TFTP ActiveX Control for Microsoft® Windows™

Copyright © Magneto Software
All rights reserved
4. Events ........................................................................................................................................ 11
   4.1. TftpCompleted .................................................................................................................... 11
   4.2. TftpQueryCompleted .......................................................................................................... 12

5. Examples ................................................................................................................................... 13
1. **Overview**

1.1. **Introduction**
The Magneto Software TFTP ActiveX control (sktftp.ocx) allows developers to integrate TFTP based file transfer capabilities into their 32-Bit or 64-Bit applications.

TFTP is a UDP based protocol that is simpler to use than the File Transfer Protocol (FTP) but less capable. An application may use TFTP to download/upload files where user authentication is not required and location is identified.

TFTP ActiveX control allows downloading and uploading multiple files simultaneously.

It is fully compliant with RFC 1350, RFC2347, RFC2348 and RFC2349

It can be used from any Windows development environment, including Visual Studio.

It comes with complete documentation, sample code, and working demo programs.

Additional information about TFTP Protocol can be found at these locations:

- RFC 1350 – THE TFTP PROTOCOL
- TFTP Option Extension
- TFTP Blocksize Option
- TFTP Timeout Interval and Transfer Size Options

1.2. **Usage**
TFTP ActiveX control is capable of performing multiple TFTP upload/download queries simultaneously.

1.3. **Interface Summary**

1.3.1. **_DSKTftp**

Specifies a collection of methods and properties to accomplish TFTP file transfer.

1.3.2. **_DSKTftpEvents**

Specifies a collection of events supported by TFTP control.

1.4. **Property Summary**

- **optBlockSize**
  Blocksize value for file transfer

- **optExtensionBlockSize**
  Toggle ‘Blocksize’ option

- **optExtensionTimeout**
  Toggle ‘Timeout’ option
**optExtensionTsize**
Toggle ‘Tsize’ option

**optMode**
Select ASCII or binary transfer

**optPort**
The UDP port where the remote TFTP server is listening

**optRetries**
Maximum number of retries for the data transfers

**optTimeout**
Timeout value for file transfer

1.5. Event Summary

**TftpCompleted**
Indicate that TFTP control has stopped processing all upload/download requests.

**TftpQueryCompleted**
Indicate that TFTP control has stopped processing single specific upload/download request.

1.6. Method Summary

**Get**
Download a file from the remote host

**Put**
Upload a file to the remote host

**TftpReset**
This method terminates any pending TFTP requests

**ResetTftpSettings**
Reset all SkTFTP settings back to default values.

**TftpGetStatus**
Use TFTP control default settings
2. **Properties**

2.1. **optBlocksize**

**Summary**

Blocksize value for file transfer

**Description**

This property determines the blocksize used during file transfer. The transfer time decreases with an increase in blocksize. The reason for the reduction in time is the reduction in the number of packets sent. For example, by increasing the blocksize from 512 octets to 1024 octets, not only is the number of data packets halved, but the number of acknowledgement packets is also halved (along with the number of times the data transmitter must wait for an ACK). A secondary effect is the efficiency gained by reducing the per-packet framing and processing overhead. This property can have any value between 8 and 65536. The default value for this property is 512 bytes.

2.2. **optExtensionBlockSize**

**Summary**

Toggles the Blocksize option.

**Summary**

The `optExtensionBlockSize` property toggles the Blocksize option in the TFTP Read Request or Write Request. By default the `optExtensionBlockSize` option is enable.

2.3. **optExtensionTimeout**

**Summary**

Toggles the Timeout option.

**Description**

The `optExtensionTimeout` property toggles the Timeout option in the TFTP Read Request or Write Request. By default the `optExtensionTimeout` option is enable.

2.4. **optExtensionTsize**
Summary
Toggles the Transfer Size option.

Description
The optExtensionTsize property toggles the tsize option in the TFTP Read Request or Write Request. By default the optExtensionTsize option is enable.

2.5. optMode

Summary
Select ASCII or binary transfer

Description
This property determines if the server will perform any type of file conversion during file transfers. This property has two possible values:

1 – Binary (default)
0 – ASCII

This property defaults to binary transfers.
FTP servers require that you specify the type of data you expect to be in the files you're transferring back and forth.

When the optMode property is set to 1 (Binary), then no file conversion is performed. This allows undisturbed transfers of non-text files, such as compressed files, archives and executables.

When the optMode property is set to 0 (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.

Unless you're sure that you will be transferring ASCII-only files, it is best to switch optMode to 1 (Binary) prior to any file transfers.

2.6. optPort

Summary
The UDP port where the remote TFTP server is listening
Description
This property can be changed at design time or at run time, but must be set before calling Get or Put method. The default value for this property is 69.

2.7. optRetries

Summary
Maximum number of retries for the data transfers.
The default value for this property is 3.

2.8. optTimeout

Summary
Timeout value for file transfer

Description
The optTimeout property specifies the get or put timeout period in seconds. This property can be changed at design time or at run time, but must be set before calling Get or Put method. The default value for this property is 5 seconds.

3. Methods

3.1. Get

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

Syntax
long Get(BSTR strHost, BSTR strRemoteFile, BSTR strLocalFile)

Description
The Get method transfers a file from the remote machine and stores it locally. The host name must be set to the name or IP address of the TFTP server.

Parameters
strHost
[in] The name of the host from where the file has to be transferred.
strRemoteFile
[in] The file on the remote host.
strLocalFile
[in] Local file name.

**Return value:**
Zero indicates that no error occurred. If the call fails, the Win32 error number is returned.

### 3.2. Put

**Summary**
Put a file on the remote host

**Syntax**
```c
long Put(BSTR strHost, BSTR strRemoteFile, BSTR strLocalFile)
```

**Description**
The Put method transfers file from the local machine to remote machine. The host name must be set to the name or IP address of the TFTP server.

**Parameters**
strHost
[in] The name of the host from where the file has to be transferred.
strRemoteFile
[in] Remote file name
strLocalFile
[in] Local file name.

**Return value:**
Zero indicates that no error occurred. If the call fails, the Win32 error number is returned.

### 3.3. ResetTftpSettings

**Summary**
Set control properties to their default values.

**Syntax**
```c
void ResetTftpSettings()
```

**Description**
Reset control properties in order to use default values:
optBlksize – 512 (bytes)
optMode – 1 (binary)
optTimeout – 5 (seconds)
optPort – 69
optRetries – 3
optTsize – 1 (enabled)

Parameters
N/A

Return value:
This function does not return a value.

3.4. TftpReset

Summary
Abort all SkFTP requests.

Syntax
void TftpReset()

Description
Aborts all files transfer in progress.

Parameters
N/A

Return value:
This function does not return a value.

3.5. TftpGetStatus

Summary
Get a SkFTP query status.

Syntax
long TftpGetStatus(VARIANT* pvarAction,
                    VARIANT* pvarHost,
VARIANT* pvarRemoteFile,
VARIANT* pvarLocalFile,
VARIANT* pvarStatus,
VARIANT* pvarMode,
VARIANT* pvarSize,
VARIANT* pvarTime)

Description
Note: This method should be used only when TFTP control is used as a COM server, not an
ActiveX control, for instance, when TFTP control is instantiated from ASP page or Windows
Scripting Host.
When TFTP is used as regular ActiveX control, notification event TftpQueryCompleted should
be used instead.

Parameters
pvarAction
[in] The name of the host from where the file has to be transferred.
pvarHost
[in] The name of the host from where the file has to be transferred.
pvarRemoteFile
[in] Remote file name.
pvarLocalFile
[in] Local file name
pvarStatus
[in] Local file name
pvarSize
[in] Local file name
pvarTime
[in] Local file name

Return value:
This function does not return a value.

4. Events

4.1. TftpCompleted
Summary
Indicate that TFTP control has stopped processing all upload/download requests.

Syntax
void TftpCompleted()

Description
Indicate that TFTP control has stopped processing all upload/download requests.

Parameters
N/A

Return value:
None

4.2. TftpQueryCompleted

Summary
Occurs when TFTP control has stopped processing single specific upload/download request.

Syntax
void TftpQueryCompleted( short nAction,
                        BSTR strHost,
                        BSTR strRemoteFile,
                        BSTR strLocalFile,
                        long lStatus,
                        short nMode,
                        long lSize,
                        long lTime)

Description
Indicate that TFTP control has stopped processing single specific upload/download request.

Parameters
nAction
[in] Type of action performed. 0 – get action, 1 – put action
strHost
[in] The name of the host from where the file has to be transferred
strRemoteFile
[in] Remote file name
strLocalFile
[in] Local file name.
lStatus
[in] Status of the query. See the complete list of supported error codes.
nMode
[in] Type of transfer performed. 1 - binary transfer, 0 – ASCII transfer
lSize
[in] Size of the file to be transferred
lTime
[in] time elapsed to complete the operation

**Return value:**
None

5. Examples
TFTP ActiveX control comes with complete documentation, VC++/VB/VB.Net sample code, and working demo programs distributed during install time. Demo programs below were written to use TFTP control to retrieve process related information.

**Figure 1 TFTP Demo**
Server: 127.0.0.1
Remote File: 2.txt
Local File: 2.txt
Options:
- Mode: Binary
- Port: 69
- Retries: 3
- Timeout, sec: 5
Extended options:
- Blocksize: 2048
- Use Blocksize option
- Use timeout option
- Use tsize option
Status: Transfer successful: 106730 bytes in 1 seconds