

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

```
1 using System;
2 using System.Data;
3 using System.Data.SqlClient;
4 using ceTe.DynamicPDF;
5 using ceTe.DynamicPDF.IO;
6 using ceTe.DynamicPDF.Text;
7 using ceTe.DynamicPDF.PageElements;
8
9 namespace Web_CSharp.GeneratorExamples
10 {
11     public class TableReport : System.Web.UI.Page
12     {
13         static string dBConnString =
14             System.Configuration.ConfigurationSettings.AppSettings["NorthWindConnectionString"];
15
16         private void Page_Load(object sender, System.EventArgs e)
17         {
18             // Create a document and set it's properties
19             ceTe.DynamicPDF.Document document = new ceTe.DynamicPDF.Document();
20             document.Creator = "TableReport.aspx";
21             document.Author = "ceTe Software";
22             document.Title = "Table Report";
23
24             // Establishes connection to the database
25             SqlConnection connection = GetOpenDBConn();
26             SqlDataReader data = GetContactListData( connection );
27
28             //Create a Table and set it's properties
29             Table2 table = new Table2( 0, 0, 512, 676, ceTe.DynamicPDF.Font.Helvetica, 12 );
30             table.Border.Width = 1f;
31             table.CellDefault.Border.Width = 1f;
32             table.Border.Color = Grayscale.Black;
33             table.RepeatColumnHeaderCount = 1;
34             table.RepeatRowHeaderCount = 1;
35
36             // Builds the report
37             BuildTable( data, table );
38             int currentRow = 0;
39             int currentColumn = 0;
40             do
41             {
42                 currentRow++;
43                 currentColumn = 1;
44                 AddTableToPage( document, table, currentRow, currentColumn );
45                 Table2 overflowTable = table.GetOverflowColumns();
46                 do
47                 {
48                     currentColumn++;
49                     AddTableToPage( document, overflowTable, currentRow, currentColumn );
50                     overflowTable = overflowTable.GetOverflowColumns();
```

```
51         } while ( overflowTable != null );
52         table = table.GetOverflowRows();
53     } while ( table != null );
54
55     // Cleans up database connections
56     data.Close();
57     connection.Close();
58
59     // Outputs the SimpleReport to the current web page
60     document.DrawToWeb( "TableExample.pdf" );
61 }
62
63 private void AddTableToPage( Document document, Table2 table, int currentRow, int currentColumn )
64 {
65     Page page = new Page( PageSize.Letter );
66     if( table != null )
67         page.Elements.Add( table );
68     page.Elements.Add( new Label( "(" + currentRow + "," + currentColumn + ")", 0, page.Dimensions.Body.Height - 12,
page.Dimensions.Body.Width, 12, Font.Helvetica, 12, TextAlign.Center ) );
69     document.Pages.Add( page );
70 }
71
72 private void BuildTable( SqlDataReader data, Table2 table )
73 {
74     CreateColumns( table );
75     CreateRowHeadings( table );
76     while( data.Read() )
77         CreateRow( table, data );
78 }
79
80 private void CreateRowHeadings( Table2 table )
81 {
82     Row2 row = table.Rows.Add( 40, Font.TimesBold, 12, RgbColor.Black, RgbColor.LightGrey );
83     row.CellDefault.Align = TextAlign.Center;
84     row.CellDefault.VAlign = VAlign.Top;
85     row.Cells.Add( "ID" );
86     row.Cells.Add( "Product Name" );
87     row.Cells.Add( "Supplier ID" );
88     row.Cells.Add( "Category ID" );
89     row.Cells.Add( "Quantity Per Unit" );
90     row.Cells.Add( "Unit Price" );
91     row.Cells.Add( "Unit In Stock" );
92     row.Cells.Add( "Units On Order" );
93     row.Cells.Add( "Reorder Level" );
94     row.Cells.Add( "Discontinued" );
95 }
96
97 private void CreateRow( Table2 table, SqlDataReader data )
98 {
99     Row2 row = table.Rows.Add( 20 );
```

```
100         row.Cells.Add( data.GetInt32(0).ToString( ), Font.Helvetica, 12, RgbColor.Black, RgbColor.LightGrey, 1 );
101         row.Cells.Add( data.GetString(1) );
102         row.Cells.Add( data.GetInt32(2).ToString( ) );
103         row.Cells.Add( data.GetInt32(3).ToString( ) );
104         row.Cells.Add( data.GetString(4) );
105         row.Cells.Add( data.GetDecimal(5).ToString( "$0.00" ) );
106         row.Cells.Add( data.GetInt16(6).ToString( ) );
107         row.Cells.Add( data.GetInt16(7).ToString( ) );
108         row.Cells.Add( data.GetInt16(8).ToString( ) );
109         row.Cells.Add( data.GetBoolean(9).ToString( ) );
110     }
111
112     private void CreateColumns( Table2 table )
113     {
114         table.Columns.Add( 25 );
115         table.Columns.Add( 150 );
116         table.Columns.Add( 90 );
117         table.Columns.Add( 90 );
118         table.Columns.Add( 120 );
119         table.Columns.Add( 60 );
120         table.Columns.Add( 90 );
121         table.Columns.Add( 90 );
122         table.Columns.Add( 90 );
123         table.Columns.Add( 90 );
124     }
125
126     private SqlConnection GetOpenDBConn()
127     {
128         // Creates and opens a database connection
129         SqlConnection connection = new SqlConnection( dBConnString );
130         connection.Open();
131         return connection;
132     }
133
134     private SqlDataReader GetContactListData( SqlConnection connection )
135     {
136         // Creates a DataReader for the report
137         SqlCommand command = connection.CreateCommand();
138         command.CommandText = "SELECT * FROM Products";
139         SqlDataReader dataReader = command.ExecuteReader();
140         return dataReader;
141     }
142 }
143 }
```