Private Type Rect

Left As Long

Top As Long

Right As Long

Bottom As Long

End Type

Private Type POINTAPI

X As Long

Y As Long

End Type

Private Type PageSetupDlg

lStructSize As Long

hwndOwner As Long

hDevMode As Long

hDevNames As Long

FLAGS As Long

ptPaperSize As POINTAPI

rtMinMargin As Rect

rtMargin As Rect

hInstance As Long

lCustData As Long

lpfnPageSetupHook As Long

lpfnPagePaintHook As Long

lpPageSetupTemplateName As String

hPageSetupTemplate As Long

End Type

Private Declare Function PageSetupDlg Lib "comdlg32.dll" Alias "PageSetupDlgA" \_

(pPagesetupdlg As PageSetupDlg) As Long

Private Declare Function StretchBlt Lib "gdi32" (ByVal hdc As Long, ByVal X As Long, \_

ByVal Y As Long, ByVal mDestWidth As Long, ByVal mDestHeight As Long, \_

ByVal hSrcDC As Long, ByVal xSrc As Long, ByVal ySrc As Long, ByVal mSrcWidth As Long, \_

ByVal mSrcHeight As Long, ByVal dwRop As Long) As Long

Private Const SRCCOPY = &HCC0020

Private Type CharRange

firstChar As Long

lastChar As Long

End Type

Private Type FormatRange

hdc As Long

hdcTarget As Long

rectRegion As Rect

rectPage As Rect

mCharRange As CharRange

End Type

Private Const WM\_USER As Long = &H400

Private Const EM\_FORMATRANGE As Long = WM\_USER + 57

Private Const EM\_SETTARGETDEVICE As Long = WM\_USER + 72

Private Declare Function SendMessage Lib "User32" Alias "SendMessageA" (ByVal hwnd As Long, ByVal msg As Long, ByVal wp As Long, Ip As Any) As Long

Dim mFormatRange As FormatRange

Dim rectDrawTo As Rect

Dim rectPage As Rect

Dim TextLength As Long

Dim newStartPos As Long

Dim dumpaway As Long

Private Declare Function CreateDC Lib "gdi32" Alias "CreateDCA" \_

(ByVal lpDriverName As String, ByVal lpDeviceName As String, \_

ByVal lpOutput As Long, ByVal lpInitData As Long) As Long

Dim mNotShow As Boolean

Dim mSizeNo As Integer

Dim mTotalPages As Integer

Public gLeftMargin As Integer

Public gRightMargin As Integer

Public gTopMargin As Integer

Public gBottomMargin As Integer

Public gprint As Boolean

Dim a

Dim b

Dim c

Dim D

Private Sub cmdClose\_Click()

Unload Me

frmMain1.SetFocus

frmMain1.Show

End Sub

Private Sub Form\_Load()

'''''''''''''

DoEvents

Set preview.picP.Picture = LoadPicture

Dim mydialog1 As Object

Dim mFromPage As Integer, mToPage As Integer, mpage As Integer

Set mydialog1 = frmMain1.dlgCommonDialog

mydialog1.DialogTitle = "Print"

mydialog1.CancelError = True

mydialog1.FLAGS = cdlPDReturnDC + cdlPDPageNums

If frmMain1.RTBox.SelLength = 0 Then

mydialog1.FLAGS = mydialog1.FLAGS + cdlPDAllPages

Else

mydialog1.FLAGS = mydialog1.FLAGS + cdlPDSelection

End If

'mydialog1.ShowPrinter

If err = MSComDlg.cdlCancel Then

Exit Sub

End If

Dim PageSetup As PageSetupDlg

With PageSetup

.lStructSize = Len(PageSetup)

.hwndOwner = hwnd

PageSetupDlg PageSetup

a = .rtMargin.Left / 1000

c = .rtMargin.Right / 1000

b = .rtMargin.Top / 1000

D = .rtMargin.Bottom / 1000

preview.LabLeft = "Left Margin = " & a

preview.LabRight = "Right Margin = " & c

preview.LabTop = "Top Margin = " & b

preview.LabBottom = "Bottom Margin = " & D

preview.LabPageWidth = "Paper Width = " & .ptPaperSize.X / 1000

preview.LabPageHeight = "Paper Height = " & .ptPaperSize.Y / 1000

End With

'''''''''

With frmMain1.RTBox

' variables to store new values

Dim sglLeft As Single

Dim sglRight As Single

' here the variables get set.

' I go through some extra steps to make sure

' I do not lose precision when converting

' integers to floatig-point values.

sglLeft = .SelIndent

sglLeft = sglLeft / 1440#

sglLeft = CInt(sglLeft \* 100#)

sglLeft = sglLeft / 100#

preview.txtLeft.Text = Trim(str(sglLeft))

sglRight = .SelRightIndent

sglRight = sglRight / 1440#

sglRight = CInt(sglRight \* 100#)

sglRight = sglRight / 100#

preview.txtRight.Text = Trim(str(sglRight))

End With

'preview.txtleft.Text = A

'preview.txtRight.Text = C

'preview.txtTop.Text = B

'preview.txtBottom.Text = D

If Val(preview.txtLeft.Text) < 0 Then

Beep 'beep..beep..beep

' show error

MsgBox "Left margin must be no less than zero inches.", 16, "Margin Out Of Range"

'select box with error

preview.txtLeft.SelStart = 0

preview.txtLeft.SelLength = Len(preview.txtLeft.Text)

'set focus to box

'txtLeft.SetFocus

ElseIf Val(preview.txtRight.Text) < 0 Then

Beep

' show error message

MsgBox "Right margin must be no less than zero inches.", 16, "Margin Out Of Range"

'select text with error

preview.txtRight.SelStart = 0

preview.txtRight.SelLength = Len(preview.txtRight.Text)

'set focus to text

'txtRight.SetFocus

Else

'passed checks

' variables to store old selection

Dim lngOldStart As Long

Dim lngOldLength As Long

With frmMain1.RTBox

' save old text selection

lngOldStart = .SelStart

lngOldLength = .SelLength

' select entire document

.SelStart = 0

.SelLength = Len(.Text)

' set new margins

' the value needs to be converted to twips

' for acuracy. There are 1440 Twips/Inch.

.SelIndent = CInt(Val(preview.txtLeft.Text) \* 1440)

.SelRightIndent = CInt(Val(preview.txtRight.Text) \* 1440)

' restore old selection

.SelStart = lngOldStart

.SelLength = lngOldLength

End With

End If

'''''''''

Screen.MousePointer = vbHourglass

gprint = False

PicZ.AutoSize = False

picP.AutoSize = False

PicX.AutoSize = False

Pic1.AutoSize = False

Pic2.AutoSize = False

Dim mNormalWidth, mNormalHeight

Dim mAdjFactor

Dim mRect, mNewRect, mfactor

'Dim mpage As Integer

DocWYSIWYG frmMain1.ActiveControl

mNormalWidth = Printer.ScaleWidth

mNormalHeight = Printer.ScaleHeight

mAdjFactor = 50 / 100

mNormalWidth = mNormalWidth \* mAdjFactor

mNormalHeight = mNormalHeight \* mAdjFactor

mRect = mNormalWidth \* mNormalHeight

PicX.Width = 12240

PicX.Height = 16200

mNewRect = mRect \* (50 / 100)

mfactor = Sqr(mNewRect / (mNormalWidth \* mNormalHeight))

Pic1.Width = CInt(mNormalWidth \* mfactor)

Pic1.Height = CInt(mNormalHeight \* mfactor)

Pic2.Width = PicX.Width \* 0.7

Pic2.Height = PicX.Height \* 0.7

preview.ScaleMode = vbPixels

PicZ.ScaleMode = vbPixels

PicX.ScaleMode = vbPixels

Pic1.ScaleMode = vbPixels

Pic2.ScaleMode = vbPixels

PicZ.AutoRedraw = True

picP.AutoRedraw = True

PicX.AutoRedraw = True

Pic1.AutoRedraw = True

Pic2.AutoRedraw = True

PicZ.BorderStyle = 1

PicX.BorderStyle = 1

Pic1.BorderStyle = 1

Pic2.BorderStyle = 1

PicZ.FillStyle = 1

picP.FillStyle = 1

PicX.FillStyle = 1

Pic1.FillStyle = 1

Pic2.FillStyle = 1

mTotalPages = PageCtnProc(preview.PicX)

txtTotalPages.Text = "Total " & CStr(mTotalPages) & " page(s)"

setPageButtons

Dim i As Integer

cboPageNo.Clear

For i = 1 To mTotalPages

cboPageNo.AddItem i

Next i

cboPageNo.Text = cboPageNo.List(0)

VScroll1.Max = 1000

HScroll1.Max = 1000

cboScale.AddItem "150"

cboScale.AddItem "100"

cboScale.Text = cboScale.List(1)

mpage = 1

FormPreviewPage preview.PicX, mpage

For i = 1 To 5

DoEvents

If MakeSizes(i) = False Then

Screen.MousePointer = vbDefault

Exit Sub

End If

Next

Screen.MousePointer = vbDefault

PicZ.Visible = True

picP.Visible = False

PicX.Visible = False

mNotShow = False

mSizeNo = 1

ChangePreview

End Sub

Private Sub cboPageNo\_click()

Dim mpage As Integer

mpage = cboPageNo.ListIndex + 1

setPageButtons

Screen.MousePointer = vbHourglass

FormPreviewPage preview.PicX, mpage

Dim i

For i = 1 To 2

DoEvents

If MakeSizes(i) = False Then

Screen.MousePointer = vbDefault

Exit Sub

End If

Next

If mSizeNo = 1 Then

mSizeNo = 2

mNotShow = True

ChangePreview

mNotShow = False

mSizeNo = 1

ChangePreview

Else

mSizeNo = mSizeNo - 1

mNotShow = True

ChangePreview

mNotShow = False

mSizeNo = mSizeNo + 1

ChangePreview

End If

Screen.MousePointer = vbDefault

End Sub

Private Sub cmdPrevPage\_Click()

If mTotalPages = 1 Then

Exit Sub

Else

If Val(cboPageNo.Text) > 1 Then

cboPageNo.Text = cboPageNo.List(cboPageNo.ListIndex - 1)

cboPageNo\_click

End If

End If

ChangePreview

End Sub

Private Sub cmdNextPage\_Click()

If mTotalPages = 1 Then

Exit Sub

Else

If Val(cboPageNo.Text) < mTotalPages Then

cboPageNo.Text = cboPageNo.List(cboPageNo.ListIndex + 1)

cboPageNo\_click

End If

End If

ChangePreview

End Sub

Private Sub setPageButtons()

If mTotalPages = 1 Then

cmdPrevPage.Enabled = False

cmdNextPage.Enabled = False

Else

If Val(cboPageNo.Text) = 1 Then

cmdPrevPage.Enabled = False

cmdNextPage.Enabled = True

ElseIf Val(cboPageNo.Text) = mTotalPages Then

cmdPrevPage.Enabled = True

cmdNextPage.Enabled = False

Else

cmdPrevPage.Enabled = True

cmdNextPage.Enabled = True

End If

End If

End Sub

Private Sub HScroll1\_Change()

Select Case mSizeNo

Case 1

Pic1.Left = -HScroll1.Value

Case 2

Pic2.Left = -HScroll1.Value

End Select

End Sub

Private Sub Pic1\_Click()

Select Case mSizeNo

Case 2

mSizeNo = 2

cboScale.Text = cboScale.List(1)

End Select

If mSizeNo = 1 Then

mSizeNo = 2

cboScale.Text = cboScale.List(0)

End If

ChangePreview

End Sub

Private Sub Pic2\_Click()

Select Case mSizeNo

Case 2

mSizeNo = 1

cboScale.Text = cboScale.List(0)

End Select

If mSizeNo = 2 Then

mSizeNo = 1

cboScale.Text = cboScale.List(1)

End If

ChangePreview

End Sub

Private Sub VScroll1\_Change()

Select Case mSizeNo

Case 1

Pic1.Top = -VScroll1.Value

Case 2

Pic2.Top = -VScroll1.Value

End Select

End Sub

Private Sub ChangePreview()

Select Case mSizeNo

Case 1

Pic1.Visible = True

Pic2.Visible = False

Case 2

Pic1.Visible = False

Pic2.Visible = True

End Select

End Sub

Private Sub cboScale\_Click()

Select Case cboScale.Text

Case "150"

mSizeNo = 2

Pic2.Visible = False

Case "100"

mSizeNo = 1

Pic1.Visible = False

End Select

ChangePreview

End Sub

Private Sub cmdPrint\_Click()

cd.ShowPrinter

'PrintRTF frmMain1.RTBox, a \* 1440, b \* 1440, c \* 1440, D \* 1440 ' 1440 Twips = 1 Inch

preview.DocPrintProc

gprint = True

End Sub

Private Function MakeSizes(ByVal mofSize As Integer) As Boolean

Dim SrcX As Long, SrcY As Long

Dim DestX As Long, DestY As Long

Dim SrcWidth As Long, SrcHeight As Long

Dim DestWidth As Long, DestHeight As Long

Dim SrcHDC As Long, DestHDC As Long

Dim mresult

SrcX = 0: SrcY = 0: DestX = 0: DestY = 0

SrcWidth = PicX.ScaleWidth

SrcHeight = PicX.ScaleHeight

SrcHDC = PicX.hdc

Select Case mofSize

Case 1

DestWidth = Pic1.ScaleWidth

DestHeight = Pic1.ScaleHeight

DestHDC = Pic1.hdc

Case 2

DestWidth = Pic2.ScaleWidth

DestHeight = Pic2.ScaleHeight

DestHDC = Pic2.hdc

End Select

mresult = StretchBlt(DestHDC, DestX, DestY, DestWidth, DestHeight, SrcHDC, \_

SrcX, SrcY, SrcWidth, SrcHeight, SRCCOPY)

If mresult = 0 Then

MakeSizes = False

Else

MakeSizes = True

End If

End Function

Function DocWYSIWYG(RTB As Control) As Long

Dim LeftMargin As Long, RightMargin As Long

Dim LineWidth As Long

Dim PrinterhDC As Long

Dim r As Long

Printer.ScaleMode = vbTwips

LeftMargin = gLeftMargin \* 1440

RightMargin = Printer.Width - gRightMargin \* 1440

LineWidth = RightMargin - LeftMargin

DocWYSIWYG = LineWidth

End Function

Sub FormPreviewPage(inControl As Control, InPage As Integer)

Dim PageCtn

Set inControl.Picture = LoadPicture

rectPage.Left = 0

rectPage.Top = 0

rectPage.Right = inControl.Width \* Screen.TwipsPerPixelX

rectPage.Bottom = inControl.Height \* Screen.TwipsPerPixelY

rectDrawTo.Left = gLeftMargin \* 1440 + 1440 \* a

rectDrawTo.Top = gTopMargin \* 1440 + 1440 \* b

rectDrawTo.Right = inControl.Width \* Screen.TwipsPerPixelX \_

- gRightMargin \* 1440 - 1440 \* c

rectDrawTo.Bottom = inControl.Height \* Screen.TwipsPerPixelY \_

- gBottomMargin \* 1440 - 1440 \* D

mFormatRange.hdc = inControl.hdc

mFormatRange.hdcTarget = inControl.hdc

mFormatRange.rectRegion = rectDrawTo

mFormatRange.rectPage = rectPage

mFormatRange.mCharRange.firstChar = 0

mFormatRange.mCharRange.lastChar = -1

TextLength = Len(frmMain1.ActiveControl.Text)

PageCtn = 1

Do

newStartPos = SendMessage(frmMain1.ActiveControl.hwnd, EM\_FORMATRANGE, True, mFormatRange)

If newStartPos >= TextLength Then

Exit Do

End If

If PageCtn = InPage Then

Exit Do

End If

Set inControl.Picture = LoadPicture

mFormatRange.mCharRange.firstChar = newStartPos

mFormatRange.hdc = inControl.hdc

mFormatRange.hdcTarget = inControl.hdc

PageCtn = PageCtn + 1

DoEvents

Loop

dumpaway = SendMessage(inControl.hwnd, EM\_FORMATRANGE, False, ByVal CLng(0))

End Sub

Function PageCtnProc(inControl As Control) As Integer

On Error GoTo errhandler

Dim mPageCtn As Integer

rectPage.Left = 0

rectPage.Top = 0

rectPage.Right = inControl.Width \* Screen.TwipsPerPixelX

rectPage.Bottom = inControl.Height \* Screen.TwipsPerPixelY

rectDrawTo.Left = gLeftMargin \* 1440 + 1440 \* a

rectDrawTo.Top = gTopMargin \* 1440 + 1440 \* b

rectDrawTo.Right = inControl.Width \* Screen.TwipsPerPixelX \_

- gRightMargin \* 1440 - 1440 \* c

rectDrawTo.Bottom = inControl.Height \* Screen.TwipsPerPixelY \_

- gBottomMargin \* 1440 - 1440 \* D

mFormatRange.hdc = inControl.hdc

mFormatRange.hdcTarget = inControl.hdc

mFormatRange.rectRegion = rectDrawTo

mFormatRange.rectPage = rectPage

mFormatRange.mCharRange.firstChar = 0

mFormatRange.mCharRange.lastChar = -1

TextLength = Len(frmMain1.RTBox.Text)

mPageCtn = 1

Do

newStartPos = SendMessage(frmMain1.ActiveControl.hwnd, EM\_FORMATRANGE, True, mFormatRange)

If newStartPos >= TextLength Then

Exit Do

End If

mFormatRange.mCharRange.firstChar = newStartPos

mFormatRange.hdc = inControl.hdc

mFormatRange.hdcTarget = inControl.hdc

mPageCtn = mPageCtn + 1

DoEvents

Loop

Set inControl.Picture = LoadPicture

dumpaway = SendMessage(inControl.hwnd, EM\_FORMATRANGE, False, ByVal CLng(0))

PageCtnProc = mPageCtn

errhandler:

End Function

Sub DocPrintProc()

On Error Resume Next

DoEvents

Set preview.picP.Picture = LoadPicture

Dim mydialog1 As Object

Dim mFromPage As Integer, mToPage As Integer, mpage As Integer

Set mydialog1 = frmMain1.dlgCommonDialog

mydialog1.DialogTitle = "Print"

mydialog1.CancelError = True

mydialog1.FLAGS = cdlPDReturnDC + cdlPDPageNums

If frmMain1.RTBox.SelLength = 0 Then

mydialog1.FLAGS = mydialog1.FLAGS + cdlPDAllPages

Else

mydialog1.FLAGS = mydialog1.FLAGS + cdlPDSelection

End If

'mydialog1.ShowPrinter

If err = MSComDlg.cdlCancel Then

Exit Sub

End If

mFromPage = mydialog1.FromPage

mToPage = mydialog1.ToPage

If frmMain1.WindowState <> 1 Then

DocWYSIWYG frmMain1.ActiveControl

frmMain1.Move 0, 0

Else

MsgBox "Cannot proceed with minimized screen"

Exit Sub

End If

Printer.Print ""

Printer.ScaleMode = vbTwips

rectPage.Left = 0

rectPage.Top = 0

rectPage.Right = Printer.ScaleWidth

rectPage.Bottom = Printer.ScaleHeight

rectDrawTo.Left = gLeftMargin \* 1440

rectDrawTo.Top = gTopMargin \* 1440

rectDrawTo.Right = Printer.ScaleWidth - gRightMargin \* 1440

rectDrawTo.Bottom = Printer.ScaleHeight - gBottomMargin \* 1440

mFormatRange.hdc = preview.picP.hdc

mFormatRange.hdcTarget = preview.picP.hdc

newStartPos = 0

mFormatRange.rectRegion = rectDrawTo

mFormatRange.rectPage = rectPage

mFormatRange.mCharRange.firstChar = newStartPos

mFormatRange.mCharRange.lastChar = -1

TextLength = Len(frmMain1.ActiveControl.Text)

mpage = 1

Do

If mpage = mFromPage Then

Exit Do

End If

newStartPos = SendMessage(frmMain1.ActiveControl.hwnd, EM\_FORMATRANGE, True, mFormatRange)

If newStartPos >= TextLength Then

Exit Do

End If

mFormatRange.mCharRange.firstChar = newStartPos ' Starting position for next page

mFormatRange.hdc = preview.picP.hdc

mFormatRange.hdcTarget = preview.picP.hdc

mpage = mpage + 1

DoEvents

Loop

dumpaway = SendMessage(Screen.ActiveForm.ActiveControl.hwnd, EM\_FORMATRANGE, False, ByVal CLng(0))

If newStartPos >= TextLength Then

Exit Sub

End If

Printer.Print ""

Printer.ScaleMode = vbTwips

mFormatRange.hdc = Printer.hdc

mFormatRange.hdcTarget = Printer.hdc

mFormatRange.mCharRange.firstChar = newStartPos

Do

newStartPos = SendMessage(frmMain1.ActiveControl.hwnd, EM\_FORMATRANGE, True, mFormatRange)

If newStartPos >= TextLength Then

Exit Do

End If

If mpage = mToPage Then

Exit Do

End If

mFormatRange.mCharRange.firstChar = newStartPos

Printer.NewPage

Printer.Print ""

mFormatRange.hdc = Printer.hdc

mFormatRange.hdcTarget = Printer.hdc

mpage = mpage + 1

DoEvents

Loop

Printer.EndDoc

dumpaway = SendMessage(Screen.ActiveForm.ActiveControl.hwnd, EM\_FORMATRANGE, False, ByVal CLng(0))

End Sub