

## 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.html.HtmlArea;
5 import com.cete.dynamicpdf.pageelements.barcoding.*;
6 import com.cete.dynamicpdf.pageelements.charting.*;
7 import com.cete.dynamicpdf.pageelements.charting.axes.*;
8 import com.cete.dynamicpdf.pageelements.charting.series.*;
9 import com.cete.dynamicpdf.pageelements.forms.*;
10 import java.io.File;
11 import java.io.FileInputStream;
12 import java.io.FileNotFoundException;
13 import java.io.IOException;
14 import java.net.URI;
15 import javax.servlet.ServletConfig;
16 import javax.servlet.ServletException;
17 import javax.servlet.ServletOutputStream;
18 import javax.servlet.http.HttpServlet;
19 import javax.servlet.http.HttpServletRequest;
20 import javax.servlet.http.HttpServletResponse;
21
22 /**
23 * A Java Servlet describing the usage of all the Page Elements.
24 */
25 public class AllPageElements extends HttpServlet {
26
27     private int pageCounter;
28
29     ServletOutputStream sOut;
30
31     public void init(ServletConfig config) throws ServletException {
32         super.init(config);
33     }
34
35     /**
36      * Custom page element.
37      */
38
39     public class StopSign extends PageElement {
40
41         private float x = 0.0f;
42         private float y = 0.0f;
43         private float w = 0.0f;
44         private float h = 0.0f;
45
46         /**
47          * Initializes a new instance of the <code>MyPageElement</code> class.
48          * @param x X coordinate of the page element.
49          * @param y Y coordinate of the page element.
50     }
```

## AllPageElements.java

---

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

## AllPageElements.java

---

```
99    }
100
101
102    public void doGet(HttpServletRequest req, HttpServletResponse res)
103        throws IOException, ServletException {
104        sOut = res.getOutputStream();
105        pageCounter = 0;
106        // Create a document and set it's properties.
107        Document document = new Document();
108        document.setCreator("AllPageElements.java");
109        document.setAuthor("ceTe Software");
110        document.setTitle("All Page Elements");
111
112        // Add the top level outline to the document
113        Outline objParentOutline = document.getOutlines()
114            .add("Top Level Outline",
115                new XYDestination(1, 0, 0));
116        objParentOutline.setColor(new WebColor("0000FF"));
117        objParentOutline.setStyle(TextStyle.BOLD);
118
119        // Add pages to the document
120        addPageElementPage(document, objParentOutline);
121        addTableAndOther(document, objParentOutline );
122        addChartsPage(document, objParentOutline);
123        addFormElementPage( document, objParentOutline );
124        addUPCEANPage(document, objParentOutline);
125        addPostalBarcodePage(document, objParentOutline);
126        addISBNISSNISMNPage(document, objParentOutline);
127        addBarcodePage(document, objParentOutline);
128        add2DBarcodePage(document, objParentOutline);
129        addWatermarkPage(document, objParentOutline);
130
131        // Outputs the document to the current web page
132        document.drawToWeb(req, res, "AllPageElements.pdf");
133
134        sOut.close();
135    }
136
137    private void addFormElementPage( Document document, Outline parentOutline ) {
138        pageCounter++;
139        parentOutline.getChildOutlines().add( "Page " + pageCounter + " - Form Page Elements", new ZoomDestination( pageCounter,
PageZoom.FIT_PAGE ) );
140
141        // Create a page to add to the document
142        com.cete.dynamicpdf.Page page = new com.cete.dynamicpdf.Page( 612, 713, 54 );
143        addTextField( page.getElements(), 0, 0 );
144        addComboBoxField( page.getElements(), 0, 93 );
145        addListField( page.getElements(), 0, 186 );
146        addRadioField( page.getElements(), 0, 279 );
```

## AllPageElements.java

---

```
147    addCheckBoxField( page.getElements(), 0, 372 );
148    addButtonField( page.getElements(), 0, 465 );
149
150    // Add the page to the document
151    document.getPages().add( page );
152 }
153
154 private void addTextField(Group pageElements, float x, float y) {
155     TextField txt = new TextField("txtfname", x + 20, y + 40, 120, 20);
156     txt.setDefaultValue( "This is a Scrollable TextField" );
157     txt.setBorderColor( RgbColor.getBlack() );
158     txt.setBackgroundColor( RgbColor.getAliceBlue() );
159     txt.setToolTip( "Scrollable" );
160     pageElements.add( txt );
161
162     TextField txt1 = new TextField("txtf1name", x + 175, y + 40, 120, 20);
163     txt1.setDefaultValue( "TextField" );
164     txt1.setPassword( true );
165     txt1.setMaxLength( 9 );
166     txt1.setBorderColor( RgbColor.getBlack() );
167     txt1.setBackgroundColor( RgbColor.getAliceBlue() );
168     txt1.setToolTip( "Password with Maximum Length" );
169     pageElements.add( txt1 );
170
171     TextField txt2 = new TextField("txtf2name", x + 330, y + 30, 150, 40);
172     txt2.setDefaultValue( "This is a TextField which goes to the next line if the text exceeds width" );
173     txt2.setMultiLine( true );
174     txt2.setBorderColor( RgbColor.getBlack() );
175     txt2.setBackgroundColor( RgbColor.getAliceBlue() );
176     txt2.setToolTip( "Multiline" );
177     pageElements.add( txt2 );
178
179     addCaptionAndRectangle( pageElements, "TextField Form Page Element:", x, y, 504, 85 );
180 }
181
182 private void addComboBoxField( Group pageElements, float x, float y ) {
183     ComboBox cb = new ComboBox("cmbname", x + 51, y + 40, 150, 20);
184     cb.setBorderColor( RgbColor.getBlack() );
185     cb.setBackgroundColor( RgbColor.getAliceBlue() );
186     cb.setFont( Font.getHelvetica() );
187     cb.setFontSize( 12 );
188     cb.getItems().add( "Item 1" );
189     cb.getItems().add( "Item 2" );
190     cb.getItems().add( "Item 3" );
191     cb.getItems().add( "Item 4" );
192     cb.getItems().add( "Editable" );
193     cb.getItems().getChoiceItem("Editable").setSelected(true);
194     cb.setEditable( true );
195     cb.setToolTip( "Editable Combo Box" );
```

## AllPageElements.java

---

```
196    pageElements.add( cb );
197
198    ComboBox cb1 = new ComboBox("cmb1name", x + 303, y + 40, 150, 20);
199    cb1.setBorderColor(RgbColor.getBlack());
200    cb1.setBackgroundColor( RgbColor.getAliceBlue() );
201    cb1.setFont(Font.getHelvetica());
202    cb1.setFontSize(12 );
203    cb1.getItems().add("Item 1");
204    cb1.getItems().add("Item 2");
205    cb1.getItems().add("Item 3");
206    cb1.getItems().add("Item 4");
207    cb1.getItems().add("Non-Editable");
208    cb1.getItems().getChoiceItem( "Non-Editable" ).setSelected(true);
209    cb1.setEditable(false);
210    cb1.setToolTip("Non-Editable Combo Box");
211    pageElements.add( cb1 );
212
213    addCaptionAndRectangle( pageElements, "ComboBox Form Page Element:", x, y, 504, 85 );
214 }
215
216 private void addListField( Group pageElements, float x, float y ) {
217     ListBox lb = new ListBox("lbname", x + 51, y + 30, 150, 40);
218     lb.setBorderColor(RgbColor.getBlack());
219     lb.setBackgroundColor( RgbColor.getAliceBlue() );
220     lb.setFont(Font.getHelvetica());
221     lb.setFontSize(12 );
222     lb.getItems().add("Item 1");
223     lb.getItems().add("Item 2");
224     lb.getItems().add("Item 3");
225     lb.getItems().add("Item 4");
226     lb.getItems().add("Non-MultiSelect");
227     lb.getItems().getChoiceItem( "Non-MultiSelect" ).setSelected(true);
228     lb.setMultiselect( false );
229     lb.setToolTip( "Simple List Box" );
230     pageElements.add( lb );
231
232     ListBox lb1 = new ListBox("lbl1name", x + 303, y + 30, 150, 40);
233     lb1.setBorderColor(RgbColor.getBlack());
234     lb1.setBackgroundColor( RgbColor.getAliceBlue() );
235     lb1.setFont(Font.getHelvetica());
236     lb1.setFontSize(12 );
237     lb1.getItems().add("Item 1");
238     lb1.getItems().add("Item 2");
239     lb1.getItems().add("Item 3");
240     lb1.getItems().add("Item 4");
241     lb1.getItems().getChoiceItem("Item 1").setSelected(true);
242     lb1.getItems().add("MultiSelect");
243     lb1.getItems().getChoiceItem("MultiSelect").setSelected(true);
244     lb1.setMultiselect( true );
```

## AllPageElements.java

---

```
245     lb1.setToolTip( "MultiSelect List Box" );
246     pageElements.add( lb1 );
247
248     addCaptionAndRectangle( pageElements, "ListBox Form Page Element:", x, y, 504, 85 );
249 }
250
251 private void addRadioField( Group pageElements, float x, float y ) {
252     RadioButton rb = new RadioButton( "rbname", x + 40, y + 42, 12, 12 );
253     rb.setChecked( true );
254     rb.setBackgroundColor( RgbColor.getAliceBlue() );
255     rb.setTextColor( RgbColor.getRed() );
256     rb.setExportValue( "Radiol" );
257     Label lb = new Label("Monthly", x + 75, y + 43, 60, 15 );
258     pageElements.add( rb );
259     pageElements.add( lb );
260
261     RadioButton rbl = new RadioButton( "rbname", x + 195, y + 42, 12, 12 );
262     rbl.setBackgroundColor(RgbColor.getAliceBlue());
263     rbl.setTextColor( RgbColor.getGreen() );
264     rbl.setExportValue( "Radio2" );
265     Label lbl = new Label("Quarterly", x + 235, y + 43, 60, 15 );
266     pageElements.add( rbl );
267     pageElements.add( lbl );
268
269     RadioButton rb2 = new RadioButton( "rbname", x + 350, y + 42, 12, 12 );
270     rb2.setBackgroundColor( RgbColor.getAliceBlue() );
271     rb2.setTextColor( RgbColor.getBlue() );
272     rb2.setExportValue( "Radio3" );
273     Label lb2 = new Label("Yearly", x + 385, y + 43, 60, 15 );
274     pageElements.add( rb2 );
275     pageElements.add( lb2 );
276
277     addCaptionAndRectangle( pageElements, "RadioButton Form Page Element:", x, y, 504, 85 );
278 }
279
280 private void addCheckBoxField( Group pageElements, float x, float y ) {
281     CheckBox cb = new CheckBox( "cbxname", x + 40, y + 42, 15, 15 );
282     cb.setSymbol(com.cete.dynamicpdf.pageelements.forms.Symbol.CHECK) ;
283     cb.setBackgroundColor( RgbColor.getAliceBlue() );
284     Label lb = new Label("Onion",x + 75, y + 43, 60, 15 );
285     pageElements.add( cb );
286     pageElements.add( lb );
287
288     CheckBox cb1 = new CheckBox( "cbxsecondname", x + 195, y + 42, 15, 15);
289     cb1.setSymbol(com.cete.dynamicpdf.pageelements.forms.Symbol.CROSS);
290     cb1.setChecked( true );
291     cb1.setBackgroundColor(RgbColor.getAliceBlue());
292     Label lbl1 = new Label("Capsicum",x + 235, y + 43, 60, 15 );
293     pageElements.add( cb1 );
```

## AllPageElements.java

---

```
294     pageElements.add( lb1 );
295
296     CheckBox cb2 = new CheckBox( "cbxthirdname", x + 350, y + 42, 15, 15);
297     cb2.setSymbol(com.cete.dynamicpdf.pageelements.forms.Symbol.STAR);
298     cb2.setBackgroundColor( RgbColor.getAliceBlue() );
299     Label lb2 = new Label("Jalapenos",x + 385, y + 43, 80, 15 );
300     pageElements.add( cb2 );
301     pageElements.add( lb2 );
302
303     addCaptionAndRectangle( pageElements, "CheckBox Form Page Element:", x, y, 504, 85 );
304 }
305
306 private void addButtonField( Group pageElements, float x, float y ) {
307     Button bt = new Button("bttonename", x + 40, y + 40, 100, 25);
308     bt.setBehavior(Behavior.PUSH);
309     bt.setBorderStyle(BorderStyle.BEVELED);
310     SubmitAction sb = new SubmitAction("http://localhost:8080/");
311     bt.setToolTip("Submit Button");
312     bt.setLabel("Submit");
313     bt.setAction(sb);
314     pageElements.add(bt);
315
316     Button bt1 = new Button( "bttwename", x + 195, y + 40, 100, 25 );
317     bt1.setBehavior( Behavior.PUSH );
318     bt1.setBorderStyle( BorderStyle.BEVELED );
319     ResetAction rb = new ResetAction();
320     bt1.setAction( rb );
321     bt1.setToolTip( "Reset Button" );
322     bt1.setLabel( "Reset" );
323     pageElements.add( bt1 );
324
325     Button bt2 = new Button( "btthreename", x + 350, y + 40, 100, 25 );
326     bt2.setBehavior( Behavior.PUSH );
327     bt2.setBorderStyle( BorderStyle.BEVELED );
328     bt2.setAction( new ZoomDestination(pageCounter, PageZoom.FIT_WIDTH ) );
329     bt2.setBehavior( Behavior.createPush("Click", "Zoom") );
330     bt2.setToolTip( "Zoom to fit page width" );
331     bt2.setLabel( "Zoom" );
332     pageElements.add( bt2 );
333
334     Button bt3 = new Button( "btfourname", x + 101, y + 90, 100, 25 );
335     bt3.setBehavior( Behavior.PUSH );
336     bt3.setBorderStyle( BorderStyle.BEVELED );
337     bt3.setAction( new XYDestination(pageCounter, 100, 100));
338     bt3.setBehavior( Behavior.createPush("Click", "Go To XY Destination") );
339     bt3.setToolTip( "Go to XY position" );
340     bt3.setLabel( "XY" );
341     pageElements.add( bt3 );
342 }
```

## AllPageElements.java

---

```
343     Button bt4 = new Button( "btfive", x + 302, y + 90, 100, 25 );
344     bt4.setBehavior( Behavior.PUSH );
345     bt4.setBorderStyle( BorderStyle.BEVELED );
346     bt4.setAction( new JavaScriptAction( "app.alert('Hello');" ) );
347     bt4.setToolTip( "Javascript Button" );
348     bt4.setLabel( "OK" );
349     pageElements.add( bt4 );
350
351     addCaptionAndRectangle( pageElements, "Button Form Page Element:", x, y, 504, 140 );
352 }
353
354 private void addListElements( Page page, float x, float y ) {
355     OrderedList orderedList = new OrderedList( x + 5, y + 20, 400, 90, Font.getHelvetica(), 10 );
356     orderedList.getListItems().add("Fruits");
357     orderedList.getListItems().add("Vegetables");
358
359     OrderedSubList orderedSubList = orderedList.getListItems().getItem(0).getSubLists().addOrderedSubList(
360         NumberingStyle.ROMAN_UPPERCASE );
361     orderedSubList.getListItems().add("Citrus");
362     orderedSubList.getListItems().add("Non-Citrus");
363
364     addCaptionAndRectangle( page.getElements(), "Ordered List Page Element:", x, y, 225, 110 );
365     OrderedSubList orderedSubList2 = orderedList.getListItems().getItem(1).getSubLists().addOrderedSubList(
366         NumberingStyle.ROMAN_UPPERCASE );
367     orderedSubList2.getListItems().add("Potato");
368     orderedSubList2.getListItems().add("Beans");
369
370     OrderedSubList subOrderedSubList = orderedSubList.getListItems().getItem(0).getSubLists().addOrderedSubList(
371         NumberingStyle.ROMAN_LOWERCASE );
372     subOrderedSubList.getListItems().add("Lime");
373     subOrderedSubList.getListItems().add("Orange");
374
375     OrderedSubList subOrderedSubList2 = orderedSubList.getListItems().getItem(1).getSubLists().addOrderedSubList(
376         NumberingStyle.ROMAN_LOWERCASE );
377     subOrderedSubList2.getListItems().add("Mango");
378     subOrderedSubList2.getListItems().add("Banana");
379
380     OrderedSubList subOrderedSubList3 = orderedSubList2.getListItems().getItem(0).getSubLists().addOrderedSubList(
381         NumberingStyle.ROMAN_LOWERCASE );
382     subOrderedSubList3.getListItems().add("Sweet Potato");
383
384     OrderedSubList subOrderedSubList4 = orderedSubList2.getListItems().getItem(1).getSubLists().addOrderedSubList(
385         NumberingStyle.ROMAN_LOWERCASE );
386     subOrderedSubList4.getListItems().add("String Bean");
387     subOrderedSubList4.getListItems().add("Lima Bean");
388     subOrderedSubList4.getListItems().add("Kidney Bean");
389
390     page.getElements().add( orderedList );
391     x += 279;
392 }
```

## AllPageElements.java

---

```
386
387     orderedList = orderedList.getOverFlowList(x + 5, y + 20);
388     addCaptionAndRectangle( page.getElements(), "Ordered List Page Element Overflow:", x, y, 225, 110 );
389     page.getElements().add( orderedList );
390
391     x = 0;
392     y += 118;
393
394     // Create an unordered list
395     UnorderedList unorderedList = new UnorderedList(x + 5, y + 20, 400, 90, Font.getHelvetica(), 10 );
396     unorderedList.getListItems().add("Fruits");
397     unorderedList.getListItems().add("Vegetables");
398
399     UnorderedSubList unorderedSubList = unorderedList.getListItems().getItem(0).getSubLists().addUnorderedSubList();
400     unorderedSubList.getListItems().add(" Citrus");
401     unorderedSubList.getListItems().add(" Non-Citrus");
402
403     addCaptionAndRectangle( page.getElements(), "Unordered List Page Element:", x , y, 225, 110 );
404
405     UnorderedSubList unorderedSubList2 = unorderedList.getListItems().getItem(1).getSubLists().addUnorderedSubList( );
406     unorderedSubList2.getListItems().add("Potato");
407     unorderedSubList2.getListItems().add("Beans");
408
409     UnorderedSubList subUnorderedSubList = unorderedSubList.getListItems().getItem(0).getSubLists().addUnorderedSubList( );
410     subUnorderedSubList.getListItems().add("Lime");
411     subUnorderedSubList.getListItems().add("Orange");
412
413     UnorderedSubList subUnorderedSubList2 = unorderedSubList.getListItems().getItem(1).getSubLists().addUnorderedSubList(
414 );
415     subUnorderedSubList2.getListItems().add("Mango");
416     subUnorderedSubList2.getListItems().add("Banana");
417
418     UnorderedSubList subUnorderedSubList3 = unorderedSubList2.getListItems().getItem(0).getSubLists().addUnorderedSubList(
419 );
420     subUnorderedSubList3.getListItems().add("Sweet Potato");
421
422     UnorderedSubList subUnorderedSubList4 = unorderedSubList2.getListItems().getItem(1).getSubLists().addUnorderedSubList(
423 );
424     subUnorderedSubList4.getListItems().add("String Bean");
425     subUnorderedSubList4.getListItems().add("Lima Bean");
426     subUnorderedSubList4.getListItems().add("Kidney Bean");
427
428     x += 279;
429     page.getElements().add( unorderedList );
430     unorderedList = unorderedList.getOverFlowList(x + 5, y + 20);
431     addCaptionAndRectangle( page.getElements(), "Unordered List Page Element Overflow:", x, y, 225, 110 );
432     page.getElements().add( unorderedList );
433 }
```

## AllPageElements.java

---

```
432     private void addPageElementPage(Document document, Outline parentOutline) {  
433         // Increments a page counter  
434         pageCounter++;  
435  
436         // Adds to outlines to the page  
437         Outline pageOutline = parentOutline.getChildOutlines().add("Page " + pageCounter);  
438  
439         pageOutline.getChildOutlines().add("Zoom - Fit Page",  
440             new ZoomDestination(pageCounter, PageZoom.FIT_PAGE));  
441         pageOutline.getChildOutlines().add("Zoom - Fit Width",  
442             new ZoomDestination(pageCounter, PageZoom.FIT_WIDTH));  
443         //create a new Page.  
444         Page page = new Page(PageSize.LETTER, PageOrientation.PORTRAIT, 54);  
445  
446         addHeader(page.getElements());  
447  
448         addBookmark(page.getElements(), 0, 41, pageOutline);  
449  
450         addCircle(page.getElements(), 279, 41);  
451  
452         addHtmlArea(page.getElements(), 0, 134);  
453  
454         addImage(page.getElements(), 0, 227);  
455  
456         addLabel(page.getElements(), 279, 227);  
457  
458         addLine(page.getElements(), 0, 320);  
459  
460         addLink(page.getElements(), 279, 320);  
461  
462         addPath(page.getElements(), 0, 413);  
463  
464         addRectangle(page.getElements(), 279, 413);  
465  
466         addTextArea(page.getElements(), 0, 506);  
467  
468         addCustom(page.getElements(), 0, 599);  
469  
470         addNote(page.getElements(), 279, 599);  
471  
472  
473  
474         // Add the page to the document  
475         document.getPages().add(page);  
476  
477  
478     }  
479  
480     private void addHeader(Group pageElements) {
```

## AllPageElements.java

---

```
481     // Adds a header to the pageElements
482     String strFormattedText = "<i>Dynamic</i><b>PDF</b>; Generator for" +
483         "Java has over 50 built-in page " +
484         "elements. Custom page elements can be built by " +
485         "creating a class that inherits from " +
486         "<b>com.cete.dynamicpdf.PageElement</b>.";
487     //create a FormattedTextArea which can contain text with Formatted tags.
488     FormattedTextArea formattedTextArea = new FormattedTextArea(strFormattedText, 0, 0, 504,
489         450, FontFamily.getHelvetica(), 12,
490         false);
491     formattedTextArea.getStyle().paragraph.align = TextAlign.CENTER;
492
493     pageElements.add(formattedTextArea);
494 }
495
496 private void addBookmark(Group pageElements, float x, float y,
497     Outline parentOutline) {
498
499     addCaptionAndRectangle(pageElements, "Bookmark Page Element:", x, y);
500
501     // Adds a bookmark to the page
502     pageElements.add(new Bookmark("Bookmarked Text", x + 5, y + 20,
503         parentOutline));
504
505     pageElements.add(new Label("This text is bookmarked.", x + 5, y + 20,
506         215, 10, Font.getTimesRoman(), 10));
507 }
508
509 private void addCircle(Group pageElements, float x, float y) {
510     addCaptionAndRectangle(pageElements, "Circle Page Element:", x, y);
511
512     // Adds a circle to the pageElements
513     pageElements.add(new Circle(x + 112.5f, y + 50f, 107.5f, 30f,
514         RgbColor.getRed(), RgbColor.getBlue(), 2,
515         LineStyle.getDashLarge()));
516 }
517
518 private void addHtmlArea(Group pageElements, float x, float y) {
519     // Adds a html area to the pageElements
520     URI uri = null;
521     File file = new File(getServletContext().getRealPath("html" + "/AllPageElementSample.html"));
522     uri = file.toURI();
523
524     HtmlArea htmlArea = new HtmlArea(uri, x + 5, y + 20, 215, 70);
525
526     // Sets the indent property
527     addCaptionAndRectangle(pageElements, "HtmlArea Page Element", x, y);
528     addCaptionAndRectangle(pageElements, "HtmlArea Overflow Text", x + 279, y);
```

## AllPageElements.java

---

```
530    pageElements.add(htmlArea);
531
532    // Set the html area object equal to the rest of the html area that did not fit
533    // if all the html area did fit, GetOverflowHtmlArea will return null
534    HtmlArea overflowHtmlArea = htmlArea.getOverflowHtmlArea(x + 284, y + 20, 70);
535    pageElements.add(overflowHtmlArea);
536 }
537
538 private void addImage(Group pageElements, float x, float y) {
539    // Adds an image to the pageElements
540    addCaptionAndRectangle(pageElements, "Image Page Element:", x, y);
541    Image objImage = null;
542    try {
543        objImage = new Image(getServletContext().getRealPath("images" +
544                           "/DPDFLogo.png"), x + 112.5f, y + 50f, 0.24f);
545
546    } catch (FileNotFoundException ex1) {
547        System.err.println("Image File Not Found: "+ex1);
548    }
549    // Image is sized and centered in the rectangle
550    objImage.setBounds(215, 60);
551    objImage.setVAlign(VAlign.CENTER);
552    objImage.setAlign(Align.CENTER);
553
554    pageElements.add(objImage);
555 }
556
557 private void addLabel(Group pageElements, float x, float y)      {
558    // Adds a label to the pageElements
559    addCaptionAndRectangle(pageElements, "Label and PageNumberingLabel Page Elements:", x, y);
560
561    String labelText = "Labels can be rotated";
562    String numberText = "PageNumberingLabels contain page numbering: %%CP%% of %%TP%% pages.";
563
564    Label label = new Label(labelText, x + 5, y + 12, 220, 80,
565                           Font.getTimesRoman(), 12, TextAlign.CENTER);
566    label.setAngle(8);
567
568    PageNumberingLabel pageNumLabel = new PageNumberingLabel(numberText,
569                  x + 5, y + 55, 220, 80, Font.getTimesRoman(), 12, TextAlign.CENTER);
570
571    pageElements.add(pageNumLabel);
572    pageElements.add(label);
573 }
574
575 private void addLine(Group pageElements, float x, float y)      {
576    // Adds a line to the pageElements
577    addCaptionAndRectangle(pageElements, "Line Page Element:", x, y);
578 }
```

## AllPageElements.java

---

```
579     pageElements.add(new Line(x + 5, y + 20, x + 220, y + 80, 3,
580                         RgbColor.getGreen()));
581     pageElements.add(new Line(x + 220, y + 20, x + 5, y + 80, 3,
582                         RgbColor.getGreen()));
583 }
584
585 private void addLink(Group pageElements, float x, float y) {
586     // Adds a link to the pageElements
587     Font objFont = Font.getTimesRoman();
588     String strText = "This is a link to DynamicPDF.com.";
589
590     addCaptionAndRectangle(pageElements, "Link Page Element:", x, y);
591
592     Label objLabel = new Label(strText, x + 5, y + 20, 215, 80, objFont, 12,
593                               RgbColor.getBlue());
594     objLabel.setUnderline(true);
595
596     Link objLink = new Link(x + 5, y + 20, objFont.getTextWidth(strText, 12),
597                             12 - objFont.getDescender(12),
598                             new UrlAction("http://www.dynamicpdf.com"));
599     pageElements.add(objLabel);
600     pageElements.add(objLink);
601 }
602
603 private void addPath(Group pageElements, float x, float y) {
604     // Adds a path to the pageElements
605     Path objPath = new Path(x + 5, y + 20, RgbColor.getBlue(), RgbColor.getRed(),
606                            2, LineStyle.getSolid(), true);
607     objPath.getSubPaths().add(new LineSubPath(x + 215, y + 40));
608     objPath.getSubPaths().add(new CurveToSubPath(x + 108, y + 80, x + 160,
609                                              y + 80));
610     objPath.getSubPaths().add(new CurveSubPath(x + 5, y + 40, x + 65, y + 80,
611                                              x + 5, y + 60));
612
613     addCaptionAndRectangle(pageElements, "Path Page Element:", x, y);
614
615     pageElements.add(objPath);
616 }
617
618 private void addRectangle(Group pageElements, float x, float y) {
619     // Adds a rectangle to the pageElements
620     addCaptionAndRectangle(pageElements, "Rectangle Page Element:", x, y);
621
622     pageElements.add(new Rectangle(x + 5, y + 20, 215, 60, RgbColor.getGreen(),
623                                   RgbColor.getBlue(), 2));
624 }
625
626 private void addTextArea(Group pageElements, float x, float y) {
627     // Adds a text area to the pageElements
```

## AllPageElements.java

---

```
628     String strText = "The TextArea page element is great for efficient " +
629         "processing of text that does not require advanced " +
630         "formatting. It offers basic control over leading " +
631         "and paragraph spacing and include text continuation" +
632         " functionality. If all the text does not fit within " +
633         "the bounds of the TextArea the hasOverflowText " +
634         "method will return a true value. The " +
635         "getOverflowTextArea method can then be called to " +
636         "create a new TextArea with the remaining text. This " +
637         "functionality makes it simple to span large amounts of " +
638         "text over multiple TextAreas and pages. " +
639         "Text can also be rotated.";
640
641     addCaptionAndRectangle(pageElements, "TextArea Page Element:", x, y);
642     addCaptionAndRectangle(pageElements, "TextArea Overflow Text:",
643         x + 279, y);
644
645     TextArea objTextArea = new TextArea(strText, x + 5, y + 20, 215, 60,
646         Font.getTimesRoman(), 10);
647     // Create an overflow text area for the overflow text
648     TextArea objOverflowTextArea = objTextArea.getOverflowTextArea(x + 284,
649         y + 20);
650
651     pageElements.add(objTextArea);
652     pageElements.add(objOverflowTextArea);
653 }
654
655 private void addCustom(Group pageElements, float x, float y) {
656     // Adds a custom page element to the page
657     addCaptionAndRectangle(pageElements, "Custom Page Element:", x, y);
658
659     pageElements.add(new StopSign(80, y + 20, 60, 60));
660 }
661
662 private void addNote(Group pageElements, float x, float y) {
663     // Add a note element to the page
664     addCaptionAndRectangle(pageElements, "Note Page Element:", x, y);
665
666     pageElements.add(new Note("This is a note", x + 100, y + 35, 50, 50,
667         NoteType.NOTE));
668 }
669
670 private void addTableAndOther(Document document, Outline parentOutline) {
671     // Increments a page counter
672     pageCounter++;
673
674     // Adds to outlines to the page
675     Outline pageOutline = parentOutline.getChildOutlines().add("Page "
676
```

## AllPageElements.java

---

```
677             + pageCounter + " - Table and Additional Page Elements",
678             new ZoomDestination(pageCounter, PageZoom.FIT_PAGE));
679
680         // Create a page to add to the document
681         Page page = new Page(612, 729, 54);
682
683         // Add the Table and other elements to the page
684         addTable(page.getElements(), 0, 0);
685         addListElements( page, 0, 205 );
686         addAdditional(page.getElements(), 0, 441);
687
688         // Add the page to the document
689         document.getPages().add(page);
690     }
691
692     private void addTable(Group pageElements, float x, float y) {
693         Label captionLabel1 = new Label("Table Page Element:", x, y, 225, 10,
694             Font.getHelveticaBold(), 10);
695         pageElements.add(captionLabel1);
696
697         Label captionLabel2 = new Label("Table Row Overflow:", x + 279, y,
698             225, 10, Font.getHelveticaBold(), 10);
699         pageElements.add(captionLabel2);
700
701         Table2 table = new Table2(x, y + 15, 225, 190, Font.getTimesRoman(), 10);
702         table.setRepeatColumnHeaderCount( 1 );
703
704         // Add columns to the table
705         table.getColumns().add(95);
706         table.getColumns().add(65);
707         table.getColumns().add(65);
708
709         // Add the Header Row to the table
710         Row2 row1 = table.getRows().add(40, Font.getHelveticaBold(), 12,
711             Grayscale.getBlack(), RgbColor.getLightBlue());
712         row1.getCellDefault().setAlign(TextAlign.CENTER);
713         row1.getCellDefault().setVAlign(VAlign.CENTER);
714         row1.getCells().add("Header 1");
715         row1.getCells().add("Header 2");
716         row1.getCells().add("Header 3");
717         row1.getCells().add("Header 4");
718         row1.getCells().add("Header 5");
719         row1.getCells().add("Header 6");
720         row1.getCells().add( "Header 7" );
721         row1.getCells().add( "Header 8" );
722         row1.getCells().add( "Header 9" );
723         row1.getCells().add( "Header 10" );
724         row1.getCells().add( "Header 11" );
725         row1.getCells().add( "Header 12" );
```

## AllPageElements.java

---

```
726     row1.getCells().add( "Header 13" );
727     row1.getCells().add( "Header 14" );
728     row1.getCells().add( "Header 15" );
729     row1.getCells().add( "Header 16" );
730
731     int cell = 0;
732     // Add the rest of the rows to the table
733     addRow(table, cell, 1);
734     addRow(table, cell, 2);
735     addRow(table, cell, 3);
736     addRow(table, cell, 4);
737     addRow(table, cell, 5);
738     addRow(table, cell, 6);
739     addRow(table, cell, 7);
740     addRow(table, cell, 8);
741     addRow(table, cell, 9);
742     addRow(table, cell, 10);
743     addRow(table, cell, 11);
744     addRow(table, cell, 12);
745     addRow(table, cell, 13);
746     addRow(table, cell, 14);
747     addRow(table, cell, 15);
748     addRow(table, cell, 16);
749
750     // Create an overflow table for the overflow columns
751     Table2 overflowTable = table.getOverflowRows(x + 279, y + 15);
752
753     pageElements.add(table);
754     pageElements.add(overflowTable);
755 }
756
757 private void addRow(Table2 table, int cellNum, int rowNum) {
758     Row2 row = table.getRows().add();
759     Cell2 cell = row.getCells().add("RowHeader " + rowNum,
760         Font.getHelveticaBold(), 12,
761         Grayscale.getBlack(), RgbColor.getLightPink(), 1);
762     cell.setAlign(TextAlign.CENTER);
763     cell.setVAlign(Valign.CENTER);
764
765     row.getCells().add("Item " + cellNum++);
766     row.getCells().add("Item " + cellNum++);
767 }
768
769 private void addAdditional(Group pageElements, float x, float y) {
770     // Adds text explaining the additional page elements
771     String strText = "An <b><u>AreaGroup</u></b> represents a group of page elements with an area.<p/>" +
772         "A <b><u>BackgroundImage</u></b> page element is used to add an image to the background of a page. <p/>" +
773         "The <b><u>Group</u></b> page element is used for grouping page elements together.<p/>" +
774         "The <b><u>LayoutGrid</u></b> draws a coordinate grid to the page.<p/>" +
```

## AllPageElements.java

---

```
775             "A <b><u>TransformationGroup</u></b> can rotate and scale a group of page elements.<p/>" +
776             "A <b><u>TransparencyGroup</u></b> can be used to add transparency to page elements.<p/>" +
777             "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/>";
778             FormattedTextArea objFormattedTextArea = new FormattedTextArea(strText, x + 5, y + 20,
779                             494, 180, FontFamily.getTimes(),
780                             12, false);
781             objFormattedTextArea.getStyle().paragraph.spacingAfter = 5;
782
783             addCaptionAndRectangle(pageElements, "Additional Page Elements:", x, y, 504, 180);
784
785             pageElements.add(objFormattedTextArea);
786         }
787
788     private void addChartsPage(Document document, Outline parentOutline) {
789         // Increments a page counter
790         pageCounter++;
791
792         // Adds outlines to the page
793         parentOutline.getChildOutlines().add("Page " + pageCounter + " Charts", new ZoomDestination(pageCounter,
PageZoom.FIT_PAGE));
794
795         // Create a page
796         Page page = new Page(PageSize.LETTER, PageOrientation.LANDSCAPE);
797         addAreaChart(page.getElements(), 0, 25);
798         addLineChart(page.getElements(), 375, 25);
799         addPieChart(page.getElements(), 0, 260);
800         addBarChart(page.getElements(), 375, 260);
801         document.getPages().add(page);
802     }
803
804     private void addAreaChart(Group elements, float x, float y) {
805         addCaptionAndRectangle(elements, "Stacked Area Chart", x, y, 325, 225);
806
807         // Create a chart
808         Chart chart = new Chart(x + 20, y + 25, 300, 200);
809
810         // Create a plot area
811         PlotArea plotArea = chart.getPrimaryPlotArea();
812
813         // Create header title and add it to the chart
814         Title title1 = new Title("Website Visitors");
815         chart.getHeaderTitles().add(title1);
816
817         // Create indexed stacked area series elements and add values to it
818         IndexedStackedAreaSeriesElement seriesElement1 = new IndexedStackedAreaSeriesElement("Website A");
819         seriesElement1.getValues().add(new float[] { 5, 7, 9, 6 });
820         IndexedStackedAreaSeriesElement seriesElement2 = new IndexedStackedAreaSeriesElement("Website B");
821         seriesElement2.getValues().add(new float[] { 4, 2, 5, 8 });
```

## AllPageElements.java

---

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

## AllPageElements.java

---

```
871 // Create a indexed line series and add values to it
872 IndexedLineSeries lineSeries1 = new IndexedLineSeries("Website A");
873 lineSeries1.getValues().add(new float[] { 5, 7, 9, 6 });
874 IndexedLineSeries lineSeries2 = new IndexedLineSeries("Website B");
875 lineSeries2.getValues().add(new float[] { 4, 2, 5, 8 });
876 IndexedLineSeries lineSeries3 = new IndexedLineSeries("Website C");
877 lineSeries3.getValues().add(new float[] { 2, 4, 6, 9 });
878
879 // Add indexed line series to the plot area
880 plotArea.getSeries().add(lineSeries1);
881 plotArea.getSeries().add(lineSeries2);
882 plotArea.getSeries().add(lineSeries3);
883
884 // Create a title and add it to the yaxis
885 Title lTitle = new Title("Visitors (in millions)");
886 lineSeries1.getYAxis().getTitles().add(lTitle);
887
888 //Adding AxisLabels to the XAxis
889 lineSeries1.getXAxis().getLabels().add(new IndexedXAxisLabel("Q1", 0));
890 lineSeries1.getXAxis().getLabels().add(new IndexedXAxisLabel("Q2", 1));
891 lineSeries1.getXAxis().getLabels().add(new IndexedXAxisLabel("Q3", 2));
892 lineSeries1.getXAxis().getLabels().add(new IndexedXAxisLabel("Q4", 3));
893 elements.add(chart);
894 }
895
896 private void addPieChart(Group elements, float x, float y) {
897     addCaptionAndRectangle(elements, "Pie Chart", x, y, 325, 225);
898
899 // Create a chart
900 Chart chart = new Chart(x + 20, y + 25, 300, 200);
901
902 // Add a plot area to the chart
903 PlotArea plotArea = chart.getPlotAreas().add(50, 50, 300, 300);
904
905 // Create the Header titles and add it to the chart
906 Title tTitle = new Title("Website Visitors (in millions)");
907
908 chart.getHeaderTitles().add(tTitle);
909
910 // Create autogradient and add it to the series
911 AutoGradient autogradient1 = new AutoGradient(90f, CmykColor.getRed(), CmykColor.getIndianRed());
912 AutoGradient autogradient2 = new AutoGradient(90f, CmykColor.getGreen(), CmykColor.getYellowGreen());
913 AutoGradient autogradient3 = new AutoGradient(120f, CmykColor.getBlue(), CmykColor.getLightBlue());
914
915 // Create a scalar datalabel
916 ScalarDataLabel da = new ScalarDataLabel(true, false, false);
917
918 // Create a pie series
919 PieSeries pieSeries = new PieSeries();
```

```
920
921     // Set scalar dataLabel to the pie series
922     pieSeries.setDataLabel(da);
923
924     // Add series to the plot area
925     plotArea.getSeries().add(pieSeries);
926
927     //Add pie series elements to the pie series
928     pieSeries.getElements().add(27, "Website A");
929     pieSeries.getElements().add(19, "Website B");
930     pieSeries.getElements().add(21, "Website C");
931     pieSeries.getElements().getElement(0).setColor(autogradient1);
932     pieSeries.getElements().getElement(1).setColor(autogradient2);
933     pieSeries.getElements().getElement(2).setColor(autogradient3);
934     elements.add(chart);
935 }
936
937 private void addBarChart(Group elements, float x, float y) {
938     addCaptionAndRectangle(elements, "Stacked Bar Chart", x, y, 325, 225);
939
940     // Create a chart
941     Chart chart = new Chart(x + 20, y + 25, 300, 200);
942
943     // Create a plot area
944     PlotArea plotArea = chart.getPrimaryPlotArea();
945
946     // Create header titles and add it to the chart
947     Title title1 = new Title("Website Visitors");
948
949     chart.getHeaderTitles().add(title1);
950
951
952     // Create a indexed stacked bar series elements and add values to it
953     IndexedStackedBarSeriesElement seriesElement1 = new IndexedStackedBarSeriesElement("Website A");
954     seriesElement1.getValues().add(new float[] { 5, 7, 9, 6 });
955     IndexedStackedBarSeriesElement seriesElement2 = new IndexedStackedBarSeriesElement("Website B");
956     seriesElement2.getValues().add(new float[] { 4, 2, 5, 8 });
957     IndexedStackedBarSeriesElement seriesElement3 = new IndexedStackedBarSeriesElement("Website C");
958     seriesElement3.getValues().add(new float[] { 2, 4, 6, 9 });
959
960     // Create a Indexed Stacked Bar Series
961     IndexedStackedBarSeries barSeries = new IndexedStackedBarSeries();
962
963     // Create autogradient and add it to the series
964     AutoGradient autogradient1 = new AutoGradient(90f, CmykColor.getRed(), CmykColor.getIndianRed());
965     AutoGradient autogradient2 = new AutoGradient(90f, CmykColor.getGreen(), CmykColor.getYellowGreen());
966     AutoGradient autogradient3 = new AutoGradient(90f, CmykColor.getBlue(), CmykColor.getLightBlue());
967     seriesElement1.setColor(autogradient1);
968     seriesElement2.setColor(autogradient2);
```

## AllPageElements.java

---

```
969     seriesElement3.setColor(autogradient3);
970
971     // Add indexed stacked bar series elements to the Indexed Stacked Bar Series
972     barSeries.add(seriesElement1);
973     barSeries.add(seriesElement2);
974     barSeries.add(seriesElement3);
975
976     //Add series to the plot area
977     plotArea.getSeries().add(barSeries);
978
979     // Create a title and add it to the xaxis
980     Title lTitle = new Title("Visitors (in millions)");
981     barSeries.getXAxis().getTitles().add(lTitle);
982
983     //Adding AxisLabels to the yaxis
984     barSeries.getYAxis().getLabels().add(new IndexedYAxisLabel("Q1", 0));
985     barSeries.getYAxis().getLabels().add(new IndexedYAxisLabel("Q2", 1));
986     barSeries.getYAxis().getLabels().add(new IndexedYAxisLabel("Q3", 2));
987     barSeries.getYAxis().getLabels().add(new IndexedYAxisLabel("Q4", 3));
988     barSeries.getXAxis().getLabels().setAngle(-65);
989     elements.add(chart);
990 }
991
992 private void addCodabar(Group elements, float x, float y) {
993     addCaptionAndRectangle(elements, "Codabar Bar Code", x, y, 225, 80);
994     // Add codabar barcode to the page
995     BarCode objBarCode = new Codabar("A1234B", x, y + 27, 48);
996     objBarCode.setX(objBarCode.getX() + (225
997         - objBarCode.getSymbolWidth()) / 2);
998     elements.add(objBarCode);
999 }
1000
1001 private void addCode128(Group elements, float x, float y) {
1002     addCaptionAndRectangle(elements, "Code 128 Bar Code", x, y, 225, 80);
1003     // Add code128 barcode to the page.
1004     BarCode objBarCode = new Code128("Code 128 Bar Code.", x, y + 27, 48,
1005         0.75f);
1006     objBarCode.setX(objBarCode.getX() + (225
1007         - objBarCode.getSymbolWidth()) / 2);
1008     elements.add(objBarCode);
1009 }
1010
1011 private void addCode25(Group elements, float x, float y) {
1012     addCaptionAndRectangle(elements, "Code 2 of 5 Bar Code", x, y, 225, 80);
1013
1014     // Add code25 barcode to the page.
1015     BarCode objBarCode = new Code25("1234567890", x, y + 27, 48);
1016     objBarCode.setX(objBarCode.getX() + (225
1017         - objBarCode.getSymbolWidth()) / 2);
```

## AllPageElements.java

---

```
1018     elements.add(objBarcode);
1019 }
1020
1021 private void addCode39(Group elements, float x, float y) {
1022     addCaptionAndRectangle(elements, "Code 3 of 9 Bar Code", x, y, 225, 80);
1023
1024     // Add code39 barcode to the page.
1025     BarCode objBarcode = new Code39("CODE 39", x, y + 27, 48);
1026     objBarcode.setX(objBarcode.getX() + (225
1027         - objBarcode.getSymbolWidth()) / 2);
1028     elements.add(objBarcode);
1029 }
1030
1031 private void addInterleaved25(Group elements, float x, float y) {
1032     addCaptionAndRectangle(elements, "Interleaved 2 of 5 Bar Code", x, y,
1033         225, 80);
1034
1035     // Add interleaved25 barcode to the page.
1036     BarCode objBarcode = new Interleaved25("1234567890", x, y + 27, 48);
1037     objBarcode.setX(objBarcode.getX() + (225
1038         - objBarcode.getSymbolWidth()) / 2);
1039     elements.add(objBarcode);
1040 }
1041 private void addIntelligentMailBarcode(Group elements, float x, float y) {
1042     addCaptionAndRectangle(elements, "Intelligent Mail Barcode", x, y, 225, 80);
1043     BarCode objBarcode = new IntelligentMailBarcode("0123456709498765432101234567891", x, y + 15, true);
1044     objBarcode.setHeight(35);
1045     objBarcode.setX(objBarcode.getX() + (225 - objBarcode.getSymbolWidth()) / 2);
1046     objBarcode.setY(objBarcode.getY() + (65 - objBarcode.getHeight()) / 2);
1047     elements.add(objBarcode);
1048 }
1049
1050 private void addPostnet(Group elements, float x, float y) {
1051     addCaptionAndRectangle(elements, "Postnet Bar Code", x, y, 225, 55);
1052
1053     // Add postnet barcode to the page.
1054     BarCode objBarcode = new Postnet("20815470412", x, y + 25);
1055     objBarcode.setX(objBarcode.getX() + (225
1056         - objBarcode.getSymbolWidth()) / 2);
1057     elements.add(objBarcode);
1058 }
1059
1060 private void addPdf417( Group elements, float x, float y ) {
1061     addCaptionAndRectangle( elements, "Pdf417 Barcode", x, y, 225, 100 );
1062     byte[] by = "PDF417 Barcode".getBytes();
1063     Pdf417 pdfc = new Pdf417( by, x, y + 15, 2, 2 );
1064     pdfc.setX(pdfc.getX() + ( 225 - pdfc.getSymbolWidth() ) / 2);
1065     pdfc.setY(pdfc.getY() + ( 85 - pdfc.getSymbolHeight() ) / 2);
1066     elements.add( pdfc );
```

## AllPageElements.java

---

```
1067    }
1068
1069    private void addPdf417Macro( Group elements, float x, float y ) {
1070        addCaptionAndRectangle( elements , "Pdf417 Macro Bar Code:", x, y, 225, 100 );
1071        byte[] values5 = null;
1072        try {
1073            FileInputStream fi = new FileInputStream( getServletContext().getRealPath("text/Textdata.txt") );
1074            long avail = fi.available();
1075            values5 = new byte[(int)avail];
1076            fi.read( values5, 0, (int)avail );
1077            fi.close();
1078        } catch( Exception e ) {
1079            System.out.println( "EXCEPTION " +e.getMessage() );
1080        }
1081
1082        // Create a MacroPdf417
1083        MacroPdf417 macroPdf417 = new MacroPdf417( values5, x, y + 15, 10, 15, 0.9f );
1084        macroPdf417.setX(macroPdf417.getX() + ( 225 - macroPdf417.getSymbolWidth() ) / 2);
1085        macroPdf417.setY(macroPdf417.getY() + ( 85 - macroPdf417.getSymbolHeight() ) / 2);
1086
1087        // Add macroPdf417 to page
1088        elements.add( macroPdf417 );
1089        addCaptionAndRectangle( elements, "Pdf417 Overflow Macro Barcode:", x , y += 108, 225, 100 );
1090        // Retrieve the overflow barcode and place it at the appropriate location on the page
1091        macroPdf417 = macroPdf417.getOverflowMacroPdf417( x, y + 15 );
1092        macroPdf417.setX(macroPdf417.getX()+( 225 - macroPdf417.getSymbolWidth() ) / 2);
1093        macroPdf417.setY(macroPdf417.getY()+(85 - macroPdf417.getSymbolHeight() ) / 2);
1094
1095        elements.add( macroPdf417 );
1096    }
1097
1098    private void addDataMatrixBarcode(Group elements, float x, float y) {
1099        addCaptionAndRectangle(elements, "Data Matrix Barcode:", x, y, 225, 100);
1100        String text = "DynamicPDF Generator can be used to add a Data Matrix Barcode to the PDF.";
1101
1102        // Create a data matrix barcode
1103        DataMatrixBarCode dataMatrixBarCode = new DataMatrixBarCode(text, x, y + 15, DataMatrixSymbolSize.R32xC32,
1104        DataMatrixEncodingType.AUTO);
1105        //dataMatrixBarCode.setXDimension(1.75f);
1106        dataMatrixBarCode.setX( dataMatrixBarCode.getX()+( 225 - dataMatrixBarCode.getSymbolWidth() ) / 2);
1107        dataMatrixBarCode.setY( dataMatrixBarCode.getY()+( 85 - dataMatrixBarCode.getSymbolHeight() ) / 2);
1108        // Add data matrix barcode to the page
1109        elements.add(dataMatrixBarCode);
1110
1111        addCaptionAndRectangle(elements, "Data Matrix Overflow Barcode:", x, y += 108, 225, 100);
1112
1113        // Retrieve the overflow barcode and place it at the appropriate location on the page
1114        dataMatrixBarCode = dataMatrixBarCode.getOverflowDataMatrixBarcode(x, y + 15);
1115        //dataMatrixBarCode.setXDimension(1.75f);
```

## AllPageElements.java

---

```
1115     dataMatrixBarCode.setX(dataMatrixBarCode.getX()+(225 - dataMatrixBarCode.getSymbolWidth()) / 2);
1116     dataMatrixBarCode.setY(dataMatrixBarCode.getY()+(85 - dataMatrixBarCode.getSymbolHeight()) / 2);
1117     elements.add(dataMatrixBarCode);
1118 }
1119
1120 private void addQrCode(Group elements, float x, float y) {
1121     addCaptionAndRectangle(elements, "Qr Code", x, y, 225, 100);
1122     byte[] by = "ceTe Software".getBytes();
1123     QrCode qrCode = new QrCode(by, x, y + 15);
1124     qrCode.setxDimension(2.5f);
1125     qrCode.setX(qrCode.getX() + (225 - qrCode.getSymbolWidth()) / 2);
1126     qrCode.setY(qrCode.getY() + (85 - qrCode.getSymbolHeight()) / 2);
1127     elements.add(qrCode);
1128 }
1129
1130 private void addUPCEANPage(Document document, Outline parentOutline) {
1131     // Increments a page counter
1132     pageCounter++;
1133
1134     // Adds to outlines to the page
1135     Outline pageOutline = parentOutline.getChildOutlines().add("Page "
1136         + pageCounter + " - UPC/EAN/JAN Bar Codes",
1137         new ZoomDestination(pageCounter,
1138             PageZoom.FIT_PAGE));
1139
1140     // Create a page to add to the document
1141     Page page = new Page(PageSize.LETTER, PageOrientation.LANDSCAPE, 54);
1142
1143     // Add the page elements to the page
1144     addEAN13(page.getElements(), 0, 0);
1145     addEAN13Sup2(page.getElements(), 240, 0);
1146     addEAN13Sup5(page.getElements(), 480, 0);
1147     addUPCVersionA(page.getElements(), 0, 135);
1148     addUPCVersionASup2(page.getElements(), 240, 135);
1149     addUPCVersionASup5(page.getElements(), 480, 135);
1150     addEAN8(page.getElements(), 0, 270);
1151     addEAN8Sup2(page.getElements(), 240, 270);
1152     addEAN8Sup5(page.getElements(), 480, 270);
1153     addUPCVersionE(page.getElements(), 0, 405);
1154     addUPCVersionESup2(page.getElements(), 240, 405);
1155     addUPCVersionESup5(page.getElements(), 480, 405);
1156
1157     // Add the page to the document
1158     document.getPages().add(page);
1159 }
1160
1161 private void addEAN13(Group elements, float x, float y) {
1162     addCaptionAndRectangle(elements, "EAN/JAN 13 Bar Code", x, y, 204, 99);
1163 }
```

## AllPageElements.java

---

```
1164     // Add ean13 barcode to the page
1165     BarCode objBarCode = new Ean13("123456789012", x, y + 21);
1166     objBarCode.setX(objBarCode.getX() + (204
1167         - objBarCode.getSymbolWidth()) / 2);
1168     elements.add(objBarCode);
1169 }
1170
1171 private void addEAN13Sup2(Group elements, float x, float y) {
1172     addCaptionAndRectangle(elements, "EAN/JAN 13 Bar Code, 2 digit " +
1173         "supplement", x, y, 204, 99);
1174     // Add ean13 2 digit supplement bar code to the page
1175     BarCode objBarCode = new Ean13Sup2("12345678901212", x, y + 21);
1176     objBarCode.setX(objBarCode.getX() + (204
1177         - objBarCode.getSymbolWidth()) / 2);
1178     elements.add(objBarCode);
1179 }
1180
1181 private void addEAN13Sup5(Group elements, float x, float y) {
1182     addCaptionAndRectangle(elements, "EAN/JAN 13 Bar Code, 5 digit " +
1183         "supplement", x, y, 204, 99);
1184
1185     // Add ean13 5 digit supplement bar code to the page.
1186     BarCode objBarCode = new Ean13Sup5("12345678901212345", x, y + 21);
1187     objBarCode.setX(objBarCode.getX() + (204
1188         - objBarCode.getSymbolWidth()) / 2);
1189     elements.add(objBarCode);
1190 }
1191
1192 private void addUPCVersionA(Group elements, float x, float y) {
1193     addCaptionAndRectangle(elements, "UPC Version A Bar Code", x, y, 204,
1194         99);
1195
1196     // Add upcversionA bar code to the page.
1197     BarCode objBarCode = new UpcVersionA("12345678901", x, y + 21);
1198     objBarCode.setX(objBarCode.getX() + (204
1199         - objBarCode.getSymbolWidth()) / 2);
1200     elements.add(objBarCode);
1201 }
1202
1203 private void addUPCVersionASup2(Group elements, float x, float y) {
1204     addCaptionAndRectangle(elements, "UPC Version A Bar Code, 2 digit " +
1205         "supplement", x, y, 204, 99);
1206
1207     // Add upcversionA 2 digit supplement bar code to the page.
1208     BarCode objBarCode = new UpcVersionASup2("1234567890112", x, y + 21);
1209     objBarCode.setX(objBarCode.getX() + (204
1210         - objBarCode.getSymbolWidth()) / 2);
1211     elements.add(objBarCode);
1212 }
```

## AllPageElements.java

---

```
1213
1214     private void addUPCVersionASup5(Group elements, float x, float y) {
1215         addCaptionAndRectangle(elements, "UPC Version A Bar Code, 5 digit " +
1216             "supplement", x, y, 204, 99);
1217
1218         // Add upcversionA 5 digit supplement bar code to the page.
1219         BarCode objBarCode = new UpcVersionASup5("1234567890112345", x, y + 21);
1220         objBarCode.setX(objBarCode.getX() + (204
1221             - objBarCode.getSymbolWidth()) / 2);
1222         elements.add(objBarCode);
1223     }
1224
1225     private void addEAN8(Group elements, float x, float y) {
1226         addCaptionAndRectangle(elements, "EAN/JAN 8 Bar Code", x, y, 204, 99);
1227
1228         // Add ean8 bar code to the page.
1229         BarCode objBarCode = new Ean8("12345670", x, y + 21);
1230         objBarCode.setX(objBarCode.getX() + (204
1231             - objBarCode.getSymbolWidth()) / 2);
1232         elements.add(objBarCode);
1233     }
1234
1235     private void addEAN8Sup2(Group elements, float x, float y) {
1236         addCaptionAndRectangle(elements, "EAN/JAN 8 Bar Code, 2 digit " +
1237             "supplement", x, y, 204, 99);
1238
1239         // Add ean8 2 digit supplement bar code to the page.
1240         BarCode objBarCode = new Ean8Sup2("1234567012", x, y + 21);
1241         objBarCode.setX(objBarCode.getX() + (204
1242             - objBarCode.getSymbolWidth()) / 2);
1243         elements.add(objBarCode);
1244     }
1245
1246     private void addEAN8Sup5(Group elements, float x, float y) {
1247         addCaptionAndRectangle(elements, "EAN/JAN 8 Bar Code, 5 digit " +
1248             "supplement", x, y, 204, 99);
1249
1250         // Add ean8 5 digit supplement bar code to the page.
1251         BarCode objBarCode = new Ean8Sup5("1234567012345", x, y + 21);
1252         objBarCode.setX(objBarCode.getX() + (204
1253             - objBarCode.getSymbolWidth()) / 2);
1254         elements.add(objBarCode);
1255     }
1256
1257     private void addUPCVersionE(Group elements, float x, float y) {
1258         addCaptionAndRectangle(elements, "UPC Version E Bar Code", x, y, 204,
1259             99);
1260
1261         // Add upcversionE bar code to the page.
```

## AllPageElements.java

---

```
1262     BarCode objBarCode = new UpcVersionE("0123456", x, y + 21);
1263     objBarCode.setX(objBarCode.getX() + (204
1264         - objBarCode.getSymbolWidth()) / 2);
1265     elements.add(objBarCode);
1266 }
1267
1268 private void addUPCVersionESup2(Group elements, float x, float y) {
1269     addCaptionAndRectangle(elements, "UPC Version E Bar Code, 2 digit " +
1270         "supplement", x, y, 204, 99);
1271
1272     // Add upcversionE 2 digit supplement bar code to the page.
1273     BarCode objBarCode = new UpcVersionESup2("012345612", x, y + 21);
1274     objBarCode.setX(objBarCode.getX() + (204
1275         - objBarCode.getSymbolWidth()) / 2);
1276     elements.add(objBarCode);
1277 }
1278
1279 private void addUPCVersionESup5(Group elements, float x, float y) {
1280     addCaptionAndRectangle(elements, "UPC Version E Bar Code, 5 digit " +
1281         "supplement", x, y, 204, 99);
1282
1283     // Add upcversionE 5 digit supplement bar code to the page.
1284     BarCode objBarCode = new UpcVersionESup5("012345612345", x, y + 21);
1285     objBarCode.setX(objBarCode.getX() + (204
1286         - objBarCode.getSymbolWidth()) / 2);
1287     elements.add(objBarCode);
1288 }
1289
1290 private void addCaptionAndRectangle(Group pageElements, String caption,
1291     float x, float y, float width,
1292     float height) {
1293     // Adds a rectangle and caption to the pageElements
1294     Rectangle objRectangle = new Rectangle(x, y + 15, width, height - 15);
1295     Label objCaption = new Label(caption, x, y, 225, 10,
1296         Font.getHelveticaBold(), 10);
1297
1298     pageElements.add(objRectangle);
1299     pageElements.add(objCaption);
1300 }
1301
1302 private void addIsbn(Group elements, float x, float y) {
1303     addCaptionAndRectangle(elements, "ISBN Barcode", x, y, 204, 99);
1304     Isbn barCode = new Isbn("978-1-23-456789-7", x, y + 21);
1305     barCode.setShowText(true);
1306     barCode.setX(barCode.getX() + (204 - barCode.getSymbolWidth()) / 2);
1307     elements.add(barCode);
1308 }
1309
1310 private void addIsbnSup2(Group elements, float x, float y) {
```

## AllPageElements.java

---

```
1311     addCaptionAndRectangle(elements, "ISBN Barcode, 2 digit supplement", x, y, 204, 99);
1312     IsbnSup2 barCode = new IsbnSup2("978-1-23-456789-7", "12", x, y + 21);
1313     barCode.setShowText(true);
1314     barCode.setX(barCode.getX() + (204 - barCode.getSymbolWidth()) / 2);
1315     elements.add(barCode);
1316 }
1317
1318 private void addIsbnSup5(Group elements, float x, float y) {
1319     addCaptionAndRectangle(elements, "ISBN Barcode, 5 digit supplement", x, y, 204, 99);
1320     IsbnSup5 barCode = new IsbnSup5("978-1-23-456789-7", "12345", x, y + 21);
1321     barCode.setShowText(true);
1322     barCode.setX(barCode.getX() + (204 - barCode.getSymbolWidth()) / 2);
1323     elements.add(barCode);
1324 }
1325
1326 private void addIsmn(Group elements, float x, float y) {
1327     addCaptionAndRectangle(elements, "ISMN Barcode", x, y, 204, 99);
1328     Ismn barCode = new Ismn("979-0-1234-5678", x, y + 21);
1329     barCode.setShowText(true);
1330     barCode.setX(barCode.getX() + (204 - barCode.getSymbolWidth()) / 2);
1331     elements.add(barCode);
1332 }
1333
1334 private void addIsmnSup2(Group elements, float x, float y) {
1335     addCaptionAndRectangle(elements, "ISMN Barcode, 2 digit supplement", x, y, 204, 99);
1336     IsmnSup2 barCode = new IsmnSup2("979-0-1234-5678", "12", x, y + 21);
1337     barCode.setShowText(true);
1338     barCode.setX(barCode.getX() + (204 - barCode.getSymbolWidth()) / 2);
1339     elements.add(barCode);
1340 }
1341
1342 private void addIsmnSup5(Group elements, float x, float y) {
1343     addCaptionAndRectangle(elements, "ISMN Barcode, 5 digit supplement", x, y, 204, 99);
1344     IsmnSup5 barCode = new IsmnSup5("979-0-1234-5678", "12345", x, y + 21);
1345     barCode.setShowText(true);
1346     barCode.setX(barCode.getX() + (204 - barCode.getSymbolWidth()) / 2);
1347     elements.add(barCode);
1348 }
1349
1350 private void addIssn(Group elements, float x, float y) {
1351     addCaptionAndRectangle(elements, "ISSN Barcode", x, y, 204, 99);
1352     Issn barCode = new Issn("977-1234-56700", x, y + 21);
1353     barCode.setShowText(true);
1354     barCode.setX(barCode.getX() + (204 - barCode.getSymbolWidth()) / 2);
1355     elements.add(barCode);
1356 }
1357
1358 private void addIssnSup2(Group elements, float x, float y) {
1359     addCaptionAndRectangle(elements, "ISSN Barcode, 2 digit supplement", x, y, 204, 99);
```

## AllPageElements.java

---

```
1360     IssnSup2 barCode = new IssnSup2("977-1234-56700", "12", x, y + 21);
1361     barCode.setShowText(true);
1362     barCode.setX(barCode.getX() + (204 - barCode.getSymbolWidth()) / 2);
1363     elements.add(barCode);
1364 }
1365
1366 private void addIssnSup5(Group elements, float x, float y) {
1367     addCaptionAndRectangle(elements, "ISSN Barcode, 5 digit supplement", x, y, 204, 99);
1368     IssnSup5 barCode = new IssnSup5("977-1234-56700", "12345", x, y + 21);
1369     barCode.setShowText(true);
1370     barCode.setX(barCode.getX() + (204 - barCode.getSymbolWidth()) / 2);
1371     elements.add(barCode);
1372 }
1373
1374 private void addSingaporePost(Group elements, float x, float y) {
1375     addCaptionAndRectangle(elements, "Singapore Post Barcode", x, y, 225, 80);
1376     BarCode barCode = new SingaporePost("208154", x, y + 15, true);
1377     barCode.setHeight(35);
1378     barCode.setX(barCode.getX() + (225 - barCode.getSymbolWidth()) / 2);
1379     barCode.setY(barCode.getY() + (65 - barCode.getHeight()) / 2);
1380     elements.add(barCode);
1381 }
1382
1383 private void addKix(Group elements, float x, float y) {
1384     addCaptionAndRectangle(elements, "Kix (Royal TNT Post) Barcode", x, y, 225, 80);
1385     BarCode barCode = new Kix("1231FZ13XHS", x, y + 15, true);
1386     barCode.setHeight(35);
1387     barCode.setX(barCode.getX() + (225 - barCode.getSymbolWidth()) / 2);
1388     barCode.setY(barCode.getY() + (65 - barCode.getHeight()) / 2);
1389     elements.add(barCode);
1390 }
1391
1392 private void addDeutschePostLeitcode(Group elements, float x, float y) {
1393     addCaptionAndRectangle(elements, "Deutsche Post Leitcode", x, y, 225, 80);
1394     BarCode barCode = new DeutschePostLeitcode("1234567890123", x, y + 15, 48);
1395     barCode.setX(barCode.getX() + (225 - barCode.getSymbolWidth()) / 2);
1396     barCode.setY(barCode.getY() + (65 - barCode.getHeight()) / 2);
1397     elements.add(barCode);
1398 }
1399
1400 private void addDeutschePostIdentcode(Group elements, float x, float y) {
1401     addCaptionAndRectangle(elements, "Deutsche Post Identcode", x, y, 225, 80);
1402     BarCode barCode = new DeutschePostIdentcode("12345678901", x, y + 15, 48);
1403     barCode.setX(barCode.getX() + (225 - barCode.getSymbolWidth()) / 2);
1404     barCode.setY(barCode.getY() + (65 - barCode.getHeight()) / 2);
1405     elements.add(barCode);
1406 }
1407
1408 private void addAustraliaPost(Group elements, float x, float y) {
```

## AllPageElements.java

---

```
1409     addCaptionAndRectangle(elements, "Australia Post Barcode", x, y, 225, 80);
1410     BarCode barCode = new AustraliaPost("1139549554", x, y + 15, true);
1411     barCode.setHeight(35);
1412     barCode.setX(barCode.getX() + (225 - barCode.getSymbolWidth()) / 2);
1413     barCode.setY(barCode.getY() + (65 - barCode.getHeight()) / 2);
1414     elements.add(barCode);
1415 }
1416
1417 private void addAztec(Group elements, float x, float y) {
1418     addCaptionAndRectangle(elements, "Aztec Barcode", x, y, 225, 100);
1419     Aztec aztec = new Aztec("Aztec Barcode", x, y + 15, AztecSymbolSize.AUTO);
1420     aztec.setXDimension(2.5f);
1421     aztec.setX(aztec.getX() + (225 - aztec.getSymbolWidth()) / 2);
1422     aztec.setY(aztec.getY() + (85 - aztec.getSymbolHeight()) / 2);
1423     elements.add(aztec);
1424 }
1425
1426 private void addStackedGs1Expanded(Group elements, float x, float y) {
1427     addCaptionAndRectangle(elements, "Stacked GS1 DataBar Expanded Barcode", x, y, 225, 100);
1428     StackedGS1DataBar barCode = new StackedGS1DataBar("(01)1234567890123", x, y + 15, 25,
StackedGS1DataBarType.EXPANDEDSTACKED);
1429     barCode.setX(barCode.getX() + (225 - barCode.getSymbolWidth()) / 2);
1430     barCode.setY(barCode.getY() + (85 - barCode.getSymbolHeight()) / 2);
1431     elements.add(barCode);
1432 }
1433
1434 private void addStackedGs1Stacked(Group elements, float x, float y) {
1435     addCaptionAndRectangle(elements, "Stacked GS1 DataBar Barcode", x, y, 225, 100);
1436     StackedGS1DataBar barCode = new StackedGS1DataBar("(01)1234567890123", x, y + 15, 25, StackedGS1DataBarType.STACKED);
1437     barCode.setX(barCode.getX() + (225 - barCode.getSymbolWidth()) / 2);
1438     barCode.setY(barCode.getY() + (85 - barCode.getSymbolHeight()) / 2);
1439     elements.add(barCode);
1440 }
1441
1442 private void addStackedGs1Omni(Group elements, float x, float y) {
1443     addCaptionAndRectangle(elements, "Stacked GS1 DataBar Omnidirectional Barcode", x, y, 225, 100);
1444     StackedGS1DataBar barCode = new StackedGS1DataBar("(01)1234567890123", x, y + 15, 25,
StackedGS1DataBarType.STACKEDOMNIDIRECTIONAL);
1445     barCode.setX(barCode.getX() + (225 - barCode.getSymbolWidth()) / 2);
1446     barCode.setY(barCode.getY() + (85 - barCode.getSymbolHeight()) / 2);
1447     elements.add(barCode);
1448 }
1449
1450 private void addGs1DataBarExpanded(Group elements, float x, float y) {
1451     addCaptionAndRectangle(elements, "GS1 DataBar Expanded Barcode", x, y, 225, 80);
1452     BarCode barCode = new GS1DataBar("(01)1234567890123", x, y + 15, 48, GS1DataBarType.EXPANDED);
1453     barCode.setX(barCode.getX() + (225 - barCode.getSymbolWidth()) / 2);
1454     barCode.setY(barCode.getY() + (65 - barCode.getHeight()) / 2);
1455     elements.add(barCode);
```

## AllPageElements.java

---

```
1456     }
1457
1458     private void addGs1DataBarLimited(Group elements, float x, float y) {
1459         addCaptionAndRectangle(elements, "GS1 DataBar Limited Barcode", x, y, 225, 80);
1460         BarCode barCode = new GS1DataBar("(01)1234567890123", x, y + 15, 48, GS1DataBarType.LIMITED);
1461         barCode.setX(barCode.getX() + (225 - barCode.getSymbolWidth()) / 2);
1462         barCode.setY(barCode.getY() + (65 - barCode.getHeight()) / 2);
1463         elements.add(barCode);
1464     }
1465
1466     private void addGs1DataBarOmnidirectional(Group elements, float x, float y) {
1467         addCaptionAndRectangle(elements, "GS1 DataBar Omnidirectional Barcode", x, y, 225, 80);
1468         BarCode barCode = new GS1DataBar("(01)9889876543210", x, y + 15, 48, GS1DataBarType.OMNIDIRECTIONAL);
1469         barCode.setX(barCode.getX() + (225 - barCode.getSymbolWidth()) / 2);
1470         barCode.setY(barCode.getY() + (65 - barCode.getHeight()) / 2);
1471         elements.add(barCode);
1472     }
1473
1474     private void addEan14(Group elements, float x, float y) {
1475         addCaptionAndRectangle(elements, "EAN-14 Barcode", x, y, 225, 80);
1476         BarCode barCode = new Ean14("1234567890123", x, y + 15, 48);
1477         barCode.setX(barCode.getX() + (225 - barCode.getSymbolWidth()) / 2);
1478         barCode.setY(barCode.getY() + (65 - barCode.getHeight()) / 2);
1479         elements.add(barCode);
1480     }
1481
1482     private void addItf14(Group elements, float x, float y) {
1483         addCaptionAndRectangle(elements, "ITF-14 Barcode", x, y, 225, 80);
1484         BarCode barCode = new Itf14("1234567890123", x, y + 15, 48);
1485         barCode.setX(barCode.getX() + (225 - barCode.getSymbolWidth()) / 2);
1486         barCode.setY(barCode.getY() + (65 - barCode.getHeight()) / 2);
1487         elements.add(barCode);
1488     }
1489
1490     private void addIata25(Group elements, float x, float y) {
1491         addCaptionAndRectangle(elements, "IATA-25 Barcode", x, y, 225, 80);
1492         BarCode barCode = new Iata25("1234567", x, y + 15, 48);
1493         barCode.setX(barCode.getX() + (225 - barCode.getSymbolWidth()) / 2);
1494         barCode.setY(barCode.getY() + (65 - barCode.getHeight()) / 2);
1495         elements.add(barCode);
1496     }
1497
1498     private void addExtendedCode93(Group elements, float x, float y) {
1499         addCaptionAndRectangle(elements, "Extended Code 93 Barcode", x, y, 225, 80);
1500         BarCode barCode = new Code93("Code 93", x, y + 15, 48);
1501         barCode.setX(barCode.getX() + (225 - barCode.getSymbolWidth()) / 2);
1502         barCode.setY(barCode.getY() + (65 - barCode.getHeight()) / 2);
1503         elements.add(barCode);
1504     }
```

## AllPageElements.java

```
1505
1506     private void addCode93(Group elements, float x, float y) {
1507         addCaptionAndRectangle(elements, "Code 93 Barcode", x, y, 225, 80);
1508         BarCode barCode = new Code93("CODE 93", x, y + 15, 48);
1509         barCode.setX(barCode.getX() + (225 - barCode.getSymbolWidth()) / 2);
1510         barCode.setY(barCode.getY() + (65 - barCode.getHeight()) / 2);
1511         elements.add(barCode);
1512     }
1513
1514     private void addExtendedCode39(Group elements, float x, float y) {
1515         addCaptionAndRectangle(elements, "Extended Code 3 of 9 Barcode", x, y, 225, 80);
1516         BarCode barCode = new Code39("Code 39", x, y + 15, 48);
1517         barCode.setX(barCode.getX() + (225 - barCode.getSymbolWidth()) / 2);
1518         barCode.setY(barCode.getY() + (65 - barCode.getHeight()) / 2);
1519         elements.add(barCode);
1520     }
1521
1522     private void addCode11(Group elements, float x, float y) {
1523         addCaptionAndRectangle(elements, "Code 11 Barcode", x, y, 225, 80);
1524         BarCode barCode = new Code11("12345678", x, y + 15, 48);
1525         barCode.setX(barCode.getX() + (225 - barCode.getSymbolWidth()) / 2);
1526         barCode.setY(barCode.getY() + (65 - barCode.getHeight()) / 2);
1527         elements.add(barCode);
1528     }
1529
1530     private void addMSICode(Group elements, int x, int y) {
1531         addCaptionAndRectangle(elements, "MSI Barcode", x, y, 225, 80);
1532         MsiBarcode msi = new MsiBarcode("1234567890", x, y, 50, true);
1533         msi.setY(msi.getY() + (80 - msi.getHeight()) / 2 + 8);
1534         msi.setX(msi.getX() + (225 - msi.getSymbolWidth()) / 2);
1535         elements.add(msi);
1536     }
1537
1538     private void addRm4sccCode(Group elements, int x, int y) {
1539         addCaptionAndRectangle(elements, "RM4SCC Barcode", x, y, 225, 80);
1540         Rm4scc rm4sccCode = new Rm4scc("1457826RA", x, y, true);
1541         rm4sccCode.setY(rm4sccCode.getY() + (65 - rm4sccCode.getHeight()) / 2 + 13);
1542         rm4sccCode.setX(rm4sccCode.getX() + (225 - rm4sccCode.getSymbolWidth()) / 2);
1543         elements.add(rm4sccCode);
1544     }
1545
1546     private void addBarCodePage(Document document, Outline parentOutline) {
1547         // Increments a page counter
1548         pageCounter++;
1549
1550         // Adds to outlines to the page
1551         parentOutline.getChildOutlines().add("Page " + pageCounter + " - Other Linear Barcodes", new
1552 ZoomDestination(pageCounter, PageZoom.FIT_PAGE));
```

## AllPageElements.java

---

```
1553 // Create a page to add to the document
1554 Page page = new Page(PageSize.LETTER, PageOrientation.PORTRAIT, 54);
1555
1556     // Uncomment the line below to add a layout grid to the page
1557 //page.Elements.Add( new LayoutGrid() );
1558 // Add the page elements to the page
1559 addCodabar(page.getElements(), 0, 0);
1560 addCode11(page.getElements(), 0, 88);
1561 addCode128(page.getElements(), 0, 176);
1562 addCode25(page.getElements(), 0, 264);
1563 addCode39(page.getElements(), 0, 352);
1564 addExtendedCode39(page.getElements(), 0, 440);
1565 addCode93(page.getElements(), 0, 528);
1566 addExtendedCode93(page.getElements(), 0, 616);
1567
1568 addIata25(page.getElements(), 279, 0);
1569 addInterleaved25(page.getElements(), 279, 88);
1570 addItf14(page.getElements(), 279, 176);
1571 addMSICode(page.getElements(), 279, 264);
1572 addEan14(page.getElements(), 279, 352);
1573 addGs1DataBarOmnidirectional(page.getElements(), 279, 440);
1574 addGs1DataBarLimited(page.getElements(), 279, 528);
1575 addGs1DataBarExpanded(page.getElements(), 279, 616);
1576
1577 document.getPages().add(page);
1578 }
1579
1580 private void add2DBarcodePage(Document document, Outline parentOutline) {
1581     // Increments a page counter
1582     pageCounter++;
1583
1584     // Adds to outlines to the page
1585     parentOutline.getChildOutlines().add("Page " + pageCounter + " - 2D and Stacked GS1 Databar Barcodes", new
1586 ZoomDestination(pageCounter, PageZoom.FIT_PAGE));
1587
1588     // Create a page to add to the document
1589     Page page = new Page(PageSize.LETTER, PageOrientation.PORTRAIT, 54);
1590
1591     // Uncomment the line below to add a layout grid to the page
1592 //page.Elements.Add( new LayoutGrid() );
1593 // Add the page elements to the page
1594 addStackedGs1Omni(page.getElements(), 0, 0);
1595 addStackedGs1Stacked(page.getElements(), 0, 108);
1596 addStackedGs1Expanded(page.getElements(), 0, 216);
1597 addAztec(page.getElements(), 0, 324);
1598 addQrCode(page.getElements(), 0, 432);
1599
1600 addPdf417(page.getElements(), 279, 0);
1601 addPdf417Macro(page.getElements(), 279, 108);
```

## AllPageElements.java

---

```
1601     addDataMatrixBarcode(page.getElements(), 279, 324);  
1602  
1603     document.getPages().add(page);  
1604 }  
1605  
1606 private void addPostalBarcodePage(Document document, Outline parentOutline) {  
1607     // Increments a page counter  
1608     pageCounter++;  
1609  
1610     // Adds to outlines to the page  
1611     parentOutline.getChildOutlines().add("Page " + pageCounter + " - Postal Barcodes", new ZoomDestination(pageCounter,  
PageZoom.FIT_PAGE));  
1612  
1613     // Create a page to add to the document  
1614     Page page = new Page(PageSize.LETTER, PageOrientation.PORTRAIT, 54);  
1615  
1616     // Uncomment the line below to add a layout grid to the page  
1617     //page.Elements.Add( new LayoutGrid() );  
1618     // Add the page elements to the page  
1619     addAustraliaPost(page.getElements(), 0, 0);  
1620     addDeutschePostIdentcode(page.getElements(), 0, 120);  
1621     addDeutschePostLeitcode(page.getElements(), 0, 240);  
1622     addKix(page.getElements(), 0, 360);  
1623  
1624     addIntelligentMailBarcode(page.getElements(), 279, 0);  
1625     addRm4sccCode(page.getElements(), 279, 120);  
1626     addPostnet(page.getElements(), 279, 240);  
1627     addSingaporePost(page.getElements(), 279, 360);  
1628  
1629     document.getPages().add(page);  
1630 }  
1631  
1632 private void addIsbnIssnIsmnPage(Document document, Outline parentOutline) {  
1633     // Increments a page counter  
1634     pageCounter++;  
1635  
1636     // Adds to outlines to the page  
1637     parentOutline.getChildOutlines().add("Page " + pageCounter + " - ISBN/ISMN/ISSN Barcodes", new  
ZoomDestination(pageCounter, PageZoom.FIT_PAGE));  
1638  
1639     // Create a page to add to the document  
1640     Page page = new Page(PageSize.LETTER, PageOrientation.LANDSCAPE, 54);  
1641  
1642     // Uncomment the line below to add a layout grid to the page  
1643     //page.Elements.Add( new LayoutGrid() );  
1644     // Add the page elements to the page  
1645     addIsbn(page.getElements(), 0, 0);  
1646     addIsbnSup2(page.getElements(), 240, 0);  
1647     addIsbnSup5(page.getElements(), 480, 0);
```

## AllPageElements.java

---

```
1648     addIsmn(page.getElements(), 0, 150);
1649     addIsmnSup2(page.getElements(), 240, 150);
1650     addIsmnSup5(page.getElements(), 480, 150);
1651     addIssn(page.getElements(), 0, 300);
1652     addIssnSup2(page.getElements(), 240, 300);
1653     addIssnSup5(page.getElements(), 480, 300);
1654
1655     // Add the page to the document
1656     document.getPages().add(page);
1657 }
1658 private void addCaptionAndRectangle(Group pageElements, String caption,
1659         float x, float y) {
1660     addCaptionAndRectangle(pageElements, caption, x, y, 225, 85);
1661 }
1662
1663 private void addWatermarkPage(Document document, Outline parentOutline) {
1664     pageCounter++;
1665
1666     // Adds to outlines to the page
1667     parentOutline.getChildOutlines().add("Page " + pageCounter + " - Watermark text", new ZoomDestination(pageCounter,
1668 PageZoom.FIT_PAGE));
1669
1670     // Create a page to add to the document
1671     Page page = new Page(PageSize.LETTER, PageOrientation.PORTRAIT, 54);
1672
1673     String caption = "Text and Image Watermark Page Elements";
1674     Label captionLabel = new Label(caption, 0, 235, 225, 10, Font.getHelveticaBold(), 10);
1675     page.getElements().add(captionLabel);
1676
1677     // Create Label and add description text
1678     String text = "Text or Image watermarks can be placed on any page";
1679     Label label = new Label(text, 5, 255, 500, 100);
1680     label.setFontSize(40);
1681     page.getElements().add(label);
1682
1683     // Create a Rectangle
1684     Rectangle rect = new Rectangle(0, 250, 500, 100);
1685     page.getElements().add(rect);
1686
1687     // Create Text watermark
1688     TextWatermark watermark = new TextWatermark("Draft");
1689     watermark.setTextColor(null);
1690     watermark.setTextOutlineColor(RgbColor.getBlack());
1691
1692     // Add Text watermark to page
1693     page.getElements().add(watermark);
1694
1695     // Set Pdf version to 1.6
1696     document.setPdfVersion(PdfVersion.v1_6);
```

## AllPageElements.java

---

```
1696  
1697     // Add page to document  
1698     document.getPages().add(page);  
1699 }  
1700 }
```