



## Technical Tip 13

### Winsock Errors and SMS Comfort

A **socket error** often occurs when the local network system aborts a connection, which might happen if the remote host didn't acknowledge data even after retries.

If you get a socket error, first see if you have **antivirus software** or **firewalls** running on both the mail server and the SMS server that may be **blocking outgoing** and/or **incoming tcp/ip** packets. If you do, make sure that the applications are set as a "trusted" application and that the ports are added or turn your scanners off.

If antivirus and firewall software is not the problem, here are a few things to verify:

- Ping the mail server on your local network. Do you get a response?
- Ping the SMS Comfort server. Does it respond?
- Telnet IP-mail server 25. from the SMS Comfort Server. Do you get a connection?
- Telnet IP-SMS Comfort 25. from the Mail Server. Does it connect?

If all these tests are positive there is nothing to keep you from installing SMS Comfort. **If not**, it is probably a **network issue**. Something is still blocking the TCP/IP connection.

1. You can verify with **netstat -a** to see if the necessary ports are open on your mail and SMS server.
2. You can try to make a SMTP connection on the local host: **telnet 127.0.0.1 25**
3. You can try a **traceroute** to see if the problem is at the target remote host or somewhere in between.
4. Install remote port scanning software "**Network Mapper**" on your machine (<http://insecure.org/nmap/>)

To do a standard tcp scan on the reserved ports of a single remote host: (= mail / SMS Server)  
**nmap host**

To scan the same network for all the services in your /etc/services via (very fast) tcp scan:  
**nmap -F host/24**

To do a tcp scan on a specified range of ports of a single host:  
**nmap host -p 1-1024**

With these tests you should be able to find the blocking point in the network.  
(Switch – router – firewall – NIC – software - ....)



The most WinSock Errors that occur with installation of SMS Comfort are:

- 10053 - **Software caused connection abort**
- 10061 - **Connection refused.**

**Default port used within SMS Comfort:**

- configuration GUI : 22280
- Web server m-banxafe : 22281
- SMTP: 25
- DongleServer: 2107

**Note1:** It is possible to change all ports on SMS Comfort to your own desires.

**Note2:** When using the **SMTP Client Tool** delivered with SMS Comfort it is necessary to perform the same test as described above. (Ping and (SMTP)Telnet to both mail and SMS servers)

**Samples:** nmap remote port scan and Telnet 25 (SMTP)

```
C:\WINDOWS\system32\CMD.exe
Microsoft Windows XP [versie 5.1.2600]
(C) Copyright 1985-2001 Microsoft Corp.

P:\>nmap 192.168.1.4

Starting Nmap 4.20 ( http://insecure.org ) at 2007-11-14 23:34 Romance (standaard
dtijd)
Interesting ports on srv4.sonal.local (192.168.1.4):
Not shown: 1688 closed ports
PORT      STATE      SERVICE
25/tcp    open       smtp
80/tcp    open       http
135/tcp   open       msrpc
139/tcp   open       netbios-ssn
443/tcp   open       https
445/tcp   open       microsoft-ds
1720/tcp  filtered  H.323/Q.931
3389/tcp  open       ms-term-serv
8080/tcp  open       http-proxy

Nmap finished: 1 IP address (1 host up) scanned in 11.362 seconds

P:\>
```

```
C:\WINDOWS\system32\CMD.exe
Microsoft Windows XP [versie 5.1.2600]
(C) Copyright 1985-2001 Microsoft Corp.

P:\>telnet 192.168.1.4 25_

Telnet 192.168.1.4
220 ESMTTP Server Ready.
```



## Comprehensive Winsock Error Code List

Return Code	Value	Description
WSAEINTR	10004	<b>Interrupted function call.</b> A blocking operation was interrupted by a call to WSACancelBlockingCall.
WSAEACCES	10013	<b>Permission denied.</b> An attempt was made to access a socket in a way forbidden by its access permissions. An example is using a broadcast address for <b>sendto</b> without broadcast permission being set using <b>setsockopt(SO_BROADCAST)</b> .  Another possible reason for the WSAEACCES error is that when the <b>bind</b> 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 <b>SO_EXCLUSIVEADDRUSE</b> option.
WSAEFAULT	10014	<b>Bad address.</b> 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 <b>sockaddr</b> structure, is smaller than the <b>sizeof(sockaddr)</b> .
WSAEINVAL	10022	<b>Invalid argument.</b> Some invalid argument was supplied (for example, specifying an invalid level to the <b>setsockopt</b> function). In some instances, it also refers to the current state of the socket—for instance, calling <b>accept</b> on a socket that is not listening.
WSAEMFILE	10024	<b>Too many open files.</b> Too many open sockets. Each implementation may have a maximum number of socket handles available, either globally, per process, or per thread.
WSAEWOULDBLOCK	10035	<b>Resource temporarily unavailable.</b> This error is returned from operations on non-blocking sockets that cannot be completed immediately, for example <b>recv</b> 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 WSAEWOLDBLOCK to be reported as the result from calling <b>connect</b> on a non-blocking <b>SOCK_STREAM</b> socket, since some time must elapse for the connection to be established.
WSAEINPROGRESS	10036	<b>Operation now in progress.</b> 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.
WSAEALREADY	10037	<b>Operation already in progress.</b>  An operation was attempted on a non-blocking socket with an operation already in progress—that is, calling <b>connect</b> a second time on a non-blocking socket that is already connecting, or canceling an asynchronous request ( <b>WSAAsyncGetXbyY</b> ) that has already been canceled or completed.
WSAENOTSOCK	10038	<b>Socket operation on nonsocket.</b> An operation was attempted on something that is not a socket. Either the socket handle parameter did not reference a valid socket, or for <b>select</b> , a member of an <b>fd_set</b> was not valid.
WSAEDESTADDRREQ	10039	<b>Destination address required.</b> A required address was omitted from an operation on a socket. For example, this error is returned if <b>sendto</b> is called with the remote address of <b>ADDR_ANY</b> .
WSAEMSGSIZE	10040	<b>Message too long.</b> 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.
WSAEPROTOTYPE	10041	<b>Protocol wrong type for socket.</b> A protocol was specified in the <b>socket</b> 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 <b>SOCK_STREAM</b> .
WSAENOPROTOOPT	10042	<b>Bad protocol option.</b> An unknown, invalid or unsupported option or level was specified in a <b>getsockopt</b> or <b>setsockopt</b> call.
WSAEPROTONOSUPPORT	10043	<b>Protocol not supported.</b> The requested protocol has not been configured into the system, or no implementation for it exists. For example, a <b>socket</b> call requests a <b>SOCK_DGRAM</b> socket, but specifies a stream protocol.
WSAESOCKTNSUPPORT	10044	<b>Socket type not supported.</b> The support for the specified socket type does not exist in this address family. For example, the optional type <b>SOCK_RAW</b> might be selected in a <b>socket</b> call, and the implementation does not support <b>SOCK_RAW</b> sockets at all.



WSAEOPNOTSUPP	10045	<p><b>Operation not supported.</b> 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.</p>
WSAEPFNOSUPPORT	10046	<p><b>Protocol family not supported.</b> 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.</p>
WSAEAFNOSUPPORT	10047	<p><b>Address family not supported by protocol family.</b> 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 <b>socket</b> call, or if an address of the wrong family is used for a socket, for example, in <b>sendto</b>.</p>
WSAEADDRINUSE	10048	<p><b>Address already in use.</b> Typically, only one usage of each socket address (protocol/IP address/port) is permitted. This error occurs if an application attempts to <b>bind</b> 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 <b>bind</b> multiple sockets to the same port number, consider using <b>setsockopt</b> (SO_REUSEADDR). Client applications usually need not call <b>bind</b> at all— <b>connect</b> chooses an unused port automatically. When <b>bind</b> 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 <b>connect</b>, <b>listen</b>, <b>WSAConnect</b>, or <b>WSAJoinLeaf</b>.</p>
WSAEADDRNOTAVAIL	10049	<p><b>Cannot assign requested address.</b> The requested address is not valid in its context. This normally results from an attempt to <b>bind</b> to an address that is not valid for the local computer. This can also result from <b>connect</b>, <b>sendto</b>, <b>WSAConnect</b>, <b>WSAJoinLeaf</b>, or <b>WSASendTo</b> when the remote address or port is not valid for a remote computer (for example, address or port 0).</p>
WSAENETDOWN	10050	<p><b>Network is down.</b> 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.</p>
WSAENETUNREACH	10051	<p><b>Network is unreachable.</b> A socket operation was attempted to an unreachable network. This usually means the local software knows no route to reach the remote host.</p>
WSAENETRESET	10052	<p><b>Network dropped connection on reset.</b> 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 <b>setsockopt</b> if an attempt is made to set SO_KEEPALIVE on a connection that has already failed.</p>
WSAECONNABORTED	10053	<p><b>Software caused connection abort.</b> An established connection was aborted by the software in your host computer, possibly due to a data transmission time-out or protocol error.</p>
WSAECONNRESET	10054	<p><b>Connection reset by peer.</b> 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, the host or remote network interface is disabled, or the remote host uses a hard close (see <b>setsockopt</b> 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.</p> <p>For more information see <a href="#">GlobalSCAPE Knowledge Base Article Q10235</a></p>
WSAENOBUFS	10055	<p><b>No buffer space available.</b> An operation on a socket could not be performed because the system lacked sufficient buffer space or because a queue was full.</p> <p>This error indicates a shortage of resources on your system. It can occur if you're trying to run too many applications (of any kind) simultaneously on your machine. If this tends to occur after running certain applications for a while, it might be a symptom of an application that doesn't return system resources (like memory) properly. It may also indicate you are not closing the applications properly.</p> <p>If it persists, exit Windows or reboot your machine to remedy the problem.</p> <p>Another possible solution is to increase the available virtual memory by increasing the size of the Windows paging file.</p> <p>For more information see <a href="#">GlobalSCAPE Knowledge Base Article Q10234</a></p>
WSAEISCONN	10056	<p><b>Socket is already connected.</b> A connect request was made on an already-connected socket. Some implementations also return this error if <b>sendto</b> is called on a connected SOCK_DGRAM socket (for SOCK_STREAM sockets, the <b>to</b> parameter in <b>sendto</b> is ignored) although other implementations treat this as a legal occurrence.</p>



WSAENOTCONN	10057	<p><b>Socket is not connected.</b> A request to send or receive data was disallowed because the socket is not connected and (when sending on a datagram socket using <b>sendto</b>) no address was supplied. Any other type of operation might also return this error—for example, <b>setsockopt</b> setting SO_KEEPALIVE if the connection has been reset.</p>
WSAESHUTDOWN	10058	<p><b>Cannot send after socket shutdown.</b> A request to send or receive data was disallowed because the socket had already been shut down in that direction with a previous <b>shutdown</b> call. By calling <b>shutdown</b> a partial close of a socket is requested, which is a signal that sending or receiving, or both have been discontinued.</p>
WSAETIMEDOUT	10060	<p><b>Connection timed out.</b> 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.</p>
WSAECONNREFUSED	10061	<p><b>Connection refused.</b> No connection could be made because the target computer 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.</p>
WSAEHOSTDOWN	10064	<p><b>Host is down.</b> 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.</p>
WSAEHOSTUNREACH	10065	<p><b>No route to host.</b> A socket operation was attempted to an unreachable host. See WSAENETUNREACH.</p>
WSAEPROCLIM	10067	<p><b>Too many processes.</b> A Windows Sockets implementation may have a limit on the number of applications that can use it simultaneously. <b>WSAStartup</b> may fail with this error if the limit has been reached.</p>
WSASYSNOTREADY	10091	<p><b>Network subsystem is unavailable.</b> This error is returned by <b>WSAStartup</b> 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:</p> <ul style="list-style-type: none"> <li>• That the appropriate Windows Sockets DLL file is in the current path.</li> <li>• 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.</li> <li>• The Windows Sockets implementation documentation to be sure all necessary components are currently installed and configured correctly.</li> </ul>
WSAVERNOTSUPPORTED	10092	<p><b>Winsock.dll version out of range.</b> 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.</p>
WSANOTINITIALISED	10093	<p><b>Successful WSAStartup not yet performed.</b> Either the application has not called <b>WSAStartup</b> or <b>WSAStartup</b> 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 <b>WSACleanup</b> has been called too many times.</p>
WSAEDISCON	10101	<p><b>Graceful shutdown in progress.</b> Returned by <b>WSARecv</b> and <b>WSARecvFrom</b> to indicate that the remote party has initiated a graceful shutdown sequence.</p>
WSATYPE_NOT_FOUND	10109	<p><b>Class type not found.</b> The specified class was not found.</p>
WSAHOST_NOT_FOUND	11001	<p><b>Host not found.</b> 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.</p>
WSATRY_AGAIN	11002	<p><b>Nonauthoritative host not found.</b> 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.</p>
WSANO_RECOVERY	11003	<p><b>This is a nonrecoverable error.</b> This indicates that some sort of non-recoverable 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.</p>
WSANO_DATA	11004	<p><b>Valid name, no data record of requested type.</b> 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 <b>gethostbyname</b> or <b>WSAAsyncGetHostByName</b>) 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.</p>



**Extra info:**

Winsock errors index - <http://support.ipswitch.com/kb/WSK-19980721-EM01.htm>