

Anwendungsmigration

Besser ein Blick auf's Kleingedruckte

Stephan La Rocca
TEAM GmbH
Paderborn

Schlüsselworte:

Migration, Oracle Reports, Oracle BI Publisher, Report as a Service (RaaS), Integration Oracle Forms, APEX, ADF

Einleitung

Begeben sie sich im Internet auf die Suche nach Unterstützung für eine Migration von Oracle Forms, werden ihnen mehr als zweieinhalb Millionen Seiten angeboten. Eine Migration von Oracle Reports bringt gerade mal etwas mehr als 400.000 Fundstellen. Die Treffermenge mag ein Maßstab für die Bedeutung aber auch für den Fokus von Migrationsprojekten widerspiegeln. Vielfach wird das Thema „Print-Services“ erst im Laufe oder gar am Ende des Migrationsprojektes betrachtet, weil es ja „nur“ darum geht, die Daten auf das Papier zu bringen. Anforderungen, Möglichkeiten und Aufwände sollen in diesem Vortrag beleuchtet werden

Aufgaben an einen Druckservice

Wenn auch der Ansatz „Reporting as a Service“ in einem Unternehmen sich noch nicht hat vollständig etablieren können, so übernimmt doch der Oracle Report Server bereits heute wesentlich mehr Aufgaben, als „nur“ Papier zu bedrucken. Er ist fester Bestandteil und nicht nur Ende einer Geschäftsprozesskette. Neben der kritischen Verfügbarkeit des Services (z.B. für Barcodes in der Logistik oder Produktion), werden häufig weitere Prozessschritte angestoßen, Schnittstellen befüllt, externe Systeme mit Dateien beliefert

In der klassischen Client-Server Architektur, in der Oracle Reports seine Anfänge hatte, war es selbstverständlich, dass der Reports-Prozess auf dem lokalen PC ausgeführt wurde und somit zum Einen auf die lokalen Drucker (natürlich auch Netzwerkdrucker) zugreifen konnte und zum anderen in der „Obhut“ des Anwenders lag. Sicherlich gab es auch Installationen, bei denen auf dem Datenbank-Server ein Reports-Client installiert wurde, um über einen Cron-Job oder Windows-Task zeitlich gesteuerte Reports ausführen zu können.

Von dieser Sicht auf die Anwendungsarchitektur haben sich viele Installationen nicht wirklich befreien können und befinden sich heute vor der Fragestellung, wie ein Reportsserver auf die lokalen Drucker zugreifen kann. Und das in einem WAN.

Ein Versuch, die Anforderungen an den Druckservice in einer 3-Tier-Architektur zu klassifizieren, könnte in folgender Einteilung münden:

- 1.) Wo wird das Dokument genutzt? In dieser Frage kann zwischen Ausgaben auf dem Client (Anzeigen von druckerfreundlichen Dokumenten in der Applikation/dem Browser und der Ansteuerung lokal angeschlossener Drucker) und Ausgaben auf dem Server (Direktdruck auf Netzwerkdrucker oder Erstellen von Dateien, die weiterverarbeitet werden)
- 2.) Wie wird das Dokument erstellt? Hier soll die Aktion betrachtet werden, die den Druck initiiert. Das kann eine Interaktion des Anwenders sein, ein zeitlicher Rhythmus oder eine Aktion eines anderen Prozesses.

- 3.) Wie wird das Dokument genutzt? Wie in den Beispielen bereits erwähnt, kann das erstellte Dokument zur Druckausgabe genutzt werden, oder als Datei für weitere Prozessschritte. Die dazu notwendige Integration in die Prozessschritte kann über SOA-Konzepte erfolgen oder schlicht als Datei, die via FTP oder E-Mail weitergereicht wird.

Je nach Kombination der obigen Klassifizierung können nun unterschiedliche Druckszenarien erzeugt werden. Hinzu kommt, dass Druckausgaben in Form (z.B. Barcode, OCR-Fonts, etc) , Geschwindigkeit (Etikettierstraßen, Label in der Produktion, etc) und Layout (Formulardruck, Schnittstellenformate, etc) durchaus zusätzliche Anforderungen an den gesamten Prozess erstellen.

Reports as a Service

Durch den Architekturwandel hin zur 3-Tier-Architektur und damit einziehenden Konsolidierung der Anwendungen auf einem Applikation Server bietet sich auch an, über die Konsolidierung der Reports-Anforderungen nachzudenken. Bevor jede Anwendung für sich diese Aufgaben löst und die dazu notwendige Infrastruktur bereitstellen muss, ist es effizienter einen Dienst als Druckservice einzurichten.

Bei dieser Begriffseinführung geht es weniger darum, die „ServiceLeistungen“ aus der Firma auszugliedern, sondern Applikationsunabhängig einen Service für die Druckausgaben anzubieten. Dazu muss es möglich sein, den Service über klassische WebServices, Java-Apis oder anderen Schnittstellen in die eigenen Applikationen einbinden zu können.

Zentrale Aufgaben, wie die Einbindung der Drucker und Schnittstellen, Autorisierung und Authentifizierung, Logging, Loadbalancing und Hochverfügbarkeit sind dann einmalig vom Reportservice realisiert.

Das Oracle Produkt BI Publisher bietet auf Grund seiner Architektur und der Vielzahl der Integrationswege eine ideale Grundlage, einen solchen Service aufzubauen.

In einem ersten Schritt muss dabei die Migration der vorhandenen Oracle Reports betrachtet werden, bevor die verschiedenen Integrationen in ADF, Apex und Forms beleuchtet werden.

Migration Oracle Reports zu BI Publisher

Im Rahmen bereits durchgeführter Migrationsprojekte hat Oracle bereits ein Verfahren zur Migration unter dem Link http://docs.oracle.com/cd/E10415_01/doc/bi.1013/e10416/converttrpts.htm im Rahmen der Dokumentation veröffentlicht.

Die Migration erfolgt in mehreren Schritten und beginnt mit der Umwandlung der Oracle Reports in eine XML-Datei. Diese wird von einem Java-Programm des BI Publishers in die verschiedenen Bereiche Layout, Datenmodell und PL/SQL-Package-Code für eine Datenbank-Package aufbereitet. Das folgende Bild zeigt schematisch den Migrationsprozess

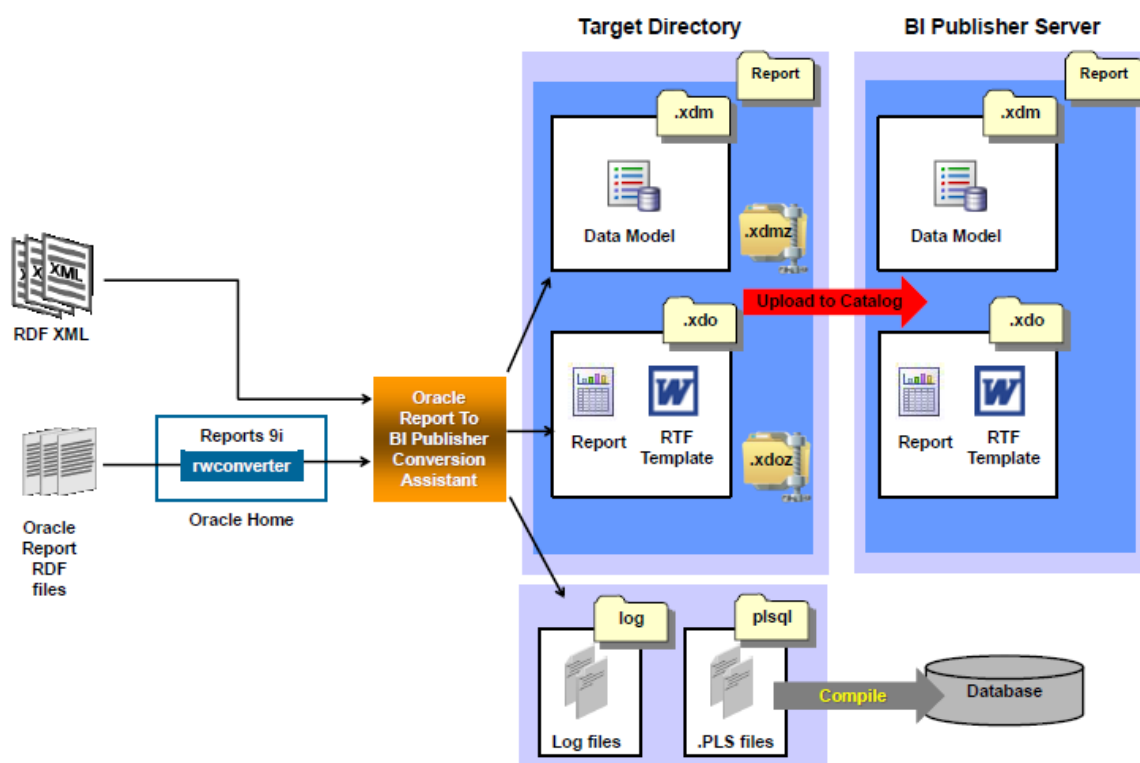


Abb. 1: Reports zu BI Publisher Migrationstool

Seit Anfang Mai 2012 steht auch für die 11g Version des BI Publishers eine Version dieses Tools zum Download bereit. Dieses Werkzeug ist im Gegensatz zur 10er Version dialoggesteuert, jedoch bleiben allerdings hier noch Aufgaben zur manuellen Nachbereitung gerade bei Formelspalten im Datenmodell und Ausnahmenformatierungen mit PL/SQL-Code.

Integration Oracle BI Publisher

Für den Zugriff auf die Reportservices gibt es je nach Anwendungsentwicklung unterschiedliche Möglichkeiten. Starten wir mit Legacy-Applikationen, die mit Oracle Forms implementiert sind. Im Rahmen einer schrittweisen Migration und unter Verwendung der Vorteile des BI Publishers ist eine solche Konstellation sicher zu betrachten. Im Gegensatz zur Integration, die von Herrn Menge und Herrn Willems von Oracle im Dokument (<http://www.oracle.com/technetwork/middleware/bi-publisher/overview/forms-bip-v22-129995.pdf>) beschrieben ist, sei hier die Alternative beleuchtet, dass eine Integration über die Java-Beans des Forms-Applets erfolgt. In der Java-Bean wird ein Webservice-Client in Forms integriert, so dass der BI Publisher Webservice nicht auf dem Applikation Server konsumiert wird, sondern auf dem Client. Das führt zu einer initial längeren Ladezeit des Forms-Applets (das jar wird aber gecached) reduziert aber dramatisch die Last auf dem Applikation Server bei hohem Druckaufkommen.

Über die die Nutzung des WebServices kann auch eine Integration des BI Publishers in jede ADF Applikation erfolgen. Hier haben wir bereits in dem DOAG Vortrag (http://www.team-pb.de/files/Oracle/doag-adf_bi-publisher.pdf) die beiden Alternativen des WebServices oder der JAVA-API gegenüber gestellt.

Durch die wachsende Verbreitung der APEX-Applikationen, wird eine Integration auch in diese Anwendungsentwicklung interessant. Hier verfolgt Oracle eine sehr enge Integration, die bereits über die Konfigurationsmöglichkeiten von APEX vorbereitet ist.

Abb. 2: Konfiguration APEX

Die weitere Integration erfolgt über die bekannten APEX-Wizards. Der Aufruf des BI Publishers ist schlussendlich ein Aufruf über eine URL in der alle Parameter an die BI-Publisher-Applikation übergeben werden.

Integrationen in die Oracle BI Produktfamilie sei hier nur erwähnt, ist aber nicht Gegenstand der Applikationsentwicklung. Allerdings ist es ein Grund mehr, auch darüber eine bereits vorhandene Infrastruktur für Druckservices zu nutzen.

Kontaktadresse:

Stephan La Rocca
TEAM GmbH
Hermann-Löns-Straße 88
D-33104 Paderborn

Telefon: +49 (0) 5254-8008 0
Fax: +49 (0) 5254-8008 19
E-Mail: sr@team-pb.de
Internet: www.team-pb.de