SimpleXMLReport.aspx

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

SimpleXMLReport.aspx.cs

```
using System;
using System.IO;
using System.Xml;
using ceTe.DynamicPDF;
using ceTe.DynamicPDF.PageElements;

namespace Web_CSharp.GeneratorExamples
{
    public class SimpleXMLReport : System.Web.UI.Page
    {
        // Template for document elements
        private Template template = new Template();

        // Page Dimensions of pages
        private static PageDimensions pageDimensions = new PageDimensions( PageSize.Letter, PageOrientation.Portrait, 54.0f );
        // Current page that elements are being added to
        private ceTe.DynamicPDF.Page currentPage = null;
        // Top Y coordinate for the body of the report
        private float bodyTop = 38;
        // Bottom Y coordinate for the body of the report
        private float bodyBottom = pageDimensions.Body.Bottom - pageDimensions.Body.Top;
        // Current Y coordinate where elements are being added
        private float currentY = 0;

        // Used to control the alternating background
        private bool alternateBG = false;

        private void Page_Load(object sender, System.EventArgs e)
        {
            // Create a document and set it's properties
            Document document = new Document();
            document.Creator = "SimpleXMLReport.aspx";
            document.Author = "ceTe Software";
            document.Title = "Simple XML Report";
            XmlTextReader reader = new XmlTextReader( MapPath( "../Data/ProductList.xml" ) );

            // Adds elements to the header template
            SetTemplate();

            // Add the template to the document
            document.Template = template;

            // Builds the report
            BuildDocument( document, reader );

            // Outputs the SimpleReport to the current web page
            document.DrawToWeb( "SimpleXMLReport.pdf" );
        }

        private void SetTemplate()
```

SimpleXMLReport.aspx.cs

```
51          {
52              // Adds elements to the header template
53              template.Elements.Add( new Label( DateTime.Now.ToString("dd MMM yyyy, H:mm:ss EST"), 0, 0, 504, 12,
    Font.HelveticaBold, 12 ) );
54              template.Elements.Add( new Label( "Northwind Product List", 0, 0, 504, 12, Font.HelveticaBold, 12, TextAlign.Center
    ) );
55              PageNumberingLabel pageNumLabel = new PageNumberingLabel( "Page %%CP%% of %%TP%%", 0, 0, 504, 12,
    Font.HelveticaBold, 12, TextAlign.Right );
56              template.Elements.Add( pageNumLabel );
57              template.Elements.Add( new Label( "Product", 2, 23, 236, 11, Font.TimesBold, 11 ) );
58              template.Elements.Add( new Label( "Qty Per Unit", 242, 23, 156, 11, Font.TimesBold, 11 ) );
59              template.Elements.Add( new Label( "Unit Price", 402, 23, 100, 11, Font.TimesBold, 11, TextAlign.Right ) );
60              template.Elements.Add( new Line( 0, 36, 504, 36 ) );
61
62              // Uncomment the line below to add a layout grid to each page
63              //template.Elements.Add( new LayoutGrid() );
64          }
65
66          private void BuildDocument( Document document, XmlTextReader reader )
67          {
68              // Builds the PDF document with data from the XML Data
69              AddNewPage( document );
70              while( reader.Read())
71              {
72                  if(reader.NodeType == XmlNodeType.Element && reader.Name == "Products")
73                  {
74                      //Add current node to the document
75                      AddRecord( document, reader );
76                  }
77              }
78          }
79
80          private void AddRecord( Document document, XmlTextReader reader )
81          {
82              // Adds a new page to the document if needed
83              if( currentY > bodyBottom ) AddNewPage( document );
84
85              // Adds alternating background to document if needed
86              if( alternateBG )
87              {
88                  currentPage.Elements.Add( new Rectangle( 0, currentY, 504, 18, RgbColor.Black, new WebColor( "E0E0FF" ), 0.0F )
    );
89              }
90
91              // Adds Labels to the document with data from the current node
92              currentPage.Elements.Add( new Label( reader.GetAttribute( "ProductName" ), 2, currentY + 3, 236, 11,
    Font.TimesRoman, 11 ) );
93              currentPage.Elements.Add( new Label( reader.GetAttribute( "QuantityPerUnit" ), 242, currentY + 3, 156, 11,
    Font.TimesRoman, 11 ) );
```

```
            currentPage.Elements.Add( new Label( reader.GetAttribute( "UnitPrice" ), 402, currentY + 3, 100, 11,


            // Toggles alternating background
            alternateBG = !alternateBG;

            // Increments the current Y position on the page
            currentY += 18;
        }

        private void AddNewPage( Document document )
        {
            // Adds a new page to the document
            currentPage = new Page( pageDimensions );
            currentY = bodyTop;
            alternateBG = false;
            document.Pages.Add( currentPage );
        }
    }
}
```