

A Generalised Reporting Application based upon XML Descriptions

Fan Shao, Mango Solutions, Chippenham, UK

ABSTRACT

Report Generator is a XML based generalized reporting application which takes XML input and produces reports of various format. It is implemented as a Java library. This paper introduces this light-weight and easy-to-use library and describes the main design issues of the library.

INTRODUCTION

Reporting is an important feature required by many enterprise systems. There is a number of reporting products in the market but most of those contain too many features that are not to be used therefore complicate the issue. In such a situation a light-weight, intuitive and easy-to-use reporting component is needed, and this XML based Report Generator is developed to solve this problem.

DESIGN

The report generator takes a report definition in the form of XML, and produces PDF, RTF or HTML reports. The style of a report is defined using CSS (Cascade Style Sheet) [1], and the style and the layout are defined separately so that the style or layout can be changed without much impact to other parts of the report..

Shown below is a section of a xml definition of an example report:

```
<style type="text/css">
    ReportStyle {width: 210mm; height: 297mm;}
</style>

<layout xsi:type="flow" direction="vertical">
    <reportItem>items/text.xml</reportItem>
    <reportItem>items/image.xml</reportItem>
    <reportItem>items/table.xml</reportItem>

    <layout xsi:type="flow" direction="horizontal">
        <reportItem>items/text.xml</reportItem>
        <reportItem>
            <image class="LargeImage">
                <file class="SmallImage" label="Test Plot A">
                    outputA.jpg
                </file>
            </image>
        </reportItem>

        <layout xsi:type="tabular" direction="vertical" columns="2"
fill="page">
            <reportItem>items/image.xml</reportItem>
            <reportItem>items/text.xml</reportItem>
            <reportItem>items/image.xml</reportItem>
            <reportItem>items/text.xml</reportItem>
        </layout>

    </layout>
</layout>
```

REPORT LAYOUT

The report generator supports flow layout and tabular layout. The flow layout puts the items side by side and it automatically put an item in a new line if it runs out of space. The tabular layout allows for displaying the items in a table. The layout elements can be nested in each other. By combining the layouts the items can be positioned wherever required.

PhUSE 2008

REPORT STYLE

The styling of a report is implemented using CSS, which is a W3C [2] standard used in HTML and other XML based technologies. To allow for best flexibility the style can be defined in multiple places. It can be passed in to the application at runtime, or defined in the XML report definition, or individual XML items.

JAVA API

The java api is straightforward. The whole process involves reading a XML report, parsing it, generating the report, and writing to file system, as shown below:

```
File reportInputXML = new File("/local/input/reportInput.xml");
String report = FileUtils.readFileToString(reportInputXML, null);

File reportDir = new File("/local/output");
InputStream reportIn = new ByteArrayInputStream(report.getBytes());
Report r = rg.readReport(reportIn, reportDir, null);
File outputFile = new File(FileNameUtils.concat("/local/output", "report.pdf"));
outputFile.createNewFile();
ProcessedReport pr = rg.processReport(r, null);

rg.export(this.graphicFormat, pr, null, new FileOutputStream(outputFile));
```

DEPENDENCIES

The Report Generator mainly depends on Apache FOP (Formatting Objects Processor) library, which is “a print formatter driven by XSL formatting objects (XSL-FO) and an output independent formatter” [3]. There are also some other libraries which this application depends on, such as Apache Commons, etc. Non of the dependencies are large in size.

CONCLUSION

The XML based generalized report generator is light-weight in size, and comes with most commonly needed features. It's been successfully used in a number of Java based projects and proved to be adequate.

REFERENCES

All the web links are validated as of 29th, Aug, 2008.

[1] CSS (Cascade Style Sheet) <http://www.w3.org/Style/CSS/>

[2] W3C (World Wide Web Consortium) <http://www.w3.org/>

[3] Apache FOP <http://xmlgraphics.apache.org/fop/>

CONTACT INFORMATION

Your comments and questions are valued and encouraged. Contact the author at:

Author:	Fan Shao
Company:	Mango Solutions
Address:	Ground Floor, Unit 2 Greenways Business Park, Bellinger Close, Chippenham, Wiltshire, UK, SN15 3YR
Work Phone:	+44- (0)1249 767 700
Fax:	+44- (0)1249 767 707
Email:	fshao@mango-solutions.com
Web:	http://www.mango-solutions.com