MailingLabels.aspx

```
1 <%@ Page Language="vb" Src="MailingLabels.aspx.vb" Inherits="Web_VB.GeneratorExamples.MailingLabels" %>
```

MailingLabels.aspx.vb

```
Imports System
Imports System.Configuration
Imports System.Data
Imports System.Data.SqlClient
Imports Microsoft.VisualBasic

Imports ceTe.DynamicPDF
Imports ceTe.DynamicPDF.PageElements

Namespace Web_VB.GeneratorExamples
    Public Class MailingLabels
        Inherits System.Web.UI.Page

        '////////////////////////////////////////////
        ' Begin Setting Region
        ' This settings are for Avery Label number 5160. Modify the settings below absed on your label number
        ' Set page dimensions
        Shared TopMargin As String = 36
        Shared BottomMargin As String = 36
        Shared RightMargin As String = 13.5
        Shared LeftMargin As String = 13.5

        'Set the number of labels per page
        Shared MaximumColumns As String = 3
        Shared MaximumRows As String = 10

        'Set the spacing between the labels
        Shared HorizontalSpace As String = 9
        Shared VerticalSpace As String = 0

        'These margins are on the labels themsleves
        Shared LabelTopBottomMargin As String = 5
        Shared LabelLeftRightMargin As String = 15
        ' End Setting Region
        '////////////////////////////////////////////

        'Dim the page level variables
        Dim MyDocument As ceTe.DynamicPDF.Document
        Dim MyPage As ceTe.DynamicPDF.Page

        Dim CurrentColumn, CurrentRow, LabelWidth, LabelHeight
        Dim CompanyName As String
```

MailingLabels.aspx.vb

```
        Dim ContactName As String
        Dim Phone As String
        Dim Fax As String
        Dim Connection As SqlConnection
        Dim Command As SqlCommand
        Dim DataReader As SqlDataReader
        Dim SQL As String

  Protected WithEvents txtStartDate As System.Web.UI.WebControls.TextBox
        Protected WithEvents txtEndDate As System.Web.UI.WebControls.TextBox
        Protected WithEvents btnGo As System.Web.UI.WebControls.Button
        Protected WithEvents btnSearchMembers As System.Web.UI.WebControls.Button

        Private Sub Page_Load(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles MyBase.Load
            'Put user code to initialize the page here
            'Create Document object and set properties

            MyDocument = New ceTe.DynamicPDF.Document
            MyPage = New ceTe.DynamicPDF.Page(PageSize.Letter, PageOrientation.Portrait)

            MyDocument.Creator = "MailingLabels.aspx"
            MyDocument.Author = "ceTe Software"
            MyDocument.Title = "Mailing Labels"

            'Entrypoint for the labels
            CurrentRow = 1
            CurrentColumn = 1

            Connection = New SqlConnection(ConfigurationSettings.AppSettings("northWindConnectionString"))
            Connection.Open()

            SQL = "Select CompanyName, ContactName, Phone, Fax FROM Customers"

            Command = New SqlCommand(SQL, Connection)

            DataReader = Command.ExecuteReader()

            'Loop over the RecordSet and add each label

            While DataReader.Read
                CompanyName = SafeDBNull(DataReader("CompanyName"))
                ContactName = SafeDBNull(DataReader("ContactName"))
```

```
                Phone = SafeDBNull(DataReader("Phone"))
                Fax = SafeDBNull(DataReader("Fax"))

                AddLabel()
            End While

            If MyPage.Elements.Count > 0 Then
                MyDocument.Pages.Add(MyPage)
            End If

            'Outputs the document to the current web page
            MyDocument.DrawToWeb("MailingLabels.pdf")
            Response.End()

        End Sub

        Function FindLabelHeight()

            FindLabelHeight = (MyPage.Dimensions.Height - (MyPage.Dimensions.TopMargin + MyPage.Dimensions.BottomMargin) -
((MaximumRows - 1) * VerticalSpace)) / MaximumRows

        End Function

        Function FindLabelWidth()

            FindLabelWidth = (MyPage.Dimensions.Width - (MyPage.Dimensions.RightMargin + MyPage.Dimensions.LeftMargin) -
((MaximumColumns - 1) * HorizontalSpace)) / MaximumColumns

        End Function

        Sub AddLabel()

            'Add a new page if you are beyond the maximum Rows
            If CurrentRow = MaximumRows + 1 Then

                MyDocument.Pages.Add(MyPage)

                CurrentRow = 1
            End If

            'Determines if the the label belongs in the first row or first column of the page
            If CurrentColumn > 1 And CurrentRow > 1 Then
```

```
125                 AddToPage()
126             ElseIf CurrentColumn > 1 And CurrentRow = 1 Then
127                 AddToFirstRow()
128             ElseIf CurrentColumn = 1 And CurrentRow > 1 Then
129                 AddToFirstColumn()
130             Else
131                 MyPage = New ceTe.DynamicPDF.Page(PageSize.Letter, PageOrientation.Portrait)
132
133                 MyPage.Dimensions.TopMargin = TopMargin
134                 MyPage.Dimensions.BottomMargin = BottomMargin
135                 MyPage.Dimensions.RightMargin = RightMargin
136                 MyPage.Dimensions.LeftMargin = LeftMargin
137
138                 'Calculates how big the Width and height of each label should be according to the margins and dimensions
    provided above
139                 LabelWidth = FindLabelWidth()
140                 LabelHeight = FindLabelHeight()
141
142                 AddToFirstRowColumn()
143             End If
144
145             'Incremment your row if you are beyond the maximum columns
146             If CurrentColumn = MaximumColumns + 1 Then
147                 CurrentRow = CurrentRow + 1
148                 CurrentColumn = 1
149             End If
150         End Sub
151
152         'Adds the label on at least row 2 column 2 of the page
153         Sub AddToPage()
154             Dim X, Y
155             Y = (CurrentRow - 1) * (LabelHeight + VerticalSpace)
156             X = (CurrentColumn - 1) * (LabelWidth + HorizontalSpace)
157             AddLabelInfo(X, Y)
158             CurrentColumn = CurrentColumn + 1
159         End Sub
160
161         'Adds the label on the first row of labels
162         Sub AddToFirstRow()
163             Dim X, Y
164             Y = 0
165             X = (CurrentColumn - 1) * (LabelWidth + HorizontalSpace)
```

```
166             AddLabelInfo(X, Y)
167             CurrentColumn = CurrentColumn + 1
168         End Sub
169 
170         'Adds the label to the First column of labels
171         Sub AddToFirstColumn()
172             Dim X, Y
173             Y = (CurrentRow - 1) * (LabelHeight + VerticalSpace)
174             X = 0
175             AddLabelInfo(X, Y)
176             CurrentColumn = CurrentColumn + 1
177         End Sub
178 
179         'Adds only the first label of every page (row 1 column 1)
180         Sub AddToFirstRowColumn()
181             Dim X, Y
182             Y = 0
183             X = 0
184             AddLabelInfo(X, Y)
185             CurrentColumn = CurrentColumn + 1
186         End Sub
187 
188         'This is where you format the look of each label
189         Sub AddLabelInfo(ByVal X, ByVal Y)
190 
191             Dim Txt1 As TextArea = New TextArea(CompanyName, X + LabelLeftRightMargin, Y + LabelTopBottomMargin, LabelWidth -
(LabelLeftRightMargin * 2), 11, Font.TimesRoman, 11)
192             Dim Txt2 As TextArea = New TextArea(ContactName, X + LabelLeftRightMargin, Y + LabelTopBottomMargin + 12, LabelWidth
- (LabelLeftRightMargin * 2), 11, Font.TimesRoman, 11)
193             Dim Txt3 As TextArea = New TextArea(Phone, X + LabelLeftRightMargin, Y + LabelTopBottomMargin + 24, LabelWidth -
(LabelLeftRightMargin * 2), 11, Font.TimesRoman, 11)
194             Dim Txt4 As TextArea = New TextArea(Fax, X + LabelLeftRightMargin, Y + LabelTopBottomMargin + 36, LabelWidth -
(LabelLeftRightMargin * 2), 11, Font.TimesRoman, 11)
195 
196             MyPage.Elements.Add(Txt1)
197             MyPage.Elements.Add(Txt2)
198             MyPage.Elements.Add(Txt3)
199             MyPage.Elements.Add(Txt4)
200 
201         End Sub
202 
203         Function SafeDBNull(ByVal Value As Object) As String
```

```
            If IsDBNull(Value) Then
                Return String.Empty
            Else
                Return CType(Value, String)
            End If
        End Function

    End Class
End Namespace
```