

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

```
1 using System;
2 using System.Data;
3 using System.Data.SqlClient;
4 using ceTe.DynamicPDF;
5 using ceTe.DynamicPDF.PageElements;
6
7 namespace Web_CSharp.GeneratorExamples.Generator
8 {
9     public class MailingLabels : System.Web.UI.Page
10    {
11        ///////////////////////////////////////////////////////////////////
12        // Begin Setting Region
13        // This settings are for Avery Label number 5160. Modify the settings below based on your label number
14
15        // Set page dimensions
16        static int topMargin = 36;
17        static int bottomMargin = 36;
18        static float rightMargin = 13.5f;
19        static float leftMargin = 13.5f;
20
21        // Set the number of labels per page
22        static int maximumColumns = 3;
23        static int maximumRows = 10;
24
25        // Set the spacing between the labels
26        static int horizontalSpace = 9;
27        static int verticalSpace = 0;
28
29        // These margins are on the labels themsleves
30        static int labelTopBottomMargin = 5;
31        static int labelLeftRightMargin = 15;
32
33        // End Setting Region
34        ///////////////////////////////////////////////////////////////////
35
36        ceTe.DynamicPDF.Document document;
37        ceTe.DynamicPDF.Page page;
38        float currentColumn, currentRow, labelWidth, labelHeight;
39        string companyName;
40        string contactName;
41        string phone;
42        string fax;
43        SqlConnection connection;
44        SqlCommand command;
45        SqlDataReader dataReader;
46        String SQL;
47
48        private void Page_Load(object sender, System.EventArgs e)
49        {
50            // Put user code to initialize the page here
```

```
51     document = new ceTe.DynamicPDF.Document();
52     page = new ceTe.DynamicPDF.Page( PageSize.Letter, PageOrientation.Portrait );
53     document.Creator = "MailingLabels.aspx";
54     document.Author = "ceTe Software";
55     document.Title = "Mailing Labels";
56
57     // Entrypoint for the labels
58     currentRow = 1;
59     currentColumn = 1;
60     connection = new SqlConnection( System.Configuration.ConfigurationSettings.AppSettings["NorthWindConnectionString"]
);
61
62     connection.Open();
63     SQL = "Select CompanyName, ContactName, Phone, Fax FROM Customers";
64     command = new SqlCommand( SQL, connection );
65     dataReader = command.ExecuteReader();
66
67     // Loop over the RecordSet and add each label
68     while( dataReader.Read() )
69     {
70         companyName = SafeDBNull( dataReader["CompanyName"] );
71         contactName = SafeDBNull( dataReader["ContactName"] );
72         phone = SafeDBNull( dataReader["Phone"] );
73         fax = SafeDBNull( dataReader["Fax"] );
74         AddLabel();
75     }
76     if ( page.Elements.Count > 0 )
77         document.Pages.Add( page );
78
79     // Outputs the document to the current web page
80     document.DrawToWeb( "Labels.pdf" );
81     Visible = false;
82 }
83
84 private float FindLabelHeight()
85 {
86     return ( page.Dimensions.Height - ( page.Dimensions.TopMargin + page.Dimensions.BottomMargin ) - (( maximumRows - 1
) * verticalSpace )) / maximumRows;
87 }
88
89 private float FindLabelWidth()
90 {
91     return ( page.Dimensions.Width - ( page.Dimensions.RightMargin + page.Dimensions.LeftMargin ) - (( maximumColumns -
1 ) * horizontalSpace )) / maximumColumns;
92 }
93
94 private void AddLabel()
95 {
96     // Add a new page if you are beyond the maximum Rows
97     if( currentRow == maximumRows + 1 )
98     {
```

```
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 // Increment 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     {
```

```
194         if( value is DBNull )
195             return String.Empty;
196         else
197             return (string)value;
198     }
199 }
200 }
```