

# Im BI Publisher steckt mehr als erwartet

Frank Effenberger und Janine Lehmann, Robotron Datenbank-Software GmbH

*Das Reporting nimmt eine bedeutende Rolle bei der Prozess- und Entscheidungs-Unterstützung in datengetriebenen Unternehmen ein. Der Oracle Business Intelligence (BI) Publisher bietet die Möglichkeit, komplexe Berichtsszenarien über RTF-Templates abzubilden. Dessen Schnittstellen ermöglichen zusätzlich eine Integration beispielsweise in Apex-Anwendungen, um die automatische Generierung von Dokumentationen in anderen Projekten zu unterstützen. Dieser Artikel zeigt diese Integrationsmöglichkeit anhand eines Praxisbeispiels.*

Aufgrund der stetig steigenden Datenmengen in Unternehmen und der damit verbundenen Anforderungen an die Datenauswertung ist das Reporting beziehungsweise Berichtswesen in Bezug auf Effizienz und Qualität eine entscheidende Komponente. Ein besonderes Augenmerk der Endanwender liegt auf der Formular- und Dokumenten-Erstellung. Das Layout der Berichte und die zugrunde liegende Datenkomplexität bilden dabei zwei wichtige Aspekte.

Im Layout bestehen die Herausforderungen in der geforderten Einheitlichkeit, Übersichtlichkeit und Exaktheit von Berichten, da diese beispielsweise als Vorlagen oder zur Weitergabe für unterschiedliche Interessengruppen verwendet werden. Bei Dokumenten ist weiterhin auf eine allgemeingültige Struktur von Berichten hinsichtlich Inhaltsverzeichnis, Seitenangaben, Verweisen oder Diagrammen zu achten.

Aus Datensicht muss auf die Datenqualität, Verfügbarkeit und die richtige Modellierung geachtet werden. Dies ist ein wesentlicher Punkt für die zu erstellenden umfangreichen Dokumente oder detaillierten Formulare. Damit sind Layouts und die

steigende Datenkomplexität die wesentlichen Herausforderungen für ein Berichtstool.

Abbildung 1 zeigt, dass der BI Publisher die Bewältigung dieser Herausforderungen unterstützt. An dieser Stelle sei angemerkt, dass der Aspekt der Sprache beziehungsweise Übersetzung von Berichten in diesem Beitrag nicht adressiert, aber bei international zugänglichen Berichten durch den BI Publisher unterstützt wird. Die dargestellte konsequente Trennung von Layout und Daten ermöglicht selbst komplexe Reports.

## Komplexe Reports

Der BI Publisher bietet vielseitige Möglichkeiten der Berichterstellung. Um einen Bericht zu erzeugen, ist im ersten Schritt ein Datenmodell zu erstellen, um die individuellen Auswertungsanforderungen abzubilden. Als Datenquelle kann jede JDBC-fähige Datenbank dienen, etwa eine Oracle-Datenbank. Im zweiten Schritt werden SQL-Abfragen erstellt, die für die Darstellung der Daten themenbezogen oder hierarchisch aufgebaut sind. Im dritten Schritt ist ein Layout für den Bericht zu generieren, das sowohl die Datenfelder als auch die Formatierung des Berichts abbildet.

Mit dem im BI Publisher integrierten, benutzerfreundlichen und durch einen Wizard gesteuerten Layout-Editor können die Daten in der Web-Oberfläche ausgegeben und Nutzer-Interaktionen ermöglicht werden.

Die Praxis der vergangenen Jahre zeigte, dass die Web-Oberfläche des BI Publisher für einfache Berichte zwar geeignet, jedoch für die gewünschten komplexen Berichtsszenarien unzureichend ist. Dies ist mit dem Fehlen von benötigten Funktionalitäten zu begründen. Deshalb ist es empfehlenswert, das BI-Publisher-Add-in für Microsoft Word zu verwenden. Dazu muss zunächst der Template Builder installiert werden, mit dessen Hilfe detaillierte Templates im Rich-Text-Format (RTF) möglich sind. Die Verwendung des BI Publisher als strategische Reporting-Lösung im Rahmen spezifischer Anforderungen bietet mit den RTF-Templates eine hohe Flexibilität im Berichterstellungsprozess.

Vor der Erstellung eines Berichts mit dem BI-Publisher-Add-in erfolgt zunächst die umfangreiche Datenmodellierung, die als Grundlage für unterschiedliche Berichte dient, wodurch sich der Aufwand des weiteren Berichterstellungsprozesses verringert. Somit kann, wie im vorherigen Kapitel beschrieben, ein Datenmodell für mehrere Layouts verwendet werden. Man sollte bereits die wesentlichen Berechnungen und Zusammenhänge im Datenmodell abbilden, um die Komplexität bei der Layouterstellung zu minimieren. Für die Erstellung aufwändiger RTF-Templates sind zusätzlich Word-Mittel, Funktionen und Variablen nutzbar. Zur Vereinfachung der Layouterstellung sind Formatvorlagen geeignet. In mehrseitigen Dokumenten kann beispielsweise die Übersichtlichkeit durch eine Referenz der Seitenzahlen im Inhaltsverzeichnis gewährleistet und Datenhierarchien können über Grup-

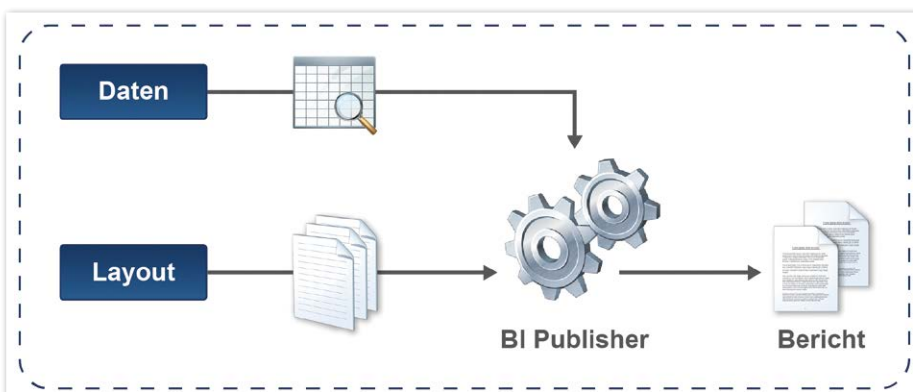


Abbildung 1: Vereinfachte Darstellung der Funktionsweise des BI Publisher

pierungen abgebildet werden. Zudem besteht die Möglichkeit, wiederkehrende und parametergestützte Bereiche eines Berichts in „Sections“ einzuteilen (siehe Abbildung 2).

Als hilfreiches Mittel bei der RTF-Erstellung dienen SQL- und XSL-Funktionen, die in BI-Publisher-Tags eingebunden sind. Als Beschreibungssprache kommen dabei XSL-Formatting Objects (XSL-FO) und XSL-Transformation (XSLT) des BI Publisher im Hintergrund zum Einsatz. Damit ist die Berichterstellung für Endanwender ohne weitreichende XML-Kenntnisse möglich.

Bei der späteren Ausführung des Berichts übersetzt der BI Publisher die Tags der Datenfelder in die standardmäßigen SQL- beziehungsweise XSL-Codes. XSLT lässt sich jedoch auch ohne Tags anwenden. Dies ist beispielsweise über XSL-Templates zu realisieren, die man in RTF-Templates referenziert. Zudem lassen sich auch einzelne Sektionen eines Berichts in andere Templates auslagern, um für weitere zu erstellende Berichte wiederverwendbar zu sein. Somit reduziert sich auch hierbei der Aufwand der Berichtsgenerierung. Darüber hinaus lassen sich Erweiterungen der Darstellungsmöglichkeiten von Diagrammen in Berichten oder die Verwendung von Variablen nur mit XSLT umsetzen. Dadurch ist jedoch ein breites Spektrum an Visualisierungsmöglichkei-

ten möglich, um eine schnellere Umsetzung von Kundenanforderungen zu erzielen.

Durch die Diversität an Ausgabeformaten lassen sich vielfältige Berichte generieren (siehe Abbildung 3). Hierbei kommen die Output-Formate PDF, HTML, Excel, PowerPoint, RTF, TXT oder CSV zum Einsatz. Für die erstellten Dokumente existieren viele Weiterleitungsmöglichkeiten. In diesem Kontext lassen sich beispielsweise Drucker, E-Mail- oder FTP-Server im BI Publisher einbinden und konfigurieren. Als Möglichkeit der umfangreichen Datenauswertung ist zudem die Integrierbarkeit der Dokumente auf Dashboards der Oracle Business Intelligence Suite zu erwähnen. Damit ergibt sich eine Interoperabilität, die die Ad-hoc-Analysen der BI Suite mit den komplexen Dokumenten des BI Publisher parametergesteuert vereint. Zudem ist die Einbindung in andere Anwendungen hervorzuheben, um Berichte entsprechend zu präsentieren. Damit all diese Funktionalitäten auch außerhalb des BI Publisher verwendet werden können, existieren mehrere Schnittstellen. Nachfolgend ist exemplarisch die Webservice-Schnittstelle näher erläutert.

### Webservice-Schnittstelle

Der BI Publisher kann über eine Webservice-Schnittstelle von heterogenen Anwendungen

verwendet werden. Der Vorteil der plattform-, programmiersprachen- und protokollunabhängigen Schnittstelle erlaubt dabei eine Vielzahl von Integrationsszenarien auf Basis XML-basierter Nachrichten (SOAP). Zur Nutzung der Webservice-Schnittstelle muss lediglich ein passender Webservice-Request an den BI Publisher gesendet und die Webservice-Response vom empfangenden Programm ausgewertet werden. Es existieren folgende Services, die verwendet werden können:

- *ReportService*  
Stellt Methoden bereit, um mit BI-Publisher-Berichtsobjekten zu interagieren
- *ScheduleService*  
Stellt Methoden bereit, um BI-Publisher-Scheduler-Aufgaben auszuführen
- *CatalogService*  
Stellt Methoden bereit, um BI-Publisher-Katalogobjekte zu verwalten
- *SecurityService*  
Stellt Methoden bereit, um Sicherheitsaufgaben zu lösen

Dabei ist es auch möglich, dass die Generierung eines Dokuments anhand geänderter Daten in der Datenbank stattfindet und so ein aktuelles Dokument auf Anfrage hin erzeugt wird (ReportService). Dies ist möglich, weil der BI Publisher zwischen den Daten und der Visualisierung trennt; so kann ein Layout fest hinterlegt, die Daten jedoch dynamisch angepasst und verändert werden.

### Einbindung in Apex

Es gibt bereits eine BI-Publisher-Integration in Apex. Dieser Beitrag und die folgenden Ausführungen adressieren jedoch explizit die Webservice-Schnittstelle des BI Publisher, ohne näher auf die Integrationslösung einzugehen. Dadurch wird die Funktionsweise und einfache Verwendung auch für andere Anwendungen verdeutlicht. Eine Apex-Anwendung kann die Webservice-Schnittstelle des BI Publisher verwenden, um etwa auf die Funktionalitäten der Dokumenten-Generierung zurückzugreifen. So kann für den Endanwender ein benötigtes Dokument erstellt werden, ohne dass dieser die eigentliche Anwendung verlassen und beispielsweise den BI Publisher öffnen und benutzen muss.

Die Apex-Anwendung benötigt dabei eine PL/SQL-Funktion, die den Webservice-Request (konkret: Report-Request) für ein

```

<?fo.page-number?> von <?fo.page-number-citation xdofo:lastpage-joinseq?>
<?xdoxslt.sysdate('DD.MM.YYYY')?>

<?for-each@section:BEREICHE?> INHALTSVERZEICHNIS
<?if.count(xdoxslt.distinct_values(BEZ_GESAMT))=1?>


|                  | UEB   | ANZ_TEILBEREICHE | SEITEN   |
|------------------|-------|------------------|----------|
| <b>BEREICH_1</b> | UEB_A | ANZAHL_A         | C3 - 3EC |
|                  | UEB_B | ANZAHL_B         | C3 - 9EC |
| <b>BEREICH_2</b> | UEB_D | ANZAHL_D         | C6 - 6EC |
|                  | UEB_E | ANZAHL_E         | C6 - 6EC |


<?end if?><?end for-each?>

```

Abbildung 2: RTF-Template mit SQL- und XSL-Funktionen, Sections und Variablen

Seite 1 von 1  
22.01.2016

### Überblick pro Jahr

	Überschrift	Teilbereiche	Seitenzahlen
<b>Aufgabengebiet 1</b>	Gebiet A	2	2 - 3
	Gebiet B	2	3 - 9
<b>Aufgabengebiet 2</b>	Gebiet D	8	9 - 34
	Gebiet E	5	34 - 56

Abbildung 3: Beispielhaftes Ergebnis eines Reports mit RTF-Template aus Abbildung 2

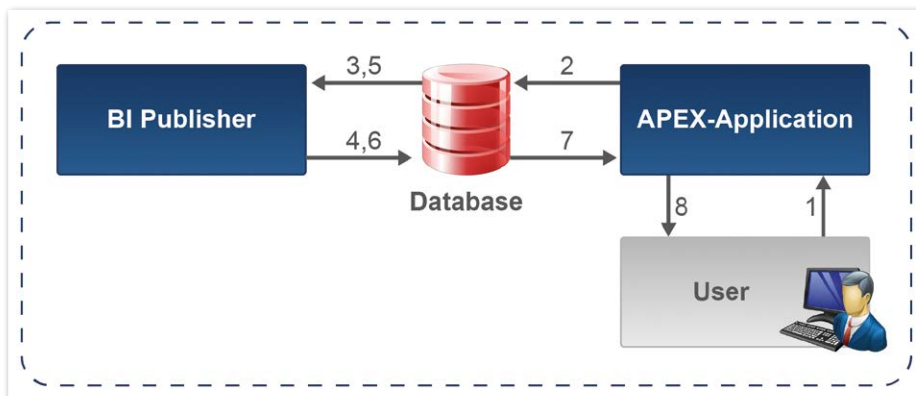


Abbildung 4: Ausführungssicht auf die Architektur für den Apex-Dokumentation-Generator

bestimmtes Dokument an den BI Publisher sendet und als Response eine binäre Datei erhält. Der Header der Datei in der Response muss dann auf das beauftragte Format gesetzt werden (etwa „application/pdf“ bei einem PDF-Dokument), um die Datei zum lesbaren Download bereitzustellen. Die Definition des Headers hängt dabei von der Art des angeforderten Dokuments ab. Das Zusammenwirken des BI Publisher und eines Apex-Projekts wird nun anhand eines konkreten und sich im Einsatz befindlichen Beispiels erklärt.

### Automatisierte Generierung von Dokumentationen

In einem Software-Projekt, das auch nach der initialen Produktübergabe weiterhin gewartet, entwickelt und damit vom Auftragnehmer unterstützt wird, finden in Absprache mit dem Auftraggeber regelmäßig Änderungen statt. Dadurch sind ursprünglich erstellte Anwender-Handbücher und Software-Dokumentationen meist schnell veraltet und haben nur noch eine bedingte Gültigkeit.

In Apex-Projekten existiert die Möglichkeit, Hilfetexte zu den jeweiligen Seiteninhalten zu generieren und zusätzlich in einer Datenbank abzulegen. Die Idee des Apex-Dokumentation-Generators liegt darin, diese Hilfetexte zu nutzen, um eine aktuelle und valide Anwender-Dokumentation für den Endbenutzer bereitzustellen. Zeitgleich kann dieses Werkzeug auch während der verschiedenen Entwicklungsphasen und Meilensteine vor der ersten Übergabe des Produkts genutzt werden, um intern die Verständlichkeit und Einheitlichkeit der Hilfetexte zu kontrollieren. Dadurch findet eine enge Kopplung der Anwender-Dokumentation an die Entwicklung statt und Änderungen müssen nur an einer zentralen Stelle anstatt an zwei Orten (Hilfetext

und das Dokument) gepflegt werden. Dies ist vor allem bei umfangreichen Software-Projekten von Vorteil.

Abbildung 4 zeigt die Architektur der Anwendung aus Sicht der Ausführung, bestehend aus einer Oracle-Datenbank mit installiertem Apex und dem BI Publisher. Die angegebenen Zahlen stellen die Schritte dar, die eine aktuell gültige Erstellung einer Anwendungsdokumentation erklären.

Schritt 1 stellt eine Dokumentationsanfrage des Nutzers dar, die in der Regel durch eine Schaltfläche in der Apex-Anwendung realisiert ist. Die Betätigung der Schaltfläche veranlasst die Apex-Anwendung dazu, intern PL/SQL-Funktionen und Prozeduren aufzurufen (Schritt 2). Der nächste Schritt beschreibt den innerhalb einer Funktion angestoßenen Webservice-Request (mit der „Make\_Request-Funktion“ des Package „Apex\_Web\_Service“) an den BI Publisher, mit dem die aktuelle Anwenderdokumentation als PDF bereitgestellt werden soll.

Der BI Publisher führt im vierten Schritt eine SQL-Query gegen die Datenbank aus und bekommt im fünften Schritt die aktuellen Hilfetexte in einer zuvor bereitgestellten Datenstruktur als Ergebnis der Abfrage zurück. Das PDF wird mit den erhaltenen Daten nun finalisiert und als Webservice-Response im sechsten Schritt zurück an die Datenbank-Funktion gegeben. Durch die Prozeduren und Funktionen wird nun von der Web-Oberfläche ein Dialog zum Download für das PDF bereitgestellt. Der User hat nun die Möglichkeit, die aktuelle Anwender-Dokumentation herunterzuladen und auf seinem Arbeitsgerät zu speichern.

Aus Sicht der Daten-Modellierung existieren drei Tabellen und zwei Views, die als Daten-Grundlage für den BI Publisher dienen. Die Tabelle „Kapitel“ stellt eine hierarchische Definition der verschiedenen

Kapitel der Anwender-Dokumentation dar. Im BI Publisher stellt die Tabelle „KAPITEL\_CBP“ eine View auf die Tabelle „Kapitel“ mit einem „Connect By Prior“-Befehl (CBP) dar. Da der BI Publisher den CBP nicht alleine ausführen kann, ist dieser Umweg über die View notwendig.

Jedes Kapitel hat eine individuelle Zuordnung zu verschiedenen Inhalten. Die Beziehung ist über die Tabellen „KAPITEL\_ZUORDNUNG“ und „INHALT“ dargestellt. Dieses generische Vorgehen erlaubt es, an jeder Position der Kapitel dynamische Inhalte in Form von Grafiken oder Texten zu hinterlegen. Dabei wird auch HTML- und CSS-Code berücksichtigt, der zum Beispiel in der Apex-Anwendung verwendet wird, um ein einheitliches Layout bereitzustellen. Um dies umzusetzen, besteht die Tabelle „INHALT“ aus einer View, in der Inhalte der Texte mit einer CSS der Apex-Anwendung verknüpft werden, damit Tabellen-Inhalte wie in der Anwendung dargestellt werden.

Auf Seite des RTF-Templates ist es aufgrund der dynamischen Erstellung der Anwender-Dokumentation empfehlenswert, das Deckblatt individuell je nach Kunde zu gestalten und das Template entsprechend anzupassen. Die Funktionalität des Inhaltsverzeichnis muss mithilfe des BI-Publisher-Add-ins zusätzlich eingebaut werden, ebenso wie automatisierte Referenzen in den Kopf- und Fußzeilen.

### Fazit

Der BI Publisher bietet eine Vielzahl von Funktionen, damit selbst komplexe Berichtsszenarien abgebildet werden können. Da für diese Szenarien die Web-Oberfläche des BI Publisher nicht ausreichend ist, muss auf RTF-Templates zurückgegriffen werden. Die Webservice-Schnittstelle des BI Publisher ermöglicht es, diese Funktionalitäten an andere heterogene Anwendungslandschaften zu koppeln. So sind beispielsweise automatische Dokumentations-Generierungen in Apex-Projekten möglich. Dies wurde anhand eines beim Kunden eingesetzten Projekts zur Erstellung eines stets aktuellen Anwenderhandbuchs demonstriert.

Frank Effenberger  
frank.effenberger@robotron.de

Janine Lehmann  
janine.lehmann@robotron.de