

# 利用 iText 包实现 Java 报表打印

丁振凡<sup>1</sup>, 王小明<sup>1</sup>, 吴小元<sup>1</sup>, 邓建明<sup>2</sup>, 周斌<sup>2</sup>

(1. 华东交通大学, 江西 南昌 330013;

2. 南昌铁路局, 江西 南昌 330001)

**摘要:** 结合报表制作的两种情形介绍了 iText 的应用方法。一种是由程序对象动态产生整个报表文件的内容, 另一种是在已存在的 PDF 报表文档中填写数据域以完成报表。给出了 Java 实现报表打印的控制方法。

**关键词:** Java 报表; iText 包; 动态报表; 填充型报表; 报表打印

中图分类号: TP393

文献标识码: A

文章编号: 1674-7720(2012)18-0084-03

## Java print table building by using iText package

Ding Zhenfan<sup>1</sup>, Wang Xiaoming<sup>1</sup>, Wu Xiaoyuan<sup>1</sup>, Deng Jianming<sup>2</sup>, Zhou Bin<sup>2</sup>

(1. East China Jiao Tong University, Nanchang 330013, China;

2. Nanchang Railway Bureau, Nanchang 330001, China)

**Abstract:** The use of iText was introduced combing two cases of report generation in the paper. One case is that the whole content of the report file is dynamically generated by program while the other case is that the data fields is just be filled in the existing PDF report documents. The control method of report print implemented by java was also given in the paper.

**Key words:** Java report; iText library; dynamic report; filling type report; report print

在信息系统应用中, 报表处理一直起着比较重要的作用。Java 报表制作中最常使用的是 iText 组件, 它是一种生成 PDF 报表的 Java 组件。本文讨论两种形式的 PDF 报表处理, 一种是通过程序对象生成整个 PDF 报表文档, 另一种是利用制作好的含报表的 PDF 文档模板, 通过在模板填写数据实现数据报表。

### 1 通过编程绘制实现报表的生成

对于内容动态变化的表格, 适合使用程序绘制办法进行生成处理。这类表格中数据项和数据均是动态存在的。

#### 1.1 使用 iText 编程生成含报表的 PDF 文档的步骤<sup>[1]</sup>

(1) 建立 Document 对象。Document 是 PDF 文件所有元素的容器。

```
Document document = new Document();
```

(2) 建立一个与 Document 对象关联的书写器(Writer)。通过书写器(Writer)对象可以将具体文档存盘成需要的格式, PDFWriter 可以将文档保存为 PDF 文件。

```
PDFWriter.getInstance(document, new FileOutputStream("my.PDF"));
```

(3) 打开文档。如: document.open();

(4) 向文档中添加内容。所有向文档添加的内容都是以对象为单位的, iText 中用文本块(Chunk)、短语(Phrase)和段落(Paragraph)处理文本。

```
document.add(new Paragraph("Hello World"));
```

//添加一个段落

值得注意的是文本中汉字的显示, 默认的 iText 字体设置不支持中文字体, 需要下载远东字体包 iTextAsian.jar, 否则不能往 PDF 文档中输出中文字体<sup>[2]</sup>。

(5) 关闭文档。如: document.close();

#### 1.2 表格绘制

要在 PDF 文件中创建表格, iText 提供了两个类——Table 和 PdfPTable。Table 类用来实现简单表格, PdfPTable 类则用来实现比较复杂的表格。本文主要讨论 PdfPTable 类的应用。

(1) 创建 PdfPTable 对象

创建 PdfPTable 对象只需要指定列数, 不用指定行数。通常生成的表格默认以 80% 的比例显示在页面上。例如定义 3 列的表格, 每列的宽度分别为 15%、25% 和 60%, 语句如下:

## 应用奇葩

Example of Application

```
float[] widths = {15f, 25f, 60f};
```

```
PdfPTable table = new PdfPTable(widths);
```

用 `setWidthPercentage(float widthPercentage)` 方法可设置表格的按百分比的宽度。而用 `setTotalWidth` 则可设置表格按像素计算的宽度。如果表格的内容超过了 300 px, 表格的宽度会自动加长。用 `setLockedWidth(true)` 方法可锁定表格宽度。通过表格对象的系列方法可设置表格的边界以及对齐、填充方式。

## (2) 添加单元格

表格创建完成以后, 可通过 `addCell(Object object)` 方法插入单元格元素 (`PdfPCell`)。其中, `Object` 对象可以是 `PdfPCell`、`String`、`Phrase`、`Image`, 也可以是 `PdfPTable` 对象本身, 即在表格中嵌套一个表格。通过单元格的方法可设定单元格的列跨度、边框粗细、对齐方式、填充间隙等。

## (3) 合并单元格

为了实现某些特殊的表格形式, 需要合并单元格。`PdfPCell` 类提供了 `setColspan(int colspan)` 方法用于合并横向单元格, 参数 `colspan` 为合并的单元格数。但要合并纵向单元格需要使用嵌套表格的方法。将某个子表加入单元格, 且安排单元格所占列数为子表中列数, 则其行跨度也就是子表中的行数。

由于实际编程时, 经常出现各类结构的嵌套情形, 可以将产生某种结构的表格模块进行封装, 编制成方法, 通过传递方法参数完成表格特定模块的绘制。

例如, 可以将生成一个整齐行列表格的代码编写成方法。方法返回表格, 填充的数据通过二维对象数组传递。代码如下:

```
public static PdfPTable creatSubTable(Object x[][]){
    PdfPTable t= new PdfPTable(x[0].length);
    t.getDefaultCell().setHorizontalAlignment(Element.ALIGN_
        CENTER);
    for (int k=0;k<x.length;k++) {
        for (int j=0;j<x[0].length;j++)
            t.addCell(new Phrase(x[k][j].toString(),
                FontChinese));
    }
    return t;
}
```

以下代码调用上述方法, 绘制图 1 表格中黑框内部分:

```
PdfPTable t2= new PdfPTable(3);
String x1[][]={ {"+601k", "10", "合格"}, {"-601k",
    "11", "合格"}, {"+601k-601k", "12", "合格"} };
PdfPCell m=new PdfPCell(creatSubTable(x1));
//将创建的子表放入单元格
m.setColspan(3); //单元格占用外层表格的 3 栏
t2.addCell(m);
```

## 2 基于 PDF 报表模板的报表生成

有些表格具有固定的格式, 实际工作中只是给表格填写数据。这类表格可转换为 PDF 文件格式的报表模

图 1 含跨行和跨列的检测记录表

板, 通过特殊工具在文件中定义若干数据域, 通过给数据域写入数据实现对报表数据的填充处理。它具有格式灵活的特点。基于报表模板的报表处理步骤如下:

- (1) 利用 Word 制作打印报表;
- (2) 利用 Adobe Acrobat 7.0 Professional 将 Word 文档转换为 PDF 格式;
- (3) 利用 Adobe Designer 7.0 对 PDF 进行设计, 定义数据域;
- (4) 利用 iText 组件实现对报表数据字段的写入。

可利用 Adobe Designer 7.0 导入某个 PDF 文件进行设计, 在任意位置添加文本域。每个文本域有一个绑定的名称和值, 在 Java 程序中正是通过文本域的名称访问文本域对象。

图 2 给出了利用 Adobe Acrobat 7.0 Professional 打开一个制作好的带数据域定义的 PDF 文档模板文件的浏览界面, 出于清晰考虑, 图中特别将数据域采用高亮度显示。



图 2 带数据域定义的 PDF 文档

以下给出了打开报表模板实现数据写入的关键代码:

```
import com.itextpdf.text.DocumentException;
import com.itextpdf.text.pdf.AcroFields;
import com.itextpdf.text.pdf.PdfReader;
```

```
import com.itextpdf.text.pdf.PdfStamper;
.....
PdfReader r=new PdfReader("d:\预检模板
(DC600V 方式).pdf"); // ①
PdfStamper s=new PdfStamper(r,new FileOutputStream
("d:\结果.pdf")); //②
AcroFields form=s.getAcroFields(); // ③
String x[]=detectlog.getYjdata(date,cheNumber,code);
//读取数据库数据
form.setField("日期", x[1]); // ④
form.setField("修规", x[2]);
.....
s.close();
```

## 【说明】

①利用 PdfReader 读取 PDF 文档；通过实例化 PdfReader 对象来获取 pdf 模板，传入的字符串就是 pdf 文件所放置的路径，可以用绝对路径表示。

②取得对象后，需要用 PdfStamper 来编辑 PdfReader 对象，同时获取一个 OutputStream 输出流作为输出对象。

③利用 PdfStamper 获取文件中定义的 AcroFields 对象。

④用 AcroFields 对象的 setField 填写各个字段的数据到表格中。

## 3 Java 打印 PDF 报表文件

在 Web 应用中要在客户端打印 PDF 文档只需要将文件送客户浏览器显示，利用浏览器客户端的文件打印功能可实现打印。以下讨论在 Java 应用程序中如何打印报表文件。

Java 实现报表打印首先要获取打印服务对象，然后利用服务对象开始一个作业的打印。以下给出了新的 JDK1.4 以上版本中实现打印的具体步骤和关键代码。

```
// ① 构建打印请求属性集
PrintRequestAttributeSet pras = new HashPrintRequest-
AttributeSet();
// ② 设置打印格式，因为未确定文件类型，
这里选择 AUTOSENSE
```

```
DocFlavor flavor=DocFlavor.INPUT_STREAM.AUTOSENSE;
// ③ 查找所有的可用打印服务
PrintService printService[] = PrintServiceLookup.lookup-
PrintServices(flavor, pras);
// ④ 定位默认的打印服务
PrintService defaultService =
PrintServiceLookup.lookupDefaultPrintService();
// ⑤ 显示打印对话框
PrintService service = ServiceUI.printDialog(null, 200,
200, printService, defaultService, flavor, pras);
if (service != null) {
DocPrintJob job = service.createPrintJob();
// ⑥ 创建打印作业
FileInputStream fis = new FileInputStream(file);
// 假设 file 为具体文件对象
DocAttributeSet das = new HashDocAttributeSet();
Doc doc = new SimpleDoc(fis, flavor, das);
// ⑦ 建立打印文件格式
job.print(doc, pras); // ⑧ 进行文件的打印
}
```

本文介绍了利用 iText 实现 PDF 报表打印的编程处理方法。分别介绍了报表处理中的两种典型应用。一种是由程序对象动态产生整个报表，重点介绍了利用 PdfPTable 等对象实现报表生成处理的技巧。另一种是在已存在的 PDF 报表文档中填写数据域完成报表，重点介绍了通过 AcroFields 对象实现数据域的填写处理。最后给出了 Java 实现报表打印的控制方法。

## 参考文献

- [1] 侯金波,王德群.iText API 在计量信息管理系统中的应用[J].计量与测试技术,2007,34(1):57-60.
- [2] 王丽萍,秦永平.基于 iText 的 PDF 报表设计[J].电脑知识与技术,2007(8):493-494.

(收稿日期:2012-05-22)

## 作者简介:

丁振凡,男,1965年生,教授,硕士生导师,主要研究方向:语义 Web,分布式计算,计算机辅助教学。