

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

```
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
8 {
9     public class ContactList : System.Web.UI.Page
10    {
11        #region Member Variables
12
13        // Top, bottom, left and right margins of report
14        private static float margin = 36;
15        // Height of the header
16        private static float headerHeight = 74;
17        // Height of the footer
18        private static float footerHeight = 14;
19        // Size of paper to use
20        private static PageDimensions pageSize = new PageDimensions( PageSize.Letter, PageOrientation.Landscape, margin );
21        // Bottom Y coordinate for the body of the report
22        private static float bodyBottom = pageSize.Height - ( margin * 2 ) - footerHeight;
23
24        // Current page that elements are being added to
25        private ceTe.DynamicPDF.Page currentPage = null;
26
27        // Current Y coordinate where elements are being added
28        private float currentY = 0;
29        // Used to control the alternating background
30        private bool alternateBG = false;
31        // Used to test for grouping
32        private string currentFirstI = string.Empty;
33        // Template for header and footer elements
34
35        private static Template template = null;
36        // Connection string to the Northwind database
37        static string dbConnString =
38            System.Configuration.ConfigurationSettings.AppSettings["NorthWindConnectionString"];
39
40        #endregion
41
42        #region Page Load
43
44        private void Page_Load(object sender, System.EventArgs e)
45        {
46            if( template == null )
47                SetTemplate();
48
49            // Create a document and set it's properties
50            Document document = new Document();
```

```
51     document.Creator = "ContactList.aspx";
52     document.Author = "ceTe Software";
53     document.Title = "Contact List";
54     document.Template = template;
55
56     // Establishes connection to the database
57     SqlConnection connection = GetOpenDBConn();
58     SqlDataReader data = GetContactListData( connection );
59
60     // Builds the report
61     BuildDocument( document, data );
62
63     // Cleans up database connections
64     data.Close();
65     connection.Close();
66
67     // Outputs the ContactList to the current web page
68     document.DrawToWeb( "ContactList.pdf" );
69 }
70
71 #endregion
72
73 #region Private Methods
74
75 private void SetTemplate()
76 {
77     template = new Template();
78
79     // Uncomment the line below to add a layout grid to the each page
80     //template.Elements.Add( new LayoutGrid() );
81
82     // Adds header elements to the template
83     template.Elements.Add( new Image( MapPath( "../Images/DPDFLogo.png" ), 0, 0, 0.21f ) );
84     template.Elements.Add( new Label( "Northwind Traders", 0, 0, 720, 18, Font.HelveticaBold, 18, TextAlign.Center ) );
85     template.Elements.Add( new Label( "Contact List", 0, 21, 720, 12, Font.Helvetica, 12, TextAlign.Center ) );
86     template.Elements.Add( new Label( DateTime.Now.ToString("dd MMM yyyy, H:mm:ss EST"), 0, 36, 720, 12, Font.Helvetica,
87 12, TextAlign.Center ) );
88     template.Elements.Add( new Rectangle( 0, 56, 720, 16, Grayscale.Black, new WebColor( "0000A0" ), 0.0F ) );
89     template.Elements.Add( new Label( "Cust ID", 2, 57, 58, 12, Font.HelveticaBold, 12, TextAlign.Left, Grayscale.White
90 ) );
91     template.Elements.Add( new Label( "Company", 62, 57, 156, 12, Font.HelveticaBold, 12, TextAlign.Left,
92 Grayscale.White ) );
93     template.Elements.Add( new Label( "Contact", 222, 57, 156, 12, Font.HelveticaBold, 12, TextAlign.Left,
94 Grayscale.White ) );
95     template.Elements.Add( new Label( "Title", 362, 57, 156, 12, Font.HelveticaBold, 12, TextAlign.Left, Grayscale.White
96 ) );
97     template.Elements.Add( new Label( "Phone", 522, 57, 86, 12, Font.HelveticaBold, 12, TextAlign.Left, Grayscale.White
98 ) );
99     template.Elements.Add( new Label( "Fax", 622, 57, 86, 12, Font.HelveticaBold, 12, TextAlign.Left, Grayscale.White )
100 );
```

```
94
95     // Adds footer elements to the template
96     PageNumberingLabel pageNumLabel = new PageNumberingLabel( "Page %%CP(i)%% of %%TP(i)%%", 0, bodyBottom + 5, 720, 10,
Font.Helvetica, 10, TextAlign.Center );
97     template.Elements.Add( pageNumLabel );
98 }
99
100 private void BuildDocument( Document document, SqlDataReader data )
101 {
102     // Builds the PDF document with data from the DataReader
103     AddNewPage( document );
104     while( data.Read() )
105     {
106         // Add current record to the document
107         AddRecord( document, data );
108     }
109 }
110
111 private void AddRecord( Document document, SqlDataReader data )
112 {
113     // Creates TextAreas that are expandable
114     TextArea companyName = new TextArea( data.GetString(2), 62, currentY + 3, 156, 11, Font.TimesRoman, 11 );
115     TextArea contactName = new TextArea( data.GetString(3), 222, currentY + 3, 136, 11, Font.TimesRoman, 11 );
116     TextArea contactTitle = new TextArea( data.GetString(4), 362, currentY + 3, 156, 11, Font.TimesRoman, 11 );
117
118     // Gets the height required for the current record
119     float requiredHeight = SetExpandableRecords( document, data, companyName, contactName, contactTitle );
120
121     // Creates non expandable Labels
122     Label customerID = new Label( data.GetString(0), 2, currentY + 3, 58, 11, Font.TimesRoman, 11 );
123     Label phone = new Label( data.GetString(5), 522, currentY + 3, 96, 11, Font.TimesRoman, 11 );
124     Label fax = new Label( data.IsDBNull(6)?"":data.GetString(6), 622, currentY + 3, 96, 11, Font.TimesRoman, 11 );
125
126     // Adds alternating background if required
127     if( alternateBG )
128     {
129         currentPage.Elements.Add( new Rectangle( 0, currentY, 720, requiredHeight + 6, RgbColor.Black, new WebColor(
"EOEOFF" ), 0.0F ) );
130     }
131
132     // Toggles alternating background
133     alternateBG = !alternateBG;
134
135     // Adds elements to the current page
136     currentPage.Elements.Add( customerID );
137     currentPage.Elements.Add( companyName );
138     currentPage.Elements.Add( contactName );
139     currentPage.Elements.Add( contactTitle );
140     currentPage.Elements.Add( phone );
141     currentPage.Elements.Add( fax );
```

```
142
143     // increments the current Y position on the page
144     currentY += requiredHeight + 6;
145 }
146
147 private float SetExpandableRecords( Document document, SqlDataReader data, TextArea companyName, TextArea contactName,
TextArea contactTitle )
148 {
149     // Gets the maximum height required of the three TextAreas
150     float requiredHeight = GetMaxRecordHeight( companyName, contactName, contactTitle );
151
152     // Add space for the section header if required
153     float sectionHeaderHeight = 0;
154     if( currentFirstI != data.GetString(1) ) sectionHeaderHeight = 26;
155
156     // Add a new page if needed
157     if( bodyBottom < currentY + requiredHeight + sectionHeaderHeight + 4 )
158     {
159         AddNewPage( document );
160         if( sectionHeaderHeight == 0 )
161         {
162             // Update Y coordinate of TextArea when placed on the new page
163             companyName.Y = currentY + 3;
164             contactName.Y = currentY + 3;
165             contactTitle.Y = currentY + 3;
166         }
167     }
168
169     // Add section header if required
170     if( sectionHeaderHeight > 0 )
171     {
172         AddSectionHeader( data );
173         companyName.Y = currentY + 3;
174         contactName.Y = currentY + 3;
175         contactTitle.Y = currentY + 3;
176     }
177     return requiredHeight;
178 }
179
180 private float GetMaxRecordHeight( TextArea companyName, TextArea contactName, TextArea contactTitle )
181 {
182     // Returns the maximum required height of the three TextAreas
183     float requiredHeight = 11;
184     float requiredHeightB = 0;
185
186     requiredHeight = companyName.GetRequiredHeight();
187     requiredHeightB = contactName.GetRequiredHeight();
188     if( requiredHeightB > requiredHeight ) requiredHeight = requiredHeightB;
189     requiredHeightB = contactTitle.GetRequiredHeight();
190     if( requiredHeightB > requiredHeight ) requiredHeight = requiredHeightB;
```

```
191         if( requiredHeight > 11 )
192         {
193             companyName.Height = requiredHeight;
194             contactName.Height = requiredHeight;
195             contactTitle.Height = requiredHeight;
196         }
197         return requiredHeight;
198     }
199
200     private void AddSectionHeader( SqlDataReader data )
201     {
202         // Adds a section header to the current Y coordinate of the current page
203         currentFirstI = data.GetString(1);
204         currentPage.Elements.Add( new Label( "-" + currentFirstI + "-", 0, currentY + 6, 720, 18, Font.HelveticaBold, 18,
TextAlign.Center ) );
205         currentY += 26;
206         alternateBG = false;
207     }
208
209     private void AddNewPage( Document document )
210     {
211         // Adds a new page to the document
212         currentPage = new Page( pageSize );
213         currentY = headerHeight;
214         alternateBG = false;
215         document.Pages.Add( currentPage );
216     }
217
218     private SqlConnection GetOpenDBConn()
219     {
220         // Creates and opens a database connection
221         SqlConnection connection = new SqlConnection( dBConnString );
222         connection.Open();
223         return connection;
224     }
225
226     private SqlDataReader GetContactListData( SqlConnection connection )
227     {
228         // Creates a DataReader for the report
229         SqlCommand command = connection.CreateCommand();
230         command.CommandText = "SELECT CustomerID, FirstL = Left(CompanyName,1), " +
231             "CompanyName, ContactName, ContactTitle, Phone, Fax " +
232             "FROM Customers ORDER BY CompanyName";
233         SqlDataReader dataReader = command.ExecuteReader();
234
235         return dataReader;
236     }
237
238     #endregion
239 }
```

240 }