

***VORSIS* Application Note**

AN2007-01A

Resolving Graphical User Interface (GUI) Network Connectivity Issues

Original October 2007 – Jeff Keith
Updated July 2008 to Add New Vorsis Products

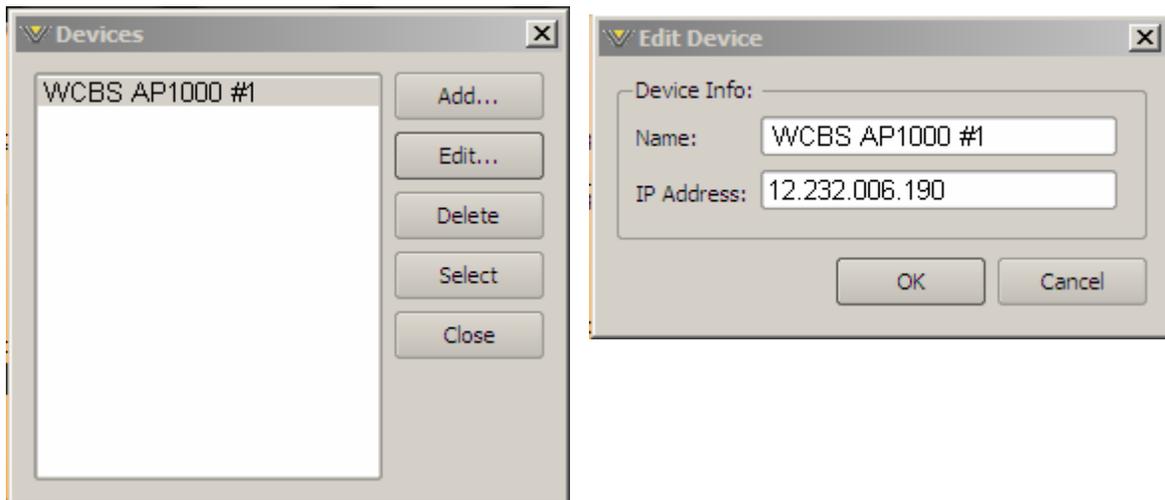
Wheatstone Corporation © 2008 All Rights Reserved
600 Industrial Drive New Bern, NC 28562 USA 252-638-7000
techsupport@wheatstone.com

NOTE: This document has been updated to include the new Vorsis products released in 2008.

Background

On occasion there may be difficulty experienced when connecting the Vorsis Windows® GUI to one of our products. When this occurs the “Trying” status message will remain illuminated at the top of the GUI application and will blink as the GUI retries the connection.

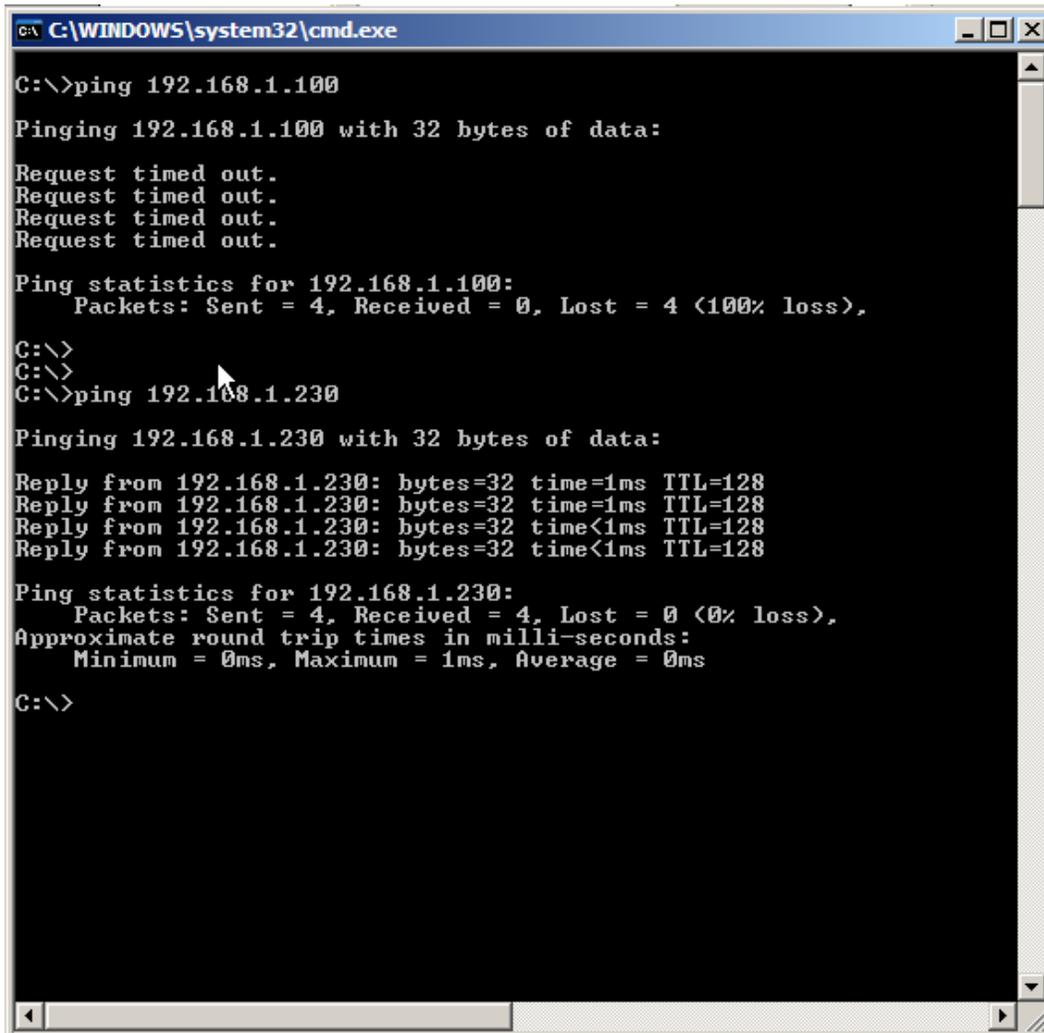
When such difficulties are experienced it is prudent to carefully review the Device configuration settings within the GUI to ensure that the Vorsis device destination TCP/IP address is consistent between the GUI and Vorsis device. There should be no blank spaces or illegal punctuation (such as a comma instead of a period) between the four octets of the IP address.



It is also necessary for the controlling PC and the Vorsis product to be on the same network subnet. For instance, if the Vorsis device is assigned IP address 192.168.1.194, then the PC’s own TCP/IP address must be between 192.168.1.1 and 192.168.1.254, noting that the Vorsis device and GUI PC cannot share the same TCP/IP address.

Note that it can be helpful (for us humans) to include leading zeros in the IP Address entry; For instance, 12.232.006.190 in the above example could be entered as 012.232.006.190 instead.

It can also be helpful to utilize the operating system's Ping command to see if the host PC can 'see' the Vorsis device on the network. The following shows two 'ping' examples; the first 'ping' shows what happens when the device is not reachable on the network while the second 'ping' attempt shows a successful 'ping' reply from a networked device.



```
C:\WINDOWS\system32\cmd.exe
C:\>ping 192.168.1.100
Pinging 192.168.1.100 with 32 bytes of data:
Request timed out.
Request timed out.
Request timed out.
Request timed out.
Ping statistics for 192.168.1.100:
    Packets: Sent = 4, Received = 0, Lost = 4 (100% loss),

C:\>
C:\>
C:\>ping 192.168.1.230
Pinging 192.168.1.230 with 32 bytes of data:
Reply from 192.168.1.230: bytes=32 time=1ms TTL=128
Reply from 192.168.1.230: bytes=32 time=1ms TTL=128
Reply from 192.168.1.230: bytes=32 time<1ms TTL=128
Reply from 192.168.1.230: bytes=32 time<1ms TTL=128
Ping statistics for 192.168.1.230:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
    Approximate round trip times in milli-seconds:
        Minimum = 0ms, Maximum = 1ms, Average = 0ms

C:\>
```

The above shows that when a device cannot be found on the network and its address is 'pinged' the Ethernet packets sent from the host PC are 'lost'.

When a pinged device is found the output of the 'ping' command reports a reply from the device. It also displays the round-trip transit time of the packets to and from the pinged device which is a measure of the delay within the network between the host PC and that device.

Vorsis Device Network Connectivity – Peeking Under The Hood

Vorsis audio processor products use both the TCP and UDP protocols for communication with the host computer. Processing control adjustments and preset datasets use TCP/IP for its bit-for-bit reliable connection between the Vorsis device and the host PC.

The UDP protocol used for metering data has no handshaking and since ‘normal’ networks rarely have perfect connectivity this means that data contained in dropped UDP metering packets will not be recovered.

We’ve used the benefits of both protocols to minimize network traffic; the 100% reliable TCP protocol handles control adjustments and preset data, and the less reliable (handshake-less) UDP protocol handles metering data.

Vorsis Product TCP and UDP Ports and default IP addresses

All Vorsis audio processing products use UDP ports 60001 through 60010 for transporting metering data. Different TCP ports, however, are utilized for “controlling” each Vorsis product and each has a Factory assigned default IP Address:

| Vorsis Model | Default IP Address | TCP Port |
|---------------------|---------------------------|-----------------|
| AP3 | 192.168.1.191 | 55888 |
| AP1000 | 192.168.1.200 & 201* | 55888 |
| HDP3 | 192.168.1.191 | 55889 |
| VP8 | 192.168.1.198 | 55890 |
| M1 | 192.168.1.190 | 55891 |
| FM5 | 192.168.1.194 | 55892 |
| AM-5HD | 192.168.1.195 | 55893 |
| FM-10HD | 192.168.1.210 | 55894 |
| AM-10HD | 192.168.1.211 | 55895 |
| AP2000 | 192.168.1.220 & 221* | 55896 |
| FM2000 | 192.168.1.230 & 231* | 55897 |

*the AP1000, AP2000, and FM2000 utilize two IP addresses – one for the main board and one for the front panel GUI controller.

In each case the Vorsis processor device is the “server” and the host PC is the “client” and each Vorsis audio processors can accept up to 4 simultaneous GUI connections back to host PC’s. This capability can be quite useful for ‘collaborative’ tweaking sessions.

How We Use the TCP and UDP Ports

Once a TCP connection is established the remote GUI running on the host PC opens a UDP port and asks the Vorsis device to send its metering data on that port. The first UDP port tried is 60001, but it is not always possible for the GUI to use this port. This is because with UDP ports only one application on a PC may open a specific port at a time.

Since we do not know (and cannot easily tell) what ports might already be in use by other applications on the host PC, and because we also want to allow more than one GUI to run at a time on any given host PC, the GUI has ten UDP Ports that it is allowed to use.

It begins by trying to open Port 60001, the first UDP port in the allowed range. If UDP Port 60001 cannot be opened, it is assumed to be busy and the GUI will then try to open Port 60002, then Port 60003, etc., up to Port 60010. Whatever port successfully opens first, that port number will be sent to the Vorsis device, and the Vorsis device will begin to stream its meter data back over that UDP port to the host PC running the remote Windows[®] GUI.

Note: if there are *two* GUI’s attached to a Vorsis device, the Vorsis device knows this (because the connected devices have different MAC addresses) and therefore streams its meter data *twice* – once to each of the opened UDP ports associated with the unique MAC addresses of the PC’s hosting the two connected GUI’s. The same sort of thing happens if more than two GUI’s are connected to the same device, up to a maximum of four GUI’s.

If the GUI fails to open *any* of the ten UDP ports the GUI will still be able to go “Online” and control the Vorsis device via the TCP port, but it will *not* receive any meter data to display. Therefore if you have a network connection that appears to control the processor but there is no metering displays on the remote Windows[®] GUI, check to ensure that there is no

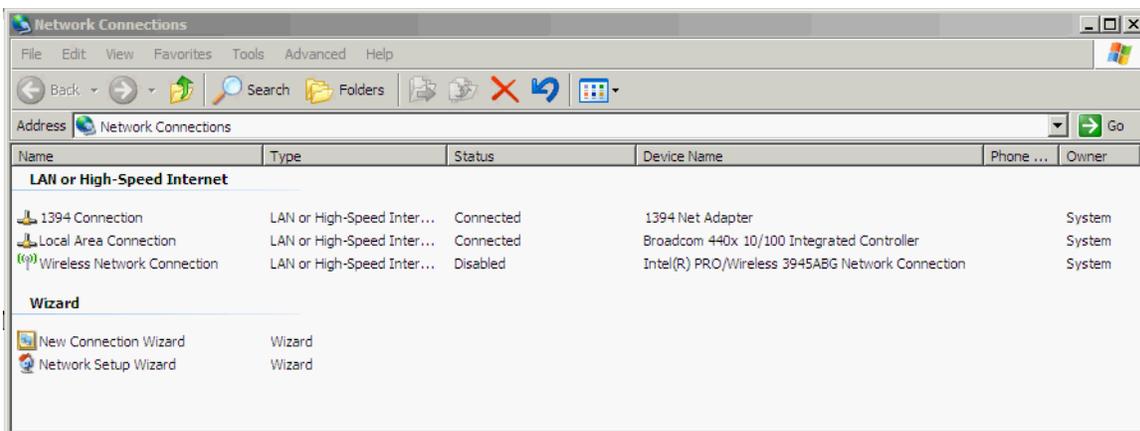
firewall or other blockage on the PC or network preventing the use of UDP ports 60001 through 60010!

In summary, network traffic on the TCP Port assigned to the Vorsis product *and* UDP Ports 60001 through 60010 *must* be allowed to traverse the segment of the network between the host computer and the Vorsis device or they will not be able to fully communicate with each other.

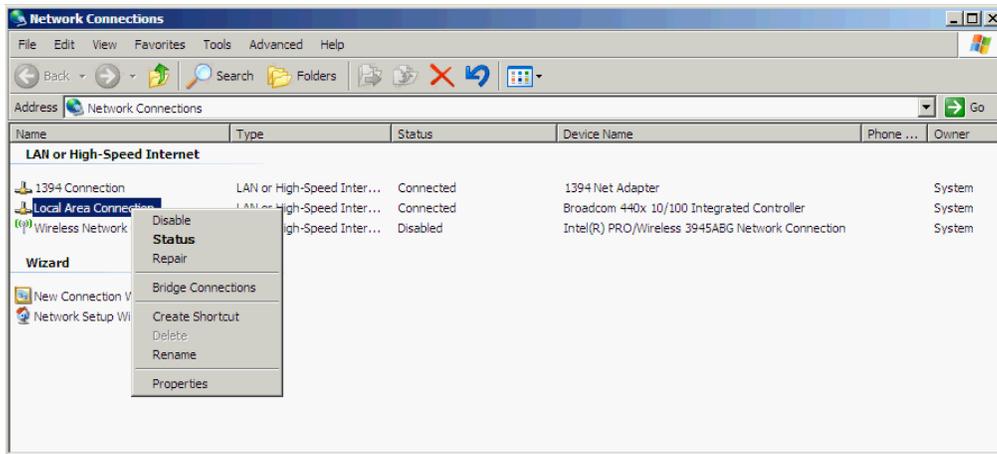
A WindowsXP Networking Issue:

We've observed that both Service Pack 1 and Service Pack 2 of WindowsXP® may sometimes exhibit inconsistent behavior during attempts to connect to another networked device (such as a Vorsis processor with the Windows® GUI). Usually this is due to old data residing in the PC's ARP (Address Resolution Protocol) cache – the first place that Windows® looks to for how to connect to an externally connected network device.

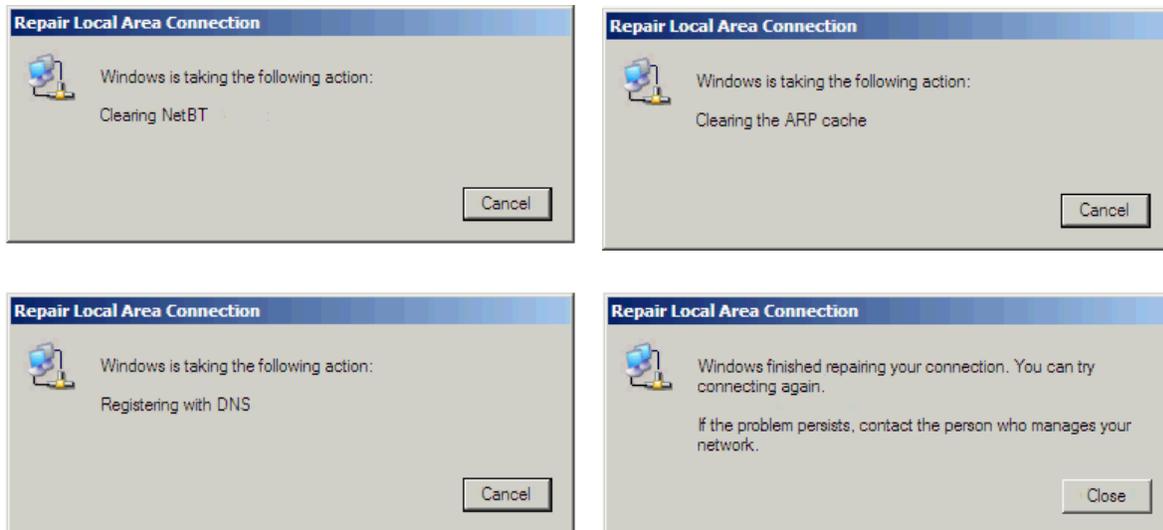
The old data in the ARP cache can be cleared manually at a command line, but it's much more easily done using the built in tools within Windows®. To do this, open the Windows® Network Connections dialog and a window similar that shown below will open.



Next, highlight the “wired” Local Area Connection and then right click it. A drop down menu similar to the following will appear:



One of the options in the drop down box is “Repair” and you should select it. What this option does is reset the PC’s Ethernet interface¹ – in essence ‘refreshing’ it with new data. During this process a message sequence similar to the following should occur:

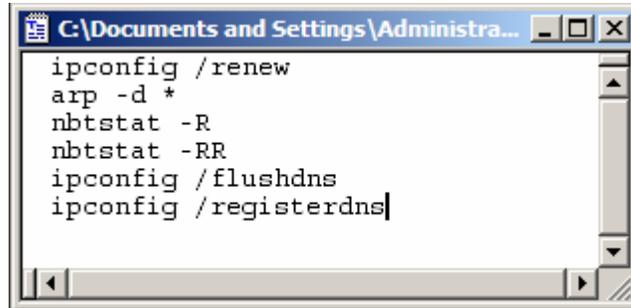


The process usually completes in a few seconds and when it does the network connection between the host PC and the Vorsis device should be fully functional.

¹ See “What Repair Connection Does”.

What the “Repair” Option Does

There are six actions that occur automatically when the “Repair” option is selected, replacing six manual commands that would need to be sequentially entered at a Windows command prompt in order to accomplish the same thing (see the graphic below for a batch file we call “fixnet.bat” that we use in our Vorsiis lab to circumvent this issue many times during a routine day).



```
C:\Documents and Settings\Administra...
ipconfig /renew
arp -d *
nbtstat -R
nbtstat -RR
ipconfig /flushdns
ipconfig /registerdns
```

The commands listed in the above graphic are shown in bold text below and accomplish the following:

- First, the Dynamic Host Configuration Protocol (DHCP) lease is renewed:
ipconfig /renew ***
- Then the Address Resolution Protocol (ARP) cache is flushed:
arp -d *
- Next it reloads the NetBIOS name cache:
nbtstat -R
- Then the NetBIOS name update is sent:
nbtstat -RR
- Next, the Domain Name System (DNS) cache is flushed:
ipconfig /flushdns
- Lastly, the DNS name is re-registered:
ipconfig /registerdns

*** This step is necessary even if DHCP is not being used by the host PC!

Conflicts between Wired and Wireless Networking

If the networking between the host PC and the Vorsis device still does not work, there may be other issues. In fact, if you have both Wired and Wireless Networking installed on the host PC there can be a conflict that is causing Windows[®] to not know *which* network path to utilize to get to an external network device. In this scenario it is quite common for Windows[®] to try to utilize the *wireless* connection first when in fact the device you wish to communicate with is connected to the *wired* network.

We have noted in our own research lab, and customers have reported to us, that if a host PC (typically a laptop computer) has both wired and wireless connections configured and both are ‘enabled’, sometimes Windows[®] can’t seem to tell which network connection to use. When this happens, the Windows[®] GUI refuses to go online even though there is known good wired network connection between the Vorsis device and the host PC.

To alleviate this you should temporarily disable the wireless interface on the host PC, connect to the Vorsis device using the Windows[®] GUI, and then re-enable the wireless interface once “wired” communication has been established.

This quirk of the Windows[®] operating system has been reported to us more often when the host PC is running versions of WindowsXP[®], than when Windows 2000[®] or Windows Vista[®] is the operating system, although the latter may be due to the relatively small Vista[®] user base as of this writing.

In summary, the ‘disable the wireless interface’ and/or ‘Repair’ actions have worked successfully in every instance where we have encountered this behavior *if there is a good network connection between the Vorsis GUI on the host PC and the Vorsis audio processor.*

If all else fails, then some friendly assistance from your local IT guru or even a call to Vorsis technical support might be required.

Vorsis[®] is a registered trademark of Wheatstone Corporation
600 Industrial Drive, New Bern, North Carolina 28562
252-638-7000
www.vorsis.com