



IPPort Control (TM) Version 1.0

[Properties](#)

[Events](#)

[Error Codes](#)

[Copyright & Registration](#)

Description

The IPPort Control facilitates TCP/IP communications by providing an easy interface to Winsock functions. It allows a client application to communicate with a server using stream sockets.

File Name

IPPORT.VBX

Object Type

IPPort

Remarks

IPPort needs a Winsock 1.1 compliant TCP/IP subsystem. WINSOCK.DLL must be available in the system before the control can be loaded. The Winsock version supported must be at least 1.1.

Our main goal in designing IPPort was ease of use. The control has a minimum of properties, and four events: [Connected](#), [ReadyToSend](#), [DataIn](#), and [Disconnected](#). The events are relatively self-explanatory. The connection is attempted by setting the [Connected](#) property to **True**, and then waiting for the [Connected](#) event. The destination is defined by setting the [Port](#) property and either the [HostName](#) or the [HostAddress](#) property. Data is sent by assigning the data string to the [DataToSend](#) property.

The operation of the control is almost completely asynchronous. All the calls except the ones that deal with host name and address resolution, operate through Windows messages (no blocking calls). The gain in performance is considerable when compared to using blocking calls. The only drawback is what some people perceive as "unnatural" programming, but if you were brave enough to come to this sentence, you will be doing fine.

If you have any questions, suggestions, or need any assistance, you can contact us via email at devsoft@aol.com. We will try to answer all messages, however, messages from registered users will have higher priority, so please include your serial number in your message for faster service.

Copyright Notice

The IPPort Control (TM) is Copyright 1994 by DevSoft. - All rights reserved.

Registration Information

The registration fee for IPPort is \$25. You can register by email, phone, or mail. Payments can be made by check, money order, or credit card. For more information, please select one of the following options:

Mail Orders

Credit Card Orders

COMPUSERVE SWREG Orders

By registering, you will be able to distribute royalty-free executables that use IPPort. You will also receive prioritized online support, code samples and tips from our mailing list, free minor updates (1.x), discounts on major updates (2.x, 3.x, etc..), and our thanks. And if for any reason you find our registration fee unsuitable or unacceptable, please write to us. Give us a good reason why, and we would be happy to send you a free copy.

Licensing

i) shareware version

You may use the shareware version of IPPort for up to 30 days in your design environment for evaluation purposes only. You may copy and distribute it freely as long as all the files in the package, including the demo programs and this help file are distributed with it and no changes or additions of any kind are made to the original package.

There is no charge for any of the above, however, you are specifically prohibited from charging, or requesting donations for any copies, however made, and from distributing IPPort and/or its accompanying files with other products (commercial or otherwise) without prior written permission from DevSoft.

ii) registered version

As a registered user, you can use IPPort in your design environment as well as distribute executables that use IPPort as a runtime component. DevSoft asks for no royalties or runtime fees for such distribution. The only requirement is that you distribute a license file (ippport.lic) which will bear your unique serial number. You will obtain that file upon registration. We also ask you as a courtesy to distribute this help file with your application, but you are not required to do so.

Please note that the rights to ippport.lic are not transferable: users of your application cannot legally use ippport.lic in their own applications, or distribute their own code using the ippport.lic with your serial number on it. Only registered users can distribute executables using IPPort.

You may install only one registered copy of IPPort in a single workstation at any time. Use of a registered copy in more than one workstation is against the terms of this licensing agreement. In particular, you are specifically prohibited from distributing a registered version of IPPort, except as a runtime component of one of your applications.

Limitation of liability:

THIS SOFTWARE AND THE ACCOMPANYING FILES ARE SOLD "AS IS" AND WITHOUT WARRANTIES AS TO PERFORMANCE OF MERCHANTABILITY OR ANY OTHER WARRANTIES WHETHER EXPRESSED OR IMPLIED. THE EXTENT OF LIABILITY OF THE SELLER IS HEREBY LIMITED EXCLUSIVELY TO PRODUCT REPLACEMENT OR REFUND OF PURCHASE PRICE. IN PARTICULAR, IN NO EVENT SHALL DEVSOFT BE

LIABLE TO YOU FOR ANY DAMAGES, INCLUDING ANY LOSS OF PROFITS, LOSS OF DATA, INCLUDING BUT NOT LIMITED TO SPECIAL, INCIDENTAL, CONSEQUENTIAL, OR INDIRECT DAMAGES ARISING FROM THE USE OF THIS SOFTWARE.

Mail Orders

You can order IPPort by mail by sending a check or money order for US\$ 25 to the following address:

DevSoft
P.O.Box 13821
Research Triangle Park, NC 27709
U.S.A.

INTERNET: devsoft@aol.com COMPUSERVE: 75244,2736

All funds must be in US dollars drawn on a US bank. For faster service, please specify an email address with your order.

Credit Card Orders

You can order with Visa, MasterCard, American Express, or Discover card from Public (software) Library by:

phone: (800) 2424-PSL or (713) 524-6394

fax: (713) 524-6398

COMPUSERVE mail: 71355,470

Please mention **product number 11682**.

The above numbers are for orders only. Any other inquiries, including inquiries about the status of the shipment of the order, refunds, registration options, product details, technical support, volume discounts, dealer pricing, site licenses, etc., must be directed to:

DevSoft
P.O.Box 13821
Research Triangle Park, NC 27709
U.S.A.

INTERNET: devsoft@aol.com COMPUSERVE: 75244,2736

Please specify your card number and expiration date with your order, as well as the name of the package you are ordering and product id (#11682).

COMPUSERVE SWREG Orders

You will be able to register through the SWREG forum on CompuServe. At the time this help file is created, we have not yet acquired an SWREG ID number. This number will be supplied with future updates. For the moment, you can make a search using the keyword **IPPort**.

We apologize for the inconvenience, but we thought it would be better to make the product available now, rather than wait until CompuServe assigns an ID.

Properties

| | |
|----------------------|-----------------------|
| <u>*AcceptData</u> | Left |
| <u>*BytesSent</u> | <u>*LocalHostName</u> |
| <u>*Connected</u> | <u>*LocalPort</u> |
| <u>*DataToSend</u> | Name |
| <u>*EOL</u> | <u>*OutBufferSize</u> |
| <u>*HostAddress</u> | <u>*Port</u> |
| <u>*HostName</u> | Top |
| <u>*InBufferSize</u> | <u>*WinSockInfo</u> |

Index

AcceptData Property

Description

Setting the property to **False** temporarily disables data reception (and the [DataIn](#) event). Setting the property to **True** reenables data reception.

Usage

[form.][ipportcontrol.]AcceptData[= value]

Default Value

True

Remarks

Use the **AcceptData** property with caution. If the data reception is disabled for too long, the other side might abort the connection.

Also, please note that when **AcceptData** is set to **False**, and [EOL](#) is not empty (""), the effect of **AcceptData = False** is not immediate. First the data in the internal buffer are delivered via the [DataIn](#) event.

This property is not available in design mode.

Data Type

Boolean (Integer)

BytesSent Property

Description

Shows the number of bytes sent after the last assignment to the [DataToSend](#) property.

Usage

[form.][\[ipportcontrol.\]BytesSent](#)

Default Value

0

Remarks

BytesSent shows how many bytes were sent after the last assignment to [DataToSend](#) . Check the [DataToSend](#) property for more information.

This property is read-only and not available in design mode.

Data Type

Integer

Connected Property

Description

Setting the property to **True** initiates a connection request. Setting it to **False** breaks the current connection (if any).

Usage

*[form.][ipportcontrol.]***Connected**[= *value*]

Default Value

False

Remarks

Connected is an action property. Use it to create and destroy connections. Before initiating a connection, make sure that the HostName (or HostAddress) and Port properties are set to the appropriate values. Otherwise, an error will occur.

After setting the property to **True**, wait for the **Connected** or ReadyToSend event before trying to send any data.

This property is not available in design mode.

Data Type

Boolean)(Integer)

DataToSend Property

Description

DataToSend is an action property. Assigning a Visual Basic string to this property makes the control send the string to the remote host (note that a Visual Basic string can contain control as well as NULL characters).

Usage

[*form.*][*ipportcontrol.*]**DataToSend**[= *value*]

Default Value

"" (*empty string*)

Remarks

If you are sending data to the remote host faster than it can process it, or faster than the network bandwidth allows, the outgoing queue might fill up. When this happens, **DataToSend** fails with error 25036: "[10035] Operation would block" (WSAEWOULDBLOCK). The BytesSent property shows how many bytes were sent (if any). You can trap the error, and then try to send the data again. If 0 bytes were sent, then you can wait for the ReadyToSend event before attempting to send data again. (However, please note that ReadyToSend is **not** fired when part of the data are successfully sent).

This property is not available in design mode.

Data Type

String

EOL Property

Description

Used to break the incoming data stream into chunks separated by the string assigned to **EOL**. For more information, see the [DataIn](#) event.

Usage

```
[form.][ipportcontrol.]EOL[ = value]
```

Default Value

"" (*empty string*)

Remarks

The **EOL** property is especially useful with ASCII files. Setting it to Chr\$(10) (newline) enables splitting of an incoming ASCII text stream into lines. In this case, one event is fired for each line received (as well as in packet boundaries). The Chr\$(10) characters are discarded.

EOL is a string. In particular, this means that it can be more than one character long.

Data Type

String

HostAddress Property

Description

Specifies the remote host IP number in Internet dotted format.

Usage

[*form.*][*ipportcontrol.*]**HostAddress**[= *value*]

Default Value

0.0.0.0

Remarks

HostAddress is an action property. Setting it to a string value initiates a domain name resolution call (in blocking mode). After the call completes, the corresponding IP address (domain name) can be found in the HostName property.

Data Type

String

HostName Property

Description

Specifies the domain name of the remote host.

Usage

*[form.][ipportcontrol.]***HostName**[= *value*]

Default Value

"" (*empty string*)

Remarks

HostName is an action property. Setting it to a string value initiates a domain name resolution call (in blocking mode). After the call completes, the corresponding IP number can be found in the HostAddress property.

Data Type

String

InBufferSize Property

Description

Specifies the size (in bytes) of the receiving queue in the underlying TCP/IP provider.

Usage

[*form.*][*ipportcontrol.*]**InBufferSize**[= *value*]

Default Value

2048

Remarks

This is the size of an internal queue in the TCP/IP provider. You can increase or decrease its size depending on the amount of data that you will be receiving. Increasing **InBufferSize** can provide drastic improvements in performance in some cases.

Some TCP/IP implementations do not support variable buffer sizes. If that is the case, when you set the Connected property to **True** (i.e. attempt a connection), InBufferSize reverts back to its allowable size. The same happens if you attempt to make it too large or too small.

Data Type

Integer

LocalHostName Property

Description

Specifies the domain name of the local host.

Usage

[*form.*][*ipportcontrol.*]**LocalHostName**

Default Value

"" (*empty string*)

Remarks

To find the address of the host, you can set the [HostName](#) property to the value specified by **LocalHostName**. The address will be provided by the [HostAddress](#) property.

This property is read-only.

Data Type

String

LocalPort Property

Description

Specifies the IP port in the local machine.

Usage

[*form.*][*ipportcontrol.*]**LocalPort**[= *value*]

Default Value

0

Remarks

The **LocalPort** property must be set before a connection is attempted. Setting it to 0 (default) enables Winsock to choose a port at random. The chosen port will be shown by the **LocalPort** property after the connection is made.

LocalPort cannot be changed once a connection is made. Any attempt to set the **LocalPort** property when a connection is active will generate an error.

The **LocalPort** property is useful when trying to connect to services that require a trusted port in the client side. An example is the remote shell (rsh) service in UNIX systems.

Data Type

Integer

OutBufferSize Property

Description

Specifies the size (in bytes) of the outgoing queue in the underlying TCP/IP provider.

Usage

[*form.*][*ipportcontrol.*]**OutBufferSize**[= *value*]

Default Value

2048

Remarks

This is the size of an internal queue in the TCP/IP provider. You can increase or decrease its size depending on the amount of data that you will be receiving. Increasing **OutBufferSize** can provide drastic improvements in performance in some cases.

Some TCP/IP implementations do not support variable buffer sizes. If that is the case, when you set the Connected property to **True** (i.e. attempt a connection), OutBufferSize reverts back to its allowable size. The same happens if you attempt to make it too large or too small.

Data Type

Integer

Port Property

Description

Specifies the IP port of the remote host.

Usage

[form.][ipportcontrol.]Port[= value]

Default Value

0

Remarks

The **Port** property must be set before a connection is attempted. The **Port** cannot be changed once a connection is made. Any attempt to set the **Port** property when a connection is active will generate an error.

Data Type

Integer

WinsockInfo Property

Description

Provides information about the underlying TCP/IP (Winsock) provider.

Usage

*[form.][ipportcontrol.]***WinsockInfo**

Default Value

"" (*empty string*)

Remarks

WinsockInfo returns a string up to 256 bytes long provided by the underlying Winsock provider. The property is read-only.

Data Type

String

Events

*Connected

*DataIn

*Disconnected

*ReadyToSend

Connected Event

Description

Occurs when a connection is made to the remote host.

Syntax

Sub *ipportcontrol_***Connected**(*StatusCode* **As Integer**, *Description* **As String**)

Remarks

If the connection succeeds, *StatusCode* is 0, and *Description* is "OK".

If the connection fails for any reason, *StatusCode* has the error code returned by Winsock. *Description* contains a description of this code. The value of *StatusCode* is obtained by adding 15001 to the corresponding Winsock error code.

Please refer to the [Error Codes](#) section for more information.

DataIn Event

Description

Occurs when data arrives from the remote host.

Syntax

Sub *ipportcontrol*_**DataIn**(*Text* **As String**, *EOL* **As Integer**)

Remarks

Trapping the **DataIn** event is your only chance to get the data coming from the other end of the connection. The incoming data are given in *Text*. *Text* is a Visual Basic string, and as such might be considered as a binary chunk of data with length Len(*Text*).

EOL indicates whether an *EOL* string was found on the end of *Text* or not. If the *EOL* string was found, then *EOL* is **True**. If *Text* was obtained at the end of a segment of data received from Winsock, then *EOL* is **False**. Please note that this also means that more than one **DataIn** event with *EOL* set to **False** can be received during a connection.

If the *EOL* property is "" (empty), then *EOL* can be disregarded. For more information on *EOL*, please refer to the description of the *EOL* property.

Disconnected Event

Description

Occurs when the connection to the remote host is closed (broken).

Syntax

Sub *ipportcontrol_Disconnected*(*StatusCode* **As Integer**, *Description* **As String**)

Remarks

If the connection is broken normally, *StatusCode* is 0, and *Description* is "OK".

If the connection is broken for any other reason, *StatusCode* has the error code returned by Winsock. *Description* contains a description of this code. The value of *StatusCode* is obtained by adding 15001 to the corresponding Winsock error code.

Please refer to the [Error Codes](#) section for more information.

ReadyToSend Event

Description

Indicates that the underlying TCP/IP subsystem is ready to receive data.

Syntax

Sub *ipportcontrol_ReadyToSend()*

Remarks

The ReadyToSend event is fired when the connection is ready to accept data again after a failed DataToSend. The event is also fired immediately after the connection to the remote host is made.

Error Codes

The following is a list of the trappable errors fired by IPPort:

IPPort Internal Errors

- 20101 You cannot change the Port at this time. Close the connection first.
- 20102 You cannot change the HostName at this time. Close the connection first..
- 20103 You cannot change the HostAddress at this time. Close the connection first.
- 20104 Already connected. Close the current connection first in order to reconnect.
- 20105 Winsock error code outside normal range.
- 20106 You cannot change the LocalPort at this time. Close the connection first.

Winsock Errors

The error message descriptions show the corresponding Winsock error number. The corresponding Visual Basic error code can be obtained by adding 15001 to the number displayed in the message and vice-versa.

| | | |
|-------|----------------------|--|
| 25005 | (WSAEINTR) | [10004] Interrupted system call. |
| 25010 | (WSAEBADF) | [10009] Bad file number. |
| 25014 | (WSAEACCES) | [10013] Permission denied. |
| 25015 | (WSAEFAULT) | [10014] Bad address. |
| 25023 | (WSAEINVAL) | [10022] Invalid argument. |
| 25025 | (WSAEMFILE) | [10024] Too many open files. |
| 25036 | (WSAEWOULDBLOCK) | [10035] Operation would block. |
| 25037 | (WSAEINPROGRESS) | [10036] Operation now in progress. |
| 25038 | (WSAEALREADY) | [10037] Operation already in progress. |
| 25039 | (WSAENOTSOCK) | [10038] Socket operation on non-socket. |
| 25040 | (WSAEDESTADDRREQ) | [10039] Destination address required. |
| 25041 | (WSAEMSGSIZE) | [10040] Message too long. |
| 25042 | (WSAEPROTOTYPE) | [10041] Protocol wrong type for socket. |
| 25043 | (WSAENOPROTOOPT) | [10042] Bad protocol option. |
| 25044 | (WSAEPROTONOSUPPORT) | [10043] Protocol not supported. |
| 25045 | (WSAESOCKTNOSUPPORT) | [10044] Socket type not supported. |
| 25046 | (WSAEOPNOTSUPP) | [10045] Operation not supported on socket. |
| 25047 | (WSAEPFNOSUPPORT) | [10046] Protocol family not supported. |
| 25048 | (WSAEAFNOSUPPORT) | [10047] Address family not supported by protocol family. |
| 25049 | (WSAEADDRINUSE) | [10048] Address already in use. |
| 25050 | (WSAEADDRNOTAVAIL) | [10049] Can't assign requested address. |
| 25051 | (WSAENETDOWN) | [10050] Network is down. |
| 25052 | (WSAENETUNREACH) | [10051] Network is unreachable. |
| 25053 | (WSAENETRESET) | [10052] Net dropped connection or reset. |
| 25054 | (WSAECONNABORTED) | [10053] Software caused connection abort. |
| 25055 | (WSAECONNRESET) | [10054] Connection reset by peer. |
| 25056 | (WSAENOBUFS) | [10055] No buffer space available. |
| 25057 | (WSAEISCONN) | [10056] Socket is already connected. |
| 25058 | (WSAENOTCONN) | [10057] Socket is not connected. |
| 25059 | (WSAESHUTDOWN) | [10058] Can't send after socket shutdown. |
| 25060 | (WSAETOOMANYREFS) | [10059] Too many references, can't splice. |
| 25061 | (WSAETIMEDOUT) | [10060] Connection timed out. |
| 25062 | (WSAECONNREFUSED) | [10061] Connection refused. |

| | | |
|-------|----------------------|--|
| 25063 | (WSAELOOP) | [10062] Too many levels of symbolic links. |
| 25064 | (WSAENAMETOOLONG) | [10063] File name too long. |
| 25065 | (WSAEHOSTDOWN) | [10064] Host is down. |
| 25066 | (WSAEHOSTUNREACH) | [10065] No Route to Host. |
| 25067 | (WSAENOTEMPTY) | [10066] Directory not empty. |
| 25068 | (WSAEPROCLIM) | [10067] Too many processes. |
| 25069 | (WSAEUSERS) | [10068] Too many users. |
| 25070 | (WSAEDQUOT) | [10069] Disc Quota Exceeded. |
| 25071 | (WSAESTALE) | [10070] Stale NFS file handle. |
| 25072 | (WSAEREMOTE) | [10071] Too many levels of remote in path. |
| 25092 | (WSASYSNOTREADY) | [10091] Network SubSystem is unavailable. |
| 25093 | (WSAVERNOTSUPPORTED) | [10092] WINSOCK DLL Version out of range. |
| 25094 | (WSANOTINITIALISED) | [10093] Successful WSASTARTUP not yet performed. |
| 25102 | (WSAHOST_NOT_FOUND) | [11001] Host not found. |
| 25103 | (WSATRY_AGAIN) | [11002] Non-Authoritative Host not found (try again). |
| 25104 | (WSANO_RECOVERY) | [11003] Non-Recoverable error. |
| 25105 | (WSANO_DATA) | [11004] Valid name, no data record for requested name. |

