

```
1 <%@ Page Language="C#" AutoEventWireup="true" CodeFile="InvoiceExample.aspx.cs" Inherits="ReportWriter_InvoiceExample" %>
```

```
1 using System;
2 using System.Collections.Generic;
3 using System.Web;
4 using System.Web.UI;
5 using System.Web.UI.WebControls;
6 using System.Data;
7 using System.Data.SqlClient;
8
9 using ceTe.DynamicPDF;
10 using ceTe.DynamicPDF.LayoutEngine;
11 using ceTe.DynamicPDF.LayoutEngine.Data;
12
13 public partial class ReportWriter_InvoiceExample : System.Web.UI.Page
14 {
15     protected void Page_Load(object sender, EventArgs e)
16     {
17         // Create the document's layout from a DLEX template
18         DocumentLayout documentLayout = new DocumentLayout(Server.MapPath("Invoice.dlex"));
19
20         InvoiceExampleData.Order order = InvoiceExampleData.Order11077;
21         NameValueLayoutData layoutData = new NameValueLayoutData();
22         layoutData.Add("OrderID", order.OrderID);
23         layoutData.Add("OrderDate", order.OrderDate);
24         layoutData.Add("CustomerID", order.CustomerID);
25         layoutData.Add("ShippedDate", order.ShippedDate);
26         layoutData.Add("ShipperName", order.ShipperName);
27         layoutData.Add("BillTo", order.BillTo);
28         layoutData.Add("ShipTo", order.ShipTo);
29         layoutData.Add("Freight", order.Freight);
30         layoutData.Add("OrderDetails", order.OrderDetails);
31
32         // Layout the document using the parameters and set it's properties
33         Document document = documentLayout.Layout(layoutData);
34         document.Author = "DynamicPDF ReportWriter";
35         document.Title = "Invoice Example";
36
37         // Outputs the document to a file
38         document.DrawToWeb("ReportWriter_Invoice.pdf");
39     }
40
41     // This method fires when the DataNeeded event occurs (before the report is laid out)
42     private static DataTableReportData RetrieveOrderReportData(string xmlFile)
43     {
44         // Data is loaded into a DataTable from XML for portability.
45         DataTable invoiceData = new DataTable("Order");
46         invoiceData.ReadXml(xmlFile);
47         return new DataTableReportData(invoiceData);
48     }
49
50     // Use this method if you have access to the Northwind sample database.
```

```
51     private static DataReaderReportData RetrieveSqlOrderReportData()
52     {
53         string connectionString = "Data Source=(local);Initial Catalog=Northwind;Integrated Security=true";
54         string queryString =
55             "SELECT o.OrderID, o.CustomerID, o.OrderDate, o.ShippedDate, Freight, o.ShipName, o.ShipAddress, o.ShipCity,
56             o.ShipPostalCode, " +
57             "        o.ShipCountry, c.CompanyName as BillName, c.Address as BillAddress, c.City as BillCity, c.PostalCode as
58             BillPostalCode, " +
59             "        c.Country as BillCountry, s.CompanyName as ShipperName, p.ProductName, od.UnitPrice, od.Quantity " +
60             "FROM Orders o " +
61             "JOIN Customers c ON o.CustomerID = c.CustomerID      JOIN Shippers s ON o.ShipVia = s.ShipperID " +
62             "JOIN [Order Details] od ON o.OrderID = od.OrderID      JOIN Products p ON od.ProductID = p.ProductID " +
63             "WHERE o.OrderID = @OrderID";
64
65         SqlConnection connection = new SqlConnection(connectionString);
66
67         SqlCommand command = new SqlCommand(queryString, connection);
68         command.Parameters.AddWithValue("@OrderID", 11077);
69
70         connection.Open();
71         SqlDataReader reader = command.ExecuteReader();
72
73         return new DataReaderReportData(connection, reader);
74     }
75 }
76
77 public class InvoiceExampleData
78 {
79     public class Order
80     {
81         public int OrderID { get; set; }
82         public DateTime OrderDate { get; set; }
83         public string CustomerID { get; set; }
84         public DateTime ShippedDate { get; set; }
85         public string ShipperName { get; set; }
86         public string ShipTo { get; set; }
87         public string BillTo { get; set; }
88         public decimal Freight { get; set; }
89         public List<OrderDetail> OrderDetails { get; set; }
90     }
91
92     public class OrderDetail
93     {
94         public int ProductID { get; set; }
95         public int Quantity { get; set; }
96         public string ProductName { get; set; }
97         public decimal UnitPrice { get; set; }
98     }
99
100     private static Order order11077 = new Order
101     {
```

```
99      OrderID = 11077,
100      OrderDate = new DateTime(2019, 1, 6),
101      CustomerID = "RATTC",
102      ShippedDate = new DateTime(2019, 1, 30),
103      ShipperName = "United Package",
104      ShipTo = "Rattlesnake Canyon Grocery\n2817 Milton Dr.\nAlbuquerque, NM 87110\nUSA",
105      BillTo = "Rattlesnake Canyon Grocery\n2817 Milton Dr.\nAlbuquerque, NM 87110\nUSA",
106      Freight = 8.53M,
107      OrderDetails = new List<OrderDetail>() {
108          new OrderDetail { ProductID = 2, Quantity = 24, ProductName = "Chang", UnitPrice = 19M },
109          new OrderDetail { ProductID = 3, Quantity = 4, ProductName = "Aniseed Syrup", UnitPrice = 10M },
110          new OrderDetail { ProductID = 4, Quantity = 1, ProductName = "Chef Anton's Cajun Seasoning", UnitPrice = 22M },
111          new OrderDetail { ProductID = 6, Quantity = 1, ProductName = "Grandma's Boysenberry Spread", UnitPrice = 25M },
112          new OrderDetail { ProductID = 7, Quantity = 1, ProductName = "Uncle Bob's Organic Dried Pears", UnitPrice = 30M
113      },
114          new OrderDetail { ProductID = 8, Quantity = 2, ProductName = "Northwoods Cranberry Sauce", UnitPrice = 40M },
115          new OrderDetail { ProductID = 10, Quantity = 1, ProductName = "Ikura", UnitPrice = 31M },
116          new OrderDetail { ProductID = 12, Quantity = 2, ProductName = "Queso Manchego La Pastora", UnitPrice = 38M },
117          new OrderDetail { ProductID = 13, Quantity = 4, ProductName = "Konbu", UnitPrice = 6M },
118          new OrderDetail { ProductID = 14, Quantity = 1, ProductName = "Tofu", UnitPrice = 23.25M },
119          new OrderDetail { ProductID = 16, Quantity = 2, ProductName = "Pavlova", UnitPrice = 17.45M },
120          new OrderDetail { ProductID = 20, Quantity = 1, ProductName = "Sir Rodney's Marmalade", UnitPrice = 81M },
121          new OrderDetail { ProductID = 23, Quantity = 2, ProductName = "Tunnbröd", UnitPrice = 9M },
122          new OrderDetail { ProductID = 32, Quantity = 1, ProductName = "Mascarpone Fabioli", UnitPrice = 32M },
123          new OrderDetail { ProductID = 39, Quantity = 2, ProductName = "Chartreuse verte", UnitPrice = 18M },
124          new OrderDetail { ProductID = 41, Quantity = 3, ProductName = "Jack's New England Clam Chowder", UnitPrice =
9.65M },
125          new OrderDetail { ProductID = 46, Quantity = 3, ProductName = "Spegesild", UnitPrice = 12M },
126          new OrderDetail { ProductID = 52, Quantity = 2, ProductName = "Filo Mix", UnitPrice = 7M },
127          new OrderDetail { ProductID = 55, Quantity = 2, ProductName = "Pâté chinois", UnitPrice = 24M },
128          new OrderDetail { ProductID = 60, Quantity = 2, ProductName = "Camembert Pierrot", UnitPrice = 34M },
129          new OrderDetail { ProductID = 64, Quantity = 2, ProductName = "Wimmers gute Semmelknödel", UnitPrice = 33.25M },
130          new OrderDetail { ProductID = 66, Quantity = 1, ProductName = "Louisiana Hot Spiced Okra", UnitPrice = 17M },
131          new OrderDetail { ProductID = 73, Quantity = 2, ProductName = "Röd Kaviar", UnitPrice = 15M },
132          new OrderDetail { ProductID = 75, Quantity = 4, ProductName = "Rhönbräu Klosterbier", UnitPrice = 7.75M },
133          new OrderDetail { ProductID = 77, Quantity = 2, ProductName = "Original Frankfurter grüne Soße", UnitPrice = 13M
134      };
135
136      public static Order Order11077
137      {
138          get { return order11077; }
139      }
140  }
```