

```
1 <%@ Page language="c#" Src="SourceView.aspx.cs" Inherits="Web_CSharp.GeneratorExamples.SourceView" Title="View Source of the  
selected example - DynamicPDF" %>
```

```
1 using System;
2 using System.IO;
3 using ceTe.DynamicPDF;
4 using ceTe.DynamicPDF.PageElements;
5
6 namespace Web_CSharp.GeneratorExamples
7 {
8     public class SourceView : System.Web.UI.Page
9     {
10         #region Member Variables
11
12         //////////////////////////////////////
13         // Begin Setting Region
14         // These settings are for editable. Modify the settings below based on your requirements
15
16         private static Font font = Font.Courier;
17         private const float fontSize = 9.0f;
18         private static PageDimensions pageDimensions = new PageDimensions( PageSize.Letter, PageOrientation.Landscape, 36.0f );
19         private static float left = font.GetTextWidth( "0000", fontSize ) + 6;
20         private static float top = Font.Helvetica.GetDefaultLeading( 10.0f ) + 3.0f;
21         private static float bottom = pageDimensions.Body.Height - 12;
22         private static float lineWidth = pageDimensions.Body.Width - left;
23         private const string tabSpaces = "    ";
24         private static Template template;
25         private const string footerText = "<p align=\"center\">This PDF was generated in real-time using
        <i>Dynamic</i><b>PDF</b>; Generator for .NET. www.dynamicpdf.com</p>";
26
27         //////////////////////////////////////
28
29         private int lineNum = 1;
30         private string currentUrl = "";
31         private string aspUrl = "";
32         private string codeBehindUrl = "";
33         private float currentPosition = 0;
34         private ceTe.DynamicPDF.Page page = null;
35         private Document document = null;
36
37         #endregion
38
39         #region Constructors
40
41         static SourceView()
42         {
43             template = new Template();
44             template.Elements.Add( new Line( 0, top - 1, pageDimensions.Body.Width, top - 1 ) );
45             template.Elements.Add( new Line( 0, top - 1, pageDimensions.Body.Width, top - 1 ) );
46             template.Elements.Add( new Line( left - 3, top - 1, left - 3, bottom ) );
47             template.Elements.Add( new Line( 0, bottom, pageDimensions.Body.Width, bottom ) );
48             template.Elements.Add( new FormattedTextArea( footerText, 0, bottom + 1.0f, pageDimensions.Body.Width, 10,
                FontFamily.Helvetica, 10, false ) );
49         }
50     }
51 }
```

```
49         template.Elements.Add( new Link( 0, bottom + 1, pageDimensions.Body.Width, 14, new UrlAction(
50     }
51     #endregion
52
53     #region Page Load
54
55     private void Page_Load(object sender, System.EventArgs e)
56     {
57         // Retrieve the class query string variable
58         string classPath = Request.QueryString["class"];
59         if( classPath == null )
60         {
61             Response.Write( " No File Selected ");
62             return;
63         }
64
65         if( classPath.IndexOf( "/" ) >= 0 || classPath.IndexOf( "\\" ) >= 0 )
66         {
67             Response.Write("For security reasons slashes are not allowed in the class name ");
68             return;
69         }
70
71         // Set the file paths based on the class variable
72         aspUrl = Server.MapPath( classPath + ".aspx" );
73         codeBehindUrl = aspUrl + ".cs" ;
74         if( !File.Exists( aspUrl ) || !File.Exists( codeBehindUrl ) )
75         {
76             Response.Write("Class name does not exist");
77             return;
78         }
79
80         // Create a document and set it's properties
81         document = new Document();
82         document.Creator = "SourceView.aspx";
83         document.Author = "ceTe Software";
84         document.Title = "Source View";
85
86         // Set the template to the document
87         document.Template = template;
88
89         // Add a bookmark to the ASPX page
90         document.Outlines.Add( "ASPX Page", new XYDestination( 1, 0, 0 ) );
91
92         // Add the source of the ASPX page to the document
93         AddSource( aspUrl );
94
95         // Add a bookmark to the code behind page
96         document.Outlines.Add( "Code Behind Page", new XYDestination( document.Pages.Count + 1, 0, 0 ) );
97     }
```

```
98         // Add the source of the code behind page to the document
99         AddSource( codeBehindUrl );
100
101         // Outputs the document to the current web page
102         document.DrawToWeb( "SourceView.pdf" );
103     }
104 #endregion
105
106 #region Private Methods
107
108 private void AddSource( string filePath )
109 {
110     // reset the line number
111     lineNum = 1;
112     currentUrl = System.IO.Path.GetFileName(filePath);
113
114     // Add the initial source page to the document
115     AddSourcePage();
116
117     // Open file
118     StreamReader sourceCode = new StreamReader( filePath );
119
120     // Loop over each line
121     string lineText;
122     while( ( lineText = sourceCode.ReadLine() ) != null )
123     {
124         AddSourceLine( lineText );
125     }
126
127     // Close the file
128     sourceCode.Close();
129 }
130
131 private void AddSourceLine( string lineText )
132 {
133     // Clean up text
134     string trimmedLineText = lineText.Trim();
135     TextArea line = new TextArea( lineText.Replace( "\t", tabSpaces ), left, currentPosition, lineWidth, bottom -
currentPosition, font, fontSize );
136     float fltHeight = line.GetRequiredHeight();
137
138     // Set line color if it is a comment
139     if( trimmedLineText.StartsWith( "//" ) || trimmedLineText.StartsWith( "'" ) ) line.TextColor = RgbColor.Green;
140
141     // Add a new page if line does not fit
142     if( currentPosition + fltHeight + 2 > bottom )
143     {
144         AddSourcePage();
145         line.Y = currentPosition;
146     }
147 }
```

```
147
148     // Add line number and line text to page
149     page.Elements.Add( new Label( lineNum.ToString(), 0, currentPosition, left - 6, fontSize, font, fontSize,
TextAlign.Right, RgbColor.Gray ) );
150     page.Elements.Add( line );
151
152     // increment for next line
153     currentPosition += fltHeight;
154     lineNum++;
155 }
156
157 private void AddSourcePage()
158 {
159     // Adds a new page to the document
160     page = new ceTe.DynamicPDF.Page( pageDimensions );
161
162     // Adds the file name to the page
163     TextArea url = new TextArea( currentUrl, 0, 0, page.Dimensions.Body.Width, 100, Font.Helvetica, 10.0f,
RgbColor.DarkBlue );
164     page.Elements.Add( url );
165
166     // Resets the current position for the page
167     currentPosition = top + 1.0f;
168
169     // Adds the page to the document
170     document.Pages.Add( page );
171 }
172 #endregion
173 }
174 }
```