NETMANAGE

POP Client ActiveX Control

The following Help Topics are available:

General

POP3 Commands

Naming Conventions

Properties

Methods

Events

Error Messages

For Help on Help, Press F1

NetManage

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 POP Client ActiveX Control implements the POP3 Protocol Client as specified by RFC 1081, *Post Office Protocol*. It provides access to Internet mail servers using the POP3 protocol. It can be used by Internet mail developers or system integrators. The major advantage of this control is its ability to retrieve mail from UNIX or other servers supporting POP3 protocol.

The main features of the POP Client Control are:

- connects to a server
- sends authentication information (user and password) to the server
- retrieves user mailbox information, such as the number of messages waiting to be retrieved
- retrieves messages from the server
- deletes messages from the server

There should be no speed overhead and response delay other than the one given by the network. This control uses and is dependent on the DocStream objects (DocInput and DocOutput).

POP3 Commands

The following table summarizes POP3 commands as described by the protocol specification in RFC 1081. All the commands are implemented in the control, although some of them are abstracted at a higher level (e.g. USER + PASS = Authorization).

Command	Usage	When Valid
DELE msg	required	TRANSACTION state
LAST	required	TRANSACTION state
LIST [msg]	required	TRANSACTION state
NOOP	required	TRANSACTION state
PASS string	required	AUTHORIZATION state
QUIT	required	AUTHORIZATION and UPDATE state
RETR msg	required	TRANSACTION state
RPOP user	optional	AUTHORIZATION state; not supported in current release
RSET	required	TRANSACTION state
STAT	required	TRANSACTION state
TOP msg n	optional	TRANSACTION state; supported if it is supported by the server.
USER name	required	AUTHORIZATION state

Each of the POP3 commands can return either:

+OK

-ERR

The reply given by the POP3 server to any command is significant only up to "+OK" and "-ERR". The client can ignore any text occurring after this reply. The only exception is the STAT command.

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.

When using a collection, object becomes the collection name. For example, icErrors collection objects are preceded by icErrors.

Properties

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.

The list of properties follows alphabetically.

Busy

DocOutput

EnableTimer

Errors

MessageCount

NotificationMode

Password

ProtocolState

ProtocolStateString

RemoteHost

RemotePort

ReplyCode

ReplyString

State

StateString

TimeOut

TopLines

TopSupported

UserId

URL

Busy

Description

Indicates a command is in progress.

Syntax

object.**Busy**

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

Boolean.

Default Value

N/A.

Range

N/A

DocOutput

Description

Object describing output information for the document being transferred.

Syntax

object.**DocOutput**

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

DocOutput.

Default Value

N/A.

Range

N/A

Comments

The DocOutput object provides a more powerful interface than the basic capabilities of the GetDoc method. However, you can use the basic functions of the control without knowledge or use of the DocInput object.

Properties of the DocOutput object may be set before calling the GetDoc method or they may be passed as arguments to this method. The DocOutput object is also used for conveying information about the progress of the document transfer, and for data linking and streaming.

For more information, see the DocOutput event and the Common Control Objects

EnableTimer

Description

Enable timer for the specified event. The event is specified by entering:

```
EnableTimer(short event)
```

Syntax

object.**EnableTimer** (*event*) [= *Boolean*]

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

Permission

W (Write Only).

Note:This is the only control property that is Write only.

Availability

R (Runtime)

Data Type

Boolean.

Default Value

False. (The timer for this event will not be enabled.)

Range

True or False

Comments

Event is an integer value that determines the type of Timeout event that will be enabled. Constants defined for enum types for events are:

Value	Meaning
prcConnectTimeout = 0	Timeout for connect. If connection is not established within the timeout period, the Timeout event will be activated.
prcReceiveTimeout = 1	Timeout for receiving data. If no data arrives within the timeout period, the Timeout event will be activated.
prcUserTimeout= 65	Timeout for user defined event. Use prcUserTimeout + [Integer] range for custom timeout events.

Errors

Description

A collection of errors that can be accessed for details about the last error that occurred. This collection should be used within an Error event if information passed through the Error event is not sufficient. For more details, see icErrors.

Syntax

object.**Errors**

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

icErrors.

Default Value

N/A.

Range

N/A

MessageCount

Description

This property specifies the number of messages in the mailbox. It is established after authentication has been successfully performed. Before that it is invalid.

Syntax

object.**MessageCount**

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

Integer.

Default Value

0.

Range

1-32767.

NotificationMode

Description

Determines when notification is issued for incoming data. Notification can also be suspended.

Syntax

object.**NotificationMode** [= *Integer*]

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

Permission

W (Read/Write).

Availability

R (Runtime) and D (Design).

Data Type

Integer.

Default Value

1.

Range

0-maximum unsigned long. At present , the values are:

Constant	Meaning
0	COMPLETE: notification is provided when there is a complete response.
1	CONTINUOUS: an event is repeatedly activated when new data arrives from the connection.

Password

Description

Password of current user on the server.

Syntax

object.**Password** [= *String*]

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

Permission

W (Read/Write).

Availability

R (Runtime) and D (Design).

Data Type

String.

Default Value

Empty.

Range

N/A

ProtocolState

Description

This property specifies the current state of the protocol.

Syntax

object.**ProtocolState**

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

Integer.

Default Value

prcBase.

Range

0-3. Constants defined for enumerated type Protocol State property:

Constant	Meaning
prcBase = 0	Base state before connection to server is established.
prcAuthorization = 1	Authorization is being performed.
prcTransaction = 2	Authorization had been performed successfully, the client has successfully identified itself to the POP3 server and the POP3 server has locked and burst the appropriate maildrop.
prcUpdate = 3	When Quit command is issued from transaction state.

ProtocolStateString

Description

String representation of ProtocolState.

Syntax

object.**ProtocolStateString** [= String]

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

BASE.

Range

N/A

ReplyCode

Description

The value of the reply code is a protocol specific number that determines the result of the last request, as returned in the ReplyString property.

Syntax

object.**ReplyCode**

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

0

Range

See RFC 1081 for a list of valid reply codes.

ReplyString

Description

Line returned to the client as a result of a request.

Syntax

object.**ReplyString**

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.

RemoteHost

Description

The remote machine to connect to if the remoteHost parameter in the Connect method is missing. You can either provide a host name or an IP address string in dotted format. For example, 127.0.0.1.

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.

RemotePort

Description

The remote port number to which to connect.

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

R (Runtime) and D (Design).

Data Type

Long.

Default Value

110.

Range

1-65535.

State

Description

This property specifies the connection state of the control.

Syntax

object.**State**

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

Integer.

Default Value

prcDisconnected.

Range

1-6. Constants defined for enum types of State property are:

Value	Meaning
prcConnecting = 1	Connecting. Connect has been requested, waiting for connect acknowledge.
prcResolvingHost = 2	Resolving Host. Occurs when RemoteHost is in name format rather than dot-delimited IP format.
prcHostResolved = 3	Resolved the host. Occurs only if ResolvingHost state has been entered previously.
prcConnected = 4	Connection established.
prcDisconnecting = 5	Connection closed. Disconnect has been initiated.
prcDisconnected = 6	Initial state when protocol object is instantiated, before Connect has been initiated, after a Connect attempt failed or after Disconnect performed.

StateString

Description

A string representation of State.

Syntax

object.**StateString** [= *String*]

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

"Disconnected".

Range

N/A

Timeout

Description

Timeout value for the specified event. The event is specified by entering:

```
Timeout(short event)
```

Syntax

object.**Timeout** (*event*) [= *Long*]

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

Permission

W (Read/Write).

Availability

R (Runtime) and D (Design).

Data Type

Long.

Default Value

0.

Range

0-maximum unsigned long. Constants defined for enum types for events are:

Value	Meaning
prcConnectTimeout = 0	Timeout for connect. If connection is not established within the timeout period, the Timeout event will be activated.
prcReceiveTimeout = 1	Timeout for receiving data. If no data arrives within the timeout period, the Timeout event will be activated.
prcUserTimeout= 65	Timeout for user defined event. Use prcUserTimeout + [Integer] range for custom timeout events.

TopLines

Description

Designates the number of lines to be retrieved in a top request.

Syntax

object.**TopLines** [= *Long*]

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

Permission

W (Read/Write).

Availability

R (Runtime) and D (Design).

Data Type

Long.

Default Value

0.

Range

0-2147483647.

TopSupported

See Also

Description

This property indicates "Top is supported". It can be queried after a connection to the server has been established. It is set to TRUE if the particular server supports the TOP command.

Syntax

object.**TopSupported**

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

Boolean.

Default Value

None.

Range

True, False.

URL

Description

URL string identifying the current document being transferred. The URL format for this control is:

```
POP://user:password@host:port /message number
```

Syntax

object.**URL** [= S*tring*]

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

Permission

W (Read/Write).

Availability

R (Runtime).

Data Type

String.

Default Value

Empty string.

Range

Valid URL.

Comments

URL may be set before calling the GetDoc method of the control, or it may be passed as an argument to these methods. If it is passed as an argument, the URL property will be set to the argument value.

In the POP Control, the URL property may identify a message being retrieved from a remote server. The URL type (first part up to the colon) may be omitted. In this case, it will default to the correct type for this control. For example, the pop string may be omitted when using the POP Control.

UserId

Description

User identification name for the client on the server.

Syntax

object.**UserId** [= *String*]

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

Permission

W (Read/Write).

Availability

R (Runtime) and D (Design).

Data Type

String.

Default Value

Empty.

Range

N/A

TopMessage Method

Methods

Methods are called to perform a particular operation on an object. After the method is successfully processed, you will receive an event with a name similar to the method called. You can then check the ReplyString value for the server response or check error codes if an error message is generated.

Authenticate

Cancel

Connect

Delete

GetDoc

Last

MessageSize

NOOP

Quit

RefreshMessageCount

Reset

RetrieveMessage

TopMessage

Authenticate

Description

Authenticates the user based on the parameters passed. If no parameters are passed, the UserId and Password properties are used. If neither the UserId nor the Password are entered, the control uses the URL. When authentication process is terminated, the Authenticate event is activated.

Return Value

Void.

Syntax

*object.***Authenticate** [*UserID,*] [*Password*]

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

Parameters

UserId

Optional. User identification string to use for authentication.
Data Type: String
Param: IN
Default Value: N/A

Password

Optional. Password to use for authentication.
Data Type: String
Param: IN
Default Value: N/A

Comments

If the UserId and/or Password are set before invoking this method, the optional parameters do not need to be specified. Optional arguments to this method override the values from corresponding UserId and Password properties. The values of the properties will not change. If you omit one or both of the arguments, the value from a corresponding property will be used to provide authentication.

Cancel

Description

Cancels a pending request.

Return Value

Void.

Syntax

object.**Cancel**

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

Parameters

None.

Connect

Description

Initiates a Connect request. The control calls the StateChanged event if a connection is established.

Return Value

Void.

Syntax

object.**Connect** [*RemoteHost*,] [*RemotePort*]

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

Parameters

RemoteHost

Optional. Remote host to which to connect. If this parameter is missing, the control connects to the host defined in the RemoteHost property.
Data Type: String
Param: IN
Default Value: N/A

RemotePort

Optional. Remote port to which to connect. If this parameter is missing, the control connects to the port defined in the RemotePort property.
Data Type: Long
Param: IN
Default Value: N/A

Comments

Optional arguments to this method override the values from corresponding RemoteHost and RemotePort properties. The values of the properties will not change. If no argument is given, the values from the properties will be used to establish the connection.

Delete

Description

Initiates a Delete request. If successful, a Delete event is activated, otherwise an Error event is activated.

Return Value

Void.

Syntax

object.**Delete** *MsgNumber*

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

Parameters

MsgNumber

Number of message to be deleted.
Data Type: Integer.
Param: IN
Default Value: None.

GetDoc

Description

A DocOutput related method that requests retrieval of a document identified by a URL.

Return Value

Void.

Syntax

object.**GetDoc** [*URL*,] [*Headers*,] [*OutputFile*]

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

Parameters

URL

Optional. The URL identifying the remote document to be retrieved.
Data Type: String
Param: IN
Default Value: DocInput.URL

Headers

Optional. Headers used for requesting the document. This argument only applies to protocols where request headers can be specified (for example, HTTP).
Data Type: DocHeaders
Param: IN
Default Value: DocInput.Headers

OutputFile

Optional. A local file to which the retrieved document will be written.
Data Type: String
Param: IN
Default Value: DocOutput.Filename

Comments

The GetDoc method in POP gets a message from the server.

The URL and (for some controls) Headers are used as inputs specifying which document is to be retrieved. The OutputFile argument indicates where the retrieved document should be written locally.

The URL type (first part up to the colon) may be omitted and will default to the correct type for this control. For example, when using the POP Control, the "pop:" string may be omitted.

For basic use of this control, arguments should be passed to GetDoc to describe the document transfer. For more powerful use of this control, the DocInput and DocOutput objects can be used in conjunction with the DocInput and DocOutput events. The arguments of GetDoc correspond to properties in the DocInput and DocOutput objects of this control. DocInput and DocOutput properties can be set before calling GetDoc to avoid passing arguments. The DocInput and DocOutput events can also be used for transferring data using streaming rather than local files.

For more information see the DocOutput property, the DocOutput event and the DocOutput object section in the Common Control Objects.

Last

Description

Initiates a LAST request. If successful, the LAST event is activated. This request is used to find the highest message number accessed by the client.

Return Value

Void.

Syntax

object.**Last**

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

Parameters

None.

MessageSize

Description

Initiates a request to retrieve the message size. If successful, a MessageSize event is activated, otherwise the Error event is activated.

Return Value

Void.

Syntax

object.**MessageSize** *MsgNumber*

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

Parameters

MsgNumber

Message number.
Data Type: Integer.
Param: IN
Default Value: None.

NOOP

Description

Initiates a NOOP request. This is used to test the connection.

Return Value

Void.

Syntax

object.**NOOP**

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

Parameters

None.

Quit

Description

Initiates a Quit request. If unsuccessful, the Error event is activated.

Return Value

Void.

Syntax

object.**Quit**

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

Parameters

None.

RefreshMessageCount

Description

This method will refresh the number of undeleted messages from your current maildrop. When the request is completed, the RefreshMessageCount event is activated, indicating the current number of undeleted messages. This method is only available if you are already connected and authenticated. You will not be notify you of new messages received by the POP server; the context is the current maildrop. The message numbers themselves will not be renumbered. If, for example, you delete message 4 out of 6, a message retrieval of 5 will get message 5 and not message 6; message 6 is not renumbered to 5. Using Reset, you can undo message deletion at anytime prior to a Quit command. If, for example, you delete messages 3 and 5 out of 6, and then call RefreshMessageCount, the new number of undeleted messages is 4. If you do a Reset and then call RefreshMessageCount, the number of messages is 6.

Return Value

Void.

Syntax

object.**RefreshMessageCount**

Parameters

None.

Reset

Description

Initiate a RSET request. Any messages marked as deleted will be unmarked. If successful, a corresponding Reset event is activated, otherwise an Error event is activated.

Return Value

Void.

Syntax

object.**Reset**

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

Parameters

None.

RetrieveMessage

Description

Initiates a RetreiveMessage request for the message specified in msgNumber.

Return Value

Void.

Syntax

object.**RetrieveMessage** *msgNumber*

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

Parameters

msgNumber

specifies number of message to be retrieved.
Data Type: Integer.
Param: IN
Default Value: None.

Comments

DocOutput event can be used to retrieve the data.

TopMessage

Description

Initiates a Top of Message request for the message specified in msgNumber.

Syntax

object.**TopMessage** *msgNumber*

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

Parameters

msgNumber

specifies number of message to be retrieved.
Data Type: Integer.
Param: IN
Default Value: None.

Comments

DocOutput event can be used to retrieve the data.

TopMessage is used in conjunction with the TopLines property. If TopLines is 0, then only header information will be retrieved.

Events

Events are used for POP client notification. They indicate that an action has been requested and processed. Any errors which occur during command processing result in the Error event being called with appropriate error codes. Error codes, state changes, and protocol return values are usually checked during event processing.

Busy

Cancel

Delete

DocOutput

Error

Last

MessageSize

NOOP

ProtocolStateChanged

RefreshMessageCount

Reset

StateChanged

TimeOut

Busy

Description

This event is activated when a command is in progress or when a command has completed.

Syntax

*object*_**Busy** (*Busy* **As Boolean**)

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

Parameters

Busy

Indicates whether or not a command is in progress.
Data Type: Boolean. If the argument is True, a command is in progress.

Cancel

Description

This event is activated after a cancellation request has been completed and satisfied. After this event the object's state changes to idle.

Syntax

*object*_**Cancel**

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

Parameters

None.

Delete

Description

This event is activated after the successful completion of a Delete request.

Syntax

*object*_**Delete**

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

Parameters

None.

DocOutput

Description

A DocOutput related event indicating that output data has been transferred.

Syntax

*object*_**DocOutput** (*DocOutput* **As DocOutput**)

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

Parameters

DocOutput

Object describing document output data for the current transfer.
Data Type: DocOutput
Param: IN
Default Value: N/A

Comments

The DocOutput event can be used in its basic form to notify the user of transfer progress, (for example, for updating a progress bar). The DocOutput.BytesTotal, DocOutput.BytesTransferred and DocOutput.State properties can be examined to determine the current status of the transfer. This event can be ignored if no progress information is needed.

To gain more power from this control, you can also use the DocOutput event for data streaming. For more information, see the DocInput object section of the Common Control Objects.

Error

Description

This event is activated when 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

The short error code. For a list of possible error codes see POP Error Codes.

Description

String containing error information.

sCode

The long Scode.

Source

Error source.

HelpFile

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.

Last

Description

This event is activated after the successful completion of a Last request. It indicates the number of the last message accessed by the client.

Syntax

*object*_**Last** *(Number* **As Long***)*

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

Parameters

Number

Number of the last message accessed by the client.
Data Type: Long.

MessageSize

Description

This event is activated after successful completion of a MessageSize request.

Syntax

*object*_**MessageSize** *(MsgSize* **As Long***)*

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

Parameters

MsgSize

Size of the message requested.
Data Type: Long.

NOOP

Description

This event is activated after the successful completion of a NOOP request.

Syntax

*object*_**NOOP**

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

Parameters

None.

ProtocolStateChanged

Description

This event is activated whenever the protocol state changes.

Syntax

*object*_**ProtocolStateChanged** (*State* **As Integer**)

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

Parameters

Refer to the ProtocolState property and ProtocolStateString for possible values of the state parameter.

RefreshMessageCount

Description

This event is activated after a successful completion of RefreshMessageCount request. The number of undeleted messages from the current maildrop is returned. (A maildrop contains the messages that can be retrieved/deleted in the current state.)

Syntax

*object*_**RefreshMessageCount**

Parameters

None.

Reset

Description

This event is activated after the successful completion of a Reset request.

Syntax

*object*_**Reset**

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

Parameters

None.

StateChanged

Description

This event is activated whenever the state of the transport state changes.

Syntax

*object*_**StateChanged** (*State* **As Integer)**

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

Parameters

Refer to the State property and StateString for possible values of the state parameter.

TimeOut

See Also

Description

This event is activated when the timer for the specified event expires. See Timeout property for pre-defined events.

Syntax

*object*_**TimeOut** (ByVal *Event* **As Integer,** *Continue* **As Boolean**)

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

Parameters

Event

Defines the event to which the time interval applies.
Data Type: Integer

Continue

Determines if the timer is active or not. Set *Continue* to TRUE to keep the timer active.
Data Type: Boolean
Default Value: False

Comments

Event is an integer value that determines the type of Timeout event that will be enabled. Constants defined for enum types for events are:

Value	Meaning
prcConnectTimeout = 0	Timeout for connect. If connection is not established within the timeout period, the Timeout event will be activated.
prcReceiveTimeout = 1	Timeout for receiving data. If no data arrives within the timeout period, the Timeout event will be activated.
prcUserTimeout= 65	Timeout for user defined event. Use prcUserTimeout + [Integer] range for custom timeout events.

TimeoutProperty

ActiveX

ActiveX is a trademark of Microsoft Corporation.

POP Error Codes

The following error codes apply only to the POP ActiveX Control.

Error Number	Error Message
2450	RetrieveMessage Command Failed. Unable to retrieve message.
2451	Delete Command Failed. Unable to delete message.
2452	Reset Command Failed. Unable to unmark deleted message(s).
2453	Last Command Failed. Unable to find the highest message number accessed by client.
2454	RefreshMessageCount Command Failed. Unable to ascertain the number of messages marked as deleted.
2455	Noop Command Failed. Unable to test the connection.
2456	Quit Command Failed. Error while quitting.
2457	TopMessage Command Failed. Unable to retrieve the TopLines of the message.