VBWsk - Winsock Custom Control

Version 0.3   27.1.94

Notice:
This is an ALPHA TEST VERSION of the control. The behaviour of the control may change significantly in the final release version - you have been warned! Click here for change details.

Description

The Winsock custom control provides access to the Windows Sockets network programming API. The control is shown here as it appears in the toolbox and as an icon on a form in design mode.

File name

VBWSK.VBX

Properties

Socket
Host
Port
Data
Sent
BufferSize
EOL
EOF
Blocking

Events

Read
Write
OutOfBand
Accept
Connect
Close

Errors
Copyright & Conditions
Future developments

Remarks

Some additional support routines and definitions for this control are contained in the file VBWSK.BAS. You should include this module in your project when using the VBWsk control.

The control provides access to the Winsock API. Each instance of the control can be either "idle" (not associated with a socket) or "active" (associated with a socket.) A control is made active by either connecting to a remote host, listening for incoming connections or accepting an incoming connection. The control can be used in blocking and non-blocking (event driven) modes. The default is non-blocking mode.

To connecting to a remote host set the Port property to the service name or port number, then set the Host property to the remote hosts name or dotted-decimal IP address. The act of assigning data to the Host property causes the connection attempt to be made. If in non-blocking mode the Connect subroutine will be called when the connection is made.

To make the control listen for incoming connections, set the Port property to the service name or port number, then set the Host property to an empty string. If in non-blocking mode the Accept subroutine will be called when an incoming connection attempt occurs. To accept an incoming connection, assign the value of the Socket property to the Socket property of a VBWsk control. (This can be either the same control, or a different control.) If in non-blocking mode the Connect subroutine will be called when the connection is made.

When the control is active and connected you can write data to the socket by assigning strings to the Data property, and read data from the socket by reading the value of the Data property.

The control supports a mode of operation known as "line mode." This allows you to read incoming data a "line" at a time. Setting the EOL property to the sequence used to terminate the line (typically CRLF, or chr$(13)+chr$(10)) enables this mode.

To close a socket, set the Socket property to the special value CLOSE_SOCKET.

Changes

From 0.2 to 0.3
- Connecting to a host specified as a dotted-decimal IP address now works properly.
- It was sometimes possible to lose data sent by a remote host if the remote service sent data and closed the socket very quickly. This shouldn't happen now.
- Non-Line mode read buffering error corrected.

From 0.1 to 0.2
- Various documentation errors corrected (doubtless others introduced.)
- Each of the six possible network events (read, write, out-of-band, accept, connect and close) now has it's own event procedure instead of the single parameterised event procedure.
- The LineReady property has now been changed to EOF.
- Sent property returns the number of characters successfully sent by the last assignment to the data property. (It's possible for this to be less than the full amount due to the underlying protocol stack's internal buffering limits.)
- Reading the Host property returns the name of the remote host - particularly useful for validating incoming connections.
- Now works with Visual Basic 2.0 (properly.)
- Error handling and diagnostics are being improved.

Copyright and Conditions

Future Development

This is an alpha release of the control - don't expect too much!

If you have any requests for changes, fixes or ideas for enhancements or additions to the control, please get in touch with the author, or post the request to the alt.winsock newsgroup.

Planned changes:
- Better error reporting and detection.

Socket property

Description

This is used to determine socket identifiers, to close sockets and to accept incoming connections.

Type

Integer

Remarks

Reading this property returns the socket identifier if the socket is currently open, or the special value INVALID_SOCKET if the socket is currently closed.

Setting this property to the special value CLOSE_SOCKET closes the socket. If the socket was already closed, this is silently ignored.

If the property is set to any value other than CLOSE_SOCKET it is assumed that the new value is the identifier of a socket listening for incoming connections, and an attempt is made to accept a connection on the new socket. If this succeeds the property is set to the identifier of the socket created as a result of the incoming connection. The same control can be used to listen for and accept the connection, in which case the listening socket is closed after the connection is accepted. Attempting to accept an incoming connection on an active control other than the listener causes an in-use error.

Host property

Description

This is the host address of the other side of the current connection.

Type

String

Remarks

Setting this property to the address of a host causes the control to attempt to connect to the host on the port specified by the setting of the Port property. Setting this to an empty string causes the control to listen for incoming connections on the port specified by the Port property. Note that in either case you **must** set the value of the Port property first. If the socket is already connected this causes an in-use error.

Reading the value of this property returns the name of the host to which the socket is connected, or an empty string if the control is idle.

Port property

Description

This specifies the port to connect to when the control is used as a client, or the port to listen on when the control is used as a server.

Type

String

Data property (default)

Description

This is used to write data to and read data from the socket.

Type

String

Remarks

If the EOL property is currently set to an empty string:

Reading the value of this property returns data read from the socket. If no data is available an empty string is returned. The value of the BufferSize property sets the maximum amount of data that can be read at a time.

If the EOL string is **not** an empty string:

Reading the value of this property returns the first complete line in the control's buffer (without the EOL string.) If no complete line is available an end-of-file error occurs. You can check the EOF property to avoid this error.

Assigning data to the property writes the data to the socket (note that any terminator specified by the EOL property is **not** automatically appended.) Because the underlying network transport has only a limited amount of buffer space it is possible for less than the full amount of the data to be sent. Check the SendCount property to find out how much data was actually sent.

Sent property

Description

This property specifies the number of bytes actually sent after assigning data to the Data property.

Type

Integer

Remarks

If the underlying network transport has run out of buffer space this may be less than the entire amount of data assigned to the Data property may actually be sent. This propery is set to the amount sent, or to SOCKET_ERROR if an error occured during the write.

Your application should check this property after sending data, and take actions to ensure that any remaining data is sent when buffer space becomes available.

BufferSize property

Description

This specifies the size of the control's input buffer. The size of this buffer sets the maximum amount of data that can be read at one time. This property is read-only at run time.

Type

Integer

EOL property

Description

This controls the line mode operation of the control.

Type

String

Remarks

Setting this property to a non-empty string puts the control into line mode. In this mode reading the Data property will only return complete lines that are terminated by the sequence of characters in this string, and Read events are only generated when a complete line is available in the buffer.

EOF property

Description

If data can be read from the Data property this control has the value False, otherwise the value is True. If the control is in line mode this property will be True unless a complete line is available.

Type

Integer (boolean)

Blocking property

Description

This determines whether the socket is used in blocking or non-blocking mode.

Type

Integer (boolean)

Remarks

Setting this property to False (the default) causes the control to be used in non-blocking mode. In this mode writing data to or reading data from the Data property will not freeze your application while waiting for new data, and network events (such as the arrival of new data, or the connection being closed) will cause an appropriate event subroutine to be called.

Setting this property to True causes the control to operate in blocking mode. In this mode your application will freeze if it attempts to read the Data property when no data is available, and will stay frozen until data arrives. The event subroutines are not called.

Read event

Description

Occurs when in non-blocking mode and data is available to be read.

Write event

Description

Occurs when it is possible to write data to the control. This event is called after a write was only partially successful because of a lack of buffer space in the underlying network transport.

OutOfBand event

Description

Occurs when out-of-band data arrives. Currently there is no way of accessing out-of-band data directly using the control - use the Winsock API recv() function directly along with the Socket property to retrieve out-of-band data.

Accept event

Description

Occurs when listening for incoming connections and a connection attempt occurs. The subroutine should accept the incoming connection by assigning the value of the Socket property to the Socket property of a VBWsk control (either the same one or a different one.)

Connect event

Description

Occurs when a connection has been established.

Close event

Description

Occurs when the socket is closed either by the local end or the remote host.

Errors

Code	Message
20098	Socket is currently inactive. This error occurs when you try to perform an operation that requires the socket to be active, e.g. reading data from the Data property.
20099	Socket is currently active. This error occurs when you try to perform some operation that requires the socket to be idle, e.g. connecting to a host.
21000 -22999	Winsock DLL errors. These are error codes returned directly from the Winsock DLL (see WINSOCK.H and the Winsock specification.)