

Table of Contents

Foreword	0
Part I About MegaPipe .NET CF Class Library	3
1 Introduction.....	3
2 How to Use It.....	3
Trial Version	3
Full Version	4
3 How to Distribute It.....	6
Part II Reference Guide	6
1 Serial Communication.....	6
Properties	6
AvailDataCount Property.....	6
BaudRate Property.....	6
CDStatus Property.....	7
CTSSStatus Property.....	7
DataBits Property.....	7
DSRStatus Property.....	7
Flow Control Property.....	8
InputData Property.....	8
InputQ Property.....	8
IsConnected Property.....	8
OutputData Property.....	9
OutputQ Property.....	9
OutputStringData Property.....	9
Parity Property.....	9
Port Property.....	10
PortOpen Property.....	10
RingStatus Property.....	10
RThreshold Property.....	10
StopBits Property.....	11
XoffChar Property.....	11
XonChar Property.....	11
Methods	11
CleanUp Method.....	11
EscapeCommFunc Method.....	12
Read Method.....	13
Events	13
CDChanged Event.....	13
CTSChanged Event.....	14
DSRChanged Event.....	14
OnComm Event.....	14
RingChanged Event.....	15
Enumerations	15
DataBits Enumeration.....	15
Flow Control Enumeration.....	15
Parity Enumeration.....	16
StopBits Enumeration.....	16

2	File Transfer.....	16
	Properties	16
	XferCurrBytes Property.....	16
	XferCurrFileName Property.....	16
	XferCurrFileSize Property.....	17
	XFerMode Property.....	17
	XFerProtocol Property.....	17
	XFerStart Property.....	17
	XFerStop Property.....	18
	XFerWorkDir Property.....	18
	Methods	18
	XferAddFile Method.....	18
	XferClearAllFiles Method.....	19
	XferSetDstFile Method.....	19
	Events	19
	AbortTransfer Event.....	19
	DoInit Event.....	20
	FinishFile Event.....	20
	FinishTransfer Event.....	20
	OneBlockDone Event.....	20
	StartFile Event.....	21
	Enumerations	21
	Action Enumeration.....	21
	Protocol Enumeration.....	22
3	Common Events.....	22
	HasError Event	22
	HasWarning Event	22
	Part III License	23
	Index	0

1 About MegaPipe .NET CF Class Library

1.1 Introduction

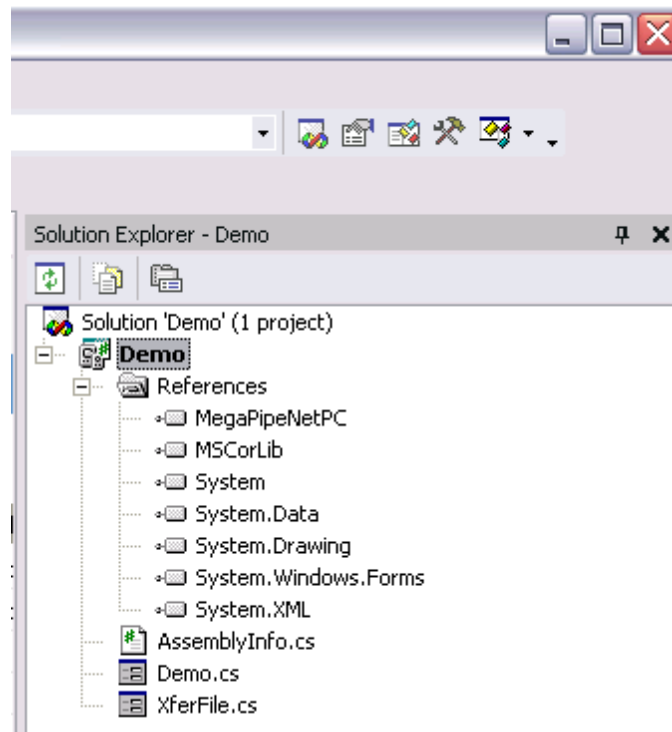
MegaPipe .NET compact framework's class library is a reliable and powerful 100% managed code component for handling serial communication and file-transfer (XModem Checksum, XModem CRC, XModem 1K, YModem, YModem-G, ZModem and Kermit). It supports Pocket PC and Windows CE and can be used in VB.NET and C#.NET.

1.2 How to Use It

1.2.1 Trial Version

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

1. Complete the installation for the trial version package.
 2. If you develop a Windows CE application, copy MegaPipeNetCE.dll in VB.NET or C# demo project sub folder in the destination folder (e.g. "C:\Program Files\MW6 .NET Components\MegaPipe CF\Trial Version\2003\Windows CE" or "C:\Program Files\MW6 .NET Components\MegaPipe CF\Trial Version\2005\Windows CE") to your project folder.
 3. If you develop a Pocket PC application, copy MegaPipeNetPC.dll in VB.NET or C# demo project sub folder in the destination folder (e.g. "C:\Program Files\MW6 .NET Components\MegaPipe CF\Trial Version\2003\Pocket PC" or "C:\Program Files\MW6 .NET Components\MegaPipe CF\Trial Version\2005\Pocket PC") to your project folder.
 4. In the "Solution Explorer" windows for your project, right click "References" and "Add Reference" for MegaPipeNetCE or MegaPipeNetPC depending on the target platform.
-



5. Now your project has access to the MegaPipe .NET CF class library, please take a look at our demo projects for more coding details.

1.2.2 Full Version

Follow the instructions listed below to add the full version MegaPipe .NET CF class library to your project:

1. Uninstall the trial version package and complete the installation for the full version package.
 2. If you develop a Windows CE application, copy MegaPipeNetCE.dll in VB.NET or C# demo project sub folder in the destination folder (e.g. "C:\Program Files\MW6 .NET Components\MegaPipe CF\Trial Version\2003\Windows CE" or "C:\Program Files\MW6 .NET Components\MegaPipe CF\Trial Version\2005\Windows CE") to your project folder to replace the trial version MegaPipeNetCE.dll.
 3. If you develop a Pocket PC application, copy MegaPipeNetPC.dll in VB.NET or C# demo project sub folder in the destination folder (e.g. "C:\Program Files\MW6 .NET Components\MegaPipe CF\Trial Version\2003\Pocket PC" or "C:\Program Files\MW6 .NET Components\MegaPipe CF\Trial Version\2005\Pocket PC") to your project folder to replace the trial version MegaPipeNetPC.dll.
 4. The license key method SetKey should be called once before you call other MegaPipe functions.
-

[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]

```
' Code for Pocket PC
Dim ValidKey As Boolean
MegaPipeNet1 = New MegaPipeNetPC.PocketPC()
ValidKey = MegaPipeNet1.SetKey("XXXXXX-XXXX")

MegaPipeNet1.BaudRate = 9600
MegaPipeNet1.Parity = MegaPipeNetPC.PocketPC.enumParity.pNone
MegaPipeNet1.DataBits = MegaPipeNetPC.PocketPC.enumDataBits.db8Bits
MegaPipeNet1.StopBits = MegaPipeNetPC.PocketPC.enumStopBits.sb1Bit
MegaPipeNet1.FlowControl = MegaPipeNetPC.PocketPC.enumFlowControl.fcNone
MegaPipeNet1.InputQ = 16384
MegaPipeNet1.OutputQ = 16384
MegaPipeNet1.RThreshold = 0
```

[C#]

```
// Code for Pocket PC
bool ValidKey;
MegaPipeNet1 = new MegaPipeNetPC.PocketPC();
ValidKey = MegaPipeNet1.SetKey("XXXXXX-XXXX");

MegaPipeNet1.BaudRate = 9600;
MegaPipeNet1.Parity = MegaPipeNetPC.PocketPC.enumParity.pNone;
MegaPipeNet1.DataBits = MegaPipeNetPC.PocketPC.enumDataBits.db8Bits;
MegaPipeNet1.StopBits = MegaPipeNetPC.PocketPC.enumStopBits.sb1Bit;
```

```
MegaPipeNet1.FlowControl = MegaPipeNetPC.PocketPC.enumFlowControl.fcNone;  
MegaPipeNet1.InputQ = 16384;  
MegaPipeNet1.OutputQ = 16384;  
MegaPipeNet1.RThreshold = 0;
```

1.3 How to Distribute It

If you want to redistribute MegaPipe .NET CF class library as part of your application, simply put MegaPipeNetCE.dll or MegaPipeNetPC.dll depending on the target platform 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;}
```

2.1.1.3 CDStatus Property

Gets 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 CTSSStatus Property

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

[Visual Basic .NET]

```
Public ReadOnly Property CTSSStatus As Boolean
```

[C#]

```
public bool CTSSStatus {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;}
```

2.1.1.6 DSRStatus Property

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

[Visual Basic .NET]

```
Public Property FlowControl As enumFlowControl
```

[C#]

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

2.1.1.8 InputData Property

Returns 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;}
```

2.1.1.10 IsConnected Property

Indicates whether a serial line 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 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;}
```

2.1.1.13 OutputStringData Property

Sets 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 type of the parity check for the serial communication.

[Visual Basic .NET]

```
Public Property Parity As enumParity
```

[C#]

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

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;}
```

2.1.1.16 PortOpen Property

This write-only 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 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 a 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;}
```

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;}
```

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;}
```

2.1.2 Methods

2.1.2.1 CleanUp Method

Performs cleanup to make sure that threads inside MegaPipe .NET CF class library are properly terminated.

[Visual Basic .NET]

```
Public Sub CleanUp()
```

[C#]

```
public void CleanUp();
```

Remarks

Call this method before your application exits.

2.1.2.2 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 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.

2.1.2.3 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 CTSChecked Event

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

[Visual Basic .NET]

```
Public Event CTSChecked()
```

[C#]

```
public event CTSChecked();
```

2.1.3.3 DSRChecked Event

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

[Visual Basic .NET]

```
Public Event DSRChecked()
```

[C#]

```
public event DSRChecked();
```

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 File Transfer

2.2.1 Properties

2.2.1.1 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;}
```

2.2.1.2 XferCurrFileName Property

Returns the name of file being transferred.

[Visual Basic .NET]

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

[C#]

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

2.2.1.3 XferCurrFileSize Property

Returns the size of file being transferred.

[Visual Basic .NET]

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

[C#]

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

2.2.1.4 XFerMode Property

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

[Visual Basic .NET]

```
Public Property XferMode As enumAction
```

[C#]

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

2.2.1.5 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;}
```

2.2.1.6 XFerStart Property

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

[Visual Basic .NET]

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

[C#]

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

2.2.1.7 XFerStop Property

Determines 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.

2.2.1.8 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;}
```

2.2.2 Methods

Enter topic text here.

2.2.2.1 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.

2.2.2.2 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.

2.2.2.3 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.

2.2.3 Events

2.2.3.1 AbortTransfer Event

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

[Visual Basic .NET]

```
Public Event AbortTransfer()
```

[C#]

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

2.2.3.2 Dolnit Event

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

[Visual Basic .NET]

```
Public Event Dolnit()
```

[C#]

```
public event Dolnit();
```

2.2.3.3 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.

2.2.3.4 FinishTransfer Event

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

[Visual Basic .NET]

```
Public Event FinishTransfer()
```

[C#]

```
public event FinishTransfer();
```

2.2.3.5 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 `XferCurrBytes` property to obtain the information about how many bytes of data have been sent or received successfully so far.

See Also

`XferCurrBytes` Property

2.2.3.6 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.

2.2.4 Enumerations

2.2.4.1 Action Enumeration

An enumeration type for all possible `XferMode` values.

Members

Name	Comment
aUpload	Upload file
aDownload	Download file

2.2.4.2 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

2.3 Common Events

2.3.1 HasError Event

This event is fired when a fatal error occurs for serial communication or file transfer process.

[Visual Basic .NET]

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

[C#]

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

Parameters

ErrorMessage

Fatal error message.

2.3.2 HasWarning Event

This event is fired when a non-fatal warning pops up for serial communication or file transfer process.

[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.