XPrint 利用の手引き

初版　　2004/9/1

第 2 版　　2010/6/25

目次

1. 概要 ・・・ 02
2. 動作環境 ・・・ 04
3. タグ ・・・ 05
4. 属性 ・・・ 07
5. 要素のレイアウト ・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・ 09
6. 要素の境界線 ・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・ 10
7. ページビューワ ・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・ 11
8. Java API 仕様 ・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・ 12
9. Java API サンプルソース ・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・ 13

1. 概要

XPrint は、HTML のように簡単な記述言語で帳票を作成することを目的とする、帳票作成アプリケーションです。

XPrint を使用して帳票を作成するためには、印刷レイアウトを定義したフォームファイルと、それに埋め込むデータファイルが必要です。Java API を使用する場合は、データファイルではなくメモリ上の Java の DOM オブジェクトを直接指定することもできます。

1-1. フォームファイル

フォームファイルは XML 形式のファイルで、XSLT と独自のタグを使用して記述します。

繰り返しデータは、XSLT の for-each タグ、動的データの埋め込みは text タグと XSLT の value-of タグを組み合わせて指定します。

```
<?xml version="1.0" encoding="UTF-8" ?>
<xsl:stylesheet version="1.0" xmlns:xsl="http://www.w3.org/1999/XSL/Transform">
<xsl:template match="/root">

<page width="595" height="841" font-family=" ＭＳ ゴシック ">

 <div border-width="0 0 1 0" font-size="16">
  <text><xsl:value-of select="*/name"/></text>
 </div>

 <div height="2"/>

 <div padding="0">
  <div border-width="0.5" align="left" width="200" background-color="#999999">
   <text align="center" content-align="center"> 名前 </text>
  </div>
  <div border-width="0.5" align="left" width="200" background-color="#999999">
   <text align="center" content-align="center"> 価格 </text>
  </div>
 </div>

 <xsl:for-each select="//item">
  <record>
   <div padding="0">
    <div border-width="0.5" align="left" width="200">
     <text><xsl:value-of select="name"/></text>
    </div>
    <div border-width="0.5" align="left" width="200">
     <text content-align="right"><xsl:value-of select="price"/></text>
    </div>
   </div>
  </record>
 </xsl:for-each>

</page>

</xsl:template>
</xsl:stylesheet>
```

リスト 1.　フォームファイル

1-2. データファイル

データファイルも XML 形式のファイルですが、こちらは任意のデータ構造を持つことができます。

```
<?xml version="1.0" encoding="UTF-8" ?>

<root>

 <name> 野菜 太郎 </name>

 <item>
  <name> きゅうり </name>
  <price>\128</price>
 </item>

 <item>
  <name> なす </name>
  <price>\158</price>
 </item>

 <item>
  <name> ピーマン </name>
  <price>\78</price>
 </item>

 <item>
  <name> 長ネギ </name>
  <price>\198</price>
 </item>

</root>
```

リスト 2.　データファイル

1-3.　ページビューワ

作成したフォームファイルとデータファイルの印刷イメージを確認するために、ページビューワを利用します。ページビューワは HTML を表示する WEB ブラウザにあたるアプリケーションで、帳票の印刷イメージを予め確認することができます。

リスト 1. フォームファイルと、リスト 2. データファイルを指定して Reload ボタンを押下すると、図 1 のような印刷イメージが表示されます。

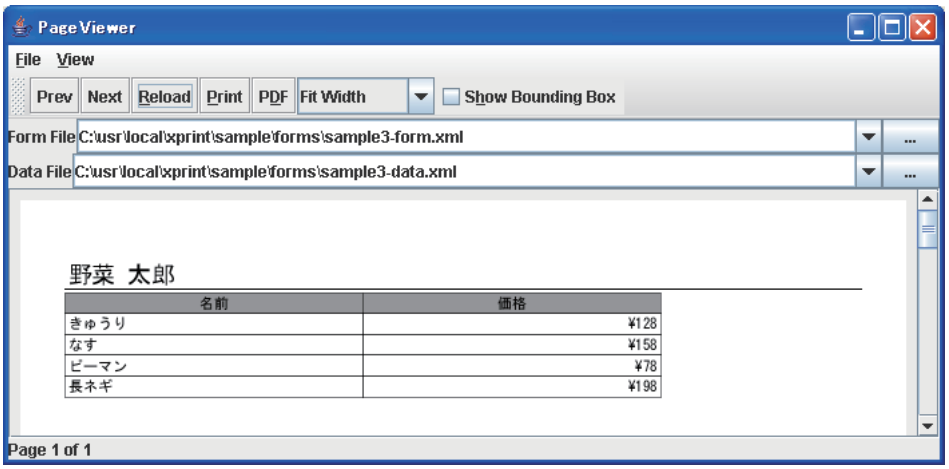

図 1.　ページビューワ

2. 動作環境

JDK(JRE)1.5 以上

3. タグ

フォームファイルで利用可能なタグを表 1. タグ一覧に示します。

page	フォームのルート要素です。 用紙サイズや、既定フォントを指定します。
div	構成要素をレイアウトするために使用します。 <text> ヘッダ </text> <text> 残りの部分 </text>
header	ヘッダを指定するために使用します。 ここで扱うヘッダとは、全ページに印刷される項目のことで、一般的にはフッタと呼ばれているものも含みます。 <header align="bottom"> <text> ページ <property select="page-no"/></text> </header>
text	文字列要素です。静的、または動的に文字を埋め込むために使用します。 静的文字の埋め込みは、text タグで囲って指定します。 <text> ようこそ </text> 動的文字の埋め込みは、XSLT の value-of を使用します。 <text><xsl:value-of select="UserName"/></text>
record	繰り返し項目内で、1 ページ内に収めたい項目をグループ化するために使用します。 <xsl:for-each select="User"> <record page-break="auto"> </record> </xsl:for-each>
image	画像です。
barcode	バーコードです。 <barcode type="JAN">012345678901</barcode>
qrcode	QR コードです。 <qrcode unit-width="0.3mm">mail@example.com</qrcode>
script	スクリプトです。 スクリプトは ECMAScript(JavaScript) で記述します。グラフなど、汎用タグでは表現できないものを描画することができます。 <script src="mygraph.js" width="200mm" height="100mm"/>

property	システムプロパティ (印刷時に決まる値) を埋め込むために使用します。 text 要素と組み合わせて、ページ番号や全ページ数を埋め込むことができます。 select 属性には、page-no(ページ番号) または page-count(総ページ数) を指定します。 <text> ページ <property select="page-no"/></text>

表 1.　タグ一覧

4. 属性

フォームファイルのタグで指定可能な属性を表 2. 属性一覧に示します。

数値項目はポイント (pt)、ミリ (mm)、センチ (cm)、インチ (in) のいずれかの単位で指定します (72pt=1inch=25.4mm)。単位が未指定の場合は、ポイントとして扱われます。

色項目は HTML と同様に、#RRGGBB 形式で指定します。色名で指定することはできません。

margin、padding、border-width は、上、右、下、左の順で、空白区切りで指定します。空白区切りをせずに単一の値を指定した場合は、上下左右全てに同じ値が設定されます。

継承項目は、親要素の値を引き継ぎます。

	page	div	header	text	record	image	barcode	qrcode	script	property	継承	
width	●	●	●	●	●	●	●	●	●			幅
height	●	●	●	●	●	●	●	●	●			高さ
align	●	●	●	●	●	●	●	●	●			レイアウト位置 • top(既定) • bottom • left • right • center
color	●	●	●	●	●	●	●	●	●			前景色 (文字および境界線)
background-color	●	●	●	●	●	●	●	●	●			背景色
padding	●	●	●	●	●	●	●	●	●			境界線内余白
margin	●	●	●	●	●	●	●	●	●			境界線外余白
border-width	●	●	●	●	●	●	●	●	●			境界線の太さ
corner-radius	●	●	●	●	●	●	●	●	●			角丸半径
stroke-dash-pattern	●	●	●	●	●	●	●	●	●			点線パターン
stroke-cap	●	●	●	●	●	●	●	●	●			終端 • butt • round • square(既定)
stroke-join	●	●	●	●	●	●	●	●	●			角の形 • miter(既定) • round • bevel
font-family	●	●	●	●	●				●		●	フォントファミリ名 システムにインストール済みのフォントである必要があります。
font-size	●	●	●	●	●				●		●	フォントサイズ
font-weight	●	●	●	●	●				●		●	フォントの太さ • plain(既定): 通常 • bold: 太字

	page	div	header	text	record	image	barcode	qrcode	script	property	継承	
page-break					●							改ページ • no(既定): なるべく改ページしない • always: 常に改ページする • auto: ページ中途で、繰り返しブロック全体がページ内に収まらない場合は改ページする
content-align				●								内容の横方向位置 • left(既定) • center • right
content-vertical-align				●								内容の縦方向位置 • top • center(既定) • bottom
src						●			●			ソース
type							●					種別
unit-width							●	●				単位幅
type-number								●				
error-correct-level								●				
select										●		データ選択 • page-no • page-count

表 2.　属性一覧

5. 要素のレイアウト

要素のレイアウトには、align 属性を使用します。align には、上下左右と中央を指定することができます。

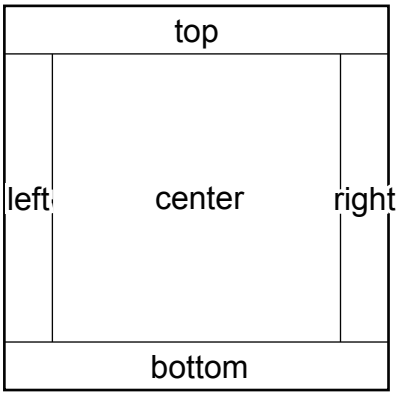

図 2. align 指定とレイアウト位置

同一階層に、同じ align の要素を記述した場合、上下左右の場合は記述した順に並べてレイアウトされます。中央の場合は、同じ領域に重なってレイアウトされます。この特徴を利用して、画像に文字を重ねたりすることができます。

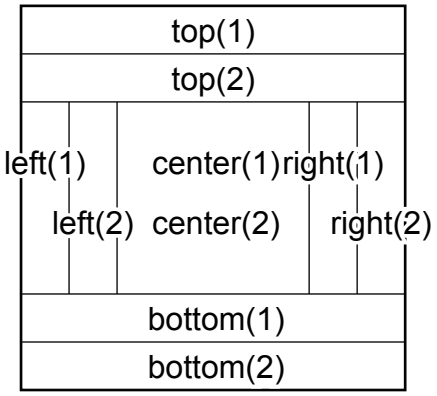

図 3. 同一階層に同じ align の要素を配置

6. 要素の境界線

要素の境界線の、外側の余白を margin、内側の余白を padding と呼びます。外接する領域との境界を計算する際には margin、内接する領域との境界を計算する際には padding が使用されます。

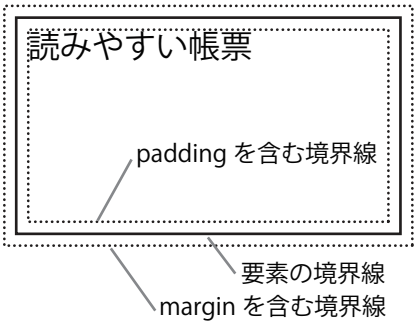

図 4. 要素の境界

7. ページビューワ

7-1. 起動方法

pageviewer.jar をダブルクリックすると、ページビューワが起動します。ダブルクリックで起動しない場合はコマンドラインから起動します。

```
java -jar pageviewer.jar
```

7-2. 操作方法

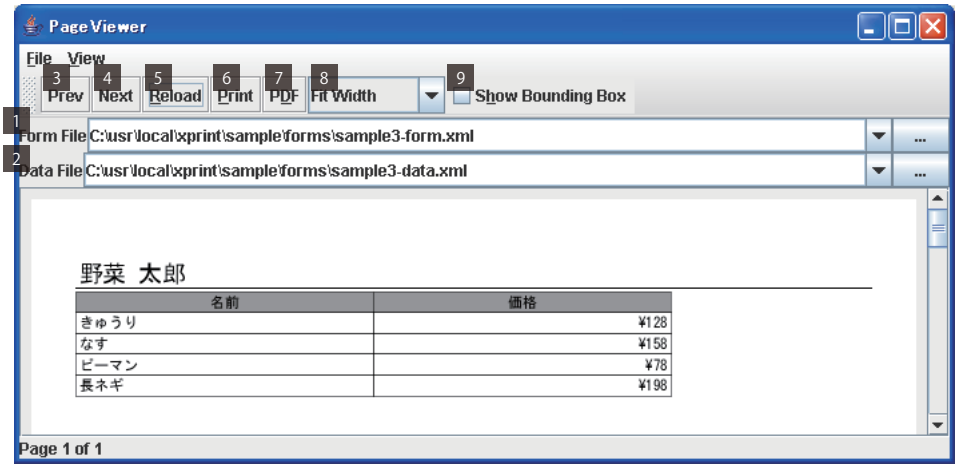

図 5. ページビューワ

	項目	機能	ショートカット
1	フォームファイル名	フォームファイル名を入力します。 右端のボタンをクリックすると、ファイルを選択するためのダイアログが開きます。	
2	データファイル名	埋め込みに使うデータファイル名を入力します。 データを埋め込まない場合は省略可能です。	
3	前ページ表示	ページが複数ある場合に、前のページを表示させます。	Alt+ 左矢印
4	次ページ表示	ページが複数ある場合に、次のページを表示させます。	Alt+ 右矢印
5	再ロード	再ロードを実行して画面表示を最新にします。	F5
6	印刷	表示中の帳票を印刷します。	Ctrl+P
7	PDF	表示中の帳票を PDF ファイルとして保存します。	
8	表示スケール	画面に表示する大きさを変更します。	
9	要素領域の表示	要素領域を表示します。	

8. Java API 仕様

XPrint では Java API として、xprint.XPrint クラスを提供します。

XPrint クラスには、引数の異なる print メソッドが用意されています。データをファイルで渡したい場合、ファイルを作らずに直接 DOM オブジェクトを渡したい場合など、用途に合わせて使い分けます。

---

public static void print(java.awt.print.PrinterJob job,
java.io.File formFile,
org.w3c.dom.Node dataNode)
throws XPrintException

フォームファイルとデータノードを指定して印刷を実行する。

パラメータ :
job - 印刷するプリンター・ジョブ
formFile - フォームファイル
dataNode - データノード
例外 :
XPrintException - 印刷中にエラーが発生

---

public static void print(java.awt.print.PrinterJob job,
java.io.File formFile,
java.io.File dataFile)
throws XPrintException

フォームファイルとデータファイルを指定して印刷を実行する。

パラメータ :
job - 印刷するプリンター・ジョブ
formFile - フォームファイル
dataFile - データファイル
例外 :
XPrintException - 印刷中にエラーが発生

---

9. Java API サンプルソース

フォームファイルはファイルを指定、埋め込みデータは動的に作成した DOM オブジェクトを指定する場合の Java ソースはリスト 3 のようになります。

ここで作成している埋め込みデータは、リスト 2 のデータファイルと同等の階層をもつデータ構造になります。

```
import java.awt.print.PrinterJob;
import java.io.File;

import javax.xml.parsers.DocumentBuilderFactory;

import org.w3c.dom.Document;
import org.w3c.dom.Node;

import xprint.XPrint;

public class Sample {

   public static void main(String[] args) throws Exception {

      // フォームファイル ( コマンドラインの第 1 引数 )
      File formFile = new File(args[0]);

      // 埋め込みデータの DOM を作成
      Document dataDoc = DocumentBuilderFactory.newInstance()
         .newDocumentBuilder().newDocument();
      Node dataNode = dataDoc.appendChild(dataDoc.createElement("root") );
      dataNode.appendChild(dataDoc.createElement("name") )
         .appendChild(dataDoc.createTextNode(" 帳票 太郎 ") );
      for (int i = 0; i < 10; i++) {
         Node itemNode = dataNode.appendChild(dataDoc.createElement("item") );
         itemNode.appendChild(dataDoc.createElement("name") )
            .appendChild(dataDoc.createTextNode(" 商品 " + i) );
         itemNode.appendChild(dataDoc.createElement("price") )
            .appendChild(dataDoc.createTextNode("\\100") );
      }

      // 既定のプリンターで印刷
      XPrint.print(PrinterJob.getPrinterJob(), formFile, dataDoc);
   }
}
```

リスト 3.　サンプルソース