

TFTP Client ActiveX Control

**Trivial File Transfer Protocol
Custom Control
for Microsoft® Windows™**

Version 5.2

**Copyright © 1995 - 2003 by Distinct Corporation
All rights reserved**

Table of Contents

| | |
|------------------------------|----|
| Table of Contents..... | 1 |
| 5.1 Overview | 3 |
| 5.1.1 Introduction..... | 3 |
| 5.1.2 Usage..... | 3 |
| 5.1.3 Property Summary..... | 4 |
| 5.1.4 Event Summary..... | 4 |
| 5.1.5 Method Summary..... | 4 |
| 5.1.6 D_TFTP.TXT..... | 4 |
| 5.2 Properties | 7 |
| 5.2.1 Action..... | 7 |
| 5.2.2 Attempts..... | 9 |
| 5.2.3 BlkSize..... | 10 |
| 5.2.4 Error..... | 11 |
| 5.2.5 Host..... | 13 |
| 5.2.6 LocalFile..... | 14 |
| 5.2.7 NegotiateFileSize..... | 15 |
| 5.2.8 RemoteFile..... | 16 |
| 5.2.9 Timeout..... | 17 |
| 5.2.10 TransferType..... | 18 |
| 5.3 Events | 19 |
| 5.3.1 OnError..... | 19 |
| 5.3.2 OnTransfer..... | 21 |
| 5.4 Methods | 23 |
| 5.4.1 Abort..... | 23 |
| 5.4.2 Ascii..... | 24 |
| 5.4.3 Binary..... | 25 |
| 5.4.4 Get..... | 26 |
| 5.4.5 Put..... | 27 |

5.1 Overview

5.1.1 Introduction

The Distinct Trivial File Transfer Protocol (TFTP) Client ActiveX allows you to integrate TFTP based file transfer capabilities into your applications. TFTP is a UDP based protocol and uses minimal resources. An application may use TFTP when it simply wants to download a known file from a server, where access is guaranteed and the location is known, with minimal use of resources.

5.1.2 Usage

See the section entitled "Using Distinct ActiveX controls in various environments" on how to add the control to your project.

After placing a TFTP Client ActiveX control into a form, some properties can be set at design time. Certain configuration options controlling the type of file transfer or the timeout may never change in a session and need therefore only be set once at design time. The TransferType property defaults to TRANSFER_TYPE_ASCII and can be changed to TRANSFER_TYPE_BINARY. The Timeout property specifies the timeout value in seconds for a file transfer. The Attempts property specifies the number of attempts the TFTP ActiveX control should make for a file transfer before it returns with a failure.

The Ascii method can be used instead of setting the TransferType property to TRANSFER_TYPE_ASCII. Similarly, the Binary method is equivalent to setting the TransferType property to TRANSFER_TYPE_BINARY.

The size of each block of data can be specified by setting the BlkSize property to the desired packet size. The default packet size is 512 octets. The file size can be determined by setting the NegotiateFileSize property to TRUE.

The Host property is usually set at run time right before a session is established. This allows the application to request the host name or internet address of the TFTP server from the user. If an application will always connect to the same host, then the Host property can also be set at design time to minimize user interaction.

The Action property can only be accessed at run time. The RemoteFile and LocalFile properties must be set before attempting a file transfer. To get a remote file and store it locally, the value of the Action property must be set to ACTION_TFTP_GET (the Get method can also be used). If the remote file exists, it is transferred and stored in the local disk with the file name specified in the LocalFile property. To put a local file and store it remotely, the value of the Action property must be set to ACTION_TFTP_PUT (the Put method can also be used). If the local file exists, it is transferred and stored in the remote directory with the file name specified in the RemoteFile property. The file transfer can be aborted at any time by setting the Action property to ACTION_TFTP_ABORT.

The Error property will contain the error code returned for an unsuccessful file transfer. The value of this property can be checked after a get or put operation to ensure no error has occurred. During a file transfer, one or more OnTransfer event will occur to inform the application of the status of the file transfer. This information is useful if the application wishes to display the progress of the file transfer.

5.1.3 Property Summary

Action

Get or put a file

Attempts

Number of attempts for file transfer

BlkSize

The size of each block of data

Error

Error code returned during file transfer

Host

Name of server or dotted decimal internet address

LocalFile

Path and name of local file to transfer

NegotiateFileSize

Check the size of the file before doing a Get or a Put.

RemoteFile

Name of remote file to transfer

Timeout

Timeout value for file transfer

TransferType

Select ASCII or binary transfer

5.1.4 Event Summary

OnError

Local error has occurred

OnTransfer

Number of bytes transferred

5.1.5 Method Summary

Abort

Abort file transfer

Ascii

Set file transfer to ASCII

Binary

Set file transfer to binary

Get

Retrieve file from server

Put

Upload file to server

5.1.6 D_TFTP.TXT

The following provides a complete listing of the D_TFTP.TXT definition file. If your application uses more than one Distinct ActiveX control in the same form, then some definitions will conflict. For example, the FTP Client ActiveX control includes the definition

```
Global Const ACTION_DISCONNECT = 3
```

in the D_FTP.TXT file and the Telnet ActiveX control includes the definition

```
Global Const ACTION_DISCONNECT = 2
```

in the D_TNET.TXT file. To avoid this conflict, you must rename at least one of the constants (for example, FTP_ACTION_DISCONNECT or TNET_ACTION_DISCONNECT).

```
' TFTP Client ActiveX Control
' (C) Copyright 1995 - 1997 by Distinct Corporation
' All rights reserved
```

```
' actions
```

```
Global Const ACTION_NONE = 0
Global Const ACTION_TFTP_GET = 1
Global Const ACTION_TFTP_PUT = 2
Global Const ACTION_TFTP_ABORT = 3
```

```
' modes
```

```
Global Const TRANSFER_TYPE_ASCII = 0
Global Const TRANSFER_TYPE_BINARY = 1
```

```
' return values
```

```
Global Const TFTP_FAILURE = -1
Global Const TFTP_GLOBALALLOC_FAILED = -2
Global Const TFTP_ILLEGAL_REQUEST = -3
Global Const TFTP_SOCKET_FAILURE = -4
Global Const TFTP_BIND_FAILED = -6
Global Const TFTP_ILLEGAL_FILE_MODE = -9
Global Const TFTP_FILE_ERROR = -10
Global Const TFTP_PARTIAL_FILE = -12
Global Const TFTP_SENDTO_ERR = -13
Global Const TFTP_READ_ERROR = -16
Global Const TFTP_CANNOT_RESOLVE_HOSTNAME = -17
Global Const TFTP_CANNOT_NEGOTIATE_OPTION = -18L
```

```
' error codes
```

```
Global Const ERR_CANNOT_CHANGE_XFER_TYPE = 1
Global Const ERR_HOST_UNDEFINED = 2
Global Const ERR_CANNOT_CONNECT = 3
Global Const ERR_CANNOT_DISCONNECT = 4
Global Const ERR_IN_ACTION = 5
Global Const ERR_NO_REMOTE_FILE = 6
Global Const ERR_NO_LOCAL_FILE = 7
Global Const ERR_CANNOT_GET = 8
Global Const ERR_CANNOT_PUT = 9
Global Const ERR_UNABLE_TO_LOAD = 10
```


5.2 Properties

5.2.1 Action

Summary

Get or put a file.

Description

The Action property controls the file transfer state of the TFTP Client ActiveX control. A TFTP get or put of a file can be performed by assigning one of the following values to the property.

| Value | Meaning |
|-------------------|----------------------------------|
| ACTION_TFTP_GET | Get a file from remote to local. |
| ACTION_TFTP_PUT | Put a file from local to remote. |
| ACTION_TFTP_ABORT | Abort file transfer in progress |

This property can be changed at run time only.

Before setting the Action property to ACTION_TFTP_GET or ACTION_TFTP_PUT, the following properties must be initialized. The Host property must be set to the name or internet address (in dotted decimal notation) of the TFTP server. The LocalFile property must be set to the name of the file on the local disk. The RemoteFile property must be set to the name of the file on the remote server. The TransferType property must be set to ASCII or binary. Most applications will prompt the user with a dialog box for the information to place into these properties, but all of them can also be specified at design time. The application can set the BlkSize property to the desired data block size (valid range is between 8 and 65464), this is useful if the file is too big. NegotiateFileSize property can be set to TRUE if the application wants the TFTP ActiveX control to determine the file size before actually receiving the file or send the file size to the server while doing a put request.

Setting the Action property to ACTION_TFTP_ABORT will abort any get or put operation in progress.

If the file was not transferred, then the OnError event will be fired. This event will occur before the next statement (i.e. the statement following the assignment of ACTION_TFTP_GET or ACTION_TFTP_PUT to the Action property) is executed. The application should set a flag in the OnError event, so that it can determine if the file has been transferred or not. In addition, the application may want to display an error message in the OnError event to inform the user that the file has not been transferred. Please check the reference page of the OnError event for a complete listing of error codes.

The Error property will contain the error code returned for an unsuccessful file transfer. The value of this property can be checked after a get or put operation to ensure no error has occurred.

During a file transfer, one or more OnTransfer events will occur to notify the application of the status of the file transfer (number of bytes copied). The application can use this information to display the progress of the file transfer.

The Get, Put and Abort methods accomplish the same as the above actions. Please check the reference pages of these methods for more detailed information on their usage.

There is no default value for this property.

Example

```
TFTPClient.Host = "speedy.distinct.com"  
TFTPClient.LocalFile = "c:\test.exe"  
TFTPClient.RemoteFile = "test.exe"  
TFTPClient.Action = ACTION_TFTP_GET
```

5.2.2 Attempts

Summary

Number of attempts for file transfer.

Description

The Attempts property indicates the number of times the TFTP client ActiveX control should try to get or put a file to the TFTP server before it fails and returns an error. Generally, the default value of 5 should be enough, but the application can customize it.

This property must be set before setting the Action property to ACTION_TFTP_GET or ACTION_TFTP_PUT (or the Get or Put method must be called). Because communication links or servers sometimes go down, the TFTP Client ActiveX control uses a default timeout of 5 seconds and a default of 5 attempts before returning an error upon unsuccessful file transferred.

This property can be read or written to at any time. The default value for this property is 5.

Example

```
TFTPClient.Attempts = 10
```

5.2.3 BlkSize

Summary

The size of each data block.

Description

The BlkSize property indicates the size of each data block that the TFTP ActiveX control will send or receive from the TFTP server. The valid range of BlkSize is between 8 and 65464 octets. The default size of each packet is 512.

This property must be set before setting the Action property to ACTION_TFTP_GET or ACTION_TFTP_PUT (or the Get or Put method must be called). If the server refuses to accept the specified BlkSize the Get or Put method will return an error.

This property can be read or written to at any time. The default value for this property is 512.

Example

```
TFTPClient.BlkSize = 1024
```

5.2.4 Error

Summary

Error code returned during file transfer.

Description

The Error property specifies the error code returned during an unsuccessful file transfer operation. On failure, the Error property reflects the error code of the last file transfer operation initiated by setting the Action property to ACTION_TFTP_GET or ACTION_TFTP_PUT (or by calling the Get or Put method).

The value of this property must be checked immediately after initiating a file transfer operation. Initiating other file transfer operations may change the value of this property.

This property is a read only property and can be read at any time. There is no default value for this property.

The following is a list of possible error codes returned by the TFTP Client ActiveX control after a file transfer operation.

| Constant | Meaning |
|------------------------------|---|
| TFTP_FAILURE | Unable to setup socket. |
| TFTP_GLOBALALLOC_FAILED | Out of memory. |
| TFTP_ILLEGAL_REQUEST | Unable to build request packet. |
| TFTP_SOCKET_FAILURE | Unable to open socket. |
| TFTP_BIND_FAILED | Unable to bind socket. |
| TFTP_ILLEGAL_FILE_MODE | Illegal file mode. |
| TFTP_FILE_ERROR | Part of file was transferred. |
| TFTP_SENDTO_ERR | Unable to send data to server. |
| TFTP_READ_ERROR | Unable to read file. |
| TFTP_CANNOT_RESOLVE_HOSTNAME | Cannot resolve host name. |
| TFTP_CANNOT_NEGOTIATE_OPTION | Cannot negotiate the BlkSize option with the server |

The following describes each error in more detail.

TFTP_FAILURE

An error occurred during a call to setup socket options or the TFTP server did not allow file creation.

TFTP_GLOBALALLOC_FAILED

An error occurred during a call to allocate memory.

TFTP_ILLEGAL_REQUEST

An attempt was made to build a file transfer request packet, but failed.

TFTP_SOCKET_FAILURE

Unable to open the socket.

TFTP_BIND_FAILED

An error occurred during a call to bind to local address.

TFTP_ILLEGAL_FILE_MODE

The requested file transfer type is invalid. The TransferType property must be set to TRANSFER_TYPE_ASCII or TRANSFER_TYPE_BINARY (or the Ascii or Binary method must be called).

TFTP_FILE_ERROR

Unable to open the specified file for read or write.

TFTP_PARTIAL_FILE

Only part of the file was transferred.

TFTP_SENDTO_ERR

An error occurred during the send operation.

TFTP_READ_ERROR

Cannot read the file.

TFTP_CANNOT_RESOLVE_HOSTNAME

Unable to resolve host name. The name you have used for the Host property is not an official host name or alias, or it cannot be found in the database(s) being queried.

TFTP_CANNOT_NEGOTIATE_OPTION

Unable to negotiate the BlkSize for each data block with the server or the BlkSize is out of range.

Example

```
Const TFTP_CANNOT_RESOLVE_HOSTNAME = -17

TFTPClient.Host = "santa.distinct.com"
TFTPClient.LocalFile = "c:\test.exe"
TFTPClient.RemoteFile = "test.exe"
TFTPClient.Action = ACTION_TFTP_GET
If TFTPClient.Error = TFTP_CANNOT_RESOLVE_HOSTNAME Then
    MsgBox "Invalid host name specified", 64, "TFTP Client Sample"
End If
```

5.2.5 Host

Summary

Name of server or dotted decimal internet address.

Description

The Host property specifies the name or internet address of an TFTP server. This property must be set before a session can be established. There are three possible ways of specifying an TFTP server.

Machine Name

An application only needs to specify the name of the TFTP server if the server is located on the same network as the local PC or if the internet address of the server is defined in the local host table. If the TFTP server is not on the local network, then the underlying protocol will route the traffic through a gateway. If the TFTP server is not defined in the local host table, then the underlying protocol will contact the domain server to resolve the internet address of the server.

Machine and Domain Name

An application needs to specify the machine name and the domain name if the TFTP server is not located on the same network as the local PC. Fully domain qualified machine names are written from left to right in ascending order (for example, *speedy.distinct.com*). If both machine and domain names are specified, then the underlying protocol will contact the domain server to resolve the internet address of the server.

Internet Address

Sometimes the user knows only the internet address of the TFTP server that he or she wants to use. In this case, the internet address can be entered in what is known as the dotted decimal notation (for example, *127.43.101.12*). If the TFTP server identified by this address is not on the local network, then the underlying protocol will route the traffic through a gateway.

This property can be changed at design time and at run time before a connection has been established. There is no default value for this property.

Example

```
TFTPClient.Host = "127.43.101.12"  
TFTPClient.LocalFile = "c:\test.exe"  
TFTPClient.RemoteFile = "test.exe"  
TFTPClient.Action = ACTION_TFTP_GET
```

5.2.6 LocalFile

Summary

Path and name of local file to transfer.

Description

The LocalFile property specifies the name of the local file to be used during a file transfer. This property must be set before a file transfer. File transfers are initiated by setting the Action property to ACTION_TFTP_GET or ACTION_TFTP_PUT.

The local file must be specified with a fully qualified path (such as "C:\MYDIR\TEST.TXT"). Since there is no local current working directory, a file specified without a path may reside in the Windows directory, in the directory from which the application was started or in the directory defined as the application's working directory.

This property can be changed at any time except during a file transfer. There is no default value for this property.

Example

```
TFTPClient.RemoteFile = "test.exe"  
TFTPClient.LocalFile = "c:\test.exe"  
TFTPClient.Action = ACTION_TFTP_GET
```

5.2.7 NegotiateFileSize

Summary

Negotiate the file size before doing a Get or a Put operation.

Description

Setting this property to TRUE allows the side receiving the file to determine the ultimate size of the file. If this property is set to TRUE the TFTP ActiveX control will add a file size request option to the read or write request packet. This is useful when the file is too large and there is not enough disk space.

Setting the property to TRUE will affect the Get (ACTION_TFTP_GET) method and the Put (ACTION_TFTP_PUT) method in the following way.

During a Get operation the TFTP ActiveX control will request the size of the file that it is going to get, if the file is too big and there is not enough disk space the TFTP ActiveX control will return an error to the server. Similarly during a Put operation the TFTP ActiveX control will send the size of the file in the request packet, if the file is too large for the server to handle it will return an error. Note that if the server does not support this file size negotiation option then the TFTP ActiveX control will not check the available disk space before getting the file.

This property can be changed at any time except during a file transfer. The default value for this property is FALSE.

Example

```
TFTPClient.NegotiateFileSize = TRUE
```

5.2.8 RemoteFile

Summary

Name of remote file to transfer.

Description

The RemoteFile property specifies the name of the remote file to be used during a file transfer. This property must be set before a file transfer. File transfers are initiated by setting the Action property to ACTION_TFTP_GET or ACTION_TFTP_PUT.

This property must be set to the name of a file in the current remote working directory for both get and put operations.

This property can be changed at any time except during a file transfer. There is no default value for this property.

Example

```
TFTPClient.RemoteFile = "test.exe"  
TFTPClient.LocalFile = "c:\test.exe"  
TFTPClient.Action = ACTION_TFTP_GET
```

5.2.9 Timeout

Summary

Timeout value for file transfer.

Description

The Timeout property specifies the get or put timeout period. The application can change the Timeout property, which specifies the timeout value in seconds for the get or put action. The application can change the value of this property at design time or at run time. The TFTP Client ActiveX control will timeout after the specified period before returning an error to the application.

This property must be set before setting the Action property to ACTION_TFTP_GET or ACTION_TFTP_PUT (or before calling the Get or Put method). Because communication links or servers sometimes go down, the TFTP Client ActiveX control uses a default timeout of 5 seconds and a default of 5 attempts before returning a error upon unsuccessful file transferred.

The application must specify the new timeout value in seconds. The value must be in the range of 1 to 999 seconds.

This property can be changed at design time or at run time. The default value for this property is 5.

Example

```
TFTPClient.Timeout = 10
```

5.2.10 TransferType

Summary

Select ASCII or binary transfer.

Description

The TransferType property determines if any type of file conversion will be performed by the server during file transfers. The value of this property can be set to either one of the following two constants.

| Value | Meaning |
|----------------------|---|
| TRANSFER_TYPE_ASCII | Convert CR and LF to LF and vice versa. |
| TRANSFER_TYPE_BINARY | Transfer files without conversion. |

When the TransferType property is set to TRANSFER_TYPE_BINARY, then no file conversion is performed. Files are sent and received exactly as they are saved on the originating machine. This allows undisturbed transfers of non-text files, such as compressed files, archives and executables.

When the TransferType property is set to TRANSFER_TYPE_ASCII, then carriage return and line feed pairs are converted to line feeds during put operations and line feeds are expanded into carriage return and line feed pairs during get operations. The ASCII mode conversion is normally used to convert text files to a format suitable for text editors on the destination machine.

The Ascii and Binary methods can also be used to set the transfer type. Please check the reference pages of the methods for more detailed information on their usage.

This property can be changed at any time except during a file transfer. The default value for this property is TRANSFER_TYPE_ASCII.

Example

```
TFTPClient.RemoteFile = "test.exe"
TFTPClient.LocalFile = "c:\test.exe"
TFTPClient.TransferType = TRANSFER_TYPE_BINARY
TFTPClient.Action = ACTION_TFTP_GET
```

5.3 Events

5.3.1 OnError

Summary

Local error has occurred.

Description

The OnError event occurs when a property is accessed in an illegal way or when a connection with the TFTP server cannot be established. The following describes all possible error codes delivered by this event.

| Value | Meaning |
|-----------------------------|---|
| ERR_CANNOT_CHANGE_XFER_TYPE | Cannot change transfer type during file transfer. |
| ERR_HOST_UNDEFINED | Remote host must be defined before connecting. |
| ERR_CANNOT_CONNECT | Unable to connect. |
| ERR_CANNOT_DISCONNECT | Unable to disconnect. |
| ERR_IN_ACTION | Another action is in progress. |
| ERR_NO_REMOTE_FILE | Remote file not specified. |
| ERR_NO_LOCAL_FILE | Local file not specified. |
| ERR_CANNOT_GET | Unable to get file. |
| ERR_CANNOT_PUT | Unable to put file. |

The following describes each error in more detail.

ERR_CANNOT_CHANGE_XFER_TYPE

Transfer type cannot be changed during a file transfer (by setting the Action property to ACTION_TFTP_GET or ACTION_TFTP_PUT or by calling the Get or Put method).

ERR_HOST_UNDEFINED

Host must be defined before transferring files. Set the Host property before setting the Action property to ACTION_TFTP_GET or ACTION_TFTP_PUT.

ERR_CANNOT_CONNECT

TFTP server is unreachable. The Host property may be set incorrectly or the host may be down.

ERR_CANNOT_DISCONNECT

TFTP server is unreachable. The host may have gone down during file transfer.

ERR_IN_ACTION

Another action is already in progress.

ERR_NO_REMOTE_FILE

Remote file is not defined. The RemoteFile property must be set.

ERR_NO_LOCAL_FILE

Local file is not defined. The LocalFile property must be set.

ERR_CANNOT_GET

An error has occurred during the get file transfer operation. Check the Error property for more information on the type of error.

ERR_CANNOT_PUT

An error has occurred during the put file transfer operation. Check the Error property for more information on the of error.

Example

```
Sub TFTPClient_OnError (ErrorCode As Integer)
  If ErrorCode = ERR_CANNOT_CONNECT Then
    MsgBox "Unable to connect to remote host", 64, "Sample Program"
  End If
End Sub
```

5.3.2 OnTransfer

Summary

Number of bytes transferred.

Description

The OnTransfer event occurs during a file transfer of data from the local machine to a remote file or from the remote machine to a local file. The event contains the number of bytes of data transferred between the two machines.

A get or put operation can be initiated by setting the Action property to ACTION_TFTP_GET or ACTION_TFTP_PUT (or by calling the Get or Put method). The local file must be specified with the LocalFile property and the remote file must be specified with the RemoteFile property. The TransferType property must be set TRANSFER_TYPE_ASCII or TRANSFER_TYPE_BINARY (or the Ascii or Binary method must be called).

As the file transfer proceeds, one or more OnTransfer events will occur to deliver the number of bytes transferred so far between the TFTP server and the client. The application can process this data and display the status of the file transfer (for example, update a progress bar or text control).

While handling the OnTransfer event, an application should not perform tasks which have the potential of requiring a lot of time to complete, such as generating a message box.

Example

```
Sub TFTPClient_OnTransfer (BytesCopied As Integer)
    Label.Caption = Str$(BytesCopied) & " transferred"
End Sub
```


5.4 Methods

5.4.1 Abort

Summary

Abort file transfer in progress.

Syntax

Boolean Abort ()

Description

The Abort method aborts a file transfer in progress. This method aborts any get or put operation in progress.

The Abort method takes no parameters and returns a boolean. If the get or put operation in progress is successfully aborted, then the method returns True; otherwise, it returns False. The application should ensure that the method was successfully executed by checking the return value.

Calling this method is equivalent to setting the Action property to ACTION_TFTP_ABORT.

Example

```
Result = TFTPClient.Abort ()  
If Result = False Then  
    MsgBox "Unable to abort file transfer", 64, "Sample Program"  
End If
```

5.4.2 Ascii

Summary

Set file transfer to ASCII.

Syntax

Boolean Ascii ()

Description

The Ascii method sets the file transfer type to ASCII. That is, the server converts carriage return (CR) and line feed (LF) pairs to line feeds during put operations and line feeds are expanded into carriage return and line feed pairs during get operations. The ASCII mode conversion is normally used to convert text files to a format suitable for text editors on the destination machine.

The Ascii method takes no parameters and returns a boolean. If the transfer type can be successfully set to ASCII type file transfer, then the method returns True; otherwise, it returns False. The application should ensure that the method was successfully executed by checking the return value.

Calling this method is equivalent setting the TransferType property to TRANSFER_TYPE_ASCII. This method can be called at any time except during a file transfer.

Example

```
Result = TFTPClient.Ascii ()  
If Result = False Then  
    MsgBox "Cannot set transfer type to ASCII", 64, "Sample Program"  
End If
```

5.4.3 Binary

Summary

Set file transfer to binary.

Syntax

Boolean Binary ()

Description

The Binary method sets the file transfer type to binary. During file transfer, no file conversion is performed. Files are sent and received exactly as they are saved on the originating machine. This allows undisturbed transfers of non-text files, such as compressed files, archives and executables.

The Binary method takes no parameters and returns a boolean. If the transfer type can be successfully set to binary type file transfer, then the method returns True; otherwise, it returns False. The application should ensure that the method was successfully executed by checking the return value.

Calling this method is equivalent to setting the TransferType property to TRANSFER_TYPE_BINARY. This method can be called at any time except during a file transfer.

Example

```
Result = TFTPClient.Binary ()  
If Result = False Then  
    MsgBox "Cannot set transfer type to binary", 64, "Sample Program"  
End If
```

5.4.4 Get

Summary

Retrieve file from server.

Syntax

Boolean Get (*Host*, *RemoteFile*, *LocalFile*)

| | |
|-------------------|--------|
| <i>Host</i> | String |
| <i>RemoteFile</i> | String |
| <i>LocalFile</i> | String |

Description

The Get method transfers a file from the remote machine and stores it locally.

The Get method takes a host name (*Host*), a remote file name (*RemoteFile*) and a local file name (*LocalFile*) as its parameters and returns a boolean. The host name must be set to the name or internet address (in dotted decimal notation) of the TFTP server. The *RemoteFile* must be set to the name of the file on the remote server to retrieve. The *LocalFile* must be set to a file name on the local disk. If the file was successfully transferred, then the method returns True; otherwise, it returns False. The application should ensure that the method was successfully executed by checking the return value.

The application can specify a block size of each data block being sent to the server by setting the BlkSize property before calling the Get method. The valid range of BlkSize is between 8 and 65464.

The TFTP ActiveX control will request the file size before doing a Get operation if the NegotiateFileSize property is set to TRUE. If the server supports the file size negotiation option it will send back the size of the file that the application is trying to get. If the file is too large and there is not enough disk space the TFTP ActiveX control will return an error. Note that this property must be set before calling the Get method or before setting the Action property to ACTION_TFTP_GET.

If the file was not transferred, then the OnError event will be fired. This event will occur before the next statement (i.e. the statement following call to the Get method) is executed. The application should set a flag in the OnError event, so that it can determine if the file has been transferred or not. In addition, the application may want to display an error message in the OnError event to inform the user that the file has not been transferred. Please check the reference page of the OnError event for a complete listing of error codes.

Calling this method is equivalent to setting the Action property to ACTION_TFTP_GET.

Example

```
Result = TFTPClient.Get ("speedy.distinct.com", "test.exe", "c:\test.exe")
If Result = False Then
    MsgBox "Cannot get file from server", 64, "Sample Program"
End If
```

5.4.5 Put

Summary

Upload file to server.

Syntax

Boolean Put (*Host*, *LocalFile*, *RemoteFile*)

| | |
|-------------------|--------|
| <i>Host</i> | String |
| <i>LocalFile</i> | String |
| <i>RemoteFile</i> | String |

Description

The Put method transfers a file from the local machine and stores it on the server.

The Put method takes a host name (*Host*), a local file name (*LocalFile*) and a remote file name (*RemoteFile*) as its parameters and returns a boolean. The host name must be set to the name or internet address (in dotted decimal notation) of the TFTP server. The *LocalFile* must be set to a file name on the local machine to upload. The *RemoteFile* must be set to the name of the file on the remote server. If the file was successfully transferred, then the method returns True; otherwise, it returns False. The application should ensure that the method was successfully executed by checking the return value.

The application can specify a block size of each data block being received from the server by setting the BlkSize property before calling the Put method. The valid range of BlkSize is between 8 and 65464.

The TFTP ActiveX control will send the file size along with the Put request if the NegotiateFileSize property is set to TRUE. If the server supports the file size negotiation option it will either accept the file size or send an error if the file is too large and there is not enough disk space. Note that this property must be set before calling the Put method or before setting the Action property to ACTION_TFTP_PUT.

If the file was not transferred, then the OnError event will be fired. This event will occur before the next statement (i.e. the statement following the call to the Put method) is executed. The application should set a flag in the OnError event, so that it can determine if the file has been transferred or not. In addition, the application may want to display an error message in the OnError event to inform the user that the file has not been transferred. Please check the reference page of the OnError event for a complete listing of error codes.

Calling this method is equivalent to setting the Action property to ACTION_TFTP_PUT.

Example

```
Result = TFTPClient.Put ("speedy.distinct.com", "c:\test.exe", "test.exe")
If Result = False Then
    MsgBox "Cannot put file to server", 64, "Sample Program"
End If
```

