

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

```
1 using System;
2 using System.Text;
3 using System.IO;
4 using System.Security.Cryptography;
5 using ceTe.DynamicPDF;
6 using ceTe.DynamicPDF.IO;
7 using ceTe.DynamicPDF.Text;
8 using ceTe.DynamicPDF.PageElements;
9 using ceTe.DynamicPDF.PageElements.Forms;
10 using ceTe.DynamicPDF.PageElements.BarCoding;
11 using ceTe.DynamicPDF.PageElements.Charting;
12 using ceTe.DynamicPDF.PageElements.Charting.Axes;
13 using ceTe.DynamicPDF.PageElements.Charting.Series;
14 using ceTe.DynamicPDF.PageElements.Charting.Values;
15 using ceTe.DynamicPDF.PageElements.Html;
16
17 namespace Web_CSharp.GeneratorExamples
18 {
19     public partial class AllPageElements : System.Web.UI.Page
20     {
21         #region Member Variables
22         private int pageCounter = 0;
23         #endregion
24
25         #region StopSign Custom Page Element Class
26         /// <summary>
27         /// Custom page element
28         /// </summary>
29         public class StopSign : ceTe.DynamicPDF.PageElement
30         {
31             #region Member Variables
32             private float x = 0.0f;
33             private float y = 0.0f;
34             private float w = 0.0f;
35             private float h = 0.0f;
36             #endregion
37
38             #region Constructors
39             /// <summary>
40             /// Initilizes a new instance of the <see cref="MyPageElement"/> class.
41             /// </summary>
42             /// <param name="x">X coordinate of the page element.</param>
43             /// <param name="y">Y coordinate of the page element.</param>
44             /// <param name="width">Width of the page element.</param>
45             /// <param name="height">Height of the page element.</param>
46             public StopSign(float x, float y, float width, float height)
47             {
48                 this.x = x;
49                 this.y = y;
50                 this.w = width;
```

```
51         this.h = height;
52     }
53     #endregion
54
55     #region Public methods
56     /// <summary>
57     /// Overriden. Draws a MyPageElement page element to the page.
58     /// </summary>
59     /// <param name="writer"><see cref="PageWriter"/> object to recieve the PDF output.</param>
60     public override void Draw(PageWriter writer)
61     {
62         float v = h / (float)3.414;
63
64         // End text mode
65         writer.Write_ET();
66         writer.Write_m_(x + v, y);
67         writer.Write_RG(RgbColor.White);
68         writer.Write_rg_(RgbColor.Red);
69
70         // Draws the axis lines to the page
71         writer.Write_l_(x + h - v, y);
72         writer.Write_l_(x + h, y + v);
73         writer.Write_l_(x + h, y + h - v);
74         writer.Write_l_(x + h - v, y + h);
75         writer.Write_l_(x + v, y + h);
76         writer.Write_l_(x, y + h - v);
77         writer.Write_l_(x, y + v);
78         writer.Write_b_();
79         char[] text = new char[] { 'S', 'T', 'O', 'P' };
80
81         // Begin text mode
82         writer.Write_BT();
83         writer.SetFont(ceTe.DynamicPDF.Font.Helvetica, 16);
84         writer.Write_rg_(RgbColor.White);
85         writer.Write_Tm((float)88.5, y + (float)(w / 1.65));
86
87         // Wirte the text
88         writer.Write_Tj_(text, false);
89
90         // End text mode
91         writer.Write_ET();
92     }
93     #endregion
94 }
95 #endregion
96
97 #region Page_Load Method
98 protected void Page_Load(object sender, System.EventArgs e)
99 {
100     // Create a document and set it's properties
```

```
101         ceTe.DynamicPDF.Document document = new ceTe.DynamicPDF.Document();
102         document.Creator = "AllPageElements.aspx";
103         document.Author = "ceTe Software";
104         document.Title = "All Page Elements";
105
106         // Add the top level outline to the document
107         Outline parentOutline = document.Outlines.Add("Top Level Outline", new XYDestination(1, 0, 0));
108         parentOutline.Color = new WebColor("0000FF");
109         parentOutline.Style = TextStyle.Bold;
110
111         // Add pages to the document
112         AddPageElementPage(document, parentOutline);
113         AddTableAndOther(document, parentOutline);
114         AddChartsPage(document, parentOutline);
115         AddFormElementPage(document, parentOutline);
116         AddUPCEANPage(document, parentOutline);
117         AddPostalBarcodePage(document, parentOutline);
118         AddIsbnIssnIsmnPage(document, parentOutline);
119         AddBarcodePage(document, parentOutline);
120         Add2DBarcodePage(document, parentOutline);
121         AddWatermark(document, parentOutline);
122
123         // Outputs the document to the current web page
124         document.DrawToWeb("AllPageElements.pdf");
125     }
126
127
128     #endregion
129
130     #region Private Methods
131
132     private void AddIsbn(Group elements, float x, float y)
133     {
134         AddCaptionAndRectangle(elements, "ISBN Barcode", x, y, 204, 99);
135         Isbn barCode = new Isbn("978-1-23-456789-7", x, y + 21);
136         barCode.ShowText = true;
137         barCode.X += (204 - barCode.GetSymbolWidth()) / 2;
138         elements.Add(barCode);
139     }
140
141     private void AddIsbnSup2(Group elements, float x, float y)
142     {
143         AddCaptionAndRectangle(elements, "ISBN Barcode, 2 digit supplement", x, y, 204, 99);
144         IsbnSup2 barCode = new IsbnSup2("978-1-23-456789-7", "12", x, y + 21);
145         barCode.ShowText = true;
146         barCode.X += (204 - barCode.GetSymbolWidth()) / 2;
147         elements.Add(barCode);
148     }
149
150     private void AddIsbnSup5(Group elements, float x, float y)
```

```
151     {
152         AddCaptionAndRectangle(elements, "ISBN Barcode, 5 digit supplement", x, y, 204, 99);
153         IsbnSup5 barCode = new IsbnSup5("978-1-23-456789-7", "12345", x, y + 21);
154         barCode.ShowText = true;
155         barCode.X += (204 - barCode.GetSymbolWidth()) / 2;
156         elements.Add(barCode);
157     }
158
159     private void AddIsmn(Group elements, float x, float y)
160     {
161         AddCaptionAndRectangle(elements, "ISMN Barcode", x, y, 204, 99);
162         Ismn barCode = new Ismn("979-0-1234-5678", x, y + 21);
163         barCode.ShowText = true;
164         barCode.X += (204 - barCode.GetSymbolWidth()) / 2;
165         elements.Add(barCode);
166     }
167
168     private void AddIsmnSup2(Group elements, float x, float y)
169     {
170         AddCaptionAndRectangle(elements, "ISMN Barcode, 2 digit supplement", x, y, 204, 99);
171         IsmnSup2 barCode = new IsmnSup2("979-0-1234-5678", "12", x, y + 21);
172         barCode.ShowText = true;
173         barCode.X += (204 - barCode.GetSymbolWidth()) / 2;
174         elements.Add(barCode);
175     }
176
177     private void AddIsmnSup5(Group elements, float x, float y)
178     {
179         AddCaptionAndRectangle(elements, "ISMN Barcode, 5 digit supplement", x, y, 204, 99);
180         IsmnSup5 barCode = new IsmnSup5("979-0-1234-5678", "12345", x, y + 21);
181         barCode.ShowText = true;
182         barCode.X += (204 - barCode.GetSymbolWidth()) / 2;
183         elements.Add(barCode);
184     }
185
186     private void AddIssn(Group elements, float x, float y)
187     {
188         AddCaptionAndRectangle(elements, "ISSN Barcode", x, y, 204, 99);
189         Issn barCode = new Issn("977-1234-56700", x, y + 21);
190         barCode.ShowText = true;
191         barCode.X += (204 - barCode.GetSymbolWidth()) / 2;
192         elements.Add(barCode);
193     }
194
195     private void AddIssnSup2(Group elements, float x, float y)
196     {
197         AddCaptionAndRectangle(elements, "ISSN Barcode, 2 digit supplement", x, y, 204, 99);
198         IssnSup2 barCode = new IssnSup2("977-1234-56700", "12", x, y + 21);
199         barCode.ShowText = true;
200         barCode.X += (204 - barCode.GetSymbolWidth()) / 2;
```

```
201     elements.Add(barCode);
202 }
203
204 private void AddIssnSup5(Group elements, float x, float y)
205 {
206     AddCaptionAndRectangle(elements, "ISSN Barcode, 5 digit supplement", x, y, 204, 99);
207     IssnSup5 barCode = new IssnSup5("977-1234-56700", "12345", x, y + 21);
208     barCode.ShowText = true;
209     barCode.X += (204 - barCode.GetSymbolWidth()) / 2;
210     elements.Add(barCode);
211 }
212
213 private void AddSingaporePost(Group elements, float x, float y)
214 {
215     AddCaptionAndRectangle(elements, "Singapore Post Barcode", x, y, 225, 80);
216     BarCode barCode = new SingaporePost("208154", x, y + 15, true);
217     barCode.Height = 35;
218     barCode.X += (225 - barCode.GetSymbolWidth()) / 2;
219     barCode.Y += (65 - barCode.Height) / 2;
220     elements.Add(barCode);
221 }
222
223 private void AddKix(Group elements, float x, float y)
224 {
225     AddCaptionAndRectangle(elements, "Kix (Royal TNT Post) Barcode", x, y, 225, 80);
226     BarCode barCode = new Kix("1231FZ13XHS", x, y + 15, true);
227     barCode.Height = 35;
228     barCode.X += (225 - barCode.GetSymbolWidth()) / 2;
229     barCode.Y += (65 - barCode.Height) / 2;
230     elements.Add(barCode);
231 }
232
233 private void AddDeutschePostLeitcode(Group elements, float x, float y)
234 {
235     AddCaptionAndRectangle(elements, "Deutsche Post Leitcode", x, y, 225, 80);
236     BarCode barCode = new DeutschePostLeitcode("1234567890123", x, y + 15, 48);
237     barCode.X += (225 - barCode.GetSymbolWidth()) / 2;
238     barCode.Y += (65 - barCode.Height) / 2;
239     elements.Add(barCode);
240 }
241
242 private void AddDeutschePostIdentcode(Group elements, float x, float y)
243 {
244     AddCaptionAndRectangle(elements, "Deutsche Post Identcode", x, y, 225, 80);
245     BarCode barCode = new DeutschePostIdentcode("12345678901", x, y + 15, 48);
246     barCode.X += (225 - barCode.GetSymbolWidth()) / 2;
247     barCode.Y += (65 - barCode.Height) / 2;
248     elements.Add(barCode);
249 }
250
```

```
251     private void AddAustraliaPost(Group elements, float x, float y)
252     {
253         AddCaptionAndRectangle(elements, "Australia Post Barcode", x, y, 225, 80);
254         BarCode barCode = new AustraliaPost("1139549554", x, y + 15, true);
255         barCode.Height = 35;
256         barCode.X += (225 - barCode.GetSymbolWidth()) / 2;
257         barCode.Y += (65 - barCode.Height) / 2;
258         elements.Add(barCode);
259     }
260
261     private void AddAztec(Group elements, float x, float y)
262     {
263         AddCaptionAndRectangle(elements, "Aztec Barcode", x, y, 225, 100);
264         Aztec aztec = new Aztec("Aztec Barcode", x, y + 15, AztecSymbolSize.Auto);
265         aztec.XDimension = 2.5f;
266         aztec.X += (225 - aztec.GetSymbolWidth()) / 2;
267         aztec.Y += (85 - aztec.GetSymbolHeight()) / 2;
268         elements.Add(aztec);
269     }
270
271     private void AddStackedGs1Expanded(Group elements, float x, float y)
272     {
273         AddCaptionAndRectangle(elements, "Stacked GS1 DataBar Expanded Barcode", x, y, 225, 100);
274         StackedGS1DataBar barCode = new StackedGS1DataBar("(01)1234567890123", x, y + 15, 25,
StackedGS1DataBarType.ExpandedStacked);
275         barCode.X += (225 - barCode.GetSymbolWidth()) / 2;
276         barCode.Y += (85 - barCode.GetSymbolHeight()) / 2;
277         elements.Add(barCode);
278     }
279
280     private void AddStackedGs1Stacked(Group elements, float x, float y)
281     {
282         AddCaptionAndRectangle(elements, "Stacked GS1 DataBar Barcode", x, y, 225, 100);
283         StackedGS1DataBar barCode = new StackedGS1DataBar("(01)1234567890123", x, y + 15, 25,
StackedGS1DataBarType.Stacked);
284         barCode.X += (225 - barCode.GetSymbolWidth()) / 2;
285         barCode.Y += (85 - barCode.GetSymbolHeight()) / 2;
286         elements.Add(barCode);
287     }
288
289     private void AddStackedGs1Omni(Group elements, float x, float y)
290     {
291         AddCaptionAndRectangle(elements, "Stacked GS1 DataBar Omnidirectional Barcode", x, y, 225, 100);
292         StackedGS1DataBar barCode = new StackedGS1DataBar("(01)1234567890123", x, y + 15, 25,
StackedGS1DataBarType.StackedOmnidirectional);
293         barCode.X += (225 - barCode.GetSymbolWidth()) / 2;
294         barCode.Y += (85 - barCode.GetSymbolHeight()) / 2;
295         elements.Add(barCode);
296     }
297
```

```
298 private void AddGs1DataBarExpanded(Group elements, float x, float y)
299 {
300     AddCaptionAndRectangle(elements, "GS1 DataBar Expanded Barcode", x, y, 225, 80);
301     Barcode barCode = new GS1DataBar("(01)1234567890123", x, y + 15, 48, GS1DataBarType.Expanded);
302     barCode.X += (225 - barCode.GetSymbolWidth()) / 2;
303     barCode.Y += (65 - barCode.Height) / 2;
304     elements.Add(barCode);
305 }
306
307 private void AddGs1DataBarLimited(Group elements, float x, float y)
308 {
309     AddCaptionAndRectangle(elements, "GS1 DataBar Limited Barcode", x, y, 225, 80);
310     Barcode barCode = new GS1DataBar("(01)1234567890123", x, y + 15, 48, GS1DataBarType.Limited);
311     barCode.X += (225 - barCode.GetSymbolWidth()) / 2;
312     barCode.Y += (65 - barCode.Height) / 2;
313     elements.Add(barCode);
314 }
315
316 private void AddGs1DataBarOmnidirectional(Group elements, float x, float y)
317 {
318     AddCaptionAndRectangle(elements, "GS1 DataBar Omnidirectional Barcode", x, y, 225, 80);
319     Barcode barCode = new GS1DataBar("(01)9889876543210", x, y + 15, 48, GS1DataBarType.Omnidirectional);
320     barCode.X += (225 - barCode.GetSymbolWidth()) / 2;
321     barCode.Y += (65 - barCode.Height) / 2;
322     elements.Add(barCode);
323 }
324
325 private void AddEan14(Group elements, float x, float y)
326 {
327     AddCaptionAndRectangle(elements, "EAN-14 Barcode", x, y, 225, 80);
328     Barcode barCode = new Ean14("1234567890123", x, y + 15, 48);
329     barCode.X += (225 - barCode.GetSymbolWidth()) / 2;
330     barCode.Y += (65 - barCode.Height) / 2;
331     elements.Add(barCode);
332 }
333
334 private void AddItf14(Group elements, float x, float y)
335 {
336     AddCaptionAndRectangle(elements, "ITF-14 Barcode", x, y, 225, 80);
337     Barcode barCode = new Itf14("1234567890123", x, y + 15, 48);
338     barCode.X += (225 - barCode.GetSymbolWidth()) / 2;
339     barCode.Y += (65 - barCode.Height) / 2;
340     elements.Add(barCode);
341 }
342
343 private void AddIata25(Group elements, float x, float y)
344 {
345     AddCaptionAndRectangle(elements, "IATA-25 Barcode", x, y, 225, 80);
346     Barcode barCode = new Iata25("1234567", x, y + 15, 48);
347     barCode.X += (225 - barCode.GetSymbolWidth()) / 2;
```



```
348         barCode.Y += (65 - barCode.Height) / 2;
349         elements.Add(barCode);
350     }
351
352     private void AddExtendedCode93(Group elements, float x, float y)
353     {
354         AddCaptionAndRectangle(elements, "Extended Code 93 Barcode", x, y, 225, 80);
355         BarCode barCode = new Code93("Code 93", x, y + 15, 48);
356         barCode.X += (225 - barCode.GetSymbolWidth()) / 2;
357         barCode.Y += (65 - barCode.Height) / 2;
358         elements.Add(barCode);
359     }
360
361     private void AddCode93(Group elements, float x, float y)
362     {
363         AddCaptionAndRectangle(elements, "Code 93 Barcode", x, y, 225, 80);
364         BarCode barCode = new Code93("CODE 93", x, y + 15, 48);
365         barCode.X += (225 - barCode.GetSymbolWidth()) / 2;
366         barCode.Y += (65 - barCode.Height) / 2;
367         elements.Add(barCode);
368     }
369
370     private void AddExtendedCode39(Group elements, float x, float y)
371     {
372         AddCaptionAndRectangle(elements, "Extended Code 3 of 9 Barcode", x, y, 225, 80);
373         BarCode barCode = new Code39("Code 39", x, y + 15, 48);
374         barCode.X += (225 - barCode.GetSymbolWidth()) / 2;
375         barCode.Y += (65 - barCode.Height) / 2;
376         elements.Add(barCode);
377     }
378
379     private void AddCode11(Group elements, float x, float y)
380     {
381         AddCaptionAndRectangle(elements, "Code 11 Barcode", x, y, 225, 80);
382         BarCode barCode = new Code11("12345678", x, y + 15, 48);
383         barCode.X += (225 - barCode.GetSymbolWidth()) / 2;
384         barCode.Y += (65 - barCode.Height) / 2;
385         elements.Add(barCode);
386     }
387
388     private void AddMSICode(Group elements, int x, int y)
389     {
390         AddCaptionAndRectangle(elements, "MSI Barcode", x, y, 225, 80);
391         MsiBarcode msi = new MsiBarcode("1234567890", x, y, 50, true);
392         msi.Y += (80 - msi.Height) / 2 + 8;
393         msi.X += (225 - msi.GetSymbolWidth()) / 2;
394         elements.Add(msi);
395     }
396     private void AddRm4sccCode(Group elements, int x, int y)
397     {
```

```
398     AddCaptionAndRectangle(elements, "RM4SCC Barcode", x, y, 225, 80);
399     Rm4scc rm4sccCode = new Rm4scc("1457826RA", x, y, true);
400     rm4sccCode.Height = 35;
401     rm4sccCode.Y += (65 - rm4sccCode.Height) / 2 + 13;
402     rm4sccCode.X += (225 - rm4sccCode.GetSymbolWidth()) / 2;
403     elements.Add(rm4sccCode);
404 }
405 private void AddBarCodePage(Document document, Outline parentOutline)
406 {
407     // Increments a page counter
408     pageCounter++;
409
410     // Adds to outlines to the page
411     parentOutline.ChildOutlines.Add("Page " + pageCounter + " - Other Linear Barcodes", new ZoomDestination(pageCounter,
PageZoom.FitPage));
412
413     // Create a page to add to the document
414     ceTe.DynamicPDF.Page page = new ceTe.DynamicPDF.Page(PageSize.Letter, PageOrientation.Portrait, 54);
415
416     // Uncomment the line below to add a layout grid to the page
417     //page.Elements.Add( new LayoutGrid() );
418
419     // Add the page elements to the page
420     AddCodabar(page.Elements, 0, 0);
421     AddCode11(page.Elements, 0, 88);
422     AddCode128(page.Elements, 0, 176);
423     AddCode25(page.Elements, 0, 264);
424     AddCode39(page.Elements, 0, 352);
425     AddExtendedCode39(page.Elements, 0, 440);
426     AddCode93(page.Elements, 0, 528);
427     AddExtendedCode93(page.Elements, 0, 616);
428
429     AddIata25(page.Elements, 279, 0);
430     AddInterleaved25(page.Elements, 279, 88);
431     AddItf14(page.Elements, 279, 176);
432     AddMSICode(page.Elements, 279, 264);
433     AddEan14(page.Elements, 279, 352);
434     AddGslDataBarOmnidirectional(page.Elements, 279, 440);
435     AddGslDataBarLimited(page.Elements, 279, 528);
436     AddGslDataBarExpanded(page.Elements, 279, 616);
437
438     document.Pages.Add(page);
439 }
440 private void Add2DBarCodePage(Document document, Outline parentOutline)
441 {
442     // Increments a page counter
443     pageCounter++;
444
445     // Adds to outlines to the page
```

```
446     parentOutline.ChildOutlines.Add("Page " + pageCounter + " - 2D and Stacked GS1 Databar Barcodes", new
447
448     // Create a page to add to the document
449     ceTe.DynamicPDF.Page page = new ceTe.DynamicPDF.Page(PageSize.Letter, PageOrientation.Portrait, 54);
450
451     // Uncomment the line below to add a layout grid to the page
452     //page.Elements.Add( new LayoutGrid() );
453
454     // Add the page elements to the page
455     AddStackedGs1Omni(page.Elements, 0, 0);
456     AddStackedGs1Stacked(page.Elements, 0, 108);
457     AddStackedGs1Expanded(page.Elements, 0, 216);
458     AddAztec(page.Elements, 0, 324);
459     AddQrCode(page.Elements, 0, 432);
460
461     AddPdf417(page.Elements, 279, 0);
462     AddPdf417Macro(page.Elements, 279, 108);
463     AddDataMatrixBarcode(page.Elements, 279, 324);
464
465     document.Pages.Add(page);
466 }
467 private void AddPostalBarCodePage(Document document, Outline parentOutline)
468 {
469     // Increments a page counter
470     pageCounter++;
471
472     // Adds to outlines to the page
473     parentOutline.ChildOutlines.Add("Page " + pageCounter + " - Postal Barcodes", new ZoomDestination(pageCounter,
PageZoom.FitPage));
474
475     // Create a page to add to the document
476     ceTe.DynamicPDF.Page page = new ceTe.DynamicPDF.Page(PageSize.Letter, PageOrientation.Portrait, 54);
477
478     // Uncomment the line below to add a layout grid to the page
479     //page.Elements.Add( new LayoutGrid() );
480
481     // Add the page elements to the page
482     AddAustraliaPost(page.Elements, 0, 0);
483     AddDeutschePostIdentcode(page.Elements, 0, 120);
484     AddDeutschePostLeitcode(page.Elements, 0, 240);
485     AddKix(page.Elements, 0, 360);
486
487     AddIntelligentMailBarcode(page.Elements, 279, 0);
488     AddRm4sccCode(page.Elements, 279, 120);
489     AddPostnet(page.Elements, 279, 240);
490     AddSingaporePost(page.Elements, 279, 360);
491
492     document.Pages.Add(page);
493 }
```

```
494     private void AddIsbnIssnIsmnPage(Document document, Outline parentOutline)
495     {
496         // Increments a page counter
497         pageCounter++;
498
499         // Adds to outlines to the page
500         parentOutline.ChildOutlines.Add("Page " + pageCounter + " - ISBN/ISMN/ISSN Barcodes", new
ZoomDestination(pageCounter, PageZoom.FitPage));
501
502         // Create a page to add to the document
503         ceTe.DynamicPDF.Page page = new ceTe.DynamicPDF.Page(PageSize.Letter, PageOrientation.Landscape, 54);
504
505         // Uncomment the line below to add a layout grid to the page
506         //page.Elements.Add( new LayoutGrid() );
507
508         // Add the page elements to the page
509         AddIsbn(page.Elements, 0, 0);
510         AddIsbnSup2(page.Elements, 240, 0);
511         AddIsbnSup5(page.Elements, 480, 0);
512         AddIsmn(page.Elements, 0, 150);
513         AddIsmnSup2(page.Elements, 240, 150);
514         AddIsmnSup5(page.Elements, 480, 150);
515         AddIssn(page.Elements, 0, 300);
516         AddIssnSup2(page.Elements, 240, 300);
517         AddIssnSup5(page.Elements, 480, 300);
518
519         // Add the page to the document
520         document.Pages.Add(page);
521     }
522     private void AddChartsPage(Document document, Outline parentOutline)
523     {
524         // Increments a page counter
525         pageCounter++;
526
527         // Adds outlines to the page
528         parentOutline.ChildOutlines.Add("Page " + pageCounter + " - Charts", new ZoomDestination(pageCounter,
PageZoom.FitPage));
529
530         // Create a page
531         ceTe.DynamicPDF.Page page = new ceTe.DynamicPDF.Page(PageSize.Letter, PageOrientation.Landscape);
532         AddAreaChart(page.Elements, 0, 25);
533         AddLineChart(page.Elements, 375, 25);
534         AddPieChart(page.Elements, 0, 260);
535         AddBarChart(page.Elements, 375, 260);
536         document.Pages.Add(page);
537     }
538     private void AddAreaChart(Group elements, float x, float y)
539     {
540         AddCaptionAndRectangle(elements, "Stacked Area Chart", x, y, 325, 225);
541     }
```

```
542     // Create a chart
543     Chart chart = new Chart(x + 20, y + 25, 300, 200);
544
545     // Create a plot area
546     PlotArea plotArea = chart.PrimaryPlotArea;
547
548     // Create header title and add it to the chart
549     Title title1 = new Title("Website Visitors");
550     chart.HeaderTitles.Add(title1);
551
552     // Create indexed stacked area series elements and add values to it
553     IndexedStackedAreaSeriesElement seriesElement1 = new IndexedStackedAreaSeriesElement("Website A");
554     seriesElement1.Values.Add(new float[] { 5, 7, 9, 6 });
555     IndexedStackedAreaSeriesElement seriesElement2 = new IndexedStackedAreaSeriesElement("Website B");
556     seriesElement2.Values.Add(new float[] { 4, 2, 5, 8 });
557     IndexedStackedAreaSeriesElement seriesElement3 = new IndexedStackedAreaSeriesElement("Website C");
558     seriesElement3.Values.Add(new float[] { 2, 4, 6, 9 });
559
560     // Create autogradient and add it to the series
561     AutoGradient autogradient1 = new AutoGradient(90f, CmykColor.Red, CmykColor.IndianRed);
562     seriesElement1.Color = autogradient1;
563     AutoGradient autogradient2 = new AutoGradient(90f, CmykColor.Green, CmykColor.YellowGreen);
564     seriesElement2.Color = autogradient2;
565     AutoGradient autogradient3 = new AutoGradient(120f, CmykColor.Blue, CmykColor.LightBlue);
566     seriesElement3.Color = autogradient3;
567
568     // Create a Indexed Stacked Area Series
569     IndexedStackedAreaSeries areaSeries = new IndexedStackedAreaSeries();
570
571     // Add indexed stacked area series elements to the Indexed Stacked Area Series
572     areaSeries.Add(seriesElement1);
573     areaSeries.Add(seriesElement2);
574     areaSeries.Add(seriesElement3);
575
576     // Add series to the plot area
577     plotArea.Series.Add(areaSeries);
578
579     // Create a title and add it to the YAxis
580     Title lTitle = new Title("Visitors (in millions)");
581     areaSeries.YAxis.Titles.Add(lTitle);
582
583     //Adding AxisLabels to the XAxis
584     areaSeries.XAxis.Labels.Add(new IndexedXAxisLabel("Q1", 0));
585     areaSeries.XAxis.Labels.Add(new IndexedXAxisLabel("Q2", 1));
586     areaSeries.XAxis.Labels.Add(new IndexedXAxisLabel("Q3", 2));
587     areaSeries.XAxis.Labels.Add(new IndexedXAxisLabel("Q4", 3));
588     elements.Add(chart);
589 }
590 private void AddLineChart(Group elements, float x, float y)
591 {
```

```
592         AddCaptionAndRectangle(elements, "Line Chart", x, y, 325, 225);
593
594         // Create a chart
595         Chart chart = new Chart(x + 20, y + 25, 300, 200);
596
597         // Create a plot area
598         PlotArea plotArea = chart.PrimaryPlotArea;
599
600         // Create header title and add it to the chart
601         Title title1 = new Title("Website Visitors");
602         chart.HeaderTitles.Add(title1);
603
604         // Create a indexed line series and add values to it
605         IndexedLineSeries lineSeries1 = new IndexedLineSeries("Website A");
606         lineSeries1.Values.Add(new float[] { 5, 7, 9, 6 });
607         IndexedLineSeries lineSeries2 = new IndexedLineSeries("Website B");
608         lineSeries2.Values.Add(new float[] { 4, 2, 5, 8 });
609         IndexedLineSeries lineSeries3 = new IndexedLineSeries("Website C");
610         lineSeries3.Values.Add(new float[] { 2, 4, 6, 9 });
611
612         // Add indexed line series to the plot area
613         plotArea.Series.Add(lineSeries1);
614         plotArea.Series.Add(lineSeries2);
615         plotArea.Series.Add(lineSeries3);
616
617         // Create a title and add it to the yaxis
618         Title lTitle = new Title("Visitors (in millions)");
619         lineSeries1.YAxis.Titles.Add(lTitle);
620
621         //Adding AxisLabels to the XAxis
622         lineSeries1.XAxis.Labels.Add(new IndexedXAxisLabel("Q1", 0));
623         lineSeries1.XAxis.Labels.Add(new IndexedXAxisLabel("Q2", 1));
624         lineSeries1.XAxis.Labels.Add(new IndexedXAxisLabel("Q3", 2));
625         lineSeries1.XAxis.Labels.Add(new IndexedXAxisLabel("Q4", 3));
626         elements.Add(chart);
627
628     }
629     private void AddPieChart(Group elements, float x, float y)
630     {
631         AddCaptionAndRectangle(elements, "Pie Chart", x, y, 325, 225);
632
633         // Create a chart
634         Chart chart = new Chart(x + 20, y + 25, 300, 200);
635
636         // Add a plot area to the chart
637         PlotArea plotArea = chart.PlotAreas.Add(50, 50, 300, 300);
638
639         // Create the Header title and add it to the chart
640         Title tTitle = new Title("Website Visitors (in millions)");
641         chart.HeaderTitles.Add(tTitle);
```

```
642
643     // Create autogradient and add it to the series
644     AutoGradient autogradient1 = new AutoGradient(90f, CmykColor.Red, CmykColor.IndianRed);
645     AutoGradient autogradient2 = new AutoGradient(90f, CmykColor.Green, CmykColor.YellowGreen);
646     AutoGradient autogradient3 = new AutoGradient(120f, CmykColor.Blue, CmykColor.LightBlue);
647
648     // Create a scalar datalabel
649     ScalarDataLabel da = new ScalarDataLabel(true, false, false);
650
651     // Create a pie series
652     PieSeries pieSeries = new PieSeries();
653
654     // Set scalar datalabel to the pie series
655     pieSeries.DataLabel = da;
656
657     // Add series to the plot area
658     plotArea.Series.Add(pieSeries);
659
660     //Add pie series elements to the pie series
661     pieSeries.Elements.Add(27, "Website A");
662     pieSeries.Elements.Add(19, "Website B");
663     pieSeries.Elements.Add(21, "Website C");
664     pieSeries.Elements[0].Color = autogradient1;
665     pieSeries.Elements[1].Color = autogradient2;
666     pieSeries.Elements[2].Color = autogradient3;
667     elements.Add(chart);
668 }
669 private void AddBarChart(Group elements, float x, float y)
670 {
671     AddCaptionAndRectangle(elements, "Stacked Bar Chart", x, y, 325, 225);
672
673     // Create a chart
674     Chart chart = new Chart(x + 20, y + 25, 300, 200);
675
676     // Create a plot area
677     PlotArea plotArea = chart.PrimaryPlotArea;
678
679     // Create header title and add it to the chart
680     Title title1 = new Title("Website Visitors");
681     chart.HeaderTitles.Add(title1);
682
683     // Create a indexed stacked bar series elements and add values to it
684     IndexedStackedBarSeriesElement seriesElement1 = new IndexedStackedBarSeriesElement("Website A");
685     seriesElement1.Values.Add(new float[] { 5, 7, 9, 6 });
686     IndexedStackedBarSeriesElement seriesElement2 = new IndexedStackedBarSeriesElement("Website B");
687     seriesElement2.Values.Add(new float[] { 4, 2, 5, 8 });
688     IndexedStackedBarSeriesElement seriesElement3 = new IndexedStackedBarSeriesElement("Website C");
689     seriesElement3.Values.Add(new float[] { 2, 4, 6, 9 });
690
691     // Create a Indexed Stacked Bar Series
```

```
692     IndexedStackedBarSeries barSeries = new IndexedStackedBarSeries();
693
694     // Create autogradient and add it to the series
695     AutoGradient autogradient1 = new AutoGradient(90f, CmykColor.Red, CmykColor.IndianRed);
696     AutoGradient autogradient2 = new AutoGradient(90f, CmykColor.Green, CmykColor.YellowGreen);
697     AutoGradient autogradient3 = new AutoGradient(90f, CmykColor.Blue, CmykColor.LightBlue);
698     seriesElement1.Color = autogradient1;
699     seriesElement2.Color = autogradient2;
700     seriesElement3.Color = autogradient3;
701
702     // Add indexed stacked bar series elements to the Indexed Stacked Bar Series
703     barSeries.Add(seriesElement1);
704     barSeries.Add(seriesElement2);
705     barSeries.Add(seriesElement3);
706
707     //Add series to the plot area
708     plotArea.Series.Add(barSeries);
709
710     // Create a title and add it to the xaxis
711     Title lTitle = new Title("Visitors (in millions)");
712     barSeries.XAxis.Titles.Add(lTitle);
713
714     //Adding AxisLabels to the yaxis
715     barSeries.YAxis.Labels.Add(new IndexedYAxisLabel("Q1", 0));
716     barSeries.YAxis.Labels.Add(new IndexedYAxisLabel("Q2", 1));
717     barSeries.YAxis.Labels.Add(new IndexedYAxisLabel("Q3", 2));
718     barSeries.YAxis.Labels.Add(new IndexedYAxisLabel("Q4", 3));
719     barSeries.XAxis.Labels.Angle = -65;
720     elements.Add(chart);
721
722 }
723
724 private void AddCodabar(Group elements, float x, float y)
725 {
726     AddCaptionAndRectangle(elements, "Codabar Barcode", x, y, 225, 80);
727     BarCode barCode = new Codabar("A1234B", x, y + 15, 48);
728     barCode.X += (225 - barCode.GetSymbolWidth()) / 2;
729     barCode.Y += (65 - barCode.Height) / 2;
730     elements.Add(barCode);
731 }
732
733 private void AddCode128(Group elements, float x, float y)
734 {
735     AddCaptionAndRectangle(elements, "Code 128 Barcode", x, y, 225, 80);
736     BarCode barCode = new Code128("Code 128 Barcode.", x, y + 15, 48, 0.75f);
737     barCode.X += (225 - barCode.GetSymbolWidth()) / 2;
738     barCode.Y += (65 - barCode.Height) / 2;
739     elements.Add(barCode);
740 }
741
```



```
742
743     private void AddCode25(Group elements, float x, float y)
744     {
745         AddCaptionAndRectangle(elements, "Code 2 of 5 Barcode", x, y, 225, 80);
746         BarCode barCode = new Code25("1234567890", x, y + 15, 48);
747         barCode.X += (225 - barCode.GetSymbolWidth()) / 2;
748         barCode.Y += (65 - barCode.Height) / 2;
749         elements.Add(barCode);
750     }
751
752     private void AddCode39(Group elements, float x, float y)
753     {
754         AddCaptionAndRectangle(elements, "Code 3 of 9 Barcode", x, y, 225, 80);
755         BarCode barCode = new Code39("CODE 39", x, y + 15, 48);
756         barCode.X += (225 - barCode.GetSymbolWidth()) / 2;
757         barCode.Y += (65 - barCode.Height) / 2;
758         elements.Add(barCode);
759     }
760
761     private void AddInterleaved25(Group elements, float x, float y)
762     {
763         AddCaptionAndRectangle(elements, "Interleaved 2 of 5 Barcode", x, y, 225, 80);
764         BarCode barCode = new Interleaved25("1234567890", x, y + 15, 48);
765         barCode.X += (225 - barCode.GetSymbolWidth()) / 2;
766         barCode.Y += (65 - barCode.Height) / 2;
767         elements.Add(barCode);
768     }
769
770     private void AddIntelligentMailBarcode(Group elements, float x, float y)
771     {
772         AddCaptionAndRectangle(elements, "Intelligent Mail Barcode", x, y, 225, 80);
773         BarCode barcode = new IntelligentMailBarcode("0123456709498765432101234567891", x, y + 15, true);
774         barcode.Height = 35;
775         barcode.X += (225 - barcode.GetSymbolWidth()) / 2;
776         barcode.Y += (65 - barcode.Height) / 2;
777         elements.Add(barcode);
778     }
779
780     private void AddPostnet(Group elements, float x, float y)
781     {
782         AddCaptionAndRectangle(elements, "Postnet Barcode", x, y, 225, 80);
783         BarCode barCode = new Postnet("20815470412", x, y + 15);
784         barCode.X += (225 - barCode.GetSymbolWidth()) / 2;
785         barCode.Y += (65 - barCode.Height) / 2;
786         elements.Add(barCode);
787     }
788
789     private void AddPdf417(Group elements, float x, float y)
790     {
791         AddCaptionAndRectangle(elements, "PDF417 Barcode", x, y, 225, 100);
```

```
792     byte[] by = ASCIIEncoding.ASCII.GetBytes("PDF417 Barcode");
793     Pdf417 pdfc = new Pdf417(by, x, y + 15, 2, 2);
794     pdfc.X += (225 - pdfc.GetSymbolWidth()) / 2;
795     pdfc.Y += (85 - pdfc.GetSymbolHeight()) / 2;
796     elements.Add(pdfc);
797 }
798
799 private void AddPdf417Macro(Group elements, float x, float y)
800 {
801     AddCaptionAndRectangle(elements, "PDF417 Macro Barcode", x, y, 225, 100);
802     byte[] values5 = null;
803     try
804     {
805         FileStream fi = new FileStream(Server.MapPath(" ../Text/Textdata.txt"), FileMode.Open, FileAccess.Read);
806         long avail = fi.Length;
807         values5 = new byte[avail];
808         fi.Read(values5, 0, (int)avail);
809         fi.Close();
810     }
811     catch (Exception e)
812     {
813         Console.WriteLine("EXCEPTION " + e.Message);
814     }
815
816     // Create a MacroPdf417
817     MacroPdf417 macroPdf417 = new MacroPdf417(values5, x, y + 15, 10, 15, 0.9f);
818     macroPdf417.X += (225 - macroPdf417.GetSymbolWidth()) / 2;
819     macroPdf417.Y += (85 - macroPdf417.GetSymbolHeight()) / 2;
820
821     // Add macroPdf417 to page
822     elements.Add(macroPdf417);
823
824     // Retrieve the overflow barcode and place it at the appropriate location on the page
825     macroPdf417 = macroPdf417.GetOverflowMacroPdf417(x, y + 124);
826     macroPdf417.X += (225 - macroPdf417.GetSymbolWidth()) / 2;
827     macroPdf417.Y += (85 - macroPdf417.GetSymbolHeight()) / 2;
828     AddCaptionAndRectangle(elements, "PDF417 Overflow Macro Barcode", x, y += 108, 225, 100);
829     elements.Add(macroPdf417);
830 }
831
832 private void AddDataMatrixBarcode(Group elements, float x, float y)
833 {
834     AddCaptionAndRectangle(elements, "Data Matrix Barcode", x, y, 225, 100);
835     string text = "The photograph attached was taken by the crew on board the Columbia.";
836
837     // Create a data matrix bar code
838     DataMatrixBarcode dataMatrixBarcode = new DataMatrixBarcode(text, x, y+15, DataMatrixSymbolSize.R32xC32,
DataMatrixEncodingType.Auto);
839     //dataMatrixBarcode.XDimension = 2f;
840     dataMatrixBarcode.X += (225 - dataMatrixBarcode.GetSymbolWidth()) / 2;
```

```
841         dataMatrixBarcode.Y += (85 - dataMatrixBarcode.GetSymbolHeight()) / 2;
842
843         // Add data matrix barcode to the page
844         elements.Add(dataMatrixBarcode);
845
846         AddCaptionAndRectangle(elements, "Data Matrix Overflow Barcode", x, y += 108, 225, 100);
847
848         // Retrieve the overflow barcode and place it at the appropriate location on the page
849         dataMatrixBarcode = dataMatrixBarcode.GetOverFlowDataMatrixBarcode(x, y + 15);
850         //dataMatrixBarcode.XDimension = 1.75f;
851         dataMatrixBarcode.X += (225 - dataMatrixBarcode.GetSymbolWidth()) / 2;
852         dataMatrixBarcode.Y += (85 - dataMatrixBarcode.GetSymbolHeight()) / 2;
853         elements.Add(dataMatrixBarcode);
854     }
855     private void AddQrCode(Group elements, float x, float y)
856     {
857         AddCaptionAndRectangle(elements, "QR Code", x, y, 225, 100);
858         byte[] by = ASCIIEncoding.ASCII.GetBytes("QR Code Barcode");
859         QrCode qrCode = new QrCode(by, x, y + 15);
860         qrCode.XDimension = 2.5f;
861         qrCode.X += (225 - qrCode.GetSymbolWidth()) / 2;
862         qrCode.Y += (85 - qrCode.GetSymbolHeight()) / 2;
863         elements.Add(qrCode);
864     }
865
866     private void AddUPCEANPage(Document document, Outline parentOutline)
867     {
868         // Increments a page counter
869         pageCounter++;
870
871         // Adds to outlines to the page
872         parentOutline.ChildOutlines.Add("Page " + pageCounter + " - UPC/EAN/JAN Barcodes", new ZoomDestination(pageCounter,
PageZoom.FitPage));
873
874         // Create a page to add to the document
875         ceTe.DynamicPDF.Page page = new ceTe.DynamicPDF.Page(PageSize.Letter, PageOrientation.Landscape, 54);
876
877         // Uncomment the line below to add a layout grid to the page
878         //page.Elements.Add( new LayoutGrid() );
879
880         // Add the page elements to the page
881         AddEAN13(page.Elements, 0, 0);
882         AddEAN13Sup2(page.Elements, 240, 0);
883         AddEAN13Sup5(page.Elements, 480, 0);
884         AddUPCVersionA(page.Elements, 0, 135);
885         AddUPCVersionASup2(page.Elements, 240, 135);
886         AddUPCVersionASup5(page.Elements, 480, 135);
887         AddEAN8(page.Elements, 0, 270);
888         AddEAN8Sup2(page.Elements, 240, 270);
889         AddEAN8Sup5(page.Elements, 480, 270);
```

```
890     AddUPCVersionE(page.Elements, 0, 405);
891     AddUPCVersionESup2(page.Elements, 240, 405);
892     AddUPCVersionESup5(page.Elements, 480, 405);
893
894     // Add the page to the document
895     document.Pages.Add(page);
896 }
897
898 private void AddEAN13(Group elements, float x, float y)
899 {
900     AddCaptionAndRectangle(elements, "EAN/JAN 13 Barcode", x, y, 204, 99);
901     Barcode barCode = new Ean13("123456789012", x, y + 21);
902     barCode.X += (204 - barCode.GetSymbolWidth()) / 2;
903     elements.Add(barCode);
904 }
905
906 private void AddEAN13Sup2(Group elements, float x, float y)
907 {
908     AddCaptionAndRectangle(elements, "EAN/JAN 13 Barcode, 2 digit supplement", x, y, 204, 99);
909     Barcode barCode = new Ean13Sup2("12345678901212", x, y + 21);
910     barCode.X += (204 - barCode.GetSymbolWidth()) / 2;
911     elements.Add(barCode);
912 }
913
914 private void AddEAN13Sup5(Group elements, float x, float y)
915 {
916     AddCaptionAndRectangle(elements, "EAN/JAN 13 Barcode, 5 digit supplement", x, y, 204, 99);
917     Barcode barCode = new Ean13Sup5("12345678901212345", x, y + 21);
918     barCode.X += (204 - barCode.GetSymbolWidth()) / 2;
919     elements.Add(barCode);
920 }
921
922 private void AddUPCVersionA(Group elements, float x, float y)
923 {
924     AddCaptionAndRectangle(elements, "UPC Version A Barcode", x, y, 204, 99);
925     Barcode barCode = new UpcVersionA("12345678901", x, y + 21);
926     barCode.X += (204 - barCode.GetSymbolWidth()) / 2;
927     elements.Add(barCode);
928 }
929
930 private void AddUPCVersionASup2(Group elements, float x, float y)
931 {
932     AddCaptionAndRectangle(elements, "UPC Version A Barcode, 2 digit supplement", x, y, 204, 99);
933     Barcode barCode = new UpcVersionASup2("1234567890112", x, y + 21);
934     barCode.X += (204 - barCode.GetSymbolWidth()) / 2;
935     elements.Add(barCode);
936 }
937
938 private void AddUPCVersionASup5(Group elements, float x, float y)
939 {
```

```
940     AddCaptionAndRectangle(elements, "UPC Version A Barcode, 5 digit supplement", x, y, 204, 99);
941     BarCode barCode = new UpcVersionASup5("1234567890112345", x, y + 21);
942     barCode.X += (204 - barCode.GetSymbolWidth()) / 2;
943     elements.Add(barCode);
944 }
945
946 private void AddEAN8(Group elements, float x, float y)
947 {
948     AddCaptionAndRectangle(elements, "EAN/JAN 8 Barcode", x, y, 204, 99);
949     BarCode barCode = new Ean8("12345670", x, y + 21);
950     barCode.X += (204 - barCode.GetSymbolWidth()) / 2;
951     elements.Add(barCode);
952 }
953
954 private void AddEAN8Sup2(Group elements, float x, float y)
955 {
956     AddCaptionAndRectangle(elements, "EAN/JAN 8 Barcode, 2 digit supplement", x, y, 204, 99);
957     BarCode barCode = new Ean8Sup2("1234567012", x, y + 21);
958     barCode.X += (204 - barCode.GetSymbolWidth()) / 2;
959     elements.Add(barCode);
960 }
961
962 private void AddEAN8Sup5(Group elements, float x, float y)
963 {
964     AddCaptionAndRectangle(elements, "EAN/JAN 8 Barcode, 5 digit supplement", x, y, 204, 99);
965     BarCode barCode = new Ean8Sup5("1234567012345", x, y + 21);
966     barCode.X += (204 - barCode.GetSymbolWidth()) / 2;
967     elements.Add(barCode);
968 }
969
970 private void AddUPCVersionE(Group elements, float x, float y)
971 {
972     AddCaptionAndRectangle(elements, "UPC Version E Barcode", x, y, 204, 99);
973     BarCode barCode = new UpcVersionE("0123456", x, y + 21);
974     barCode.X += (204 - barCode.GetSymbolWidth()) / 2;
975     elements.Add(barCode);
976 }
977
978 private void AddUPCVersionESup2(Group elements, float x, float y)
979 {
980     AddCaptionAndRectangle(elements, "UPC Version E Barcode, 2 digit supplement", x, y, 204, 99);
981     BarCode barCode = new UpcVersionESup2("012345612", x, y + 21);
982     barCode.X += (204 - barCode.GetSymbolWidth()) / 2;
983     elements.Add(barCode);
984 }
985
986 private void AddUPCVersionESup5(Group elements, float x, float y)
987 {
988     AddCaptionAndRectangle(elements, "UPC Version E Barcode, 5 digit supplement", x, y, 204, 99);
989     BarCode barCode = new UpcVersionESup5("012345612345", x, y + 21);
```

```
990         barCode.X += (204 - barCode.GetSymbolWidth()) / 2;
991         elements.Add(barCode);
992     }
993
994     private void AddPageElementPage(Document document, Outline parentOutline)
995     {
996         // Increments a page counter
997         pageCounter++;
998
999         // Adds to outlines to the page
1000         Outline pageOutline = parentOutline.ChildOutlines.Add("Page " + pageCounter);
1001         pageOutline.ChildOutlines.Add("Zoom - Fit Page", new ZoomDestination(pageCounter, PageZoom.FitPage));
1002         pageOutline.ChildOutlines.Add("Zoom - Fit Width", new ZoomDestination(pageCounter, PageZoom.FitWidth));
1003
1004         // Create a page to add to the document
1005         ceTe.DynamicPDF.Page page = new ceTe.DynamicPDF.Page(PageSize.Letter, PageOrientation.Portrait, 54);
1006
1007         // Uncomment the line below to add a layout grid to the page
1008         //page.Elements.Add( new LayoutGrid() );
1009
1010         // Add the page elements to the page
1011         AddHeader(page.Elements);
1012         AddBookmark(page.Elements, 0, 41, pageOutline);
1013         AddCircle(page.Elements, 279, 41);
1014         AddHtmlArea(page.Elements, 0, 134);
1015         AddImage(page.Elements, 0, 227);
1016         AddLabel(page.Elements, 279, 227);
1017         AddLine(page.Elements, 0, 320);
1018         AddLink(page.Elements, 279, 320);
1019         AddPath(page.Elements, 0, 413);
1020         AddRectangle(page.Elements, 279, 413);
1021         AddTextArea(page.Elements, 0, 506);
1022         AddCustom(page.Elements, 0, 599);
1023         AddNote(page.Elements, 279, 599);
1024
1025         // Add the page to the document
1026         document.Pages.Add(page);
1027     }
1028
1029     private void AddHeader(Group pageElements)
1030     {
1031         // Adds a header to the pageElements
1032         string formattedHtml = "<html><head><title>Header</title><style>body{font: 10pt helvetica;text-align:center}</style></head><body><i>Dynamic</i><b>PDF</b> Generator for .NET has over 50 built-in page " +
1033             "elements. Custom page elements can be built by creating a class that inherits from " +
1034             "<b>ceTe.DynamicPDF.PageElement</b>.</body></html>";
1035         HtmlArea html = new HtmlArea(formattedHtml, 0, 0, 520, 450);
1036         //FormattedTextArea formattedTextArea = new FormattedTextArea(formattedHtml, 0, 0, 504, 450, FontFamily.Helvetica,
1037         12, false);
1038         //formattedTextArea.Style.Paragraph.Align = TextAlign.Center;
```

```
1038     pageElements.Add(html);
1039 }
1040
1041 private void AddBookmark(Group pageElements, float x, float y, Outline parentOutline)
1042 {
1043     // Adds a bookmark to the page
1044     AddCaptionAndRectangle(pageElements, "Bookmark Page Element", x, y);
1045     pageElements.Add(new Bookmark("Bookmarked Text", x + 5, y + 20, parentOutline));
1046     pageElements.Add(new Label("This text is bookmarked.", x + 5, y + 20, 215, 10, Font.TimesRoman, 10));
1047 }
1048
1049 private void AddCircle(Group pageElements, float x, float y)
1050 {
1051     // Adds a circle to the pageElements
1052     AddCaptionAndRectangle(pageElements, "Circle Page Element", x, y);
1053     pageElements.Add(new Circle(x + 112.5f, y + 50f, 107.5f, 30f, RgbColor.Red, RgbColor.Blue, 2, LineStyle.DashLarge));
1054 }
1055
1056
1057 private void AddHtmlArea(Group pageElements, float x, float y)
1058 {
1059     // Adds a html area to the pageElements
1060     Uri uri = new Uri(Server.MapPath("../Html/AllPageElementSample.html"));
1061     HtmlArea htmlArea = new HtmlArea(uri, x + 5, y + 20, 215, 70);
1062
1063     // Sets the indent property
1064     AddCaptionAndRectangle(pageElements, "HtmlArea Page Element", x, y);
1065     AddCaptionAndRectangle(pageElements, "HtmlArea Overflow Text", x + 279, y);
1066     pageElements.Add(htmlArea);
1067
1068     // Set the html area object equal to the rest of the html area that did not fit
1069     // if all the html area did fit, GetOverflowHtmlArea will return null
1070     HtmlArea overflowHtmlArea = htmlArea.GetOverflowHtmlArea(x + 284, y + 20, 70);
1071     pageElements.Add(overflowHtmlArea);
1072
1073 }
1074
1075 private void AddImage(Group pageElements, float x, float y)
1076 {
1077     // Adds an image to the pageElements
1078     AddCaptionAndRectangle(pageElements, "Image Page Element", x, y);
1079     Image image = new Image(Server.MapPath("../Images/DPDFLogo.png"), x + 112.5f, y + 50f, 0.24f);
1080
1081     // Image is sized and centered in the rectangle
1082     image.SetBounds(215, 60);
1083     image.VAlign = VAlign.Center;
1084     image.Align = Align.Center;
1085     pageElements.Add(image);
1086 }
1087
```

```
1088     private void AddLabel(Group pageElements, float x, float y)
1089     {
1090         // Adds a label to the pageElements
1091         AddCaptionAndRectangle(pageElements, "Label & PageNumberingLabel Page Elements", x, y);
1092         string labelText = "Labels can be rotated";
1093         string numberText = "PageNumberingLabels contain page numbering: %%CP%% of %%TP%% pages.";
1094         Label label = new Label(labelText, x + 5, y + 12, 220, 80, Font.TimesRoman, 12, TextAlign.Center);
1095         label.Angle = 8;
1096         PageNumberingLabel pageNumLabel = new PageNumberingLabel(numberText, x + 5, y + 55, 220, 80, Font.TimesRoman, 12,
TextAlign.Center);
1097         pageElements.Add(pageNumLabel);
1098         pageElements.Add(label);
1099     }
1100
1101     private void AddLine(Group pageElements, float x, float y)
1102     {
1103         // Adds a line to the pageElements
1104         AddCaptionAndRectangle(pageElements, "Line Page Element", x, y);
1105         pageElements.Add(new Line(x + 5, y + 20, x + 220, y + 80, 3, RgbColor.Green));
1106         pageElements.Add(new Line(x + 220, y + 20, x + 5, y + 80, 3, RgbColor.Green));
1107     }
1108
1109     private void AddLink(Group pageElements, float x, float y)
1110     {
1111         // Adds a link to the pageElements
1112         Font font = Font.TimesRoman;
1113         string text = "This is a link to DynamicPDF.com.";
1114         AddCaptionAndRectangle(pageElements, "Link Page Element", x, y);
1115         Label label = new Label(text, x + 5, y + 20, 215, 80, font, 12, RgbColor.Blue);
1116         label.Underline = true;
1117         Link link = new Link(x + 5, y + 20, font.GetTextWidth(text, 12),
1118             12 - font.GetDescender(12), new UrlAction("http://www.dynamicpdf.com"));
1119         pageElements.Add(label);
1120         pageElements.Add(link);
1121     }
1122
1123     private void AddPath(Group pageElements, float x, float y)
1124     {
1125         // Adds a path to the pageElements
1126         ceTe.DynamicPDF.PageElements.Path path = new ceTe.DynamicPDF.PageElements.Path(x + 5, y + 20, RgbColor.Blue,
RgbColor.Red, 2, LineStyle.Solid, true);
1127         path.SubPaths.Add(new LineSubPath(x + 215, y + 40));
1128         path.SubPaths.Add(new CurveToSubPath(x + 108, y + 80, x + 160, y + 80));
1129         path.SubPaths.Add(new CurveSubPath(x + 5, y + 40, x + 65, y + 80, x + 5, y + 60));
1130         AddCaptionAndRectangle(pageElements, "Path Page Element", x, y);
1131         pageElements.Add(path);
1132     }
1133
1134     private void AddRectangle(Group pageElements, float x, float y)
1135     {
```



```
1136         // Adds a rectangle to the pageElements
1137         AddCaptionAndRectangle(pageElements, "Rectangle Page Element", x, y);
1138         pageElements.Add(new Rectangle(x + 5, y + 20, 215, 60, RgbColor.Green, RgbColor.Blue, 2));
1139     }
1140
1141     private void AddTextArea(Group pageElements, float x, float y)
1142     {
1143         // Adds a text area to the pageElements
1144         string text = "The TextArea page element is great for efficient processign of text that does " +
1145             "not require advanced formatting. It offers basic control over leading and paragraph spacing and " +
1146             "include text continuation functionality. If all the text does not fit within the bounds of " +
1147             "the TextArea the HasOverflowText method will return a true value. The GetOverflowTextArea " +
1148             "method can then be called to create a new TextArea with the remaining text. This " +
1149             "functionality makes it simple to span large amounts of text over multiple TextAreas and pages. Text can also be
rotated.";
1150         AddCaptionAndRectangle(pageElements, "TextArea Page Element", x, y);
1151         AddCaptionAndRectangle(pageElements, "TextArea Overflow Text", x + 279, y);
1152         TextArea textArea = new TextArea(text, x + 5, y + 20, 215, 60, Font.TimesRoman, 10);
1153
1154         // Create an overflow text area for the overflow text
1155         TextArea overflowTextArea = textArea.GetOverflowTextArea(x + 284, y + 20);
1156         pageElements.Add(textArea);
1157         pageElements.Add(overflowTextArea);
1158     }
1159
1160     private void AddCustom(Group pageElements, float x, float y)
1161     {
1162         // Adds a custom page element to the page
1163         AddCaptionAndRectangle(pageElements, "Custom Page Element", x, y);
1164         pageElements.Add(new StopSign(80, y + 20, 60, 60));
1165     }
1166
1167     private void AddNote(Group pageElements, float x, float y)
1168     {
1169         // Add a note element to the page
1170         AddCaptionAndRectangle(pageElements, "Note Page Element", x, y);
1171         pageElements.Add(new Note("This is a note", x + 100, y + 35, 50, 50, NoteType.Note));
1172     }
1173
1174     private void AddTableAndOther(Document document, Outline parentOutline)
1175     {
1176         // Increments a page counter
1177         pageCounter++;
1178
1179         // Adds to outlines to the page
1180         parentOutline.ChildOutlines.Add("Page " + pageCounter + " - Table and Additional Page Elements", new
ZoomDestination(pageCounter, PageZoom.FitPage));
1181
1182         // Create a page to add to the document
1183         ceTe.DynamicPDF.Page page = new ceTe.DynamicPDF.Page(PageSize.Letter, PageOrientation.Portrait, 54);
```

```
1184
1185     // Uncomment the line below to add a layout grid to the page
1186     //page.Elements.Add( new LayoutGrid() );
1187
1188     // Add the Table and other elements to the page
1189     AddTable(page.Elements, 0, 0);
1190     AddListElements(page, 0, 225);
1191     AddAdditional(page.Elements, 0, 500);
1192
1193     // Add the page to the document
1194     document.Pages.Add(page);
1195 }
1196
1197 private void AddTable(Group pageElements, float x, float y)
1198 {
1199     Label captionLabel1 = new Label("Table Page Element", x, y, 225, 10, Font.HelveticaBold, 10);
1200     pageElements.Add(captionLabel1);
1201     Label captionLabel2 = new Label("Table Row Overflow", x + 279, y, 225, 10, Font.HelveticaBold, 10);
1202     pageElements.Add(captionLabel2);
1203     Table2 table = new Table2(x, y + 15, 225, 190, Font.TimesRoman, 10);
1204     table.RepeatColumnHeaderCount = 1;
1205
1206     // Add columns to the table
1207     table.Columns.Add(95);
1208     table.Columns.Add(65);
1209     table.Columns.Add(65);
1210
1211     // Add the Header Row to the table
1212     Row2 row1 = table.Rows.Add(40, Font.HelveticaBold, 12, RgbColor.Black, RgbColor.LightBlue);
1213     row1.CellDefault.Align = TextAlign.Center;
1214     row1.CellDefault.VAlign = VAlign.Center;
1215     row1.Cells.Add("Header 1");
1216     row1.Cells.Add("Header 2");
1217     row1.Cells.Add("Header 3");
1218     row1.Cells.Add("Header 4");
1219     row1.Cells.Add("Header 5");
1220     row1.Cells.Add("Header 6");
1221     row1.Cells.Add("Header 7");
1222     row1.Cells.Add("Header 8");
1223     row1.Cells.Add("Header 9");
1224     row1.Cells.Add("Header 10");
1225     row1.Cells.Add("Header 11");
1226     row1.Cells.Add("Header 12");
1227     row1.Cells.Add("Header 13");
1228     row1.Cells.Add("Header 14");
1229     row1.Cells.Add("Header 15");
1230     row1.Cells.Add("Header 16");
1231     int cell = 0;
1232
1233     // Add the rest of the rows to the table
```

```
1234         AddRow(table, ref cell, 1);
1235         AddRow(table, ref cell, 2);
1236         AddRow(table, ref cell, 3);
1237         AddRow(table, ref cell, 4);
1238         AddRow(table, ref cell, 5);
1239         AddRow(table, ref cell, 6);
1240         AddRow(table, ref cell, 7);
1241         AddRow(table, ref cell, 8);
1242         AddRow(table, ref cell, 9);
1243         AddRow(table, ref cell, 10);
1244         AddRow(table, ref cell, 11);
1245         AddRow(table, ref cell, 12);
1246         AddRow(table, ref cell, 13);
1247         AddRow(table, ref cell, 14);
1248         AddRow(table, ref cell, 15);
1249         AddRow(table, ref cell, 16);
1250
1251         // Create an overflow table for the overflow columns
1252         Table2 overflowTable = table.GetOverflowRows(x + 279, y + 15);
1253         pageElements.Add(table);
1254         pageElements.Add(overflowTable);
1255     }
1256
1257     private void AddRow(Table2 table, ref int cellNum, int rowNum)
1258     {
1259         Row2 row = table.Rows.Add();
1260         Cell2 cell = row.Cells.Add("RowHeader " + rowNum, Font.HelveticaBold, 12,
1261             ConsoleColor.Black, ConsoleColor.LightPink, 1);
1262         cell.Align = TextAlign.Center;
1263         cell.VAlign = VAlign.Center;
1264         row.Cells.Add("Item " + cellNum++);
1265         row.Cells.Add("Item " + cellNum++);
1266     }
1267
1268     private void AddAdditional(Group pageElements, float x, float y)
1269     {
1270         // Adds text explaining the additional page elements
1271         string text = "An <b><u>AreaGroup</u></b> represents a group of page elements with an area.<p/>" +
1272             "A <b><u>BackgroundImage</u></b> page element is used to add an image to the background of a page. <p/>" +
1273             "The <b><u>Group</u></b> page element is used for grouping page elements together.<p/>" +
1274             "The <b><u>LayoutGrid</u></b> draws a coordinate grid to the page.<p/>" +
1275             "A <b><u>TransformationGroup</u></b> can rotate and scale a group of page elements.<p/>" +
1276             "A <b><u>TransparencyGroup</u></b> can be used to add transparency to page elements.<p/>" +
1277             "An <b><u>Anchor Group </u></b> can have Page Elements within it. It can be anchored to the edge or margins of
the page.<p/>";
1278         FormattedTextArea formattedTextArea = new FormattedTextArea(text, x + 5, y + 20, 494, 180, FontFamily.Times, 12,
false);
1279         formattedTextArea.Style.Paragraph.SpacingAfter = 5;
1280         AddCaptionAndRectangle(pageElements, "Additional Page Elements", x, y, 504, 180);
1281         pageElements.Add(formattedTextArea);
```

```
1282     }
1283
1284     private void AddCaptionAndRectangle(Group pageElements, string caption, float x, float y, float width, float height)
1285     {
1286         // Adds a rectangle and caption to the pageElements
1287         Rectangle rectangle = new Rectangle(x, y + 15, width, height - 15);
1288         Label captionLabel = new Label(caption, x, y, 225, 10, Font.HelveticaBold, 10);
1289         pageElements.Add(rectangle);
1290         pageElements.Add(captionLabel);
1291     }
1292
1293     private void AddCaptionAndRectangle(Group pageElements, string caption, float x, float y)
1294     {
1295         AddCaptionAndRectangle(pageElements, caption, x, y, 225, 85);
1296     }
1297
1298     private void AddFormElementPage(Document document, Outline parentOutline)
1299     {
1300         pageCounter++;
1301         parentOutline.ChildOutlines.Add("Page " + pageCounter + " - Form Page Elements", new ZoomDestination(pageCounter,
PageZoom.FitPage));
1302
1303         // Create a page to add to the document
1304         ceTe.DynamicPDF.Page page = new ceTe.DynamicPDF.Page(PageSize.Letter, PageOrientation.Portrait, 54);
1305         AddTextField(page.Elements, 0, 0);
1306         AddComboField(page.Elements, 0, 105);
1307         AddListField(page.Elements, 0, 210);
1308         AddRadioField(page.Elements, 0, 315);
1309         AddCheckBoxField(page.Elements, 0, 420);
1310         AddButtonField(page.Elements, 0, 525);
1311
1312         // Add the page to the document
1313         document.Pages.Add(page);
1314     }
1315
1316     private void AddListElements(Page page, float x, float y)
1317     {
1318         OrderedList orderedList = new OrderedList(x + 5, y + 20, 400, 90, Font.Helvetica, 10);
1319         orderedList.Items.Add("Fruits");
1320         orderedList.Items.Add("Vegetables");
1321
1322         OrderedSubList orderedSubList = orderedList.Items[0].SubLists.AddOrderedSubList(NumberingStyle.RomanUpperCase);
1323         orderedSubList.Items.Add("Citrus");
1324         orderedSubList.Items.Add("Non-Citrus");
1325
1326         AddCaptionAndRectangle(page.Elements, "Ordered List Page Element", x, y, 225, 110);
1327         OrderedSubList orderedSubList2 = orderedList.Items[1].SubLists.AddOrderedSubList(NumberingStyle.RomanUpperCase);
1328         orderedSubList2.Items.Add("Potato");
1329         orderedSubList2.Items.Add("Beans");
1330     }
```

```
1331         OrderedSubList subOrderedSubList =
1332             subOrderedSubList.Items.Add("Lime");
1333             subOrderedSubList.Items.Add("Orange");
1334
1335         OrderedSubList subOrderedSubList2 =
orderedSubList.Items[1].SubLists.AddOrderedSubList(NumberingStyle.RomanLowerCase);
1336             subOrderedSubList2.Items.Add("Mango");
1337             subOrderedSubList2.Items.Add("Banana");
1338
1339         OrderedSubList subOrderedSubList3 =
orderedSubList2.Items[0].SubLists.AddOrderedSubList(NumberingStyle.RomanLowerCase);
1340             subOrderedSubList3.Items.Add("Sweet Potato");
1341
1342         OrderedSubList subOrderedSubList4 =
orderedSubList2.Items[1].SubLists.AddOrderedSubList(NumberingStyle.RomanLowerCase);
1343             subOrderedSubList4.Items.Add("String Bean");
1344             subOrderedSubList4.Items.Add("Lima Bean");
1345             subOrderedSubList4.Items.Add("Kidney Bean");
1346
1347         page.Elements.Add(orderedList);
1348         x += 279;
1349
1350         orderedList = orderedList.GetOverflowList(x + 5, y + 20);
1351         AddCaptionAndRectangle(page.Elements, "Ordered List Page Element Overflow", x, y, 225, 110);
1352         page.Elements.Add(orderedList);
1353
1354         x = 0;
1355         y += 138;
1356
1357         // Create an unordered list
1358         UnorderedList unorderedList = new UnorderedList(x + 5, y + 20, 400, 90, Font.Helvetica, 10);
1359         unorderedList.Items.Add("Fruits");
1360         unorderedList.Items.Add("Vegetables");
1361
1362         UnorderedSubList unorderedSubList = unorderedList.Items[0].SubLists.AddUnorderedSubList();
1363         unorderedSubList.Items.Add(" Citrus");
1364         unorderedSubList.Items.Add(" Non-Citrus");
1365
1366         AddCaptionAndRectangle(page.Elements, "Unordered List Page Element", x, y, 225, 110);
1367
1368         UnorderedSubList unorderedSubList2 = unorderedList.Items[1].SubLists.AddUnorderedSubList();
1369         unorderedSubList2.Items.Add("Potato");
1370         unorderedSubList2.Items.Add("Beans");
1371
1372         UnorderedSubList subUnorderedSubList = unorderedSubList.Items[0].SubLists.AddUnorderedSubList();
1373         subUnorderedSubList.Items.Add("Lime");
1374         subUnorderedSubList.Items.Add("Orange");
1375
1376         UnorderedSubList subUnorderedSubList2 = unorderedSubList.Items[1].SubLists.AddUnorderedSubList();
```

```
1377         subUnorderedSubList2.Items.Add("Mango");
1378         subUnorderedSubList2.Items.Add("Banana");
1379
1380         UnorderedSubList subUnorderedSubList3 = unorderedSubList2.Items[0].SubLists.AddUnorderedSubList();
1381         subUnorderedSubList3.Items.Add("Sweet Potato");
1382
1383         UnorderedSubList subUnorderedSubList4 = unorderedSubList2.Items[1].SubLists.AddUnorderedSubList();
1384         subUnorderedSubList4.Items.Add("String Bean");
1385         subUnorderedSubList4.Items.Add("Lima Bean");
1386         subUnorderedSubList4.Items.Add("Kidney Bean");
1387
1388         x += 279;
1389         page.Elements.Add(unorderedList);
1390         unorderedList = unorderedList.GetOverflowList(x + 5, y + 20);
1391         AddCaptionAndRectangle(page.Elements, "Unordered List Page Element Overflow", x, y, 225, 110);
1392         page.Elements.Add(unorderedList);
1393     }
1394
1395     private void AddTextField(Group pageElements, float x, float y)
1396     {
1397         TextField txt = new TextField("txtfname", x + 20, y + 40, 120, 20);
1398         txt.DefaultValue = "This is a Scrollable TextField";
1399         txt.BorderColor = RgbColor.Black;
1400         txt.BackgroundColor = RgbColor.AliceBlue;
1401         txt.ToolTip = "Scrollable";
1402         pageElements.Add(txt);
1403
1404         TextField txt1 = new TextField("txtflname", x + 175, y + 40, 120, 20);
1405         txt1.DefaultValue = "TextField";
1406         txt1.Password = true;
1407         txt1.MaxLength = 9;
1408         txt1.BorderColor = RgbColor.Black;
1409         txt1.BackgroundColor = RgbColor.AliceBlue;
1410         txt1.ToolTip = "Password with Maximum Length";
1411         pageElements.Add(txt1);
1412
1413         TextField txt2 = new TextField("txtf2name", x + 330, y + 30, 150, 40);
1414         txt2.DefaultValue = "This is a TextField which goes to the next line if the text exceeds width";
1415         txt2.MultiLine = true;
1416         txt2.BorderColor = RgbColor.Black;
1417         txt2.BackgroundColor = RgbColor.AliceBlue;
1418         txt2.ToolTip = "Multiline";
1419         pageElements.Add(txt2);
1420
1421         AddCaptionAndRectangle(pageElements, "TextField Form Page Element", x, y, 504, 85);
1422     }
1423
1424     private void AddComboField(Group pageElements, float x, float y)
1425     {
1426         ComboBox cb = new ComboBox("cmbname", x + 51, y + 40, 150, 20);
```

```
1427     cb.BorderColor = RgbColor.Black;
1428     cb.BackgroundColor = RgbColor.AliceBlue;
1429     cb.Font = Font.Helvetica;
1430     cb.FontSize = 12;
1431     cb.Items.Add("Item 1");
1432     cb.Items.Add("Item 2");
1433     cb.Items.Add("Item 3");
1434     cb.Items.Add("Item 4");
1435     cb.Items.Add("Editable");
1436     cb.Items["Editable"].Selected = true;
1437     cb.Editable = true;
1438     cb.ToolTip = "Editable Combo Box";
1439     pageElements.Add(cb);
1440
1441     ComboBox cb1 = new ComboBox("cmb1name", x + 303, y + 40, 150, 20);
1442     cb1.BorderColor = RgbColor.Black;
1443     cb1.BackgroundColor = RgbColor.AliceBlue;
1444     cb1.Font = Font.Helvetica;
1445     cb1.FontSize = 12;
1446     cb1.Items.Add("Item 1");
1447     cb1.Items.Add("Item 2");
1448     cb1.Items.Add("Item 3");
1449     cb1.Items.Add("Item 4");
1450     cb1.Items.Add("Non-Editable");
1451     cb1.Items["Non-Editable"].Selected = true;
1452     cb1.Editable = false;
1453     cb1.ToolTip = "Non-Editable Combo Box";
1454     pageElements.Add(cb1);
1455
1456     AddCaptionAndRectangle(pageElements, "ComboBox Form Page Element", x, y, 504, 85);
1457 }
1458
1459 private void AddListField(Group pageElements, float x, float y)
1460 {
1461     ListBox lb = new ListBox("lbname", x + 51, y + 30, 150, 40);
1462     lb.BorderColor = RgbColor.Black;
1463     lb.BackgroundColor = RgbColor.AliceBlue;
1464     lb.Font = Font.Helvetica;
1465     lb.FontSize = 12;
1466     lb.Items.Add("Item 1");
1467     lb.Items.Add("Item 2");
1468     lb.Items.Add("Item 3");
1469     lb.Items.Add("Item 4");
1470     lb.Items.Add("Non-MultiSelect");
1471     lb.Items["Non-MultiSelect"].Selected = true;
1472     lb.Multiselect = false;
1473     lb.ToolTip = "Simple List Box";
1474     pageElements.Add(lb);
1475
1476     ListBox lbl = new ListBox("lblname", x + 303, y + 30, 150, 40);
```

```
1477     lbl.BorderColor = RgbColor.Black;
1478     lbl.BackgroundColor = RgbColor.AliceBlue;
1479     lbl.Font = Font.Helvetica;
1480     lbl.FontSize = 12;
1481     lbl.Items.Add("Item 1");
1482     lbl.Items.Add("Item 2");
1483     lbl.Items.Add("Item 3");
1484     lbl.Items.Add("Item 4");
1485     lbl.Items["Item 1"].Selected = true;
1486     lbl.Items.Add("MultiSelect");
1487     lbl.Items["MultiSelect"].Selected = true;
1488     lbl.Multiselect = true;
1489     lbl.ToolTip = "MultiSelect List Box";
1490     pageElements.Add(lbl);
1491
1492     AddCaptionAndRectangle(pageElements, "ListBox Form Page Element", x, y, 504, 85);
1493 }
1494
1495 private void AddRadioField(Group pageElements, float x, float y)
1496 {
1497     RadioButton rb = new RadioButton("rbname", x + 40, y + 42, 12, 12);
1498     rb.DefaultChecked = true;
1499     rb.BackgroundColor = RgbColor.AliceBlue;
1500     rb.TextColor = RgbColor.Red;
1501     rb.ExportValue = "Radio1";
1502     Label lb = new Label("Monthly", x + 75, y + 43, 60, 15);
1503     pageElements.Add(rb);
1504     pageElements.Add(lb);
1505
1506     RadioButton rb1 = new RadioButton("rbname", x + 195, y + 42, 12, 12);
1507     rb1.BackgroundColor = RgbColor.AliceBlue;
1508     rb1.TextColor = RgbColor.Green;
1509     rb1.ExportValue = "Radio2";
1510     Label lb1 = new Label("Quarterly", x + 235, y + 43, 60, 15);
1511     pageElements.Add(rb1);
1512     pageElements.Add(lb1);
1513
1514     RadioButton rb2 = new RadioButton("rbname", x + 350, y + 42, 12, 12);
1515     rb2.BackgroundColor = RgbColor.AliceBlue;
1516     rb2.TextColor = RgbColor.Blue;
1517     rb2.ExportValue = "Radio3";
1518     Label lb2 = new Label("Yearly", x + 385, y + 43, 60, 15);
1519     pageElements.Add(rb2);
1520     pageElements.Add(lb2);
1521
1522     AddCaptionAndRectangle(pageElements, "RadioButton Form Page Element", x, y, 504, 85);
1523 }
1524
1525 private void AddCheckBoxField(Group pageElements, float x, float y)
1526 {
```



```
1527     CheckBox cb = new CheckBox("cbxname", x + 40, y + 42, 15, 15);
1528     cb.Symbol = ceTe.DynamicPDF.PageElements.Forms.Symbol.Check;
1529     cb.BackgroundColor = RgbColor.AliceBlue;
1530     Label lb = new Label("Onion", x + 75, y + 43, 60, 15);
1531     pageElements.Add(cb);
1532     pageElements.Add(lb);
1533
1534     CheckBox cb1 = new CheckBox("cbxsecondname", x + 195, y + 42, 15, 15);
1535     cb1.Symbol = ceTe.DynamicPDF.PageElements.Forms.Symbol.Cross;
1536     cb1.DefaultChecked = true;
1537     cb1.BackgroundColor = RgbColor.AliceBlue;
1538     Label lbl = new Label("Capsicum", x + 235, y + 43, 60, 15);
1539     pageElements.Add(cb1);
1540     pageElements.Add(lbl);
1541
1542     CheckBox cb2 = new CheckBox("cbxthirdname", x + 350, y + 42, 15, 15);
1543     cb2.Symbol = ceTe.DynamicPDF.PageElements.Forms.Symbol.Star;
1544     cb2.BackgroundColor = RgbColor.AliceBlue;
1545     Label lb2 = new Label("Jalapenos", x + 385, y + 43, 80, 15);
1546     pageElements.Add(cb2);
1547     pageElements.Add(lb2);
1548
1549     AddCaptionAndRectangle(pageElements, "CheckBox Form Page Element", x, y, 504, 85);
1550 }
1551
1552 private void AddButtonField(Group pageElements, float x, float y)
1553 {
1554     Button bt = new Button("btonename", x + 40, y + 40, 100, 25);
1555     bt.Behavior = Behavior.Push;
1556     bt.BorderStyle = BorderStyle.Beveled;
1557     SubmitAction sb = new SubmitAction(Request.Path);
1558     bt.ToolTip = "Submit Button";
1559     bt.Label = "Submit";
1560     bt.Action = sb;
1561     pageElements.Add(bt);
1562
1563     Button bt1 = new Button("bttwoname", x + 195, y + 40, 100, 25);
1564     bt1.Behavior = Behavior.Push;
1565     bt1.BorderStyle = BorderStyle.Beveled;
1566     ResetAction rb = new ResetAction();
1567     bt1.Action = rb;
1568     bt1.ToolTip = "Reset Button";
1569     bt1.Label = "Reset";
1570     pageElements.Add(bt1);
1571
1572     Button bt2 = new Button("btthreename", x + 350, y + 40, 100, 25);
1573     bt2.Behavior = Behavior.Push;
1574     bt2.BorderStyle = BorderStyle.Beveled;
1575     bt2.Action = new ZoomDestination(pageCounter, PageZoom.FitWidth);
1576     bt2.Behavior = Behavior.CreatePush("Click", "Zoom");
```

```
1577         bt2.ToolTip = "Zoom to fit page width";
1578         bt2.Label = "Zoom";
1579         pageElements.Add(bt2);
1580
1581         Button bt3 = new Button("btfourname", x + 101, y + 90, 100, 25);
1582         bt3.Behavior = Behavior.Push;
1583         bt3.BorderStyle = BorderStyle.Beveled;
1584         bt3.Action = new XYDestination(pageCounter, 100, 100);
1585         bt3.Behavior = Behavior.CreatePush("Click", "Go To XY Destination");
1586         bt3.ToolTip = "Go to XY position";
1587         bt3.Label = "XY";
1588         pageElements.Add(bt3);
1589
1590         Button bt4 = new Button("btfivename", x + 302, y + 90, 100, 25);
1591         bt4.Behavior = Behavior.Push;
1592         bt4.BorderStyle = BorderStyle.Beveled;
1593         bt4.Action = new JavaScriptAction("app.alert('Hello');");
1594         bt4.ToolTip = "Javascript Button";
1595         bt4.Label = "OK";
1596         pageElements.Add(bt4);
1597
1598         AddCaptionAndRectangle(pageElements, "Button Form Page Element", x, y, 504, 140);
1599     }
1600
1601     private void AddWatermark(Document document, Outline parentOutline)
1602     {
1603         // Increments a page counter
1604         pageCounter++;
1605
1606         // Adds to outlines to the page
1607         parentOutline.ChildOutlines.Add("Page " + pageCounter + " - Watermark text", new ZoomDestination(pageCounter,
PageZoom.FitPage));
1608
1609         // Create a page to add to the document
1610         ceTe.DynamicPDF.Page page = new ceTe.DynamicPDF.Page(PageSize.Letter, PageOrientation.Portrait, 54);
1611
1612         // Create Text watermark
1613         TextWatermark watermark = new TextWatermark("Draft");
1614         watermark.TextColor = null;
1615         watermark.TextOutlineColor = RgbColor.Black;
1616
1617         string caption = "Text and Image Watermark Page Elements";
1618         Label captionLabel = new Label(caption, 0, 235, 225, 10, Font.HelveticaBold, 10);
1619         page.Elements.Add(captionLabel);
1620
1621         // Create Label and add discription text
1622         string text = "Text or Image watermarks can be placed on any page";
1623         Label label = new Label(text, 5, 255, 500, 100);
1624         label.FontSize = 40;
1625         page.Elements.Add(label);
```

```
1626
1627     // Create a Rectangle
1628     Rectangle rect = new Rectangle(0,250,500,100);
1629     page.Elements.Add(rect);
1630
1631     // Add Text watermark to page
1632     page.Elements.Add(watermark);
1633
1634     // Set Pdf version to 1.6
1635     document.PdfVersion = PdfVersion.v1_6;
1636
1637     // Add page to document
1638     document.Pages.Add(page);
1639
1640 }
1641
1642 #endregion
1643
1644 #region Web Form Designer generated code
1645 override protected void OnInit(EventArgs e)
1646 {
1647     //
1648     // CODEGEN: This call is required by the ASP.NET Web Form Designer.
1649     //
1650     InitializeComponent();
1651     base.OnInit(e);
1652 }
1653
1654 /// <summary>
1655 /// Required method for Designer support - do not modify
1656 /// the contents of this method with the code editor.
1657 /// </summary>
1658 private void InitializeComponent()
1659 {
1660 }
1661 #endregion
1662 }
1663 }
```