

BI Publisher Style- und Subtemplates

Rainer Willems
Oracle Deutschland B.V. & Co. KG

Schlüsselworte:

BI Publisher, Templates, Styles, Style-Templates, Subtemplates, Boilerplate Templates

Einleitung

Style- und Subtemplates bieten eine weitere Modularisierung im BI Publisher. Während Style-Templates dazu da sind, einheitliche Formatierungen über Berichtsgrenzen hinweg festzulegen, können mit Subtemplates Bestandteile des Layouts ausgelagert werden. Dies kann zum einen zur besseren Übersichtlichkeit eines Templates dienen, zum anderen ermöglicht es die Wiederverwendung von Layoutbausteinen

Style-Templates

Ein Style-Template ist ein RTF-Dokument, in welchem verschiedene Layout-Informationen wie Farben, Fonts oder auch eine Kopf-/Fußzeile zur Wiederverwendung definiert werden können. Die so definierten Styles können dann zur Laufzeit RTF-Templates zugeordnet werden, damit diese das look-and-feel der Style-Templates erben. Somit lassen sich zentral Formate definieren und auf Layouts übertragen.

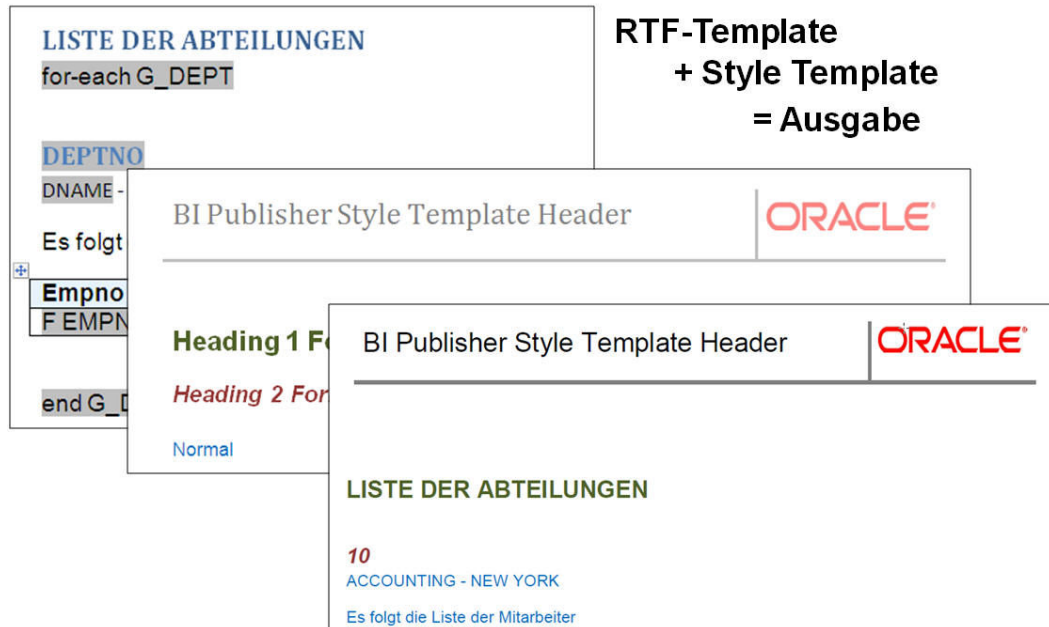


Abb. 1: RTF-Template + Style-Template = gewünschte Ausgabe

Hierzu definiert man im Style-Template in MS Word Paragraphen und Tabellen, formatiert diese wie gewünscht und versieht sie mit einem Namen. Dieser Style-Name wird später zum Übertragen der Layoutinformationen verwendet.



Abb. 2.: Verwendung von Style-Templates

Im Ziellayout definiert man nun seine Paragraphen und Tabellen und versieht diese mit ebendiesen Style-Name. Diese müssen hierzu nicht auf dem lokalen Rechner vorhanden sein, bzw. gar mit den gleichen Eigenschaften wie im Style-Template versehen sein, sondern lediglich der Style-Name spielt eine Rolle (siehe Abbildung 2).

Somit müssen dies Style-Definition nicht über Rechner-Grenzen hinweg auf alle Development-Clients verteilt werden. Im Server wird zur Laufzeit die Style-Definition aus dem Style-Template auf dementsprechende Bereiche des Ziellayouts übertragen. Beinhaltet das Style-Template eine Kopf- und/oder Fußzeile, werden diese ebenfalls übernommen. Änderungen am Style-Template ziehen bereits beim nächsten Berichtslauf. Das Metamodell sieht zwar schon einen Gültigkeitszeitraum vor, dieses kommt aber noch nicht zur Anwendung.

Style-Templates stellen einen neuen neuen Objekt-Typ im Katalog (siehe Abbildung 3) dar und werden als solche definiert. Zu erkennen sind diese an der Extension xss. Zur Internationalisierung lassen sich Sprach/Regions-abhängige Style-Templates verwenden.

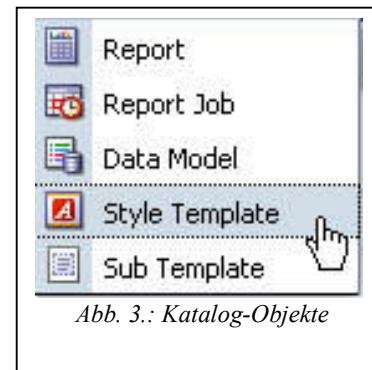


Abb. 3.: Katalog-Objekte

Die Zuordnung eines Style-Templates zu einem RTF-Template erfolgt in der Sicht „View a list“. Dort läßt sich ein Style-Template aus dem Katalog auswählen und per Checkbox den RTF-Template zuweisen (siehe Abbildung 4). In der Standard-Sicht auf die Templates finden sich diese Einstellungsmöglichkeiten nicht. Im BI Publisher Desktop kann man im Menu-Punkt Optionen ein Style-Template auswählen um bereits zum Entwicklungszeitpunkt die gewünschte Ausgabe zu erzeugen. Es ist allerdings nicht möglich unterschiedlichen Layouts eines Berichtes auch unterschiedliche Style-Templates zuzuordnen. Es kann nur ein Style-Template pro Bericht geben und hier kann pro RTF-Template entschieden werden, ob es zur Anwendung kommen soll oder nicht.

Benötigt man unterschiedliche Style-Templates, müssen die Layouts in unterschiedlichen Berichten sein.

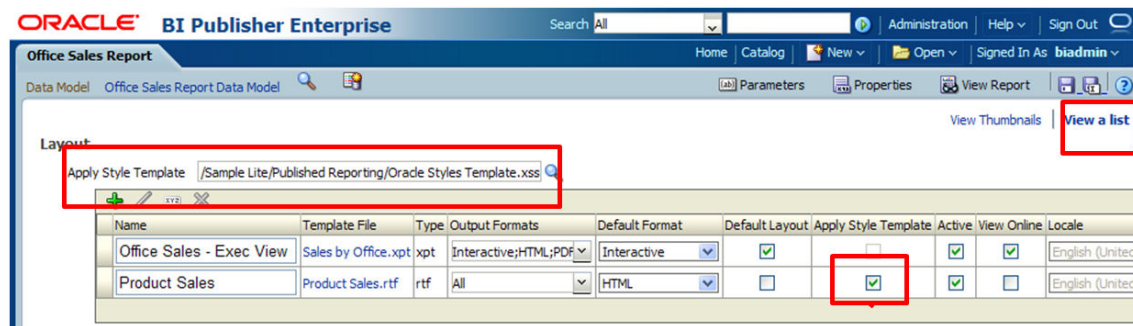


Abb. 4.: Zuordnung Style-Templates zu RTF-Templates

Boilerplate Templates

Style-Templates lassen sich nur auf RTF-Templates anwenden. Möchte man mit dem Online Template Builder im Browser arbeiten, gibt es eine solche Möglichkeit nicht. Hier kann man aber zumindest mit den sogenannten Boilerplate Templates einen Start für ein Online-Layout erstellen.

Es gibt bereits ein paar vordefinierte Boilerplate-Templates, die mit der Installation mitkommen. Diese liegen in den Shared Folders des Kataloges im Unterordner Components. Hier kann man über das Hinzufügen eines neuen Layouts auch eigene Boilerplate-Templates erstellen (siehe Abbildung 5). Diese sind zwar im Gegensatz zu Style-Templates nicht in der Lage, optische Merkmale für die Ausgabe von Daten festzulegen, aber zumindest läßt sich die gewünschte Layoutstruktur als Vorlage definieren. Diese beinhalten auch Kopf- und Fußzeilen oder Bilder.

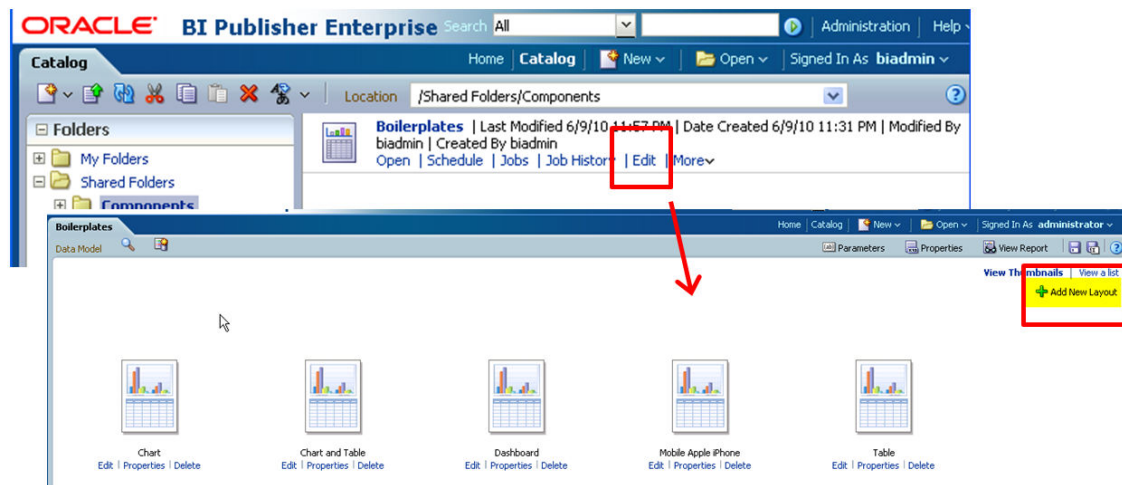


Abb. 5.: Definition von Boilerplate Templates

Subtemplates

Subtemplates sind wiederverwendbare Bausteine, die in RTF- oder XSL-Format vorgehalten werden können.



Abb. 6: Definition & Aufruf eines Subtemplates

Dies kann innerhalb eines RTF-Templates geschehen um die Übersichtlichkeit eines Templates zu erhöhen (z.B. falls eine komplexere Logik in einer kleinen Tabellenzelle untergebracht werden soll), oder einen mehrfach zu verwendenden Baustein nur einmal codieren zu müssen. Letzteres kommt aber vor allem in der Form vor, dass ein Baustein in mehreren Dokumenten benötigt wird. Hierbei kann es sich beispielsweise um Adress-Blöcke oder Kopf- und Fußzeilen handeln. Diese können dann auch in einem eigenen File gespeichert werden und von mehreren Layouts referenziert werden. So läßt sich zum einen eine einheitliche Formatierung durchsetzen (Beispiel: Adress-Block) oder Änderungen müssen nur an einer Stelle anstatt in vielen Dokumenten durchgeführt werden (Beispiel: Footer mit Firmeninformationen).

Ein File mit Subtemplates kann als eigenes Objekt im Katalog gespeichert werden. Dort ist das Objekt dann an der Extension xsb zu erkennen. Im Gegensatz zu den Style-Templates kann die Zuweisung zu den Layout-Vorlagen (noch) nicht deklarativ geschehen. Die Zuordnung geschieht mittels des Import-Befehls, der als erstes Kommando in der Layoutvorlage erscheinen muss. Im Falle der Verwendung eines Subtemplate-Files als Katalog-Objekt sieht die Referenz dann wie folgt aus:

```
<?import:xdoxsl:///pfad/name.xsb?>
```

Referenziert werden kann aber auch über eine File- oder HTTP-Referenz:

```
<?import:file:///pfad/name.ext?>
```

```
<?import:http:///pfad/name.ext?>
```

Zu beachten ist hierbei, dass das externe Referenzieren nach der Installation des BI Publishers aus Sicherheitsgründen erst einmal unterbunden wird. Die dementsprechende Eigenschaft muss also generell und für die betroffenen Berichte angepaßt werden (siehe Abbildung 7).

Disable variable header support	<input type="checkbox"/>	False
Add prefix to IDs when merging FO	<input type="checkbox"/>	False
Enable multithreading	<input type="checkbox"/>	False
Disable external references	<input checked="" type="checkbox"/>	True
FO Parsing Buffer Size	<input type="text" value="1000000"/>	1000000
Enable XSLT runtime optimization for	<input type="checkbox"/>	True

Abb. 7: Externe Referenzen zulassen

Generell kann in Subtemplates, auch wenn diese extern liegen, auf die Datenstrukturen des Berichtes zugegriffen werden. Im Falle von homogenen Benamungen läßt sich also beispielsweise eine Adress-Block recht einfach darstellen (siehe Abbildung 8).

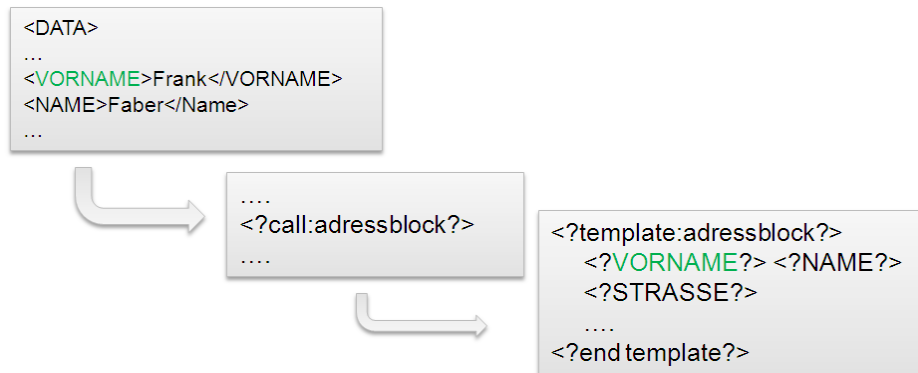


Abb. 8: Datenverwendung in Subtemplates - I

Möchte man ein solches Subtemplate aber für heterogene Benamungen verwenden, so kann man diese parametrisieren. Hierzu definiert man im Subtemplate Parameter, welche dann beim Aufruf des Subtemplates aus der Datenstruktur (oder auch statisch) belegt werden (siehe Abbildung 9)

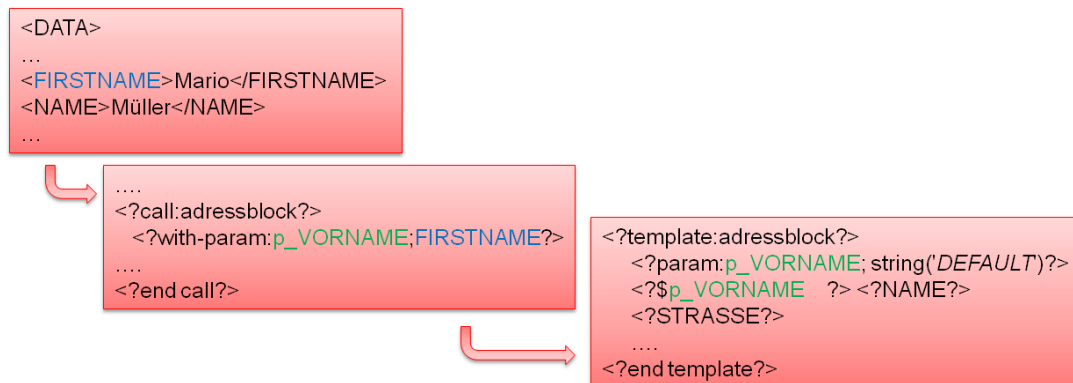


Abb.9: Datenverwendung in Subtemplates - II

Diese Parameter müssen natürlich nicht nur zur Ausgabe verwendet werden, sondern können auch für die dynamische Gestaltung eines Subtemplates (`<?if:$p_Parameter=...>`) dienen.

Kontaktadresse:

Rainer Willems
 Oracle Deutschland B.V. & Co.KG
 Robert-Bosch-Str. 5
 D-63303 Dreieich
 Telefon: +49 (0) 6103-397-290
 E-Mail: rainer.willems@oracle.com
 Internet: www.oracle.de