InvoiceExample.aspx

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

InvoiceExample.aspx.cs

```
using System;
using System.Collections.Generic;
using System.Web;
using System.Web.UI;
using System.Web.UI.WebControls;
using System.Data;
using System.Data.SqlClient;

using ceTe.DynamicPDF;
using ceTe.DynamicPDF.LayoutEngine;
using ceTe.DynamicPDF.LayoutEngine.Data;

public partial class ReportWriter_InvoiceExample : System.Web.UI.Page
{
    protected void Page_Load(object sender, EventArgs e)
    {
        // Create the document's layout from a DLEX template
        DocumentLayout documentLayout = new DocumentLayout(Server.MapPath("Invoice.dlex"));

        InvoiceExampleData.Order order = InvoiceExampleData.Order11077;
        NameValueLayoutData layoutData = new NameValueLayoutData();
        layoutData.Add("OrderID", order.OrderID);
        layoutData.Add("OrderDate", order.OrderDate);
        layoutData.Add("CustomerID", order.CustomerID);
        layoutData.Add("ShippedDate", order.ShippedDate);
        layoutData.Add("ShipperName", order.ShipperName);
        layoutData.Add("BillTo", order.BillTo);
        layoutData.Add("ShipTo", order.ShipTo);
        layoutData.Add("Freight", order.Freight);
        layoutData.Add("OrderDetails", order.OrderDetails);

        // Layout the document using the parameters and set it's properties
        Document document = documentLayout.Layout(layoutData);
        document.Author = "DynamicPDF ReportWriter";
        document.Title = "Invoice Example";

        // Outputs the document to a file
        document.DrawToWeb("ReportWriter_Invoice.pdf");
    }

    // This method fires when the DataNeeded event occurs (before the report is laid out)
    private static DataTableReportData RetrieveOrderReportData(string xmlFile)
    {
        // Data is loaded into a DataTable from XML for portability.
        DataTable invoiceData = new DataTable("Order");
        invoiceData.ReadXml(xmlFile);
        return new DataTableReportData(invoiceData);
    }

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

```
        OrderID = 11077,
        OrderDate = new DateTime(2019, 1, 6),
        CustomerID = "RATTC",
        ShippedDate = new DateTime(2019, 1, 30),
        ShipperName = "United Package",
        ShipTo = "Rattlesnake Canyon Grocery\n2817 Milton Dr.\nAlbuquerque, NM 87110\nUSA",
        BillTo = "Rattlesnake Canyon Grocery\n2817 Milton Dr.\nAlbuquerque, NM 87110\nUSA",
        Freight = 8.53M,
        OrderDetails = new List<OrderDetail>() {
                new OrderDetail { ProductID = 2, Quantity = 24, ProductName = "Chang", UnitPrice = 19M },
                new OrderDetail { ProductID = 3, Quantity = 4, ProductName = "Aniseed Syrup", UnitPrice = 10M },
                new OrderDetail { ProductID = 4, Quantity = 1, ProductName = "Chef Anton's Cajun Seasoning", UnitPrice = 22M },
                new OrderDetail { ProductID = 6, Quantity = 1, ProductName = "Grandma's Boysenberry Spread", UnitPrice = 25M },
                new OrderDetail { ProductID = 7, Quantity = 1, ProductName = "Uncle Bob's Organic Dried Pears", UnitPrice = 30M
},
                new OrderDetail { ProductID = 8, Quantity = 2, ProductName = "Northwoods Cranberry Sauce", UnitPrice = 40M },
                new OrderDetail { ProductID = 10, Quantity = 1, ProductName = "Ikura", UnitPrice = 31M },
                new OrderDetail { ProductID = 12, Quantity = 2, ProductName = "Queso Manchego La Pastora", UnitPrice = 38M },
                new OrderDetail { ProductID = 13, Quantity = 4, ProductName = "Konbu", UnitPrice = 6M },
                new OrderDetail { ProductID = 14, Quantity = 1, ProductName = "Tofu", UnitPrice = 23.25M },
                new OrderDetail { ProductID = 16, Quantity = 2, ProductName = "Pavlova", UnitPrice = 17.45M },
                new OrderDetail { ProductID = 20, Quantity = 1, ProductName = "Sir Rodney's Marmalade", UnitPrice = 81M },
                new OrderDetail { ProductID = 23, Quantity = 2, ProductName = "Tunnbröd", UnitPrice = 9M },
                new OrderDetail { ProductID = 32, Quantity = 1, ProductName = "Mascarpone Fabioli", UnitPrice = 32M },
                new OrderDetail { ProductID = 39, Quantity = 2, ProductName = "Chartreuse verte", UnitPrice = 18M },
                new OrderDetail { ProductID = 41, Quantity = 3, ProductName = "Jack's New England Clam Chowder", UnitPrice =
9.65M },
                new OrderDetail { ProductID = 46, Quantity = 3, ProductName = "Spegesild", UnitPrice = 12M },
                new OrderDetail { ProductID = 52, Quantity = 2, ProductName = "Filo Mix", UnitPrice = 7M },
                new OrderDetail { ProductID = 55, Quantity = 2, ProductName = "Pâté chinois", UnitPrice = 24M },
                new OrderDetail { ProductID = 60, Quantity = 2, ProductName = "Camembert Pierrot", UnitPrice = 34M },
                new OrderDetail { ProductID = 64, Quantity = 2, ProductName = "Wimmers gute Semmelknödel", UnitPrice = 33.25M },
                new OrderDetail { ProductID = 66, Quantity = 1, ProductName = "Louisiana Hot Spiced Okra", UnitPrice = 17M },
                new OrderDetail { ProductID = 73, Quantity = 2, ProductName = "Röd Kaviar", UnitPrice = 15M },
                new OrderDetail { ProductID = 75, Quantity = 4, ProductName = "Rhönbräu Klosterbier", UnitPrice = 7.75M },
                new OrderDetail { ProductID = 77, Quantity = 2, ProductName = "Original Frankfurter grüne Soße", UnitPrice = 13M
}
            }
    };

    public static Order Order11077
    {
        get { return order11077; }
    }
}
```