

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

```
1 using System;
2 using System.Data;
3 using System.Data.SqlClient;
4 using ceTe.DynamicPDF;
5 using ceTe.DynamicPDF.PageElements;
6
7 namespace Web_CSharp.GeneratorExamples.Generator
8 {
9     public class SimpleDataGridReport : 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        // Connection string to the Northwind database
29        static string dbConnString =
30            System.Configuration.ConfigurationSettings.AppSettings["NorthWindConnectionString"];
31        // Initializing the DataGrid
32        System.Web.UI.WebControls.DataGrid DataGrid1 = new System.Web.UI.WebControls.DataGrid();
33
34        private void Page_Load(object sender, System.EventArgs e)
35        {
36            // Create a document and set it's properties
37            Document document = new Document();
38            document.Creator = "SimpleDataGridReport.aspx";
39            document.Author = "ceTe Software";
40            document.Title = "Simple Data Grid Report";
41
42            // Adds elements to the header template
43            SetTemplate();
44            document.Template = template;
45
46            // Establishes connection to the database
47            SqlConnection connection = GetOpenDBConn();
48            SqlDataAdapter sqlDataAdapter = new SqlDataAdapter("SELECT ProductName, QuantityPerUnit, UnitPrice FROM Products",
connection);
49            System.Data.DataSet dataSet = new System.Data.DataSet();
```

```
50
51     // Populate the DataSet with the results from the query
52     sqlDataAdapter.Fill(dataSet,"Products");
53
54     // Binding DataGrid with DataSet
55     DataGrid1.DataSource = dataSet;
56     AddNewPage( document );
57
58     // Iterating through each row of the DataGrid.
59     for (int i=1; i<=dataSet.Tables["Products"].Rows.Count; i++)
60     {
61         AddRecord( document, dataSet.Tables["Products"].Rows[DataGrid1.SelectedIndex+i]);
62     }
63
64     // Cleans up database connections
65     connection.Close();
66
67     // Outputs the SimpleReport to the current web page
68     document.DrawToWeb( "SimpleDataGridReport.pdf" );
69 }
70
71 private void SetTemplate()
72 {
73     // Adds elements to the header template
74     template.Elements.Add( new Label( DateTime.Now.ToString("dd MMM yyyy, H:mm:ss EST"), 0, 0, 504, 12,
Font.HelveticaBold, 12 ) );
75     template.Elements.Add( new Label( "Northwind Product List", 0, 0, 504, 12, Font.HelveticaBold, 12, TextAlign.Center
) );
76     PageNumberingLabel pageNumLabel = new PageNumberingLabel( "Page %%CP%% of %%TP%%", 0, 0, 504, 12,
Font.HelveticaBold, 12, TextAlign.Right );
77     template.Elements.Add( pageNumLabel );
78     template.Elements.Add( new Label( "Product", 2, 23, 236, 11, Font.TimesBold, 11 ) );
79     template.Elements.Add( new Label( "Qty Per Unit", 242, 23, 156, 11, Font.TimesBold, 11 ) );
80     template.Elements.Add( new Label( "Unit Price", 402, 23, 100, 11, Font.TimesBold, 11, TextAlign.Right ) );
81     template.Elements.Add( new Line( 0, 36, 504, 36 ) );
82
83     // Uncomment the line below to add a layout grid to each page
84     //template.Elements.Add( new LayoutGrid() );
85 }
86
87 private void AddRecord( Document document, DataRow row )
88 {
89     // Adds a new page to the document if needed
90     if( currentY > bodyBottom ) AddNewPage( document );
91
92     // Adds alternating background to document if needed
93     if( alternateBG )
94     {
95         currentPage.Elements.Add( new Rectangle( 0, currentY, 504, 18, RgbColor.Black, new WebColor( "E0E0FF" ), 0.0F )
);
```

```
96     }
97
98     // Adds Labels to the document with data from current record
99     currentPage.Elements.Add( new Label( row["ProductName"].ToString(), 2, currentY + 3, 236, 11, Font.TimesRoman, 11 )
100 );
101     currentPage.Elements.Add( new Label( row["QuantityPerUnit"].ToString(), 242, currentY + 3, 156, 11, Font.TimesRoman,
11 ) );
102     currentPage.Elements.Add( new Label( row["UnitPrice"].ToString(), 402, currentY + 3, 100, 11, Font.TimesRoman, 11,
103 TextAlign.Right ) );
104
105     // Toggles alternating background
106     alternateBG = !alternateBG;
107
108     // Increments the current Y position on the page
109     currentY += 18;
110 }
111
112 private void AddNewPage( Document document )
113 {
114     // Adds a new page to the document
115     currentPage = new Page( pageDimensions );
116     currentY = bodyTop;
117     alternateBG = false;
118     document.Pages.Add( currentPage );
119 }
120
121 private SqlConnection GetOpenDBConn()
122 {
123     // Creates and opens a database connection
124     SqlConnection connection = new SqlConnection( dBConnString );
125     connection.Open();
126     return connection;
127 }
```