VTSocket (Windows Sockets) Control

Properties Events Error Codes About

Description

The VTSOCKET control provides network communications for your application by allowing transmission and reception of TCP/IP data via the Windows Sockets (winsock) standard.

Object Type

WinSock

Remarks

The windows socket control provides the TCP/IP functionality for communicating as either a server process or a client process or both. Event driven and polled communications are supported. BSD style streaming sockets are fully supported; out of band data, broadcasting, and UDP type sockets are not supported.

Client communications allow applications written in Visual Basic or Visual C++ to communicate with server processes. A client process may be established with VTSocket using as few as 6 lines of procedural code (even less if host names and ports are assigned at design time).

For example:

```
Sub Form_Load ( )
    Client1.IPName = "remote.host"
    Client1.Port = 1234
    Client1.Open = True
    Client1.Send = "This is a test message"
    answer$ = Client1.Recv
    Client1.Open = False
```

A receive event may also be defined to accept input from remote hosts on an event-driven basis.

Server communications allow applications written in Visual Basic or Visual C++ to communicate with several client processes at the same time. A server process relies on event procedures for notification when a client process connects or disconnects. A control array is used to differentiate multiple client connections from one another.

For example:

```
Sub Form_Load ( )
    Server1(0).Port = 1234
    Server1(0).Open = True
Sub Server1_Connect (Index As Integer, ID As Integer)
    Load Server1(ID)
Sub Server1_Recv(Index As Integer)
    indata$ = Server1(Index).Recv
Sub Server1_Disconnect(Index As Integer)
    Unload Server1(Index)
```

Each client control corresponds to one connection to a remote host based on the host name and port number. If you need to access more than one host at the same time multiple client controls should be used. Multiple client controls may exist as separate controls, or as elements of a single control array.

Each server control corresponds to one port number at which multiple connections may be accepted.   A server control ***MUST*** be defined as element zero of a control array. Connections received by this control are then serviced by establishing a new (non-zero) control array element for each connection.   The Visual Basic   *Load* and *Unload* commands are used for this purpose.

Distribution Note When you create and distribute applications that use the VTSocket control, you should install the file VTSOCKET.VBX in the customer's Microsoft Window's \SYSTEM sub directory.

Applications developed using the VTSocket control will cease to properly function after 30 days unless a VTPROD.LIC module is present on the development system.   A VTPROD.LIC module may be obtained by licensing the VT Socket control from Visual Technology Products.

About the VTSocket Control

Properties Events Error Codes About

The VT Socket control was developed by Visual Technology Products. For additional information, they may be contacted at:

Visual Technology Products
4317 Olley Lane
Fairfax, VA 22032

Fax: (703) 978-7389

Internet mail: vtprod@access.digex.com

VTSocket Control Properties

Properties Events Error Codes About

Properties	Description
Client_or_Server	Sets and returns the operating mode of this connection (design time only access)
Description	Returns the description information provided by the winsock.dll in response to a WSAStartup() function call. This is usually the vendor name and version number of the windows sockets DLL that is in use.
Interval (effective	Sets and returns the timeout interval for linger during close only if Linger is set to true)
IPName	Sets and returns the remote host domain name or IP number
Linger	Sets and returns an option to wait for data transmission completion during close processing
MyIP	Returns your local IP number as an unsigned long integer value.
Open	Sets and returns the connection status of this control. This event immediately initiates connection processing when set to True, and immediately terminates an active connection when set to False.
Port	Sets and returns the port number for a remote connection
Recv	Returns and removes data received from a remote host (read only)
RecvBufSize	Sets and returns the size of buffers used during data reception
Send	Receives a string of characters to be transmitted to a remote host (write only)
SendBufSize	Sets and returns the size of buffers used during data transmission

VTSocket Control Events

Properties Events Error Codes About

Events	Description
Connect	The connect event is invoked in response to a successful open (WinSock.Open = True) for the local host, or when the VTSocket control receives a request for connection from a remote host.
Disconnect	The disconnect event is invoked in response to a successful close (WinSock.Open = False) for the local host, or when the VTSocket control detects a disconnect notification from a previously connected remote host.
Recv	The receive event is invoked when the VTSocket control detects the availability of data sent by a currently connected remote host. This event is active for both client and server operating modes.

Client_or_Server Property

Properties Events Error Codes About

Description

Sets and returns the socket communications mode for this control. A setting of Client (0) causes the control to establish active communications with a remote host when connection is requested via the Open property. A setting of Server (1) causes the control to listen for a connection request initiated by a remote host.

Usage

setting % = [*form*.]*Winsock*.**Client_or_Server**

Remarks

Control is read-only at run time. The following table lists the Client_or_Server property settings for the VTSocket control:

Setting	Description
0	(Default) Control is a client socket.
1	Control is a server socket.

Data Type

Integer (Enumerated)

Description Property

Properties Events Error Codes About

Description

Returns the description information provided by winsock.dll in response to a WSAStartup() function call.   This is usually the vendor name and version number of the windows sockets DLL that is in use.

Usage

description $ = [*form*.]*Winsock*.**Description**

Remarks

Control is read-only at both design-time and run-time.

Data Type

String

Interval

Properties Events Error Codes About

Description

Sets and returns the timeout interval for linger during close (effective only if Linger is set to true). Interval is the number of seconds to wait for a graceful close. If the interval property is set to zero and the linger property is true, a hard close will be forced.

Usage

[*form*.]*Winsock*.**Interval** = *interval* %
interval % = [*form*.]*Winsock*.**Interval**

Data Type

Integer

IPName Property

Properties Events Error Codes About

Description

The IPName property sets and returns the name or IP number of a remote host to be contacted when the open property is set true.

Usage

[*form*.]*Winsock*.**IPName** = *ipname* $
ipname $ = [*form*.]*Winsock*.**IPName**

Remarks

IPName may be set and returned for both client and server connections, but is ignored when a server connection is specified.

Data Type

String

Linger Property

Properties Events Error Codes About

Description

When the Client_or_Server property is set to Client (0), this property sets and returns the port number of a remote host to be contacted when open processing is initiated.   When the Client_or_Server property is set to Server (1), this property sets and returns the port number at which the control should listen for remote host connection requests.

Usage

[*form*.]*Winsock*.**Linger** = { **True** | **False** }
linger % = [*form*.]*Winsock*.**Linger**

Remarks

The following table lists the Linger property settings for the VTSocket control:

Setting	**Description**
False	(Default) Do not wait for data to quiesce during close.
True	Wait "Interval" seconds for data to quiesce during close.

Data Type

Integer (Boolean)

MyIP Property

Properties Events Error Codes About

Description

The MyIP property returns the local IP address.

Usage

*myipnumbe*r % = [*form*.]*Winsock*.**MyIP**

Remarks

The MyIP property is read only, and is available at run-time only. It provides a valid local IP number only while a client or server connection is open (the Open property is set to True).

Data Type

Long

Open Property

Properties Events Error Codes About

Description

When the Client_or_Server property is set to Client (0), this property initiates a connection (or terminates a connection) with the remote host identified by the IPName and Port property settings. When the Client_or_Server property is set to Server (1), this property causes the control to listen (or terminate listening) for connection requests from remote hosts using the port identified by the Port property.

Usage

[*form*.]*Winsock*.**Open** = { **True** | **False** }
connection_status % = [*form*.]*Winsock*.**Open**

Remarks

The Open property is available at run-time only. The following table lists the Open property settings for the VTSocket control:

Setting	**Description**
False	Terminate a currently connected TCP/IP socket.
True	Attempt to connect a currently disconnected TCP/IP socket..

Data Type

Integer (Boolean)

Port Property

Properties Events Error Codes About

Description

When the Client_or_Server property is set to Client (0), this property sets and returns the port number of a remote host to be contacted when open processing is initiated.   When the Client_or_Server property is set to Server (1), this property sets and returns the port number at which the control should listen for remote host connection requests.

Usage

[*form*.]*Winsock*.**Port** = *port_number* %
port_number % = [*form*.]*Winsock*.**Port**

Data Type

Integer

Recv Property

Properties Events Error Codes About

Description

The Recv property receives and removes data from a currently connected remote host. This property may be used to poll for pending data by repeatedly reading the property, or may be used within the Recv event procedure to read pending data.

Usage

received_data $ = [*form*.]*Winsock*.**Recv**

Remarks

The Recv property is a read-only property, and is available only at run-time.

Data Type

String

RecvBufSize Property

Properties Events Error Codes About

Description

The RecvBufSize property sets and returns the size in bytes of the buffer used by windows sockets to receive inbound data from a remote host.

Usage

[*form*.]*Winsock*.**RecvBufSize** = *buffer_size* %
buffer_size % = [*form*.]*Winsock*.**RecvBufSize**

Data Type

Integer

Send Property

Properties Events Error Codes About

Description

The Send property is used to transmit data to a currently connected remote host.

Usage

[*form*.]*Winsock*.**Send** = *send_data* $

Remarks

The Send property is a write-only property, and is available only at run-time.

Data Type

String

SendBufSize Property

Properties Events Error Codes About

Description

The SendBufSize property sets and returns the size in bytes of the buffer used by windows sockets to transmit outbound data to a remote host.

Usage

[*form*.]*Winsock*.**SendBufSize** = *buffer_size* %
buffer_size % = [*form*.]*Winsock*.**SendBufSize**

Data Type

Integer

Connect Event

Description

Occurs when a remote host attempts to connect to a VT Socket control for which the Client_or_Server property is set to Server, and the Open property is True. Also occurs in response to a successful connection by the local host when the Open property is set True.

Syntax

Sub *Winsock*_**Connect** (Index **As Integer**, ID **As Integer**)

Remarks

When the Client_or_Server property is set to Server, the index value for a Connect event should always be zero, indicating that the connection is for the listening control entry in a control array. ID contains an index value to be used in communicating with the newly connected remote host. A new instance of the control should be created to handle communications with the remote host. For example:

```
Sub Winsock_Connect (Index As Integer, ID As Integer)
    Load Winsock(ID)
```

Disconnect Event

Properties Events Error Codes About

Description

Occurs when a remote host terminates a connection to a VT Socket control for which the Client_or_Server property is set to Server, and the Open property is True.   Also occurs in response to a successful disconnection by the local host when the Open property is set False.

Syntax

Sub *Winsock*_**Disconnect** (Index **As Integer**)

Remarks

When the Client_or_Server property is set to Server, the Index value for a Disconnect event should always be non-zero, indicating that the disconnect has been received from an instance of the control that was previously created during Connect event processing. The instance of the control that was previously created to handle communications with the remote host should be removed.

For example:

```
Sub Winsock_Disconnect (Index As Integer)
    Unload Winsock(Index)
```

Recv Event

Properties Events Error Codes About

Description

Occurs when a remote host sends data to a VT Socket control.

Syntax

Sub *Winsock*_**Recv** ( )

Remarks

The Recv event provides a convenient way for an application to recognize that new data is available from a remote host. The pending data may be immediately fetched using the Recv property, or may be deferred to a later time. However, no further Recv events will be posted until the currently pending data is fetched.

For example:

```
Sub Winsock_Recv ( )
    recvdata$ = Winsock.Recv
```

Error Codes and Messages

Properties Events Error Codes About

Remarks

All errors returned by VTSocket are, in fact, errors that originate with the Windows Sockets DLL (winsock.dll). These errors are defined in the following table. Error codes and tags are also defined in the file named VTSOCKET.INC, in a format suitable for inclusion Visual Basic applications. (The error codes are already defined in winsock.h for inclusion in Visual C++ applications.)

Err	Message	Err	Message	Err	Message
10004	INTR	10009	BADF	10013	ACCESS
10014	FAULT	10022	INVAL	10024	MFILE
10035	WOULDBLOCK	10036	INPROGRESS	10037	ALREADY
10038	NOTSOCK	10039	DESTADDRREQ	10040	MSGSIZE
10041	PROTOTYPE	10042	NOPROTOOPT	10043	PROTONOSPT
10044	SOCKTNOSPPT	10045	OPNOTSUPP	10046	PFNOSUPPORT
10047	AFNOSUPPORT	10048	ADDRINUSE	10049	ADDRNOTAVAIL
10050	NETDOWN	10051	NETUNREACH	10052	NETRESET
10053	CONNABORTED	10054	CONNRESET	10055	NOBUFS
10056	ISCONN	10057	NOTCONN	10058	SHUTDOWN
10059	TOOMANYREFS	10060	TIMEDOUT	10061	CONNREFUSED
10062	LOOP	10063	NAMETOOLONG	10064	HOSTDOWN
10065	HOSTUNREACH	10066	NOTEMPTY	10067	PROCLIM
10068	USERS	10069	DQUOT	10070	STALE
10071	REMOTE	10091	SYSNOTREADY	10092	VERNOTSPT
10093	NOTINITIALISED	11001	HOST_NOT_FOUND	11002	TRY_AGAIN
11003	NO_RECOVERY	11004	NO_DATA		