



SMTP Client ActiveX Control

The following Help Topics are available:

[General](#)

[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 SMTP Client Control implements the basic client SMTP Protocol as specified by RFC821, *Simple Mail Transfer Protocol*. It is used to send Internet mail messages to SMTP servers.

The SMTP Control can be used for developing applications that communicate with SMTP servers to send mail messages. It provides a reusable component that gives applications access to SMTP mail servers and mail posting capabilities.

The SMTP Client Control supports a high level interface, that incorporates all SMTP commands used in sending out a mail message. Using this interface, a mail message can be sent with a single call.

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 object (DocInput).

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

Properties set the attributes for SMTP behavior.

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.

Following is a description of all properties supported by the SMTP Client Control.

[Busy](#)

[DocInput](#)

[EnableTimer](#)

[Errors](#)

[NotificationMode](#)

[ProtocolState](#)

[ProtocolStateString](#)

[RemoteHost](#)

[RemotePort](#)

[ReplyCode](#)

[ReplyString](#)

[State](#)

[StateString](#)

[TimeOut](#)

[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

DocInput

Description

Object describing input information for the document being transferred.

Syntax

object.DocInput

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

DocInput.

Default Value

N/A.

Range

N/A

Comments

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

Properties of the DocInput object may be set before calling the SendDoc method or they may be passed as arguments to this method. The DocInput 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 [DocInput](#) 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

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

0.

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.

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

0-1. Constants defined for the enum types of ProtocolState property are:

Value	Meaning
prcBase = 0	Base state before connection to server is established.
prcTransaction = 1	Connection to server is established. This is the valid state for calling methods on the control.

ProtocolStateString

Description

String representation of ProtocolState.

Syntax

object.**ProtocolStateString**

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.

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) and R (Runtime).

Data Type

String.

Default Value

Mail.

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

25.

Range

1-65535.

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 821 for a list of valid reply codes.

ReplyString

Description

Lists the last reply string sent by the SMTP Server 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.

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

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).

Data Type

Long.

Default Value

0.

Range

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

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

URL

Description

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

```
SMTP://host:port/
```

Syntax

`object.URL [= 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) and R (Runtime).

Data Type

String.

Default Value

Empty string.

Range

Valid URL.

Comments

URL may be set before calling the GetDoc or SendDoc 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.

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 smtp: string may be omitted when using the SMTP Control.

Methods

The methods performed by the SMTP Client Control are:

[Cancel](#)

[Connect](#)

[Expand](#)

[Help](#)

[Noop](#)

[Quit](#)

[Reset](#)

[SendDoc](#)

[Verify](#)

Cancel

Description

Initiates a Cancel request to cancel a pending request. If successful, the Cancel event is called. In case of an error, the Error event is called.

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. If successful, the [StateChanged](#) is called if connection is established. In case of an error, the Error event is called.

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 connect to.

Data Type: String

Param: IN

Default Value: N/A

RemotePort

Optional. Remote port to connect to.

Data Type: Long

Param: IN

Default Value: N/A

Comments

Optional arguments to this method will override the values from the corresponding properties RemoteHost and RemotePort. The values of the properties will change. If no argument is given, the values from the properties will be used to establish the connection.

Expand

Description

Initiates a EXPN request. If successful, Verify event will be called when the request completes. [ReplyString](#) will contain the reply from the server. In case of an error, the Error event is called.

Return Value

Void.

Syntax

object.Expand [= *name*]

Parameters

name:

email id or name that will be expanded

Data Type: String

Param: IN

Default Value

N/A

Help

Description

Initiates a HELP request. If successful, the Help event will be called when the request completes. [ReplyString](#) will contain the reply from the server. In case of an error, the Error event is called.

Return Value

Void.

Syntax

object.Help [topic]

Parameters

topic

Optional. Topic you want help on.

Data Type: String

Param: IN

Default Value: N/A

Noop

Description

Initiates a NOOP request. Noop event will be called. In case of error, the Error event will be called. Noop verify that the connection is alive.

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 to Quit the session and disconnect. In case of an error, the Error event is called.

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.

Reset

Description

Initiates a RSET request. If successful, the Reset event will be called. In case of an error, the Error event is called.

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.

SendDoc

See Also

Description

Initiates a Request to send a document identified by a URL. In case of an error, the Error event is called.

Return Value

Void.

Syntax

object.SendDoc [*URL*,] [*Headers*,] [*InputData*,] [*InputFile*,] [*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 sent.

Data Type: String

Param: IN

Default Value: DocInput.URL

Headers

Optional. Headers used for sending the document.

Data Type: DocHeaders

Param: IN

Default Value: DocInput.Headers

InputData

Optional. A data buffer containing the document to be sent.

Data Type: VARIANT

Param: IN

Default Value: DocInput.SetData

InputFile

Optional. A local file containing the document to be sent.

Data Type: String

Param: IN

Default Value: DocInput.Filename

OutputFile

Optional. A local file to which a reply document is written. This argument only applies for protocols that return a reply document (for example, HTTP).

Data Type: String

Param: IN

Default Value: DocOutput.Filename

Comments

The SendDoc method makes it possible to send a document. For the SMTP Control this means sending a mail message to the server.

The URL and Headers are used as inputs describing the document to be sent. The InputData and InputFile arguments (only one can be specified) contain the document to be sent.

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 SMTP Control, the "smtp:" string may be omitted .

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

Verify

Description

Initiates a VRFY request. If successful, the Verify event will be called when the request completes. ReplyString will contain the reply from the server. In case of an error, the Error event is called.

Return Value

Void.

Syntax

object.Verify [= *name*]

Parameters

name

email id or name that will be verified

Data Type: String

Param: IN

Default Value: N/A

[DocInput](#) properties

[DocInput](#) events,

[DocInput](#) Common objects

Events

Events are used for SMTP 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](#)

[DocInput](#)

[Error](#)

[Expand](#)

[Help](#)

[Noop](#)

[ProtocolStateChanged](#)

[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.

DocInput

See Also

Description

Indicates that input data has been transferred.

Syntax

object **DocInput** (ByVal *DocInput* As DocInput)

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

Parameters

DocInput

Object describing document input data for the current transfer.

Data Type: DocInput

Param: IN

Default Value: N/A

Comments

The DocInput event can be used in its basic form for notification of transfer progress, (for example, for updating a progress bar). The DocInput.BytesTotal, DocInput.BytesTransferred and DocInput.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 [DocInput](#) event for data streaming.

DocInput Common 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 SMTP error codes see [SMTP 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.

Expand

Description

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

Syntax

object_Expand

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

Parameters

None.

Help

Description

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

Syntax

object_Help

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

Parameters

None.

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 for possible values of the state parameter.

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 for possible values of the state parameter.

TimeOut

See Also

Description

This event is activated when the timer for the specified event expires.

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.

[Timeout](#) Property

ActiveX

ActiveX is a trademark of Microsoft Corporation.

SMTP Error Codes

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

Error Number	Error Message
2302	Can't create temporary mail file.
2303	Unable to send mail.

