SocketWrench/VB
Frequently Asked Questions

This document will provide answers to those questions most frequently asked by those developers who are using SocketWrench/VB or are interested in writing Visual Basic programs against the Windows Sockets API.

For specific information about control properties and events, please refer to the documentation and help files included with SocketWrench/VB.

This FAQ is broken into several sections:

I. General Information
II. Development Questions

I. GENERAL INFORMATION

1. What exactly is SocketWrench/VB?

   SocketWrench/VB is a custom control which uses the Windows Sockets API to provide basic TCP/IP networking functionality to your application. Both client and server capability is provided through the control, as well as support for both TCP and UDP protocols. Virtually any type of application which needs to communicate with another, either on a local network or over the Internet, can be written using SocketWrench/VB.

2. What is the difference between SocketWrench/VB and other commercial or shareware controls that are available?

   SocketWrench/VB is a complete, fully functional control that supports the complete Windows Sockets API. Unlike many other controls that break functionality into several components (such as one control for TCP, another for UDP and yet another for server functionality) SocketWrench/VB provides everything that you need in one control.

   Another significant difference is that SocketWrench/VB is free. There are no licensing or runtime fees, and the control can be used to develop any type of package (commercial, in-house, shareware or freeware). Refer to the licensing agreement included with the package for all of the details.

3. Why was SocketWrench/VB made freely available?

   We decided to make SocketWrench/VB freely available to developers in order to introduce them to our company, Catalyst Software, and contribute something to the community of developers that we will ultimately be marketing to. It also provided us with critical feedback on the features and functionality of the control. This is important because our commercial product, Catalyst SocketTools, shares many of the features found in SocketWrench/VB.

4. If SocketWrench/VB is free, why is there a registration form?

   Even though we are making SocketWrench/VB freely available, we would still like to hear from those developers who are using it. The registration form lets us know who you are, what platforms you're using SocketWrench/VB on, and what you think about the product. Also, as an added bonus, registered SocketWrench/VB users will receive an extra $50 discount on Catalyst SocketTools. This discount can be applied to the special introductory price. It also gives us an opportunity to let you know when other Catalyst products are available that you might be interested in.

5. What is SocketTools?

   Catalyst SocketTools is a collection of TCP/IP controls written for developers who need higher-level access to specific application protocols. Included in the package are controls for file transfer (FTP), remote login and terminal emulation (TELNET), sending and receiving electronic mail (SMTP/POP3), accessing USENET newsgroups (NNTP) and the World Wide Web (HTTP/HTML).

6. What platforms can SocketWrench/VB be used on?

   SocketWrench/VB will work with any TCP/IP stack that provides a Windows Sockets 1.1 compliant library. This includes the TCP/IP stacks that are part of the base operating system for Windows 95 and Windows NT. A free 32-bit TCP/IP stack can be obtained from Microsoft for Windows for Workgroups 3.11.

   SocketWrench/VB can be used with Visual Basic 2.0 or later, including the 16-bit version of Visual Basic 4.0. Since VBXs are 16-bit controls, it cannot be used with the 32-bit version of VB 4.0.

7. Will there be a 32-bit version of SocketWrench/VB?

   Absolutely. A 32-bit OCX (OLE custom control) is included in SocketTools and will be available shortly. For further information about the availability of the OCX, send e-mail to info@catsoft.com or contact us by FAX or telephone. The README file includes this contact information.

8. What is the current version of SocketWrench/VB and where can I download the product?

   The current version of the control is 1.1, and is available at a number of sites. On CompuServe it is available in the MSBASIC forum as CSWSK110.ZIP. On the Internet, it is available on your local SimTel mirror site in the /win3/visbasic directory as cswsk110.zip. You can always find the latest copy of the package on the Catalyst Software Web site at http://www.earthlink.net/~catalyst/.

9. I want to use SocketWrench/VB in my application, but my employer feels uncomfortable with me using free software.

   Keep in mind that SocketWrench/VB is part of a commercial package, and will be fully supported by Catalyst Software. We would also consider selling a source license to qualified organizations on a case-by-case basis. Contact our business office for further information about source licensing.

10. If I have a problem or question, what is the best way to contact you?

    The best way to contact Catalyst Software is via e-mail. If you have a product-related question, send it to info@catsoft.com. If you have a technical support question or problem, you can address it to support@catsoft.com.

    If you have a technical support problem, please remember to include information about what TCP/IP stack you're using along with a detailed description of the problem. If possible, a sample project that reproduces the problem would be very helpful.

II. DEVELOPMENT QUESTIONS

1. I'm not very familiar with TCP/IP and sockets programming. How do you recommend I get started?

   Included with SocketWrench/VB is a tutorial in Microsoft Word 6 format. It's in the file TCPINTRO.DOC, and convers the basics of TCP/IP programming with SocketWrench/VB. There are also a number of good books available on the subject, with a few of them specific to Windows Sockets programming in Visual Basic.

2. I'm familiar with sockets programming under UNIX, and need to write some client applications for Windows. Will your product work for me?

   Absolutely. If you're familiar with sockets programming under UNIX, you should find the control particularly easy to work with in creating your client applications with Visual Basic.

3. I want to use SocketWrench/VB to develop an program which uses the XYZ application protocol. How do I do that?

   The best resource for learning about the common application protocols are RFC's. These Request For Comment documents form the foundation of the application protocols used over the Internet. There are a great number of them, but some of the more common ones are:

RFC 821	Simple Mail Transfer Protocol (SMTP)
RFC 959	File Transfer Protocol (FTP)
RFC 977	Network News Transfer Protocol (NNTP)
RFC 1341	Multipurpose Internet Mail Extensions (MIME)
RFC 1436	Internet Gopher Protocol
RFC 1459	Internet Relay Chat Protocol (IRC)
RFC 1725	Post Office Protocol Version 3 (POP3)

4. When I try to include SocketWrench/VB in my project, I get an error message stating that it cannot load the control. What's wrong?

   Check to make sure that a Windows Sockets library is installed on the system, and that it's either in the \WINDOWS directory, or a directory that's specified in the PATH environment variable. The control will fail to load if this library, WINSOCK.DLL, cannot be loaded. Also check to make sure that there is enough free memory available to load both the control and the Windows Sockets library.

5. When I place the control on a form, I get a general protection fault (GPF) right away or shortly afterwards. Why?

   Check to make sure that there is only one copy of a Windows Sockets library installed on your system, and that the correct library is being loaded by the control. You can use the WPS utility (included with the Visual Basic 3.0 Professional edition or downloadable from Microsoft) to determine where the WINSOCK.DLL is being loaded from.

6. Why do I get errors that some properties in the control could not be set when I try to run a program?

   This problem manifested itself in early versions of the control when it could not determine the local host name. Upgrading to the current version of SocketWrench/VB will resolve the problem.

7. When I place the control on a form, the toolbox icon isn't drawn correctly. After that, if I try to place another control on the form, Visual Basic will GPF. What's going on?

   This is a problem with older versions of the control that would attempt to determine the local host name and IP address as soon as it was loaded on a form. This caused a problem for those developers who only had a dialup SLIP or PPP connection and either had a dynamically configurable IP address or relied on DNS and did not have any entries in their local hosts file for their own system. The current version of the control resolves this problem.

8. I have a dialup connection to the Internet, but don't want to have to be connected while I develop my software. Is there any way that I can do this with your control?

   Included with SocketWrench/VB is a "loopback" Windows Sockets library which can be installed on your system. This library contains all of the functions of a standard Windows Sockets library, but can only be used to exchange data on the local system. Unlike some stub libraries which return errors for every function, the loopback library will allow you to develop and test working software.

9. I installed SocketWrench/VB, but I don't see the loopback library anywhere. How do I install it?

   The loopback library is only installed by default if you have no Windows Sockets library already present on your system. You can install the library by extracting the WINSOCK.DL_ and WINSOCK.INI files from the ZIP archive manually. Note that the library is compressed, so you will need to run Microsoft's EXPAND utility on it.

10. I have a dialup connection to the Internet and I'm using Windows 95. I've noticed that some applications will automatically bring up the dialer, but SocketWrench/VB doesn't. Why is that?

    Because SocketWrench/VB is a 16-bit VBX, it uses the 16-bit Windows Sockets library, WINSOCK.DLL. For reasons known only to Microsoft, they decided that only the 32-bit socket library would be capable of automagically invoking the dialer.

11. Why can I use SocketWrench/VB in my Visual Basic programs without any problems, but when I try to use it with Visual C++, I get an error?

    SocketWrench/VB was developed against the Visual Basic 2.0 control specification. Unfortunately, Visual C++ only supports controls written to the 1.0 specification.

12. When I try to use SocketWrench/VB with Borland's Delphi, the control will load, but I'm not able to see any of it's properties. And when I try to use it, error messages are displayed. Why?

    For the same reason that the control will not work with Visual C++, namely because Delphi only supports VBXs written to the 1.0 specification.

13. I have Sheridan's VB Assist loaded, and whenever I press the right mouse button to look at the control's properties, I get a GPF. I can see all the properties in the property box and everything is fine if I don't have VB Assist loaded.

    SocketWrench/VB implements some custom properties which have the same names as standard control properties. This is fine for Visual Basic, but seems to confuse VB Assist. Sheridan is aware of the problem, so it may have been corrected in the latest latest version of their software.

14. Why do I only get empty strings when I read the LocalName or LocalAddress properties?

    SocketWrench/VB attempts to determine your local host name and address by first calling a function to determine your local name(through a call to the gethostname() function, and then using that name, your IP address through a call to the gethostbyname() function.

    Some Windows Sockets libraries, notably the one included in Novell's Lan Workplace product, will not return the required information unless the system has an entry for itself in the local hosts file. This problem may also occur if you have a dialup network connection, using DNS for name resolution, and your host does not have a reverse (PTR) entry in the DNS database.

15. When I try to send UDP datagrams using SocketWrench/VB, they are never received by the remote host. What's wrong?

    Earlier versions of the control would not report any errors if it was unable to send or receive a UDP datagram. This problem has been corrected, and it is recommended that you upgrade to the current release.

16. When I try to use the TINYFTP sample included with SocketWrench/VB, I get errors from the server saying that it was unable to build a data connection. What does this mean?

    There was a bug in the FTPListen() function in the sample program. The problem was the creation of the PORT command which tells the server what your IP address is and what port you're socket is listening on. The code where the CtlData string is being created should be changed to the following:

    ```
    '
    ' Split the local port number into high and low bytes by
    ' converting it to hex, pulling it apart, and then converting
    ' the pieces back to decimal
    '
    HexPort = Hex$(Port)
    If Len(HexPort) = 3 Then HexPort = "0" + HexPort
    CtlData = "PORT " & Address & "," &
           (Val("&h" + Left$(HexPort, 2))) & "," & (Port And &HFF)
    ```

    The last two lines should be joined, and you will need to declare the HexPort variable as a string in the function.

    The problem fundamentally exists because port numbers are unsigned 16-bit integers, but Visual Basic only uses signed integers. The conversion to hexadecimal and back works around this limitation.

17. I am using non-blocking sockets in my application and responding to Read events in the control. Why do I sometimes get an error when I read the RecvData property that says that the socket operation would block?

    This is a problem with earlier versions of the control and has been corrected in the current release. The DataLength argument passed to the function would sometimes be incorrect, and by setting the RecvLen property to that value, would cause the control to attempt to read more data than is available on the socket, which would erroneously generate the blocking error message.

18. Why do I get strange values when I check the RecvNext property to determine the number of bytes available to be read from the socket?

    This is a problem that has been found on some Windows Sockets implementations where a call to the ioctlsocket() function returns a wildly incorrect value. This problem typically manifests itself on non-blocking, unbuffered TCP sockets.

19. I am trying to write a client application, and after the first connection, I'm getting an error stating that the address is already in use. What am I doing wrong?

    You're not setting the LocalPort property to IPPORT_ANY before you establish the connection. By default, the LocalPort property has a value of zero (which is the same as IPPORT_ANY). What this tells the socket library is that you don't care what your local port number is, and for the library to assign you an unused port number greater than 1024.

    After you connect to the server, the LocalPort property is modified to reflect the actual port number that was assigned to your socket. If, the next time you attempt to connect using that socket, the LocalPort property does not have a value of zero, it will attempt to use that specific port number, which may result in the error being generated. To correct the problem, simply insure that the LocalPort property has a value of zero before you attempt the connection.

20. My application needs to broadcast UDP datagrams over the local network. How can I do this using SocketWrench/VB?

Refer to the UDPECHO sample program for a generic example of using UDP with SocketWrench/VB. In order to broadcast datagrams (that is, each machine on the local network will receive the datagram sent from your local host), you need to set the Broadcast property to True, and set the HostAddress property to the broadcast IP address for your network.

21. I'm developing a server application using SocketWrench/VB under Windows NT. Everything seems to work well, until I attempt to connect to it with a client running on the same local host. Why does it seems to slow down considerably at that point?

    There is a significant performance penalty when 16-bit and 32-bit TCP/IP applications connect to each other on the same local system. No doubt this is due to the WOW (Windows on Windows) facility used by Windows NT to execute 16-bit applications.