

# Excel Reporting mit Oracle Reports - mit Formatierung

**Volker Winter**  
**BASF-IT Services Consult GmbH**  
**Hamburg**

## **Schlüsselworte**

Oracle Reports Excel Layout Format HTML.

## **Einleitung**

Es besteht häufig der Wunsch nach Auswertungen aus der Datenbank, die dann gleich eine schön formatierte Excel Datei herausbringen sollen. Natürlich kann der SQL Developer Excel Ausgaben herstellen. Ebenso kann man mit Oracle Reports mit dem Ausgabotyp SPREADSHEET Exceldateien erzeugen. Alle haben aber den Nachteil, dass sie einfach den Tabellen- bzw. Abfrageinhalt ohne weitere Formatierung darstellen.

## **Excel Reporting mit Oracle Reports - mit Formatierung**

Der Ansatz in diesem Vortrag ist ein anderer: Ausgehend von einer Excelvorlage, in der umfangreichere Formatierungen, berechnete Felder, eine bedingte Formatierung usw. vorliegt, soll der Datenbankinhalt in diese Vorlage importiert werden, so dass bei Aufruf eine Exceldatei erstellt wird, die auf der einen Seite die Datenbankdaten enthält, auf der anderen Seite aber die Formatierungen der Vorlage weiterhin beinhaltet.

Für uns war es selbst überraschend, wie einfach es ist, diese Anforderung umzusetzen: Im wesentlichen wird die schön formatierte Vorlage genommen und als HTML gespeichert. Die HTML Source wird manuell in den Code Editor von Oracle Reports eingefügt, Taglibs eingebaut und dann eine SQL Query und ein Iterator eingepflegt. Wenn man zusätzlich den application-mimetype "Excel" angibt, wird die erzeugte HTML Datei gleich mit Excel geöffnet (wenn die erzeugte Dateiendung .xls ist und Windows .xls mit Excel öffnet). Anschließend wird die Datei als JSP von Oracle Reports aus gespeichert. Fertig! Alle Formatierungen sind beibehalten und die Daten an den gewünschten Stellen eingefügt.

Natürlich muss die JSP auf einen Application Server transportiert werden, der die Oracle Reports Taglibs auch versteht. Dort kann die JSP dann ganz normal über einen HTML Request aufgerufen werden.

## **Technische Umsetzung**

### ***Basis***

Es sollte grundsätzlich bekannt sein, was Java Server Pages (JSP's), Taglibs, Tags in HTML, Servlets usw. sind. Diese werden bei der Umsetzung verwendet und im Vortrag rudimentär besprochen.

### ***Vorgehen***

Im ersten Schritt wird eine Excel-Datei mit umfangreichen Formatierungen, Formeln usw. verwendet. Bedingung für die Exceldatei ist, dass ein Bereich vorgesehen wird, in den die Datenbankdaten, typischerweise in Tabellenform, eingefügt werden können.

Die Exceldatei kann dabei Kopfbereiche, Summenbereiche und auch Formeln, normale und bedingte Formatierungen, Feste Feldwerte usw. in den Feldern haben, die aus der Datenbank kommen. Es muss

lediglich ein „Block“ definiert werden, in dem mit Hilfe des Iterator Tags die Daten eingefügt werden können.

Ebenso ist es natürlich nicht notwendig, die erwartete Anzahl Zeilen bereits „leer“ in der Exceldatei vorliegen zu haben. Der Iterator erzeugt für jede Zeile aus der Datenbank eine neue Zeile in der Exceldatei. Die neue Zeile verwendet dabei die Formatierungen und Formeln für jedes Feld, so wie sie in der Vorlage für eine Zeile vorgegeben waren.

Beispiele für Tags aus der Oracle Reports Tag-Lib:

`<rw:dataArea>`

definiert einen Bereich, in dem Daten aus einer DataGroup verfügbar sind

`<rw:foreach>`

durchläuft die Datensätze der Data Group

`<rw:field>`

liefert den Inhalt eines Datenfeldes

`<rw:getValue>`

holt Werte aus einer Bean (Datenbankspalte)

### Beispielcode

#### 1. Einbau Tags

```
<tr>
  <td class=x12814827>&nbsp;</td>
  <td class=x12814827>&nbsp;</td>
</tr>
```

#### 2. Einbau Query

```
SELECT fc_get_label(:P_REPORT_NAME, 'Title', :P_LANGUAGE) RTLabel,
       fc_get_align(:P_REPORT_NAME, 'Title', :P_LANGUAGE) RTAlign,
       fc_get_valign(:P_REPORT_NAME, 'Title', :P_LANGUAGE) RTVAlign,
       max(rd.char_value_01) ProformaReference
FROM rer_data rd
WHERE rd.rer_instance_id = :P_INSTANCE_ID;
```

#### 3. Einbau Iterator

```
<rw:dataArea id="MGDATAGRPF41">
  <rw:foreach id="RGDATA411" src="G_DATA">
  ...
  </rw:foreach>
</rw:dataArea> <!-- id="MGDATAGRPF41" -->
```

#### 4. Einbau Fields

```
<td class=x13114827
  align="<rw:field id="FDESCRIPTIONALIGN166" src="DESCRIPTIONALIGN">
</rw:field>" valign="<rw:field id="FDESCRIPTIONVALIGN166"
src="DESCRIPTIONVALIGN"></rw:field>">
  <rw:field id="FDESCRIPTION166" src="DESCRIPTION"></rw:field>
</td>
```

## **Vorführung**

Im Vortrag werden wir, soweit es die Zeit erlaubt, folgende Arbeitsschritte live vorführen und erklären:

- Excel Beispiel-Datei zeigen (normales Excel Format mit Formatierungen, Formeln etc.) und als Single File Web Page speichern (.mht)
- .mht im einfachen Texteditor öffnen und kurz den Aufbau zeigen
- Vorbereiteten leeren Report im Reports Developer öffnen
- Datenmodell im Reports Builder erstellen (Query)
- Web Source zeigen
- Relevante Teile der mht in Web Source kopieren
- Data Source einfügen
- Kopieren in Reports Container
- Starten des Reports über eine Browser URL
- Anpassen der Source Datei (andere Formatierung, ...)

## **Entwicklungsumgebung**

- Oracle Reports Developer (hier Version 10.1.2.0.2 verwendet)
- Microsoft Excel
- Internet Explorer
- Texteditor
- Oracle Application Server 10g mit Oracle Reports Container

## **Kontaktadresse:**

Volker Winter  
BASF-IT Services Consult GmbH  
Christoph-Probst-Weg 26  
D-20251 Hamburg

Telefon: +49 (0) 40-41000-112  
Fax: +49 (0) 40-41000-100  
E-Mail: Volker.1.Winter@basf.com  
Internet: www.information-services.basf.com