Getting Started with the CIFTP Control

Related Topics

This topic introduces File Transfer Protocol (FTP) and the Crescent Internet ToolPak FTP Control (CIFTP). You can access the following topics:

- What FTP is
- What the CIFTP control is
- What the CIFTP programming tasks are

The programming tasks topics identify each task that CIFTP supports; it lists the properties to set and the methods to call, then it illustrates the task with a code sample.

What is FTP?

Related Topics

File Transfer Protocol (FTP) is an Internet Protocol used to copy files between computers without regard for operating system imposed differences in the files structure. FTP can be used to copy text or binary files.

How Does FTP Work?

FTP uses a two channel model to transfer data. One channel passes the commands/requests between the client and host machines (this is known as the access channel); while the other channel passes the data between client and host (this is known as the data channel). Once you open the access channel, it stays open until the client notifies the server to close it. The data channel is opened and closed in response to specific data transfer requests. For example, if you want to copy a file from a host to a local workstation, you open the access channel and request the file. The server then opens the data channel and copies the file to the local machine. When the file transfer is complete, the server closes the data channel. To copy another file, you make the request via the access channel. The server opens another data channel, copies the file and closes the new data channel. The access channel remains constant while data channels are created and destroyed throughout the session in response to requests.

By default, FTP servers listens for access requests on port 21. When the client requests a file from the server, the client tells the server what port number the server should use to connect to pass data to the client. The server connects to that port number on the client system and downloads the files via that port.

FTP is a transaction oriented protocol (like HTTP). You request data, when the server returns the data, the connection (to the data channel) terminates.

What is the CIFTP Control?

Related Topics

The Crescent Internet ToolPak File Transfer Protocol (CIFTP) control lets Visual Basic programmers build customized FTP client applications that can copy files to and from FTP servers. The CIFTP control complies with the FTP standards defined in RFC 959. You can use the FTP control to perform any valid FTP function. You might use it to build applications like these:

- Build a Visual Basic application that monitors VB SourceSafe for the latest software builds. When the application detects a new build, the application copies the build from SourceSafe to an FTP server where it can be accessed by users, testers, etc.
- Build a Visual Basic application that monitors an FTP site for new files. As new files appear at the FTP site, they can be emailed to specified users. (Integrate CIFTP and CISMTP.)
- Build a Visual Basic application that performs FTP functions but provides the functionality in a way that users do not need to know the FTP commands.

CIFTP has a simple programming interface in which you set property values and call methods to perform the desired tasks. The methods (which generally do not take arguments) take the necessary data directly from the property sheet, and submits the request to the server. When the server returns its response, the control parses the data and updates the appropriate properties or populates a ListBox. You can simply read the data from a property value or a ListBox object.

When the server passes data to the client in response to a request from CIFTP, it passes the data in stream mode. *Stream mode* transfers open a dedicated channel for a single file transfer and closes it when at the end of the file. The data is passed as a a raw stream of bytes.

CIFTP supports two types of methods: low-level and high-level. A *low-level method* performs a single function like providing a single piece of information to the FTP server, such as a password. A complete task, therefore, is comprised of multiple low-level method calls. A *high-level method* performs a complete task. It encapsulates all of the low-level methods necessary to perform the complete task into a single method call, thus significantly reducing the amount of code that you must write. CIFTP includes these three high-level methods: GetDirectory, GetFile, and PutFile.

Programming Tasks

Related Topics

The following topics describe how to program some simple tasks with the CIFTP control.

- Connecting to an FTP Server
- Disconnecting From an FTP Server
- Copying a File From an FTP Server Using Low-level Methods
- Copying a File From an FTP Server Using High-level Methods
- Obtain a Directory Listing from an FTP Server Using Low-level Methods
- Obtain a Directory Listing from an FTP Server Using High-level Methods Each topic lists the properties that you must set, and the methods that you must call to perform a specific task, and identifies the events that are fired. A code fragment that illustrates the task follows the task list.

Connecting to an FTP Server

Related Topics

Each client application must contain code that first connects to an FTP server and opens the access channel (also called the *access control channel*). Once the access control channel is open, the client can request data from the FTP Server. Here are the steps

1. Set these properties:
   HostName (or HostAddress)
   AccessPort
   LoginName
   Password
   ServerOSType

2. Call the ConnectToAccessControlChannel method.

3. When the AccessControlChannelConnection event fires, call these methods in this order:
   USER
   PASS
   SYST (only required if ServerOSType property was set to Automatic).

4. Call the PASV method.

5. When the DataPortSet event fires, call the ConnectToDataChannel method.

6. When the DataControlChannelConnection event fires, invoke any of the methods listed or the SendFTPCommand method for an FTP command not explicitly supported.

Heres what the code looks like:

```
Form_Load

   CIFTP1.HostName = ftp.myhost.com
   CIFTP1.AccessPort = 21
   CIFTP1.LoginName = MyName
   CIFTP1.Password = MyPass
   CIFTP1.ServerOSType = 1
   CIFTP1.ConnectToAccessControlChannel

   CIFTP1_AccessControlChannelConnection()
   CIFTP1.USER
   CIFTP1.PASS
   CIFTP1.PASV

   CIFTP1_DataPortSet()
   CIFTP1.ConnectToDataChannel

   CIFTP1_DataControlChannelConnection()

   Call any method here...
```

▪ Disconnecting From an FTP Server

Related Topics

If you have an active data access channel and have performed all of the FTP tasks that you want to with the FTP server, follow these steps to end the FTP session.

1. Call the QUIT method.
   When QUIT completes, the following occurs:
   CIFTP fires the AccessControlPacketSent event.
   The server issues a response to the client which when received causes:
   AccessControlChannelClosed event to fire and the
   SocketClosed event to fire.

The code looks like this:

```
CIFTP1.QUIT
```

▪ Copying a File From an FTP Server Using Low-level Methods

Related Topics

Once youve established a connection to the access control channel and the data control channel, you can perform any FTP command like copying a file from the FTP Server. (See the *Connecting to an FTP Server* section for the connection steps.) Here are the steps:

1. Once you are connected and the DataControlChannelConnection event fires, set these properties:
   RemoteFileName
   LocalFileName
   RepresentationType

2. Call the RETR method.

   When the RETR completes, CIFTP fires the AccessControlPacketSent event.

   When the client receives the data from the server, CITFP fires the AccessControlPacketReceived and the DataControlPacketReceived events.The control writes the data to the file named by the LocalFileName property. After the data is written to the local file, the FileClosed event fires and the DataControlChannelClosed and the TotalFileBytesReceived events fire.

   Heres what the code looks like

```
CIFTP1_DataControlChannelConnection()
CIFTP1.RemoteFileName = accounts.dat
CIFTP1.LocalFileName = C:\Myfile.dat
CIFTP1.RepresentationType = 0
CIFTP1.RETR
```

▪ Copying a File From an FTP Server Using High-level Methods

Related Topics

The high-level method GetFile performs all of the necessary connection to the data control and access control channels. You must simply set the necessary properties before calling GetFile.

Here are the steps:

1. Set these properties:
   AccessPort
   HostName (or HostAddress)
   LocalFileName
   LoginName
   Password
   RemoteFileName
   RepresentationType
   ServerOSType
   WorkingDirectory
2. Call the GetFile method.

   When GetFile successfully completes, CIFTP fires the GotFile. If GetFile is unsuccessful, the InternetError event fires.

   CIFTP writes the data to the file named by the LocalFileName property.

   Heres what the code looks like

```
Form_Load
   CIFTP1.HostName = ftp.myhost.com
   CIFTP1.AccessPort = 21
   CIFTP1.LoginName = MyName
   CIFTP1.Password = MyPass
   CIFTP1.ServerOSType = 1
   CIFTP1.RemoteFileName = accounts.dat
   CIFTP1.LocalFileName = C:\Myfile.dat
   CIFTP1.RepresentationType = 0
   CIFTP1.WorkingDirectory = C:\
   CIFTP1.GetFile
```

▪ Obtain a Directory Listing from an FTP Server Using Low-level Methods

Related Topics

Once youve established a connection to the access control channel and the data control channel, you can perform any FTP command like obtaining a directory listing. (See the *Connecting to an FTP Server* section for the connection steps.) Here are the steps:

1. Once you are connected and the DataControlChannelConnection event fires, set these properties:
   DirectoryListBoxName
   FileListBoxName
   RepresentationType
   WorkingDirectory

2. Call the LIST method.

   After you call LIST, the AccessControlPacketSent event fires. When the server responds the following occurs: the AccessControlPacketReceived event fires, the DataControlPacketReceived event fires, the MethodStateChanged event fires, and the TotalFileBytesReceived event fires (multiple times).

Once all of the data is received the following occurs: the DataControlChannelClosed event fires, MethodStateChanged event fires, the ListBoxesPopulated event fires, the FileClosed event fires and the TotalFileBytesReceived event fires.

▪ Obtain a Directory Listing from an FTP Server Using High-level Methods

Related Topics

The high-level method GetDirectory performs all of the necessary connection to the data control and access control channels. You must simply set the necessary properties before calling GetDirectory.

Here are the steps:

1. Set these properties:
   AccessPort
   FileListBoxName
   DirectoryListBoxName
   HostName (or HostAddress)
   LocalFileName
   LoginName
   Password
   RemoteFileName
   RepresentationType
   ServerOSType
   WorkingDirectory
2. Call the GetDirectory method.

If GetDirectory succeeds, the GotDirectory event fires and the Directory and File ListBox objects are populated with the requested data. If it fails, the InternetError event fires.

The Crescent Internet ToolPak FTP Control

Related Topics

Control File

CIFTP.OCX
CIFTP16.OCX

Object Type

CIFTP

Purpose

The Crescent Internet ToolPak File Transfer Protocol (CIFTP) control enables you to build customized FTP applications such as:

- Copying files from an FTP server.
- Copying files to an FTP server.
- Monitoring files on an FTP site and notifying specified users when an update occurs.

The CIFTP control complies with the FTP standards defined in RFC 959.

Properties

About	AccessPort c	Container
DataPort c	DirectoryListBoxName c	Drag
DragIcon	DragMode	EventState c
FileListBoxName c	Height	HostAddress c
HostName c	Index	Left
LocalFileName c	LoginName c	MethodState c
Move	Name	Object
Parent	Password c	RemoteFileName c
RepresentationType c	ServerOSType c	ShowWhatsThis
Top	Visible	WhatThisHelpID
Width	WorkingDirectory c	

Events

AccessControlChannelClosed c	AccessControlChannelConnection c	AccessControlPacketReceived c
AccessControlPacketSent c	DataControlChannelClosed c	DataControlChannelConnection c
DataControlPacketReceived c	DataPortSet c	DragDrop
DragOver	EventStateChanged c	FileClosed c
GotDirectory c	GotFile c	InternetError c
ListBoxesPopulated c	MethodStateChangedc	PutFile c
SocketClosedc	TotalFileBytesReceivedc	WSAErrorc

Methods

CDUP c	CleanupConnection c	ConnectToAccessControlChannel c
ConnectToDataChannel c	CWD c	GetDirectory c
GetFile c	LIST c	PASS c
PASV c	PutFile c	PWD c
QUIT c	RETR c	SendFTPCommand c
STOR c	SYST c	TYPE c
USER c	ZOrder	

c A custom or modified property, method, or event.

AccessPort Property

Applies To

CIFTP

Purpose

The AccessPort property sets the port number used for access control requests and replies.

Syntax

[*Form.***]***CIFTP*.AccessPort**[** = *Integer%***]**

Data Type

Integer

Usage

Read/Write at design time and runtime.

Comments

The AccessPort is the port on which the AccessControlChannel operates. The AccessControlChannel relays commands and responses between the client and server. Set this property before attempting to connect to an FTP server.

See Also

AccessControlChannelConnection, ConnectToAccessControlChannel, DataPort

DataPort Property

Applies To

CIFTP

Purpose

The DataPort Property returns the port number used for the receipt of data.

Syntax

[*Form.***]***CIFTP*.DataPort**[** = *Integer%***]**

Data Type

Integer

Usage

Read/Write at runtime and design time.

Comments

The FTP server notifies the client which port to establish as the port on which to send the data. This property is updated by CIFTP once a response to the PASV method is received.

When it is set, CIFTP fires the DataPortSet event. Once the DataPortSet event fires, you can call the ConnectToDataPort method to establish the DataPort connection. You can start any FTP task once the DataChannelConnection event fires.

See Also

AccessControlChannelConnection, AccessPort, ConnectToAccessControlChannel, ConnectToDataChannel, DataPortSet, PASV

DirectoryListBoxName Property

Applies To

CIFTP

Purpose

The DirectoryListBoxName property sets the name of a ListBox object that is bound to the CIFTP control. When the control retrieves a directory listing from the FTP server, it populates the listbox with the data.

Syntax

```
Set [Form.]CIFTP.DirectoryListBoxName [ = ObjectName]
```

Data Type

Object

Usage

Read/Write at runtime only.

Comments

Set DirectoryListBoxName, FileListBoxName and WorkingDirectory before invoking the LIST or the PWD methods. After the DirectoryListBox and the FileListBox objects are populated, the ListBoxesPopulated event fires.

See Also

FileListBoxName, LIST, PWD, WorkingDirectory

EventState Property

Applies To

CIFTP

Purpose

The EventState property returns the event currently in progress.

Syntax

```
[Integer%] = [Form.]CIFTP.EventState
```

Data Type

Integer

Usage

Read/Write at runtime.

Comments

CIFTP updates EventState when any of the CIFTP events (except the PacketReceived events) fire. When the EventState property changes, CIFTP fires the EventStateChanged event.

The valid EventStateproperty values are listed in the following table and are also in the CITPAK.BAS file.

This event...	has this constant...	with this integer value...
FileClosed	CIFTP_FCLOSED	100
SocketClosed	CIFTP_SCLOSED	101
AccessControlChannelClosed	CIFTP_ACCLOSED	102
DataControlChannelClosed	CIFTP_DCCLOSED	103
AccessControlChannelConnection	CIFTP_ACCONN	104
DataControlChannelConnection	CIFTP_DCCONN	105
DataPortSet	CIFTP_DPORTSET	106
ListBoxesPopulated	CIFTP_LBPOP	107

You can test EventState to initiate an action or control program flow.

See Also

EventStateChanged

FileListBoxName Property

Applies To

CIFTP

Purpose

The FileListBoxName property sets the name of the listBox object that is bound to the CIFTP control. When the control retrieves a file list from the FTP server, it populates the listbox with the data.

Syntax

```
Set [Form.]CIFTP.FileListBoxName [ = ObjectName]
```

Data Type

Object

Usage

Read/Write at runtime only.

Comments

Set DirectoryListBoxName, FileListBoxName and WorkingDirectory before invoking the LIST or the PWD methods.

The LIST or PWD method populates the FileListBox object. After the FileListBox and the DirectoryListBox objects are populated the ListBoxesPopulated event fires.

See Also

DirectoryListBoxName, LIST, PWD, WorkingDirectory

HostAddress Property

Applies To

CIFTP

Purpose

The HostAddress property sets or returns the IP address of the FTP server.

Syntax

[*Form.***]***CIFTP*.HostAddress**[** = String$**]**

Data Type

String

Usage

Read/Write at design time.

Comments

The HostAddress identifies the host to the network (the Internet). HostAddress consists of the network number and the local host number. You must use either the HostName or HostAddress to send or retrieve data from an FTP server.

An example of an IP address is:

```
198.137.64.1
```

NOTE You must use either the HostName or HostAddress property to connect to an FTP server. When you provide both, CIFTP uses HostName.

See Also

HostName

HostName Property

Applies To

CIFTP

Purpose

The HostName property sets or returns the DNS name of the FTP server.

Syntax

[*Form.***]***CIFTP*.HostName**[** = *String$***]**

Data Type

String

Usage

Read/Write at design time and runtime.

Comments

The HostName is the name that identifies an FTP server to the Internet. Depending on your connection to the Internet, the HostName might contain simply a machine name like `crescentserver`, or it can be a fully qualified Internet name that follows this format: `machine.organizationname.com`. Contact your site administrator to determine the correct format to supply.

Set HostAddress or HostName before calling the ConnectToFTPServer method.

NOTE You must use either the HostName or the HostAddress property to connect to an FTP server. When you provide both, CIFTP uses HostName.

See Also

HostAddress

LocalFileName Property

Applies To

CIFTP

Purpose

The LocalFileName property sets the path and the name of a file on the client (local machine).

Syntax

[*Form.***]***CIFTP*.LocalFileName**[** = *String$***]**

Data Type

String

Usage

Read/Write at design time and runtime.

Comments

When you use the RETR method to copy data from the FTP server, CIFTP updates the local file specified by the LocalFileName property. The data is appended to the end of the file giving it the potential to grow quite large.

When you use the STOR method to copy data to the FTP server, CIFTP copies the data from the file specified by the LocalFileName property to the server.

See Also

RETR, STOR

LoginName Property

Applies To

CIFTP

Purpose

The LoginName property sets the login name used to connect to an FTP server.

Syntax

[*Form.***]***CIFTP*.LoginName**[** = *String$***]**

Data Type

String

Usage

Read/Write at design time and runtime.

Comments

The USER method passes the LoginName to the FTP server.

NOTE Your applications users must have an account on the FTP server with the the appropriate access level to the files that the application accesses. In addition, some FTP servers support a user name of *anonymous*. Anyone can login as anonymous, but they will have limited privileges.

See Also

PASS, Password, USER

MethodState Property

Applies To

CIFTP

Purpose

The MethodState property returns the method currently in progress.

Syntax

`[`*`Integer%`*`] = [`*`Form.`*`]`*`CIFTP.`*`MethodState`

Data Type

Integer

Usage

Read/Write at runtime.

Comments

CIFTP updates MethodState when you call a method like PASS, PASV, or PWD. When the MethodStateproperty changes, CIFTP fires the MethodStateChanged event.

The valid MethodState property values are listed in the following table and are also in the CITPAK.BAS file.

This method...	has this constant...	with this integer value...
PASS	CIFTP_PASS	1
USER	CIFTP_USER	2
SYST	CIFTP_SYST	3
TYPE	CIFTP_TYPE	4
LIST	CIFTP_LIST	5
QUIT	CIFTP_QUIT	6
PASV	CIFTP_PASV	7
RETR	CIFTP_RETR	8
STOR	CIFTP_STOR	9
CWD	CIFTP_CWD	10
PWD	CIFTP_PWD	11
NLST	CIFTP_NLST	12
CDUP	CIFTP_CDUP	13

You can test the MethodState to initiate an action or control program flow.

See Also

MethodStateChanged

Password Property

Applies To

CIFTP

Purpose

The Password property sets the password.

Syntax

[*Form.***]***CIFTP*.Password**[** = *String$***]**

Data Type

String

Usage

Read/Write at design time and runtime.

Comments

The PASS method passes the Password to the FTP server. The password must be valid for the login name passed by USER.

See Also

LoginName, PASS, USER

RemoteFileName Property

Applies To

CIFTP

Purpose

The RemoteFileName property sets the path and name of the remote file that you want to retrieve from the FTP.

Syntax

[*Form.***]***CIFTP*.RemoteFileName**[** = *String$***]**

Data Type

String

Usage

Read/Write at design time and runtime.

Comments

The RETR method passes the RemoteFileName to the server. If the file does not exist, you will not receive any data.

If the file exists, and you have access privileges to it, the server passes the data in stream mode to the client. When the complete file is received by the client, the data channel on which the data was passed is closed and the following occurs multiple times depending on the files size:

1. The DataControlPacketReceived event fires.
2. The DataControlChannelClosed event fires.
3. The TotalFileBytesReceived event fires.
4. The LocalFile is populated and the FileClosed event fires.

See Also

DataControlChannelClosed, DataControlPacketReceived, FileClosed, LocalFileName, RETR, TotalFileBytesReceived

RepresentationType Property

Applies To

CIFTP

Purpose

The RepresentationType property sets the data format that you want the server to use when it transmits the data to the client.

Syntax

[*Form.***]***CIFTP*.RepresentationType**[** = *String$***]**

Data Type

String

Usage

Read/Write at design time and runtime.

Comments

The TYPE method passes the RepresentationType to the FTP server.

Use this value...	**to retrieve this kind of data...**
A	ASCII
I	Binary
E	EBCDIC

See Also

TYPE

ServerOSType Property

Applies To

CIFTP

Purpose

The ServerOSType property sets or returns the name of the FTP servers operating system.

Syntax

[*Form.***]***CIFTP*.ServerOSType**[** = *integer%***]**

Data Type

Integer

Usage

Read/Write at design time and runtime.

Comments

The enumerated values for this property are:

This value . . .	**sets ServerOSType to this . . .**
0	Automatic
1	Windows NT
2	UNIX

If you are not sure of the servers operating system type, set the ServerOSType to Automatic (0), then use the SYST method to obtain it. CIFTP uses this information to properly format directory and file listings in the ListBox objects.

See Also

SYST

WorkingDirectory Property

Applies To

CIFTP

Purpose

The WorkingDirectory property sets or returns the path and name of the current working directory on the FTP server.

Syntax

[*Form.***]***CIFTP*.WorkingDirectory**[** = *String$***]**

Data Type

String

Usage

Read/Write at design time and runtime.

Comments

If you know the directory on the FTP server that you want to access, set the WorkingDirectory property before calling the CWD method. If you do not set a WorkingDirectory, CIFTP logs you in to the default directory for the login. You can then use CWD or CDUP to traverse the servers file system.

The PWD method updates the WorkingDirectory property to the current directory on the FTP server.

See Also

CWD, FileListBoxName, LIST, WorkingDirectory

AccessControlChannelClosed Event

Applies To

CIFTP

Purpose

The AccessControlChannelClosed event fires when the connection to the FTP server terminates.

Syntax

```
Sub CIFTP_AccessControlChannelClosed()
```

Comments

You terminate a connection to an FTP server with the QUIT method.

See Also

ConnectToAccessControlChannel, QUIT

AccessControlChannelConnection Event

Applies To

CIFTP

Purpose

The AccessControlChannelConnection event fires when a connection to an FTP server is established. The connection is to an access control channel that passes requests and replies (but no data) between client and server.

Syntax

```
Sub CIFTP_AccessControlChannelConnection()
```

Comments

Use the ConnectToAccessControlChannel method to request a connection to an FTP server. Once the AccessControlChannelConnection event fires, your application can start to interact with the FTP server. You cannot however start to retrieve data from an FTP server until you also establish a data channel connection with the ConnectToDataControlChannel method.

See Also

AccessControlChannelConnection, AccessPort, ConnectToAccessControlChannel, ConnectToDataChannel, PASV

AccessControlPacketReceived Event

Applies To

CIFTP

Purpose

The AccessControlPacketRecevied event fires when CIFTP receives a packet, over the access control channel, from the FTP server.

Syntax

```
Sub CIFTP_AccessControlPacketReceived(ByVal Packet As String)
```

Comments

The *Packet* contains information from the server about whether an access control request was granted or denied. The server responds on the access control channel to every CIFTP method that you call.

See Also

AccessControlPacketSent, DataControlPacketReceived

AccessControlPacketSent Event

Applies To

CIFTP

Purpose

The AccessControlPacketSent event fires when CIFTP sends an access control request to an FTP server.

Syntax

```
Sub CIFTP_AccessControlPacketSent()
```

Comments

CIFTP issues an access control request when you invoke any method (except CleanupConnection). The FTP protocol requires that the access control channel be used for all requests for data as well as responses to requests between the client and server.

See Also

AccessControlPacketReceived

DataControlChannelClosed Event

Applies To

CIFTP

Purpose

The DataControlChannelClosed event fires when the client receives the data it requested from the server. Once the data is received, the server terminates the data control channel.

Syntax

```
Sub CIFTP_DataControlChannelClosed()
```

Comments

If you want to copy multiple files from the same FTP server, you must establish another data control channel connection for each file that you want to copy.

See Also

AccessControlChannelClosed, AccessControlChannelConnection, DataControlChannelConnection, FileClosed

DataControlChannelConnection Event

Applies To

CIFTP

Purpose

The DataControlChannelConnection event fires when a data connection to an FTP server is established.

Syntax

```
Sub CIFTP_AccessControlChannelConnection()
```

Comments

Use the ConnectToDataControlChannel method to request a data connection to an FTP server. Once the DataControlChannelConnection event fires, your application can request data from the FTP server. You cannot establish a data connection to an FTP server until you establish a connection to an AccessControlChannel connection.

See Also

AccessControlChannelConnection, ConnectToAccessControlChannel, ConnectToDataChannel

DataControlPacketReceived Event

Applies To

CIFTP

Purpose

The DataControlPacketReceived event fires when CIFTP receives a packet, over the data control channel, from the FTP server.

Syntax

```
Sub CIFTP_DataControlPacketReceived(ByVal Packet As String, ByVal bytes_in As Integer)
```

Comments

The *Packet* contains the data received from the server. It can contain a file listing, a directory listing, the contents of a file, or other data depending on the method used to obtain the packet. The *bytes_in* value is the number of bytes in the current packet.

See Also

AccessControlPacketReceived, AccessControlPacketSent

DataPortSet Event

Applies To

CIFTP

Purpose

The DataPortSet event fires when the DataPort property is populated as a result of the PASV method.

Syntax

```
Sub CIFTP_DataPortSet()
```

Comments

Once you create an access control connection to an FTP server, you must request that the server provide you with a port number to support data access requests. Use the PASV method to request a data access channel port. Once the server provides the DataPort and you connect to a data channel (call the ConnectToDataControlChannel method and wait for the DataControlChannelConnection event to fire), you can start retrieving and sending data to the FTP server via that port.

See Also

DataPort, PASV

EventStateChanged Event

Applies To

CIFTP

Purpose

The EventStateChanged event fires when the value of the EventState property changes.

Syntax

```
Sub CIFTP_EventStateChanged(ByVal state As Integer)
```

Comments

TheEventState property records the state of CIFTP; its value changes in response to other CIFTP events.You can test the EventState property to determine a state change and thus initiate an action or control program flow.

The valid EventStateproperty values are listed in the following table and are also in the CITPAK.BAS file.

This event...	has this constant...	with this integer value...
FileClosed	CIFTP_FCLOSED	100
SocketClosed	CIFTP_SCLOSED	101
AccessControlChannelClosed	CIFTP_ACCLOSED	102
DataControlChannelClosed	CIFTP_DCCLOSED	103
AccessControlChannelConnection	CIFTP_ACCONN	104
DataControlChannelConnection	CIFTP_DCCONN	105
DataPortSet	CIFTP_DPORTSET	106
ListBoxesPopulated	CIFTP_LBPOP	107

See Also

EventState

FileClosed Event

Applies To

CIFTP

Purpose

The FileClosed event fires when the file stream (specified by the LocalFileName property) on the client closes.

Syntax

```
Sub CIFTP_FileClosed()
```

Comments

The FTP server terminates the connection once it has successfully transferred the data to the client. The FileClosed event indicates that communications with the FTP server has terminated, and that all data has been retrieved and parsed.

See Also

DataControlChannelClosed, FileClosed, ListBoxesPopulated, LocalFileName

GotDirectory Event

Applies To

CIFTP

Purpose

The GotDirectory event signals the successful completion of the GetDirectory high-level method.

Syntax

```
Sub CIFTP_GotDirectory()
```

Comments

The GotDirectory event fires when the GetDirectory method successfully completes.

See Also

GetDirectory

GotFile Event

Applies To

CIFTP

Purpose

The GotFile event signals the successful completion of the GetFile method.

Syntax

```
Sub CIFTP_GotFile()
```

Comments

The GotFile event fires when the GetFile method successfully completes.

See Also

GetFile

InternetError Event

Applies To

CIFTP

Purpose

The InternetError event signals Internet errors specific to the FTP high level methods (GetDirectory, GetFile, and PutFile).

Syntax

```
Sub CIFTP_InternetError(ByVal error_number As Long, ByVal error_message As String)
```

Comments

The InternetError event fires whenever the FTP server returns an error in response to the GetDirectory, GetFile, and PutFile.

The error_number&:parameter is the error number returned by the FTP server. (It is usually 12003). The error_message& parameter is the error text as returned by the FTP server.

See Also

GetDirectory, GetFile, PutFile (method), GotDirectory, GotFile, PutFile (event)

ListBoxesPopulated Event

Applies To

CIFTP

Purpose

The ListBoxesPopulated event fires when the DirectoryListBox and FileListBox objects have been filled.

Syntax

```
Sub CIFTP_ListBoxesPopulated()
```

Comments

The event fires once after the ListBox objects are populated.

See Also

DirectoryListBoxName, FileListBoxName

MethodStateChanged Event

Applies To

CIFTP

Purpose

The MethodStateChanged event fires when the value of the MethodState property changes.

Syntax

```
Sub CIFTP_MethodStateChanged(ByVal state As Integer)
```

Comments

The MethodState property records the state of CIFTP; its value changes when a CIFTP method is called.You can test the MethodState property to determine a state change and thus initiate an action or control program flow.

The valid MethodState property values are listed in the following table and are also in the CITPAK.BAS file.

This method...	has this constant...	with this integer value...
PASS	CIFTP_PASS	1
USER	CIFTP_USER	2
SYST	CIFTP_SYST	3
TYPE	CIFTP_TYPE	4
LIST	CIFTP_LIST	5
QUIT	CIFTP_QUIT	6
PASV	CIFTP_PASV	7
RETR	CIFTP_RETR	8
STOR	CIFTP_STOR	9
CWD	CIFTP_CWD	10
PWD	CIFTP_PWD	11
NLST	CIFTP_NLST	12
CDUP	CIFTP_CDUP	13

See Also

MethodState

PutFile Event

Applies To

CIFTP

Purpose

The PutFile event signals the successful completion of the PutFile method.

Syntax

```
Sub CIFTP_PutFile()
```

Comments

The PutFile event fires when the PutFile high-level method successfully completes. The InternetError event fires when the PutFile high-level method fails.

See Also

InternetError, PutFile

SocketClosed Event

Applies To

CIFTP

Purpose

The SocketClosed event fires when the socket closes.

Syntax

```
Sub CIFTP_SocketClosed()
```

Comments

The socket should remain open during program execution. If the SocketClosed event fires while your application is running, an error condition has occurred.

See Also

CleanupConnection

TotalFileBytesReceived Event

Applies To

CIFTP

Purpose

The TotalFileBytesReceived event fires repeatedly as packets are received from the server.

Syntax

```
Sub CIFTP_TotalFileBytesReceived(ByVal bytes_in As Long)
```

Comments

The *bytes_in* value is the total number of bytes received during the file transfer. It increments each time a packet is received. Use *bytes_in* as the byte counter for the file transfer.

See Also

FileClosed, LIST, RETR

WSAError Event

Applies To

CIFTP

Purpose

The WSAError event fires when CIFTP receives a WINSOCK error.

Syntax

```
Sub CIFTP_WSAError(ByVal error_number As Integer)
```

Comments

Use the WSAError event to monitor the WINSOCK activity. The *error_number* identifies the WINSOCK error that occurred. These errors are listed in the CITPAK.BAS file.

See Also

CITPAK.BAS

CDUP Method

Applies To

CIFTP

Purpose

The CDUP method requests that the server change the current working directory to its parent directory.

Syntax

CIFTP.CDUP

Data Type

None

Comments

If the current directory is the root directory, this command has no effect.

See Also

CWD

CleanupConnection Method

Applies To

CIFTP

Purpose

The CleanupConnection method closes the socket and cleans up WINSOCK.

Syntax

CIFTP.CleanupConnection

Data Type

None

Comments

By default CIFTP opens the socket when you invoke the ConnectToAccessChannel method and the ConnectToDataControlChannel method. It closes the socket when you issue the QUIT method.

Use CleanupConnection to clear errors that might occur when a method fails, or in situations when the socket does not close properly, and you need to reconnect to the server. The CleanupConnection method causes the SocketClosed event to fire.

Use this method when all else fails.

See Also

ConnectToAccessControlChannel, QUIT, WSAError, SocketClosed

ConnectToAccessControlChannel Method

Applies To

CIFTP

Purpose

The ConnectToAccessControlChannel method establishes a connection to the FTP servers access control channel. The access control channel passes requests and replies (but no data) between client and server.

Syntax

```
nResult% = CIFTP.ConnectToAccessControlChannel
```

Data Type

Integer

Comments

Use the ConnectToAccessControlChannel method to request a connection to an FTP server. When the method succeeds, the AccessControlChannelConnection event fires, and your application can start to interact with the FTP server. You cannot retrieve data from an FTP server until you also establish a data channel connection with the ConnectToDataChannel method.

To establish a connection you must set the HostName **or** HostAddress and the AccessPort properties. When the connection attempt succeeds, *nResult* is an integer that represents the socket number opened by the ConnectToAccessControlChannel method. The return of a socket number does not indicate successful connection; rather the AccessControlChannelConnection event fires when the connection attempts succeeds. When the connection attempt fails, *nResult* is 0 and the WSAError event fires.

See Also

AccessControlChannelConnection, AccessPort, ConnectToDataChannel, PASV

ConnectToDataChannel Method

Applies To

CIFTP

Purpose

The ConnectToDataChannel method establishes a connection to a data channel. The data control channel passes data between client and server.

Syntax

```
nResult = CIFTP.ConnectToDataChannel
```

Data Type

Integer

Comments

Use the ConnectToDataChannel method to request a data connection to an FTP server.

You cannot establish a data channel connection until you first establish an access control channel (ConnectToAccessControlChannel method) and receive a port number (DataPort property) for the data access channel to use from the FTP server (PASV method).

To establish a connection you must set the HostName **or** HostAddress property. When the ConnectToDataChannel method succeeds, *nResult* is an integer that represents the socket number opened by the ConnectToDataControlChannel method. The return of a socket number does not indicate successful connection; rather the DataChannelConnection event fires when the connection attempts succeeds. When the connection attempt fails, *nResult* is 0 and the WSAError event fires.

See Also

AccessControlChannelConnection, AccessPort, ConnectToAccessControlChannel, DataPort, PASV

CWD Method

Applies To

CIFTP

Purpose

The CWD method requests that the server change the current working directory.

Syntax

CITFP.CWD

Data Type

None

Comments

Set the WorkingDirectory property before calling the CWD method. The following code sample illustrates how you might verify that the FTP server was able to change to the new directory

```
NewDir$ = "/pub/crescent"
With CIFTP1
   .WorkingDirectory = NewDir$
   .CWD
   .WorkingDirectory = ""
   .PWD
   Do While .WorkingDirectory = ""
      DoEvents
   Loop
End With
```

If WorkingDirectory = NewDir$, you've succeeded, otherwise the server could not change to the directory in NewDir$.

See Also

CWD, WorkingDirectory

GetDirectory Method

Applies To

CIFTP

Purpose

The GetDirectory Method obtains a directory listing from an FTP server.

Syntax

CITFP.GetDirectory

Data Type

Integer

Comments

When GetDirectory succeeds, the GotDirectory event fires. When it fails the InternetError event fires.

GetDirectory encapsulates these five CIFTP methods required to obtain a directory listing (in this order): ConnectToAccessControlChannel, USER, PASS, LIST, and QUIT.

NOTE If any of the five encapsulated methods fails, then GetDirectory also fails.

Set these properties before calling GetDirectory: AccessPort, FileListBoxName, DirectoryListBoxName, HostName (or HostAddress), LocalFileName, LoginName, Password, RemoteFileName, RepresentationType, ServerOSType, WorkingDirectory. GetDirectory populates the FileListBox and the DirectoryListBox objects. Note that the events for the encapsulated methods (ConnectToAccessControlChannel, USER, PASS, LIST, and QUIT) do not fire.

See Also

AccessPort,ConnectToAccessControlChannel, DirectoryListBoxName, FileListBoxName, GotDirectory, HostAddress, HostName InternetError, LIST, LocalFileName, LoginName, Password, PWD, QUIT, RemoteFileName, RepresentationType, ServerOSType, USER,WorkingDirectory

GetFile Method

Applies To

CIFTP

Purpose

The GetFile method retrieves a file from an FTP server.

Syntax

CITFP.GetFile

Data Type

None

Comments

When GetFile succeeds, the GotFile event fires. When it fails, the InternetError event fires.

GetFile encapsulates these nine CIFTP methods required to obtain a file(in this order): ConnectToAccessControlChannel, USER, PASS, PASV, ConnectToDataAccessChannel, SYST, TYPE, RETR, QUIT.

NOTE If any of the nine encapsulated methods fails, then GetFile also fails.

Set these properties before calling GetFile: AccessPort, HostName (or HostAddress), LocalFileName, LoginName, Password, RemoteFileName, RepresentationType, ServerOSType, WorkingDirectory. Note that the events for the encapsulated methods (ConnectToAccessControlChannel, USER, PASS, PASV, ConnectToDataAccessChannel, SYST, TYPE, RETR, QUIT) do not fire.

See Also

AccessPort, ConnectToAccessControlChannel, ConnectToDataChannel, GotFile, HostAddress, HostName InternetError, LocalFileName, LoginName, PASS, Password, PASV, QUIT, RemoteFileName, RepresentationType, RETR, ServerOSType, SYST, TYPE, USER, WorkingDirectory

LIST Method

Applies To

CIFTP

Purpose

The LIST method requests a directory listing of the current working directory from the FTP server.

Syntax

CITFP.LIST

Data Type

None

Comments

Set the DirectoryListBoxName property and the FileListBoxName property before calling LIST if you want the results of the request to populate a ListBox object. The working directory is the directory to which you are currently logged in.

See Also

CDUP, CWD, DirectoryListBoxName

PASS Method

Applies To

CIFTP

Purpose

The PASS method is part of the login procedure along with the USER method. It provides the FTP server with the logins password.

Syntax

CIFTP.PASS

Data Type

None

Comments

Set the Password property before calling the PASS method. Call the PASS method immediately after USER. The combination of USER and PASS identifies the user to the FTP host server.

See Also

LoginName, Password, USER

PASV Method

Applies To

CIFTP

Purpose

The PASV method requests a data port from the FTP server.

Syntax

CIFTP.PASV

Data Type

Integer

Comments

PASV updates the DataPort property. It is this port that the client and FTP server will use for the data control channel. When the DataPortSet event fires, the DataPort property has been updated.

See Also

ConnectToDataChannel, DataPortSet

PutFile Method

Applies To

CIFTP

Purpose

The PutFile method sends a file to an FTP server.

Syntax

CIFTP.PutFile

Data Type

Integer

Comments

When PutFile succeeds, the PutFile event fires. When it fails, the InternetError event fires.

PutFile encapsulates these nine CIFTP methods required to obtain a file (in this order): ConnectToAccessControlChannel, USER, PASS, PASV, ConnectToDataAccessChannel, SYST, TYPE, STOR, QUIT.

NOTE If any of the nine encapsulated methods fails, then PutFile also fails.

Set these properties before calling GetFile: AccessPort, HostName (or HostAddress), LocalFileName, LoginName, Password, RemoteFileName, RepresentationType, ServerOSType, WorkingDirectory. Note that the events for these methods do not fire.

See Also

AccessPort, ConnectToAccessControlChannel, ConnectToDataChannel, HostAddress, HostName, InternetError, LocalFileName, LoginName, PASS, Password, PASV, PutFile, QUIT, RemoteFileName, RepresentationType, RETR, ServerOSType, STOR, SYST, TYPE, USER, WorkingDirectory

PWD Method

Applies To

CIFTP

Purpose

The PWD method requests that the server print the current working directory.

Syntax

CIFTP.PWD

Data Type

None

Comments

PWD updates the WorkingDirectory property.

See Also

WorkingDirectory

QUIT Method

Applies To

CIFTP

Purpose

The QUIT method requests that the connection to the FTP server be terminated.

Syntax

CIFTP.QUIT

Data Type

None

Comments

QUIT notifies the FTP server that the client is done. and the server closes the access control channel connection.

If a file transfer is in progress, the FTP server waits until it is complete and then closes the channel.

See Also

ConnectToAccessControlChannel

RETR Method

Applies To

CIFTP

Purpose

The RETR method requests that the server copy a file from the FTP server to the client.

Syntax

CIFTP.RETR

Data Type

None

Comments

Set the RemoteFilename property before calling the RETR method. RETR updates the file identified by the LocalFileName property. If the file does not exist, or the user does not have privileges to the file, the server will not return any data to the client and the DataControlPacketReceived event will not fire.

See Also

DataControlPacketReceived, LocalFileName, RemoteFileName

SendFTPCommand Method

Applies To

CIFTP

Purpose

The SendFTPCommand method requests that the server issue FTP commands. Use this method to issue commands not implemented as CIFTP methods.

Syntax

```
SendFTPCommand(Packet As String)
```

Data Type

String

Comments

The *Packet* is the FTP command that you want to issue. It has this format:
FTPCommand

Where *FTPCommand* is the actual command that you want to issue. If the FTP command that you issue takes parameters, the command and parameters must be separated by a space. When you call the SendFTPCommand method the PacketSent event fires.

See Also

ConnectToAccessControlChannel, ConnectToDataChannel

STOR Method

Applies To

CIFTP

Purpose

The STOR method requests that the FTP server accept a file from the client.

Syntax

CIFTP.STOR

Data Type

None

Comments

STOR copies the file identified by the LocalFileName propety to the FTP server. The FTP server stores it in the location specified by the RemoteFileName property.

If the file already exists, the FTP server overwrites the existing file with the new file. If the file does not exist, it is created.

NOTE You might not have access privileges to deposit files on to a particular FTP server.

See Also

LocalFileName, RemoteFileName

SYST Method

Applies To

CIFTP

Purpose

The SYST method asks the FTP server to identify its operating system, for example, Windows NT or UNIX.

Syntax

CIFTP.SYST

Data Type

None

Comments

SYST updates the ServerOSType property. The ServerOSType identifies for the control, the format of the incoming directory or file listing.

See Also

DirectoryListBoxName, FileListBoxName, ServerOSType

TYPE Method

Applies To

CIFTP

Purpose

The TYPE method requests that the server translate the data into a specified format before passing it to the client.

Syntax

CIFTP.TYPE

Data Type

None

Comments

Set the RepresentationType property before calling the TYPE method. RepresentationType has these values:

Use this value...	to retrieve this kind of data...
0	ASCII
1	Binary

See Also

RepresentationType

USER Method

Applies To

CIFTP

Purpose

The USER method is part of the login procedure along with the PASS method. It provides the FTP server with the logins username.

Syntax

CIFTP.USER

Data Type

Integer

Comments

Set the LoginName property before calling the USER method.

Call USER after the AccessControlChannelConnection event fires. Call PASS immediately after USER. The combination of USER and PASS identifies the user to the FTP host server.

NOTE Your applications users must have an account on the FTP server with the the appropriate access level to the files that the application accesses. Some FTP servers support a user name of *anonymous*. Anyone can login as anonymous, but they might have limited privileges.

See Also

USER, LoginName, Password

General Tab

Applies To

CIFTP

This custom tab lets you set these custom properties:

FTP Configuration:

Access Port	Data Port	Host Name
Host Address	Login Name	Password

Transaction Related:

Local File	Remote File	Representation Type
FTP Server OS	Working Directory	

SoundEvents Tab

Applies To

CIFTP

This custom tab lets you associate sounds to custom events. The following table lists the hidden properties that define the sound object and the event to which they are applied:

Event	Hidden Property
AccessControlChannelConnection	AccessControlChannelWAV
AccessChannelClosed	AccessChannelClosedWAV
DataChannelConnection	DataChannelConnectionWAV
DataChannelClosed	DataChannelClosedWAV
FileClosed	FileClosedWAV
ListBoxesPopulated	ListBoxesPopulatedWAV
SocketClosed	SocketClosedWAV
WSAError	WSAErrorEventWAV

You can set the sound associations at design time or runtime. You can test the sounds from within the property page. At runtime, an event is fired and a sound is played asynchronously where an association is made.

To associate a sound, you can type in the full qualified path for any events you want to associate or click the Browse... button and select it from the dialog box. After you enter a sound, you can click the test button to hear it. If the sound file is invalid or missing you will hear nothing when you click Test.