Trumpet Winsock

Version 2.0

By Peter R. Tattam

Managed by Trumpet Software International Pty Ltd.

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 that 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 released as shareware, and as such you are permitted to evaluate it for 30 days. If you are satisfied with its usefulness, a registration form is provided which you should 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 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.

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.

Using Trumpet Winsock over Internal SLIP/PPP

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. SLIP is usually accessed via a phone line, and with the advent of high speed modems, TCP/IP is a reality over a dial-up connection.

PPP (Point to Point Protocol) is a protocol that is more complicated than SLIP, which offers error correction and is more reliable than SLIP.

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.

Installing the Winsock for use over Internal SLIP/PPP

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 are:

winsock.dll the guts of the TCP/IP driver

tcpman.exe controlling program for the Winsock

sendreg.exe registration program

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 your computer or by executing autoexec.bat again.

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

From windows, start up tcpman by selecting File/Run from the file manager, then type "tcpman". If this fails, the path is probably not set up correctly, so fix it. Later, you can set up tcpman as an icon so it can be started 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 the following details to enable the TCP package to function. 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 or Internal PPP. Some of the parameters will be greyed and others ungreyed.

IP address your Internet IP address or "bootp" in lower case only. Only use bootp if your server supports it, otherwise the winsock will delay for up to 2 minutes and the message "Unable to perform bootp" will come up. The winsock will not function should the bootp fail.

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). For SLIP, we suggest 256 initially.

TCP RWIN TCP Receive Window. It is recommended that this value be roughly 3 to 4 times the value of TCP MSS (Numeric). For SLIP, we suggest 848.

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). For SLIP, we suggest 212 initially.

SLIP port your comms port number ..1=com1, 2=com2 etc., (numeric).

baud rate the speed you wish to run at (numeric). Up to 115200 is supported although speeds greater than 19200 require suitable hardware.

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. You will need to make sure that your modem has a default power on setting of AT&C1 for this to function.

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 with internal SLIP

Check your baud rates...

If using hardware hand shaking with an external modem, make sure the cable is correctly wired.

By default, all dialling must be done with 8 bits, no parity. This may not work for you... You may need to choose Dialler/Options to select Control Panel parity/word size if you are not using 8 bits, no parity.

If all else fails... contact us !!

Once you have determined your login sequence, you can set up a login script. A sample script is provided along with the disk.

Many Internet dialup providers will have their own scripts for connecting to their systems. Please contact your provider for a suitable script. Trumpet Software International can offer you assistance in setting up your script if required.

Using Trumpet Winsock with Packet Driver.

Firstly, if you don't know what a packet driver is; it is 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 a 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 expressed in hexadecimal, with the range 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, otherwise a comprehensive collection of public domain packet drivers 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.

Installing the Winsock for use with Packet Drivers

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 controlling program for the winsock

sendreg.exe registration program

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 your computer or by executing autoexec.bat again.

Setting up your packet driver.

The packet driver command lines in the examples below would normally be part of a batch file used to connect to your network although it could be placed in autoexec.bat or in a batch file to start Windows in which case you would need to add a third line to start Windows.

The most basic setup of packet driver and WINPKT would look something like this example (for an NE2000 compatible network card).

Note that the packet driver name for other types of network cards will of course be different to ne2000:

```
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.
The next example shows the setup for a Western Digital 8-bit network card using vector 0x61, IRQ5 and I/O address 0x320.

```
wd8003e 0x61 5 0x320
WINPKT 0x61
```

These of course are only examples so your mileage will vary. Of prime importance is the need to make sure your network card IRQ, I/O address settings and shared memory addresses don’ t conflict with other cards in your computer.

Some more example configurations are described later in this document under the heading Sample Configurations for Packet Driver. 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 by selecting File/Run from the file manager, then type "tcpman". If this fails, the path is probably not set up correctly, so fix it. Later, you can setup tcpman as an icon so it can be started 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 the following details to enable the TCP package to function. 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” in lower case. If you use “ bootp” or “ rarp” , be sure to have a bootp or rarp 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 to be 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). Defaults to 4096 but can be larger.

TCP MSS TCP Maximum Segment Size (numeric). Usually MTU - 40. Use the default of 1460.

The rest of the details should be greyed out and you need not try to fill them in. The Internal SLIP and PPP check boxes 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 must 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 a message about not finding a packet driver or unable to load TCP, then check that the packet driver loaded properly; you can do this by observing the screen as the .bat file loading the packet driver executes. This will also tell you if WINPKT managed to find it. WINPKT will also tell you if the correct vector was chosen by 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. Examples are also provided of installation using NetWare. Some care is required to configure these protocols.

Possible causes for tcpman load errors specific to packet drivers:

unable to bind protocol 0806/0800 another TCP stack is using the packet driver... remove it.

WINPKT or pktdrv not foundcouldn't find the correct packet driver. Also check the vector number in TCPMAN (if you set it yourself rather than letting TCPMAN do it for you).

unable to allocate network buffers critical error... try to free up some special driver memory by removing windows device drivers. See section on low memory.

network buffers low not critical but unadvisable... try to free up some special driver memory by removing windows device drivers.

If WINPKT cannot 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 cannot get any response, you are probably accessing the wrong protocol. If you have the ARP trace on, you will possibly get "ARP timed out" messages. The first parameter of ODIPKT selects the correct protocol. Try adjusting this.

If you get the " unable to load network buffers" or " network buffers low" message, there are ways to free up more memory for the winsock. You can try providing more DOS memory before Windows is loaded by removing unwanted TSR' s or drivers. Also there can be occasions where Windows will attempt to load the full winsock into low memory resulting in insufficient driver memory being available. The workaround for this is to try loading the winsock at a different time. Often this error occurs when loading automatically at start up.

Anything else... contact Technical Support staff at Trumpet Software International, who will try to figure out what is wrong; but first browse the samples provided.By email: tech-support@trumpet.com.au

Scripting Language

Basic Commands

Each of the following commands is a scripting primitive and will return success or failure depending on whether the command succeeded or not. Commands always return success unless otherwise specified.

Note that the parameters enclosed in < > characters are compulsory for that command and those enclosed within [ ] are optional for that command.

abort Abort the script. This command always returns failure.

address <timeout> Parse the current stream for an IP address. This value will be copied into the Winsock' s primary IP address. The address change will not be effective until the script finishes. Returns success if an IP address was found, or failure if it timed out.

bootp Tell the winsock to attempt a bootp when the script has finished executing. This will normally only happen at the end of the login.cmd script and not the bye.cmd script. This will modify any TCP/IP setup parameters that have been specified in the returned bootp response.

display <string> Display <string> on the tcpman console display.

echo [on | off] Turn the echoing of received characters on or off respective. The default is ' on' which will display any received data while dialling onto the tcpman debug screen.

exec [<program command>]Execute the windows command specified. This will usually be to start up a program from the script, eg., pingw to check the connectivity.

expect <timeout> <target string>
Check that the current input stream contains the target string. It must match exactly, and will not search the stream. If the command timed out or the string received does not match, the command returns failure.

input <timeout> <target string>
Scan the current input stream for the matching string. The target string must be found for the command to be successful. If no string is found within the timeout period, the command returns failure.

load <variable> [<string variable> [<.ini file>]]
Returns <variable> which may be (integer or real) from <ini file>.
<string variable> is the section name within the ini file and defaults to: default vars. <ini file> is the name of the Winsock ini file and defaults to: trumpwsk.ini. Note that the file extension must be specified with the file name.

online Tell the winsock to go online immediately. SLIP packets will begin to be processed, and the state of the comms port will be adjusted to normal SLIP conditions.

output <string> Send the specified string to the output stream on the comms port. Any characters received during this process will be echoed to the display if the echo flag is on.

outputecho [timeout] <string>

Send the specified string to the output stream on the comms port. As each character is output, the script processor waits for the corresponding character to be echoed. If the received character does not match, the command will abort and a failure is returned. If a timeout is not provided, a default of 10 seconds is used. The timeout applies to the whole command. In addition, the carriage-return is handled specially by waiting for a linefeed to be read after the carriage return is read. Both the sequences <CR><LF> and <CR><CR><LF> are acceptable.

password [<prompt string>] Bring up a message box with the optional prompt for the password to be entered. Any characters typed will be echoed with a " * " character. If the user cancels the message box, the command returns failure. Any password entered will automatically be encrypted.

query <string variable> [<prompt string>]

Bring up a message box for the string variable to be entered. An optional prompt may be provided. If the user cancels the message box, the command returns failure.

quit Post a quit message to the winsock tcp manager. This will start a normal quit process.

read <timeout> <string variable>

Read the corresponding string variable from the comms input stream. A whole line is read. The line should be terminated by a line feed character. Any carriage returns are stripped. Returns failure if the command timed out.

save <variable> [<string variable> [<ini file>]]

Saves <variable> which may be (integer or real) to <ini file>. <string variable> is the section name within the ini file and defaults to: default vars. <ini file> is the name of the Winsock ini file and defaults to: trumpwsk.ini. Note that the file extension must be specified with the file name.

set (dtr | rts) [on | off] Set the modem controls DTR or RTS to the given state.

sleep <timeout> Pause for timeout seconds.

trace [on | off] Enable or disable the tracing of scripting commands. This command is very useful for debugging scripts that may be misbehaving.

username [<prompt string>] Bring up a message box for a username to be entered. An optional prompt may be entered. If the user cancels the message box, the command returns failure.

wait <timeout> [dsr | cts | dcd | rlsd]

Pause until the given modem signal changes state or a timeout occurs. If the command times out, a failure is returned.

String formats

When a string is required for a parameter, all characters up until the end of the command line are processed as the string. Strings conform to the following format.

A quote character (“) means that all characters are to be taken as is without any special meaning until a corresponding closing quote (“) has been found. The string is not permitted to extend over more than one line.

If a reverse slash “ \” is found, it denotes that the character following has a special meaning, or is to be taken literally. Here is a list of the special character meanings.

\b place a back space character into the string (control character number 8)

\c place the port number that the SLIP driver is using into the string as a decimal number.

\e place an escape character into the string (control character 27)

\f place a form feed character into the string (control character 12)

\i place the current IP address into the string.

\l place a line feed character into the string (control character 10)

\n place an end of line sequence into the string (control characters 13, then 10)

\p place the current password into the string.

\r place a carriage return character into the string (control character 13)

\t place a tab character into the string (control character number 9)

\u place the current username into the string.

\0 to \9 this denotes a decimal number of a character to be added into the stream. Up to three digits may be supplied. eg. \0\27 \255

If a ‘ #’ symbol is found that is not inside quotes, then it means that the rest of the logical line is to be ignored. This can be used to annotate the script.

If a ‘ $’ symbol is found, it denotes a string variable. If the variable has been assigned a value when the command is executed, it’ s value will be placed in the string. If not, an error message is displayed, and the script continues.

If a ‘ %’ symbol is found, it denotes an integer variable. If the variable has been assigned a value when the command is executed, it’ s value will be placed in the string. If not, an error message is displayed, and the script continues. Integers are 32 bit signed integers in the range -2147483648 to 2147483647.

If you wish to use the characters \, #, $ or % inside a string, they must be quoted with “ or have a \ character in front of them. The “ character may only be formed by preceding it with a \ character.

Program control commands.

In addition to the basic command primitives, there are the following special control statements.

If Statement.

if <condition>
  <statement list>
[else
  <statement list>]
end
If the condition evaluates as true, the first statement list is executed. If the condition evaluates as false, the first statement list is ignored, and if an else clause is present, it is executed instead.

While Loop.

while <condition>
   <statement list>
end

While the condition evaluates as true, the statement list is executed.

Repeat Loop.

repeat
  <statement list>
until <condition>

The statement list is repeatedly executed until the condition evaluates as true.

Assignments and Expressions.

Assignments are used to store new values into variables. They take the following forms.

<integer variable> = <integer expression>
eg. %attempts = %attempts + 1

<string variable> = <string expression>
eg. $name = $first + $last

The variable part may be a string variable (either unspecified or starting with a $), or an integer variable (starting with a %).

Integer expressions may be composed of the following operands.

numbers		eg.	1	43 7373
quoted strings	eg.	“ fred”	“ OK”	
string variables		eg.	$name	$response
integer variables		eg.	%I	%count

The following string functions may be used as operands.

copy ($s, %,p %l) Copy characters from string $s at character %p for %l and return them as an Note that string counting starts at 1 - not

len ($s) Returns length of string $s as an integer

lower ($s) Converts string $s to all lower case.

pos ($s1,$s2) Returns an integer number corresponding to the position of the first character of $s1 as found in string $s2. A zero is returned if $s1 is not found $s2.

upper($s) Converts string $s to all upper case.

Integer expressions use the following operators in order of priority.

( )
* / % meaning multiplication, division and modulo division.
+, - meaning addition and subtraction

String expressions may use the following operators: ( ) + concatenation

Also, integer operands may be formed by the comparison of strings.

eg. $A = " FRED" will evaluate to an integer operand of 0 or 1 depending on the value of $A.

Conditional expressions may also be formed using conditions and boolean operators.

These operators have the following priorities.

= <> < > <= >= equal, not equal, less than, greater than, less than or equal, greater than or equal. They return 0 or 1 on the result of the condition.

! boolean not operator
& boolean and operator
| boolean or operator

A more exact syntax is provided as an appendix.

Commands as operands

As well as the normal type of operands one may expect with expressions, one can also use a scripting command as part of an expression in the following manner by surrounding the command with the [and] symbols as follows.

eg. [input 10 OK\n]

This operand would have the value 1 if the command succeeded or 0 if it failed.
By the prudent use of such operators, quite sophisticated scripts can be formed.

For example, the following segment of script could be used to attempt a repeated dial of a given number. Note the use of the outputecho rather than output so that any characters echoed from the command will be consumed.

```
%attempts = 0
repeat
  %attempts = %attempts + 1
  outputecho 60 atdt345772371\r
until [input 30 CONNECT\n] | %attempts = 10
```

This section of script is fine, but would take 30 seconds for the input function to timeout if a response other than CONNECT were returned. It could be refined further into the following lines

```
%attempts = 0
repeat
  %attempts = %attempts + 1
  outputecho 60 atdt345772371\r
  read 30 $result
until $result = " CONNECT"  | %attempts = 10
```

This piece of script would be fine unless the modem failed to respond, in which case the script would abort. A further refinement would be the following.

```
%attempts = 0
repeat
  %attempts = %attempts + 1
  outputecho 60 atdt345772371\r
  %timeout = [read 30 $result]
  if %timeout
    display " Dial up timed out." \n
  end
until $result = " CONNECT"  | %attempts = 10 | %timeout
```

These portions of script are only examples to demonstrate the use of scripting. In practice, most modems do not generate simple messages after the atdt command. You will have to skip extra lines etc., to get a working script.

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 with 0.0.0.0 before dialling and try again. RARP is impossible to send via SLIP so 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 (an exact multiple if
)
on the IP trace to see if fragmentation is occurring on TCP
If so, then reduce MSS until it stops. UDP packets
still be fragmented, but nothing can be done about that. On the
TCP is type
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. Try quoting the strings to be sure.

For other problems, contact us at

tech-support@trumpet.com.au, or try the trumpet news groups. 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.
Specification of the -n switch of the packet driver is important. 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 by FTP Software, and the other, a public domain one. This example refers 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. Windows for Workgroups 3.11 Setup, courtesy of B. Armstrong and Douglas W. Jones.

There is an automatic 3.11 installer available free of charge from our ftp server:

ftp.trumpet.com.au:/ftp/pub/winsock/wfwsetup/twswfwg.zip

This is provided AS IS with no warranty and is for personal use only. You will also need to pick up a copy of of the DIS_PKT9 program. It currently only supports NDIS2, not NDIS3.

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

Ashok says...
“ 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." (Ashok Aiyar)

(Ed. Note... You may also require the use of a special comms buffer to enhance the buffering capabilities of Windows when using slipper/cslipper. A 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 that can be overridden:

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 archie.au (139.130.23.2), 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 accessable 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

The originating site for ODIPKT is...

Host newdev.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.

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 service on-line. will only if using Ethernet, and there is an 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 Suffixa space separated list of suffixes to be tried when looking up names via the name

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

Internal PPP click on this for internal PPP 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 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/Firewall setup calls up the firewall setup dialog

PPP options allows PPP authentication protocol to be configured

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

Special/Info displays a list of active sockets

Special/Kill Socket allows any socket to be killed; use with caution

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
the subroutine 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
Trace/Comms trace serial port communications

Trace/PPP trace PPP negotiations

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/Manual Login invoke the dialler manually. Use <esc> to exit
from the manual dialler

Dialler/Edit Scripts invokes notepad to edit any script

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 win
) Automatic login & logout must
enabled for this to close the SLIP
A value of 0 disables the

Automatic redial when
disconnected Redials once per minute while disconnected

Use standard SLIP settings for parity and word size

Use Control Panel settings for parity and word size
if your SLIP server does not accept 8 bits and no
parity when logging in
on the Use Control Panel

Any additional scripts will appear in the dialler menu

Help/About Display the version number and copyright

Trumpet General Discussion Group.

The machine newsroom.trumpet.com.au is now running a local news service with the news groups

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

You can ask questions, or discuss any aspect of any Trumpet program through these groups. Feedback is always welcome. There is also an anonymous FTP area at ftp. trumpet.com.au with all the latest Trumpet programs and pre-releases. However, our site currently supports a maximum of 20 concurrent ftp sessions - please be patient if you are unable to connect.

Bugs or Comments

Send to

tech-support@trumpet.com.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. Due to the high demand for the Trumpet Winsock, our mail box can be overloaded at times, therefore, priority will be given to registered users. We request that registered users identify themselves as such by including their registration name with all correspondence. Be patient... someone will answer you. You may also call us for assistance or send a FAX, our numbers are:

International Phone: +61-02-450220
International FAX: +61-02-450210

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 should register it.

The basic registration fee for a single user version of the Trumpet Winsock is US$25. See a later section for details on multi-user site licenses.
Australian users should contact Trumpet Software International regarding Australian pricing information and availability.

We accept MASTERCARD, VISA, BANKCARD, AMERICAN EXPRESS and DINERS CLUB. Credit card details may be given by e-mail, fax or telephone.

Trumpet Winsock Version 2.0 has a " Send Registration" option which will automatically post encrypted credit card details to Trumpet Software International. Select File/Register to take advantage of this feature.

Phone : International +61-02-450220, Australia (002) 450220
FAX: International +61-02-450210, Australia (002) 450210

We also accept Cheques which should be drawn in favour of:
Trumpet Software International Pty Ltd

and sent to:

Trumpet Software International Pty Ltd.
GPO Box 1649,
HOBART, TAS AUSTRALIA 7001

Please fill out the following order form and send it along with your payment to the above address.

Order Form

```
+---------------------------------------------+
|   O R D E R   F O R M   |
| for Trumpet Winsock version 2.0 Software|
+---------------------------------------------+
```

Ship to: Bill to:

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

Please supply the following items:

Licence to use Trumpet Winsock 2.0 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 31-Dec-1994

Single User license

1 user $25 US

Multi-user site license

A multi-user site license for Trumpet Winsock will be charged by the number of installed users.

The pricing structure for commercial users is thus:

1-99 users	$25 US per user
100-499 users	$2500 US + $10 US per additional user over 100
500-999 users	$6500 US + $5 US per additional user over 500
1000+ users	$9000 US + $2 US per additional user over 1000

site restriction 10km radius (negotiable)

Unlimited Commercial Site License

$12,500 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	$25 US per user
100+	$2500 US

site restriction limited to a campus

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 current 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 Pty Ltd reserves the right not to accept any offer which is not considered acceptable.

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

Phone: International +61-02-450220, Australia (002) 450220
FAX: International +61-02-450210, Australia (002) 450210