Invoice.aspx

```
<%@ Page language="c#" Inherits="Web_CSharp.MergerExamples.Invoice" CodeFile="Invoice.aspx.cs" %>
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.0 Transitional//EN" >

<html>
  <head>
    <title>DynamicPDF Merger v10.0 for .NET - Invoice Example</title>
    <link rel="stylesheet" type="text/css" href="../DynamicPDF.css">
  </head>
  <body>

    <div id="banner">
      <img src="../Images/DynamicPDF_top.gif" align=right><img src="../Images/ceTeSoftware_top.gif">
      <div id="header"><i>Dynamic</i><b>PDF</b> Merger v10.0 for .NET Examples</div>
      <h1>Invoice Example</h1>
    </div>

      <div id="content">
      <form id="FormFill" method="get" target="_blank" runat="server">
        <h1>Select the invoices to include in the PDF.</h1>
        <asp:CheckBoxList id="chkInvoices" runat="server">
          <asp:ListItem Value="10248">10248</asp:ListItem>
          <asp:ListItem Value="10249">10249</asp:ListItem>
          <asp:ListItem Value="10250">10250</asp:ListItem>
          <asp:ListItem Value="10251">10251</asp:ListItem>
          <asp:ListItem Value="10252">10252</asp:ListItem>
          <asp:ListItem Selected=True Value="10360">10360</asp:ListItem>
          <asp:ListItem Selected=True Value="10979">10979</asp:ListItem>
          <asp:ListItem Value="11077">11077</asp:ListItem>
        </asp:CheckBoxList>
        <asp:Button id="btnCreateInvoices" Text="Create Invoices" runat="server" />
      </form>
    </div>

  </body>
</html>
```

Invoice.aspx.cs

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

namespace Web_CSharp.MergerExamples
{
    public partial class Invoice : System.Web.UI.Page
    {

        // Connection string to the Northwind database
        static string dBConnString =
            System.Configuration.ConfigurationSettings.AppSettings["NorthWindConnectionString"];

        protected void Page_Load(object sender, System.EventArgs e)
        {
            if( IsPostBack )
            {
                CreateInvoices();
            }
        }

        private void CreateInvoices()
        {
            // Create a document and set it's properties
            Document document = new Document();
            document.Creator = "Invoice.aspx";
            document.Author = "ceTe Software";
            document.Title = "Invoice";

            // Establises connection to the database
            SqlConnection connection = GetOpenDBConn();

            // Add Invoices to the document
            foreach( System.Web.UI.WebControls.ListItem item in chkInvoices.Items )
            {
                if( item.Selected )
                {
                    MyInvoice invoice = new MyInvoice( this.Server );
                    invoice.Draw( connection, document, int.Parse( item.Value ) );
                }
            }

            // Cleans up database connections
            connection.Close();

            // Outputs the Invoices to the current web page
```

Invoice.aspx.cs

```
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,
    10 );
94                  page.Elements.Add( new ceTe.DynamicPDF.PageElements.Label( "StackTrace:", 0, stackTraceTop - 12, 512, 12,
    Font.HelveticaBold, 10 ) );
95                  page.Elements.Add( stackTraceArea );
96              }
97              document.Pages.Add( page );
98              document.DrawToWeb();
```

```
            Response.End();
        }

        #region MyInvoice Class

        private class MyInvoice
        {

            #region Member Variables

            private static string templatePath;
            private decimal subTotal = 0;
            private decimal freight = 0;
            private static bool templatePathSet = false;

            #endregion

            #region Constructors

            public MyInvoice( System.Web.HttpServerUtility server )
            {
                // Sets the template files path
                // This is done in the constructor because MapPath is not available within the subclass
                SetTemplatePath( server );
            }

            #endregion

            #region Private Methods

            private void SetTemplatePath( System.Web.HttpServerUtility server )
            {
                // Adds the image to page template if it is not already added
                if(!templatePathSet )
                {
                    templatePath = server.MapPath( "../PDFs/Invoice.pdf" );
                    templatePathSet = true;
                }
            }

            public void Draw( SqlConnection connection, Document document, int invoiceNumber )
            {
                // Each Invoice should begin a new section
                document.Sections.Begin();

                // Gets the Invoice data
                SqlDataReader data = GetInvoiceData( connection, invoiceNumber );

                // Adds the invoice if there is data
                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 aditional 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                     {
```

```
                        invoiceFinished = false;
                        break;
                    }
                }

                // Add the page to the document
                document.Pages.Add( page );

                // If Invoice is finished return the page else return null so another page will be added
                if( invoiceFinished )
                {
                    return page;
                }
                else
                {
                    page.Elements.Add( new Label( "Continued...", 454, 704, 82, 12, Font.Helvetica, 12, TextAlign.Right ) );
                    return null;
                }
            }

            private void DrawInvoiceDetails( SqlDataReader data, Page page )
            {
                // Adds Invoice details to the page
                page.Elements.Add( new Label( data[ "OrderID" ].ToString(), 437, 25, 100, 12, Font.Helvetica, 12 ) );
                page.Elements.Add( new Label( ((DateTime)data[ "OrderDate" ]).ToString( "d MMM yyyy" ), 437, 39, 100, 12,
Font.Helvetica, 12 ) );
                page.Elements.Add( new Label( data[ "CustomerID" ].ToString(), 437, 53, 100, 12, Font.Helvetica, 12 ) );
                if( !( data[ "ShippedDate" ] is DBNull ) )
                {
                    page.Elements.Add( new Label( ((DateTime)data[ "ShippedDate" ]).ToString( "d MMM yyyy" ), 437, 67, 100, 12,
Font.Helvetica, 12 ) );
                }
                page.Elements.Add( new Label( data[ "ShipperName" ].ToString(), 437, 81, 100, 12, Font.Helvetica, 12 ) );
                page.Elements.Add( new Interleaved25( data[ "OrderID" ].ToString(), 380, 4, 18, false ) );
                page.Elements.Add( new PageNumberingLabel( "Page %%SP%% of %%ST%% ", 450, 253, 90, 20, Font.HelveticaBold, 12,
TextAlign.Center ) );
            }

            private void DrawBillTo( SqlDataReader data, Page page )
            {
                // Adds bill to address
                string strBillToAddress =
                    data[ "BillName" ].ToString() + "\n" +
                    data[ "BillAddress" ].ToString() + "\n" +
                    data[ "BillCity" ].ToString() + ", " +
                    data[ "BillPostalCode" ].ToString() + "\n" +
                    data[ "BillCountry" ].ToString();
                page.Elements.Add( new TextArea( strBillToAddress, 28, 138, 194, 70, Font.Helvetica, 12 ) );
            }

```

```
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                 page.Elements.Add( new Label( data[ "ProductName" ].ToString(), 64, 2 + yOffset, 292, 12, Font.Helvetica, 12 )
    );
268                 page.Elements.Add( new Label( unitPrice.ToString( "#,##0.00" ), 364, 2 + yOffset, 82, 12, Font.Helvetica, 12,
    TextAlign.Right ) );
269                 page.Elements.Add( new Label( lineTotal.ToString( "#,##0.00" ), 454, 2 + yOffset, 82, 12, Font.Helvetica, 12,
    TextAlign.Right ) );
270                 yOffset += 18;
271 
272                 // Advanses the recordset and returns if there are additional records
273                 return data.Read();
274             }
275 
276             private void DrawTotals( SqlDataReader data, Page page )
277             {
278                 // Add totals to the bottom of the Invoice
279                 decimal grandTotal = subTotal + freight;
280                 page.Elements.Add( new Label( subTotal.ToString( "#,##0.00" ), 454, 668, 82, 12, Font.Helvetica, 12,
    TextAlign.Right ) );
281                 page.Elements.Add( new Label( freight.ToString( "#,##0.00" ), 454, 686, 82, 12, Font.Helvetica, 12,
    TextAlign.Right ) );
282                 page.Elements.Add( new Label( grandTotal.ToString( "#,##0.00" ), 454, 704, 82, 12, Font.Helvetica, 12,
    TextAlign.Right ) );
283             }
284 
285             private SqlDataReader GetInvoiceData( SqlConnection connection, int invoiceNumber )
286             {
287                 // Creates a DataReader for the Invioce
288                 SqlCommand command = connection.CreateCommand();
```

```
                command.CommandText =
                    "SELECT o.OrderID, o.CustomerID, o.OrderDate, o.ShippedDate, Freight, " +
                    "        o.ShipName, o.ShipAddress, o.ShipCity, o.ShipPostalCode, o.ShipCountry, " +
                    "        c.CompanyName as BillName, c.Address as BillAddress, " +
                    "        c.City as BillCity, c.PostalCode as BillPostalCode, " +
                    "        c.Country as BillCountry, s.CompanyName as ShipperName, " +
                    "        p.ProductName, od.UnitPrice, od.Quantity " +
                    "FROM    Orders o " +
                    "JOIN    Customers c ON " +
                    "        o.CustomerID = c.CustomerID " +
                    "JOIN    Shippers s ON " +
                    "        o.ShipVia = s.ShipperID " +
                    "JOIN    [Order Details] od ON " +
                    "        o.OrderID = od.OrderID " +
                    "JOIN    Products p ON " +
                    "        od.ProductID = p.ProductID " +
                    "WHERE    o.OrderID = " + invoiceNumber;
                SqlDataReader dataReader = command.ExecuteReader();
                return dataReader;
            }

            #endregion
        }

        #endregion

        #region Web Form Designer generated code
        override protected void OnInit(EventArgs e)
        {
            //
            // CODEGEN: This call is required by the ASP.NET Web Form Designer.
            //
            InitializeComponent();
            base.OnInit(e);
        }

        /// <summary>
        /// Required method for Designer support - do not modify
        /// the contents of this method with the code editor.
        /// </summary>
        private void InitializeComponent()
        {
        }
        #endregion
    }
}
```