

## AllPageElements.java

---

```
1 import com.cete.dynamicpdf.*;
2 import com.cete.dynamicpdf.io.PageWriter;
3 import com.cete.dynamicpdf.pageelements.*;
4 import com.cete.dynamicpdf.pageelements.barcoding.*;
5 import com.cete.dynamicpdf.pageelements.charting.*;
6 import com.cete.dynamicpdf.pageelements.charting.axes.*;
7 import com.cete.dynamicpdf.pageelements.charting.series.*;
8 import com.cete.dynamicpdf.pageelements.forms.*;
9 import java.io.FileInputStream;
10 import java.io.FileNotFoundException;
11 import java.io.IOException;
12 import javax.servlet.ServletConfig;
13 import javax.servlet.ServletException;
14 import javax.servlet.ServletOutputStream;
15 import javax.servlet.http.HttpServlet;
16 import javax.servlet.http.HttpServletRequest;
17 import javax.servlet.http.HttpServletResponse;
18
19 /**
20  * A Java Servlet describing the usage of all the Page Elements.
21  */
22 public class AllPageElements extends HttpServlet {
23
24     private int pageCounter;
25
26     ServletOutputStream sOut;
27
28     public void init(ServletConfig config) throws ServletException {
29         super.init(config);
30
31     }
32
33     /**
34      * Custom page element.
35      */
36     public class StopSign extends PageElement {
37
38         private float x = 0.0f;
39         private float y = 0.0f;
40         private float w = 0.0f;
41         private float h = 0.0f;
42
43         /**
44          * Initializes a new instance of the <code>MyPageElement</code> class.
45          * @param x X coordinate of the page element.
46          * @param y Y coordinate of the page element.
47          * @param width Width of the page element.
48          * @param height Height of the page element.
49          */
```

```
50     public StopSign(float x, float y, float width, float height) {
51         this.x = x;
52         this.y = y;
53         this.w = width;
54         this.h = height;
55     }
56
57     /**
58     * Draws a MyPageElement page element to the page.
59     * @param writer <code>PageWriter</code> object to receive the
60     *           PDF output.
61     * @see PageWriter.
62     */
63     public void draw(PageWriter writer) {
64
65         float v = h / (float)3.414;
66
67         // End text mode
68         writer.write_ET();
69         writer.write_m_(x + v, y);
70         writer.write_RG(RgbColor.getWhite());
71         writer.write_rg_(RgbColor.getRed());
72
73         // Draws the axis lines to the page
74         writer.write_l_(x + h - v, y);
75         writer.write_l_(x + h, y + v);
76         writer.write_l_(x + h, y + h - v);
77         writer.write_l_(x + h - v, y + h);
78         writer.write_l_(x + v, y + h);
79         writer.write_l_(x, y + h - v);
80         writer.write_l_(x, y + v);
81         writer.write_b_();
82         char[] text = new char[] { 'S', 'T', 'O', 'P' };
83
84         // Begin text mode
85         writer.write_BT();
86         writer.setFont(Font.getHelvetica(), 16);
87         writer.write_rg_(RgbColor.getWhite());
88         writer.write_Tm((float)88.5, y + (float)(w / 1.65));
89
90         // Write the text
91         writer.write_Tj_(text, false);
92
93         // End text mode
94         writer.write_ET();
95     }
96 }
97
98
```

```
99     public void doGet(HttpServletRequest req, HttpServletResponse res)
100     throws IOException, ServletException {
101         sOut = res.getOutputStream();
102         pageCounter = 0;
103         // Create a document and set it's properties.
104         Document document = new Document();
105         document.setCreator("AllPageElements.java");
106         document.setAuthor("ceTe Software");
107         document.setTitle("All Page Elements");
108
109         // Add the top level outline to the document
110         Outline objParentOutline = document.getOutlines()
111         .add("Top Level Outline",
112             new XYDestination(1, 0, 0));
113         objParentOutline.setColor(new WebColor("0000FF"));
114         objParentOutline.setStyle(TextStyle.BOLD);
115
116         // Add pages to the document
117         addPageElementPage(document, objParentOutline);
118         addTableAndOther(document, objParentOutline );
119         addChartsPage(document, objParentOutline);
120         addFormElementPage( document, objParentOutline );
121         addBarCodePage(document, objParentOutline);
122         addUPCEANPage(document, objParentOutline);
123
124         // Outputs the document to the current web page
125         document.drawToWeb(req, res, sOut, "AllPageElements.pdf");
126
127         sOut.close();
128     }
129
130     private void addFormElementPage( Document document, Outline parentOutline ) {
131         pageCounter++;
132         parentOutline.getChildOutlines().add( "Page " + pageCounter + " - Form Page Elements", new ZoomDestination( pageCounter,
PageZoom.FIT_PAGE ) );
133
134         // Create a page to add to the document
135         com.cete.dynamicpdf.Page page = new com.cete.dynamicpdf.Page( 612, 713, 54 );
136         addTextField( page.getElements(), 0, 0 );
137         addComboField( page.getElements(), 0, 93 );
138         addListField( page.getElements(), 0, 186 );
139         addRadioField( page.getElements(), 0, 279 );
140         addCheckBoxField( page.getElements(), 0, 372 );
141         addButtonField( page.getElements(), 0, 465 );
142
143         // Add the page to the document
144         document.getPages().add( page );
145     }
146
```

```
147     private void addTextField(Group pageElements, float x, float y ) {
148         TextField txt = new TextField("txtfname", x + 20, y + 40, 120, 20);
149         txt.setDefaultValue( "This is a Scrollable TextField" );
150         txt.setBorderColor( RgbColor.getBlack() );
151         txt.setBackgroundColor( RgbColor.getAliceBlue() );
152         txt.setToolTipText( "Scrollable" );
153         pageElements.add( txt );
154
155         TextField txt1 = new TextField("txtflname", x + 175, y + 40, 120, 20);
156         txt1.setDefaultValue( "TextField" );
157         txt1.setPassword( true );
158         txt1.setMaxLength( 9 );
159         txt1.setBorderColor(RgbColor.getBlack());
160         txt1.setBackgroundColor( RgbColor.getAliceBlue() );
161         txt1.setToolTipText("Password with Maximum Length" );
162         pageElements.add( txt1 );
163
164         TextField txt2 = new TextField("txtf2name", x + 330, y + 30, 150, 40);
165         txt2.setDefaultValue( "This is a TextField which goes to the next line if the text exceeds width" );
166         txt2.setMultiLine( true );
167         txt2.setBorderColor( RgbColor.getBlack() );
168         txt2.setBackgroundColor( RgbColor.getAliceBlue() );
169         txt2.setToolTipText( "Multiline" );
170         pageElements.add( txt2 );
171
172         addCaptionAndRectangle( pageElements, "TextField Form Page Element:", x, y, 504, 85 );
173     }
174
175     private void addComboField( Group pageElements, float x, float y ) {
176         ComboBox cb = new ComboBox("cmbname", x + 51, y + 40, 150, 20);
177         cb.setBorderColor( RgbColor.getBlack() );
178         cb.setBackgroundColor( RgbColor.getAliceBlue());
179         cb.setFont(Font.getHelvetica());
180         cb.setFontSize( 12 );
181         cb.getItems().add("Item 1");
182         cb.getItems().add("Item 2");
183         cb.getItems().add("Item 3");
184         cb.getItems().add("Item 4");
185         cb.getItems().add("Editable");
186         cb.getItems().getChoiceItem("Editable").setSelected(true);
187         cb.setEditable( true );
188         cb.setToolTipText( "Editable Combo Box" );
189         pageElements.add( cb );
190
191         ComboBox cb1 = new ComboBox("cmb1name", x + 303, y + 40, 150, 20);
192         cb1.setBorderColor(RgbColor.getBlack());
193         cb1.setBackgroundColor( RgbColor.getAliceBlue());
194         cb1.setFont(Font.getHelvetica());
195         cb1.setFontSize(12 );
```

```
196     cbl.getItems().add("Item 1");
197     cbl.getItems().add("Item 2");
198     cbl.getItems().add("Item 3");
199     cbl.getItems().add("Item 4");
200     cbl.getItems().add("Non-Editable");
201     cbl.getItems().getChoiceItem( "Non-Editable" ).setSelected(true);
202     cbl.setEditable(false);
203     cbl.setToolTipText("Non-Editable Combo Box");
204     pageElements.add( cbl );
205
206     addCaptionAndRectangle( pageElements, "ComboBox Form Page Element:", x, y, 504, 85 );
207 }
208
209 private void addListField( Group pageElements, float x, float y ) {
210     ListBox lb = new ListBox("lbname", x + 51, y + 30, 150, 40);
211     lb.setBorderColor(RgbColor.getBlack());
212     lb.setBackgroundColor( RgbColor.getAliceBlue() );
213     lb.setFont(Font.getHelvetica());
214     lb.setFontSize(12);
215     lb.getItems().add("Item 1");
216     lb.getItems().add("Item 2");
217     lb.getItems().add("Item 3");
218     lb.getItems().add("Item 4");
219     lb.getItems().add("Non-MultiSelect");
220     lb.getItems().getChoiceItem( "Non-MultiSelect" ).setSelected(true);
221     lb.setMultiselect( false );
222     lb.setToolTipText( "Simple List Box" );
223     pageElements.add( lb );
224
225     ListBox lbl = new ListBox("lblname", x + 303, y + 30, 150, 40);
226     lbl.setBorderColor(RgbColor.getBlack());
227     lbl.setBackgroundColor( RgbColor.getAliceBlue() );
228     lbl.setFont(Font.getHelvetica());
229     lbl.setFontSize(12);
230     lbl.getItems().add("Item 1");
231     lbl.getItems().add("Item 2");
232     lbl.getItems().add("Item 3");
233     lbl.getItems().add("Item 4");
234     lbl.getItems().getChoiceItem("Item 1").setSelected(true);
235     lbl.getItems().add("MultiSelect");
236     lbl.getItems().getChoiceItem("MultiSelect").setSelected(true);
237     lbl.setMultiselect( true );
238     lbl.setToolTipText( "MultiSelect List Box" );
239     pageElements.add( lbl );
240
241     addCaptionAndRectangle( pageElements, "ListBox Form Page Element:", x, y, 504, 85 );
242 }
243
244 private void addRadioField( Group pageElements, float x, float y ) {
```

```
245     RadioButton rb = new RadioButton( "rbname", x + 40, y + 42, 12, 12 );
246     rb.setDefaultChecked( true );
247     rb.setBackgroundColor( RgbColor.getAliceBlue() );
248     rb.setTextColor( RgbColor.getRed() );
249     rb.setExportValue( "Radio1" );
250     Label lb = new Label("Monthly", x + 75, y + 43, 60, 15 );
251     pageElements.add( rb );
252     pageElements.add( lb );
253
254     RadioButton rbl = new RadioButton( "rbname", x + 195, y + 42, 12, 12 );
255     rbl.setBackgroundColor(RgbColor.getAliceBlue());
256     rbl.setTextColor( RgbColor.getGreen() );
257     rbl.setExportValue( "Radio2" );
258     Label lbl = new Label("Quarterly", x + 235, y + 43, 60, 15 );
259     pageElements.add( rbl );
260     pageElements.add( lbl );
261
262     RadioButton rb2 = new RadioButton( "rbname", x + 350, y + 42, 12, 12 );
263     rb2.setBackgroundColor( RgbColor.getAliceBlue() );
264     rb2.setTextColor( RgbColor.getBlue() );
265     rb2.setExportValue( "Radio3" );
266     Label lb2 = new Label("Yearly", x + 385, y + 43, 60, 15 );
267     pageElements.add( rb2 );
268     pageElements.add( lb2 );
269
270     addCaptionAndRectangle( pageElements, "RadioButton Form Page Element:", x, y, 504, 85 );
271 }
272
273 private void addCheckBoxField( Group pageElements, float x, float y ) {
274     CheckBox cb = new CheckBox( "cbxname", x + 40, y + 42, 15, 15 );
275     cb.setSymbol(com.cete.dynamicpdf.pageelements.forms.Symbol.CHECK) ;
276     cb.setBackgroundColor( RgbColor.getAliceBlue() );
277     Label lb = new Label("Onion",x + 75, y + 43, 60, 15 );
278     pageElements.add( cb );
279     pageElements.add( lb );
280
281     CheckBox cbl = new CheckBox( "cbxsecondname", x + 195, y + 42, 15, 15);
282     cbl.setSymbol(com.cete.dynamicpdf.pageelements.forms.Symbol.CROSS);
283     cbl.setDefaultChecked( true );
284     cbl.setBackgroundColor(RgbColor.getAliceBlue() );
285     Label lbl = new Label("Capsicum",x + 235, y + 43, 60, 15 );
286     pageElements.add( cbl );
287     pageElements.add( lbl );
288
289     CheckBox cb2 = new CheckBox( "cbxthirdname", x + 350, y + 42, 15, 15);
290     cb2.setSymbol(com.cete.dynamicpdf.pageelements.forms.Symbol.STAR);
291     cb2.setBackgroundColor( RgbColor.getAliceBlue() );
292     Label lb2 = new Label("Jalapenos",x + 385, y + 43, 80, 15 );
293     pageElements.add( cb2 );
```

```
294     pageElements.add( lb2 );
295
296     addCaptionAndRectangle( pageElements, "CheckBox Form Page Element:", x, y, 504, 85 );
297 }
298
299 private void addButtonField( Group pageElements, float x, float y ) {
300     Button bt = new Button("btonename", x + 40, y + 40, 100, 25);
301     bt.setBehavior(Behavior.PUSH);
302     bt.setBorderStyle(BorderStyle.BEVELED);
303     SubmitAction sb = new SubmitAction("http://localhost:8080/");
304     bt.setToolTip("Submit Button");
305     bt.setLabel("Submit");
306     bt.setAction(sb);
307     pageElements.add(bt);
308
309     Button bt1 = new Button( "bttwoname", x + 195, y + 40, 100, 25 );
310     bt1.setBehavior( Behavior.PUSH );
311     bt1.setBorderStyle( BorderStyle.BEVELED );
312     ResetAction rb = new ResetAction();
313     bt1.setAction( rb );
314     bt1.setToolTip( "Reset Button" );
315     bt1.setLabel( "Reset" );
316     pageElements.add( bt1 );
317
318     Button bt2 = new Button( "btthreename", x + 350, y + 40, 100, 25 );
319     bt2.setBehavior( Behavior.PUSH );
320     bt2.setBorderStyle( BorderStyle.BEVELED );
321     bt2.setAction( new ZoomDestination(pageCounter, PageZoom.FIT_WIDTH ));
322     bt2.setBehavior( Behavior.createPush("Click","Zoom"));
323     bt2.setToolTip( "Zoom to fit page width" );
324     bt2.setLabel( "Zoom" );
325     pageElements.add( bt2 );
326
327     Button bt3 = new Button( "btfourname", x + 101, y + 90, 100, 25 );
328     bt3.setBehavior( Behavior.PUSH );
329     bt3.setBorderStyle( BorderStyle.BEVELED );
330     bt3.setAction( new XYDestination(pageCounter, 100, 100));
331     bt3.setBehavior( Behavior.createPush("Click", "Go To XY Destination"));
332     bt3.setToolTip( "Go to XY position" );
333     bt3.setLabel( "XY" );
334     pageElements.add( bt3 );
335
336     Button bt4 = new Button( "btfivename", x + 302, y + 90, 100, 25 );
337     bt4.setBehavior( Behavior.PUSH );
338     bt4.setBorderStyle( BorderStyle.BEVELED );
339     bt4.setAction( new JavaScriptAction( "app.alert('Hello');" ));
340     bt4.setToolTip( "Javascript Button" );
341     bt4.setLabel( "OK" );
342     pageElements.add( bt4 );
```

```
343
344     addCaptionAndRectangle( pageElements, "Button Form Page Element:", x, y, 504, 140 );
345 }
346
347 private void addListElements( Page page, float x, float y ) {
348     OrderedList orderedList = new OrderedList( x + 5, y + 20, 400, 90, Font.getHelvetica(), 10 );
349     orderedList.getListItems().add("Fruits");
350     orderedList.getListItems().add("Vegetables");
351
352     OrderedSubList orderedSubList = orderedList.getListItems().getItem(0).getSubLists().addOrderedSubList(
353 NumberingStyle.ROMAN_UPPERCASE );
354     orderedSubList.getListItems().add("Citrus");
355     orderedSubList.getListItems().add("Non-Citrus");
356
357     addCaptionAndRectangle( page.getElements(), "Ordered List Page Element:", x, y, 225, 110 );
358     OrderedSubList orderedSubList2 = orderedList.getListItems().getItem(1).getSubLists().addOrderedSubList(
359 NumberingStyle.ROMAN_UPPERCASE );
360     orderedSubList2.getListItems().add("Potato");
361     orderedSubList2.getListItems().add("Beans");
362
363     OrderedSubList subOrderedSubList = orderedSubList.getListItems().getItem(0).getSubLists().addOrderedSubList(
364 NumberingStyle.ROMAN_LOWERCASE );
365     subOrderedSubList.getListItems().add("Lime");
366     subOrderedSubList.getListItems().add("Orange");
367
368     OrderedSubList subOrderedSubList2 = orderedSubList.getListItems().getItem(1).getSubLists().addOrderedSubList(
369 NumberingStyle.ROMAN_LOWERCASE );
370     subOrderedSubList2.getListItems().add("Mango");
371     subOrderedSubList2.getListItems().add("Banana");
372
373     OrderedSubList subOrderedSubList3 = orderedSubList2.getListItems().getItem(0).getSubLists().addOrderedSubList(
374 NumberingStyle.ROMAN_LOWERCASE );
375     subOrderedSubList3.getListItems().add("Sweet Potato");
376
377     OrderedSubList subOrderedSubList4 = orderedSubList2.getListItems().getItem(1).getSubLists().addOrderedSubList(
378 NumberingStyle.ROMAN_LOWERCASE );
379     subOrderedSubList4.getListItems().add("String Bean");
380     subOrderedSubList4.getListItems().add("Lima Bean");
381     subOrderedSubList4.getListItems().add("Kidney Bean");
382
383     page.getElements().add( orderedList );
384     x += 279;
385
386     orderedList = orderedList.getOverflowList(x + 5, y + 20);
387     addCaptionAndRectangle( page.getElements(), "Ordered List Page Element Overflow:", x, y, 225, 110 );
388     page.getElements().add( orderedList );
389
390     x = 0;
391     y += 118;
```



```
386
387 // Create an unordered list
388 UnorderedList unorderedList = new UnorderedList(x + 5, y + 20, 400, 90, Font.getHelvetica(), 10 );
389 unorderedList.getListItems().add("Fruits");
390 unorderedList.getListItems().add("Vegetables");
391
392 UnorderedSubList unorderedSubList = unorderedList.getListItems().getItem(0).getSubLists().addUnorderedSubList();
393 unorderedSubList.getListItems().add(" Citrus");
394 unorderedSubList.getListItems().add(" Non-Citrus");
395
396 addCaptionAndRectangle( page.getElements(), "Unordered List Page Element:", x , y, 225, 110 );
397
398 UnorderedSubList unorderedSubList2 = unorderedList.getListItems().getItem(1).getSubLists().addUnorderedSubList( );
399 unorderedSubList2.getListItems().add("Potato");
400 unorderedSubList2.getListItems().add("Beans");
401
402 UnorderedSubList subUnorderedSubList = unorderedSubList.getListItems().getItem(0).getSubLists().addUnorderedSubList( );
403 subUnorderedSubList.getListItems().add("Lime");
404 subUnorderedSubList.getListItems().add("Orange");
405
406 UnorderedSubList subUnorderedSubList2 = unorderedSubList.getListItems().getItem(1).getSubLists().addUnorderedSubList(
407 );
408 subUnorderedSubList2.getListItems().add("Mango");
409 subUnorderedSubList2.getListItems().add("Banana");
410
411 UnorderedSubList subUnorderedSubList3 = unorderedSubList2.getListItems().getItem(0).getSubLists().addUnorderedSubList(
412 );
413 subUnorderedSubList3.getListItems().add("Sweet Potato");
414
415 UnorderedSubList subUnorderedSubList4 = unorderedSubList2.getListItems().getItem(1).getSubLists().addUnorderedSubList(
416 );
417 subUnorderedSubList4.getListItems().add("String Bean");
418 subUnorderedSubList4.getListItems().add("Lima Bean");
419 subUnorderedSubList4.getListItems().add("Kidney Bean");
420
421 x += 279;
422 page.getElements().add( unorderedList );
423 unorderedList = unorderedList.getOverflowList(x + 5, y + 20);
424 addCaptionAndRectangle( page.getElements(), "Unordered List Page Element Overflow:", x, y, 225, 110 );
425 page.getElements().add( unorderedList );
426 }
427
428 private void addPageElementPage(Document document, Outline parentOutline) {
429 // Increments a page counter
430 pageCounter++;
431
432 // Adds to outlines to the page
433 Outline pageOutline = parentOutline.getChildOutlines().add("Page "
434 + pageCounter);
```

```
432
433     pageOutline.getChildOutlines().add("Zoom - Fit Page",
434         new ZoomDestination(pageCounter, PageZoom.FIT_PAGE));
435     pageOutline.getChildOutlines().add("Zoom - Fit Width",
436         new ZoomDestination(pageCounter, PageZoom.FIT_WIDTH));
437     //create a new Page.
438     Page page = new Page(PageSize.LETTER, PageOrientation.PORTRAIT, 54);
439
440     addHeader(page.getElements());
441
442     addBookmark(page.getElements(), 0, 41, pageOutline);
443
444     addCircle(page.getElements(), 279, 41);
445
446     addFormattedTextArea(page.getElements(), 0, 134);
447
448     addImage(page.getElements(), 0, 227);
449
450     addLabel(page.getElements(), 279, 227);
451
452     addLine(page.getElements(), 0, 320);
453
454     addLink(page.getElements(), 279, 320);
455
456     addPath(page.getElements(), 0, 413);
457
458     addRectangle(page.getElements(), 279, 413);
459
460     addTextArea(page.getElements(), 0, 506);
461
462     addCustom(page.getElements(), 0, 599);
463
464     addNote(page.getElements(), 279, 599);
465
466
467     // Add the page to the document
468     document.getPages().add(page);
469
470
471 }
472
473 private void addHeader(Group pageElements) {
474     // Adds a header to the pageElements
475     String strFormattedText = "<i>Dynamic</i><b>PDF</b>; Generator for" +
476         "Java has over 50 built-in page " +
477         "elements. Custom page elements can be built by " +
478         "creating a class that inherits from " +
479         "<b>com.cete.dynamicpdf.PageElement</b>.";
480     //create a FormattedTextArea which can contain text with Formatted tags.
```

```
481     FormattedTextArea formattedTextArea = new FormattedTextArea(strFormattedText, 0, 0, 504,
482         450, FontFamily.getHelvetica(), 12,
483         false);
484     formattedTextArea.getStyle().paragraph.align = TextAlign.CENTER;
485
486     pageElements.add(formattedTextArea);
487 }
488
489 private void addBookmark(Group pageElements, float x, float y,
490     Outline parentOutline) {
491
492     addCaptionAndRectangle(pageElements, "Bookmark Page Element:", x, y);
493
494     // Adds a bookmark to the page
495     pageElements.add(new Bookmark("Bookmarked Text", x + 5, y + 20,
496         parentOutline));
497
498     pageElements.add(new Label("This text is bookmarked.", x + 5, y + 20,
499         215, 10, Font.getTimesRoman(), 10));
500
501 }
502
503 private void addCircle(Group pageElements, float x, float y) {
504     addCaptionAndRectangle(pageElements, "Circle Page Element:", x, y);
505
506     // Adds a circle to the pageElements
507     pageElements.add(new Circle(x + 112.5f, y + 50f, 107.5f, 30f,
508         RgbColor.getRed(), RgbColor.getBlue(), 2,
509         LineStyle.getDashLarge()));
510
511 }
512
513 private void addFormattedTextArea(Group pageElements, float x, float y) {
514     // Adds an Formatted text area to the pageElements
515     String strFormattedText = "<p><i>Dynamic</i><b>PDF</b>; Generator " +
516         "for Java has an Formatted text area page " +
517         "element. This provides rich formatting support " +
518         "for text that appears in the document. You have " +
519         "complete control over 8 paragraph properties: " +
520         "spacing before, spacing after, first line " +
521         "indentation, left indentation, right " +
522         "indentation, alignment, allowing orphan lines, " +
523         "and white space preservation; 6 font properties:" +
524         "<font face='Times'>font face, </font>" +
525         "<font pointSize='6'>font " +
526         "size, </font><font color='FF0000'>color, " +
527         "</font><b>bold, </b><i>italic and " +
528         "</i><u>underline</u>; and 2 line properties: " +
529         "leading, and leading type. Text can also be " +
530         "rotated.</p>";
```

```
530     FormattedTextArea objFormattedTextArea = new FormattedTextArea(strFormattedText, x + 5,
531         y + 20, 215, 60,
532         FontFamily.getHelvetica(),
533         9, false);
534     // Sets the indent property
535     objFormattedTextArea.getStyle().paragraph.indent = 18;
536
537     addCaptionAndRectangle(pageElements, "FormattedTextArea Page Element:", x,
538         y);
539     addCaptionAndRectangle(pageElements, "FormattedTextArea Overflow Text:",
540         x + 279, y);
541
542     pageElements.add(objFormattedTextArea);
543
544     // Create an overflow Formatted text area for the overflow text
545     FormattedTextArea objOverflowFormattedTextArea = objFormattedTextArea
546         .getOverflowFormattedTextArea(x
547         + 284,y + 20);
548
549     pageElements.add(objOverflowFormattedTextArea);
550
551 }
552
553 private void addImage(Group pageElements, float x, float y) {
554     // Adds an image to the pageElements
555     addCaptionAndRectangle(pageElements, "Image Page Element:", x, y);
556     Image objImage = null;
557     try {
558         objImage = new Image(getServletContext().getRealPath("/images" +
559             "/DPDFLogo.png"), x + 112.5f, y + 50f, 0.24f);
560
561     } catch (FileNotFoundException ex1) {
562         System.err.println("Image File Not Found: "+ex1);
563     }
564     // Image is sized and centered in the rectangle
565     objImage.setBounds(215, 60);
566     objImage.setVAlign(VAlign.CENTER);
567     objImage.setAlign(Align.CENTER);
568
569     pageElements.add(objImage);
570 }
571
572 private void addLabel(Group pageElements, float x, float y) {
573     // Adds a label to the pageElements
574     addCaptionAndRectangle(pageElements, "Label and PageNumberingLabel Page Elements:", x, y);
575
576     String labelText = "Labels can be rotated";
577     String numberText = "PageNumberingLabels contain page numbering: %%CP%% of %%TP%% pages.";
578 }
```

```
579     Label label = new Label(labelText, x + 5, y + 12, 220, 80,
580         Font.getTimesRoman(), 12, TextAlign.CENTER);
581     label.setAngle(8);
582
583     PageNumberingLabel pageNumLabel = new PageNumberingLabel(numberText,
584         x + 5, y + 55, 220, 80, Font.getTimesRoman(), 12, TextAlign.CENTER);
585
586     pageElements.add(pageNumLabel);
587     pageElements.add(label);
588 }
589
590 private void addLine(Group pageElements, float x, float y)    {
591     // Adds a line to the pageElements
592     addCaptionAndRectangle(pageElements, "Line Page Element:", x, y);
593
594     pageElements.add(new Line(x + 5, y + 20, x + 220, y + 80, 3,
595         RgbColor.getGreen()));
596     pageElements.add(new Line(x + 220, y + 20, x + 5, y + 80, 3,
597         RgbColor.getGreen()));
598 }
599
600
601
602
603 private void addLink(Group pageElements, float x, float y) {
604     // Adds a link to the pageElements
605     Font objFont = Font.getTimesRoman();
606     String strText = "This is a link to DynamicPDF.com.";
607
608     addCaptionAndRectangle(pageElements, "Link Page Element:", x, y);
609
610     Label objLabel = new Label(strText, x + 5, y + 20, 215, 80, objFont, 12,
611         RgbColor.getBlue());
612     objLabel.setUnderline(true);
613
614     Link objLink = new Link(x + 5, y + 20, objFont.getTextWidth(strText, 12),
615         12 - objFont.getDescender(12),
616         new UrlAction("http://www.dynamicpdf.com"));
617     pageElements.add(objLabel);
618     pageElements.add(objLink);
619 }
620
621 private void addPath(Group pageElements, float x, float y) {
622     // Adds a path to the pageElements
623     Path objPath = new Path(x + 5, y + 20, RgbColor.getBlue(), RgbColor.getRed(),
624         2, LineStyle.getSolid(), true);
625     objPath.getSubPaths().add(new LineSubPath(x + 215, y + 40));
626     objPath.getSubPaths().add(new CurveToSubPath(x + 108, y + 80, x + 160,
627         y + 80));
```

```
628     objPath.getSubPaths().add(new CurveSubPath(x + 5, y + 40, x + 65, y + 80,
629         x + 5, y + 60));
630
631     addCaptionAndRectangle(pageElements, "Path Page Element:", x, y);
632
633     pageElements.add(objPath);
634 }
635
636 private void addRectangle(Group pageElements, float x, float y) {
637     // Adds a rectangle to the pageElements
638     addCaptionAndRectangle(pageElements, "Rectangle Page Element:", x, y);
639
640     pageElements.add(new Rectangle(x + 5, y + 20, 215, 60, RgbColor.getGreen(),
641         RgbColor.getBlue(), 2));
642 }
643
644 private void addTextArea(Group pageElements, float x, float y) {
645     // Adds a text area to the pageElements
646     String strText = "The TextArea page element is great for efficient " +
647         "processing of text that does not require advanced " +
648         "formatting. It offers basic control over leading " +
649         "and paragraph spacing and include text continuation" +
650         " functionality. If all the text does not fit within " +
651         "the bounds of the TextArea the hasOverflowText " +
652         "method will return a true value. The " +
653         "getOverflowTextArea method can then be called to " +
654         "create a new TextArea with the remaining text.This " +
655         "functionality makes it simple to span large amounts of " +
656         "text over multiple TextAreas and pages. " +
657         "Text can also be rotated.";
658
659     addCaptionAndRectangle(pageElements, "TextArea Page Element:", x, y);
660     addCaptionAndRectangle(pageElements, "TextArea Overflow Text:",
661         x + 279, y);
662
663     TextArea objTextArea = new TextArea(strText, x + 5, y + 20, 215, 60,
664         Font.getTimesRoman(), 10);
665     // Create an overflow text area for the overflow text
666     TextArea objOverflowTextArea = objTextArea.getOverflowTextArea(x + 284,
667         y + 20);
668
669     pageElements.add(objTextArea);
670     pageElements.add(objOverflowTextArea);
671 }
672
673 private void addCustom(Group pageElements, float x, float y) {
674     // Adds a custom page element to the page
675     addCaptionAndRectangle(pageElements, "Custom Page Element:", x, y);
676 }
```

```
677     pageElements.add(new StopSign(80, y + 20, 60, 60));
678 }
679
680 private void addNote(Group pageElements, float x, float y) {
681     // Add a note element to the page
682     addCaptionAndRectangle(pageElements, "Note Page Element:", x, y);
683
684     pageElements.add(new Note("This is a note", x + 100, y + 35, 50, 50,
685         NoteType.NOTE));
686
687 }
688
689 private void addTableAndOther(Document document, Outline parentOutline) {
690     // Increments a page counter
691     pageCounter++;
692
693     // Adds to outlines to the page
694     Outline pageOutline = parentOutline.getChildOutlines().add("Page "
695         + pageCounter + " - Table and Additional Page Elements",
696         new ZoomDestination(pageCounter, PageZoom.FIT_PAGE));
697
698     // Create a page to add to the document
699     Page page = new Page(612, 729, 54);
700
701     // Add the Table and other elements to the page
702     addTable(page.getElements(), 0, 0);
703     addListElements( page, 0, 205 );
704     addAdditional(page.getElements(), 0, 441);
705
706     // Add the page to the document
707     document.getPages().add(page);
708 }
709
710 private void addTable(Group pageElements, float x, float y) {
711     Label captionLabel1 = new Label("Table Page Element:", x, y, 225, 10,
712         Font.getHelveticaBold(), 10);
713     pageElements.add(captionLabel1);
714
715     Label captionLabel2 = new Label("Table Row Overflow:", x + 279, y,
716         225, 10, Font.getHelveticaBold(), 10);
717     pageElements.add(captionLabel2);
718
719     Table2 table = new Table2(x, y + 15, 225, 190, Font.getTimesRoman(), 10);
720     table.setRepeatColumnHeaderCount( 1 );
721
722     // Add columns to the table
723     table.getColumns().add(95);
724     table.getColumns().add(65);
725     table.getColumns().add(65);
```

```
726
727     // Add the Header Row to the table
728     Row2 row1 = table.getRows().add(40, Font.getHelveticaBold(), 12,
729         Grayscale.getBlack(), RgbColor.getLightBlue());
730     row1.getCellDefault().setAlign(TextAlign.CENTER);
731     row1.getCellDefault().setVAlign(VAlign.CENTER);
732     row1.getCells().add("Header 1");
733     row1.getCells().add("Header 2");
734     row1.getCells().add("Header 3");
735     row1.getCells().add("Header 4");
736     row1.getCells().add("Header 5");
737     row1.getCells().add("Header 6");
738     row1.getCells().add("Header 7");
739     row1.getCells().add("Header 8");
740     row1.getCells().add("Header 9");
741     row1.getCells().add("Header 10");
742     row1.getCells().add("Header 11");
743     row1.getCells().add("Header 12");
744     row1.getCells().add("Header 13");
745     row1.getCells().add("Header 14");
746     row1.getCells().add("Header 15");
747     row1.getCells().add("Header 16");
748
749     int cell = 0;
750     // Add the rest of the rows to the table
751     addRow(table, cell, 1);
752     addRow(table, cell, 2);
753     addRow(table, cell, 3);
754     addRow(table, cell, 4);
755     addRow(table, cell, 5);
756     addRow(table, cell, 6);
757     addRow(table, cell, 7);
758     addRow(table, cell, 8);
759     addRow(table, cell, 9);
760     addRow(table, cell, 10);
761     addRow(table, cell, 11);
762     addRow(table, cell, 12);
763     addRow(table, cell, 13);
764     addRow(table, cell, 14);
765     addRow(table, cell, 15);
766     addRow(table, cell, 16);
767
768     // Create an overflow table for the overflow columns
769     Table2 overflowTable = table.getOverflowRows(x + 279, y + 15);
770
771     pageElements.add(table);
772     pageElements.add(overflowTable);
773 }
774
```



```
775     private void addRow(Table2 table, int cellNum, int rowNum) {
776         Row2 row = table.getRows().add();
777         Cell2 cell = row.getCells().add("RowHeader " + rowNum,
778             Font.getHelveticaBold(), 12,
779             Grayscale.getBlack(), RgbColor.getLightPink(), 1);
780         cell.setAlign(TextAlign.CENTER);
781         cell.setVAlign(VAlign.CENTER);
782
783         row.getCells().add("Item " + cellNum++);
784         row.getCells().add("Item " + cellNum++);
785     }
786
787     private void addAdditional(Group pageElements, float x, float y) {
788         // Adds text explaining the additional page elements
789         String strText = "An <b><u>AreaGroup</u></b> represents a group of page elements with an area.<p/>" +
790             "A <b><u>BackgroundImage</u></b> page element is used to add an image to the background of a page. <p/>" +
791             "The <b><u>Group</u></b> page element is used for grouping page elements together.<p/>" +
792             "The <b><u>LayoutGrid</u></b> draws a coordinate grid to the page.<p/>" +
793             "A <b><u>TransformationGroup</u></b> can rotate and scale a group of page elements.<p/>" +
794             "A <b><u>TransparencyGroup</u></b> can be used to add transparency to page elements.<p/>" +
795             "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/>";
796         FormattedTextArea objFormattedTextArea = new FormattedTextArea(strText, x + 5, y + 20,
797             494, 180, FontFamily.getTimes(),
798             12, false);
799         objFormattedTextArea.getStyle().paragraph.spacingAfter = 5;
800
801         addCaptionAndRectangle(pageElements, "Additional Page Elements:", x, y, 504, 180);
802
803         pageElements.add(objFormattedTextArea);
804     }
805
806     private void addChartsPage(Document document, Outline parentOutline) {
807         // Increments a page counter
808         pageCounter++;
809
810         // Adds outlines to the page
811         parentOutline.getChildOutlines().add("Page " + pageCounter + " Charts", new ZoomDestination(pageCounter,
PageZoom.FIT_PAGE));
812
813         // Create a page
814         Page page = new Page(PageSize.LETTER, PageOrientation.LANDSCAPE);
815         addAreaChart(page.getElements(), 0, 25);
816         addLineChart(page.getElements(), 375, 25);
817         addPieChart(page.getElements(), 0, 260);
818         addBarChart(page.getElements(), 375, 260);
819         document.getPages().add(page);
820     }
821 }
```

```
822 private void addAreaChart(Group elements, float x, float y) {
823     addCaptionAndRectangle(elements, "Stacked Area Chart", x, y, 325, 225);
824
825     // Create a chart
826     Chart chart = new Chart(x + 20, y + 25, 300, 200);
827
828     // Create a plot area
829     PlotArea plotArea = chart.getPrimaryPlotArea();
830
831     // Create header title and add it to the chart
832     Title title1 = new Title("Website Visitors");
833     chart.getHeaderTitles().add(title1);
834
835     // Create indexed stacked area series elements and add values to it
836     IndexedStackedAreaSeriesElement seriesElement1 = new IndexedStackedAreaSeriesElement("Website A");
837     seriesElement1.getValues().add(new float[] { 5, 7, 9, 6 });
838     IndexedStackedAreaSeriesElement seriesElement2 = new IndexedStackedAreaSeriesElement("Website B");
839     seriesElement2.getValues().add(new float[] { 4, 2, 5, 8 });
840     IndexedStackedAreaSeriesElement seriesElement3 = new IndexedStackedAreaSeriesElement("Website C");
841     seriesElement3.getValues().add(new float[] { 2, 4, 6, 9 });
842
843     // Create autogradient and add it to the series
844     AutoGradient autogradient1 = new AutoGradient(90f, CmykColor.getRed(), CmykColor.getIndianRed());
845     seriesElement1.setColor(autogradient1);
846     AutoGradient autogradient2 = new AutoGradient(90f, CmykColor.getGreen(), CmykColor.getYellowGreen());
847     seriesElement2.setColor(autogradient2);
848     AutoGradient autogradient3 = new AutoGradient(120f, CmykColor.getBlue(), CmykColor.getLightBlue());
849     seriesElement3.setColor(autogradient3);
850
851     // Create a Indexed Stacked Area Series
852     IndexedStackedAreaSeries areaSeries = new IndexedStackedAreaSeries();
853
854     // Add indexed stacked area series elements to the Indexed Stacked Area Series
855     areaSeries.add(seriesElement1);
856     areaSeries.add(seriesElement2);
857     areaSeries.add(seriesElement3);
858
859     // Add series to the plot area
860     plotArea.getSeries().add(areaSeries);
861
862     // Create a title and add it to the YAxis
863     Title lTitle = new Title("Visitors (in millions)");
864     areaSeries.getYAxis().getTitles().add(lTitle);
865
866     //Adding AxisLabels to the XAxis
867     areaSeries.getXAxis().getLabels().add(new IndexedXAxisLabel("Q1", 0));
868     areaSeries.getXAxis().getLabels().add(new IndexedXAxisLabel("Q2", 1));
869     areaSeries.getXAxis().getLabels().add(new IndexedXAxisLabel("Q3", 2));
870     areaSeries.getXAxis().getLabels().add(new IndexedXAxisLabel("Q4", 3));
```

```
871     elements.add(chart);
872 }
873
874 private void addLineChart(Group elements, float x, float y) {
875     addCaptionAndRectangle(elements, "Line Chart", x, y, 325, 225);
876
877     // Create a chart
878     Chart chart = new Chart(x + 20, y + 25, 300, 200);
879
880     // Create a plot area
881     PlotArea plotArea = chart.getPrimaryPlotArea();
882
883     // Create header titles and add it to the chart
884     Title title1 = new Title("Website Visitors");
885
886     chart.getHeaderTitles().add(title1);
887
888
889     // Create a indexed line series and add values to it
890     IndexedLineSeries lineSeries1 = new IndexedLineSeries("Website A");
891     lineSeries1.getValues().add(new float[] { 5, 7, 9, 6 });
892     IndexedLineSeries lineSeries2 = new IndexedLineSeries("Website B");
893     lineSeries2.getValues().add(new float[] { 4, 2, 5, 8 });
894     IndexedLineSeries lineSeries3 = new IndexedLineSeries("Website C");
895     lineSeries3.getValues().add(new float[] { 2, 4, 6, 9 });
896
897     // Add indexed line series to the plot area
898     plotArea.getSeries().add(lineSeries1);
899     plotArea.getSeries().add(lineSeries2);
900     plotArea.getSeries().add(lineSeries3);
901
902     // Create a title and add it to the yaxis
903     Title lTitle = new Title("Visitors (in millions)");
904     lineSeries1.getYAxis().getTitles().add(lTitle);
905
906     //Adding AxisLabels to the XAxis
907     lineSeries1.getXAxis().getLabels().add(new IndexedXAxisLabel("Q1", 0));
908     lineSeries1.getXAxis().getLabels().add(new IndexedXAxisLabel("Q2", 1));
909     lineSeries1.getXAxis().getLabels().add(new IndexedXAxisLabel("Q3", 2));
910     lineSeries1.getXAxis().getLabels().add(new IndexedXAxisLabel("Q4", 3));
911     elements.add(chart);
912 }
913
914 private void addPieChart(Group elements, float x, float y) {
915     addCaptionAndRectangle(elements, "Pie Chart", x, y, 325, 225);
916
917     // Create a chart
918     Chart chart = new Chart(x + 20, y + 25, 300, 200);
919
```

```
920 // Add a plot area to the chart
921 PlotArea plotArea = chart.getPlotAreas().add(50, 50, 300, 300);
922
923 // Create the Header titles and add it to the chart
924 Title tTitle = new Title("Website Visitors (in millions)");
925
926 chart.getHeaderTitles().add(tTitle);
927
928 // Create autogradient and add it to the series
929 AutoGradient autogradient1 = new AutoGradient(90f, CmykColor.getRed(), CmykColor.getIndianRed());
930 AutoGradient autogradient2 = new AutoGradient(90f, CmykColor.getGreen(), CmykColor.getYellowGreen());
931 AutoGradient autogradient3 = new AutoGradient(120f, CmykColor.getBlue(), CmykColor.getLightBlue());
932
933 // Create a scalar datalabel
934 ScalarDataLabel da = new ScalarDataLabel(true, false, false);
935
936 // Create a pie series
937 PieSeries pieSeries = new PieSeries();
938
939 // Set scalar datalabel to the pie series
940 pieSeries.setDataLabel(da);
941
942 // Add series to the plot area
943 plotArea.getSeries().add(pieSeries);
944
945 //Add pie series elements to the pie series
946 pieSeries.getElements().add(27, "Website A");
947 pieSeries.getElements().add(19, "Website B");
948 pieSeries.getElements().add(21, "Website C");
949 pieSeries.getElements().getElement(0).setColor(autogradient1);
950 pieSeries.getElements().getElement(1).setColor(autogradient2);
951 pieSeries.getElements().getElement(2).setColor(autogradient3);
952 elements.add(chart);
953 }
954
955 private void addBarChart(Group elements, float x, float y) {
956     addCaptionAndRectangle(elements, "Stacked Bar Chart", x, y, 325, 225);
957
958     // Create a chart
959     Chart chart = new Chart(x + 20, y + 25, 300, 200);
960
961     // Create a plot area
962     PlotArea plotArea = chart.getPrimaryPlotArea();
963
964     // Create header titles and add it to the chart
965     Title title1 = new Title("Website Visitors");
966
967     chart.getHeaderTitles().add(title1);
968 }
```

```
969
970 // Create a indexed stacked bar series elements and add values to it
971 IndexedStackedBarSeriesElement seriesElement1 = new IndexedStackedBarSeriesElement("Website A");
972 seriesElement1.getValues().add(new float[] { 5, 7, 9, 6 });
973 IndexedStackedBarSeriesElement seriesElement2 = new IndexedStackedBarSeriesElement("Website B");
974 seriesElement2.getValues().add(new float[] { 4, 2, 5, 8 });
975 IndexedStackedBarSeriesElement seriesElement3 = new IndexedStackedBarSeriesElement("Website C");
976 seriesElement3.getValues().add(new float[] { 2, 4, 6, 9 });
977
978 // Create a Indexed Stacked Bar Series
979 IndexedStackedBarSeries barSeries = new IndexedStackedBarSeries();
980
981 // Create autogradient and add it to the series
982 AutoGradient autogradient1 = new AutoGradient(90f, CmykColor.getRed(), CmykColor.getIndianRed());
983 AutoGradient autogradient2 = new AutoGradient(90f, CmykColor.getGreen(), CmykColor.getYellowGreen());
984 AutoGradient autogradient3 = new AutoGradient(90f, CmykColor.getBlue(), CmykColor.getLightBlue());
985 seriesElement1.setColor(autogradient1);
986 seriesElement2.setColor(autogradient2);
987 seriesElement3.setColor(autogradient3);
988
989 // Add indexed stacked bar series elements to the Indexed Stacked Bar Series
990 barSeries.add(seriesElement1);
991 barSeries.add(seriesElement2);
992 barSeries.add(seriesElement3);
993
994 //Add series to the plot area
995 plotArea.getSeries().add(barSeries);
996
997 // Create a title and add it to the xaxis
998 Title lTitle = new Title("Visitors (in millions)");
999 barSeries.getXAxis().getTitles().add(lTitle);
1000
1001 //Adding AxisLabels to the yaxis
1002 barSeries.getYAxis().getLabels().add(new IndexedYAxisLabel("Q1", 0));
1003 barSeries.getYAxis().getLabels().add(new IndexedYAxisLabel("Q2", 1));
1004 barSeries.getYAxis().getLabels().add(new IndexedYAxisLabel("Q3", 2));
1005 barSeries.getYAxis().getLabels().add(new IndexedYAxisLabel("Q4", 3));
1006 barSeries.getXAxis().getLabels().setAngle(-65);
1007 elements.add(chart);
1008 }
1009
1010
1011 private void addMSIcode(Group elements, int x, int y) {
1012     addCaptionAndRectangle(elements, "MSI Barcode", x, y, 225, 80);
1013     MsiBarcode msi = new MsiBarcode("1234567890", x, y, 50, true);
1014     msi.setY(msi.getY() + (80 - msi.getHeight()) / 2 + 8);
1015     msi.setX(msi.getX() + (225 - msi.getSymbolWidth()) / 2);
1016     elements.add(msi);
1017 }
```

```
1018
1019
1020 private void addRm4sccCode(Group elements, int x, int y) {
1021     addCaptionAndRectangle(elements, "RM4SCC Barcode", x, y, 225, 80);
1022     Rm4scc rm4sccCode = new Rm4scc("1457826RA", x, y);
1023     rm4sccCode.setY(rm4sccCode.getY() + (65 - rm4sccCode.getHeight()) / 2 + 13);
1024     rm4sccCode.setX(rm4sccCode.getX() + (225 - rm4sccCode.getSymbolWidth()) / 2);
1025     elements.add(rm4sccCode);
1026 }
1027
1028
1029 private void addBarCodePage(Document document, Outline parentOutline) {
1030     // Increments a page counter
1031     pageCounter++;
1032
1033     // Adds to outlines to the page
1034     Outline pageOutline = parentOutline.getChildOutlines().add("Page "
1035         + pageCounter + " - Bar Codes",
1036         new ZoomDestination(pageCounter,
1037             PageZoom.FIT_PAGE));
1038
1039     // Create a page to add to the document
1040     com.cete.dynamicpdf.Page page = new com.cete.dynamicpdf.Page(PageSize.LETTER, PageOrientation.PORTRAIT, 54);
1041
1042     // Add the page elements to the page
1043     addCodabar(page.getElements(), 0, 0);
1044     addCode128(page.getElements(), 0, 88);
1045     addCode25(page.getElements(), 0, 176);
1046     addCode39(page.getElements(), 0, 264);
1047     addInterleaved25(page.getElements(), 0, 352);
1048     addPdf417( page.getElements(), 279, 0 );
1049     addPdf417Macro( page.getElements(), 279, 88 );
1050     addDataMatrixBarcode(page.getElements(), 279, 264);
1051     addQrCode(page.getElements(),279,440);
1052     addMSICode(page.getElements(), 0, 440);
1053     addIntelligentMailBarcode(page.getElements(), 0, 528);
1054     addPostnet(page.getElements(), 279, 528);
1055     addRm4sccCode(page.getElements(), 0, 616);
1056     // Add the page to the document
1057     document.getPages().add(page);
1058 }
1059
1060 private void addCodabar(Group elements, float x, float y) {
1061     addCaptionAndRectangle(elements, "Codabar Barcode", x, y, 225, 80);
1062     // Add codabar barcode to the page
1063     BarCode objBarCode = new Codabar("A1234B", x, y + 27, 48);
1064     objBarCode.setX(objBarCode.getX() + (225
1065         - objBarCode.getSymbolWidth()) / 2);
1066     elements.add(objBarCode);
```

```
1067     }
1068
1069     private void addCode128(Group elements, float x, float y) {
1070         addCaptionAndRectangle(elements, "Code 128 Barcode", x, y, 225, 80);
1071         // Add code128 barcode to the page.
1072         Barcode objBarcode = new Code128("Code 128 Barcode.", x, y + 27, 48,
1073             0.75f);
1074         objBarcode.setX(objBarcode.getX() + (225
1075             - objBarcode.getSymbolWidth()) / 2);
1076         elements.add(objBarcode);
1077     }
1078
1079     private void addCode25(Group elements, float x, float y) {
1080         addCaptionAndRectangle(elements, "Code 2 of 5 Barcode", x, y, 225, 80);
1081
1082         // Add code25 barcode to the page.
1083         Barcode objBarcode = new Code25("1234567890", x, y + 27, 48);
1084         objBarcode.setX(objBarcode.getX() + (225
1085             - objBarcode.getSymbolWidth()) / 2);
1086         elements.add(objBarcode);
1087     }
1088
1089     private void addCode39(Group elements, float x, float y) {
1090         addCaptionAndRectangle(elements, "Code 3 of 9 Barcode", x, y, 225, 80);
1091
1092         // Add code39 barcode to the page.
1093         Barcode objBarcode = new Code39("CODE 39", x, y + 27, 48);
1094         objBarcode.setX(objBarcode.getX() + (225
1095             - objBarcode.getSymbolWidth()) / 2);
1096         elements.add(objBarcode);
1097     }
1098
1099     private void addInterleaved25(Group elements, float x, float y) {
1100         addCaptionAndRectangle(elements, "Interleaved 2 of 5 Barcode", x, y,
1101             225, 80);
1102
1103         // Add interleaved25 barcode to the page.
1104         Barcode objBarcode = new Interleaved25("1234567890", x, y + 27, 48);
1105         objBarcode.setX(objBarcode.getX() + (225
1106             - objBarcode.getSymbolWidth()) / 2);
1107         elements.add(objBarcode);
1108     }
1109     private void addIntelligentMailBarcode(Group elements, float x, float y) {
1110         addCaptionAndRectangle(elements, "Intelligent Mail Barcode", x, y, 225, 80);
1111         Barcode objBarcode = new IntelligentMailBarcode("0123456709498765432101234567891", x, y + 15, true);
1112         objBarcode.setX(objBarcode.getX() + (225 - objBarcode.getSymbolWidth()) / 2);
1113         objBarcode.setY(objBarcode.getY() + (65 - objBarcode.getHeight()) / 2 - 6);
1114         elements.add(objBarcode);
1115     }
```

```
1116
1117 private void addPostnet(Group elements, float x, float y) {
1118     addCaptionAndRectangle(elements, "Postnet Barcode", x, y, 225, 80);
1119     BarCode objBarCode = new Postnet("20815470412", x, y + 15);
1120     objBarCode.setX(objBarCode.getX() + (225 - objBarCode.getSymbolWidth()) / 2);
1121     objBarCode.setY(objBarCode.getY() + (65 - objBarCode.getHeight()) / 2);
1122     elements.add(objBarCode);
1123 }
1124
1125
1126 private void addPdf417( Group elements, float x, float y ) {
1127     addCaptionAndRectangle( elements, "Pdf417 Barcode", x, y, 225, 80 );
1128     byte[] by = "ceTe Software".getBytes();
1129     Pdf417 pdfc = new Pdf417( by, x, y + 26, 2, 2 );
1130     pdfc.setX(pdfc.getX() + ( 225 - pdfc.getSymbolWidth() ) / 2);
1131     elements.add( pdfc );
1132 }
1133
1134 private void addPdf417Macro( Group elements, float x, float y ) {
1135     addCaptionAndRectangle( elements , "Pdf417 Macro Barcode:", x, y, 225, 80 );
1136     byte[] values5 = null;
1137     try {
1138         FileInputStream fi = new FileInputStream( getServletContext().getRealPath("/text/Textdata.txt"));
1139         long avail = fi.available();
1140         values5 = new byte[(int)avail];
1141         fi.read( values5, 0, (int)avail );
1142         fi.close();
1143     } catch( Exception e ) {
1144         System.out.println( "EXCEPTION "+e.getMessage() );
1145     }
1146
1147     // Create a MacroPdf417
1148     MacroPdf417 macroPdf417 = new MacroPdf417( values5, x, y + 27, 10,15, 0.9f );
1149     macroPdf417.setX(macroPdf417.getX() + ( 225 - macroPdf417.getSymbolWidth() ) / 2);
1150
1151
1152     // Add macroPdf417 to page
1153     elements.add( macroPdf417 );
1154
1155     // Retrieve the overflow barcode and place it at the appropriate location on the page
1156     macroPdf417 = macroPdf417.getOverflowMacroPdf417( x, y + 115 );
1157     macroPdf417.setX(macroPdf417.getX()+ ( 225 - macroPdf417.getSymbolWidth() ) / 2);
1158     addCaptionAndRectangle( elements, "Pdf417 Overflow Macro Barcode:", x , y += 88, 225, 80 );
1159     elements.add( macroPdf417 );
1160 }
1161
1162 private void addDataMatrixBarcode(Group elements, float x, float y) {
1163     addCaptionAndRectangle(elements, "Data Matrix Barcode:", x, y, 225, 80);
1164     String text = "DynamicPDF Generator can be used to add a Data Matrix Barcode to the PDF.";
```



```
1165
1166     // Create a data matrix barcode
1167     DataMatrixBarCode dataMatrixBarCode = new DataMatrixBarCode(text, x, y + 22, DataMatrixSymbolSize.R32xC32,
DataMatrixEncodingType.AUTO);
1168     dataMatrixBarCode.setXDimension(1.75f);
1169     dataMatrixBarCode.setX( dataMatrixBarCode.getX()+ (225 - dataMatrixBarCode.getSymbolWidth()) / 2);
1170
1171     // Add data matrix barcode to the page
1172     elements.add(dataMatrixBarCode);
1173
1174     addCaptionAndRectangle(elements, "Data Matrix Overflow Barcode:", x, y += 88, 225, 80);
1175
1176     // Retrieve the overflow barcode and place it at the appropriate location on the page
1177     dataMatrixBarCode = dataMatrixBarCode.getOverFlowDataMatrixBarcode(x, y + 22);
1178     dataMatrixBarCode.setXDimension(1.75f);
1179     dataMatrixBarCode.setX(dataMatrixBarCode.getX()+(225 - dataMatrixBarCode.getSymbolWidth()) / 2);
1180     elements.add(dataMatrixBarCode);
1181 }
1182 private void addQrCode(Group elements, float x, float y)
1183 {
1184     addCaptionAndRectangle(elements, "Qr Code", x, y, 225, 80);
1185     byte[] by = "ceTe Software".getBytes();
1186     QrCode qrCode = new QrCode(by, x, y + 20);
1187     qrCode.setXDimension(2.5f);
1188     qrCode.setX(qrCode.getX()+(225 - qrCode.getSymbolWidth()) / 2);
1189     elements.add(qrCode);
1190 }
1191 private void addUPCEANPage(Document document, Outline parentOutline) {
1192     // Increments a page counter
1193     pageCounter++;
1194
1195     // Adds to outlines to the page
1196     Outline pageOutline = parentOutline.getChildOutlines().add("Page "
1197         + pageCounter + " - UPC/EAN/JAN Bar Codes",
1198         new ZoomDestination(pageCounter,
1199             PageZoom.FIT_PAGE));
1200
1201     // Create a page to add to the document
1202     Page page = new Page(PageSize.LETTER, PageOrientation.LANDSCAPE, 54);
1203
1204     // Add the page elements to the page
1205     addEAN13(page.getElements(), 0, 0);
1206     addEAN13Sup2(page.getElements(), 240, 0);
1207     addEAN13Sup5(page.getElements(), 480, 0);
1208     addUPCVersionA(page.getElements(), 0, 135);
1209     addUPCVersionASup2(page.getElements(), 240, 135);
1210     addUPCVersionASup5(page.getElements(), 480, 135);
1211     addEAN8(page.getElements(), 0, 270);
1212     addEAN8Sup2(page.getElements(), 240, 270);
```

```
1213     addEAN8Sup5(page.getElements(), 480, 270);
1214     addUPCVersionE(page.getElements(), 0, 405);
1215     addUPCVersionESup2(page.getElements(), 240, 405);
1216     addUPCVersionESup5(page.getElements(), 480, 405);
1217
1218     // Add the page to the document
1219     document.getPages().add(page);
1220 }
1221
1222 private void addEAN13(Group elements, float x, float y) {
1223     addCaptionAndRectangle(elements, "EAN/JAN 13 Barcode", x, y, 204, 99);
1224
1225     // Add ean13 barcode to the page
1226     BarCode objBarCode = new Ean13("123456789012", x, y + 21);
1227     objBarCode.setX(objBarCode.getX() + (204
1228         - objBarCode.getSymbolWidth()) / 2);
1229     elements.add(objBarCode);
1230 }
1231
1232 private void addEAN13Sup2(Group elements, float x, float y) {
1233     addCaptionAndRectangle(elements, "EAN/JAN 13 Barcode, 2 digit " +
1234         "supplement", x, y, 204, 99);
1235     // Add ean13 2 digit supplement barcode to the page
1236     BarCode objBarCode = new Ean13Sup2("12345678901212", x, y + 21);
1237     objBarCode.setX(objBarCode.getX() + (204
1238         - objBarCode.getSymbolWidth()) / 2);
1239     elements.add(objBarCode);
1240 }
1241
1242 private void addEAN13Sup5(Group elements, float x, float y) {
1243     addCaptionAndRectangle(elements, "EAN/JAN 13 Barcode, 5 digit " +
1244         "supplement", x, y, 204, 99);
1245
1246     // Add ean13 5 digit supplement barcode to the page.
1247     BarCode objBarCode = new Ean13Sup5("12345678901212345", x, y + 21);
1248     objBarCode.setX(objBarCode.getX() + (204
1249         - objBarCode.getSymbolWidth()) / 2);
1250     elements.add(objBarCode);
1251 }
1252
1253 private void addUPCVersionA(Group elements, float x, float y) {
1254     addCaptionAndRectangle(elements, "UPC Version A Barcode", x, y, 204,
1255         99);
1256
1257     // Add upcversionA barcode to the page.
1258     BarCode objBarCode = new UpcVersionA("12345678901", x, y + 21);
1259     objBarCode.setX(objBarCode.getX() + (204
1260         - objBarCode.getSymbolWidth()) / 2);
1261     elements.add(objBarCode);
```

```
1262     }
1263
1264     private void addUPCVersionASup2(Group elements, float x, float y) {
1265         addCaptionAndRectangle(elements, "UPC Version A Barcode, 2 digit " +
1266             "supplement", x, y, 204, 99);
1267
1268         // Add upcversionA 2 digit supplement barcode to the page.
1269         BarCode objBarCode = new UpcVersionASup2("1234567890112", x, y + 21);
1270         objBarCode.setX(objBarCode.getX() + (204
1271             - objBarCode.getSymbolWidth()) / 2);
1272         elements.add(objBarCode);
1273     }
1274
1275     private void addUPCVersionASup5(Group elements, float x, float y) {
1276         addCaptionAndRectangle(elements, "UPC Version A Barcode, 5 digit " +
1277             "supplement", x, y, 204, 99);
1278
1279         // Add upcversionA 5 digit supplement barcode to the page.
1280         BarCode objBarCode = new UpcVersionASup5("1234567890112345", x, y + 21);
1281         objBarCode.setX(objBarCode.getX() + (204
1282             - objBarCode.getSymbolWidth()) / 2);
1283         elements.add(objBarCode);
1284     }
1285
1286     private void addEAN8(Group elements, float x, float y) {
1287         addCaptionAndRectangle(elements, "EAN/JAN 8 Barcode", x, y, 204, 99);
1288
1289         // Add ean8 barcode to the page.
1290         BarCode objBarCode = new Ean8("12345670", x, y + 21);
1291         objBarCode.setX(objBarCode.getX() + (204
1292             - objBarCode.getSymbolWidth()) / 2);
1293         elements.add(objBarCode);
1294     }
1295
1296     private void addEAN8Sup2(Group elements, float x, float y) {
1297         addCaptionAndRectangle(elements, "EAN/JAN 8 Barcode, 2 digit " +
1298             "supplement", x, y, 204, 99);
1299
1300         // Add ean8 2 digit supplement barcode to the page.
1301         BarCode objBarCode = new Ean8Sup2("1234567012", x, y + 21);
1302         objBarCode.setX(objBarCode.getX() + (204
1303             - objBarCode.getSymbolWidth()) / 2);
1304         elements.add(objBarCode);
1305     }
1306
1307     private void addEAN8Sup5(Group elements, float x, float y) {
1308         addCaptionAndRectangle(elements, "EAN/JAN 8 Barcode, 5 digit " +
1309             "supplement", x, y, 204, 99);
1310     }
```

```
1311 // Add ean8 5 digit supplement barcode to the page.
1312 Barcode objBarCode = new Ean8Sup5("1234567012345", x, y + 21);
1313 objBarCode.setX(objBarCode.getX() + (204
1314     - objBarCode.getSymbolWidth()) / 2);
1315 elements.add(objBarCode);
1316 }
1317
1318 private void addUPCVersionE(Group elements, float x, float y) {
1319     addCaptionAndRectangle(elements, "UPC Version E Barcode", x, y, 204,
1320         99);
1321
1322     // Add upcversionE barcode to the page.
1323     Barcode objBarCode = new UpcVersionE("0123456", x, y + 21);
1324     objBarCode.setX(objBarCode.getX() + (204
1325         - objBarCode.getSymbolWidth()) / 2);
1326     elements.add(objBarCode);
1327 }
1328
1329 private void addUPCVersionESup2(Group elements, float x, float y) {
1330     addCaptionAndRectangle(elements, "UPC Version E Barcode, 2 digit " +
1331         "supplement", x, y, 204, 99);
1332
1333     // Add upcversionE 2 digit supplement barcode to the page.
1334     Barcode objBarCode = new UpcVersionESup2("012345612", x, y + 21);
1335     objBarCode.setX(objBarCode.getX() + (204
1336         - objBarCode.getSymbolWidth()) / 2);
1337     elements.add(objBarCode);
1338 }
1339
1340 private void addUPCVersionESup5(Group elements, float x, float y) {
1341     addCaptionAndRectangle(elements, "UPC Version E Barcode, 5 digit " +
1342         " supplement", x, y, 204, 99);
1343
1344     // Add upcversionE 5 digit supplement barcode to the page.
1345     Barcode objBarCode = new UpcVersionESup5("012345612345", x, y + 21);
1346     objBarCode.setX(objBarCode.getX() + (204
1347         - objBarCode.getSymbolWidth()) / 2);
1348     elements.add(objBarCode);
1349 }
1350
1351 private void addCaptionAndRectangle(Group pageElements, String caption,
1352     float x, float y, float width,
1353     float height) {
1354     // Adds a rectangle and caption to the pageElements
1355     Rectangle objRectangle = new Rectangle(x, y + 15, width, height - 15);
1356     Label objCaption = new Label(caption, x, y, 225, 10,
1357         Font.getHelveticaBold(), 10);
1358
1359     pageElements.add(objRectangle);
```

```
1360     pageElements.add(objCaption);
1361 }
1362
1363 private void addCaptionAndRectangle(Group pageElements, String caption,
1364     float x, float y) {
1365     addCaptionAndRectangle(pageElements, caption, x, y, 225, 85);
1366 }
1367 }
```