

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

```
1 using System;
2 using System.IO;
3 using System.Xml;
4 using ceTe.DynamicPDF;
5 using ceTe.DynamicPDF.PageElements;
6
7 namespace Web_CSharp.GeneratorExamples
8 {
9     public class SimpleXMLReport : System.Web.UI.Page
10    {
11        // Template for document elements
12        private Template template = new Template();
13
14        // Page Dimensions of pages
15        private static PageDimensions pageDimensions = new PageDimensions( PageSize.Letter, PageOrientation.Portrait, 54.0f );
16        // Current page that elements are being added to
17        private ceTe.DynamicPDF.Page currentPage = null;
18        // Top Y coordinate for the body of the report
19        private float bodyTop = 38;
20        // Bottom Y coordinate for the body of the report
21        private float bodyBottom = pageDimensions.Body.Bottom - pageDimensions.Body.Top;
22        // Current Y coordinate where elements are being added
23        private float currentY = 0;
24
25        // Used to control the alternating background
26        private bool alternateBG = false;
27
28        private void Page_Load(object sender, System.EventArgs e)
29        {
30            // Create a document and set it's properties
31            Document document = new Document();
32            document.Creator = "SimpleXMLReport.aspx";
33            document.Author = "ceTe Software";
34            document.Title = "Simple XML Report";
35            XmlTextReader reader = new XmlTextReader( MapPath( "../Data/ProductList.xml" ) );
36
37            // Adds elements to the header template
38            SetTemplate();
39
40            // Add the template to the document
41            document.Template = template;
42
43            // Builds the report
44            BuildDocument( document, reader );
45
46            // Outputs the SimpleReport to the current web page
47            document.DrawToWeb( "SimpleXMLReport.pdf" );
48        }
49
50        private void SetTemplate()
```

```
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 ) );
```

```
94         currentPage.Elements.Add( new Label( reader.GetAttribute( "UnitPrice" ), 402, currentY + 3, 100, 11,
95
96         // Toggles alternating background
97         alternateBG = !alternateBG;
98
99         // Increments the current Y position on the page
100        currentY += 18;
101    }
102
103    private void AddNewPage( Document document )
104    {
105        // Adds a new page to the document
106        currentPage = new Page( pageDimensions );
107        currentY = bodyTop;
108        alternateBG = false;
109        document.Pages.Add( currentPage );
110    }
111 }
112 }
```