NETMANAGE

# WinSock UDP ActiveX Control

The following Help Topics are available:

For Help on Help, Press F1

## Net*Manage*

NetManage develops, markets and supports an integrated set of TCP/IP inter-networking applications and development tools for Microsoft Windows. NetManage software facilitates communication, productivity and the administration of personal computers across dissimilar networking environments. The Company's award-winning product families include Chameleon and ECCO.

The company is located at 10725 North De Anza Blvd. Cupertino, CA 95014, USA

Phone: 408-973-7171 Fax: 408-973-8272.

International phone: +972-4-8550234 Fax +972-4-8550122

## General

The WinSock UDP ActiveX Control implements WinSock UDP (User Datagram Protocol) for both client and server. The control represents a communication point utilizing UDP network services. It can be used to send and retrieve UDP data.

Invisible to the user, the UDP Control, provides easy access to UDP network services. It can be used by both Delphi and C++ programmers. To write UDP applications you do not need to understand the details of UDP or to call low level WinSock APIs. By setting properties and calling methods on the control, you can easily send data to a remote machine or retrieve data from the network. Events are used to notify users of network activities.

## Properties

Properties supported by the UDP Control are listed here.

Note: Some common ActiveX properties of the control, such as Name, Index, About Box, and others, may appear in the Object Browser but are not documented here.

LocalHostName

LocalIP

LocalPort

RemoteHost

RemoteHostIP

RemotePort

SocketHandle

## Naming Conventions

Objects described in the Properties, Methods and Events section are preceded by the required parameter: object. During execution object translates to the name of the control. The actual object name will be:

`UDP`*`n`*

where *n* is the number identifier. For example, the first UDP in a form becomes UDP1, the second is UDP2 and so forth.

## LocalHostName

### Description

This property defines the local machine name.

### Syntax

*object.***LocalHostName**

The object placeholder is required and evaluates to the name of the relevant control or collection during execution.

### Permission

R (Read-only).

### Availability

R (Runtime).

### Data Type

String.

### Default Value

Empty.

### Range

N/A

## LocalIP

### Description

This property specifies the IP address of the local machine. It has the format:

```
    number.number.number.number
```

### Syntax

*object*.**LocalIP**

The object placeholder is required and evaluates to the name of the relevant control or collection during execution.

### Permission

R (Read-only).

### Availability

R (Runtime).

### Data Type

String.

### Default Value

Empty.

### Range

N/A

## LocalPort

### Description

Designates the local port to use.

### Syntax

*object*.**LocalPort** [= Long]

The object placeholder is required and evaluates to the name of the relevant control or collection during execution.

### Permission

W (Read/Write).

### Availability

D (Design).

### Data Type

Long.

### Default Value

0.

### Range

0 - 65535

## RemoteHost

### Description

The remote machine to which to send UDP data.   You can enter either a host name or an IP address string in dotted format (for example, 156.10.5.298).

Note:This is the default property of the control.

### Syntax

*object*.**RemoteHost** [= *String*]

The object placeholder is required and evaluates to the name of the relevant control or collection during execution.

### Permission

W (Read/Write).

### Availability

D (Design).

### Data Type

String.

### Default Value

Empty.

### Range

N/A

## RemoteHostIP

### Description

After the GetData method, this property contains the IP address of the remote machine sending the UDP data.

### Syntax

*object*.**RemoteHostIP**

The object placeholder is required and evaluates to the name of the relevant control or collection during execution.

### Permission

R (Read-only).

### Availability

R (Runtime).

### Data Type

String.

### Default Value

Empty.

### Range

N/A

## RemotePort

### Description

This property specifies the remote port number on the remote machine to which UDP data is sent. After the GetData method, this property contains the remote port that is sending the UDP data.

### Syntax

*object*.**RemotePort** [= *Long*]

The object placeholder is required and evaluates to the name of the relevant control or collection during execution.

### Permission

W (Read/Write).

### Availability

D (Design).

### Data Type

Long.

### Default Value

0.

### Range

0 - 65535

## SocketHandle

**Description**

This is the socket handle the control uses to communicate with the WinSock layer.

**Syntax**

*object*.**SocketHandle**

The object placeholder is required and evaluates to the name of the relevant control or collection during execution.

**Permission**

R (Read only).

**Availability**

R (Runtime).

**Data Type**

Long.

**Default Value**

-1.

**Range**

-1 - 65535

**Comments**

This property is for advanced programmers.   You can use SocketHandle in direct WinSock API calls. However, you should be aware that if WinSock calls are used directly, certain events may not be activated appropriately.

**Property Page**

One property page is provided for viewing and editing the following properties:

- RemoteHost
- RemotePort
- LocalPort

## Methods

The methods performed by the UDP Control are:

GetData

SendData

## GetData

### Description

Retrieves data.

### Return Value

Void.

### Syntax

*object*.**GetData** *data,* [*type*]

The object placeholder is required and evaluates to the name of the relevant control or collection during execution.

### Parameters

*Data*

Stores retrieved data after the method returns successfully. If there is no data available, *data* will be set to Empty.
Data Type: VARIANT
Param: OUT

*Type*

Type of data to be retrieved. It can be either vbString or byte array.
Data Type: VARIANT
Param: IN
Default Value: vbArray + vbByte

### Comments

If the type is specified as vbString, string data is converted to UNICODE before returning to the user.

## SendData

### Description

This method sends data to remote machine.

### Return Value

Void.

### Syntax

*object*.**SendData** *data*

The object placeholder is required and evaluates to the name of the relevant control or collection during execution.

### Parameters

*Data*

Data to be sent. For binary data, byte array should be used.
Data Type: VARIANT
Param: IN

Currently, the following variant types are supported.

| Type | C++ | VB Type |
|---|---|---|
| Byte | VT_UI1 | vbByte |
| Integer | VT_I2 | vbInteger |
| Long | VT_I4 | vbLong |
| Single | VT_R4 | vbSingle |
| Double | VT_R8 | vbDouble |
| Currency | VT_CY | vbCurrency |
| Date | VT_DATE | vbDate |
| Boolean | VT_BOOL | vbBoolean |
| SCODE | VT_ERROR | vbError |
| String | VT_String | vbString |
| Byte Array | VT_ARRAY \| VT_UI1 | vbArray + vbByte |

### Comments

The RemoteHost and RemotePort properties should be set before calling this method.

When a UNICODE string is passed in, it is converted to an ANSI string before being sent out on the network.

## Events

Events are used for UDP client notification.   They indicate that an action has been requested and processed.

DataArrival

Error

## Error

### Description

The event is activated whenever an error occurs in background processing (for example, failed to connect, or failed to send or receive in the background).

### Syntax

*object*_**Error** (*ErrCode* **As Integer**, *Description* **As String**, *Scode* **As Long**, *Source* **As String**, *HelpFile* **As String**, *HelpContext* **As Long**, *CancelDisplay* **As Boolean**)

The object placeholder is required and evaluates to the name of the relevant control or collection during execution.

### Parameters

*ErrCode*

An integer that defines the error code.   For a list of possible error codes see WinSock Error Codes.

*Description*

String containing error information.

*Scode*

The long SCODE.

*Source*

String describing the error source.

*HelpFile*

String containing help file name.

*HelpContext*

Help file context.

*CancelDisplay*

Indicates whether to cancel the display.   The default is TRUE (no display of the default error message box ).   If you do want to use the default message box, set CancelDisplay to FALSE.

## DataArrival

### Description

The event is activated when a new UDP packet arrives.

### Syntax

*object*_**DataArrival** (*BytesTotal* **As Long**)

The object placeholder is required and evaluates to the name of the relevant control or collection during execution.

### Parameters

*BytesTotal*

The total amount of data, in bytes, currently available.
Data Type: Long
Param: IN

## Localization

The resources for the control's about box, property page, and strings are in resource DLL nmorenu.dll. The resource DLL is localized for each language.

## WinSock Error Codes

The following error codes apply to the WinSock ActiveX Controls.

| Error Number | Error Message |
|---|---|
| 10004 | The operation is canceled |
| 10013 | The requested address is a broadcast address, but flag is not set |
| 10014 | Invalid argument |
| 10022 | Socket not bound, invalid address or listen is not invoked prior to accept |
| 10024 | No more file descriptors are available, accept queue is empty |
| 10035 | Socket is non-blocking and the specified operation will block |
| 10036 | A blocking Winsock operation is in progress |
| 10037 | The operation is completed. No blocking operation is in progress. |
| 10038 | The descriptor is not a socket |
| 10039 | Destination address is required |
| 10040 | The datagram is too large to fit into the buffer and is truncated |
| 10041 | The specified port is the wrong type for this socket |
| 10042 | Option unknown, or unsupported |
| 10043 | The specified port is not supported |
| 10044 | Socket type not supported in this address family |
| 10045 | Socket is not a type that supports connection oriented service |
| 10047 | Address Family is not supported |
| 10048 | Address in use |
| 10049 | Address is not available from the local machine |
| 10050 | Network subsystem failed |
| 10051 | The network cannot be reached from this host at this time |
| 10052 | Connection has timed out when SO_KEEPALIVE is set |
| 10053 | Connection is aborted due to |

| | |
|---|---|
| | timeout or other failure |
| 10054 | The connection is reset by remote side |
| 10055 | No buffer space is available |
| 10056 | Socket is already connected |
| 10057 | Socket is not connected |
| 10058 | Socket has been shut down |
| 10060 | The attempt to connect timed out |
| 10061 | Connection is forcefully rejected |
| 10201 | Socket already created for this object |
| 10202 | Socket has not been created for this object |
| 11001 | Authoritative answer: Host not found |
| 11002 | Non-Authoritative answer: Host not found |
| 11003 | Non-recoverable errors |
| 11004 | Valid name, no data record of requested type |