



## IPPort Control (TM) Version 1.11

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.

This minor release includes internal enhancements as well a new functionality.

### 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 path 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, DataIn, Disconnected, ReadyToSend. 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.

To disconnect, you just set the Connected property to **False**. The Linger controls how the connection is terminated.

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.

We also strongly recommend that you visit our WWW site at **<http://www.dev-soft.com/devsoft>**. There you will find the latest versions of our shareware products as well as other helpful information.

AcceptDataPREF\_AcceptData  
ActivePREF\_Active  
BytesSentPREF\_BytesSent  
ConnectedPREF\_Connected  
DataInPREF\_DataIn  
DataToSendPREF\_DataToSend  
EOLPREF\_EOL  
HostPREF\_Host  
HostAddressPREF\_HostAddress  
HostNamePREF\_HostName  
InBufferSizePREF\_InBufferSize  
LingerPREF\_Linger  
ListeningPREF\_Listening  
LocalHostPREF\_LocalHost  
LocalHostNamePREF\_LocalHostName  
LocalPortPREF\_LocalPort  
NullsToSendPREF\_NullsToSend  
OutBufferSizePREF\_OutBufferSize  
PortPREF\_Port  
RemoteHostPREF\_RemoteHost  
RemotePortPREF\_RemotePort  
WinsockInfoPREF\_WinsockInfo  
ActionPREF\_Action  
EncodedDataPREF\_EncodedData  
DecodedDataPREF\_DecodedData  
FileNamePREF\_FileName  
FileCntPREF\_FileCnt  
FileCntPREF\_FileCnt  
FormatPREF\_Format  
IntellicodePREF\_Intellicode  
MaxFileSizePREF\_MaxFileSize  
OverwritePREF\_Overwrite  
ProgressStepPREF\_ProgressStep

DecodingUREF\_ENCODING  
EncodingUREF\_ENCODING  
UUDecodingUREF\_UU\_ENCODING  
UUEncodingUREF\_UU\_ENCODING  
Base64 DecodingUREF\_BASE64\_ENCODING  
Base64 EncodingUREF\_BASE64\_ENCODING  
Quoted Printable DecodingUREF\_QP\_ENCODING  
Quoted Printable EncodingUREF\_QP\_ENCODING

ConnectedEREF\_Connected  
ConnectionRequestEREF\_ConnectionRequest  
DataInEREF\_DataIn  
DisconnectedEREF\_Disconnected  
ReadyToSendEREF\_ReadyToSend  
ProgressEREF\_Progress

EncodeFREF\_Encode  
DecodeFREF\_Decode

**True**

**False**  
**Boolean (Integer)**  
**Integer**  
**Long**  
**String**  
"" (*empty string*)

".uue", ".b16", or ".q\_p"

Error CodesERROR\_CODES  
Exported FunctionsEXPORTED\_FUNCTIONS

## **IPPort**

IPPORT.VBX

*ipportcontrol*

1.1

Copyright (C) 1995 **devSoft Inc.** - All Rights Reserved.

**\$25**

**#11682**

**3961**

## Copyright Notice

The **IPPort** Custom Control (TM) is Copyright (C) 1995 **devSoft Inc.** - All Rights Reserved.

## Registration Procedure

The prices below are for the **licenses only** and do not include media distribution. We only send you a set of keys to unlock the software and verify registration by e-mail. All technical support questions should be directed to:

INTERNET: **devsoft@aol.com**  
COMPUERVE: **75244,2736**

The cost of a single user developer is **\$25**. You can order via any one of the following channels:

### ***i) ordering through CompuServe Software Registration Service (SWREG)***

You can register via CompuServe by going to the Shareware Registration Forum (**GO SWREG**) and following the forum instructions. The Registration ID for **IPPort** is **3961**. You can also do a keyword search using the keyword **IPPort**.

### ***ii) ordering by Check or Money Order***

To order by check or money order please send the attached order form and a check or a money order (**payments must be in US Dollars drawn on a US Bank**) to:

**devSoft Inc.**  
P.O. Box 13821  
Research Triangle Park, NC 27709 U.S.A.

### ***iii) ordering by Credit Card***

To order by Visa or MasterCard by E-mail, fax or snail mail send the attached order form to the above address or:

INTERNET: **devsoft@aol.com**  
COMPUERVE: **75244,2736**  
FAX: **(919) 493-5805**

## Where To Find Our Shareware

The first place to look at is **<http://www.dev-soft.com/devsoft>** . There you will find the latest versions of our products, release notes, questions and answers, documentation, press releases, everything you would want to know about us and our products. We strongly recommend that you access that site before contacting us directly.

We will also upload our products in the CompuServe MSBASIC Forum (GO MSBASIC) in Library 17 (3rd Party Tools), as well as in America Online. Usually, the name of the product will be listed as a keyword, so if you try it, you will certainly get a hit.

We will also announce our new releases to the newsgroups of the comp.lang.basic.visual hierarchy, and **comp.lang.basic.visual.3rdparty** in particular.

## Licensing

### ***i) shareware version***

You may use the shareware version of **IPPort** for up to 30 days in your design environment and 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 it's

accompanying files with other products (commercial or otherwise) without prior written permission from **devSoft Inc.**

***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 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 the license file are not transferable: users of your application cannot legally use the license for their own applications, or distribute their own code using the a license file 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.

---

---

**devSoft Inc.**

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

---

---

**ORDER FORM / INVOICE**

*Prices are guaranteed through December 1995.*

*Registration codes will be sent by electronic mail only. If you need a disk (3.5") or a paper copy of your license, please enclose an additional \$5.00 with your order.*

Product	No. of Copies	Price	Total
IPPort	_____ x	\$25.00 =	_____
IPDaemon	_____ x	\$25.00 =	_____
UDPPort	_____ x	\$25.00 =	_____
NetCode	_____ x	\$30.00 =	_____
<b>Add Disk and/or Paper copy of License</b>		\$5.00 =	_____
		<b>Total</b>	_____

---

Name:

Phone:

Date:

Credit card used /Exp. Date

Company Name:

Address:

City, State, Zip/Country:

Name of registrant:

E-Mail address:

Comments:

---

---

## Properties

<u>*AcceptData</u>	Left
<u>*BytesSent</u>	<u>*Linger</u>
<u>*Connected</u>	<u>*LocalHostName</u>
<u>*DataToSend</u>	<u>*LocalPort</u>
<u>*EOL</u>	Name
<u>*HostAddress</u>	<u>*OutBufferSize</u>
<u>*HostName</u>	<u>*Port</u>
<u>*InBufferSize</u>	Top
Index	<u>*WinsockInfo</u>

## AcceptData Property

### Description

Setting the property to **False** temporarily disables data reception (and the DataIn event) from the specified *ConnectionID*. Setting the property to **True** reenables data reception.

### Usage

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

### Default Value

**True**

### Remarks

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

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 assignement 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

This property is used to create and close connections, as well as show the current status of the control (connected or not).

### Usage

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

### Default Value

**False**

### Remarks

**Connected** is an action property. Use it to create and close connections. Setting **Connected** to **True** initiates a connection attempt. If successful, after the connection is achieved, the value of the property changes to **True** and the Connected event is fired.

How and when the connection is closed is controlled by the Linger property. Please refer to its description for more information.

The **Connected** 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(ConnectionID)[ = 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 write-only and 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 description of 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 Visual Basic String. In particular, this means that it can be more than one character long, and it can contain NULL (0) characters as well.

The **EOL** property is shared among incoming connections.

### Data Type

**String**

## HostAddress Property

### Description

Specifies the IP number of the remote host in Internet dotted format.

### Usage

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

### Default Value

0.0.0.0

### Remarks

The **HostAddress** property is an action property, but in this case the *action* is deferred until the HostName property is queried (read). When this happens, a DNS operation is initiated, trying to find the corresponding domain name for the address. After the DNS call finishes, the host name can be found in the HostName property.

**HostAddress** cannot be changed once connected. You need to close the current connection before attempting to set a new address, or an error will be fired.

### Data Type

**String**

## HostName Property

### Description

Specifies the domain name of the remote host.

### Usage

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

### Default Value

"" (*empty string*)

### Remarks

The **HostName** property is an action property. When set, a DNS operation is initiated, trying to find the corresponding IP number for the domain name (**HostName**). After the DNS call finishes, the IP number can be found in the HostAddress property.

**HostName** cannot be changed once connected. You need to close the current connection before attempting to set a new address, or an error will be fired.

### 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 a new connection is accepted, **InBufferSize** reverts back to its allowable size. The same happens if you attempt to make it too large or too small.

### Data Type

**Integer**

## Linger Property

### Description

This property controls how a connection is closed. The default is **True**. In this case the connection is closed only after all the data is sent. Setting it to **False** forces an abrupt (hard) disconnection. Any data that were in the sending queue might be lost.

### Usage

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

### Default Value

**True**

### Remarks

The **Linger** property is shared among connections. It's value controls how the next connection will be closed.

The default behaviour (which is also the default mode for Winsock stream sockets) might result in an indefinite delay in closing the connection. Even though IPPort returns control immediately, Winsock might indefinitely hold system resources until all pending data are sent (even after your application closes). This means that valuable system resources might be wasted.

Setting **Linger** to **False** forces an immediate disconnection. If you know that the other side has received all the data you had sent (by a client acknowledgment, for example), setting **Linger** to **False** might be the appropriate course of action.

### Data Type

**Boolean (Integer)**



## LocalHostName Property

### Description

Specifies the domain name of the local host.

### Usage

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

### Default Value

"" (*empty string*)

### Remarks

This property is read-only.

### Data Type

**String**

## LocalPort Property

### Description

Specifies the IP port of the local host.

### Usage

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

### Default Value

0

### Remarks

The **LocalPort** property must be set before a connection is attempted. It instructs the control to *bind* to a specific port (or communication endpoint) in the local machine.

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 a new connection is accepted, **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.]RemotePort[ = value]`

### Default Value

0

### Remarks

The **Port** property must be set before a connection is attempted. The port number 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 subsystem.

If Winsock fails to initialize successfully, **WinsockInfo** contains the string "Not Initialized." followed by a description of the error condition.

The property is read-only.

### Data Type

**String**

## Events

\*Connected

\*DataIn

\*Disconnected

\*ReadyToSend

## Connected Event

### Description

Occurs when the connection to the remote host is created.

### Syntax

**Sub** *ipportcontrol\_Connected*(*StatusCode* **As Integer**, *Description* **As String**)

### Remarks

If a connection is successfully created, *StatusCode* is **0**, and *Description* is "**OK**".

If the connection fails, *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\_ConnectionRequest*(*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 the 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 accept data and send them to the remote host.

### 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 a connection to the remote host is created.

## 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.
- 20105 Already connected. Close the current connection first in order to reconnect.
- 20106 Winsock error code outside normal range.
- 20107 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	(WSAEPROTOPTYPE)	[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.

