

# Table of Contents

Foreword	0
<b>Part I About MegaPipe ActiveX (OCX)</b>	<b>3</b>
1 Introduction.....	3
2 How to Use It.....	3
Trial Version .....	3
Full Version .....	4
3 How to Distribute It.....	5
<b>Part II Reference Guide</b>	<b>5</b>
1 Serial Communication.....	5
<b>Properties</b> .....	<b>5</b>
AvailDataCount Property.....	5
BaudRate Property.....	6
CDStatus Property.....	6
CTSSStatus Property.....	6
DataBits Property.....	7
DSRStatus Property.....	7
Flow Control Property.....	7
InputData Property.....	7
InputQ Property.....	8
IsConnected Property.....	8
OutputData Property.....	8
OutputQ Property.....	9
OutputStringData Property.....	9
Parity Property.....	9
Port Property.....	9
PortOpen Property.....	10
RingStatus Property.....	10
RThreshold Property.....	10
StopBits Property.....	10
XoffChar Property.....	11
XonChar Property.....	11
<b>Methods</b> .....	<b>11</b>
EscapeCommFunc Method.....	11
Read Method.....	12
<b>Events</b> .....	<b>13</b>
CDChanged Event.....	13
CTSChanged Event.....	13
DSRChanged Event.....	13
OnComm Event.....	13
RingChanged Event.....	14
<b>Enumerations</b> .....	<b>14</b>
DataBits Enumeration.....	14
Flow Control Enumeration.....	14
Parity Enumeration.....	15
StopBits Enumeration.....	15

<b>2</b>	<b>Modem</b>	<b>15</b>
	<b>Properties</b>	<b>15</b>
	DropCall Property	15
	MakeCall Property	15
	ModemCount Property	16
	ModemSelIndex Property	16
	PhoneNumber Property	16
	WaitForCall Property	17
	<b>Methods</b>	<b>17</b>
	CloseTAPI Method	17
	GetModemName Method	17
	GetModemPort Method	18
	InitTAPI Method	18
	<b>Events</b>	<b>19</b>
	LineStatus Event	19
<b>3</b>	<b>File Transfer</b>	<b>20</b>
	<b>Properties</b>	<b>20</b>
	XferCurrBytes Property	20
	XferCurrFileName Property	20
	XferCurrFileSize Property	20
	XferMode Property	20
	XferProtocol Property	21
	XferStart Property	21
	XferStop Property	21
	XferTimeoutInterval	21
	XferTimeoutsAllowed	21
	XferWorkDir Property	22
	XferZCREnabled Property	22
	<b>Methods</b>	<b>22</b>
	XferAddFile Method	22
	XferClearAllFiles Method	23
	XferSetDstFile Method	23
	<b>Events</b>	<b>23</b>
	AbortTransfer Event	23
	DoInit Event	23
	FinishFile Event	24
	FinishTransfer Event	24
	OneBlockDone Event	24
	StartFile Event	25
	<b>Enumerations</b>	<b>25</b>
	Action Enumeration	25
	Protocol Enumeration	25
<b>4</b>	<b>Common Events</b>	<b>25</b>
	HasError Event	25
	HasWarning Event	26

<b>Part III License</b>	<b>26</b>
-------------------------	-----------

<b>Index</b>	<b>0</b>
--------------	----------

# 1 About MegaPipe ActiveX (OCX)

## 1.1 Introduction

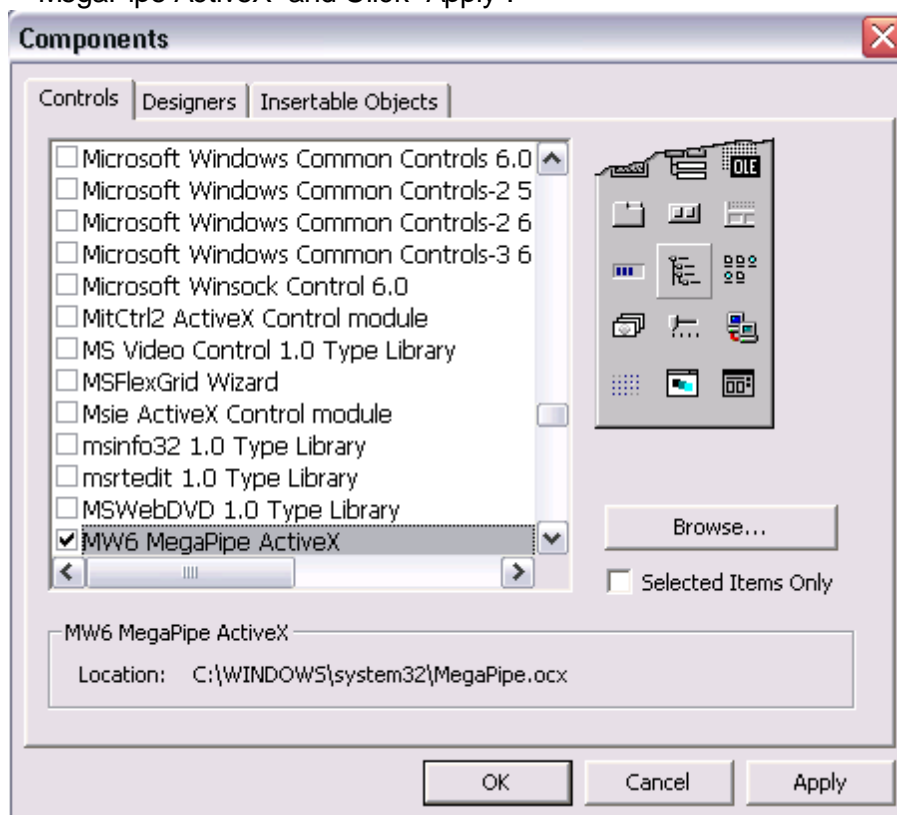
MegaPipe ActiveX (OCX) written in Visual Basic 6.0 is a reliable and powerful control for handling serial communication, modem operation and file-transfer (XModem Checksum, XModem CRC, XModem 1K, YModem, YModem-G, ZModem and Kermit). It can be used in any ActiveX-compliant environment such as VC++, VB, VB.Net, C#.Net, Access, Delphi or Borland C++.

## 1.2 How to Use It

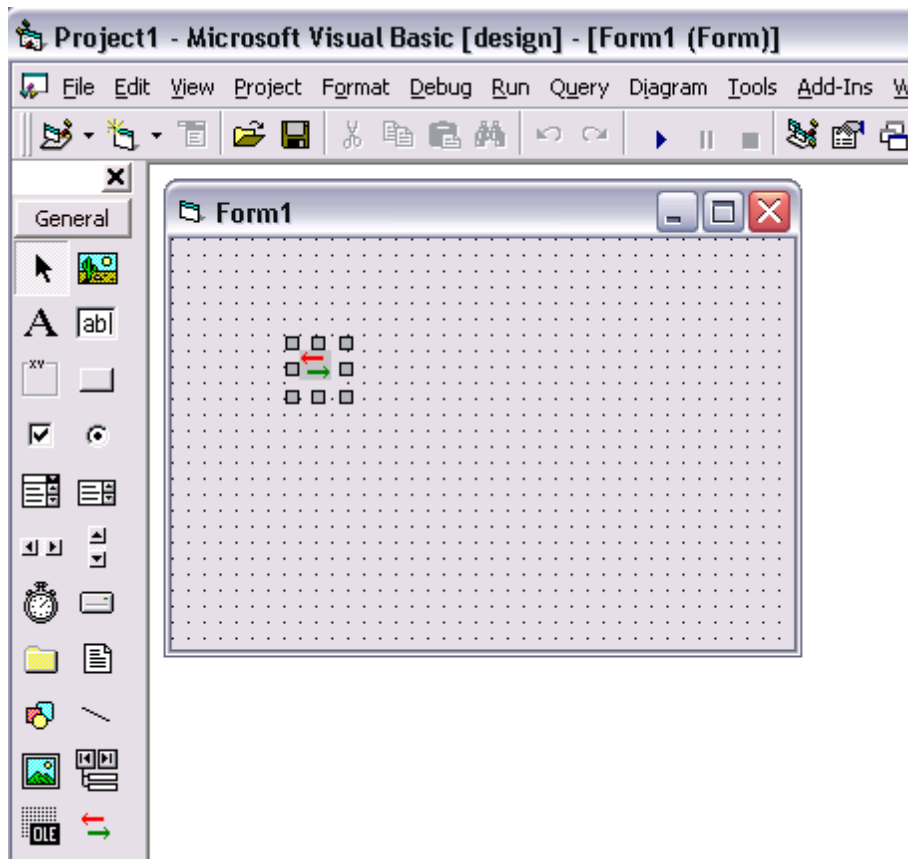
### 1.2.1 Trial Version

It is very easy to add the MegaPipe ActiveX (OCX) to your project in any ActiveX compliant IDE such as Visual Basic 6.0, Visual C++ 6.0, Access or Visual FoxPro.

1. After the installation for the trial version is finished, your ActiveX compliant IDE should be able to recognize the MegaPipe ActiveX (OCX).
2. Assume you are using Visual Basic 6.0, click "Project" > "Components", choose "MW6 MegaPipe ActiveX" and Click "Apply".



3. Drag and Drop the MegaPipe ActiveX (OCX) on your windows form.



### 1.2.2 Full Version

Follow the instructions listed below to add the full version MegaPipe ActiveX (OCX) to your project:

1. Uninstall the trial version and complete the installation for the full version.
2. The license key method SetKey should be called before you call any other MegaPipe function.

[Visual Basic]

```
Public Function SetKey(ByVal Key As String) As Boolean
```

#### Parameters

*Key*

10 digits license key obtained from us.

#### Return Value

If the license key is valid, the return value is true, otherwise the return value is false.

## Examples

[Visual Basic]

```
Dim ValidKey As Boolean
ValidKey = MegaPipeCtrl1.SetKey("XXXXXX-XXXX")

MegaPipeCtrl1.BaudRate = 9600
MegaPipeCtrl1.InputQ = 16384
MegaPipeCtrl1.ModemSelIndex = -1
MegaPipeCtrl1.OutputQ = 16384
MegaPipeCtrl1.PhoneNumber = Nothing
MegaPipeCtrl1.RThreshold = 0
MegaPipeCtrl1.XferWorkDir = Nothing
```

## 1.3 How to Distribute It

If you want to redistribute the MegaPipe ActiveX (OCX) as part of your application, please follow the instructions below:

- 1) For 32-bit version Windows OS, put **MegaPipe.ocx** into the windows 32-bit system folder (e.g. "c:\windows\system32" or "c:\winnt\system32") on the target machine and run "regsvr32 MegaPipe.ocx" to register it.
- 2) For 64-bit version Windows OS, put **MegaPipe.ocx** into the SysWOW64 folder (e.g. "c:\windows\SysWOW64") on the target machine, and run the following commands to register it:
  - cd \windows\SysWOW64
  - regsvr32 MegaPipe.ocx
- 3) For Windows Vista or above, you need to use an elevated Command Prompt to run *regsvr32.exe* command, click "**Start**" > "**All Programs**" > "**Accessories**", right-click "**Command Prompt**", and then click "**Run as administrator**".

## 2 Reference Guide

### 2.1 Serial Communication

#### 2.1.1 Properties

##### 2.1.1.1 AvailDataCount Property

Returns the number of incoming data bytes available in the input buffer.

[Visual Basic]

Public Property AvailDataCount() As Integer

### Remarks

This is a read-only property.

### See Also

InputData Property

#### 2.1.1.2 BaudRate Property

Gets or sets the baud rate of the serial communication.

[Visual Basic]

Public Property BaudRate() As Long

### Remarks

If you use MegaPipe modem-related properties (e.g. MakeCall Property and WaitForCall Property) to establish a phone line connection, you don't need to touch this property, Microsoft TAPI will take care of it automatically.

#### 2.1.1.3 CDStatus Property

Gets the status of the serial communication CD (carrier detect) line.

[Visual Basic]

Public Property CDStatus() As Boolean

### Remarks

This is a read-only property.

#### 2.1.1.4 CTSSStatus Property

Gets the status of the serial communication CTS (clear to send) line.

[Visual Basic]

Public Property CTSSStatus() As Boolean

### Remarks

This is a read-only property.

---

#### 2.1.1.5 DataBits Property

Gets or sets the number of data bits per byte for the serial communication.

[Visual Basic]

```
Public Property DataBits() As DataBitsType
```

##### Remarks

If you use MegaPipe modem-related properties (e.g. MakeCall Property and WaitForCall Property) to establish a phone line connection, you don't need to touch this property, Microsoft TAPI will take care of it automatically.

#### 2.1.1.6 DSRStatus Property

Gets the status of the serial communication DSR (data-set-ready) line.

[Visual Basic]

```
Public Property DSRStatus() As Boolean
```

##### Remarks

This is a read-only property.

#### 2.1.1.7 FlowControl Property

Gets or sets the type of the flow control used for the serial communication.

[Visual Basic]

```
Public Property FlowControl() As FlowControlType
```

##### Remarks

If you use MegaPipe modem-related properties (e.g. MakeCall Property and WaitForCall Property) to establish a phone line connection, you don't need to touch this property, Microsoft TAPI will take care of it automatically.

#### 2.1.1.8 InputData Property

Returns the incoming data in the input buffer.

[Visual Basic]

```
Public Property InputData() As Variant
```

##### Remarks

This is a read-only property.

**See Also**

AvailDataCount Property

**2.1.1.9 InputQ Property**

Gets or sets the size of input buffer for the serial communication.

[Visual Basic]

Public Property InputQ() As Integer

**Remarks**

If you use MegaPipe modem-related properties (MakeCall Property or WaitForCall Property) to establish a phone line connection, you don't need to touch this property, Microsoft TAPI will take care of it automatically.

**2.1.1.10 IsConnected Property**

Indicates whether a serial line (direct cable) or a phone line (modem based) is connected or disconnected.

[Visual Basic]

Public Property IsConnected() As Boolean

**Remarks**

This is a read-only property.

**2.1.1.11 OutputData Property**

Sets the outgoing data stream with Null characters (ASCII value 0).

[Visual Basic]

Public Property OutputData() As Variant

**Remarks**

This is a write-only property.

---



#### 2.1.1.12 OutputQ Property

Gets or sets the size of output buffer for the serial communication.

[Visual Basic]

```
Public Property OutputQ() As Integer
```

##### Remarks

If you use MegaPipe modem-related properties (MakeCall Property or WaitForCall Property) to establish a phone line connection, you don't need to touch this property, Microsoft TAPI will take care of it automatically.

#### 2.1.1.13 OutputStringData Property

Sets the outgoing data stream without Null characters (ASCII value 0).

[Visual Basic]

```
Public Property OutputStringData() As String
```

##### Remarks

This is a write-only property.

#### 2.1.1.14 Parity Property

Gets or sets the type of the parity check for the serial communication.

[Visual Basic]

```
Public Property Parity() As ParityType
```

##### Remarks

If you use MegaPipe modem-related properties (e.g. MakeCall Property and WaitForCall Property) to establish a phone line connection, you don't need to touch this property, Microsoft TAPI will take care of it automatically.

#### 2.1.1.15 Port Property

Gets or sets the port name (i.e. "COM1") used for the serial communication.

[Visual Basic]

```
Public Property Port() As String
```

##### Remarks

If you use MegaPipe modem-related properties (MakeCall Property or WaitForCall Property) to establish a phone line connection, you don't need to touch this property, Microsoft TAPI will

take care of it automatically.

#### 2.1.1.16 PortOpen Property

Indicates whether to open a serial port or not.

[Visual Basic]

Public Property PortOpen() As Boolean

#### Remarks

This is a write-only property.

#### 2.1.1.17 RingStatus Property

Gets the status of the serial communication ring indicator line.

[Visual Basic]

Public Property RingStatus() As Boolean

#### Remarks

This is a read-only property.

#### 2.1.1.18 RThreshold Property

Gets or sets the minimum number of characters sitting in the input buffer before the OnComm event is triggered.

[Visual Basic]

Public Property RThreshold() As Integer

#### Remarks

if this property is 0, OnComm event will never be fired.

#### See Also

OnComm event

#### 2.1.1.19 StopBits Property

Gets or sets the number of stop bits for the serial communication.

[Visual Basic]

---

```
Public Property StopBits() As StopBitsType
```

### Remarks

If you use MegaPipe modem-related properties (e.g. MakeCall Property and WaitForCall Property) to establish a phone line connection, you don't need to touch this property, Microsoft TAPI will take care of it automatically.

#### 2.1.1.20 XoffChar Property

Gets or sets the ASCII value of the XOFF character for the serial communication.

```
[Visual Basic]
```

```
Public Property XoffChar() As Integer
```

### Remarks

If you use MegaPipe modem-related properties (e.g. MakeCall Property and WaitForCall Property) to establish a phone line connection, you don't need to touch this property, Microsoft TAPI will take care of it automatically.

#### 2.1.1.21 XonChar Property

Gets or sets the ASCII value of the XON character for the serial communication.

```
[Visual Basic]
```

```
Public Property XonChar() As Integer
```

### Remarks

If you use MegaPipe modem-related properties (e.g. MakeCall Property and WaitForCall Property) to establish a phone line connection, you don't need to touch this property, Microsoft TAPI will take care of it automatically.

## 2.1.2 Methods

#### 2.1.2.1 EscapeCommFunc Method

Performs an extended function for the serial communication.

```
[Visual Basic]
```

```
Public Function EscapeCommFunc(ByVal dwFunc As Integer) As Boolean
```

### Parameters

*dwFunc*

Extended function to be performed, this parameter can be one of the following values.

dwFunc Value	Comment
--------------	---------

3	Sets RTS (request-to-send) line
4	Clears RTS (request-to-send) line
5	Sets DTR (data-terminal-ready) line
6	Clears DTR (data-terminal-ready) line.
7	Resets device if possible
8	Sets the device break line
9	Clears the device break line

### Return Value

If the function succeeds, the return value is true, otherwise the return value is false.

### Remarks

If you want to shut down a phone line connection established by modem AT commands and still keep port open, you can use this function to achieve it by assigning 6 to dwFunc parameter.

#### 2.1.2.2 Read Method

Reads a certain number of bytes from the input buffer.

[Visual Basic .NET]

```
Public Function Read(ByVal Count As Long) As Variant
```

### Parameters

*Count*

The number of bytes to read from the input buffer.

### Return Value

A Variant that contains a byte array read from the input buffer.

### Remarks

You can use this method to read partial data in the input buffer, the InputData property returns all available data in the input buffer.

### See Also

---

InputData Property

## 2.1.3 Events

### 2.1.3.1 CDChanged Event

This event is fired when status of the CD (carrier detect) line changes.

[Visual Basic]

```
Public Event CDChanged()
```

### 2.1.3.2 CTSChecked Event

This event is fired when status of the CTS (clear to send) line changes.

[Visual Basic]

```
Public Event CTSChecked()
```

### 2.1.3.3 DSRChanged Event

This event is fired when status of the DSR (data-set-ready) line changes.

[Visual Basic]

```
Public Event DSRChanged()
```

### 2.1.3.4 OnComm Event

This event is fired when amount of incoming data bytes is greater than or equal to RThreshold.

[Visual Basic]

```
Public Event OnComm(ByVal Count As Integer)
```

## Parameters

*Count*

The number of data bytes sitting in the input buffer.

## Remarks

If RThreshold property is 0, this event will never be fired.

## See Also

RThreshold Property

### 2.1.3.5 RingChanged Event

This event is fired when status of the ring line changes.

[Visual Basic]

```
Public Event RingChanged()
```

## 2.1.4 Enumerations

### 2.1.4.1 DataBits Enumeration

An enumeration type for all possible DataBits values.

#### Members

Name	Comment
db4Bits	4 data bits
db5Bits	5 data bits
db6Bits	6 data bits
db7Bits	7 data bits
db8Bits	8 data bits

### 2.1.4.2 FlowControl Enumeration

An enumeration type for all possible flow control values.

#### Members

Name	Comment
fcNone	No flow control
fcXonXoff	Xon/Xoff software control
fcHardware	Hardware control
fcDTR	Enable DTR
fcRTS	Enable RTS
fcDTR_RTS	Enable DTR & RTS

#### Remarks

Choose different options for FlowControl property if your application can't send/receive characters correctly.

---

### 2.1.4.3 Parity Enumeration

An enumeration type for all possible Parity values.

#### Members

Name	Comment
pNone	No parity
pOdd	Odd parity
pEven	Even parity
pMark	Mark parity
pSpace	Space parity

### 2.1.4.4 StopBits Enumeration

An enumeration type for all possible StopBits values.

#### Members

Name	Comment
sb1Bit	1 stop bit
sb1x5Bits	1.5 stop bits
sb2Bits	2 stop bits

## 2.2 Modem

Enter topic text here.

### 2.2.1 Properties

#### 2.2.1.1 DropCall Property

Determines whether current established phone line connection should be cut off or not.

[Visual Basic]

Public Property DropCall() As Boolean

#### Remarks

This is a write-only property.

#### 2.2.1.2 MakeCall Property

Indicates whether to make a phone call or not.

[Visual Basic]

Public Property MakeCall() As Boolean

**Remarks**

This is a write-only property, you can use LineStatus event to monitor the status of call or wait-for-call process.

**See Also**

LineStatus Event

**2.2.1.3 ModemCount Property**

Returns the number of the modems installed on PC.

[Visual Basic]

```
Public Property ModemCount() As Integer
```

**Remarks**

This is a read-only property.

**2.2.1.4 ModemSelIndex Property**

Specifies which modem is used to establish a phone line connection.

[Visual Basic]

```
Public Property ModemSelIndex() As Integer
```

**Remarks**

This is a write-only property, if the value is -1, either no modem is involved or the modem is not handled using TAPI operations (AT-modem commands are used instead).

**2.2.1.5 PhoneNumber Property**

Specifies the phone number of the remote party to dial.

[Visual Basic]

```
Public Property PhoneNumber() As String
```

**Remarks**

This is a write-only property.

---



### 2.2.1.6 WaitForCall Property

Indicates whether to wait for an incoming call or not.

[Visual Basic]

```
Public Property WaitForCall() As Boolean
```

#### Remarks

This is a write-only property, you can use LineStatus event to monitor the status of call or wait-for-call process.

#### See Also

LineStatus Event

## 2.2.2 Methods

### 2.2.2.1 CloseTAPI Method

Closes TAPI after you finish TAPI-related modem operation(s).

[Visual Basic]

```
Public Function CloseTAPI() As Boolean
```

#### Return Value

If the function succeeds, the return value is true, otherwise the return value is false.

#### Remarks

If InitTAPI API is called, this method must be called in order to shut down TAPI functions properly.

If a phone line connection is established, be sure to set the DropCall property to true before call this function.

#### See Also

InitTAPI Method | DropCall Property

### 2.2.2.2 GetModemName Method

Returns the name of a modem.

[Visual Basic]

```
Public Function GetModemName(ByVal Index As Integer) As String
```

## Parameters

### *Index*

0-based modem index in the modem list, this parameter value must be between 0 and total number of modems - 1.

## Return Value

The return value is the name of a modem.

## See Also

ModemCount Property

### 2.2.2.3 GetModemPort Method

Returns the name of the port associated with a modem.

[Visual Basic]

```
Public Function GetModemPort(ByVal Index As Integer) As String
```

## Parameters

### *Index*

0-based modem index in the modem list, this parameter value must be between 0 and total number of modems - 1.

## Return Value

The return value is the name of the port associated with a modem.

## See Also

ModemCount Property

### 2.2.2.4 InitTAPI Method

Initializes TAPI before your application conducts TAPI-related modem operation(s).

[Visual Basic]

```
Public Function InitTAPI() As Boolean
```

## Return Value

---

If the function succeeds, the return value is true, otherwise the return value is false.

### Remarks

Call CloseTAPI method to shut down TAPI functions properly after you call this method.

### See Also

CloseTAPI Method

## 2.2.3 Events

### 2.2.3.1 LineStatus Event

This event is fired when the status for the phone line changes.

[Visual Basic]

```
Public Event LineStatus(ByVal StatusID As Integer)
```

### Parameters

*StatusID*

This parameter indicates different status for the phone line and can be one of the following values.

StatusID Value	Comment
0	There is an incoming call.
1	The call is proceeding.
2	The line is connected.
3	The line is disconnected.
4	The line is busy, please dial later.
5	No dial tone was detected.
6	The remote side does not answer.

### Remarks

You can use this event to monitor the status of call or wait-for-call process.

**See Also**

MakeCall Property | WaitForCall Property | DropCall Property

## 2.3 File Transfer

### 2.3.1 Properties

#### 2.3.1.1 XferCurrBytes Property

Returns the number of data bytes sent/received so far.

[Visual Basic]

Public Property XferCurrBytes() As Long

**Remarks**

This is a read-only property.

#### 2.3.1.2 XferCurrFileName Property

Returns the name of file being transferred.

[Visual Basic]

Public Property XferCurrFileName() As String

**Remarks**

This is a read-only property.

#### 2.3.1.3 XferCurrFileSize Property

Returns the size of file being transferred.

[Visual Basic]

Public Property XferCurrFileSize() As Long

**Remarks**

This is a read-only property.

#### 2.3.1.4 XFerMode Property

Gets or sets the file-transfer mode which can be either file-download or file-upload.

[Visual Basic]

---

```
Public Property XferMode() As ActionType
```

#### 2.3.1.5 XFerProtocol Property

Gets or sets the file-transfer protocol type, which can be XModem Checksum, XModem CRC, XModem 1K, YModem, YModem-G, ZModem or Kermit.

```
[Visual Basic]
```

```
Public Property XferProtocol() As ProtocolType
```

#### 2.3.1.6 XFerStart Property

Determines whether to start a file-transfer session or not.

```
[Visual Basic]
```

```
Public Property XferStart() As Boolean
```

#### Remarks

This is a write-only property.

#### 2.3.1.7 XFerStop Property

Determines whether to stop a file-transfer session or not.

```
[Visual Basic]
```

```
Public Property XferStop() As Boolean
```

#### Remarks

This is a write-only property, you can use this property to terminate a file-transfer session.

#### 2.3.1.8 XferTimeoutInterval

Gets or sets the timeout interval in milliseconds during a file-transfer handshake.

```
[Visual Basic]
```

```
Public Property XferTimeoutInterval() As Long
```

#### Remarks

The default value is 5000 milliseconds(5 seconds).

#### 2.3.1.9 XferTimeoutsAllowed

Gets or sets the maximum allowed number of timeouts during a file-transfer handshake.

```
[Visual Basic]
```

```
Public Property XferTimeoutsAllowed() As Integer
```

**Remarks**

The default value is 5.

**2.3.1.10 XFerWorkDir Property**

Gets or sets the work directory for a file-upload or a file-download session.

```
[Visual Basic]
```

```
Public Property XferWorkDir() As String
```

**2.3.1.11 XferZCREnabled Property**

Determines whether to allow ZModem crash recovery or not when downloading file(s) from the remote side.

```
[Visual Basic]
```

```
Public Property XferZCREnabled() As Boolean
```

**Remarks**

If this property is set to TRUE and you lose ZModem connection after initiating a download, partial files are always saved so that an aborted ZModem transfer can be recovered.

**2.3.2 Methods**

Enter topic text here.

**2.3.2.1 XferAddFile Method**

Informs MegaPipe of the name of file which will be uploaded to the remote side.

```
[Visual Basic]
```

```
Public Sub XferAddFile(ByVal FileName As String)
```

**Parameters**

*FileName*

The relevant path name of file which will be uploaded to the remote side.

**Remarks**

All XModem protocols can only transfer 1 file during one file-transfer session, your application only needs to call this API once for one file-transfer session.

---

YModem, YModem-G, ZModem or Kermit can upload multiple files during one file-transfer session, your application needs to call this API a few times if multiple files are uploaded.

#### 2.3.2.2 XferClearAllFiles Method

Clears the file name information in MegaPipe memory on the upload side.

[Visual Basic]

```
Public Sub XferClearAllFiles()
```

#### Remarks

Call this API before you call XferAddFile() API.

#### See Also

XferAddFile Method

#### 2.3.2.3 XferSetDstFile Method

Informs MegaPipe of the name of file which will be created on the download side for all XModem protocols.

[Visual Basic]

```
Public Sub XferSetDstFile(ByVal FileName As String)
```

#### Parameters

*FileName*

The relevant path name of file which will be created on the download side for all XModem protocols. .

#### Remarks

YModem, YModem-G, ZModem or Kermit doesn't need to touch this API, since the name of file on the download side is identical to the name of file on the upload side.

### 2.3.3 Events

#### 2.3.3.1 AbortTransfer Event

This event is fired when a file-transfer session is aborted.

[Visual Basic]

```
Public Event AbortTransfer()
```

#### 2.3.3.2 DoInit Event

This event is fired when initialization for a file-transfer session is in progress.

[Visual Basic]

```
Public Event DoInit()
```

#### 2.3.3.3 FinishFile Event

This event is fired when one file is uploaded or downloaded successfully during a file-transfer session.

```
[Visual Basic]
```

```
Public Event FinishFile(ByVal FileName As String)
```

#### Parameters

*FileName*

The relevant path name of file which has been uploaded or downloaded successfully.

#### Remarks

Multiple files can be uploaded or downloaded with YModem, YModem-G, ZModem or Kermit during a file-transfer session.

#### 2.3.3.4 FinishTransfer Event

This event is fired when current file transfer session is done.

```
[Visual Basic]
```

```
Public Event FinishTransfer()
```

#### 2.3.3.5 OneBlockDone Event

This event is fired when one block data are sent or received successfully.

```
[Visual Basic]
```

```
Public Event OneBlockDone()
```

#### Remarks

In your event handler for this event, you can utilize the XferCurrBytes property to obtain the information about how many bytes of data have been sent or received successfully so far.

#### See Also

XferCurrBytes Property

---



### 2.3.3.6 StartFile Event

This event is fired when one file is starting to be uploaded or downloaded.

[Visual Basic]

```
Public Event StartFile(ByVal FileName As String)
```

#### Parameters

*FileName*

The relevant path name of file which is starting to be uploaded or downloaded.

## 2.3.4 Enumerations

### 2.3.4.1 Action Enumeration

An enumeration type for all possible XferMode values.

#### Members

Name	Comment
aUpload	Upload file
aDownload	Download file

### 2.3.4.2 Protocol Enumeration

An enumeration type for all possible file-transfer protocol values.

#### Members

Name	Comment
pXModemChecksum	XModem Checksum protocol
pXModemCRC	XModem CRC protocol
pXModem1K	XModem 1K protocol
pYModem	YModem protocol
pYModemG	YModem-G protocol
pZModem	ZModem protocol
pKermit	Kermit protocol

## 2.4 Common Events

### 2.4.1 HasError Event

This event is fired when a fatal error occurs for serial communication, modem operation, file transfer process or Kermit command execution.

[Visual Basic]

```
Public Event HasError(ByVal ErrorMessage As String)
```

```
[VC++]  
public event HasError(string ErrorMsg);
```

### Parameters

*ErrorMsg*

Fatal error message.

## 2.4.2 HasWarning Event

This event is fired when a non-fatal warning pops up for serial communication, modem operation, file transfer process or Kermit command execution.

```
[Visual Basic]  
Public Event HasWarning(ByVal WarningMsg As String)
```

```
[VC++]  
public event HasWarning(string WarningMsg);
```

### Parameters

*WarningMsg*

Non-fatal warning message.

## 3 License

### License agreement

This License Agreement ("LA") is the legal agreement between you and MW6 Technologies, Inc. ("MW6") for the software, the font, and any electronic documentation ("Package"). By using, copying or installing the Package, you agree to be bound by the terms of this LA. If you don't agree to the terms in this LA, immediately remove unused Package.

#### 1. License

\* The Single Developer License allows 1 developer in your organization the royalty-free distribution (up to 10,000 users) of the software to the third parties, **each individual developer requires a separate Single Developer License as long as he or she needs access to MW6's product(s) and document(s).**

\* The 2 Developer License allows 2 developers in your organization the royalty-free distribution (up to 10,000 users) of the software to the third parties.

\* The 3 Developer License allows 3 developers in your organization the royalty-free distribution (up to 10,000 users) of the software to the third parties.

\* The 4 Developer License allows 4 developers in your organization the royalty-free distribution (up to 10,000 users) of the software to the third parties.

---

---

\* The 5 Developer License allows 5 developers in your organization the royalty-free distribution (up to 10,000 users) of the software to the third parties.

\* The Unlimited Developer License allows unlimited number of developers in your organization the royalty-free distribution (unlimited number of users) of the software to the third parties.

## **2. User Disclaimer**

The software is provided "as is" without warrant of any kind, either expressed or implied, including, but not limited to, the implied warranties of merchantability, fitness for a particular purpose, or noninfringement. MW6 assumes no liability for damages, direct or consequential, which may result from the use of the software. Further, MW6 assumes no liability for losses caused by misuse or abuse of the software. This responsibility rests solely with the end user.

## **3. Copyright**

The software and any electronic documentation are the proprietary products of MW6 and are protected by copyright and other intellectual property laws.

---