

WSNWDEMO
Version 1.0 (Alpha)
Copyright ©1994, Mark Clouden

WSNWDEMO is distributed as 'freeware' with only one exception; the DLL files that are distributed are a part of a SDK named WSNETWRK. This SDK is still in the Alpha test phase, but will likely be made available as a commercial product at some time in the near future. Therefore, **any distribution of information pertaining to the use of, or the actual use of the DLL files for anything other than as runtime libraries for WSNWDEMO would be considered a breach of the common law copyright claimed by the authors.**

The files in question are; WSNETWRK.DLL, ICMPSOCK.DLL and ECHOSOCK.DLL.

Standard restrictions:

1) No commercial use of WSNWDEMO may be made. WSNWDEMO is distributed free of charge, and shall remain free to anyone who desires a copy of it. No profit of any kind may be made from the distribution or use of WSNWDEMO.

2) The authors will not be held liable for any use or misuse of the WSNWDEMO package. Any damage caused by it's use is solely the responsibility of the user.

And finally:

As should be evident by the use of the copyright symbol, the authors reserve ALL rights to the WSNWDEMO application as well as any related materials.

Within the restrictions described above, WSNWDEMO may be distributed and used as much as you would like.

Contacting the authors:

Comments are always welcome, as are inquiries into the availability of the WSNETWRK SDK. The authors may be reached via internet EMail at **icode@teleport.com**

Known Bugs:

- 1) When a TCP server is started and accepts a connection - If the new connection is still active when the server is stopped, then when the server is restarted a bind error occurs which keeps the server from being created. *This may be the way this is supposed to work but I doubt it. SO_REUSE is enabled for the server socket and for the accepted socket, but this has no effect (I didn't think it would).*
- 2) Better notifications of errors, including connect failures
- 3) Periodically seeing times that are not likely to be correct

About WSNWDEMO:

The purpose of WSNWDEMO is to demonstrate the capabilities of an application designed with the WSNETWRK SDK. It has the added benefit of being at least marginally useful as a common utility providing Echo testing and a Finger protocol client.

Echo testing in general refers to both the ICMP 'Ping' and the ECHO client tools. Not every WINSOCK provides support for ICMP (or RAW) sockets, so ECHO, using either TCP or UDP sockets was added to fill this void.

On the opposite end of an ECHO (not a 'Ping') sits a server that does the echoing. One of these was provided as a tool, and it supports both TCP and UDP clients. (ICMP packets are handled directly by the TCP/IP protocol stack - there is no server).

The Finger client was added shortly before the release of WSNWDEMO, and should be considered still a test. There really isn't too much to it though, and it should prove to be fairly stable (The bug in the display dealing with the last column on the right sometimes being lost is not mine - All my MFC CEdit objects do that).

ECHO And Ping Statistics:

Echo Attempts:	The number of actual attempts at sending a packet
Echo Failures:	Count of packets for which we never received a reply
Late Replies:	Count of packets previously falling under the failure category but for which we received a reply outside of the interval window
Error Count:	Count of low-level send/recv errors
Average Time:	Includes only successful send/receives
Minimum Time:	Includes only successful send/receives
Maximum Time:	Includes only successful send/receives

All times are in microseconds (1/1000th's of a second).

Average, minimum, and maximum times include some processing overhead. They are not to be used as measurements of a network's speed (Although most Ethernet networks should have an average time of 0 - not measurable - and a standard SLIP connection over 14400 modem comes in around 150).

ECHO and Ping Timing:

The interval which may be set must be between 3000 and 60000 (3 and 60 seconds). This is how often an echo attempt will be made. If when it is time to send the next packet the previous packet has not been replied to, it counts as a failure and the new packet is sent.

ECHO - TCP Connected or UDP sockets:

If the checkbox is highlighted then the socket created will use TCP. You will receive connection errors if the host is unavailable. If the checkbox is not highlighted, the socket created will use UDP. If the host is not available, or there is no ECHO server running there, you will only see a continuous stream of failures as a UDP socket does not get connected.

ECHO Servers:

There is button to start and stop both a TCP server and a UDP server. Stats for each are kept also in the form of the number of echo replies sent.

A TCP server will spawn off child sockets that will be represented by the 'number of servers' item in the window. The server may stop listening at any time, even with sessions in progress. The child sockets will continue to function even if the server is not accepting new connects (not listening). Note the bug? detailed above. If you stop listening while child sockets are active, you will not be able to resume listening until they are closed (do so by closing the ECHO Server window itself - there is no button to kill the child sockets explicitly).

Bit O' Trivia: The reason the child sockets were left intact and not stopped explicitly by a button on the window is to test the robustness of WSNETWRK in detecting when the owning window or application goes away. This way WSNETWRK can release all resources associated with a socket when the owner ends - even if the owner forgets to close the sockets, or if the application GPF's.

Finger Client:

Not every host server accepts the Verbose option. Some don't even accept the user name. For these you will need to clear the Verbose checkbox and/or leave the user name blank and get the full listing.

Some servers send back a listing with only CR to terminate a line. Many more send a CRLF pair. The multi-line edit box in this window needs a CRLF pair, so we translate most server output to make sure that it bends to our will. This will not *always* work, just most of the time.

Maintaining host lists:

The list of host names and any saved settings for each host are saved in an INI file named WSNWDEMO.INI. I did not put much in the way of validation in the loading of these INI entries, so muck about with them at your own risk.

Use the New Host button to pop up a one line text entry dialog box that accepts the name of the new host. It will then be available in the list box.

Use Delete Host to remove the currently selected host name from the list box.

ECHO Client is the only one right now that saves more than the host name. It also saves the last TCP/UDP socket type and the alternate port number (for non-standard port assignments on the host).

Multiple Windows:

The reason these tools are in an MDI window was to see just how much could be pumped through a single application at once. Personally I prefer the MDI approach anyway.

Create as many child windows as the operating system will allow. They can be talking to as many hosts at the same time as there are child windows.

Mark Clouden
17 May, 1994