

Wie der BI Publisher in Oracle Forms eingebunden werden kann, wurde von Oracle schon vor längerer Zeit in einem Whitepaper beschrieben [1]. Dieser Artikel zeigt eine für den Applikations-Server ressourcenschonende Alternative des Zusammenspiels zwischen Forms und dem BI Publisher auf.

Alternative Einbindung des BI Publisher in Forms

Stephan La Rocca und Christian Piasecki, TEAM GmbH

Bei einem Kunden mit einer gewachsenen Infrastruktur- und System-Landschaft entstand der Bedarf an einer neuen, unternehmensrelevanten Applikation. Aus der Historie heraus besteht die vorliegende Anwendungs-Landschaft nahezu durchgängig aus Forms-Anwendungen, die auf Basis eines Oracle-Applikations-Servers betrieben werden. Das Forms-Know-how ist stark ausgeprägt und der Kunde will die Pflege der entwickelten Anwendungen selbst fortführen. Auf Basis dieser Randbedingungen entschied man sich erneut für eine Forms-Oberfläche mit einer Oracle-Datenbank als Daten-Basis. Die Anwendung realisiert eine Adressverwaltung mit hochsensiblen Datenbeständen, aus der es den Benutzern möglich sein soll, einzelne oder mehrere Personen unkompliziert per Post oder E-Mail anzuschreiben. Zusätzlich sollen die Adressen für einen Etikettendruck zur Verfügung stehen und exportiert werden können.

Der BI Publisher kommt ins Spiel

Die genannten Reporting-Anforderungen lassen instinktiv an den BI Publisher als passgenaues Werkzeug denken. Durch seine Vorteile, verschiedene Ausgabekanäle mit unterschiedlichen Ausgabeformaten zu bedienen, platziert er sich für die Aufgabenstellung als geeignete Lösung. Insbesondere bedient er mühelos die unterschiedlichen Aufgabenstellungen, etwa einem Endbenutzern zu ermöglichen, Briefe sowohl als PDF-Ausgabe direkt auf einem Drucker ausgeben zu lassen, wenn gewünscht, Personen direkt per E-Mail anzuschreiben oder, abhängig von der Kontaktadresse der Person, beide Aufgaben gleichzeitig zu erledigen.

Da die Benutzer für diese Aufgaben nicht zwischen der Web-Oberfläche des BI Publisher und der Forms-Applikation wechseln sollen, sollten der BI Publisher in die Forms-Anwendung integriert und das Zusammenspiel zwischen Benutzern, Forms und BI Publisher umgesetzt werden. Dieses Zusammenspiel lässt sich in mehrere Aspekte unterteilen.

Berichts- und Vorlagen-Erstellung nur für Administratoren

Neue Berichte und Layout-Vorlagen sollen nur durch Administratoren erstellt werden, der normale Benutzer darf in der Anwendung nur aus den verschiedenen Berichten und Layout-Vorlagen auswählen. Da es Administratoren zumutbar ist, mit der Web-Oberfläche des BI Publisher zu arbeiten, wurden hier keine Anpassungen vorgenommen, sondern die Möglichkeiten genutzt, die die Web-Oberfläche und der Word-Template-Builder bieten. Das war der einfache Part.

Im nächsten Schritt ging es darum, den BI Publisher an Forms anzubinden und den Benutzern das Erstellen von Briefen, Berichten, E-Mails etc. zu ermöglichen.

Die Endbenutzer sollten diese über die Forms-Anwendung möglichst komfortabel erstellen können, indem sie einfach Kontakte, einen Bericht und Layout auswählen und dann aus Forms heraus nur noch das Generieren des Reports anstoßen.

Nutzung des Java-API als MBean

Als Erstes wurde die Lösung aus dem genannten Whitepaper [1] von Oracle über die Nutzung des BI Publisher in Forms eruiert. Die beschriebene Lö-

sung für diesen Use-Case besitzt einen Nachteil: Durch die hier beschriebene Implementierung wäre jeder neu angestoßene Bericht ein neuer JVM-Prozess auf dem Applikations-Server und könnte so bei Benutzern zu Performance-Problemen führen. Um dieses zu umgehen, fiel die Entscheidung auf die Möglichkeit, alles in eine MBean zu kapseln, diese in Forms einzubinden und so den JVM-Prozess auf dem Client ablaufen zu lassen.

Oracle Forms bietet seit Einführung der Web-Variante die Möglichkeit, Java an verschiedenen Stellen einzubinden. Das sind im Einzelnen die Nutzung von Pluggable Java Components, Java Beans oder von PL/SQL-Wrappern. Der BI Publisher stellt ein Java-API zur Verfügung, das sowohl über einen PL/SQL-Wrapper in Forms-PL/SQL-Code eingebunden werden kann (so wie es in dem erwähnten Papier beschrieben wurde) oder aber auch für eine Java Bean ansprechbar ist. Dieser Java Bean wird in Forms eine Implementierungsklasse zugewiesen, die die gesamte Kommunikation zwischen den Forms-Properties und den Attributen des Java-API übernimmt.

Mit den Built-ins „SET_CUSTOM_PROPERTY“ und „GET_CUSTOM_PROPERTY“ können aus der Java Bean Methoden aufgerufen oder Werte abgefragt werden. Die Implementierungsklasse bereitet diese Informationen dann für das BI-Publisher-API auf.

In der Konfiguration des Forms-Servers werden die Implementierungsklasse und das Java-API als Jar-Archiv signiert, ähnlich wie schon für die von Oracle beigestellten Web-Utilities, und zum Download auf den Client zur Verfügung gestellt. Nach dem ersten

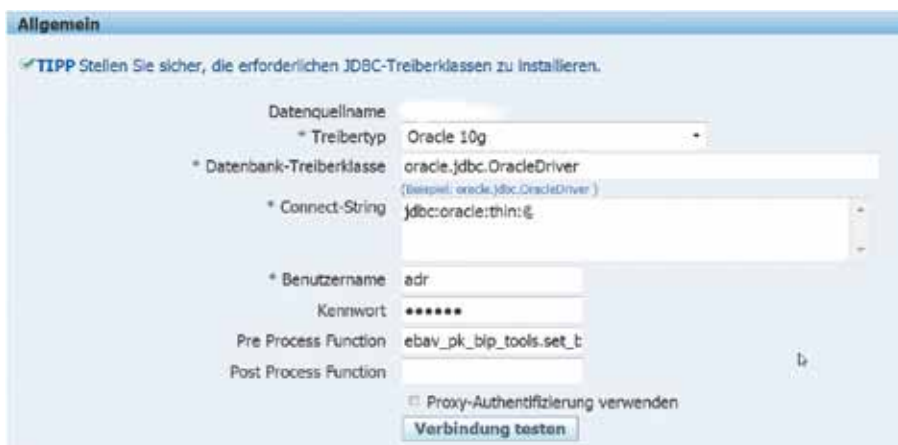


Abbildung 1: Der Pre-Trigger

Download auf den Client übernimmt das BI-Publisher-API die Kommunikation zum BI-Publisher-Server und somit ist der Applikations-Server von dieser Last und dem Traffic befreit.

Pre-Trigger und Bursting

Zwei weitere BI-Publisher-Features, die in diesem Projekt genutzt werden, sind die Pre-Trigger der Datenquellen und das Bursting. Die Pre-Trigger der Datenquellen sind perfekt dafür geeignet, den Kontext zu setzen, in dem der BI Publisher unterwegs ist.

Da in der Datenbank-Verbindung explizit ein Benutzer für den Datenbank-Connect eingegeben werden muss, es aber sicherzustellen gilt, dass jeder Forms-Anwender nur seine eigenen Daten beziehungsweise die Daten, an denen er Rechte besitzt, sehen darf, wird in dem Trigger über eine Datenbank-Funktion der Endbenutzer an die Datenbank weitergegeben, um so die Daten zu filtern (siehe Abbildung 1).

Außerdem ist der Einsatz der Bursting-Funktionalität entscheidend für das Projekt. In Abhängigkeit von der Adress-Art (E-Mail, Brief etc.) müssen verschiedene Ausgabekanäle des BI Publisher angesprochen und zudem unterschiedliche Layout-Vorlagen für die Dokumente genutzt werden. Um dieses möglichst komfortabel für den Endbenutzer zu gestalten, werden die Berichte durch die Bursting-Funktionalität aufgesplittet und mit den passenden Layout-Vorlagen an die entsprechenden Ausgabekanäle verteilt.

Fazit

Durch diese Implementierung und den Einsatz der verschiedenen BI-Publisher-Funktionen ist es gelungen, eine Forms-Anwendung mit einer modernen Reporting-Lösung zu schaffen. Zudem ist es dem Kunden nun möglich, den BI Publisher in Zukunft als zentrales Reporting-Tool in seiner bestehenden System-Landschaft zu nutzen.

Literaturhinweis

- [1] How to integrate Oracle BI Publisher via Web Services in Oracle Forms, 2008, Dr. Jürgen Menge, Rainer Willems

Stephan La Rocca
sr@team-pb.de



Christian Piasecki
cpi@team-pb.de



■ Neu: Oracle stellt Database Appliance X3-2 vor

Die neue Version bietet zweimal höhere Leistungsfähigkeit und viermal mehr Storage-Kapazität im Vergleich zu früher. Kunden können damit das verfügbare Datenvolumen umgehend vergrößern, indem sie einfach ein Storage Expansion Shelf anschließen, ohne dass weitere Verwaltungsaufgaben notwendig sind. Zudem enthält Oracle Database Appliance X3-2 optional eine virtualisierte Plattform, um auf Basis von Oracle VM mit der ISV vollständige Out-of-the-Box-Lösungen paketieren und ausliefern können.

Die Oracle Database Appliance Virtual Platform nutzt die Möglichkeiten von Oracle VM zum Hard-Partitioning. Außerdem erweitert sie das „Pay-as-you-grow“-Software-Lizenzierungsmodell auf sämtliche Software von Oracle.

Bei der Oracle Database Appliance handelt es sich um ein vollständiges Paket aus Software, Server, Storage und Netzwerk-Komponenten, das auf Einfachheit und Hochverfügbarkeit hin zusammen entwickelt wurde. Damit senken Unternehmen ihren Aufwand für die Installation, die Wartung und den Support ihrer Datenbanken.

Hardwareseitig hat Oracle die Oracle Database Appliance X3-2 mit 512 GB Speicher, 18 TB SAS Festplatten und 800 GB Flash Memory ausgestattet, um die Leistung im Online Transaction Processing und Data Warehousing zu beschleunigen. Im Vergleich zum Vorgängermodell ist sie damit bis zu zweimal schneller, verfügt über viermal mehr Storage-Kapazität, dreimal so viel Flash Speicher und über zweieinhalbmal so viel Arbeitsspeicher.

Weitere Informationen unter <http://www.oracle.com/us/products/database/database-appliance/overview/index.html>