

The Winning Team - BI Publisher Integration in Oracle ADF

Jürgen Menge
Oracle Deutschland B.V. & Co. KG
München

Schlüsselworte

Oracle BI Publisher, Oracle ADF, Oracle JDeveloper, Web Service

Einleitung

Beim Oracle BI Publisher handelt es sich um eine leistungsfähige Lösung zur Definition und Generierung von Berichten bzw. Dokumenten. Obwohl Oracle Reports weiterhin Bestandteil der Fusion Middleware sein wird, ist der Oracle BI Publisher die strategische Reporting-Komponente, die bereits heute in vielen Oracle-Produkten zum Einsatz kommt (1).

Im Kontext der Anwendungsentwicklung ergibt sich häufig die Anforderung, bestimmte Berichte bzw. Dokumente aus der Anwendung heraus zu generieren. Die mit dem BI Publisher ausgelieferte grafische Bedienoberfläche kann dabei aus folgenden Gründen nicht zum Einsatz kommen:

- der Einsatzfall sieht eine enge Integration mit der Anwendung vor (z.B. automatische Übergabe von Parametern)
- eine Anpassung der grafischen Bedienoberfläche des BI Publisher an das Look&Feel der Anwendung ist weder dokumentiert noch wird sie offiziell unterstützt.

Aus diesem Grund muss es andere Mittel und Wege geben, die Erstellung der Berichte in die Anwendung zu integrieren.

Möglichkeiten der Integration

Da für den BI Publisher bereits bei der Entstehung des Produktes eine Integration in unterschiedliche Umgebungen vorgesehen war, gibt es mehrere, gut ausgebaute Möglichkeiten der Integration.

Schnittstelle		Besonderheiten	Remote Zugriff	BIP Server notwendig
HTTP-Aufruf		<ul style="list-style-type: none"> • kein Scheduling • nur synchrone Aufrufe • Verwendung bestehender Reports-Definitionen möglich • Authentifizierung oder Single Sign-On notwendig 	ja	ja
Java API	Java-Klassen	<ul style="list-style-type: none"> • Data Engine • Core API • Advanced API 	nein	nein
	Servlet, JSP/JSF		ja	nein
	EJB Session Bean		ja	nein
Web Service API		<ul style="list-style-type: none"> • Verwendung bestehender Reports-Definitionen möglich • Integration in .Net-Umgebungen möglich • Orchestrierung mittels BPEL möglich 	ja	ja

Aufruf der Berichte über HTTP-Request (URL)

Dabei handelt es sich um die am einfachsten umsetzbare Methode, die allerdings mit Einschränkungen verbunden ist. Da der Aufruf eines Berichtes im Security-Kontext des BI Publisher Servers ausgeführt wird, ist normalerweise eine erneute Authentifizierung des Benutzers (innerhalb der Applikation) notwendig. Um diese zu vermeiden, gibt es folgende Optionen:

- Username/Passwort beim Aufruf übergeben (Verwendung von SSL empfohlen)
- Speicherung der aufzurufenden Berichte im Guest Folder
- Verwendung einer Single Sign-On-Lösung (z.B. Oracle Access Manager)

Der BI Publisher gibt dabei eine gewissen Unterstützung, indem man sich verschiedene Varianten einer URL zum Aufruf eines Berichtes anzeigen lassen kann.

Aufruf des BI Publisher Java API

Die Besonderheit dieser Option besteht darin, dass sich der Anwender seinen eigenen Reporting-Server programmiert. Dem Vorteil einer sehr schlanken Lösung z.B. für eingebettete Applikationen stehen aber große Nachteile gegenüber:

- Bestehende Berichtsdefinitionen im BI Catalog können nicht genutzt werden.
- Die umfangreiche Funktionalität der ausgelieferten BI Publisher-Implementierung kann nicht genutzt werden.

Zudem wurde ab dem Release 11g die Klasse *DataProcessor* zur Erzeugung der XML-Daten aus dem Package *xdo-core.jar* in das Package *xdo-server.jar* verschoben, da es inzwischen Abhängigkeiten zu serverseitigen Funktionen aufweist. Damit fehlt eine wesentliche Funktion, um eine schlanke eingebettete Lösung zu entwickeln.

Aufruf der BI Publisher Web Services

Diese Möglichkeit wird sich für viele Anwendungsfälle als die am besten geeignete erweisen, da sie dem Anwendungsentwickler nahezu die gesamte Funktionalität des BI Publishers zur Verfügung stellt.(2)

Die Web Services existieren in zwei Implementierungen:

1. Als leichtgewichtige Web Services auf Basis von Apache Axis für Umgebungen, in denen nicht die komplette Infrastruktur der Oracle Fusion Middleware installiert ist. Eine Authentifizierung erfolgt entweder über Parameter bei jedem Aufruf einer Methode oder über ein Session Cookie, das zu Beginn einer Session vom BI Publisher Server angefordert wird.
2. Als Fusion Middleware Web Services für Umgebungen, in denen die komplette Infrastruktur der Oracle Fusion Middleware (Java Required Files) installiert ist. Die FMW Web Services verwenden Web Service Policies, um den Aufruf der Berichte abzusichern. Die FMW Web Services wurden bisher im Oracle Fusion-Projekt eingesetzt und sind bisher nicht offiziell dokumentiert.

Pull vs. Push

Bei der Erstellung von Berichten sind zwei grundsätzlich unterschiedliche Szenarien denkbar:

1. Pull (Lose Kopplung)
Aus der Anwendung wird ein vordefinierter Bericht aufgerufen, der die Daten aus dem zugehörigen Datenmodell bezieht. Einschränkende Parameter (Filter) können aus der Anwendung übergeben werden, sind aber nicht zwingend erforderlich. Es besteht kein direkter Zusammenhang zwischen der aufrufenden Anwendung und dem aufgerufenen Bericht.
2. Push (Enge Kopplung)
Aus der Anwendung wird ein vordefinierter Bericht aufgerufen, der die Daten aus der Anwendung bezieht und entsprechend dem festgelegten Layout-Template formatiert. Es erfolgt kein Zugriff auf eine bereits definierte Datenquelle (Data Source) des Berichts. Der fertige Bericht enthält Daten, die aus den vorhergehenden Schritten in der Anwendung resultieren.

Integration in Oracle ADF

Werden die Applikationen mit dem Oracle Application Development Framework (ADF) entwickelt, stehen verschiedene Möglichkeiten zur Verfügung, um BI Publisher-Berichte in die Anwendungen zu integrieren. Die ersten beiden Abschnitte behandeln die Integration über einen Push-Mechanismus.

Verwendung der Methode writeXML() des View Objects

Werden die ADF Business Components (ADFbc) verwendet, kann die Methode writeXML() die Daten eines View Objects (VO) als XML-Datenstrom zur Verfügung stellen. Anschließend kann entweder das Java API (3) oder das Web Service API des BI Publisher verwendet werden, um aus den XML-Daten des View Objects einen Bericht zu erzeugen. Da die XML-Daten direkt aus dem View Object bereitgestellt werden, ist die Verwendung der Klasse *DataProcessor* nicht erforderlich.

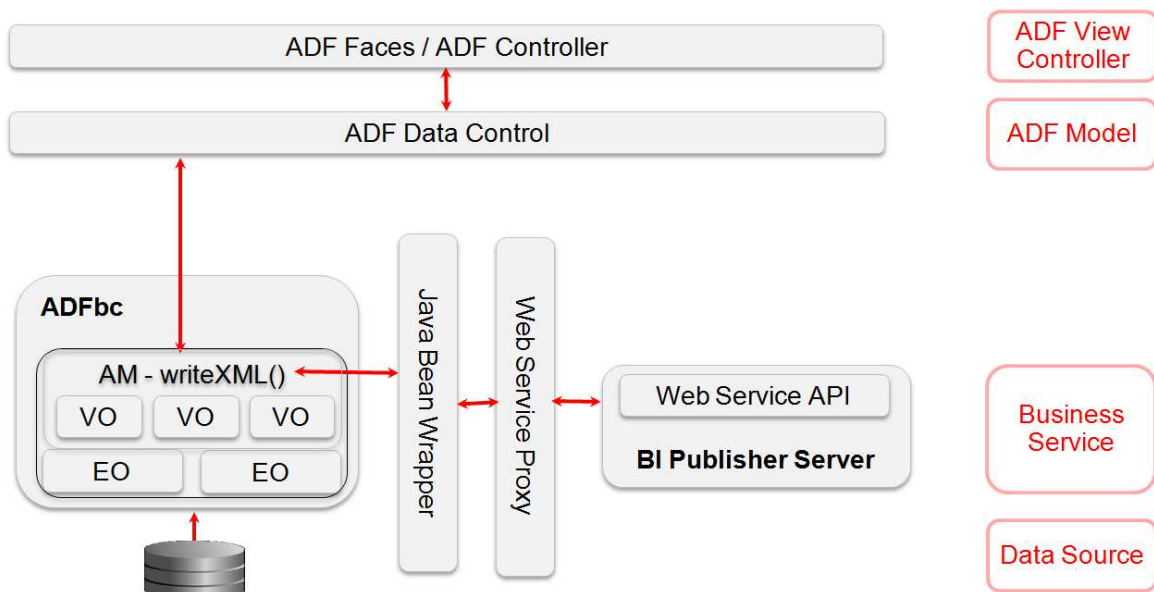


Abb. 1: Verwendung der Methode writeXML() des View Objects

Publizieren des View Objects als SDO Service

Werden die ADF Business Components (ADFbc) verwendet, ist es ebenfalls möglich, die Daten des View Objects (VO) als SDO Service zu publizieren. Wenn der SDO Service die XML-Daten bereitstellt, können das Java API oder das Web Service API des BI Publisher verwendet werden, um den gewünschten Bericht zu erzeugen.

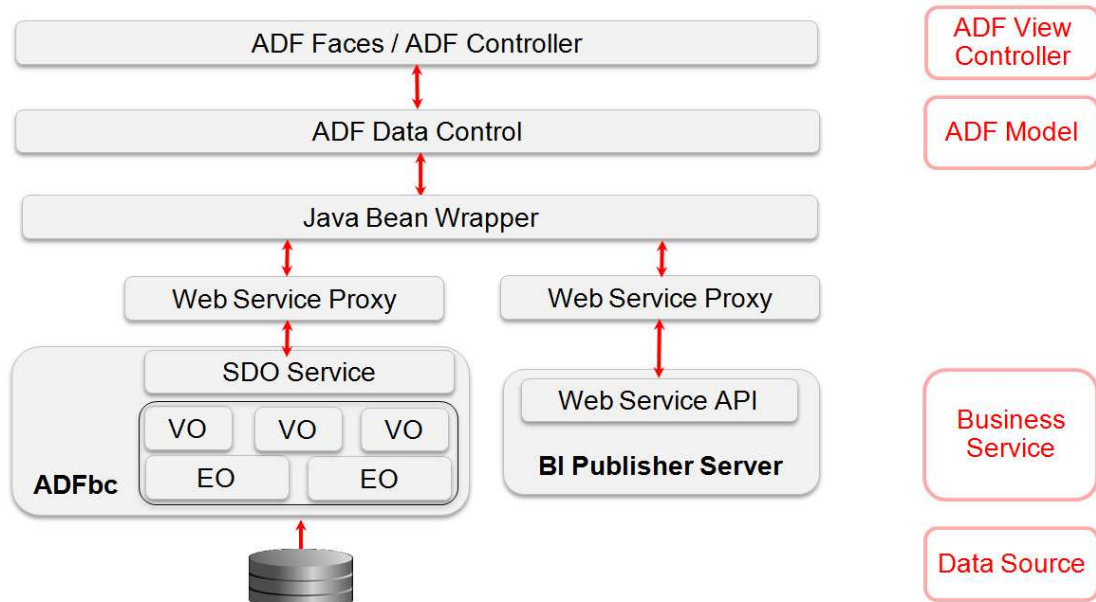


Abb. 2: Publizieren des View Objects als SDO Service

Verwendung der BI Publisher Web Services als ADF Business Service

Sollen vordefinierte BI Publisher-Berichte aufgerufen werden, so bietet sich das Web Service API des BI Publisher an. In diesem Fall werden die Daten aus den Datenquellen bereitgestellt, die im Datenmodell des Berichts definiert wurden (Pull-Mechanismus).

Innerhalb von Oracle ADF kann das Web Service Data Control oder besser eine Kombination von Web Service Proxy und POJO Data Control eingesetzt werden, um die ADF Model-Schicht zu erzeugen.

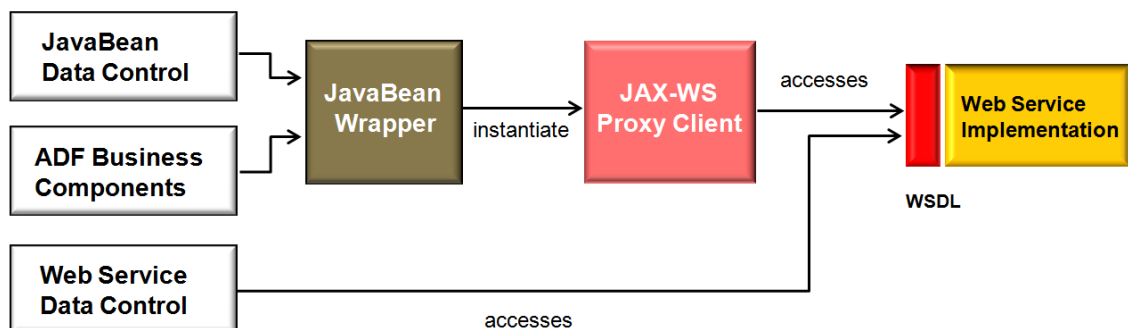


Abb.3: SOAP Service-Integration in Oracle ADF

Notwendige Anpassungen bzw. Einschränkungen werden beim Aufruf des Berichts über Parameter realisiert. Dabei kann es sich um:

- berichts-spezifische Parameter (Locale, Ausgabeformat, Template, Zielverzeichnis etc.)
- fach spezifische Parameter (Bind-Variablen)

handeln.

Da es zwischen den Parametern Abhängigkeiten gibt, so hängen z.B. die verfügbaren Ausgabeformate von der Wahl des Templates ab, sind in der Regel mehrere sequenzielle Web Service Calls zum BI Publisher Server erforderlich, um einen Bericht zu erzeugen.

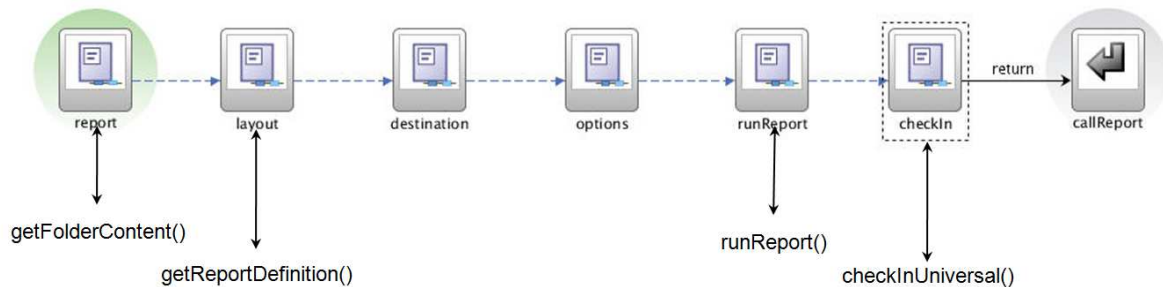


Abb.4: Sequenzielle Web Service-Aufrufe des BI Publisher Servers als ADF Bounded Taskflow

Aus Sicht des Application Development Frameworks bietet es sich an, den Berichtsaufwurf in einem Bounded Taskflow (BTF) zu kapseln. Diese Konzept nutzt die Stärken von Oracle DB und bietet folgende Vorteile:

- Der BTF kann als Train implementiert werden, der in einer Wizard-ähnlichen Benutzerführung den Anwender bei der Eingabe der Parameter unterstützt. Der Anwender bewegt sich Schritt für Schritt durch die Dialoge, bekommt am Ende eine Übersicht der ausgewählten Parameter und kann anschließend die Ausführung des Berichts starten.
- Der BTF kann mit Parametern versehen (z.B. URL des BI Publisher Server) als Bibliothek exportiert und anderen ADF-Projekten zur Verfügung gestellt werden. Die Entwickler der konsumierenden Applikationen brauchen keine Kenntnisse des BI Publisher Web Service API zu haben, um diese Funktionalität zu nutzen.

Verwendung eines View Objects als Datenquelle für einen Bericht

Ein spezieller Fall der Verwendung von ADF View Objects ist die Möglichkeit, ein View Object (VO) als Datenquelle für einen Bericht zu definieren. (4)

Diese Technik ermöglicht den Zugriff des BI Publishers auf die Daten eines VO über den Aufruf eines Web Service. Dazu muss das Application Module (AM) auf dem WebLogic Server deployed werden. Auf der ADF-Seite werden nur statische VO unterstützt. Eine dynamische Verwendung von View Criteria zur Laufzeit ist nicht möglich.

Auf der BI Publisher-Seite werden im Data Set Gruppenwechsel, Gruppenfunktionen und Data Links nicht unterstützt.

JDeveloper-Integration

Durch die beschriebene Verwendung einer ADF Library kann die Funktionalität des Berichtsaufrufs als wiederverwendbare Ressource im Resource Catalog des JDevelopers bereitgestellt werden. Zusätzlich gibt es im Release 11.1.1.6 und 11.1.1.7 des Oracle JDeveloper eine spezielle ADF Faces-Komponente, die als Region in einer Seite oder einem Seitenfragment implementiert werden kann und die den Aufruf eines Berichtes kapselt.(5)

Da diese Komponente zum gegenwärtigen Zeitpunkt die Fusion Middleware Web Services des BI Publishers benutzt, ist deren Anwendung auf Kunden beschränkt, die die Fusion Applications einsetzen.

Verweise

- (1) <http://www.oracle.com/technetwork/issue-archive/2010/toolssod-3-129969.pdf>
- (2) http://docs.oracle.com/cd/E28280_01/bi.1111/e22259/p1_webservices.htm
- (3) <http://husaindalal.blogspot.de/2009/11/integrating-bi-publisher-standalone.html>
- (4) http://docs.oracle.com/cd/E23549_01/bi.1111/e13880/T526682T565378.htm
- (5) http://docs.oracle.com/cd/E28271_01/fusionapps.1111/e26385/jdev.htm#BIPDV015

Kontaktadresse:

Dr. Jürgen Menge
Oracle B.V. & Co. KG
Riesstr. 25
D-80992 München

Telefon: +49 (0) 89-1430 2239
Fax: +49 (0) 89-1430 2150
E-Mail: juergen.menge@oracle.com
Internet: www.oracle.com