

Windows



Connect, Inc.
1701 Quincy Avenue, Suites 5 & 6, Naperville, IL 60540
Ph: (630) 717-7200 Fax: (630) 717-7243
www.connectrf.com

Table of Contents

Placing Static IP Entries into NT Platform ARP Table	1
AirLinc on Windows NT/2000 in a Novell Environment	3
NT/2000 and Multiple NIC Adapters.....	4
Inability to Start 19-20 Sessions on NT/2000	5
Windows 2000 Errors on Starting PowerNet	6
Telxon Windows 2000 Radio Parameters for PowerNet	7
Inability to See Administrator PowerNet OpenAir NT/2000	8
Windows 2000 & NT Permission Problems	9
How to Configure Telnet Server in Windows 2000.....	11
AS400 Alarm Function.....	14
The Windows OS Will Not Stop the MU Processes in OpenAir	15
Windows 32 Standard/Socket Error Listing	16

Placing Static IP Entries into NT Platform ARP Table

Problem Description

Disconnects are caused by the NT Operating/network system when operating PowerNet in an NT platform environment, especially in a switched IP environment.

Configure the switches so that they pass the MAC address of the RF terminals in use. RF terminals that operate in a switched environment perform their cell change at about 60ms. Unfortunately, the switches themselves most likely do not update each other's tables that fast, or they need to be configured to do so. Otherwise, you may find that the RF terminals that move between cells and cross switches or ports on the same switch will be disconnected by the switch.

Once the switch configuration is done, a secondary issue with NT and switches needs to be addressed. If the NT platform network does not hear from an address within 4.7 seconds time, it will query the MAC address with an ICMP packet. At the end of that time, the NT network platform removes the MAC address from the ARP table, resulting in random RF terminal disconnects. The operator of the RF terminal continues to work, and sends the next data to the application, but there is no longer a session with NT. This forces the RF terminals to enter the "Waiting for Data" state, and an RF timeout causes the operator to reboot the terminal.

Resolution

To solve this issue with NT networking, specify static entries in the NT ARP table for each RF terminal being used.

Log in all RF terminals and run the "ARP -a>>arp.lst" to create a list of all the ARP/MAC addresses for the RF terminals. Edit the list file so that each MAC address becomes a static entry. Save the file as an "arp.bat" file.

Run the "arp.bat" file to add the entries in the static table. Add the "arp.bat" file to NT's start up processing to ensure that the entries get added on an NT reboot.

Remember to update this file and run it to update the entries as RF terminals are added and/or repaired

ARP -a [inet_addr] [-N if_addr]

- a Displays current ARP entries by interrogating the current protocol data. If inet_addr is specified, the IP and Physical addresses for only the specified computer are displayed. If more than one network interface uses ARP, entries for each ARP table are displayed.
- g Same as -a.
- inet_addr Specifies an internet address.
- N if_addr Displays the ARP entries for the network interface specified by if_addr.
- d Deletes the host specified by inet_addr.
- s Adds the host and associates the Internet address inet_addr with the Physical address eth_addr. The Physical address is given as 6 hexadecimal bytes separated by hyphens. The entry is permanent.

eth_addr Specifies a physical address.
if_addr If present, this specifies the Internet address of the
interface whose address translation table should be modified.
If not present, the first applicable interface will be used.

Example:

```
> arp -s 157.55.85.212 00-aa-00-62-c6-09 .... Adds a static entry.  
> arp -a .... Displays the arp table.
```

AirLinc on Windows NT/2000 in a Novell Environment

Problem Description

The client's server is in an NT environment. However, it is the only machine in the NT environment. All other machines reside in a Novell network. The Access database to which the user wants to connect through ODBC in AirLinc is located on a machine in the Novell environment. There is an apparent permission problem with MS ODBC pulling data from Novell into Windows 2000.

Resolution

Move the Access database to the NT machine.

NT/2000 and Multiple NIC Adapters

Problem Description

When PowerNet installs, it defaults to the first NIC adapter found in the Registry that may or may not be the correct segment where the AP's are connected. NT/2000 gets confused as to which adapter is the default for the network service when the name of the machine and the host list entry is the same on multiple adapter machines.

Resolution

Select the correct settings from within AP server setup, save them, and then stop and restart the AP server. The machine name must be unique from the adapter address in order for apbase to run.

Note: Verify that a NT/2000 service is not running on port 1800 from the target segment of the AP's.

Customer suggestion

The service running on the NT/2000 server is bound to an address other than that specified in Access Point---> Server. Change the terminal to use the other address in the host list, and it should connect on the next try. This issue occurs when an NT/2000 server has multiple IP segments attached. This issue is not expected when an NT/2000 server has only one IP segment attached.

Inability to Start 19-20 Sessions on NT/2000

Problem Description

An attempt to start the 19 or 20th sessions results in TAP terminated.

Windows NT and 2000 have a setting buried in the registry that controls how much memory is available to non-interactive applications for user interface components like windows and menus. Non-interactive applications are programs that start as services that do not have the "Interact with desktop" option selected. Apbase is a non-interactive application in this sense. Since apbase kicks off VTERM that in turn kicks off AirLinc, they all are non-interactive applications.

The catch is that Windows NT defaults the amount of memory for these components (known as the desktop heap) to 3 MB. Windows 2000 defaults it to 512 KB. That is why the problem exists on 2000, but not on NT. If you raise the value in the registry on 2000, it functions as normal.

Resolution

To change the value, run regedit and open the following key (folder):

HKEY_LOCAL_MACHINE\System\CurrentControlSet\Control\SessionManager\SubSystems

Edit the value of the field called "Windows". It contains a bunch of values strung together.

Near the beginning, identify one that looks like "SharedSection=1024,3097,512,512".

The third value (512) needs to be increased. Using 3097 is a tested solution. Reboot after making the change.

Note: You may see different values in the other positions of SharedSection, and there may not be a fourth value. It is the 3rd value that requires modification. Do not be concerned if the other values are different than this example.

Note: An upper limit may exist. One should contact Microsoft for a description of what that may be and how it affects memory utilization under their Operating Systems.

Windows 2000 Errors on Starting PowerNet

Problem Description

It has happened that PowerNet does not start on Windows 2000 and CGI script errors are received when using the browser to administer PowerNet. A Windows error box pops up with the following "abase.exe application error. The application failed to initialize properly 0xc0000142 click ok to terminate the app.". Errors can also occur with remotesm.exe and S1base.exe.

Resolution

PowerNet was not installed under Windows 2000 correctly. To correct this as the root user on the 2000 console, Select Start, Control Panel, Administrative Tools, Services. Stop all PowerNet processes. Right Click each process select the LogOn tab and make sure you have selected both "Local System Account" and "Allow service to interact with Desktop".

Telxon Windows 2000 Radio Parameters for PowerNet

Problem Description

Telxon Windows 2000 requires radio parameters for PowerNet.

Resolution

Set access points up as follows:

AIR I/O AP2000E V5.21 Configuration Radio Menu AP2000E_2162f9

Option	Value	Description
1 - Sid	[2]	- System identifier
2 - Bitrate	[2000]	- Data bit rate in kilobits / second
3 - Frequency	[2442]	- Center frequency in MHz
4 - Root	[on]	- Enable root mode
5 - Autoscan	[on]	- Enable auto scan mode
6 - Install	[menu]	- Installation utilities
7 - Extended	[menu]	- Extended parameters

Enter an option number or name, "=" main menu, <ESC> previous menu.

Settings in the c:\PowerNet\twinclient\terminal\Telxon\8660-960\2000\pktds.def file:

```
[SpreadPacket]
BitRate=0
Channel=3
RadioType=0
RadioId=0xFFFF
RouterId=0x0000
SystemId=0x00000002
MaxDatagram=2048
MaxRetries=0
ReceiveMode=0
RootRepeater=0
SpecifiedRouter=0x000000000000
FastPoll=0
FastPollDecay=0
FastPollDelay=0
ARL_Threshold=0
ARL_Decay=0
SpecRegTimeout=0
RegistrationMode=2
RegistrationFill=0
SlowPoll=0
RefreshRate=0
NetworkId=0x000000000000
LocalTalkAddress=0x00
CodeFormat=0
NumChannels=0
Channel1=0
Channel2=0
Channel3=0
Channel4=0
```

Inability to See Administrator PowerNet OpenAir NT/2000

Introduction

There sometimes occurs the inability to see Administrator PowerNet OpenAir NT/2000.

Problem Description

After installing the PowerNet NT/2000 OpenAir product, you cannot access the user interface for administering the PowerNet product. The difficulty seems to arise within the OS as to how the Network adapter and the computer are identified as nodes.

Resolution

After starting the administrator with the WEB browser, specify the IP address of the Network adapter you are using for the RF link. Example: Replace the 127.0.0.1 with the address of the NIC adapter 207.241.78.225.

Note: Care should be taken if multiple NIC adapters are installed.

Windows 2000 & NT Permission Problems

Introduction

The OpenAir NT product has a Browser Interface. It will launch the default WEB browser to run the OpenAir Administrator.

We have seen problems where the browser will not have full permission to access files and/or services under NT or WIN 2000.

This can cause a variety of errors to include:

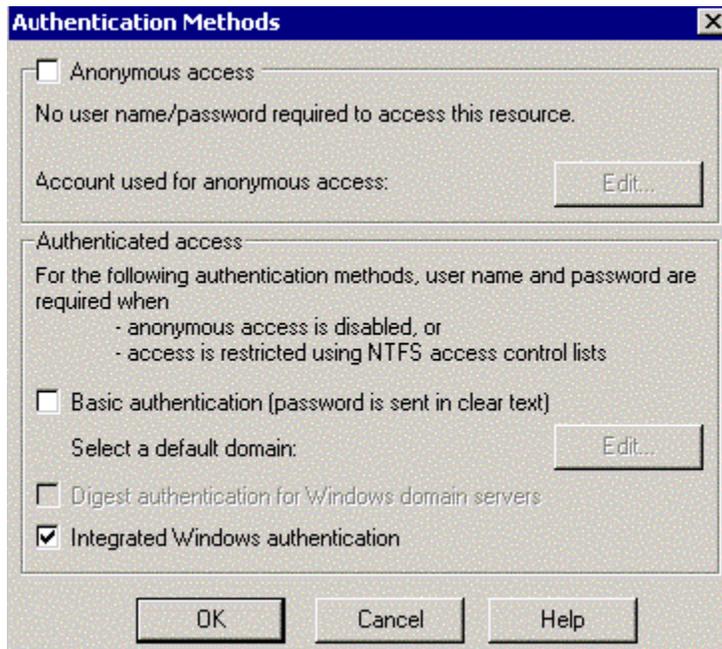
- Inability to run OpenAir Administrator from the browser - Will fail with an "Unable to access".
- Inability to enter an Authorization code - Will complain of "Invalid Code" when in fact, it is fine.
- Inability to setup local TCP/IP address - Will complain about the inability to move the "host file".

Resolution

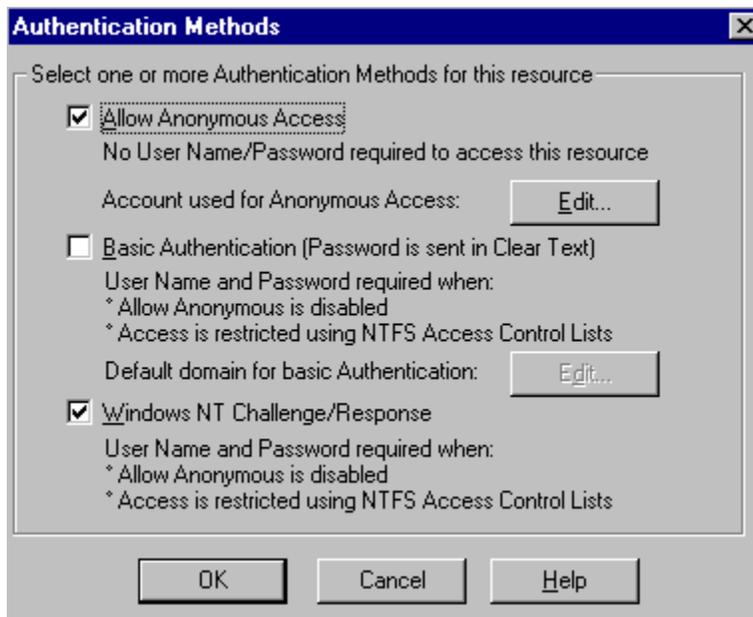
Turn off Anonymous Access for the WEB server - Either IIS or Peer to Peer.

When you run the OpenAir administrator, it will now prompt for User ID and Password. If you use the administrator password, all functions will work.

Note: The screen shot below is from the WEB setup under WIN 2000 for reference. It is found under Control Panel/Administrative tools/Internet Information Services/WWW Service Master Properties/Authentication Methods.



Windows NT Screen Shot



How to Configure Telnet Server in Windows 2000

Introduction

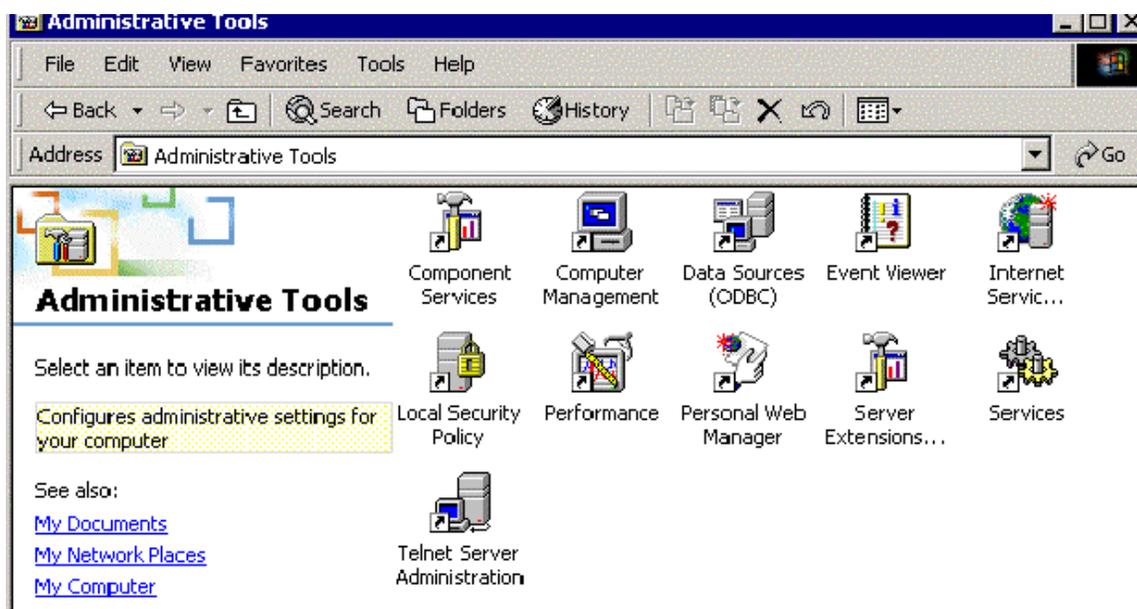
The following are instructions for configuring the Telnet server under Windows 2000.

Problem Description

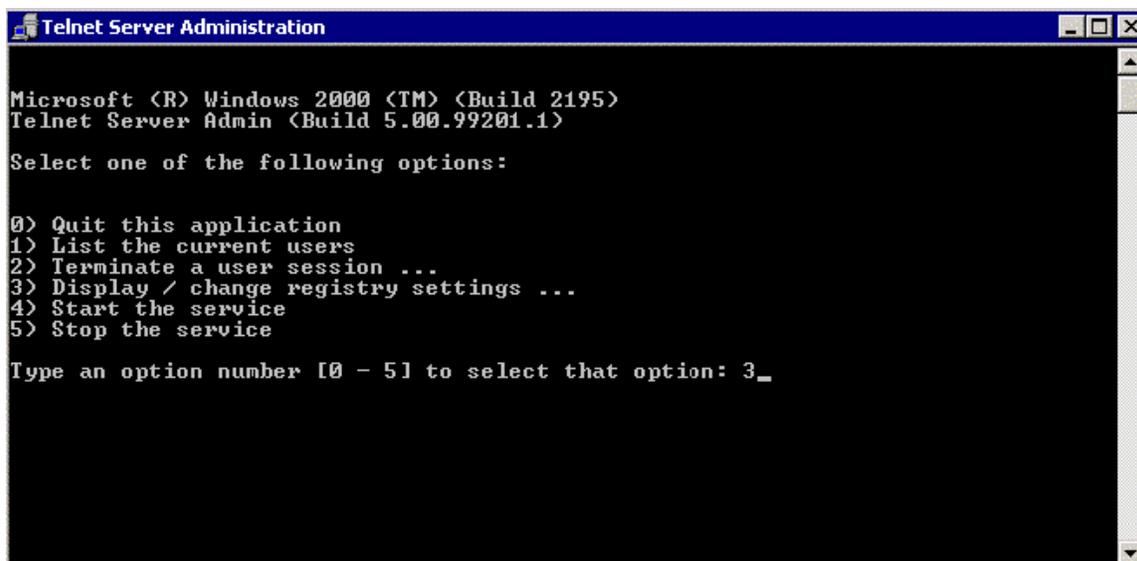
The terminal does not communicate with the Windows 2000 Telnet server.

Resolution

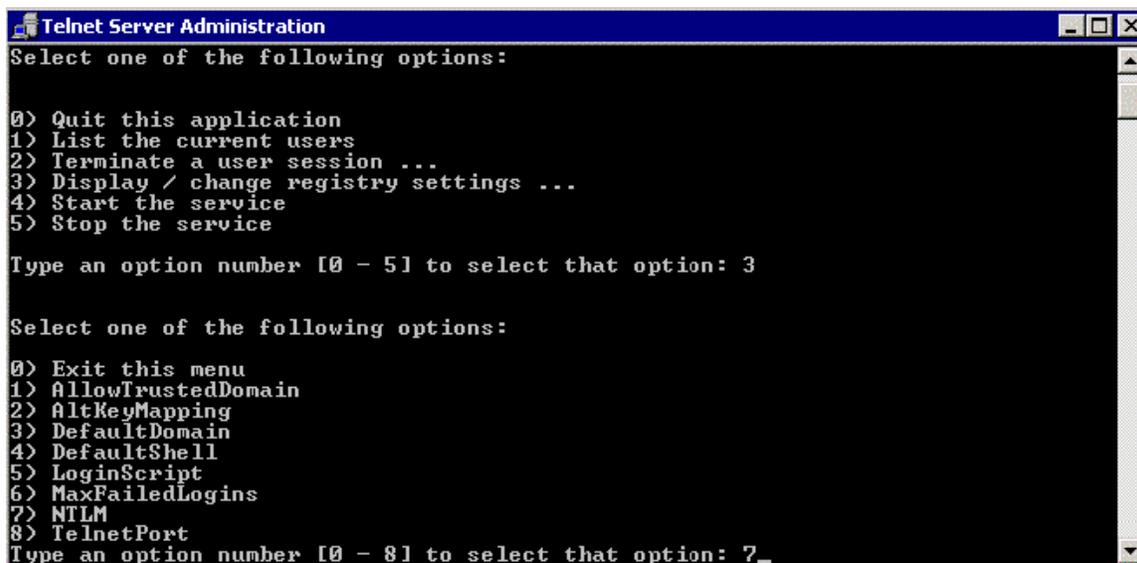
Select START, then Control Panel, then Telnet Server Administration.



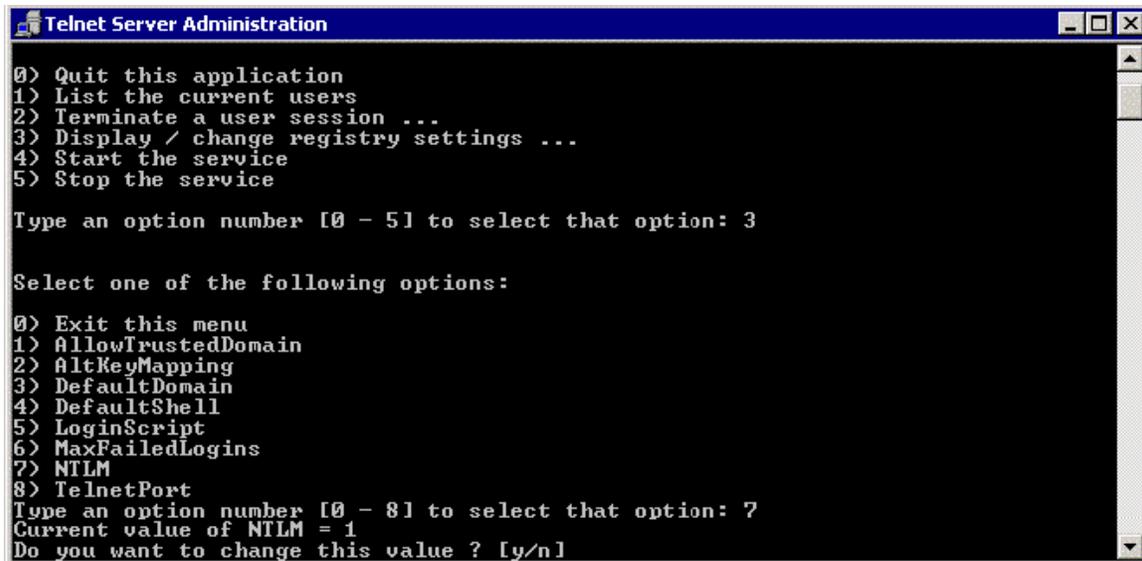
This should bring up the following window and select 3 to change the settings.



Now select the NTLM (NT LAN Manager) option (7):



Change the setting to 1 and save it.



Now Stop and Start the Windows Telnet server.

AS400 Alarm Function

Introduction

The following is a sample display file program for the AS400 Alarm Function.

Resolution

```
A      R VERSION
A*%%TS SD 20000803 083012 QSECOFR  REL-V4R2M0 5769-PW1
A              OVRDTA
A              PUTOVR
A              BLINK
A              ALARM          <= to beep on a terminal
A              1 2'Validation Tests'
A              2 3'5250/3270 V1.0'
A              3 1'Enter'
A              3 8'to'
A              3 11'Continue'
A      SSS1      2A B 4 8
```

The Windows OS Will Not Stop the MU Processes in OpenAir

Introduction

The protocol manager (apbase or s1base) cannot stop the handler processes in OpenAir.

Problem Description

We have found that, under the right conditions, a kill signal issued by the protocol manager will be ignored and the process will not end. This can happen when the system is busy, overloaded, etc. We have made the kill process more aggressive for these conditions.

The terminal trace will usually show this type of line: *ERROR rxxx.xxx already active*

Solution

Try the DLL and apbase.exe at the links below. These have been successful on problems with the handler processes hanging (rfvt.exe, 5250tn.exe, and 3270tn.exe).

The new SMI driver goes in the System32 directory. (The SMI device is our program-to-program data link, similar in function to IPC-like shared memory, pipes, etc.)

You will have to make sure to shut down all of the processes before copying it in. This includes RemoteSMI.exe which you have to stop using Control Panel / Services.

<http://www.connectrf.com/Documents/Crflpc.zip>

<http://www.connectrf.com/Documents/apbase.zip>

Windows 32 Standard/Socket Error Listing

Introduction

The following documents the standard/socket Windows 32 error numbers.

These error numbers and meanings are common across most versions of Windows. They are returned to a running application if there is an error with a Windows system or library call.

You may, at times, see these show up in PowerNet trace files.

Platform SDK

Win32 Error Codes

The following tables provide a list of Win32 error codes:

<http://www.connectrf.com/Documents/0to1156.pdf>
<http://www.connectrf.com/Documents/1157to1246.pdf>
<http://www.connectrf.com/Documents/1247to1744.pdf>
<http://www.connectrf.com/Documents/1745to2017.pdf>

Socket Errors:

Platform SDK: Windows Sockets

Error Codes in the API

The following is a list of possible error codes returned by the **WSAGetLastError** call, along with their extended explanations. Errors are listed in alphabetical order by error macro. Some error codes defined in Winsock2.h are not returned from any function—these are not included in this topic.

WSAEACCES

(10013)

Permission denied.

An attempt was made to access a socket in a way forbidden by its access permissions. An example is using a broadcast address for **sendto** without broadcast permission being set using **setsockopt(SO_BROADCAST)**.

Another possible reason for the **WSAEACCES** error is that when the **bind** function is called (on Windows NT 4 SP4 or later), another application, service, or kernel mode driver is bound to the same address with exclusive access. Such exclusive access is a new feature of Windows NT 4 SP4 and later, and is implemented by using the **SO_EXCLUSIVEADDRUSE** option.

WSAEADDRINUSE

(10048)

Address already in use.

Typically, only one usage of each socket address (protocol/IP address/port) is permitted. This error occurs if an application attempts to **bind** a socket to an IP address/port that has already been used for an existing socket, or a socket that was not closed properly, or one that is still in the process of closing. For server applications that need to **bind**

multiple sockets to the same port number, consider using **setsockopt(SO_REUSEADDR)**. Client applications usually need not call **bind** at all—**connect** chooses an unused port automatically. When **bind** is called with a wildcard address (involving **ADDR_ANY**), a **WSAEADDRINUSE** error could be delayed until the specific address is committed. This could happen with a call to another function later, including **connect**, **listen**, **WSAConnect**, or **WSAJoinLeaf**.

WSAEADDRNOTAVAIL
(10049)

Cannot assign requested address.

The requested address is not valid in its context. This normally results from an attempt to **bind** to an address that is not valid for the local machine. This can also result from **connect**, **sendto**, **WSAConnect**, **WSAJoinLeaf**, or **WSASendTo** when the remote address or port is not valid for a remote machine (for example, address or port 0).

WSAEAFNOSUPPORT
(10047)

Address family not supported by protocol family.

An address incompatible with the requested protocol was used. All sockets are created with an associated address family (that is, **AF_INET** for Internet Protocols) and a generic protocol type (that is, **SOCK_STREAM**). This error is returned if an incorrect protocol is explicitly requested in the **socket** call, or if an address of the wrong family is used for a socket, for example, in **sendto**.

WSAEALREADY
(10037)

Operation already in progress.

An operation was attempted on a nonblocking socket with an operation already in progress—that is, calling **connect** a second time on a nonblocking socket that is already connecting, or canceling an asynchronous request (**WSAAsyncGetXbyY**) that has already been canceled or completed.

WSAECONNABORTED
(10053)

Software caused connection abort.

An established connection was aborted by the software in your host machine, possibly due to a data transmission time-out or protocol error.

WSAECONNREFUSED
(10061)

Connection refused.

No connection could be made because the target machine actively refused it. This usually results from trying to connect to a service that is inactive on the foreign host—that is, one with no server application running.

WSAECONNRESET
(10054)

Connection reset by peer.

An existing connection was forcibly closed by the remote host. This normally results if the peer application on the remote host is suddenly stopped, the host is rebooted, or the remote host uses a hard close (see **setsockopt** for more information on the **SO_LINGER** option on the remote socket.) This error may also result if a connection was broken due to keep-alive activity detecting a failure while one or more operations are in progress. Operations that were in progress fail with **WSAENETRESET**. Subsequent operations fail with **WSAECONNRESET**.

WSAEDESTADDRREQ
(10039)

Destination address required.

A required address was omitted from an operation on a socket. For example, this error is returned if **sendto** is called with the remote address of **ADDR_ANY**.

WSAEFAULT

(10014)

Bad address.

The system detected an invalid pointer address in attempting to use a pointer argument of a call. This error occurs if an application passes an invalid pointer value, or if the length of the buffer is too small. For instance, if the length of an argument, which is a **SOCKADDR** structure, is smaller than the size of (**SOCKADDR**).

WSAEHOSTDOWN

(10064)

Host is down.

A socket operation failed because the destination host is down. A socket operation encountered a dead host. Networking activity on the local host has not been initiated. These conditions are more likely to be indicated by the error **WSAETIMEDOUT**.

WSAEHOSTUNREACH

(10065)

No route to host.

A socket operation was attempted to an unreachable host. See **WSAENETUNREACH**.

WSAEINPROGRESS

(10036)

Operation now in progress.

A blocking operation is currently executing. Windows Sockets only allows a single blocking operation—per- task or thread—to be outstanding, and if any other function call is made (whether or not it references that or any other socket) the function fails with the **WSAEINPROGRESS** error.

WSAEINTR

(10004)

Interrupted function call.

A blocking operation was interrupted by a call to **WSACancelBlockingCall**.

WSAEINVAL

(10022)

Invalid argument.

Some invalid argument was supplied (for example, specifying an invalid level to the **setsockopt** function). In some instances, it also refers to the current state of the socket—for instance, calling **accept** on a socket that is not listening.

WSAEISCONN

(10056)

Socket is already connected.

A connect request was made on an already-connected socket. Some implementations also return this error if **sendto** is called on a connected **SOCK_DGRAM** socket (for **SOCK_STREAM** sockets, the *to* parameter in **sendto** is ignored) although other implementations treat this as a legal occurrence.

WSAEMFILE

(10024)

Too many open files.

Too many open sockets. Each implementation may have a maximum number of socket handles available, either globally, per process, or per thread.

WSAEMSGSIZE

(10040)

Message too long.

A message sent on a datagram socket was larger than the internal message buffer or some other network limit, or the buffer used to receive a datagram was smaller than the datagram itself.

WSAENETDOWN

(10050)

Network is down.

A socket operation encountered a dead network. This could indicate a serious failure of the network system (that is, the protocol stack that the Windows Sockets DLL runs over), the network interface, or the local network itself.

WSAENETRESET

(10052)

Network dropped connection on reset.

The connection has been broken due to keep-alive activity detecting a failure while the operation was in progress. It can also be returned by **setsockopt** if an attempt is made to set SO_KEEPALIVE on a connection that has already failed.

WSAENETUNREACH

(10051)

Network is unreachable.

A socket operation was attempted to an unreachable network. This usually means the local software knows no route to reach the remote host.

WSAENOBUFFS

(10055)

No buffer space available.

An operation on a socket could not be performed because the system lacked sufficient buffer space or because a queue was full.

WSAENOPROTOPT

(10042)

Bad protocol option.

An unknown, invalid or unsupported option or level was specified in a **getsockopt** or **setsockopt** call.

WSAENOTCONN

(10057)

Socket is not connected.

A request to send or receive data was disallowed because the socket is not connected and (when sending on a datagram socket using **sendto**) no address was supplied. Any other type of operation might also return this error—for example, **setsockopt** setting SO_KEEPALIVE if the connection has been reset.

WSAENOTSOCK

(10038)

Socket operation on nonsocket.

An operation was attempted on something that is not a socket. Either the socket handle parameter did not reference a valid socket, or for **select**, a member of an **fd_set** was not valid.

WSAEOPNOTSUPP

(10045)

Operation not supported.

The attempted operation is not supported for the type of object referenced. Usually this occurs when a socket descriptor to a socket that cannot support this operation is trying to accept a connection on a datagram socket.

WSAEPFNOSUPPORT

(10046)

Protocol family not supported.

The protocol family has not been configured into the system or no implementation for it exists. This message has a slightly different meaning from **WSAEAFNOSUPPORT**. However, it is interchangeable in most cases, and all Windows Sockets functions that return one of these messages also specify **WSAEAFNOSUPPORT**.

WSAEPROCLIM

(10067)

Too many processes.

A Windows Sockets implementation may have a limit on the number of applications that can use it simultaneously. **WSAStartup** may fail with this error if the limit has been reached.

WSAEPROTOSUPPORT

(10043)

Protocol not supported.

The requested protocol has not been configured into the system, or no implementation for it exists. For example, a **socket** call requests a SOCK_DGRAM socket, but specifies a stream protocol.

WSAEPROTOTYPE

(10041)

Protocol wrong type for socket.

A protocol was specified in the **socket** function call that does not support the semantics of the socket type requested. For example, the ARPA Internet UDP protocol cannot be specified with a socket type of SOCK_STREAM.

WSAESHUTDOWN

(10058)

Cannot send after socket shutdown.

A request to send or receive data was disallowed because the socket had already been shut down in that direction with a previous **shutdown** call. By calling **shutdown** a partial close of a socket is requested, which is a signal that sending or receiving, or both have been discontinued.

WSAESOCKTNOSUPPORT

(10044)

Socket type not supported.

The support for the specified socket type does not exist in this address family. For example, the optional type SOCK_RAW might be selected in a **socket** call, and the implementation does not support SOCK_RAW sockets at all.

WSAETIMEDOUT

(10060)

Connection timed out.

A connection attempt failed because the connected party did not properly respond after a period of time, or the established connection failed because the connected host has failed to respond.

WSATYPE_NOT_FOUND

(10109)

Class type not found.

The specified class was not found.

WSAEWOULDBLOCK

()

Resource temporarily unavailable.

This error is returned from operations on nonblocking sockets that cannot be completed immediately, for example **recv** when no data is queued to be read from the socket. It is a nonfatal error, and the operation should be retried later. It is normal for WSAEWOULDBLOCK to be reported as the result from calling **connect** on a nonblocking SOCK_STREAM socket, since some time must elapse for the connection to be established.

WSAHOST_NOT_FOUND

(11001)

Host not found.

No such host is known. The name is not an official host name or alias, or it cannot be found in the database(s) being queried. This error may also be returned for protocol and

service queries, and means that the specified name could not be found in the relevant database.

WSA_INVALID_HANDLE

(OS dependent)

Specified event object handle is invalid.

An application attempts to use an event object, but the specified handle is not valid.

WSA_INVALID_PARAMETER

(OS dependent)

One or more parameters are invalid.

An application used a Windows Sockets function which directly maps to a Win32 function. The Win32 function is indicating a problem with one or more parameters.

WSAINVALIDPROCTABLE

(OS dependent)

Invalid procedure table from service provider.

A service provider returned a bogus procedure table to Ws2_32.dll. (Usually caused by one or more of the function pointers being null.)

WSAINVALIDPROVIDER

(OS dependent)

Invalid service provider version number.

A service provider returned a version number other than 2.0.

WSA_IO_INCOMPLETE

(OS dependent)

Overlapped I/O event object not in signaled state.

The application has tried to determine the status of an overlapped operation which is not yet completed. Applications that use **WSAGetOverlappedResult** (with the *fWait* flag set to FALSE) in a polling mode to determine when an overlapped operation has completed, get this error code until the operation is complete.

WSA_IO_PENDING

(OS dependent)

Overlapped operations will complete later.

The application has initiated an overlapped operation that cannot be completed immediately. A completion indication will be given later when the operation has been completed.

WSA_NOT_ENOUGH_MEMORY

(OS dependent)

Insufficient memory available.

An application used a Windows Sockets function that directly maps to a Win32 function. The Win32 function is indicating a lack of required memory resources.

WSANOTINITIALISED

(10093)

Successful WSASStartup not yet performed.

Either the application has not called **WSASStartup** or **WSASStartup** failed. The application may be accessing a socket that the current active task does not own (that is, trying to share a socket between tasks), or **WSACleanup** has been called too many times.

WSANO_DATA

(11004)

Valid name, no data record of requested type.

The requested name is valid and was found in the database, but it does not have the correct associated data being resolved for. The usual example for this is a host name-to-address translation attempt (using **gethostbyname** or **WSAAsyncGetHostByName**) which uses the DNS (Domain Name Server). An MX record is returned but no A record—indicating the host itself exists, but is not directly reachable.

WSANO_RECOVERY

(11003)

This is a nonrecoverable error.

This indicates some sort of nonrecoverable error occurred during a database lookup. This may be because the database files (for example, BSD-compatible HOSTS, SERVICES, or PROTOCOLS files) could not be found, or a DNS request was returned by the server with a severe error.

WSAPROVIDERFAILEDINIT

(OS dependent)

Unable to initialize a service provider.

Either a service provider's DLL could not be loaded (**LoadLibrary** failed) or the provider's **WSPStartup/NSPStartup** function failed.

WSASYSCALLFAILURE

(OS dependent)

System call failure.

Returned when a system call that should never fail does. For example, if a call to **WaitForMultipleObjects** fails or one of the registry functions fails trying to manipulate the protocol/name space catalogs.

WSASYSNOTREADY

(10091)

Network subsystem is unavailable.

This error is returned by **WSAStartup** if the Windows Sockets implementation cannot function at this time because the underlying system it uses to provide network services is currently unavailable. Users should check:

- That the appropriate Windows Sockets DLL file is in the current path.
- That they are not trying to use more than one Windows Sockets implementation simultaneously. If there is more than one Winsock DLL on your system, be sure the first one in the path is appropriate for the network subsystem currently loaded.
- The Windows Sockets implementation documentation to be sure all necessary components are currently installed and configured correctly.

WSATRY_AGAIN

(11002)

Nonauthoritative host not found.

This is usually a temporary error during host name resolution and means that the local server did not receive a response from an authoritative server. A retry at some time later may be successful.

WSAVERNOTSUPPORTED

(10092)

Winsock.dll version out of range.

The current Windows Sockets implementation does not support the Windows Sockets specification version requested by the application. Check that no old Windows Sockets DLL files are being accessed.

WSAEDISCON

(10101)

Graceful shutdown in progress.

Returned by **WSARecv** and **WSARecvFrom** to indicate that the remote party has initiated a graceful shutdown sequence.

WSA_OPERATION_ABORTED

(OS dependent)

Overlapped operation aborted.

An overlapped operation was canceled due to the closure of the socket, or the execution of the **SIO_FLUSH** command in **WSAIoctl**.

About This Document

This document is based on the following Technical Documents in our Notes Database that have been made obsolete: A1024, A1066, A1076, T1128, T1162, T1167, T1168, T1173, T1175, T1177, T1180, and T1202.

Please let us know about any errors in this document at:

<http://207.241.78.223/isoxpert/calltrak.nsf/WebTracking?OpenForm>.