VBStak®

TCP/IP Custom Control

Description

The VBStak custom control provides a TCP/IP interface for Visual Basic and Visual C++ programs, opening Unix and Internet connectivity to Windows applications. VBStak handles the low level access to the Winsock dynamic link library, presenting a set of properties to the program.

The VBStak icon looks like the following when added to the Visual Basic tool box:

File Name

VBStak.vbx

Object Type

VBStak

Remarks

Constants

Data constants for VBStak are defined in VBSTAK.TXT.

Distribution Notes: When you create and distribute applications that use VBStak you should install VBSTAK.VBX in windows\system directory.

Properties

Action Property

Description

Initiate a socket action. Write only at run time.

Visual Basic

[form.]*VbStakControl* . **Action** *[= ActionConstant%]*

Remarks

Initiate the asynchronous action/request contained in *ActionValue* Upon completion of the action or request VBStak fires the Message event with the message parameter set to indicate source of the event. If an error occurred in the processing of the action/request the Error property will be set to the last error that occurred.

Action values are :

STAK_ACTION_CONNECT - Connect to the host on the service using the family and protocols. The Family, Protocol, HostAddress, Type and RemotePort properties must be set prior to this action. Upon completion of the request VBStak fires the Message event with FD_CONNECT as the parameter.

STAK_ACTION_CLOSE - Close the socket. If the VBStak is in STAK_STATE_ACCEPTED state, the secondary socket is closed and the master socket is restored to a listening state (STAK_STATE_LISTENING). To close the master socket issue the STAK_ACTION_CLOSE again.

STAK_ACTION_GET_SERVICE - Lookup the service name using the text in the Serviceproperty. Upon return VBStak fires the Messageevent with STAK_EVENT_SERVICE as the parameter. If the service is found the RemotePort property is set to the port number of the service.

STAK_ACTION_GET_HOST - Lookup the IP address of the host entered in the Host property. Upon return VBstak fires the Message event with STAK_EVENT_HOST as a parameter. If the host has been found the HostAddress property is set to the dotted IP address of the host.

STAK_ACTION_GET_PROTOCOL Find the protocol number for the text in the ProtocolName property. Upon completion of the request VBStak fires the Message event with STAK_EVENT_PROTOCOL as the parameter. If the search was successful set the Protocol property to the integer value of the protocol.

STAK_ACTION_LISTEN Prepare the socket as a server. Prior to this action the LocalPort should be set to a value above IPPORT_RESERVED. Upon receiving a request for a connection the Message event STAK_EVENT_ACCEPT will occur. At this point two sockets will be open: the master socket (MasterSocket) and the secondary socket (Socket). The secondary socket will be available for data transactions and the master socket will be idle. Issuing a STAK_ACTION_CLOSE closes the secondary socket and reactivates the master socket.

STAK_ACTION SEND Send the datagram to the socket. Upon completion of the request

(assuming a response is pending) VBStak fires the the Message event with FD_READ as the parameter. The Outputproperty should be set prior to setting this action.

Note: The constants are contained in VBStak.txt file.

Data Type

Integer

Concepts

Introduction

VBStak is a Visual Basic Control (DLL) which provides a programming interface to the Winsock services through properties and events. VBStak is accessed through VBSTAK.VBX in conjunction with the constants file VBSTAK.TXT which are added to your project file.

Setup

The recommended platform for VBStak development is the Windows for Workgroups (WFW) version 3.11 with the Microsoft WFWT32 driver set. This package is distributed free and can be downloaded from Compuserve and other BBS's. A handy Telnet program included in the package allows you to connect to the host and monitor the connections and activity on the host at the same time as you debug your VBStak application.

Install the WFWT32 driver set into its own directory then install the driver using the WFW network setup utility. Make sure that your local IP address and local host name (your workstation) are assigned properly.

Create/edit the \windows\hosts file to include your host. The format for each host entry is:
hostipAddress *hostName*

Asynchronous Interface

VBStak utilizes asynchronous processing provided by Winsock to build a well behaved Windows program. The programming technique required to take advantage of this is presented in Asynchronous Interface topic. Detailed examples are provided in the VBMail and StakMan program samples.

Basic Steps (Client)

TCP\IP client connectivity involves the four basic steps.

1. Configure local communication port (can be set at design time)
   - set the Family,Type properties
   - set the Protocol property.
   - set the LocalPort property

2. Connect to remote host.
   - set the HostAddress of the remote host.
   - set the RemotePort property.
   - connect .

3. Transmit and receive data.
   - load the Output property
   - send the data using the Action property with STAK_ACTION_SEND.
   - respond to the receipt of data in the Message event.
   - receive the data using the Input property.

4. Close the connection.
   - close the socket using the Action property and STAK_ACTION_CLOSE.

Basic Steps (Server)

TCP\IP server connectivity involves the four basic steps.

1. Configure local communication port (can be set at design time)
   - set the Family,Type properties
   - set the Protocol property.
   - set the LocalPort property.

2. Listen for connection requests.
   - set the Actionproperty to STAK_ACTION_LISTEN.
   - wait for the Message event with FD_ACCEPT to indicate an accepted connection.

3. Transmit and receive data.
   - load the Output property
   - send the data using the Action property with STAK_ACTION_SEND.
   - respond to the receipt of data in the Message event.
   - receive the data using the Input property.

4. Close the connection.
   - close the socket using the Action property and STAK_ACTION_CLOSE. This restores the VBStak to the STAK_STATUS_LISTENING state.

Database functions

VBStak provides two interface methods for Winsock services, the direct access and the indirect database lookup method.

1. The direct access method requires that you specify the host address, protocol and host port using the numeric values such as the dotted host address format - 192.23.3.1. This requires that you know, prior to running the program, the absolute value of these properties.

2. The second (and preferred) method uses the Unix database facilities to provide a translation between text based names and the socket parameters. The host name is entered in the familiar *user@company.com* format. The address of the host is resolved by the network name resolution functions. Protocol names and services are entered as their 'well know names' (eg. tcp and telnet). This interface allows greater flexibility in addressing and user defined services.
Three files installed in your \windows directory define the databases provided. Alternately if your network has Domain Name System installed the files exist on the name server in the /etc/ directory. The file are:

hosts	a list of the hosts on the network, including your local workstation.
services	a list of the network services, port numbers and protocols.
protocol	a list of the available protocols

Copyright and Shareware Licence

Thank you for considering VBStak for your TCP/IP development project. Our objective is to provide communication solutions that are accessable and reasonably priced. We do this by concentrating our resources on development and customer support, not advertising and marketing. If your project can benefit from this control please jump to Registration to view the benefits of registering this product.

Distribution of the Shareware version is permitted under the following terms:

DEFINITION OF SHAREWARE

Shareware distribution gives users a chance to try software before buying it. If you try a Shareware program and continue using it, you are expected to register. Individual programs differ on details -- some request registrationwhile others require it, some specify a maximum trial period. With registration, you get anything from the simple right to continue using the software to an updated program with printed manual.

Copyright laws apply to both Shareware and commercial software, and the copyright holder retains all rights, with a few specific exceptions as stated below. Shareware authors are accomplished programmers, just like commercial authors, and the programs are of comparable quality. (In both cases, there are good programs and bad ones!) The main difference is in the method of distribution. The author specifically grants the right to copy and distribute the software, either to all and sundry or to a specific group. For example, some authors require written permission before a commercial disk vendor may copy their Shareware.

Shareware is a distribution method, not a type of software. You should find software that suits your needs and pocketbook, whether it's commercial or Shareware. The Shareware system makes fitting your needs easier, because you can try before you buy. And because the overhead is low, prices are low also. Shareware has the ultimate money-back guarantee -- if you don't use the product, you don't pay for it.

DISCLAIMER - AGREEMENT

Users of VBStak must accept this disclaimer of warranty:

"VBStakis supplied as is. The author disclaims all warranties, expressed or implied, including, without limitation, the warranties of merchantability and of fitness for any purpose. The author assumes no liability for damages, direct or consequential, which may result from the use of VBStak ."

VBStak is a "shareware program" and is provided at no charge to the user for evaluation. Feel free to share it with your friends, but please do not give it away altered or as part of another system. The essence of "user-supported" software is to provide personal computer users with quality software without high prices, and yet to provide incentive for programmers to continue to develop new products. If you find this program useful and find that you are using VBStak and continue to use VBStak after a reasonable trial period, you must make a

registration payment of $49.00 US to Third Stone. The $49.00 US registration fee will license one copy for use on any one computer at any one time. You must treat this software just like a book. An example is that this software may be used by any number of people and may be freely moved from one computer location to another, so long as there is no possibility of it being used at one location while it's being used at another. Just as a book cannot be read by two different persons at the
same time.

Commercial users of VBStak must register and pay for their copies of VBStak within 30 days of first use or their license is withdrawn. Site-License arrangements may be made by contacting Third Stone.

Anyone distributing VBStak for any kind of remuneration must first contact Third Stone at the address below for authorization.This authorization will be automatically granted to distributors recognized by the (ASP) as adhering to its guidelines for shareware distributors, and such distributors may begin offering VBStak immediately (However Third Stone must still be advised so that the distributor can be kept up-to-date with the latest version of VBStak .).

You are encouraged to pass a copy of VBStak along to your friends for evaluation. Please encourage them to register their copy if they find that they can use it. All registered users will receive a copy of the latest version of the VBStak system.

Registration

What do I get for registering?

1. Royalty free licence with distribution VBStak.vbx.

2. Free updates for 6 months.

3. 1 Year of TIP's Electronic newsletter with TCP/IP connectivity examples, programming tips and configuration details.

How do I register?

Credit Card Orders:
(NorthStar Solutions credit card order processing - not tech support)

Phone: 1-800-699-6395 (10:00 a.m. to 10:00 p.m., Eastern Standard Time)
Fax 1-803-699-5465 (24 hours)
E-mail 71561.2751@compuserve.com

or

Compuserve registration: GO SWREG use Registration #: **5222**

or

Check/Money orders:

Fill in VBStak.reg and send $49.00 to :

Third Stone Engineering, Inc.
5215 Sepulveda Blvd. Suite 6A
Culver City, CA 90230

Regardless of how you register, please have the following information ready:

1.The program and version number ([NAME] version [x.x]) you are registering.
2. Where the latest version should be mailed.
3. Your Visa or MasterCard # and its expiration date (if using Visa/MC).
4. Your drive types, 5.25 inch or 3.5 inch.

NOTES: 1) NorthStar processes registrations only, please contact Third Stone for any product/technical support.
2) E-mailed or faxed registrations are encouraged due to their low cost & high efficiency, but any registration is appreciated!

VBStak Registration

Product: VBSTAK.VBX Version 1.2
Name:
Company:
Address:
City: State/Prov:

Country:
Phone:
EMail:
Number of Licences: x $US 49:

Host Property

Description
Sets and returns the name of the remote computer. Read/write.

Visual Basic
[form.]*VbStakControl .* **Host** *[= hostName$]*

Remarks

The host property is the name of the host computer in the user@entity.type form. This is used with the STAK_ACTION_GET_HOST action parameter to look up the host address. The Winsock dll uses the **hosts** file in the windows directory to search for a host/address association in the absence of a Domain Name System (DNS) server.

To locate a host:

1. Set the Host property to the name of the host eg. "haddock@tse.com"

2. Set the Action property to STAK_ACTION_GET_HOST.

Response:

1. VBStak sets the HostAddress property with the IP address (nnn.nnn.nnn.nnnn format) of the host if located. The Error property is set if an error occurred.

2. VBStak fires the Message event with STAK_EVENT_HOST.

Data Type
String

Example:

Define a form with a button called CheckHost and a VBStak control called VBStak.

```
Sub CheckHost_Click()

    VBStak.Service = "haddock@tse.com"
    VBStak.Action = STAK_ACTION_GET_HOST

End Sub

Sub VBStak_Message(message as Integer)

    ' Check the return response
    Select Case message
    Case STAK_EVENT_HOST
    ' ....
        If VBStak.Error = 0 then
            HostAddressBox.Text =  VBStak.HostAddress
```

```
            Else
                    MsgBox("Host " & VBStak.Host & " not found - Error: " & Str(VBStak.Error))
            End If
        ' ....
    End Select
End Sub
```

Input Property

Description

Receive the number of bytes set by the InputLen property. Read only at run time.

Visual Basic

[form.]*VbStakControl .* **Input**

Remarks

If there are more bytes in the input buffer than defined in InputLen VBStak will retrieve the InputLen number of bytes and then post another FD_READ message event indicating that there is more data available.

If there are less then InputLen bytes available VBStak will retrieve an empty string and set the Error property to STAK_ERROR_DATA_NOT_AVAILABLE.

Note: If you use InputLen to set the input size, it is important to check the InBufferCount value prior to accessing the Input property. Winsock only sends the FD_READ message when data is first available or after an Input access with an InputLen which is less than the InBufferCount. This means that if you attempt to input more than the buffer count you will not get subsequent FD_READ messages. To prevent this deadlock you must clear the buffer with the Input property.

Prior to receiving data the socket must be connected to a host . This is accomplished by setting the Action property to STAK_ACTION_CONNECT. The connect action enables data reception on the socket and directs the Winsock messages to the control event Message.

Data Type

String

Error Property

Description

Sets and returns the error result of the last socket operation. Read only at run time.

Visual Basic

[form.]*VbStakControl .* **Error**

Remarks

VBStak errors are defined below and in VBStak.txt. The text string associated with an error are defined in VBStak.txt as the error name with a "_S" suffix eg. WSANOTINITIALISED_S. These can be loaded into an array at startup time.

WSANOTINITIALISED = "A successful WSAStartup() must occur before using this API."

WSAENETDOWN = "The Windows Sockets implementation has detected that the network subsystem has failed."

WSAEADDRINUSE = "The specified address is already in use. (See the SO_REUSEADDR socket option under setsockopt().)"

WSAEFAULT = "The namelen argument is too small (less than the size of a struct sockaddr)."

WSAEINTR = "The (blocking) call was canceled via WSACancelBlockingCall()"

WSAEINPROGRESS = "A blocking Windows Sockets call is in progress."

WSAEAFNOSUPPORT = "The specified address family is not supported by this protocol."

WSAEINVAL = "The socket is already bound to an address."
WSAENOBUFS = " Not enough buffers available, too many connections."
WSAENOTSOCK = "The descriptor is not a socket."
WSAEADDRNOTAVAIL = "The specified address is not available from the local machine."
WSAECONNREFUSED = "The attempt to connect was forcefully rejected."
WSAEDESTADDREQ = "A destination address is required."
WSAEISCONN = "The socket is already connected."
WSAEMFILE = "No more file descriptors are available."
WSAENETUNREACH = "The network can't be reached from this host at this time."
WSAETIMEDOUT = "Attempt to connect timed out without establishing a connection"
WSAEWOULDBLOCK = "The socket is marked as non-blocking and the connection cannot be completed immediately. It is possible to select() the socket while it is connecting by select()ing it for writing."
WSAHOST_NOT_FOUND = "Authoritative Answer Host not found."
WSATRY_AGAIN = "Non-Authoritative Host not found, or SERVERFAIL."
WSANO_RECOVERY = "Non recoverable errors, FORMERR, REFUSED, NOTIMP."
WSANO_DATA = "Valid name, no data record of requested type."

WSAENOPROTOOPT = "The option is unknown or unsupported. In particular."
WSAEACCES = "The requested address is a broadcast address, but the appropriate flag was not set."
WSAENETRESET = "The connection must be reset because the Windows Sockets implementation dropped it."
WSAENOTCONN = "The socket is not connected."
WSAEOPNOTSUPP = "MSG_OOB was specified, but the socket is not of type SOCKTREAM."
WSAESHUTDOWN = "The socket has been shutdown; it is not possible to send() on a socket after shutdown() has been invoked with how set to 1 or 2."
WSAEMSGSIZE = "The socket is of type SOCK_DGRAM, and the datagram is larger than the maximum supported by the Windows Sockets implementation."
WSAECONNABORTED = "The virtual circuit was aborted due to timeout or other failure."
WSAECONNRESET = "The virtual circuit was reset by the remote side."

Data Type

Integer

Service Property

Description

Sets and returns the name of the service to use on the remote host. Read and write design and run time.

Visual Basic

[form.]*VbStakControl*.Service *[= serviceName$]*

Remarks

The service property sets the service to connect to on the host. Use the Service property to verify that the service exists on the host and to define the RemotePort property. A list of the services is maintained in the **/etc/services** file on the host and in the **services** file in the local Windows directory.

To define a custom service, add the definition to the **services** file in your Windows directory. Each service is defined on one lin in the form:

serviceName *portNumber/protocolName* *comment*

eg.

echo 7/tcp return all data

To locate the host port number:

1. Set the Service property to the service name.
2. Set the ProtocolName property to the desired protocol (tcp/udp)
3. Set the Action property to STAK_ACTION_GET_SERVICE.

Upon completion of the request VBStak sets the RemotePort property to the port number and fires the Message event with STAK_EVENT_SERVICE.

Data Type

String

See Also: Host property, Action property

Example:

Define a form with a button called CheckService and a VBStak control called VBStak.

```
Sub CheckService_Click()

    VBStak.Service = "echo"
    VBStak.ProtocolName = "tcp"
    VBStak.Action = STAK_ACTION_GET_SERVICE
```

```
 End Sub

Sub VBStak_Message(message as Integer)

      ' Check the return response
      Select Case message
      Case STAK_EVENT_SERVICE
      ' ....
           If VBStak.Error = 0 then
                  RemotePortBox.Text =    VBStak.RemotePort
           Else
                    MsgBox("Service Request Failed")
          End If
      ' ....
     End Select
End Sub
```

RemotePort Property

Description

Sets and returns the numeric value of the port on the remote host . Read/Write.

Visual Basic

[form.]*VbStakControl .* **RemotePort** *[= portNumber%]*

Remarks

The RemotePort property can be set directly to define the port number of the host service . The preferred method of defining the RemotePort value is to set the Service property to the symbolic name for the service then set the Action property to STAK_ACTION_GET_SERVICE.

Data Type

Integer

HostAddress Property

Description

Sets and returns the IP address of the remote computer.

Visual Basic

[form.]*VbStakControl .* **HostAddress** *[= ipAddress$]*

Remarks

The HostAddress property is the string representation of the 32 bit address assigned to the host . This is used by VBStak to reach the remote computer.

The HostAddress property can be set directly to quickly test a host response or if the host address will not change in the furture. However the recommended method of defining the HostAddress property is to set the Host property to the symbolic address of the host then setting the Action property to STAK_ACTION_GET_HOST.

Data Type

String

Family Property

Description

Sets and returns the family of protocols to use on the socket connection.

Visual Basic

[form.]*VbStakControl .* **Family** *[= familyConstant%]*

Remarks

The family values are defined in VBSTAK.TXT. The usual value for Family property is set AF_INET for remote host to host communication.

Data Type

Integer

Protocol Property

Description

Sets and returns the protocol of the local socket. Read/Write.

Visual Basic

[form.]*VbStakControl .* Protocol *[= protocolValue%]*

Remarks

The Protocol value may be set directly or may be defined by setting the ProtocolNameproperty then setting the Action property to STAK_ACTION_GET_PROTOCOL . This will tell VBStak to look up the protocol value.

Protocol values are stored in the windows\system\protocols file distributed with Winsock.dll.

Data Type

Integer

Type Property

Description

Sets and returns the type of the local socket. Read/Write.

Visual Basic

[form.]*VbStakControl* . **Type** *[= ipAddress$]*

Remarks

The type values are:

SOCK_STREAM - Stream socket (use with TCP protocol)
SOCK_DGRAM - Datagram socket (use with UDP protocol)
SOCK_RAW - Raw protocol interface
SOCK_RDM - Reliably delivered message
SOCK_SEQPACKET - Sequenced packet stream

Common values for type are SOCK_STREAM or SOCK_DGRAM .
Type values may be set at design time using the drop down list provided with VBStak.

Data Type

Integer

Output Property

Description

Load the output buffer with a string. Run time write only.

Visual Basic

[form.]*VbStakControl .* **Output** *[= sendString$]*

Remarks

Setting this property makes the strng available for transmission. To actually send the data the Action property must be set to STAK_ACTION_SEN.

The string may contain byte values between 0 and 255.

Data Type

String

Message Service Example

This example is a template for the asynchronous serviceing of the Winsock requests initiated by setting the Action property. To run this example setup a form with a VBStak control named VBStak. Add controls:

StatusBox - (TextBox) The status of the socket connection. Make this a multi-line control.
InputBox - (TextBox) - User input.
ActionButton (Button) - Start the action

```
Sub VBStak_Message (message As Integer)
 '   Service the message event
If VBStak.Error = 0 then
        Select Case message
        Case FD_CONNECT
              ' The socket has connected to the host
              StatusBox.Text = "Connected to " & VBStak.HostAddress & "/"

        Case FD_READ
               ' Copy to the output text box.
              InputBuffer.Text = InputBuffer.Text & VBStak.Input & Chr(13) & Chr(10)

                Case FD_CLOSE
              VBStak.Action = STAK_ACTION_CLOSE
                          StatusBox.text = "Closed"

        Case STAK_EVENT_SERVICE
              ' The remote port has been loaded
              StatusBox.Text = "Host Port: " & VBStak.RemotePort

        Case STAK_EVENT_PROTOCOL
              ' The protocol has been loaded
              StatusBox.Text = "Protocol Number: " & VBStak.Protocol

        Case STAK_EVENT_HOST
              ' The host address has been loaded
              StatusBox.Text = "Remote Host Address: " & VBStak.HostAddress
                          ConnectButton.Enabled = true
        End Select
    Else
        StatusBox.Text = "Error: " & VBStak.Error
    End If
End Sub
```

Asynchronous Interface

VBStak utilizes the asynchronous request facilities of the Winsock dll. This approach provides a well behaved Windows interface because other processes are able to execute while waiting for a request to complete. This fits Visual basic's event driven architecture very well as long as we utilize it properly.

The effective utilization of VBStak involves separating the action initiation from the request result. In a single user / process environment the following code is permissible:

```
Sub WeHaveAllDay()
      Call InitiateAction()
      While not Responded()   and not TimedOut()
            ' Wait here
      Loop
      If TimedOut() then
         Call ProcessTimeOutError()
      End If
 End Sub
```

In a multi-processing environment (especially Windows without any pre-emption) this code would prevent any other program from getting access to the processor or at least waste a lot of processor time. Event driven programs separate the initiation / result portions of the program as follows:

```
' Initiating procedure
Sub StartThingsOff()
       RequestTimer.Interval = THE_MAXIMUM_TOLERABLE_WAIT
       RequestTimer.Enabled = True
       VBStak.Output = "Echo this string" & Chr(0)
       VBStak.Action = STAK_ACTION_SEND
End Sub

' Receiving procedure - VBStak messages are processed here.
Sub VBStak_message(message as integer)
       Select Case   message
              '...
       Case FD_READ
              ' Service the request
              Text1.text = VBStak.Input
              ' Initiate the next transmission
              '...
       End Select
End Sub

'To prevent an indefinite wait a timer is used:
Sub RequestTimer_Time
       ' Alert user and/or retry
       MsgBox("Request Timed Out")
End Sub
```

This trilog of procedures seems more complicated than the direct send/wait loop. However in fact the programming steps are the same, they have just been separated into independent, asynchronous modules which allows other processes to run while your program waits for input.

The control of sequential operations is implemented by building a state machine in the Message event. As each step in the sequence is completed (the appropriate resonse from the host) the state of the machine is set to the next step in the sequence. If an error occurs the state machine is reset to an idle state. A working example of this technique is contained in the VBMail sample program.

LocalPort Property

Description

Sets and returns the port number to use on the local socket. Read/write.

Visual Basic

[form.]*VbStakControl .* **LocalPort** *[= portNumber%]*

Remarks

The local port value may be set to 0 to allow the socket to pick the next available port. The LocalPort property must be set prior to connecting to the remote host . If VBStak is to be used as a server the LocalPort property must be set to the value that will be used by the remote computer to connect to the server.

Standard port values are defined in the Services file distributed with Winsock. A user defined port may be used if it is in above the IPPORT_RESERVED (1024) range.

Data Type

Integer

Debug Property

Description

Sets and returns the socket debug setting. Read/write.

Visual Basic

[form.]*VbStakControl .* Debug *[= debugSetting%]*

Remarks

VBStak debug mode displays messages indicating the phases of the Action property actions.

Data Type

Integer (Boolean)

Status Property

Description

Returns a value indicating the state of the VBStak control. Read only at run time.

Visual Basic

[form.]*VbStakControl .* Status

Remarks

The integer constants are defined in VBStak.txt

STAK_STATUS_IDLE - Control has been loaded but not initialized.

STAK_STATUS_INITIALIZED - VBStak has been initialized.

STAK_STATUS_OPENED - Socket is opened.

STAK_STATUS_BOUND - Socket bound to protocol.

STAK_STATUS_CONNECTED - Socket has conected to a host .

STAK_STATUS_SENDING - Sending to host.

STAK_STATUS_WAITING - Waiting for response.

STAK_STATUS_REC_READY - Input available.

Data Type

Integer

Socket Property

Description

Returns the handle of the local socket. Run time read/write.

Visual Basic

[form.]*VbStakControl .* **Socket**

Remarks

The number of the socket assigned by the Winsock dll.

Data Type

Integer

ProtocolName Property

Description

Sets and returns the name of the TCP/IP protocol to use. Read/write.

Visual Basic

[form.]*VbStakControl .* **ProtocoName** *[= protocolString$]*

Remarks

The Protocol property defines the symbolic value for the protocol to use in TSP/IP communication. The common values are TCP and UDP.

The symbolic protocol name is used to retrieve the numeric Protocol property through an Action property setting with the STAK_ACTION_GET_PROTCOL value.

Data Type

String

Message Event

Description

Notifies the application that an event has occured which may require servicing.

Visual Basic

*Sub VBStakControl_*Message *(messageNumber as integer)*

Remarks

This is the entry point for Winsock message postings. Asynchronous calls to the Winsock dll are routed back the VBStak control upon completion. The VBStak fires the message event to notify your program of the message. The messages are:

FD_ACCEPT - A connection has been requested by a client on a VBSTAK which has been enabled for listening. See Action property STAK_ACTION_LISTEN.

FD_CLOSE - The socket has been closed at the host end. This occurs with services such as Daytime which respond with a single datagram .

FD_CONNECT - Socket has completed a connect to a host.

FD_OOB - Out of band data has arrived on the socket.

FD_READ - Socket data is available for reading. Data can be accessed through the Input tproperty.

FD_WRITE - Socket has completed the last write operation.

STAK_EVENT_HOST - The STAK_ACTION_GET_HOST action has completed.

STAK_EVENT_SERVICE - The STAK_ACTION_GET_SERVICE action has completed. If an error occurred the Error property will contain the error value.

STAK_MESSAGE_PROTOCOL - The STAK_ACTION_GET_PROTOCOL has completed.

See Also: ActionProperty, Errors, ErrorProperty, InputProperty

Troubleshooting

The following list is a quick troubleshooting guide.

Cannot locate host
  Check:
    The existence of \windows\hosts file.
    Host name entry in the form:
      ipAddress(nnn.nnn.nnn.nnn form) hostName
    Correct host name.

Cannot locate service/port number
  Check:
    The existence of \windows\services file.
    Entry of the service you are trying to connect to such as:
      echo 7/tcp
    Host - /etc/services file.

Cannot locate the protocolName
  Check:
    The existence of \windows\protocol file.
    The entry of the protocol you want such as:
      tcp 6 TCP # Transmission control protocol
    Host - /etc/protocol file.

Cannot find Winsock library.
  Check:
    Installed Winsock library. The TCP/IP WFWT32.EXE library can be downloaded from
      Compuserve and other services.
    Installed protocol - see network setup for WFW.

Cannot connect to host
  Check:
    Network and host availabiltity - use the Telnet function built into the TCP/IP library to
      connect to the host.
    Host services, inetd running.
    Host address correct - check \windows\hosts file.

InputLen Property

Description

Define the number of bytes to receive using the Input property. Read/write.

Visual Basic

[form.]*VbStakControl .* **InputLen** *[= inputLength%]*

Remarks

To receive the complete buffer set InputLen to 0. To check the number of bytes available in the input buffer use the InBufferCount property prior to using the Input property.

Data Type

String

InBufferCount Property

Description

Returns the number of bytes of data in the input buffer.

Visual Basic

[form.]*VbStakControl .* **InBufferCount**

Remarks

InBufferCount can be used to check for a complete fixed-length string. If you use the InputLen property to set the size of the input string you should always check the value of the InBufferCount prior to accessing the Input property.

Data Type

Integer

LocalAddress Property

Description

Sets and returns the IP address of the local socket.

Visual Basic

[form.]*VbStakControl .* **LocalAddress** *[= ipAddress$]*

Remarks

The primary use of the LocalAddress Property is with multi-homed servers such as Compuserve. The IP address assigned to the local socket changes on each connection. This information is valid only after a successful STAK_ACTION_CONNECT.

Data Type

String

MasterSocket Property

Description

Returns/sets the handle of the master server socket associated with the socket. Run time read only.

Visual Basic

[form.]*VbStakControl .* **MasterSocket**

Remarks

The number of the master socket used to create the secondary data socket. This property is used by VBStak to restore the server socket to the listening state after a data transaction is complete. The main use of this property for the VB programmer is in debugging server applications.

Data Type

Integer

Glossary

D
datagram

H
host

I
IP address

S
service

W
well known service

host
A TCP/IP service provider (server).

service
A program function made available to outside computers. The services are typically defined in the host's /etc/services database.

datagram
A single TCP/IP transmission chunk containing the source, destination, port, and data information.

well known service
A predefined list of services provided by typical Unix systems.

IP address

The unique numeric address assigned to a computer using the TCP/IP protocol. The VBstak text representation of the IP address is the dotted address form: 123.456.789.012.