



FTP Client ActiveX Control

The following Help Topics are available:

[General](#)

[Naming Convention](#)

[Properties](#)

[Property Pages](#)

[Methods](#)

[Events](#)

[FTPDirItem Object](#)

[Error Messages](#)

[Localization](#)

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 FTP (File Transfer Protocol) Client ActiveX Control enables file and data transfer between a remote and local machine.

Invisible to the user, the FTP Client Control provides easy access for Internet FTP services and can be used by both Delphi and C++ programmers. To write applications that use FTP you do not need to understand the details of FTP 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 and retrieve data from the network. Events are used to notify you of network activities.

The FTP Client Control also provides file and directory parsing of the List and NameList commands, making it possible for you to access the file size, attributes, name, or other fields without querying the server to determine the type of operating system (and hence file system) running on the remote server. For more details, see [FTPDirItem Objects](#)

Properties

Properties set the attributes for FTP Server behavior. They can be set at any time, although some of them may not have any affect until the client is connected to a server (for example, setting the AppendToFile property).

FTP Client properties include properties relating to the authentication process, data streaming, all client protocols and FTP specific.

[AppendToFile](#)

[Busy](#)

[DocInput](#)

[DocOutput](#)

[EnableTimer](#)

[Errors](#)

[ListItemNotify](#)

[NotificationMode](#)

[Operation](#)

[Password](#)

[ProtocolState](#)

[ProtocolStateString](#)

[RemoteDir](#)

[RemoteFile](#)

[RemoteHost](#)

[RemotePort](#)

[ReplyCode](#)

[ReplyString](#)

[State](#)

[StateString](#)

[TimeOut](#)

[URL](#)

[UserId](#)

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 relative control or collection name.

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

AppendToFile

Description

This property applies to PutFile and SendDoc to indicate whether the data should be appended to the file (True) or whether the file should be replaced (False).

Syntax

object.AppendToFile [= *Boolean*]

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

Permission

W (Read/Write).

Availability

D (Design).

Data Type

Boolean.

Default Value

False.

Range

True or False

Busy

Description

Indicates a command is in progress.

Syntax

object.**Busy**

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

Permission

R (Read only).

Availability

R (Runtime).

Data Type

Boolean.

Default Value

N/A.

Range

True or False

DocInput

See Also

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 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.

DocInput Event

Common Control Objects

DocOutput

See Also

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 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.

[DocOutput](#) Event

[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 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:

<u>Value</u>	<u>Meaning</u>
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 control or collection during execution.

Permission

R (Read only).

Availability

R (Runtime).

Data Type

icErrors.

Default Value

N/A.

Range

N/A

ListItemNotify bmc SeeAlso.bmp}

Description

Causes the container to receive events for every directory element received during a List or NameList command.

Syntax

object.ListItemNotify [= *Boolean*]

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

Permission

W (Read/Write).

Availability

D (Design).

Data Type

Boolean.

Default Value

False. (FTPDirItem objects are not parsed in the ListItem event.)

Range

True or False

Comments

If this property is TRUE, the directory listing is parsed and events activated for every directory element. If this property is FALSE, the list data is sent in blocks to the data target during ProcessData notifications.

ListItem

DocInput, DocOutput Properties

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 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:

Const	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.

Operation

See Also

Description

Allows you to determine the last method performed that caused data to be received. This property is normally used when processing the DocOutput event.

Syntax

object.Operation

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

Permission

R (Read only).

Availability

R (Runtime).

Data Type

FTPOperationConstants.

Default Value

ftpList.

Range

- ftpFile = 0
- ftpList = 1
- ftpNameList = 2

Comments

The GetFile, GetDoc, List, and NameList methods all transfer data via the DocOutput event. Using this property allows you to determine which method activated the DocOutput event, making it possible to distinguish between the various types of data.

[GetFile](#)

[GetDoc](#)

[List](#)

[NameList](#)

Password

Description

Password of current user on the FTP Server.

Syntax

object.Password [= *String*]

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

Permission

R (Read-only).

Availability

R (Runtime).

Data Type

Integer.

Default Value

prcBase.

Range

0-2. Constants defined for enum types of ProtocolState property are:

Value	Meaning
ftpBase = 0	Default. The state before connection server is established.
ftpAuthorization = 1	Authorization is performed.
ftpTransaction = 2	Authorization successful. The client has successfully identified itself to the FTP server.

ProtocolStateString

Description

String representation of ProtocolState.

Syntax

object.ProtocolStateString

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

Permission

R (Read-only).

Availability

R (Runtime).

Data Type

STRING.

Default Value

BASE.

Range

N/A.

RemoteDir

Description

The remote directory name. This read only property is set each time a `changeDir` or a `PrintDir` method is invoked.

Syntax

object.**RemoteDir**

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

Permission

R (Read only).

Availability

R (Runtime).

Data Type

String.

Default Value

"c:\\"

Range

N/A.

Comments

This property is not used to set the remote working directory. It provides the convenience of a parsed string containing the current working directory of the remote machine.

RemoteFile

Description

The remote file name used during GetFile and PutFile operations.

Syntax

object.RemoteFile [= *String*]

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

Permission

W (Read/Write).

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 control or collection during execution.

Permission

W (Read/Write).

Availability

R (Runtime).

Data Type

String.

Default Value

127.0.0.1

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 control or collection during execution.

Permission

W (Read/Write).

Availability

R (Runtime) and D (Design).

Data Type

Long.

Default Value

21.

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 control or collection during execution.

Permission

R (Read-only).

Availability

R (Runtime).

Data Type

Long.

Default Value

0

Range

See RFC 959 for Valid reply codes.

ReplyString

Description

Lists the last reply string sent by the FTP Server to the client as a result of a request. This string contains both a number code and a status string that the server creates for the last command.

Syntax

object.**ReplyString**

The object placeholder is required and evaluates to the name of the 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 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 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 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
<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 (Universal Resource Locator) string identifying the current document being transferred. The URL format when using the FTP Control is:

`FTP://username.password@host/documentnameand path`

Syntax

`object.URL [= String]`

The object placeholder is required and evaluates to the name of the 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](#) 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.

In the FTP control, the URL property identifies a remote file transferred via FTP. 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 `ftp` string may be omitted when using the FTP control.

Userld

Description

User identification name for the client on the server.

Syntax

object.Userld [= *String*]

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

Permission

W (Read/Write).

Availability

R (Runtime) and D (Design).

Data Type

STRING.

Default Value

Empty.

Range

N/A

Property Pages

Property pages provide a standard interface for setting or accessing the properties exposed by a control. The FTP properties that are listed on the property page are listed in the following table:

Property Pages	Properties Exposed
Common Property Pages	RemoteHost, RemotePort
Authenticate Property Pages	Password UserName
FTP Specific Property Pages	AppendToFile ListItemNotify RemoteFile

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 ReplyCode for the server response or check error codes if an error message is generated.

The methods supported by the FTP Client Control are

[Abort](#)

[Account](#)

[Authenticate](#)

[Cancel](#)

[ChangeDir](#)

[Connect](#)

[CreateDir](#)

[DeleteDir](#)

[DeleteFile](#)

[Disconnect](#)

[GetDoc](#)

[GetFile](#)

[Help](#)

[List](#)

[Mode](#)

[NameList](#)

[NOOP](#)

[ParentDir](#)

[PrintDir](#)

[PutFile](#)

[Reinitialize](#)

[SendDoc](#)

[Site](#)

[State](#)

System

Type

Abort

Description

Requests a FTP Server to abort the last data transfer request. Similar to the FTP RFC-959 ABORT command.

Return Value

Void.

Syntax

object.**Abort**

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

Parameters

None.

Comments

This event usually terminates any data connection while leaving the control connection intact.

Account

Description

Sends account information to remote host. Similar to the FTP RFC-959 ACCT command.

Return Value

Void.

Syntax

object.**Account** *Account*

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

Parameters

Account

String containing new account information.

Data Type: String

Param: Out

Default Value: Empty

Comments

Use the ReplyString property to determine the result of this call.

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` or `Password` is 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 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 control or collection during execution.

Parameters

None.

ChangeDir

Description

Requests FTP Server to change the remote host current directory to the specified directory. Similar to the FTP RFC-959 CWD command.

Return Value

Void.

Syntax

object.**ChangeDir** *directory*

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

Parameters

Directory

String containing new directory name.

Data Type: String

Param: IN

Default Value: Empty

Comments

Use the ReplyString property to determine the result of this call.

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 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.

CreateDir

Description

Creates the specified directory on the remote host. Similar to the FTP RFC-959 MKD command.

Return Value

Void.

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

Syntax

object.CreateDir Directory

Parameters

Directory

String containing the directory name.

Data Type: String

Default Value: Empty

Comments

Use the ReplyString property to determine the result of this call.

DeleteDir

Description

Deletes the specified directory file from the remote host. Similar to the FTP RFC-959 RMD command.

Return Value

Void.

Syntax

object.DeleteDir Directory

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

Parameters

Directory

String containing directory to delete.

Data Type: String

Param: IN

Default Value: Empty

Comments

Use the ReplyString property to determine the result of this call.

DeleteFile

Description

Deletes the specified file from the remote host. Similar to the FTP RFC-959 DELE command.

Return Value

Void.

Syntax

object.DeleteFile [*FileName*]

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

Parameters

FileName

Optional. String containing file to delete. This argument overrides the values from the corresponding RemoteFile property. The value of the property will not change. If the argument is omitted, the value from the corresponding property will be used to provide a filename for the delete operation.

Data Type: String

Param: IN

Default Value: Empty

Comments

Use the ReplyString property to determine the result of this call.

Disconnect

Description

Disconnects session with remote host and terminates any data connection.

Return Value

Void.

Syntax

object.**Disconnect**

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

Parameters

None.

Execute

Description

Issues the RFC 959 Quote command to the server.

Return Value

Void.

Syntax

object.Execute cmd

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

Parameters

Cmd

String containing the command to be invoked on the remote FTP server. This forces the FTP server to process a specific command(s) without actually issuing the command from the client.

Data Type: String

Param: IN

Default Value: Empty

Comments

Use the ReplyString property to check the return value during the Execute event notification.

GetDoc

See Also

Description

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

Return Value

Void.

Syntax

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

The object placeholder is required and evaluates to the name of the 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

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 FTP means retrieving a file 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 FTP control, the "ftp:" 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.

[DocInput](#) Property

[DocOutput](#) Property

[DocInput](#) Event

[DocOutput](#) Event

[Common Control Objects](#)

GetFile

See Also

Description

Gets the specified file from the remote host and places it in the current directory.

Return Value

Void.

Syntax

object.**GetFile** [*RemoteFile*,] [*LocalFile*]

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

Parameters

RemoteFile

Optional. String containing the remote file to retrieve.

Data Type: String

Param: IN

Default Value: Empty

LocalFile

Optional. String containing the local filename to use when saving the remote file.

Data Type: String

Param: IN

Default Value: Empty

Comments

The arguments override the values from the corresponding RemoteFile and LocalFile properties. The value of the properties will not change. If the arguments are omitted, the value from the corresponding property will be used to provide a filenames for the get file operation.

Both local and remote names should be specified, even if they are the same.

Use the ReplyString to determine the result of this call. The data from this method is sent to the DocStream interface via the DocOutput event. During processing of the DocOutput event, the Operation property is set to FTPFile.

[Operation](#) Property

[DocOutput](#) Event

Help

Description

Gets FTP help from the remote host. Similar to the FTP RFC-959 HELP command.

Return Value

Void.

Syntax

object.Help Help

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

Parameters

Help

String containing Help Commands supported

Data Type: String

Param: IN

Default Value: Empty

Comments

Use the ReplyString property to determine the result of this call.

List

See Also

Description

Requests a detailed directory listing of the specified directory from the remote host. Similar to the FTP RFC-959 LST command.

Return Value

Void.

Syntax

object.List [*directory*]

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

Parameters

Directory

Optional. String containing path of remote host from which to list directories.

Data Type: String

Default Value: Empty

Comments

The data from this method is sent to the DocStream interface via the DocOutput event. During processing of the DocOutput event, the Operation property is set to ftpList. If the ListItemNotify property is set to True, the ListItem event is also generated for every item in the directory listing.

[DocOutput](#) and [ListItem](#) Event

[Operation](#) and [ListItemNotify](#) Property

Mode

Description

Sets data transfer mode of remote host. Similar to the FTP RFC-959 MODE command.

Return Value

Void.

Syntax

object.Mode Mode

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

Parameters

Mode

Enumerated type containing new transfer mode information. The FTPModeConstants may have one of the following values.

- ftpStream = 0
- ftpBlock = 1
- ftpCompressed = 2

Data Type: FTPModeConstants

Param: IN

Default Value: Empty

NameList

See Also

Description

Requests a directory listing of the specified directory from the remote host. Similar to the FTP RFC 959 NLST command.

Return Value

Void.

Syntax

object.NameList [directory]

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

Parameters

Directory

Optional. String containing remote host path from which to list directories.

Data Type: String

Param: IN

Default Value: Empty

Comments

The data from this method is sent to the DocStream interface via the DocOutput event. During processing of the DocOutput event, the Operation property is set to ftpNameList. If the ListItemNotify property is set to True, the ListItem event is also generated for every item in the directory listing.

[DocOutput](#) and [ListItem](#) Event

[Operation](#) and [ListItemNotify](#) Property

NOOP

Description

Issues the NOOP command to the server.

Return Value

Void.

Syntax

object.**NOOP**

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

Parameters

None.

Comments

Use the ReplyString property to determine the result of this call.

ParentDir

Description

Requests the FTP Server change to the parent of the current directory, if one exists.

Return Value

Void.

Syntax

object.**ParentDir**

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

Parameters

None.

Comments

Use the ReplyString property to determine the result of this call.

PrintDir

Description

Requests the FTP Server query the current directory of the remote host. Similar to the FTP RFC-959 PWD command.

Return Value

Void.

Syntax

object.PrintDir

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

Parameters

None.

Comments

Use the ReplyString property to determine the result of this call. You will need to parse the ReplyString to determine the directory name. You can also obtain this information from the RemoteDir property.

PutFile

See Also

Description

Puts specified file on the Server's current directory.

Return Value

Void.

Syntax

object.PutFile [*LocalFile*] [,*RemoteFile*,]

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

Parameters

LocalFile

Optional. String containing the name of the local file to be transferred to the remote machine.

Data Type: String

Param: IN

Default Value: Empty

RemoteFile

Optional. String containing name of file to be placed on the remote machine.

Data Type: String

Param: IN

Default Value: Empty

Comments

The arguments override the values from the corresponding RemoteFile and LocalFile properties. The value of the properties will not change. If the arguments are omitted, the value from the corresponding property will be used to provide a filenames for the put file operation.

Both local and remote names should be specified, even if they are the same.

Use the ReplyString property to determine the result of this call during DocInput event processing. The DocStream DocInput event is activated when file data is streamed out of the FTP Control.

DocInput Event

Reinitialize

Description

Issues the Relnit command to the server.

Return Value

Void.

Syntax

object.**Reinitialize**

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

Parameters

None.

Comments

Use the ReplyString property to determine the result of this call.

SendDoc

See Also

Description

A DocInput related method that requests sending a document identified by a URL.

Return Value

Void.

Syntax

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

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

Parameters

URL

Optional. The URL identifying the remote document to be sent. If specified, the URL property will be set to this value.

Data Type: STRING

Param: IN

Default Value: DocInput.URL

Headers

Optional. Headers used for sending the document. This argument only applies to protocols where document headers can be sent (for example, SMTP and HTTP).

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 in FTP means putting a file on the server.

The URL and (for some controls) 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.

For controls such as HTTP that return a reply document, the OutputFile argument indicates where the reply 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 FTP control, the "ftp:" 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.

[DocInput](#) Property

[DocOutput](#) Property

[DocInput](#) Event

[DocOutput](#) Event

[Common Control Objects](#)

Site

Description

Issues a Site command to the remote server. This command is used during logon to determine the file system supported on the server.

Return Value

Void.

Syntax

object.Site [Sitecmd]

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

Parameters

Sitecmd

Optional. String containing Site command line options to be processed by the server (for example, "DIRSTYLE")

Data Type: String

Param: IN

Default Value: Empty

Comments

Use the ReplyString property to determine the result of this call.

State

Description

Requests status from the remote host. Similar to the FTP RFC-959 STAT command.

Return Value

Void.

Syntax

object.State [*State*]

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

Parameters

State

Optional. String containing status string.

Data Type: String

Param: OUT

Default Value: Empty

Comments

Use the ReplyString property to determine the result of this call during the State event notification.

System

Description

Issues a system command to the remote server. It is similar to the FTP RFC-959 SYST command.

Return Value

Void.

Syntax

object.**System**

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

Parameters

None.

Comments

Use the ReplyString property to determine the result of this call.

Type

Description

Issues a Type command to the remote server. This command is entered prior to a data transfer to set the transfer type.

Return Value

Void.

Syntax

object.Type ftpType

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

Parameters

ftpType

Enumerated type containing type of data. The server attempts to use this value for data representation if it is supported. Possible values are:

- ftpAscii = 0
- ftpImage = 2
- ftpEBCDIC = 1
- ftpBinary = 3

Data Type: FTPTTypeConstants

Param: IN

Default Value: ftpAscii

Events

Events are used for FTP 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.

Almost all FTP Control methods have an associated event that is activated after the server processes the command (either successfully or unsuccessfully). During the event notification of any given command, you need to check the ReplyString property to determine the outcome of the command. For example, you may call the Site method, and receive a Site event. During notification of the Site event, you would check the ReplyString property to see the result of the Site command.

The FTP methods GetFile, GetDoc, List, and NameList all retrieve data and share the same DocOutput event notification. As a developer you need to know which of the operations caused the event and redirect the data to the appropriate place (i.e., send List data to a list box control, and send file data to an edit control for display.) To do this, check the Operation property to determine the activating event.

[Abort](#)

[Account](#)

[Authenticate](#)

[Busy](#)

[Cancel](#)

[ChangeDir](#)

[CreateDir](#)

[DelDir](#)

[DelFile](#)

[Disconnect](#)

[DocInput](#)

[DocOutput](#)

[Error](#)

[Execute](#)

[Help](#)

[ListItem](#)

[Mode](#)

[NOOP](#)

[ParentDir](#)

[PrintDir](#)

[ProtocolStateChanged](#)

[Reinitialize](#)

[Site](#)

[StateChanged](#)

[System](#)

[TimeOut](#)

[Type](#)

Abort

Description

This event is activated after the [Abort](#) method is called. It aborts any active FTP process.

Syntax

object **Abort**

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

Parameters

None.

Account

Description

This event is activated after the [Account](#) method is called. It requests the remote host set the account to the one specified.

Syntax

*object*_Account

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

Parameters

None.

Comments

Use the ReplyString property to determine the server reply after event processing.

Authenticate

Description

This event is activated after the authentication initiated by [Authenticate](#) method has been completed.

Syntax

object_ **Authenticate** (*Succeeded As Boolean*)

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

Parameters

Succeeded

Indicates result code of the authentication.

Data Type: Boolean

Comments

Use the ReplyString property to determine the server reply after authentication

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 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 control or collection during execution.

Parameters

None.

ChangeDir

Description

This event is activated after the [ChangeDir](#) method is called. It changes the current working directory.

Syntax

*object*_ChangeDir

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

Parameters

None.

Comments

Use the ReplyString property to determine the server reply after event processing.

CreateDir

Description

This event is activated after the [CreateDir](#) method is called. It creates a new directory.

Syntax

*object_***CreateDir**

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

Parameters

None.

DelDir

Description

This event is activated after the [DeleteDir](#) method is called. It deletes the specified directory on the remote host.

Syntax

object_DelDir

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

Parameters

None.

Comments

Use the ReplyString property to determine the server reply after event processing.

DelFile

Description

This event is activated after the [DeleteFile](#) method is called. It deletes the specified file located in the path specified on the remote host.

Syntax

*object_***DelFile**

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

Parameters

None.

Disconnect

Description

This event is activated after a connection has been successfully closed. It is sent as a result of calling the [Disconnect](#) method.

Syntax

object_Disconnect

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

Parameters

None.

Comments

Use the ReplyString property to determine the server reply after event processing.

DocInput

See Also

Description

A DocInput related event that indicates the input data has been transferred.

Syntax

object_DocInput (DocInput As DocInput)

The object placeholder is required and evaluates to the name of the 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.

Common Control Objects

DocOutput

See Also

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 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.

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 control or collection during execution.

Parameters

ErrCode

The short error code. For a complete list of FTP error codes, see [FTP 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.

Execute

Description

This event is activated after the [Execute](#) method is called. It issues a command to the server for processing.

Syntax

*object_***Execute**

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

Parameters

None.

Comments

Use the ReplyString property to determine the results of the Execute command during event notification.

Help

Description

This event is activated after the [HELP](#) method is called. It requests the remote host send help information with the specified HELP parameters.

Syntax

object_Help

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

Parameters

None.

Comments

Use the ReplyString property to determine the server reply.

ListItem

See Also

Description

This event is activated for every element in a directory listing when the [ListItemNotify](#) property is set to TRUE. This lets you parse the directory elements after issuing a List or NameList command.

Syntax

object_ListItem (*Item As ftpDirItem*)

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

Parameters

Item

Object describing a directory listing element. The object contains the filename, size, date, and attributes of the current listing item.

Data Type: ftpDirItem

Param: OUT

Default Value: N/A

[FTPDirItem Objects](#)

[ListItemNotify](#) Property

Mode

Description

This event is activated after the [Mode](#) method is called. It sets the remote host data transfer mode to the mode specified.

Syntax

object_Mode

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

Parameters

None.

Comments

Use the ReplyString property to determine the server reply.

NOOP

Description

This event is activated after the [NOOP](#) method is called. It requests an OK reply from the server.

Syntax

*object_***NOOP**

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

Parameters

None.

Comments

Use the ReplyString property to determine the response from the server.

ParentDir

Description

This event is activated after the [ParentDir](#) method is called. It changes the current directory on the remote host to the parent directory, if one exists.

Syntax

*object*_ParentDir

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

Parameters

None.

Comments

Use the ReplyString property to determine the server reply.

PrintDir

Description

This event is activated after the [PrintDir](#) method is called. It requests the remote host print the working directory.

Syntax

*object*_ParentDir

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

Parameters

Refer to the [ProtocolState](#) property and [ProtocolStateString](#) for possible values of the state parameter.

Reinitialize

Description

This event is activated after the [Reinitialize](#) method is called. It causes the server to log off the current user while maintaining an open connection.

Syntax

*object*_Reinitialize

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

Parameters

None.

Comments

Use the ReplyString property to determine the result of the Reinitialize method during event notification.

Site

Description

This event is activated after the [Site method](#) is called. It requests directory and file formatting information from the remote host.

Syntax

object_Site

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

Parameters

None.

Comments

Use the ReplyString property to determine the server reply.

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 control or collection during execution.

Parameters

Refer to the [State](#) property and [StateString](#) for possible values of the state parameter.

System

Description

This event is activated after the [System](#) method is called. It requests the type of operating system on the server.

Syntax

object_System

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

Parameters

None.

Comments

Use the ReplyString property to determine the server reply.

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 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

Type

Description

This event is activated the Type method is called. Specifies how the remote host should handle the transferred data.

Syntax

object_Type

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

Parameters

None.

Comments

Use the ReplyString property to determine the server reply.

FTPDirItem Object

The FTPDirItem object is used for directory and file parsing. Generally when a List or NameList command is processed, the server returns a byte stream whose format is dependent on the operating system running on the remote server.

Directory parsing is turned on by setting the [ListItemNotify](#) property to TRUE. Whenever a List command is processed, the FTP Control generates ListItem events for every directory or file element in the listing. The parameter list of the ListItem event includes an FTPDirItem object which contains the filename, size, date, and attributes for the current listing element.

If the ListItemNotify property is set to FALSE; the directory listing is sent to the Document Output object which in turn sends event notifications as streams of data are received. This is useful when you want to put directory listing data into a static control or an edit field in an 'as-is' state.

[Example](#)

[Properties](#)

Example

The following example demonstrates how to use the FTPDirItem object with directory parsing.

1. Set the ListitenNotify property to TRUE.

```
1. ftpctl1.ListItemNotify = TRUE
```

1. Invoke a List or NameList command

```
1. ftpctl1.List
```

1. As you receive the ListItem events, the item parameter contains the current directory element :

```
Private Sub Ftpctl1_OnListItem(ByVal FTPDirItem As Object)
    Form2.text1.Text = Form2.text1.Text & Chr$(13) & Chr$(10) & FTPDirItem.Filename &
        Chr$(9) & FTPDirItem.Size & Chr$(9) & FTPDirItem.Date & Chr$(9) & FTPDirItem.Attributes
        & Chr$(9) & Chr$(9) & FTPDirItem.Detail
End Sub
```

Properties

FTPDirItem object properties let you access the filename, date, file size, and file attributes of a directory listing.

[Attributes](#)

[Date](#)

[Detail](#)

[Filename](#)

[Size](#)

Attributes

Description

File or directory attributes of the current directory listing item.

Syntax

object.Attributes

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

Permission

R (Read only).

Availability

D (Design).

Data Type

Integer.

Default Value

0.

Range

This value is an integer value that may be any of the following

<u>Attribute</u>	<u>Value</u>
FtpUnknown	0x0000
FtpDir	0x0001
FtpFile	0x0002
FtpLink	0x0010
FTPDisk	0x0004

Date

Description

Last modified date of the current directory listing item.

Syntax

object.Date

The object placeholder is required and evaluates to the name of the relevant CONTROL OR COLLECTIONObject during execution.

Permission

R (Read only).

Availability

D (Design).

Data Type

String.

Default Value

Empty.

Range

N/A

Detail

Description

Contains the raw unparsed data that the FTP server sends back for this directory item.

Syntax

object.**Detail**

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

N/A

Filename

Description

File or directory name of the current directory listing item.

Syntax

object.**Filename**

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

Permission

R (Read only).

Availability

D (Design).

Data Type

String.

Default Value

Empty.

Range

N/A

Size

Description

File or directory name of the current directory listing item.

Syntax

object.**Size**

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

Permission

R (Read only).

Availability

D (Design).

Data Type

Long.

Default Value

0.

Range

Size of Long.

FTP Error Codes

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

Error Code	Error Message
2104	Port Command Failed. Unable to open Port.
2105	Abort Command Failed. Unable to Abort last command.
2106	Account Command Failed. Unable to complete.
2107	Change Directory Command Failed. Unable to change to specified directory.
2108	Connect Command Failed. Unable to connect to remote host.
2109	Create Directory Command Failed. Unable to create specified directory.
2110	Delete Directory Command Failed. Unable to delete specified directory.
2111	Delete File Command Failed. Unable to delete specified file.
2112	Disconnect Command Failed. Unable to disconnect from remote host.
2113	Get File Command Failed. Unable to retrieve specified file.
2114	Help Command Failed. Unable to retrieve help from remote host.
2115	NOOP Command Failed. Control connection error.
2116	Name List Command Failed. Unable to retrieve Named list; possible data connection error.
2117	List Command Failed. Unable to retrieve detailed list; possible data connection error.
2118	Parent directory Command Failed. Unable to change directory up.
2119	Print Directory Command Failed. Unable to print current directory of remote host.
2120	Put File Command Failed. Unable to put file on remote host.
2121	Put Unique File Command Failed. Unable to put unique file on remote host.

- 2122 Reinitialize Command Failed.
Unable to reinitialize login on
remote host.
- 2123 Rename File Command Failed.
Unable to rename specified file on
remote host.
- 2124 Retrieve File Command Failed.
Unable to retrieve specified file from
remote host.
- 2125 Status Command Failed. Unable
to retrieve status from remote host.
- 2126 System Command Failed. Unable to
issue SYST command to remote
host.
- 2127 Type Command Failed. Unable to
set transfer type on remote host.
- 2128 Error setting OutputDocStream
property
- 2129 Error setting InputDocStream
property
- 2157 Command not implemented.
- 2171 Maximum connection(<number>
reached.

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.

ActiveX

ActiveX is a trademark for Microsoft Corporation.

