MailingLabels.aspx

```
1 <%@ Page language="c#" Src="MailingLabels.aspx.cs" Inherits="Web_CSharp.GeneratorExamples.Generator.MailingLabels" %>
```

MailingLabels.aspx.cs

```
using System;
using System.Data;
using System.Data.SqlClient;
using ceTe.DynamicPDF;
using ceTe.DynamicPDF.PageElements;

namespace Web_CSharp.GeneratorExamples.Generator
{
    public class MailingLabels : System.Web.UI.Page
    {
        ////////////////////////////////////////////////
        // Begin Setting Region
        // This settings are for Avery Label number 5160. Modify the settings below based on your label number

        // Set page dimensions
        static int topMargin = 36;
        static int bottomMargin = 36;
        static float rightMargin = 13.5f;
        static float leftMargin = 13.5f;

        // Set the number of labels per page
        static int maximumColumns = 3;
        static int maximumRows = 10;

        // Set the spacing between the labels
        static int horizontalSpace = 9;
        static int verticalSpace = 0;

        // These margins are on the labels themsleves
        static int labelTopBottomMargin = 5;
        static int labelLeftRightMargin = 15;

        // End Setting Region
        ////////////////////////////////////////////////

        ceTe.DynamicPDF.Document document;
        ceTe.DynamicPDF.Page page;
        float currentColumn, currentRow, labelWidth, labelHeight;
        string companyName;
        string contactName;
        string phone;
        string fax;
        SqlConnection connection;
        SqlCommand command;
        SqlDataReader dataReader;
        String SQL;

        private void Page_Load(object sender, System.EventArgs e)
        {
            // Put user code to initialize the page here
```

```
            document = new ceTe.DynamicPDF.Document();
            page = new ceTe.DynamicPDF.Page( PageSize.Letter, PageOrientation.Portrait );
            document.Creator = "MailingLabels.aspx";
            document.Author = "ceTe Software";
            document.Title = "Mailing Labels";

            // Entrypoint for the labels
            currentRow = 1;
            currentColumn = 1;
            connection = new SqlConnection( System.Configuration.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();
            }
            if ( page.Elements.Count > 0 )
                document.Pages.Add( page );

            // Outputs the document to the current web page
            document.DrawToWeb( "Labels.pdf" );
            Visible = false;
        }

        private float FindLabelHeight()
        {
            return ( page.Dimensions.Height - ( page.Dimensions.TopMargin + page.Dimensions.BottomMargin ) - (( maximumRows - 1
) * verticalSpace )) / maximumRows;
        }

        private float FindLabelWidth()
        {
            return ( page.Dimensions.Width - ( page.Dimensions.RightMargin + page.Dimensions.LeftMargin ) - (( maximumColumns -
1 ) * horizontalSpace )) / maximumColumns;
        }

        private void AddLabel()
        {
            // Add a new page if you are beyond the maximum Rows
            if( currentRow == maximumRows + 1 )
            {
```

```
98                  document.Pages.Add( page );
99                  currentRow = 1;
100             }
101
102             // Determines if the the label belongs in the first row or first column of the page
103             if( currentColumn > 1 & currentRow > 1 )
104             {
105                 AddToPage();
106             }
107             else if( currentColumn > 1 & currentRow == 1 )
108             {
109                 AddToFirstRow();
110             }
111             else if( currentColumn == 1 & currentRow > 1 )
112             {
113                 AddToFirstColumn();
114             }
115             else
116             {
117                 page = new ceTe.DynamicPDF.Page( PageSize.Letter, PageOrientation.Portrait );
118                 page.Dimensions.TopMargin = topMargin;
119                 page.Dimensions.BottomMargin = bottomMargin;
120                 page.Dimensions.RightMargin = rightMargin;
121                 page.Dimensions.LeftMargin = leftMargin;
122                 labelWidth = FindLabelWidth();
123                 labelHeight = FindLabelHeight();
124                 AddToFirstRowColumn();
125             }
126
127             // Incremment your row if you are beyond the maximum columns
128             if( currentColumn == maximumColumns + 1 )
129             {
130                 currentRow = currentRow + 1;
131                 currentColumn = 1;
132             }
133         }
134
135         // Adds the label on at least row 2 column 2 of the page
136         private void AddToPage()
137         {
138             float x;
139             float y;
140             y = ( currentRow - 1 ) * ( labelHeight + verticalSpace );
141             x = ( currentColumn - 1 ) * ( labelWidth + horizontalSpace );
142             AddLabelInfo( x, y );
143             currentColumn = currentColumn + 1;
144         }
145
146         // Adds the label on the first row of labels
147         private void AddToFirstRow()
```

```
148         {
149             float x;
150             float y;
151             y = 0;
152             x = ( currentColumn - 1 ) * ( labelWidth + horizontalSpace );
153             AddLabelInfo( x, y );
154             currentColumn = currentColumn + 1;
155         }
156 
157         // Adds the label to the First column of labels
158         private void AddToFirstColumn()
159         {
160             float x;
161             float y;
162             y = ( currentRow - 1 ) * ( labelHeight + verticalSpace );
163             x = 0;
164             AddLabelInfo( x, y );
165             currentColumn = currentColumn + 1;
166         }
167 
168         // Adds only the first label of every page (row 1 column 1)
169         private void AddToFirstRowColumn()
170         {
171             float x;
172             float y;
173             y = 0;
174             x = 0;
175             AddLabelInfo( x, y );
176             currentColumn = currentColumn + 1;
177         }
178 
179         // This is where you format the look of each label
180         private void AddLabelInfo( float x, float y )
181         {
182             TextArea txt1 = new TextArea( companyName, x + labelLeftRightMargin, y + labelTopBottomMargin, labelWidth - (
    labelLeftRightMargin * 2 ), 11, Font.TimesRoman, 11 );
183             TextArea txt2 = new TextArea( contactName, x + labelLeftRightMargin, y + labelTopBottomMargin + 12, labelWidth - (
    labelLeftRightMargin * 2 ), 11, Font.TimesRoman, 11 );
184             TextArea txt3 = new TextArea( phone, x + labelLeftRightMargin, y + labelTopBottomMargin + 24, labelWidth - (
    labelLeftRightMargin * 2 ), 11, Font.TimesRoman, 11 );
185             TextArea txt4 = new TextArea( fax, x + labelLeftRightMargin, y + labelTopBottomMargin + 36, labelWidth - (
    labelLeftRightMargin * 2 ), 11, Font.TimesRoman, 11 );
186             page.Elements.Add( txt1 );
187             page.Elements.Add( txt2 );
188             page.Elements.Add( txt3 );
189             page.Elements.Add( txt4 );
190         }
191 
192         private string SafeDBNull( object value )
193         {
```

```
            if( value is DBNull )
                return String.Empty;
            else
                return (string)value;
        }
    }
}
```