

Invoice.aspx

```
1 <%@ Page language="c#" Inherits="Web_CSharp.MergerExamples.Invoice" CodeFile="Invoice.aspx.cs" %>
2 <!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.0 Transitional//EN" >
3
4 <html>
5   <head>
6     <title>DynamicPDF Merger v10.0 for .NET - Invoice Example</title>
7     <link rel="stylesheet" type="text/css" href="../DynamicPDF.css">
8   </head>
9   <body>
10
11   <div id="banner">
12     
13     <div id="header"><i>Dynamic</i><b>PDF</b> Merger v10.0 for .NET Examples</div>
14     <h1>Invoice Example</h1>
15   </div>
16
17   <div id="content">
18     <form id="FormFill" method="get" target="_blank" runat="server">
19       <h1>Select the invoices to include in the PDF.</h1>
20       <asp:CheckBoxList id="chkInvoices" runat="server">
21         <asp:ListItem Value="10248">10248</asp:ListItem>
22         <asp:ListItem Value="10249">10249</asp:ListItem>
23         <asp:ListItem Value="10250">10250</asp:ListItem>
24         <asp:ListItem Value="10251">10251</asp:ListItem>
25         <asp:ListItem Value="10252">10252</asp:ListItem>
26         <asp:ListItem Selected=True Value="10360">10360</asp:ListItem>
27         <asp:ListItem Selected=True Value="10979">10979</asp:ListItem>
28         <asp:ListItem Value="11077">11077</asp:ListItem>
29       </asp:CheckBoxList>
30       <asp:Button id="btnCreateInvoices" Text="Create Invoices" runat="server" />
31     </form>
32   </div>
33
34 </body>
35 </html>
```

Invoice.aspx.cs

```
1  using System;
2  using System.Data;
3  using System.Data.SqlClient;
4  using ceTe.DynamicPDF;
5  using ceTe.DynamicPDF.Merger;
6  using ceTe.DynamicPDF.PageElements;
7  using ceTe.DynamicPDF.PageElements.BarCoding;
8
9 namespace Web_CSharp.MergerExamples
{
10    public partial class Invoice : System.Web.UI.Page
11    {
12
13        // Connection string to the Northwind database
14        static string dBConnString =
15            System.Configuration.ConfigurationSettings.AppSettings["NorthWindConnectionString"];
16
17        protected void Page_Load(object sender, System.EventArgs e)
18        {
19            if( IsPostBack )
20            {
21                CreateInvoices();
22            }
23        }
24
25
26        private void CreateInvoices()
27        {
28            // Create a document and set it's properties
29            Document document = new Document();
30            document.Creator = "Invoice.aspx";
31            document.Author = "ceTe Software";
32            document.Title = "Invoice";
33
34            // Establishes connection to the database
35            SqlConnection connection = GetOpenDBConn();
36
37            // Add Invoices to the document
38            foreach( System.Web.UI.WebControls.ListItem item in chkInvoices.Items )
39            {
40                if( item.Selected )
41                {
42                    MyInvoice invoice = new MyInvoice( this.Server );
43                    invoice.Draw( connection, document, int.Parse( item.Value ) );
44                }
45            }
46
47            // Cleans up database connections
48            connection.Close();
49
50            // Outputs the Invoices to the current web page
51        }
52    }
53}
```

```
51    try
52    {
53        document.DrawToWeb( "Invoice.pdf" );
54    }
55    catch( EmptyDocumentException )
56    {
57        // Show an empty document error
58        ShowErrorDocument( "Document is empty. No invoices were selected.", "" );
59    }
60    catch( Exception general )
61    {
62        // Show a general error
63        ShowErrorDocument( general.Message, general.StackTrace );
64    }
65    finally
66    {
67        Visible = false;
68    }
69 }
70
71 private SqlConnection GetOpenDBConn()
72 {
73     // Creates and opens a database connection
74     SqlConnection connection = new SqlConnection( dBConnString );
75     connection.Open();
76     return connection;
77 }
78
79 private void ShowErrorDocument( string message, string stackTrace )
80 {
81     // Shows the error in a PDF document
82     Response.Clear();
83     Document document = new Document();
84
85     ceTe.DynamicPDF.Page page = new ceTe.DynamicPDF.Page( PageSize.Letter, PageOrientation.Portrait );
86     TextArea messageArea = new TextArea( message, 0, 0, 512, 36, Font.Helvetica, 18, RgbColor.Red );
87     messageArea.Height = messageArea.GetRequiredHeight();
88
89     page.Elements.Add( messageArea );
90     if( stackTrace.Trim().Length > 0 )
91     {
92         float stackTraceTop = messageArea.Y + messageArea.Height + 20;
93         TextArea stackTraceArea = new TextArea( stackTrace, 0, stackTraceTop, 512, 692 - stackTraceTop, Font.Helvetica,
94 10 );
95         page.Elements.Add( new ceTe.DynamicPDF.PageElements.Label( "StackTrace:", 0, stackTraceTop - 12, 512, 12,
Font.HelveticaBold, 10 ) );
96         page.Elements.Add( stackTraceArea );
97     }
98     document.Pages.Add( page );
99     document.DrawToWeb();
```

Invoice.aspx.cs

```
99         Response.End();
100    }
101
102    #region MyInvoice Class
103
104    private class MyInvoice
105    {
106
107        #region Member Variables
108
109        private static string templatePath;
110        private decimal subTotal = 0;
111        private decimal freight = 0;
112        private static bool templatePathSet = false;
113
114    #endregion
115
116    #region Constructors
117
118    public MyInvoice( System.Web.HttpServerUtility server )
119    {
120        // Sets the template files path
121        // This is done in the constructor because MapPath is not available within the subclass
122        SetTemplatePath( server );
123    }
124
125    #endregion
126
127    #region Private Methods
128
129    private void SetTemplatePath( System.Web.HttpServerUtility server )
130    {
131        // Adds the image to page template if it is not already added
132        if(!templatePathSet )
133        {
134            templatePath = server.MapPath( "../PDFs/Invoice.pdf" );
135            templatePathSet = true;
136        }
137    }
138
139    public void Draw( SqlConnection connection, Document document, int invoiceNumber )
140    {
141        // Each Invoice should begin a new section
142        document.Sections.Begin();
143
144        // Gets the Invoice data
145        SqlDataReader data = GetInvoiceData( connection, invoiceNumber );
146
147        // Adds the invoice if there is data
148        if( data.Read() )
```

```
149         {
150             // Sets the Freight amount
151             freight = (decimal)data[ "Freight" ];
152
153             // Draws the invoice data, returns a page object if it is the last page
154             Page lastPage = DrawInvoiceData( data, document );
155
156             // Draws additional pages if necessary
157             while( lastPage == null )
158             {
159                 lastPage = DrawInvoiceData( data, document );
160             }
161
162             // Draws the totals to the bottom of the last page of the Invoice
163             DrawTotals( data, lastPage );
164         }
165
166         // Close DataReader
167         data.Close();
168     }
169
170     private Page DrawInvoiceData( SqlDataReader data, Document document )
171     {
172         // Tracks if the invoice is finished
173         bool invoiceFinished = true;
174
175         // Tracks the y position on the page
176         float yOffset = 288;
177
178         // Create a page using "Invoice.pdf" as a template
179         Page page = new ImportedPage( templatePath, 1 );
180
181         // Uncomment the line below to show a layout grid.
182         //page.Elements.Add( new LayoutGrid() );
183
184         // Add Details to the Invoice
185         DrawInvoiceDetails( data, page );
186
187         // Add bill to address
188         DrawBillTo( data, page );
189
190         // Add ship to address
191         DrawShipTo( data, page );
192
193         // Add line items to the Invoice
194         while( DrawLineItem( data, page, ref yOffset ) )
195         {
196             // Break if at the bottom of the page
197             if( yOffset >= 666 )
198             {
```

```
199             invoiceFinished = false;
200         }
201     }
202
203     // Add the page to the document
204     document.Pages.Add( page );
205
206     // If Invoice is finished return the page else return null so another page will be added
207     if( invoiceFinished )
208     {
209         return page;
210     }
211     else
212     {
213         page.Elements.Add( new Label( "Continued...", 454, 704, 82, 12, Font.Helvetica, 12, TextAlign.Right ) );
214         return null;
215     }
216 }
217
218
219 private void DrawInvoiceDetails( SqlDataReader data, Page page )
220 {
221     // Adds Invoice details to the page
222     page.Elements.Add( new Label( data[ "OrderID" ].ToString(), 437, 25, 100, 12, Font.Helvetica, 12 ) );
223     page.Elements.Add( new Label( ((DateTime)data[ "OrderDate" ]).ToString( "d MMM yyyy" ), 437, 39, 100, 12,
Font.Helvetica, 12 ) );
224     page.Elements.Add( new Label( data[ "CustomerID" ].ToString(), 437, 53, 100, 12, Font.Helvetica, 12 ) );
225     if( !( data[ "ShippedDate" ] is DBNull ) )
226     {
227         page.Elements.Add( new Label( ((DateTime)data[ "ShippedDate" ]).ToString( "d MMM yyyy" ), 437, 67, 100, 12,
Font.Helvetica, 12 ) );
228     }
229     page.Elements.Add( new Label( data[ "ShipperName" ].ToString(), 437, 81, 100, 12, Font.Helvetica, 12 ) );
230     page.Elements.Add( new Interleaved25( data[ "OrderID" ].ToString(), 380, 4, 18, false ) );
231     page.Elements.Add( new PageNumberingLabel( "Page %%SP%% of %%ST%%", 450, 253, 90, 20, Font.HelveticaBold, 12,
TextAlign.Center ) );
232 }
233
234
235 private void DrawBillTo( SqlDataReader data, Page page )
236 {
237     // Adds bill to address
238     string strBillToAddress =
239         data[ "BillName" ].ToString() + "\n" +
240         data[ "BillAddress" ].ToString() + "\n" +
241         data[ "BillCity" ].ToString() + ", " +
242         data[ "BillPostalCode" ].ToString() + "\n" +
243         data[ "BillCountry" ].ToString();
244     page.Elements.Add( new TextArea( strBillToAddress, 28, 138, 194, 70, Font.Helvetica, 12 ) );
245 }
```

```
246     private void DrawShipTo( SqlDataReader data, Page page )
247     {
248         // Adds ship to address
249         string strShipToAddress =
250             data[ "ShipName" ].ToString() + "\n" +
251             data[ "ShipAddress" ].ToString() + "\n" +
252             data[ "ShipCity" ].ToString() + ", " +
253             data[ "ShipPostalCode" ].ToString() + "\n" +
254             data[ "ShipCountry" ].ToString();
255         page.Elements.Add( new TextArea( strShipToAddress, 318, 138, 194, 70, Font.Helvetica, 12 ) );
256     }
257
258     private bool DrawLineItem( SqlDataReader data, Page page, ref float yOffset )
259     {
260         // Adds a line item to the invoice
261         Int16 quantity = (Int16)data[ "Quantity" ];
262         decimal unitPrice = (decimal)data[ "UnitPrice" ];
263         decimal lineTotal = unitPrice * quantity;
264
265         subTotal += lineTotal;
266         page.Elements.Add( new Label( quantity.ToString(), 4, 2 + yOffset, 52, 12, Font.Helvetica, 12, TextAlign.Right ) );
267
268         page.Elements.Add( new Label( data[ "ProductName" ].ToString(), 64, 2 + yOffset, 292, 12, Font.Helvetica, 12, TextAlign.Right ) );
269         page.Elements.Add( new Label( unitPrice.ToString( "#,##0.00" ), 364, 2 + yOffset, 82, 12, Font.Helvetica, 12, TextAlign.Right ) );
270         page.Elements.Add( new Label( lineTotal.ToString( "#,##0.00" ), 454, 2 + yOffset, 82, 12, Font.Helvetica, 12, TextAlign.Right ) );
271         yOffset += 18;
272
273         // Advances the recordset and returns if there are additional records
274         return data.Read();
275     }
276
277     private void DrawTotals( SqlDataReader data, Page page )
278     {
279         // Add totals to the bottom of the Invoice
280         decimal grandTotal = subTotal + freight;
281         page.Elements.Add( new Label( subTotal.ToString( "#,##0.00" ), 454, 668, 82, 12, Font.Helvetica, 12, TextAlign.Right ) );
282         page.Elements.Add( new Label( freight.ToString( "#,##0.00" ), 454, 686, 82, 12, Font.Helvetica, 12, TextAlign.Right ) );
283         page.Elements.Add( new Label( grandTotal.ToString( "#,##0.00" ), 454, 704, 82, 12, Font.Helvetica, 12, TextAlign.Right ) );
284     }
285
286     private SqlDataReader GetInvoiceData( SqlConnection connection, int invoiceNumber )
287     {
288         // Creates a DataReader for the Invoice
289         SqlCommand command = connection.CreateCommand();
```

```
289     command.CommandText =
290         "SELECT o.OrderID, o.CustomerID, o.OrderDate, o.ShippedDate, Freight, " +
291         "      o.ShipName, o.ShipAddress, o.ShipCity, o.ShipPostalCode, o.ShipCountry, " +
292         "      c.CompanyName as BillName, c.Address as BillAddress, " +
293         "      c.City as BillCity, c.PostalCode as BillPostalCode, " +
294         "      c.Country as BillCountry, s.CompanyName as ShipperName, " +
295         "      p.ProductName, od.UnitPrice, od.Quantity " +
296         "FROM Orders o " +
297         "JOIN Customers c ON " +
298         "      o.CustomerID = c.CustomerID " +
299         "JOIN Shippers s ON " +
300         "      o.ShipVia = s.ShipperID " +
301         "JOIN [Order Details] od ON " +
302         "      o.OrderID = od.OrderID " +
303         "JOIN Products p ON " +
304         "      od.ProductID = p.ProductID " +
305         "WHERE      o.OrderID = " + invoiceNumber;
306     SqlDataReader dataReader = command.ExecuteReader();
307     return dataReader;
308 }
309
310 #endregion
311 }
312
313 #endregion
314
315 #region Web Form Designer generated code
316 override protected void OnInit(EventArgs e)
317 {
318     //
319     // CODEGEN: This call is required by the ASP.NET Web Form Designer.
320     //
321     InitializeComponent();
322     base.OnInit(e);
323 }
324
325 /// <summary>
326 /// Required method for Designer support - do not modify
327 /// the contents of this method with the code editor.
328 /// </summary>
329 private void InitializeComponent()
330 {
331 }
332 #endregion
333 }
334 }
```