Table of Contents

1 About MegaPipe .NET Control

1.1 Introduction

MegaPipe .NET control is a reliable and powerful 100% managed code component for handling serial communication, modem operations and file-transfer (XModem Checksum, XModem CRC, XModem 1K, YModem, YModem-G, ZModem and Kermit). It can be used in VB.NET and C#.NET.

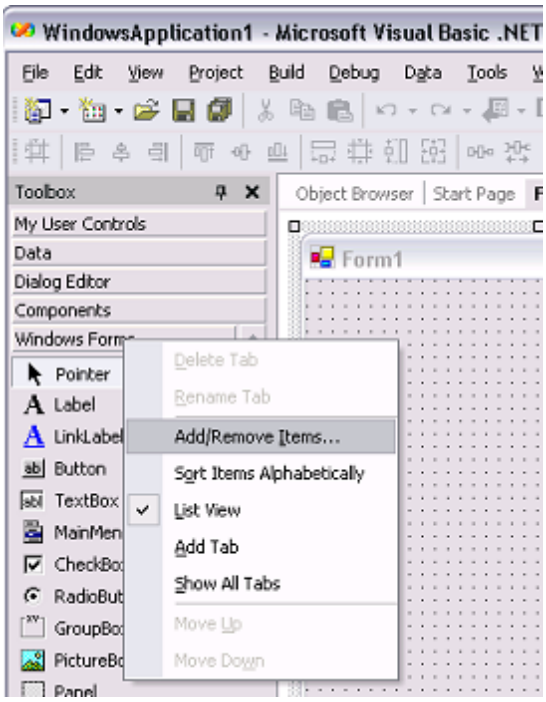
1.2 How to Use It

1.2.1 Trial Version

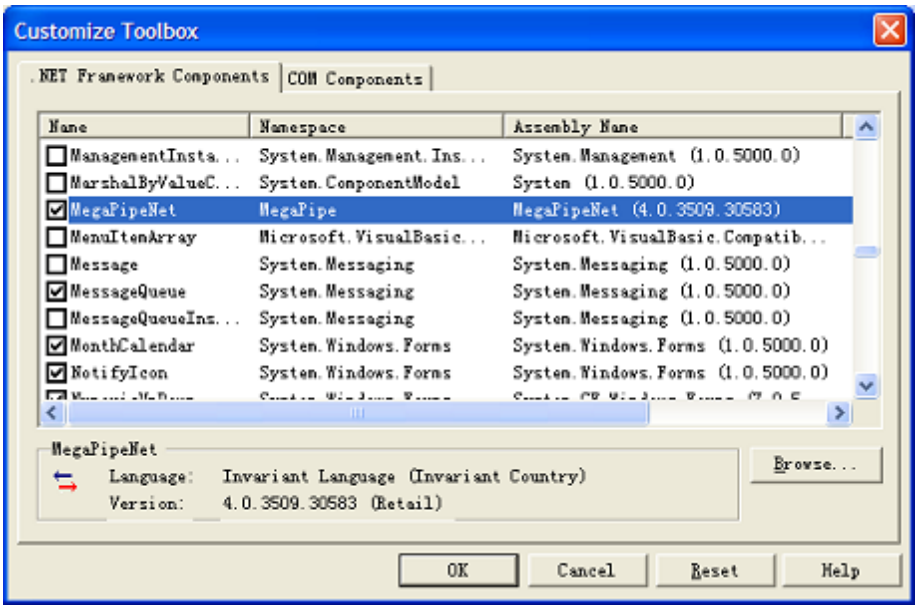
It is straightforward to add the trial version MegaPipe .NET control to your project in Visual Studio .NET environment:

1. After the installation for the trial version is finished, copy MegaPipeNet.dll in VB.NET demo project sub folder or C# demo project sub folder in the destination folder (e.g. "C:\Program Files\MW6 .NET Components\MegaPipe\Trial Version") to your project folder.

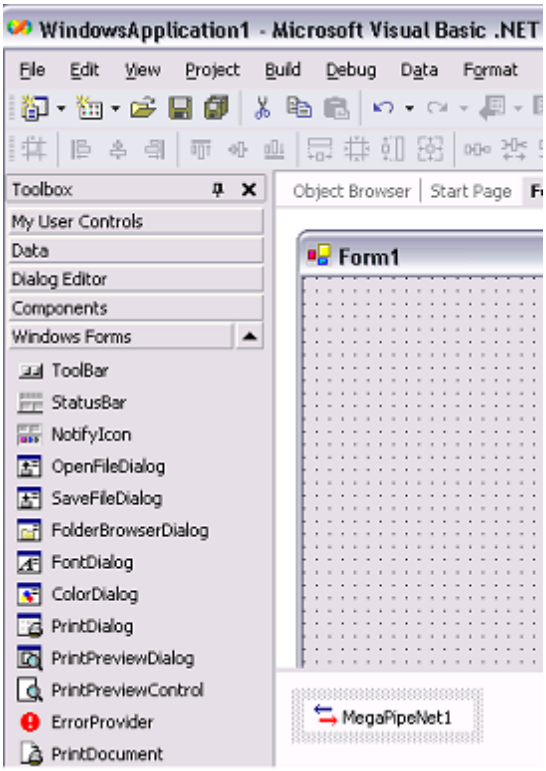
2. Right click anywhere on the Toolbox to select "Add/Remove Items...".



3. In the "Customize Toolbox" dialog, select ".NET Framework Components", click "Browse" and select MegaPipeNet.dll.



4. To use the control in your Windows Form, just drag it from the Toolbox and drop it onto your form.



1.2.2 Full Version

Follow the instructions listed below to add the full version MegaPipe .NET control to your project:

1. Uninstall the trial version and complete the installation for the full version, copy MegaPipeNet.dll in VB.NET demo project sub folder or C# demo project sub folder in the destination folder (e.g. "C:\Program Files\MW6 .NET Components\MegaPipe\Full Version") to your project folder to replace the trial version MegaPipe .NET control.

2. The license key method SetKey should be called before you call any other MegaPipe function.

[Visual Basic .NET]

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

[C#]

```
public bool SetKey(string Key);
```

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 .NET]

```
Dim ValidKey As Boolean
MegaPipeNet1 = New MegaPipe.MegaPipeNet(Me.components)
ValidKey = MegaPipeNet1.SetKey("XXXXXX-XXXX")

MegaPipeNet1.BaudRate = 9600
MegaPipeNet1.InputQ = 16384
MegaPipeNet1.ModemSelIndex = CType(0, Short)
MegaPipeNet1.OutputQ = 16384
MegaPipeNet1.PhoneNumber = Nothing
MegaPipeNet1.RThreshold = 0
MegaPipeNet1.XferWorkDir = Nothing
```

[C#]

```
bool ValidKey;
MegaPipeNet1 = new MegaPipe.MegaPipeNet(this.components);
ValidKey = MegaPipeNet1.SetKey("XXXXXX-XXXX");
```

```
MegaPipeNet1.BaudRate = 9600;
MegaPipeNet1.InputQ = 16384;
MegaPipeNet1.ModemSelIndex = ((short)(0));
MegaPipeNet1.OutputQ = 16384;
MegaPipeNet1.PhoneNumber = null;
MegaPipeNet1.RThreshold = 0;
MegaPipeNet1.XferWorkDir = null;
```

1.3 How to Distribute It

If you want to redistribute MegaPipe .NET control as part of your application, simply put MegaPipeNet.dll into application local folder on the target machine.

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 .NET]

```
Public ReadOnly Property AvailDataCount As Integer
```

[C#]

```
public int AvailDataCount {get;}
```

See Also

InputData Property

2.1.1.2 BaudRate Property

Gets or sets the baud rate of the serial communication.

[Visual Basic .NET]

```
Public Property BaudRate As Long
```

[C#]

```
public long BaudRate {get; set;}
```

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 .NET]

```
Public ReadOnly Property CDStatus As Boolean
```

[C#]

```
public bool CDStatus {get;}
```

2.1.1.4 CTSStatus Property

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

[Visual Basic .NET]

```
Public ReadOnly Property CTSStatus As Boolean
```

[C#]

```
public bool CTSStatus {get;}
```

2.1.1.5 DataBits Property

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

[Visual Basic .NET]

```
Public Property DataBits As enumDataBits
```

[C#]

```
public enumDataBits DataBits {get; set;}
```

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 .NET]

```
Public ReadOnly Property DSRStatus As Boolean
```

[C#]

```
public bool DSRStatus {get;}
```

2.1.1.7 FlowControl Property

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

[Visual Basic .NET]

```
Public Property FlowControl As enumFlowControl
```

[C#]

```
public enumFlowControl FlowControl {get; set;}
```

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 .NET]

```
Public ReadOnly Property InputData As Byte()
```

[C#]

```
public byte[] InputData {get;}
```

See Also

AvailDataCount Property

2.1.1.9 InputQ Property

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

[Visual Basic .NET]

```
Public Property InputQ As Integer
```

[C#]

```
public int InputQ {get; set;}
```

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 .NET]

```
Public ReadOnly Property IsConnected As Boolean
```

[C#]

```
public bool IsConnected  {get;}
```

2.1.1.11 OutputData Property

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

[Visual Basic .NET]

```
Public WriteOnly Property OutputData As Byte()
```

[C#]

```
public byte[] OutputData {set;}
```

2.1.1.12 OutputQ Property

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

[Visual Basic .NET]

```
Public Property OutputQ As Integer
```

[C#]

```
public int OutputQ {get; set;}
```

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 .NET]

```
Public WriteOnly Property OutputStringData As String
```

[C#]

```
public string OutputStringData {set;}
```

2.1.1.14 Parity Property

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

[Visual Basic .NET]

```
Public Property Parity As enumParity
```

[C#]

```
public enumParity Parity {get; set;}
```

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 .NET]

```
Public Property Port As String
```

[C#]

```
public string Port {get; set;}
```

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 .NET]

```
Public WriteOnly Property PortOpen As Boolean
```

[C#]

```
public bool PortOpen  {set;}
```

2.1.1.17 RingStatus Property

Gets the status of the serial communication ring indicator line.

[Visual Basic .NET]

```
Public ReadOnly Property RingStatus As Boolean
```

[C#]

```
public bool RingStatus {get;}
```

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 .NET]

```
Public Property RThreshold As Integer
```

[C#]

```
public int RThreshold {get; set;}
```

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 .NET]

```
Public Property StopBits As enumStopBits
```

[C#]

```
public enumStopBits StopBits {get; set;}
```

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 .NET]

```
Public Property XoffChar As Integer
```

[C#]

```
public int XoffChar {get; set;}
```

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 .NET]

```
Public Property XonChar As Integer
```

[C#]

```
public int XonChar {get; set;}
```

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 .NET]

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

[C#]

```
public bool EscapeCommFunc(int dwFunc);
```

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 Byte()
```

[C#]

```
public byte[] Read(long Count);
```

Parameters

Count

The number of bytes to read from the input buffer.

Return Value

A byte array containing the bytes 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 .NET]
Public Event CDChanged()
```

```
[C#]
public event CDChanged();
```

2.1.3.2 CTSChanged Event

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

```
[Visual Basic .NET]
Public Event CTSChanged()
```

```
[C#]
public event CTSChanged();
```

2.1.3.3 DSRChanged Event

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

[Visual Basic .NET]

```
Public Event DSRChanged()
```

[C#]

```
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 .NET]

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

[C#]

```
public event OnComm(int count);
```

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 .NET]

```
Public Event RingChanged()
```

[C#]

```
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

2.2.1 Properties

2.2.1.1 DropCall Property

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

[Visual Basic .NET]

```
Public WriteOnly Property DropCall As Boolean
```

[C#]

```
public bool DropCall {set;}
```

2.2.1.2 MakeCall Property

Indicates whether to make a phone call or not.

[Visual Basic .NET]

```
Public WriteOnly Property MakeCall As Boolean
```

[C#]

```
public bool MakeCall {set;}
```

Remarks

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 modems installed on PC.

[Visual Basic .NET]

```
Public ReadOnly Property ModemCount As Integer
```

[C#]

```
public int ModemCount {get;}
```

2.2.1.4 ModemSelIndex Property

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

[Visual Basic .NET]

```
Public WriteOnly Property ModemSelIndex As Integer
```

[C#]

```
public int ModemSelIndex {set;}
```

Remarks

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 .NET]

```
Public WriteOnly Property PhoneNumber As String
```

[C#]

```
public string PhoneNumber {set;}
```

2.2.1.6 WaitForCall Property

Indicates whether to wait for an incoming call or not.

[Visual Basic .NET]

```
Public WriteOnly Property WaitForCall As Boolean
```

[C#]

```
public bool WaitForCall {set;}
```

Remarks

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 .NET]

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

[C#]

```
public bool CloseTAPI();
```

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 .NET]

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

[C#]

```
public string GetModemName(int Index);
```

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 .NET]

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

[C#]

```
public string GetModemPort(int Index);
```

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 .NET]

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

```
[C#]
public bool InitTAPI();
```

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 before your application exits.

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 .NET]
Public Event LineStatus(ByVal  StatusID As Integer)
```

```
[C#]
public event LineStatus(int StatusID);
```

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 which have been sent/received so far.

[Visual Basic .NET]

```
Public ReadOnly Property XferCurrBytes As Long
```

[C#]

```
public long XferCurrBytes {get;}
```

2.3.1.2 XferCurrFileName Property

Returns the name of file being transferred.

[Visual Basic .NET]

```
Public ReadOnly Property XferCurrFileName As String
```

[C#]

```
public string XferCurrFileName {get;}
```

2.3.1.3 XferCurrFileSize Property

Returns the size of file being transferred.

[Visual Basic .NET]

```
Public ReadOnly Property XferCurrFileSize As Long
```

[C#]

```
public long XferCurrFileSize {get;}
```

2.3.1.4 XFerMode Property

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

[Visual Basic .NET]

```
Public Property XferMode As enumAction
```

[C#]

```
public enumAction XferMode {get; set;}
```

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 .NET]

```
Public Property XferProtocol As enumProtocol
```

[C#]

```
public enumProtocol XferProtocol {get; set;}
```

2.3.1.6 XFerStart Property

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

[Visual Basic .NET]

```
Public WriteOnly Property XferStart As Boolean
```

[C#]

```
public bool XferStart {set;}
```

2.3.1.7 XFerStop Property

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

[Visual Basic .NET]

```
Public WriteOnly Property XferStop As Boolean
```

[C#]

```
public bool XferStop {set;}
```

Remarks

You can use this property to terminate a file-transfer session.

2.3.1.8 XFerWorkDir Property

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

[Visual Basic .NET]

```
Public Property XferWorkDir As String
```

[C#]

```
public string XferWorkDir {get; set;}
```

2.3.2 Methods

2.3.2.1 XferAddFile Method

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

[Visual Basic .NET]

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

[C#]

```
public void XferAddFile(string FileName);
```

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 file name information in MegaPipe memory on the upload side.

[Visual Basic .NET]

```
Public Sub XferClearAllFiles()
```

[C#]

```
public void 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 .NET]

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

[C#]

```
public void XferSetDstFile(string FileName);
```

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 .NET]

```
Public Event AbortTransfer()
```

[C#]

```
public event AbortTransfer();
```

2.3.3.2 DoInit Event

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

[Visual Basic .NET]

```
Public Event DoInit()
```

```
[C#]
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 .NET]
Public Event FinishFile(ByVal FileName As String)
```

```
[C#]
public event FinishFile(string FileName);
```

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 completed.

```
[Visual Basic .NET]
Public Event FinishTransfer()
```

```
[C#]
public event FinishTransfer();
```

2.3.3.5 OneBlockDone Event

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

```
[Visual Basic .NET]
Public Event OneBlockDone()
```

```
[C#]
```

```
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 .NET]
Public Event StartFile(ByVal FileName As String)
```

```
[C#]
public event StartFile(string FileName);
```

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 Kermit Commands

2.4.1 Methods

2.4.1.1 KCopy Method

Performs a Kermit "Copy" command.

[Visual Basic .NET]

```
Public Sub KCopy(ByVal SrcFile As String, ByVal DstFile As String)
```

[C#]

```
public void KCopy(string SrcFile, string DstFile);
```

Parameters

SrcFile

The relevant path name of source file on Kermit server.

DstFile

The relevant path name of destination file on Kermit server.

Remarks

After this command is executed on Kermit server, Kermit sends the command execution status back to MegaPipe and KCommandDone event is fired indicating whether this command is executed successfully or execution fails (an error message is passed back).

See Also

KCommandDone Event

2.4.1.2 KCWD Method

Performs a Kermit "Change Work Directory" command.

[Visual Basic .NET]

```
Public Sub KCWD(ByVal Directory As String, ByVal Password As String)
```

[C#]

```
public void KCWD(string Directory, string Password);
```

Parameters

Directory

The absolute path name of the work directory on Kermit server.

Password

Used for authorization check before the "Work Directory" is changed. If it is an empty string, no authorization check is involved.

Remarks

After this command is executed on Kermit server, Kermit sends the command execution status back to MegaPipe and KCommandDone event is fired indicating whether this command is executed successfully or execution fails (an error message is passed back).

See Also

KCommandDone Event

2.4.1.3 KDiskUsage Method

Performs a Kermit "Disk Usage Query" command.

[Visual Basic .NET]

```
Public Sub KDiskUsage()
```

[C#]

```
public void KDiskUsage();
```

Remarks

After this command is executed on Kermit server, Kermit sends the command execution status back to MegaPipe and KCommandDone event is fired indicating whether this command is executed successfully or execution fails (an error message is passed back).

See Also

KCommandDone Event

2.4.1.4 KEraseFile Method

Performs a Kermit "Erase File" command.

[Visual Basic .NET]

```
Public Sub KEraseFile(ByVal File As String)
```

[C#]

```
public void KEraseFile(string File);
```

Parameters

File

The relevant path name of file to be deleted on Kermit server.

Remarks

After this command is executed on Kermit server, Kermit sends the command execution status back to MegaPipe and KCommandDone event is fired indicating whether this command is executed successfully or execution fails (an error message is passed back).

See Also

KCommandDone Event

2.4.1.5 KFinish Method

Performs a Kermit "Finish" command.

[Visual Basic .NET]

```
Public Sub KFinish()
```

[C#]

```
public void KFinish();
```

Remarks

After this command is executed on Kermit server, Kermit sends the command execution

status back to MegaPipe and KCommandDone event is fired indicating whether this command is executed successfully or execution fails (an error message is passed back).

See Also

KCommandDone Event

2.4.1.6 KListDirectory Method

Performs a Kermit "Directory" command.

[Visual Basic .NET]

```
Public Sub KListDirectory(ByVal Mask As String)
```

[C#]

```
public void KListDirectory(string Mask);
```

Parameters

Mask

Used to filter out files by file name for a specified folder on Kermit server.

Remarks

After this command is executed on Kermit server, Kermit sends the command execution status back to MegaPipe and KCommandDone event is fired indicating whether this command is executed successfully or execution fails (an error message is passed back).

See Also

KCommandDone Event

2.4.1.7 KLogin Method

Performs a Kermit "Login" command.

[Visual Basic .NET]

```
Public Sub KLogin()
```

[C#]

```
public void KLogin();
```

Remarks

After this command is executed on Kermit server, Kermit sends the command execution status back to MegaPipe and KCommandDone event is fired indicating whether this command is executed successfully or execution fails (an error message is passed back).

See Also

KCommandDone Event

2.4.1.8 KLogout Method

Performs a Kermit "Logout" command.

[Visual Basic .NET]

```
Public Sub KLogout()
```

[C#]

```
public void KLogout();
```

Remarks

After this command is executed on Kermit server, Kermit sends the command execution status back to MegaPipe and KCommandDone event is fired indicating whether this command is executed successfully or execution fails (an error message is passed back).

See Also

KCommandDone Event

2.4.1.9 KRename Method

Performs a Kermit "Rename" command.

[Visual Basic .NET]

```
Public Sub KRename(ByVal OldFile As String, ByVal NewFile As String)
```

[C#]

```
public void KRename(string OldFile, string NewFile);
```

Parameters

OldFile

The relevant path name of the original file on Kermit server.

NewtFile

The relevant path name of the new file on Kermit server.

Remarks

After this command is executed on Kermit server, Kermit sends the command execution status back to MegaPipe and KCommandDone event is fired indicating whether this command is executed successfully or execution fails (an error message is passed back).

See Also

KCommandDone Event

2.4.2 Events

2.4.2.1 KCommandDone Event

This event is fired after one Kermit command is executed successfully or it fails

```
[Visual Basic .NET]
Public Event KCommandDone(ByVal Success As Boolean, ByVal Length As Integer, ByVal Data As Byte())
```

```
[C#]
public event KCommandDone(bool Success, int Length, byte[] Data);
```

Parameters

Success

Kermit command execution status flag, if it is true, execution is successful, otherwise execution fails.

Length

Length for array of bytes for the data stream about the Kermit command execution status sent back from Kermit server.

Data

Array of bytes for the data stream about the Kermit command execution status sent back from Kermit server.

Remarks

You can obtain the Kermit command execution status via this event.

2.5 Common Events

2.5.1 HasError Event

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

[Visual Basic .NET]

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

[C#]

```
public event HasError(string ErrorMsg);
```

Parameters

ErrorMsg

Fatal error message.

2.5.2 HasWarning Event

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

[Visual Basic .NET]

```
Public Event HasWarning(ByVal WarningMsg As String)
```

[C#]

```
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.