

## TableReport.aspx

---

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

## TableReport.aspx.cs

---

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

## TableReport.aspx.cs

---

```
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 );
```

## TableReport.aspx.cs

---

```
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 }
```