

```
'Copyright 2006 Dan Morgan - VisuMAX Inc.
Imports Microsoft.VisualBasic
Imports System
Imports System.Windows.Forms
Imports System.IO
Imports System.Diagnostics

Public Class frmBinaryCoder
    Public MyBin As String
    Public MyAscii As String
    Public MyAsciiStream As String
    Public MyBinaryStream As String
    Private WithEvents docToPrint As New Printing.PrintDocument
    Public MyAsciiPrint As String

    Private Function DecToBin(ByVal dIn As Double) As String
        Dim StrLen As Integer

        DecToBin = ""
        While dIn >= 1
            DecToBin = IIf(dIn Mod 2 = 0, "0", "1") & DecToBin
            dIn = dIn \ 2
        End While

        StrLen = Len(DecToBin) 'Ensure our binary number is 8 bits
        While StrLen <> 8
            DecToBin = 0 & DecToBin
            StrLen = StrLen + 1
        End While

        MyBin = MyBin & DecToBin
    End Function

    Private Function BinToDec(ByVal sIn As String) As Double
        Dim x As Integer
        Dim MyChar As String

        BinToDec = 0
        For x = 1 To Len(sIn)
            BinToDec = BinToDec + (CInt(Mid(sIn, x, 1)) * (2 ^ (Len(sIn) - x)))
        Next x

        MyChar = Chr(BinToDec)
        MyAscii = MyAscii & MyChar
    End Function

    Private Sub EncodeTextToBinary()
        Dim MyCharInt As Integer
        Dim StrLen As Integer
        Dim i As Integer = 0
        Dim MyChar As String

        MyBin = ""
        txtBinary.Text = ""

        StrLen = Len(txtAscii.Text)
        While i <> StrLen
            i = i + 1
            MyChar = Microsoft.VisualBasic.Mid(txtAscii.Text, i, 1)

            MyCharInt = Asc(MyChar)
            DecToBin(MyCharInt)
        End While
    End Sub
End Class
```

```
txtBinary.Text = MyBin
End Sub
```

```
Private Sub DecodeBinaryToText()
    Dim StrLen As Integer
    Dim i As Integer = 0
    Dim j As Integer = 0
    Dim k As Integer = 1
    Dim MyBinary As String
    Dim StrSearch As String = Chr(13)
    Dim StrSearch2 As String = " "
    Dim StrPosit As Integer
    Dim NewBinaryString As String = ""
    Dim NewBinaryString2 As String = ""
    Dim BinaryTest As Integer = 8
    Dim TestValue As Double
```

```
StrLen = Len(txtBinary.Text)
```

```
MyAscii = ""
txtAscii.Text = ""
'All LF and CF removed
```

```
While j <> StrLen
    j = j + 1
```

```
StrPosit = InStr(k, txtBinary.Text, StrSearch)
```

```
If StrPosit = 0 Then
```

```
    NewBinaryString = NewBinaryString & Microsoft.VisualBasic.Mid(txtBinary.
Text, j, StrLen + 1 - j)
    Exit While
```

```
End If
```

```
k = StrPosit
```

```
    NewBinaryString = NewBinaryString & Microsoft.VisualBasic.Mid(txtBinary.Text,
j, k - j)
```

```
k = k + 1
```

```
j = k
```

```
End While
```

```
'Remove all spaces
```

```
NewBinaryString = Microsoft.VisualBasic.Trim(NewBinaryString)
```

```
StrLen = Len(NewBinaryString)
```

```
k = 1
```

```
j = 0
```

```
While j < StrLen
```

```
    j = j + 1
```

```
StrPosit = InStr(k, NewBinaryString, StrSearch2)
```

```
If StrPosit = 0 Then
```

```
    NewBinaryString2 = NewBinaryString2 & Microsoft.VisualBasic.Mid
(NewBinaryString, j, StrLen + 1 - j)
```

```
    Exit While
```

```
End If
```

```
k = StrPosit
```

```
    NewBinaryString2 = NewBinaryString2 & Microsoft.VisualBasic.Mid
(NewBinaryString, j, k - j)
```

```
k = k + 1
```

```
j = k - 1
```

```
End While
```

```
StrLen = Len(NewBinaryString2)
```

```
TestValue = StrLen Mod BinaryTest 'Test for a good binary string
```

```
If TestValue = 0 Then
```

```
    While i <> StrLen
```

```
        i = i + 1
```

```
        MyBinary = Microsoft.VisualBasic.Mid(NewBinaryString2, i, 8)
```

```

        i = i + 7
        BinToDec(MyBinary)
    End While
    txtAscii.Text = MyAscii

```

```

Else
    MsgBox("Encoded Message is incomplete! Unable to Decode!", MsgBoxStyle.
    Exclamation, "Decoding Error")

```

```

End If
End Sub

```

```

Private Sub SendEncodedTextMail()

```

```

    Dim oOAPP As Outlook.Application
    Dim oOMail As Outlook.MailItem
    Dim Vendor As String
    Dim NoWrapString As String = ""
    Dim StrLen As Integer
    Dim MsgInfo As Integer

```

```

    Dim j As Integer = 1
    Dim k As Integer = 36

```

```

    If txtToMail.Text = "" Then
        MsgBox("You must add a recipient to send encoded message to.", MsgBoxStyle.
        Information, "No Email Recipient")
        Exit Sub

```

```

    ElseIf txtBinary.Text = "" Then
        MsgBox("You don't have any data encoded. Enter Text and click encode then try
        to send again.", MsgBoxStyle.Information, "No Data To Send")
        Exit Sub

```

```

    ElseIf txtSubjectMail.Text = "" Then
        MsgInfo = MsgBox("You are about to send mail without a subject. Do you wish to
        continue?", MsgBoxStyle.YesNo, "No Subject")

```

```

        If MsgInfo = 7 Then
            Exit Sub
        ElseIf MsgInfo = 6 Then
            StrLen = Len(txtBinary.Text)
            While j < StrLen
                NoWrapString = NoWrapString & Microsoft.VisualBasic.Mid(txtBinary.Text,
                j, 36) & Chr(13)
                j = j + 36
            End While

```

```

            Vendor = Chr(13) & "Encoded with BinaryCoder. VisuMAX.com" & Chr(13)
            Vendor = Vendor & "Copy the above binary into BinaryCoder to decoded the
            message." & Chr(13)
            Vendor = Vendor & "Download from: http://www.visumax.com/binarycoder/"

```

```

            oOAPP = CreateObject("Outlook.Application")
            oOMail = oOAPP.CreateItem(Outlook.OlItemType.olMailItem)

```

```

            With oOMail
                .To = txtToMail.Text
                .Subject = txtSubjectMail.Text
                .Body = NoWrapString & Vendor
                .Send()
            End With

```

```

        End If
    Else
        StrLen = Len(txtBinary.Text)
        While j < StrLen
            NoWrapString = NoWrapString & Microsoft.VisualBasic.Mid(txtBinary.Text, j,
            36) & Chr(13)
            j = j + 36

```

```
End While
```

```
Vendor = Chr(13) & "Encoded with BinaryCoder. VisuMAX.com" & Chr(13)
Vendor = Vendor & "Copy the above binary into BinaryCoder to decoded the
message." & Chr(13)
Vendor = Vendor & "Download from: http://www.visumax.com/binarycoder/"
```

```
oOAPP = CreateObject("Outlook.Application")
oOMail = oOAPP.CreateItem(Outlook.OlItemType.olMailItem)
```

```
With oOMail
    .To = txtToMail.Text
    .Subject = txtSubjectMail.Text
    .Body = NoWrapString & Vendor
    .Send()
End With
End If
```

```
End Sub
```

```
Private Sub Clear()
    txtToMail.Text = ""
    txtSubjectMail.Text = ""
    txtAscii.Text = ""
    txtBinary.Text = ""
    MyBin = ""
    MyAscii = ""
    MyAsciiPrint = ""

```

```
End Sub
```

```
Private Sub OpenStream()
```

```
Call Clear()
```

```
' Displays an OpenFileDialog so the user can select a textfile.
```

```
Dim objStreamReader As StreamReader
Dim strLine As String
Dim openFileDialog1 As New OpenFileDialog()
Dim FileExt As Integer
```

```
openFileDialog1.Filter = "Text Files (*.txt)|*.txt|Binary Files (*.bbn)|*.bbn"
openFileDialog1.Title = "Select a text|binary File"
```

```
' Show the Dialog.
' If the user clicked OK in the dialog and
' a .txt file was selected, open it.
If openFileDialog1.ShowDialog() = Windows.Forms.DialogResult.OK Then
```

```
FileExt = openFileDialog1.FilterIndex
```

```
If FileExt = 1 Then
```

```
' Assign the cursor in the Stream to the Form's Cursor property.
MyAsciiStream = openFileDialog1.FileName
If Not (MyAsciiStream Is Nothing) Then
    'Code to read the stream
```

```
'Pass the file path and the file name to the StreamReader constructor.
objStreamReader = New StreamReader(MyAsciiStream)
```

```
'Read the first line of text.
strLine = objStreamReader.ReadLine
```

```
'Continue to read until you reach the end of the file.
```

```
        Do While Not strLine Is Nothing

            'Write the line to the textbox and for possible printing.

            txtAscii.Text = txtAscii.Text & strLine
            MyAsciiPrint = MyAsciiPrint & strLine & Chr(13) & Chr(10)

            'Read the next line.
            strLine = objStreamReader.ReadLine
        Loop

        'Close the file.
        objStreamReader.Close()
    End If

    ElseIf FileExt = 2 Then
        MyBinaryStream = openFileDialog1.FileName
        If Not (MyBinaryStream Is Nothing) Then
            'Code to read the stream

            'Pass the file path and the file name to the StreamReader constructor.
            objStreamReader = New StreamReader(MyBinaryStream)

            'Read the first line of text.
            strLine = objStreamReader.ReadLine

            'Continue to read until you reach the end of the file.
            Do While Not strLine Is Nothing

                'Write the line to the textbox.

                txtBinary.Text = txtBinary.Text & strLine

                'Read the next line.
                strLine = objStreamReader.ReadLine
            Loop

            'Close the file.
            objStreamReader.Close()

        End If

    End If
End Sub

Private Sub SaveAsciiStream()
    ' Displays an OpenFileDialog so the user can select a textfile.

    Dim objStreamWriter As StreamWriter
    Dim SaveFileDialog1 As New SaveFileDialog()
    Dim FileExt As Integer
    Dim i As Integer
    Dim j As Integer
    Dim StrLen As Integer

    SaveFileDialog1.Filter = "Text Files (*.txt)|*.txt|Binary Files (*.bbn)|*.bbn"
    SaveFileDialog1.Title = "Select a text File"

    ' Show the Dialog.
    ' If the user clicked OK in the dialog and
    ' a .txt file was selected, open it.
    SaveFileDialog1.FileName = MyAsciiStream
```

```
    If SaveFileDialog1.ShowDialog() = Windows.Forms.DialogResult.OK Then

        'txt=1
        'bbn=2
        FileExt = SaveFileDialog1.FilterIndex

        If FileExt = 1 Then

            StrLen = Len(SaveFileDialog1.FileName)
            i = Microsoft.VisualBasic.InStr(StrLen - 4, SaveFileDialog1.FileName, ".
txt")

            ' Assign the cursor in the Stream to the Form's Cursor property.

            MyAsciiStream = SaveFileDialog1.FileName
            If Not (MyAsciiStream Is Nothing) Then
                'Code to read the stream

                'Pass the file path and the file name to the Streamreader constructor.
                objStreamWriter = New StreamWriter(MyAsciiStream)

                'Write a line of text.
                objStreamWriter.WriteLine(txtAscii.Text)

                'Close the file.
                objStreamWriter.Close()
            End If

        ElseIf FileExt = 2 Then

            StrLen = Len(SaveFileDialog1.FileName)
            j = Microsoft.VisualBasic.InStr(StrLen - 4, SaveFileDialog1.FileName, ".
bbn")

            ' Assign the cursor in the Stream to the Form's Cursor property.

            MyBinaryStream = SaveFileDialog1.FileName
            If Not (MyBinaryStream Is Nothing) Then
                'Code to read the stream

                'Pass the file path and the file name to the Streamreader constructor.
                objStreamWriter = New StreamWriter(MyBinaryStream)

                'Write a line of text.
                objStreamWriter.WriteLine(txtBinary.Text)

                'Close the file.
                objStreamWriter.Close()
            End If

        Else
            MsgBox("Please save with either the extension .txt for text file or .bbn
for binary file.", MsgBoxStyle.Information, "Add file extension")
        End If
    End If
End Sub

Private Sub butToBinary_Click(ByVal sender As System.Object, ByVal e As System.
EventArgs) Handles butToBinary.Click
    Call EncodeTextToBinary()
End Sub

Private Sub butToText_Click(ByVal sender As System.Object, ByVal e As System.
EventArgs) Handles butToText.Click
    Call DecodeBinaryToText()
End Sub
```

```
Private Sub butSendMail_Click(ByVal sender As System.Object, ByVal e As System.  
EventArgs) Handles butSendMail.Click  
    Call SendEncodedTextMail()  
End Sub  
  
Private Sub NewToolStripMenuItem_Click(ByVal sender As System.Object, ByVal e As  
System.EventArgs) Handles NewToolStripMenuItem.Click  
    Call Clear()  
End Sub  
  
Private Sub EncodeToolStripMenuItem_Click(ByVal sender As System.Object, ByVal e As  
System.EventArgs) Handles EncodeToolStripMenuItem.Click  
    Call EncodeTextToBinary()  
End Sub  
  
Private Sub DecodeToolStripMenuItem_Click(ByVal sender As System.Object, ByVal e As  
System.EventArgs) Handles DecodeToolStripMenuItem.Click  
    Call DecodeBinaryToText()  
End Sub  
  
Private Sub SendCodeToolStripMenuItem_Click(ByVal sender As System.Object, ByVal e As  
System.EventArgs) Handles SendCodeToolStripMenuItem.Click  
    Call SendEncodedTextMail()  
End Sub  
  
Private Sub OpenAsciiToolStripMenuItem_Click(ByVal sender As System.Object, ByVal e As  
System.EventArgs) Handles OpenAsciiToolStripMenuItem.Click  
    Call OpenStream()  
  
End Sub  
  
Private Sub SaveToolStripMenuItem_Click(ByVal sender As System.Object, ByVal e As  
System.EventArgs) Handles SaveToolStripMenuItem.Click  
  
    Call SaveAsciiStream()  
End Sub  
  
Private Sub PrintToolStripMenuItem_Click(ByVal sender As System.Object, ByVal e As  
System.EventArgs)  
  
    ' Allow the user to choose the page range he or she would  
    ' like to print.  
    'PrintDialog1.AllowSomePages = True  
  
    ' Show the help button.  
    'PrintDialog1.ShowHelp = True  
  
    ' Set the Document property to the PrintDocument for  
    ' which the PrintPage Event has been handled. To display the  
    ' dialog, either this property or the PrinterSettings property  
    ' must be set  
    'PrintDialog1.Document = docToPrint  
  
    'Dim result As DialogResult = PrintDialog1.ShowDialog()  
  
    ' If the result is OK then print the document.  
    'If (result = Windows.Forms.DialogResult.OK) Then  
    '    docToPrint.Print()  
    'End If  
  
End Sub  
  
' The PrintDialog will print the document  
' by handling the document's PrintPage event.  
'Private Sub document_PrintPage(ByVal sender As Object, ByVal e As System.Drawing.  
Printing.PrintPageEventArgs) Handles docToPrint.PrintPage
```

```
' ' Insert code to render the page here.
' ' This code will be called when the control is drawn.
```

```
' ' The following code will render a simple
' ' message on the printed document.
' Dim text As String = MyAsciiPrint
' Dim printFont As New System.Drawing.Font("Arial", 12, System.Drawing.FontStyle.
Regular)
```

```
' ' Draw the content.
' e.Graphics.DrawString(text, printFont, System.Drawing.Brushes.Black, 10, 10)
'End Sub
```

```
Private Sub CopyToolStripMenuItem_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles CopyToolStripMenuItem.Click
    If txtAscii.Focused = True Then
        If txtAscii.SelectionLength > 0 Then
            txtAscii.Copy()
        End If
    ElseIf txtBinary.Focused = True Then
        If txtBinary.SelectionLength > 0 Then
            txtBinary.Copy()
        End If
    End If
End Sub
```

```
Private Sub ExitToolStripMenuItem_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles ExitToolStripMenuItem.Click
    Me.Close()
End Sub
```

```
Private Sub PasteToolStripMenuItem_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles PasteToolStripMenuItem.Click
    ' Determine if there is any text in the Clipboard to paste into the text box.
    If Clipboard.GetDataObject().GetDataPresent(DataFormats.Text) = True Then
        ' Determine if any text is selected in the text box.
        If txtAscii.Focused = True Then
            If txtAscii.SelectionLength > 0 Then
                ' Ask user if they want to paste over currently selected text.
                If MessageBox.Show("Do you want to paste over current selection?", _
                    "Cut", MessageBoxButtons.YesNo) = Windows.Forms.DialogResult.No
                Then
                    ' Move selection to the point after the current selection and
                    paste.
                    txtAscii.SelectionStart = txtAscii.SelectionStart + txtAscii.
                    SelectionLength
                End If
            End If
            ' Paste current text in Clipboard into text box.
            txtAscii.Paste()
        ElseIf txtBinary.Focused = True Then
            If txtBinary.SelectionLength > 0 Then
                ' Ask user if they want to paste over currently selected text.
                If MessageBox.Show("Do you want to paste over current selection?", _
                    "Cut", MessageBoxButtons.YesNo) = Windows.Forms.DialogResult.No
                Then
                    ' Move selection to the point after the current selection and
                    paste.
                    txtBinary.SelectionStart = txtBinary.SelectionStart + txtBinary.
                    SelectionLength
                End If
            End If
            ' Paste current text in Clipboard into text box.
            txtBinary.Paste()
        End If
    End If
```

```
End If
```

```
End Sub
```

```
Private Sub CutToolStripMenuItem_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles CutToolStripMenuItem.Click
```

```
    ' Ensure that text is currently selected in the text box.
```

```
    If txtAscii.Focused = True Then
```

```
        If txtAscii.SelectedText <> "" Then
```

```
            ' Cut the selected text in the control and paste it into the Clipboard.  
            txtAscii.Cut()
```

```
        End If
```

```
    ElseIf txtBinary.Focused = True Then
```

```
        ' Ensure that text is currently selected in the text box.
```

```
        If txtBinary.SelectedText <> "" Then
```

```
            ' Cut the selected text in the control and paste it into the Clipboard.  
            txtBinary.Cut()
```

```
        End If
```

```
End If
```

```
End Sub
```

```
Private Sub UndoToolStripMenuItem_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles UndoToolStripMenuItem.Click
```

```
    ' Determine if last operation can be undone in text box.
```

```
    If txtAscii.Focused = True Then
```

```
        If txtAscii.CanUndo = True Then
```

```
            ' Undo the last operation.
```

```
            txtAscii.Undo()
```

```
            ' Clear the undo buffer to prevent last action from being redone.
```

```
            txtAscii.ClearUndo()
```

```
        End If
```

```
    ElseIf txtBinary.Focused = True Then
```

```
        If txtBinary.CanUndo = True Then
```

```
            ' Undo the last operation.
```

```
            txtBinary.Undo()
```

```
            ' Clear the undo buffer to prevent last action from being redone.
```

```
            txtBinary.ClearUndo()
```

```
        End If
```

```
End If
```

```
End Sub
```

```
Private Sub HelpBinToolStripMenuItem_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles HelpBinToolStripMenuItem.Click
```

```
    ' Call the Process.Start method to open the default browser
```

```
    ' with a URL:
```

```
    System.Diagnostics.Process.Start("http://www.visumax.com/binarycoder/")
```

```
End Sub
```

```
Private Sub SelectAllToolStripMenuItem_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles SelectAllToolStripMenuItem.Click
```

```
    If txtAscii.Focused = True Then
```

```
        txtAscii.SelectAll()
```

```
    ElseIf txtBinary.Focused = True Then
```

```
        txtBinary.SelectAll()
```

```
End If
```

```
End Sub
```

```
Private Sub AboutToolStripMenuItem_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles AboutToolStripMenuItem.Click
```

```
    Dim frmaboutbox As New AboutBox
```

```
    frmaboutbox.Show()
```

```
End Sub  
End Class
```