numeric)

SLIP port	your comms port number ..1=com1, 2=com2 etc. (numeric)
baud rate	the speed you wish to run at. (numeric)
hardware handshake	recommended if your link supports it.
Van Jacobson CSLIP	
Compression	if your server will support it. You may also have to adjust MTU, MSS & RWIN. to be suitable.
Online Status	
Detection	if your modem will support it, select DCD or DSR on-line status detection.

The rest of the details should be greyed out and you need not try to fill them in.

When you are done, click on <OK> and if all goes well, the Trumpet Winsock will be initialised. You are now ready to start using the winsock.

Logging in to the server.

You can use either the manual login or the automatic scripting to access your server. For the time being, choose manual and log into your server with the appropriate commands. Don't forget to use the <esc> key to get out when you have finished dialling in. After logging in, you may need to go and set your IP address if it is allocated dynamically.

If you wish to use another terminal program to dial in to the server, don't forget to issue AT&D0, or disable DTR dropping when exiting the program, or the connection will be severed when the application closes the comms port.

Try out pingw to a well known host IP address to see if all is well.

Problems

Check your baud rates...

If using hardware hand shaking with an external modem, make sure the

cable is correctly wired.

At the moment, all dialling must be done with 8bits, no parity. This may not work for you... you will then need to use an external dialler. The next revision will have an extension to the dialler to allow this.

If all else fails... contact me !!

Once you have determined your login sequence, you can set up a login script. A sample script is provided along with a listing of a typical session.

Automatic dialling.

Minimal scripting is supported, and the script commands are

input <timeout> <string>	wait for string received.
output <string>	send string.
display <string>	display string on display.
wait <timeout> { DSR CTS RLSD DCD }	wait for DSR or CTS or RLSD(DCD)
trace (on off)	useful for debugging scripts
echo (on off)	defaults to on
password <prompt>	message box for password
username <prompt>	message box for username
address <timeout>	parse IP address
set (DTR RTS) (on off)	set/reset the modem lines.
sleep <seconds>	pause for so many seconds.
exec <string>	program will be started up concurrently using winexec().
online	enter SLIP mode. commands depending on received characters will not work correctly after this command is issued since the

winsock will interpret data as SLIP frames. Useful before an exec command which uses the winsock.

BOOTP

inform the winsock that a BOOTP will be required after the script has finished.

means start comment except inside string

string arguments

\l	line feed
\r	return
\n	cr/lf pair
\f	form feed
\t	tab
\b	backspace
\nnn	ASCII value in decimal
\i	IP address
\p	password
\u	username
\c	comm port number (as you have configured it)

a sample script is given for logging in to our Xylogics terminal server.

```

output atz\13
input 10 OK\n
#output atd242284\13
output atd241644\13
input 30 CONNECT
input 30 \n
wait 30 dsr
output \13
input 30 username:
output tattam\13
input 30 password:
password Enter your password
output \p\13
input 30 >
output who\13
input 30 >
output slip\13
input 30 Your address is
address 30
input 30 \n
display \n
display Connected. Your IP address is \i.\n

```

```
exec pingw 131.217.10.1
```

Here's a log of a typical session. Names have been blanked out for security.

```
Trumpet Winsock Version 1.00 Alpha #18
Copyright (c) 1993 by Peter R. Tattam
All Rights Reserved.
SLIP ENABLED
Internal SLIP driver COM3 Baud rate = 38400 Hardware handshaking
My ip = 131.217.8.4 netmask = 255.255.0.0 gateway = 131.217.250.1
Executing script c:\dev\tcpip\winsock\login.cmd
SLIP DISABLED
atz
OK
atd241644
CONNECT 38400
```

Annex Command Line Interpreter * Copyright 1991 Xylogics, Inc.

```
Checking authorization, Please wait...
Annex username: xxxxxx
Annex password:
```

```
Permission granted
University of Tasmania
AARNet Terminal Server
```

```
SLIP users:
Use a maximum segment size (MSS) of 209
and a maximum transmission unit (MTU) of 255.
```

```
Async AppleTalk users:
Configure MacTCP to use the Computing Centre zone.
*** Note change in procedures for starting async AppleTalk.
*** After typing atalk you will be prompted for your password
*** again.
```

```
AARNET TS5 >who
Port What User      Location    When      Idle Address
2  CLI xxxxxxxx    ---        8:01pm    [local]
+1  'telnet tasman.cc'
3  CLI xxxxxxxx    ---        8:15pm    [local]
+1  'rlogin franklin.cc'
4  SLIP modem4     ---        8:19pm    ants
5  CLI xxxxxxxx    ---        8:34pm    [local]
6  CLI xxxxxxxx    ---        7:19pm    [local]
+1  'rlogin baudin.cc'
17 SLIP modem18    ---        6:39pm    ants
AARNET TS5 >slip
```

```
Switching to SLIP.
Annex address is 131.217.250.10. Your address is 131.217.8.5.
```

```
Connected. Your IP address is 131.217.8.5.
```

```
Script completed
SLIP ENABLED
```

Dialler problems.

Q. tcpman just pauses when starting up, then gives the message "unable to load TCP".

A. You've probably got BOOTP set. Replace it by 0.0.0.0 before dialling and try again. RARP is impossible to send via SLIP so don't bother with that.

Q. The connection appears to be too slow compared to Xmodem.

A. Possibly the MTU/MSS & RWIN settings are not right. Try to make RWIN about 3 to 4 times MSS and an exact multiple if possible. Turn on the IP trace to see if fragmentation is occurring on TCP connections. If so, then reduce MSS until it stops. UDP packets will still be fragmented, but nothing can be done about that. On the trace, TCP is type 6 while UDP is type 17.

Q. Some input commands in the script don't work.

A. Check for upper case/lower case conflicts. Also check for blanks at the end of the lines.

For other problems, contact me at

trumpet-bugs@petros.psychol.utas.edu.au, or subscribe to the Trumpet discussion group and ask your question. Details are at the end of this document. As time goes on, various FAQ's will be constructed to cope with the more common problems.

Sample Configurations for Packet Driver.

1. Plain ne2000 packet driver using WINPKT.

```
ne2000 0x60 2 0x300
WINPKT 0x60
```

2. Ne2000 packet driver with Novell NetWare access using WINPKT.
Important is the specification of the -n switch of the packet driver. Some packet drivers don't support this switch. In that case, you may be forced to use ODI instead. An example could be the Xircom Pocket Adapter.

```
ne2000 -n 0x60 2 0x300
WINPKT 0x60
pdipx
netx
path c:\dos;c:\network\win31
f:
login
```

3. Ne2000 packet driver with Novell NetWare access using PKTMUX.
Notice that WINPKT is not required since PKTMUX does a similar job.

```
ne2000 -n 0x60 2 0x300
pktmux 4
pktdrv
pktdrv
pktdrv
pktdrv
pdipx
netx
path c:\dos;c:\network\win31
f:
login
```

4. ODI setup with NetWare access.

You will need ODIPKT. The latest known release is 2.4 It is important that ODIPKT reference the correct protocol for IP access. This can be specified as the first parameter to ODIPKT (0=1st, 1=2nd and so forth)

Here's a sample of my network attach batch file.

```
@echo off
cd \
```



```

lh lsl
lh \odi\ne2000
cd \net
lh ipxodi
lh odipkt
lh WINPKT 0x69
lh netx
path c:\dos;c:\net\win31
f:
echo on
login

```

Also, your net.cfg must be suitably configured. Here are the relevant excerpts from my net.cfg

```

Link Support
  Buffers 8 1586
  MemPool 16384

```

```

Link Driver NE2000
  Port #1 300 20
  Int #1 2
  Frame Ethernet_II
  Frame Ethernet_802.3
  Protocol IPX 0 Ethernet_802.3

```

The ordering of the frame protocols is important for the default setup of ODIPKT. Also, users should be aware that there are two versions of ODIPKT, one released I believe by FTP Software, and the public domain one. I refer to the public domain version. Also note that there are two programs with the same name of "ne2000.com". One is a packet driver and is referred to in an earlier section. The one referred to in this section is actually an ODI driver and won't function as a packet driver at all.

5. NDIS & Windows for Workgroups setup. (courtesy of Peter Whisker, WhiskerP@LGWCT.LOGICA.COM)

Installation of Trumpet Winsock makes use of the DIS_PKT9.DOS or DIS_PKT.DOS shims which provide a Packet Driver interface the NDIS. The version I have tested is found on a number of sites as DISPKT11.ZIP, and contains DIS_PKT.DOS dated 28/4/93. The example is based upon an actual configuration using DEC Pathworks version 4.1 (DECNET version) with a DEPCA Ethernet card. This installation assumes that you have NDIS up and running and have a valid PROTOCOL.INI file.

You need to perform three basic steps:

- Load the DIS_PKT driver in CONFIG.SYS following the load of PROTMAN.SYS:

```
DEVICEHIGH=\DECNET\PROTMAN.SYS /I:C:\DECNET
DEVICEHIGH=\DECNET\DEPCA.DOS
DEVICEHIGH=\DECNET\DIS_PKT.DOS
```

- Add a few lines to PROTOCOL.INI (here labelled [PKTDRV]) in order to bind it to the Ethernet driver. In this example, DIS_PKT.DOS is configured with packet vector 60.

```
[DEPCA.DOS]
DRIVERNAME = DEPCA$
```

```
.
```

```
.
```

```
.
```

```
[PKTDRV]
DRIVERNAME = PKTDRV
BINDINGS = DEPCA.DOS
INTVEC = 0x60
```

- In AUTOEXEC.BAT, load WINPKT or PKTMUX as described in examples 2 and 3 above. This should be done **after** the NETBIND command in your NDIS startup has been executed. In the case of Pathworks, this command is normally contained in STARTNET.BAT. For example:

```
call \decnet\startnet.bat
WINPKT 0x60
```

6. Some more packet driver installations courtesy of Ashok Aiyar (ashok@biochemistry.bioc.crwu.edu)

Configuration for Cabletron Network Cards. The packet driver provided by Cabletron is a little confusing as it doesn't use the same parameters as packet-drivers that use the Crynwr skeleton.

Typically the Cabletron driver is loaded as:

```
"csipd_e /s:62 /h:7 /p:300"
```

In this example the software interrupt is 0x62. Load winpkt.com as

"WINPKT 0x62"

Release 11 of the Crynwr packet drivers includes a driver for Cabletron cards written by Kai Getrost using the Crynwr skeleton that uses the same parameters as the other Crynwr drivers. This driver (CTRONDNI.COM) seems to work well with E1020/1040 and E2020 Cabletron cards. Indeed I see a performance gain over the Cabletron driver. Your mileage may vary.

C/SLIPPER with PKTMUX. Although the Trumpet Winsock has built in support for C/SLIP, there are situations when in addition to Winsock applications there is a need to run packet driver applications simultaneously over a SLIP link. For such situations, PKTMUX is of utility.

Example:

```
CSLIPPER vec=65 com1 irq=04H baud=57600 ether
PKTMUX 4 65 /4 .... (support for a maximum of 4 virtual packet drivers)
PKTDRV 60 65
```

Configure the Trumpet Winsock to use the virtual packet driver at 0x60. All other virtual packet drivers (PKTDRV) can be loaded in the DOS Windows in which they are used. They need not be loaded before entering Windows.

(Ed. Note... You may also require the use of a special comms buffer to enhance the buffering capabilities of Windows when using slipper/cslipper. An FAQ on doing this is available from biochemistry.bioc.cwru.edu via gopher or FTP. It is not needed when using the internal SLIP functions of the Winsock)

Extra Info

You may use environment variables or command line options to override some of the network parameters. They have the same names as the saved parameters in trmpwsk.ini. This file normally resides in the winsock directory rather than the windows directory since this facilitates setting up the winsock in a networked environment. IP addresses can be overridden by using the environment variables, or the command line.

example of command line.

```
tcpman -ip=123.231.213.123 -netmask=255.255.255.0
```

example of environment variable

```
set ip=123.231.213.123
set netmask=255.255.255.0
```

Here's a list of parameters.

ip/myip	your IP address or 'bootp' or 'rarp' (lower case only)
netmask	your netmask. eg. 255.255.0.0
gateway/mygateway	your gateway (IP address)
dns	list of DNS IP addresses
time	list of time server IP addresses
domain	list of domain name suffixes
vector	packet driver vector in hex
MTU	Maximum Transmission Unit
RWIN	TCP Receive Window.
MSS	TCP Maximum Segment Size
slip-enabled	0 = off, 1 = on
slip-port	port number (1-9)
slip-baudrate	baud rate in decimal
slip-handshake	0 = off, 1 = on
slip-compressed	0 = off, 1 = on

The Crynwr packet driver collection

Availability

The Crynwr packet driver collection is available by mail, by FTP, by e-mail, by UUCP and by modem. The drivers are distributed in three files: drivers.zip, which contains executables and documentation, drivers1.zip, which contains the first half of the .ASM files, and drivers2.zip, which

contains the second half of the .ASM files.

Mail:

Columbia University distributes packet drivers by mail. The formats are 9-track 1600 bpi tapes in ANSI, tar, or OS SL format, or PC diskettes (360K 5.25" and 720K 3.5"). The exact terms and conditions have yet to be worked out, please call (212) 854-3703 for ordering information, or write to:

Kermit Distribution, Dept PD
Columbia University Center for Computing Activities
612 West 115th Street
New York, NY 10025

or send e-mail to kermit@watsun.cc.columbia.edu (Internet) or KERMIT@CUVMA (BITNET/EARN).

FTP/e-mail:

The packet driver collection has its own directory devoted to it, `pd1:<msdos.pktdrvr>`. The drivers are there, along with many free programs that use the packet drivers.

SIMTEL20 files are also available from mirror sites OAK.Oakland.Edu (141.210.10.117), wuarchive.wustl.edu (128.252.135.4), ftp.uu.net (192.48.96.9), nic.funet.fi (128.214.6.100), src.doc.ic.ac.uk (146.169.3.7) or rana.cc.deakin.oz.au (128.184.1.4), or by e-mail through the BITNET/EARN file servers.

Modem:

If you cannot access them via FTP or e-mail, most SIMTEL20 MSDOS files, including the PC-Blue collection, are also available for downloading from Detroit Download Central (313) 885-3956. DDC has multiple lines which support 300/1200/2400/9600/14400 bps (103/212/V22bis/HST/V32bis/V42bis/MNP). This is a subscription system with an average hourly cost of 17 cents. It is also accessible on Telenet via PC Pursuit and on Tymnet via StarLink outdial. New files uploaded to SIMTEL20 are usually available on DDC within 24 hours.

CD-ROM:

Public, private or corporate institutions and libraries interested in the SIMTEL20 MSDOS collection in CD-ROM format bundled with library card-catalog type access and duplication software can contact Coyote Data, Ltd. by mail at 1142 N. Main, Rochester, MI 48307 or by FAX at (313) 651-4071.

UUCP:

The packet driver files are available from UUNET's 1-900-GOT-SRCS, in uunet!~/systems/msdos/simtel20/pktdrvr. See UUNET.DOC for details.

ODIPKT location

I am told that the originating site for ODIPKT is the following...

Host hsdndev.harvard.edu

Location: /pub/odipkt

FILE -rwxr-xr-x 2915 Aug 21 20:01 odipkt.com

A copy of the NDIS shim is there also.

Trumpet General Discussion Group.

The machine petros.psychol.utas.edu.au is now running a local news service with the news groups

trumpet.announce
trumpet.bugs
trumpet.feedback
& trumpet.questions

If you do not have access directly to this service, these news groups are gatewayed to the following mailing list.

You may join the new Trumpet mailing list by sending a message to

listproc@petros.psychol.utas.edu.au

with just one line in the body.

subscribe trumpet-user Your Full Name

Where "Your Full Name" should be replaced by your actual full name.

The list is called

trumpet-user@petros.psychol.utas.edu.au

and is running on a 486/50 FreeBSD system. Hopefully it will cope ;-)

You can ask questions, or discuss any aspect of any Trumpet program on this group. Feedback is always welcome. There is also an anonymous FTP area with all the latest Trumpet programs and pre-releases. If you do use a pre-release, be prepared for unexpected problems since such programs are in alpha/beta test.

Bugs or Comments

Send to

trumpet-bugs@petros.psychol.utas.edu.au

For bug reports, please send a copy of config.sys, autoexec.bat, trumpwsk.ini, and any other relevant network configurations. In the case of ODI, also send net.cfg. We will do my best to sort out your problem. Due to the high demand for the Trumpet Winsock, my mail box can be overloaded at times. Be patient... someone will answer you.

TCPMAN - The Trumpet Winsock TCP Manager**Menu options.**

File/Setup	calls up the setup dialog for configuration
IP address	your IP address, "bootp" or "rarp" (lower case). BOOTP will only work if there is a BOOTP service on-line. RARP will only work if using Ethernet, and there is an RARP service on-line.
Netmask	your network mask.
Default gateway	your default Internet gateway or router.
Name server	your Domain Name Server address.
Time server	(unused leave empty)
Domain Suffix	A space separated list of suffixes to be tried when looking up names via the name server.
Packet Vector	for accessing the packet driver in hex.
MTU	Maximum Transmission Unit.
TCP RWIN	TCP Receive Window
TCP MSS	TCP Maximum Segment Size
Demand Load Timeout	Number of seconds tcpman stays loaded after the application has finished with it.
Internal SLIP	Click on this for internal SLIP support & dialler support.
SLIP port	which comms port to use.
Baud Rate	speed of the connection.
Hardware Handshaking	turn on for RTS/CTS handshaking. May require the AT&K3 modem command to function properly.
Van Jacobson CSLIP compression	turn on for CSLIP TCP header compression.
Online Status Detection	needed for dialler autologin / autologout enabling.
None	no online status detection
DCD (RLSD) check	may require AT&C1 modem command to

function.

DSR check **may require AT&S1 modem command to function.**

File/Register **calls up the registration dialog.**
File/Exit **quits the TCP manager, forcing the winsock to be unloaded.**

Edit/Copy **copy selected text on tcpman display to the clipboard**
Edit/Clear **clear the tcpman display**

Tracing options. Use with care since some applications may crash when the traces are active. Should a program crash with stack overflow, the winsock may remain loaded in memory even though tcpman has exited. It is advisable to restart windows if this happens and possibly even to reboot your machine. Also, timing measurements of the winsock throughput will be severely affected by the trace options.

Trace/TCP	turn TCP trace on/off
Trace/UDP	turn UDP trace on/off
Trace/IP	turn IP tracing on/off
Trace/ARP	turn ARP tracing on/off
Trace/RARP	turn RARP tracing on/off
Trace/Ethernet	add Ethernet headers to IP/ARP/RARP traces.
Trace/Extra detail	add some extra detail to TCP, UDP & IP traces.
Trace/Socket calls	trace each winsock call.
	most parameters are displayed as well.
Trace/DNS	trace Domain Name Server operations.
	Use with care, stack overflows can be frequent.
Trace/Messages	trace Async Socket messages.

Dialler/Login	invoke the login.cmd dialler script.
Dialler/Bye	invoke the bye.cmd dialler script.
Dialler/Other	invoke other scripts. a file selection dialog of *.cmd will be displayed.
Dialler/Options	call up the dialler options dialog.

No automatic login

Automatic login on startup only.

Automatic login and logout on demand.

SLIP inactivity timeout (minutes) Number of minutes to wait before exiting winsock. (when no application is using the winsock.) Automatic login & logout must be enabled for this to close the SLIP connection. A value of 0 disables the timeout.

Dialler/Manual Login	invoke the dialler manually. Use <esc> to exit from the manual dialler.
Dialler/Edit Scripts	invokes notepad to edit any script.
Help/About	Display the version number and copyright.

Registration

Registration of the Trumpet Winsock is encouraged since it funds further development of the winsock. It involves sending in a registration form filled in with your registration name and other details. On receipt of your registration, you will receive a password which will remove the UNREGISTERED VERSION notice and replace it with your registration name. As part of this registration, you will receive enough support to get you going within the existing capabilities of the winsock at the present time, and preference will be given to registered users when it comes to bug fixes or future enhancements to the winsock. Packet drivers using Ethernet and SLIP are presently the only supported network access. The winsock will function through the use of packet driver shims for ODI, NDIS and token ring, but the use of these is not supported, neither is the winsock supported should you be using PKTMUX.

The Trumpet Winsock is currently distributed as shareware. You may use the Trumpet Winsock for 30 days to evaluate its usefulness. If at the end of that time you are satisfied with the Trumpet Winsock as a product, you can register it. The basic registration fee for a single user version of the Trumpet Winsock is US\$20. See a later section for details on multi-user site licenses.

Australian users should contact me regarding Australian pricing information and availability.

Cheques or Postal Orders should be made out to

Trumpet Software International

and sent to

**Trumpet Software International
GPO Box 1649,
HOBART, TAS AUSTRALIA 7001**

You should fill out the following order form and send it along with your cheque or Postal Order to the above address.

FAX/Phone International 61-02-487049, Australia 002-487049

Please note that International mail can be rather slow, and it may take up to 2 months to receive your registration.

```

+-----+
|   ORDER FORM   |
| for Trumpet Winsock version 1.0 Software |
+-----+

```

Ship to:

Bill to:

[]	[]
[]	[]
[]	[]
[]	[]
[]	[]

Please supply the following items:

Licence to use Trumpet Winsock 1.0 for [] users
 US\$[]

Tick at least one of the following options.

- ☐ 5.25" disk with the latest version of Trumpet Winsock + password
☐ 3.5" disk with the latest version of Trumpet Winsock + password
☐ registration password via post
☐ registration password via e-mail

Your registration name (required) (will appear on program)

[_____]

Your e-mail address (optional - print clearly)

[_____]

Date sent [_____] Expected delivery Date [_____]

Site Licenses

A site license is defined as being a sale to an organisation or company, and may not be resold or redistributed for profit. It may only be used within that organisation.

prices valid until 30-Jun-1994

Single User license

1 user \$20 US

Multi-user site license

Trumpet Winsock will be charged by the number of simultaneous users.
The pricing structure for commercial users is thus

1-99 users	\$20 US per user
100-499 users	\$2000 US + \$10 US per additional user over 100
500-999 users	\$6000 US + \$5 US per additional user over 500
1000+ users	\$8500 US + \$2 US per additional user over 1000

site restriction 10km radius (negotiable)

Unlimited Commercial Site License

\$10,000 US for first year.
subsequent years, 25% of unlimited site license fee for that year.
site restriction 100km radius (negotiable)

The pricing structure for educational users is thus

1-100 users	\$20 US per user
100+	\$2000 US

site restriction unlimited.

Your site license will give you support for up to 12 months from the date of purchase. Such support will include upgrades and bug fixes within that 12 months within the constraints of the program's existing capabilities. Future upgrades will be 25% of the original license fee per annum. Arrangements will also be made for conversion of smaller licenses to larger ones.

Should you wish to obtain the Trumpet Winsock to distribute with other programs, you should make a suitable offer to Trumpet Software International, and it will be considered. Source code will not be made available under any circumstances, and Trumpet Software International reserves the right not to accept any offer which is not considered acceptable.

**Trumpet Software International
GPO Box 1649,
HOBART, TAS AUSTRALIA 7001**

FAX/Phone International 61-02-487049, Australia 002-487049