jrxml2pdf - ein Reporting-Tool für Apex?

Ulrike Brenner
click-click IT Solutions e.U.
Perchtoldsdorf

Schlüsselworte
APEX, Reporting, PL_JRXML2PDF

Einleitung
Um aus Oracle APEX ein PDF zu erzeugen gibt es die verschiedensten Produkte und Reporting-Tools, wie z.B. BI Publisher, Jasper Reports, Oracle Reports, BIRT, PL/PDF usw..
Vor allem für kleinere Applikationen wäre es toll, wenn der Report dabei direkt in der Datenbank erzeugt und keine zusätzliche Middleware oder Java-tier benötigt wird. Und für das Design wäre eine graphische Umgebung sehr hilfreich, bei der einzelne Elemente mit drag-and-drop an die gewünschte Position verschoben werden, ohne großen Programmieraufwand und viel Code.
Genau diese Kombination bietet das von Andreas Weiden erzeugte Open Source Tool PL_jrxml2pdf.
Aber ist es auch praxistauglich?
Um auf diese Frage eine Antwort zu finden wird Frau Brenner anhand einer Live-Demo die Funktionalität von PL_jrxml2pdf vorstellen. Dabei werden neben klassischen Reporting-Anforderungen, sowohl Images und Charts, als auch Mehrsprachigkeit ein Thema sein.
Zum Schluss wird Fr. Brenner ein Beispiel aus der Praxis vorstellen, bei dem PL_jrxml2pdf eine bestehende Reporting-Lösung abgelöst hat.

PL_jrxml2pdf (http://sourceforge.net/projects/pljrxml2pdf/)
PL_jrxml2pdf ist ein rein PL/SQL-basierendes Tool, bei dem aufgrund einer XML-Definition in der Datenbank ein PDF erzeugt wird.
Die Definition des Layouts erfolgt im iReport Designer von JasperSoft und kann direkt in iReport getestet werden. Das dabei generierte jrxml wird in der Datenbank gespeichert und ohne zusätzliche Verwendung einer Reporting-Engine wird auch das PDF direkt in der Datenbank generiert. Ob das erzeugt BLOB dann mit APEX am Bildschirm ausgegeben, in einer Tabelle gespeichert oder als File zu einer E-Mail attachet wird, ist dem Entwickler freigestellt.

PL_jrxml2pdf war von Andreas Weiden (http://andreas.weiden.orcl.over-blog.de/) ursprünglich als „proof-of-concept“ gedacht, das sich aber zu einer vollwertigen Reporting-Lösung entwickelt hat.
APEX ist direkt in der Datenbank und daher liegt es auch nahe ein Reporting-Tool zu suchen, das auch direkt in der Datenbank das PDF generiert. Dafür gibt es verschiedene Werkzeuge (PL/PDF, AS_PDF_MINI u.a.), doch wird dabei das Layout durch Programmieren definiert, was zwischendurch etwas mühsam sein kann. Daher ist die Idee entstanden mit dem Package AS_PDF3 von Anton Scheffer zur Generierung des PDFs als Backend und einem xml-basierenden UI-Werkzeug als Frontend (iReport) eine „echte“ Reporting-Engine zu programmieren.

iReport Designer (http://sourceforge.net/projects/ireport/)
Mit iReport von JasperSoft steht ein graphischer Designer zu Verfügung, mit dem das Layout ‚gezeichnet’ werden kann. Als Datenquelle wird für PL_jrxml2pdf immer eine SQL-Query vorausgesetzt, also zumindest ein `SELECT 1 FROM DUAL` ist notwendig. Da die Query von PL_jrxml2pdf in der Datenbank geparst und per DBMS_SQL ausgeführt wird, stehen hier alle SQL-Möglichkeiten – abhängig von der zugrunde liegenden Datenbankversion – zur Verfügung.

Der iReport Designer unterteilt sich in folgende Bereiche:

- der Report Designer: hier wird das Design graphisch definiert; Die einzelnen Elemente werden per drag-and-drop positioniert bzw. verschoben
- der Report Inspector, zeigt die Struktur des gesamten Reports, inklusive aller Styledefinitionen, Parameter, Felder etc. an.
  Hier sind auch die Verschachtelungen der Regionen und Zugehörigkeiten zu Bändern ersichtlich.
- die Palette aller Elemente, in der alle möglichen Design-Elemente angezeigt werden (Text Field, Chart, Line, Break etc.)
- und dem Property Sheet: hier werden alle änderbaren Eigenschaften des ausgewählten Objects angezeigt.

Ein Report ist in einzelne „Bänder“ unterteilt, die entweder nur einmalig (z.B. Titel, Summary), am Beginn/Ende einer Seite, vor/nach der Detail-Region oder aber für jeden Ergebnisdatensatz der SQL-Query angezeigt werden. Je nachdem in welchem Band ein Objekt liegt, wird es nur einmal oder mehrfach angezeigt. Bei PL_jrxml2pdf ist zu beachten, dass folgende Regionen derzeit nicht unterstützt werden: Last page footer und no-data

Nach Auswahl der Datenbankverbindung und Eingabe einer SQL-Query stehen die Datenbankobjekte im Report Inspector zur Verfügung und können per drag-and-drop in dem jeweilige Band positioniert werden. Wird ein Spalte aus der SQL-Query in einem Detail-Band positioniert, so wird es als Textfeld erstellt und zusätzlich automatisch ein Label im „Header“ erzeugt. Dieser entspricht dem Spaltennamen.

Die einzelnen Objekte können zusätzlich zu Gruppen zusammengefasst werden, wobei diese, abhängig von logischen Bedingungen (Expressions) gedruckt werden können (z.B. ist es möglich, eine Region inklusive aller Felder nur dann anzuzeigen, wenn eine definierte Datenbankspalte NOT NULL ist).

.jrxml

Ist das Design des Reports fertig, wird eine jrxml-Definition erzeugt. In dieser ist sowohl das SQL-Query als auch die Definition jedes einzelnen Feldes enthalten.

z.B. die minimale Ausprägung eines Reports

```
<?xml version="1.0" encoding="UTF-8"?>
<jasperReport xmlns="http://jasperreports.sourceforge.net/jasperreports" xmlns:xsi="http://www.w3.org/2001/XMLSchema-i
    <queryString>
        <![CDATA[SELECT Dummy
  FROM DUAL]]>
    </queryString>
    <field name="DUMMY" class="java.lang.String"/>
    <title>
        <band height="79" splitType="Stretch"/>
    </title>
    <columnHeader>
        <band height="61" splitType="Stretch">
            <staticText>
                <reportElement uuid="1c30a3d4-8e6d-4b70-90f5-a8e0c31c35ce" x="0" y="0" width="100" height="20"/>
                <textElement/>
                <text><![CDATA[DUMMY]]></text>
            </staticText>
        </band>
    </columnHeader>
    <detail>
        <band height="125" splitType="Stretch">
            <textField>
                <reportElement uuid="38bf2804-1ad5-4e2e-8536-7d872b2cace0" x="0" y="0" width="100" height="20"/>
                <textElement/>
                <textFieldExpression><![CDATA[$F{DUMMY}]]></textFieldExpression>
            </textField>
        </band>
    </detail>
</jasperReport>
```

Erzeugen eines PDFs
In der Demo-APEX-Applikation, die mit PL_jrxml2pdf mit geliefert wird, gibt es direkt auf der ersten Seite die Möglichkeit einen neuen Report anzulegen. Dabei kopiert man die zuvor generierte jrxml-Definition in das Textfeld, gibt dem Report einen sprechenden Namen und kann sofort den neu erstellten Report ausführen (vorausgesetzt die Daten/Tabellen sind im Schema, in dem PL_jrxml2pdf installiert wurde, verfügbar).

Demo-Applikation
Bei der Installation von PL_jrxml2pdf gibt es 2 Möglichkeiten: entweder man installiert die mitgelieferte Demo-APEX-Applikation und legt mittels „Supporting Objects“ auch gleich alle Datenbankobjekte an, oder man installiert die Objekte (Tabellen, Packages, Konfigurationsdaten etc.) „by-hand“ mit Hilfe von sqlplus.
Vor allem bei ersten Versuchen mit PL_jrxml2pdf bietet die Applikation den Vorteil, sich bei den verschiedene Demo-Reports Ideen zu holen. Dabei können die zur Verfügung gestellten jrxml-Definitionen auch in den iReport Designer kopiert werden um so das Design zu erhalten.
Bei diese Demo-Reports wird auf die Oracle-DEMO_Tabellen (inklusive der Tabellen EMP und DEPT) zugegriffen.
Zusätzlich können – wie in der Live-Demo vorgestellt – selbst erstellte Reports sehr schnell ausgeführt werden.

Images
PL_jrxml2pdf unterstützt sowohl ‚statische’ Bilder (z.B. Logos), die in einer eigenen jrxml2pdf-Tabelle gespeichert werden, als auch dynamische (Daten-)Images, die in einem BLOB einer Tabelle gespeichert sind. Derzeit werden die Formate GIF, JPG und PNG unterstützt. Wichtig dabei ist, dass beim Bild kein ‚Alpha-Channel’ aktiviert ist (durchsichtiger Hintergrund).
Bilder (BLOBs) können - um für PL_jrxml2pdf eine lesbare Definition zu erhalten - nicht direkt aus dem Report Inspector in ein Band gezogen werden, sondern es wird, ein Image-Objekt aus der Palette in das gewünschte Band gezogen um dann die entsprechenden Konfigurationen in den Properties vorzunehmen. Sobald Datenbank-Bilder in iReport integriert wurden, können Reports allerdings nicht mehr im ‚Preview’-Modus des iReport Designers dargestellt werden.

Mehrsprachigkeit
In Zeiten der Globalisierung ist es immer öfter notwendig nicht nur Applikationen in mehreren Sprachen zur Verfügung zu stellen, sondern auch die PDFs in der Sprache des Kunden zu erstellen.
Die von PL_jrxml2pdf vorgesehene Lösung sind eigene Ressource-Files.
Mit der $R{}-Syntax werden die entsprechenden Labels aus den Ressource-Tabellen in der gewünschten Sprache ersetzt. Dabei kann ein Parameter für die gewünschte Sprache übergeben werden, oder aber es wird aus den NLS-Settings der Datenbank die Sprache gewählt.

Eine Alternative ist, direkt die Übersetzungs-Funktion von APEX zu verwenden. Dabei werden keine hardcodierten Labels verwendet, sondern jedes Label wird aus der Datenbank gelesen (ist somit auf PL_jrxml2pdf-Sicht ein ‚normales’ Textfeld). Der Vorteil dieser Lösung ist, dass es nur einen Übersetzungsmechanismus gibt (den von APEX) und die Übersetzungen im Report exakt gleich der Übersetzung der Applikation ist. Der Nachteil ist, dass viel mehr Daten in der SQL-Query selektiert werden müssen.

Fonts
Standardmäßig werden die Schriftarten Times New Roman, Arial, Courier New und WingDings unterstützt, d.h. bei diesen Sprachen reicht es in iReport den entsprechenden Schrifttyp auszuwählen und zu entscheiden, ob die Schrift fett, kursiv oder normal dargestellt wird.

Zusätzlich gibt es die Möglichkeit eigene TTF-Fonts zu laden. Bei TTF-Fonts wird pro Style (fett, kursiv, ...) ein eigenes File geladen wobei der Entwickler die Möglichkeit hat sich für eine komprimierte Speicherung zu entscheiden, was den Vorteil einer geringeren File-Größe hat. Oder aus Gründen der Performance die Fonts nicht compressed zu speichern.
Als „fall-back"-Schriftart wird Arial verwendet.

Master/Detail-Reports (Subreports)
Innerhalb eines Bandes kann ein Subreport aufgerufen werden. Ein Subreport ist eine eigenständige jrxml-Definition, die innerhalb des aufrufenden Reports gerendert wird. Somit kann eine Master/Detail-Darstellung erzeugt werden. Zwischen Master- und Detail-Report können Parameter übergeben werden.
Derzeit wird aus Performancegründen empfohlen nur einen Subreport pro Band zu verwenden.

Variablen in PL_jrxml2pdf
Seit der letzten Version von PL_jrxml2pdf gibt es auch die Möglichkeit Variablen z.B. für die Summierung von Beträgen zu verwenden. D.h. damit kann der Workaround, die gewünschten Aggregate in der SQL-Query zu selektieren, entfallen. Als Reset-Typen werden folgende Level angeboten: Report, Group, Page bzw. für jedes Record. Damit ist PL_jrxml2pdf wieder um ein wichtiges Element gewachsen.

Die hier beschriebenen Möglichkeiten sind nur ein Auszug der Möglichkeiten. Es werden viele der gängigen Formatierungsmöglichkeiten (Rahmen über Regionen, links/rechtsbündige Darstellung etc.) und Text-Objekte (Charts, HTML-Regionen, Fließtexte etc.) angeboten. Bisher sind uns keine Optionen abgegangen, und wenn doch, konnten sie durch einen Workaround gelöst werden.

Fazit
Mit PL-jrxml2pdf werden mit Hilfe eines grafischen Designers umfangreiche Berichte erstellt, die schnell und einfach ausgerollt werden können.

Der Vorteil, dass das PDF in der Datenbank generiert wird und wir somit keine Middleware für eine Reporting-Engine benötigen, kann sich auch als Nachteil erweisen. Das Erzeugen von PDFs (BLOBs) in der Datenbank kostet Zeit. Dieser Umstand ist vor allem dann zu berücksichtigen, wenn die Datenbank bereits den Flaschenhals im System darstellt.

Auf jeden Fall ist Andreas Weiden hier ein absolut sehenswertes Tool für kleinere und mittlere Instanzen gelungen, das es Wert ist ausprobiert zu werden.

Kontaktadresse:

Ulrike Brenner
click-click IT Solutions e.U.
Aspettenstraße 48
A-2380 Perchtoldsdorf

Telefon: +43 (0) 1 3119425 - 30
E-Mail ulrike.brenner@click-click.at
Internet: www.click-click.at