

Enterprise Repository - Wer Harvester schreibt wird Assets ernten

Sören Halter
Oracle B.V. & Co. KG
Dreieich

Schlüsselworte

Oracle Fusion Middleware, Oracle Enterprise Repository, OER, SOA, Governance, Harvester, Assets, Artefakte, Verwaltung, Asset Lifecycle Management, Repository Extensibility Framework, Web Services, Axis, Java, JDeveloper, JMS.

Einleitung

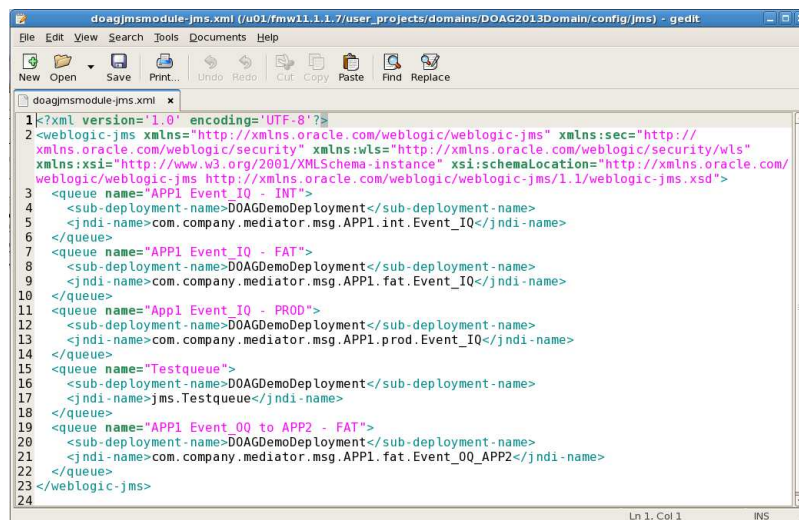
Mit dem Oracle Enterprise Repository (OER) ist die Verwaltung von technischen Artefakten, wie WSDLs, SCAs, XSDs etc. sehr einfach und automatisiert möglich.

Das Enterprise Repository bietet ferner die Möglichkeit, beliebige weitere Assetstypen zu erstellen und die entsprechenden Artefakte zu verwalten. Diese können manuell erfasst oder durch Schreiben eines eigenen Harvesters automatisiert in das Enterprise Repository aufgenommen werden. Im Rahmen des Vortrags werden die hierfür notwendigen Konzepte dargestellt und am Beispiel eines Harvesters für JMS Queues, die in einem WebLogic Server konfiguriert sind, demonstriert.

Die Herausforderung

Einer meiner Kunden setzt schon seit mehreren Jahren den Oracle Service Bus (OSB) als zentralen Mediationslayer zum Datenaustausch zwischen seinen Kernanwendungen ein.

Ein großer Teil der Kommunikation findet dabei asynchron über JMS Queues statt. Die hierzu notwendigen Komponenten, wie Queues, Connection Factories etc. sind im OSB bzw. dem darunterliegenden WebLogic Server (WLS) konfiguriert. Diese Konfiguration wird von WLS in XML Dateien im Domänenverzeichnis abgelegt (Abb. 1).



```
1<?xml version='1.0' encoding='UTF-8' ?>
2<weblogic-jms xmlns="http://xmlns.oracle.com/weblogic/weblogic-jms" xmlns:sec="http://
  xmlns.oracle.com/weblogic/security" xmlns:wls="http://xmlns.oracle.com/weblogic/security/wls"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:schemaLocation="http://xmlns.oracle.com/
  weblogic/weblogic-jms http://xmlns.oracle.com/weblogic/weblogic-jms/1.1/weblogic-jms.xsd">
3  <queue name="APP1 Event IQ - INT">
4    <sub-deployment-name>DOAGDemoDeployment</sub-deployment-name>
5    <jndi-name>com.company.mediator.msg.APP1.int.Event_IQ</jndi-name>
6  </queue>
7  <queue name="APP1 Event IQ - FAT">
8    <sub-deployment-name>DOAGDemoDeployment</sub-deployment-name>
9    <jndi-name>com.company.mediator.msg.APP1.fat.Event_IQ</jndi-name>
10 </queue>
11 <queue name="App1 Event IQ - PROD">
12 <sub-deployment-name>DOAGDemoDeployment</sub-deployment-name>
13 <jndi-name>com.company.mediator.msg.APP1.prod.Event_IQ</jndi-name>
14 </queue>
15 <queue name="Testqueue">
16 <sub-deployment-name>DOAGDemoDeployment</sub-deployment-name>
17 <jndi-name>jms.Testqueue</jndi-name>
18 </queue>
19 <queue name="APP1 Event 00 to APP2 - FAT">
20 <sub-deployment-name>DOAGDemoDeployment</sub-deployment-name>
21 <jndi-name>com.company.mediator.msg.APP1.fat.Event_00_APP2</jndi-name>
22 </queue>
23 </weblogic-jms>
24
```

Abb. 1: Beispiel JMS Konfiguration

Um im Laufe der Zeit den Überblick nicht zu verlieren, werden bisher vom Kunden weitere Informationen zu den Queues in einer Tabelle gepflegt (Abb. 2).

| 1 | Generic Queue Name | Environment | Owner | Mediator Queue Name | Mediator JMS Module | Mediator Stage | Bridge Used |
|----|---|-------------|--------|--|---------------------|----------------|-------------|
| 2 | com.company.mediator.msg.APP1.env.Event_IQ | FAT | Owner1 | com.company.mediator.msg.APP1.fat.Event_IQ | DOAGJMSModule | FAT | N |
| 3 | com.company.mediator.msg.APP1.env.Event_IQ | INT | Owner1 | com.company.mediator.msg.APP1.int.Event_IQ | DOAGJMSModule | INT | N |
| 4 | com.company.mediator.msg.APP1.env.Event_IQ | PROD | Owner1 | com.company.mediator.msg.APP1.prod.Event_IQ | DOAGJMSModule | PROD | N |
| 5 | com.company.mediator.msg.APP1.env.Event_OQ_APP2 | FAT | Owner1 | com.company.mediator.msg.APP1.fat.Event_OQ_APP2 | DOAGJMSModule | FAT | N |
| 6 | com.company.mediator.msg.APP1.env.Event_OQ_APP2 | INT | Owner1 | com.company.mediator.msg.APP1.int.Event_OQ_APP2 | DOAGJMSModule | INT | N |
| 7 | com.company.mediator.msg.APP1.env.Event_OQ_APP2 | PROD | Owner1 | com.company.mediator.msg.APP1.prod.Event_OQ_APP2 | DOAGJMSModule | PROD | N |
| 8 | com.company.mediator.msg.APP2.env.Event_IQ | FAT | Owner2 | com.company.mediator.msg.APP2.fat.Event_IQ | DOAGJMSModule | FAT | N |
| 9 | com.company.mediator.msg.APP2.env.Event_IQ | INT | Owner2 | com.company.mediator.msg.APP2.int.Event_IQ | DOAGJMSModule | INT | N |
| 10 | com.company.mediator.msg.APP2.env.Event_IQ | PROD | Owner2 | com.company.mediator.msg.APP2.prod.Event_IQ | DOAGJMSModule | PROD | N |

Abb. 2: Tabelle mit Zusatzinformationen

Hierbei hat man sich für die Namen der Queues auf folgendes Format festgelegt:

`com.company.mediator.msg.<Applikation der die Queue gehört>.<Umgebung>.<Zweck>_[IQ|OQ_<Applikationsziel des Outputs>`

Im Zeitverlauf nahm die Anzahl der Applikationen, die über den Mediationslayer miteinander kommunizieren, stark zu; und mit ihnen auch die Anzahl Queues. Aktuell muss das Projekt weit über 1000 Queues pflegen. In Zukunft sollen noch mehr Informationen und Querbeziehungen verwaltet werden.

Es war klar, dass der bisherige Ansatz zur Verwaltung der Queues nicht mehr lange funktioniert.

Im Rahmen von zwei Workshops mit dem Kunden wurde Oracle Enterprise Repository (OER) als Baustein für die Governance identifiziert. Insbesondere auch weil das Oracle Enterprise Repository schon einen Harvester für den Oracle Service Bus mitbringt und sich das Metadatenmodell sehr flexibel und einfach an eigene Anforderungen anpassen lässt.

Allerdings war auch klar, dass der Harvester, der mit OER geliefert wird, weder die WLS Konfigurationsdateien noch die zusätzlichen in der Tabelle gespeicherten Informationen verarbeiten kann.

Die schiere Anzahl an zu erfassenden Informationen schloss eine manuelle Erfassung aus und legte den Grundstein für die im weiteren Verlauf vorgestellte Lösung: Einen projektspezifischen Harvester.

Überblick Oracle Enterprise Repository

Das Oracle Enterprise Repository (OER) ist eine Design-time Komponente für die Pflege von Informationen über Artefakte oder Assets. Insbesondere die Pflege von SOA Assets wird durch einen hohen Grad der Automatisierung unterstützt. Hierzu zählt die Möglichkeit, Artefakte aus SOA Projekten (WSDLs, XSDs, SCAs, Servicebuskonfigurationen etc.) automatisiert über den Harvester in das Repository zu laden.

Hierfür liefert OER schon eine ganze Reihe von vorgefertigten Assettypen, Kategorienarten, Beziehungstypen etc. mit, erlaubt es aber auch - durch eine konfigurierbares Metamodell - selber neue Typen für die Erfassung eigener Artefakte anzulegen.

Der normale Anwender arbeitet über eine Weboberfläche mit OER (Abb. 3).

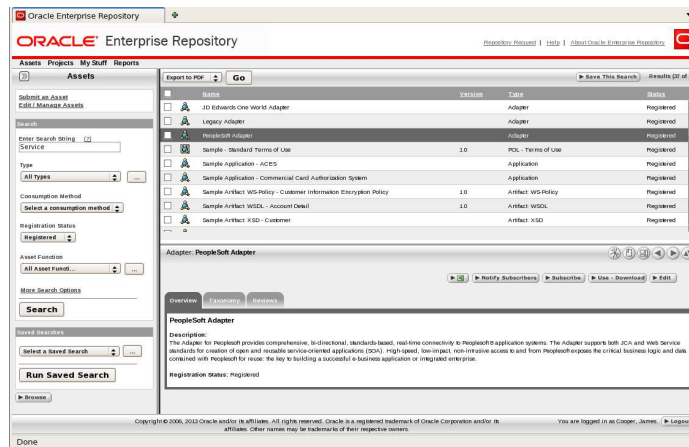


Abb. 3: Die Weboberfläche von OER

Über diese kann er Assets suchen oder browsen, verbundene Dokumente herunterladen, Assets bewerten, Reports erzeugen und vieles mehr.

| Role | View | Use | Download | Review | Notify | Edit | Approve Tabs | Register | Edit Access Settings | Create/Submit | Launch Asset Editor | Edit Artifact Status | Edit Types |
|------------------------|------|-----|----------|--------|--------|------|--------------|----------|----------------------|---------------|---------------------|----------------------|------------|
| advancedSubmitter | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| registrar | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| registrarAdministrator | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| user | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |

Abb. 4: OER Rollen und ihre Privilegien

Mit ausreichenden Privilegien (Abb. 4), hat der Anwender Zugriff auf den Asset Editor (Siehe Abb. 5) und/oder den Type Manager (Abb. 6).

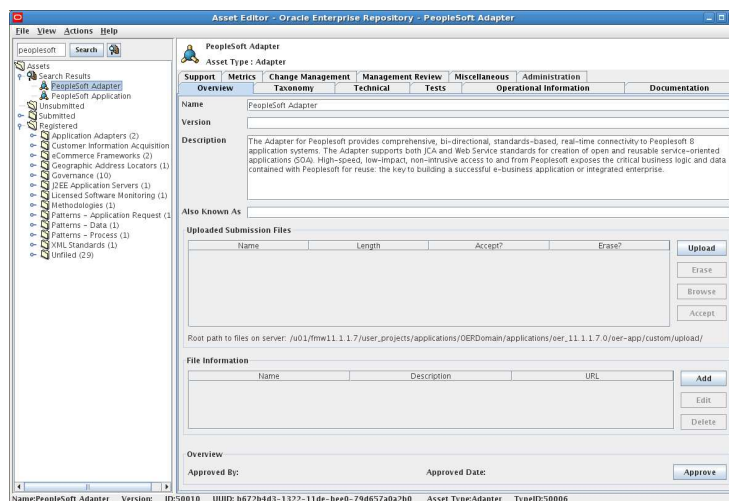


Abb. 5: Asset Editor

Der Asset Editor erlaubt die Erfassung neuer Assets, die Bearbeitung bestehender Assets und die Transition eines Assets von einem Zustand seines Lebenszyklus in den nächsten.

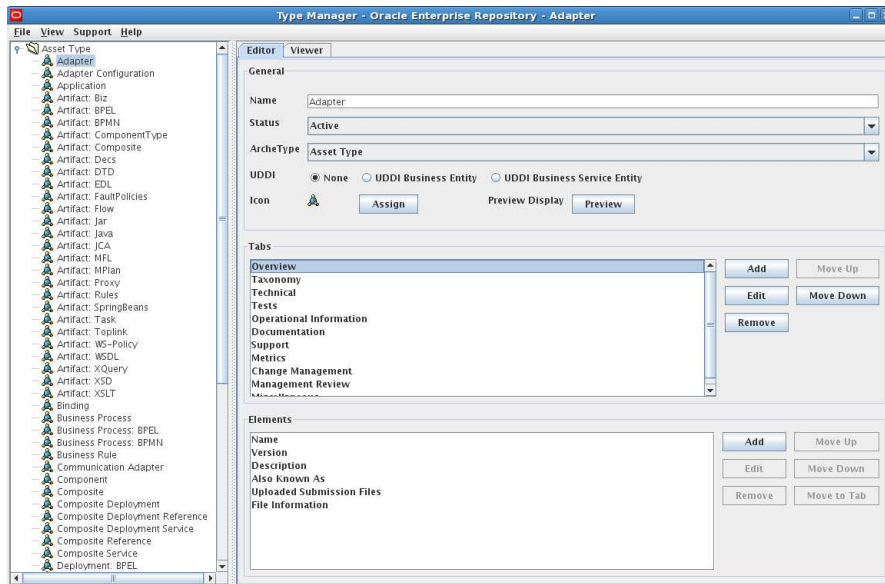


Abb. 6: Type Manager

Der Type Manager ermöglicht es, neue Assettypen zu erstellen bzw. bestehende Assettypen zu bearbeiten; und das, sowohl für die Sicht auf das Asset vom Asset Editor aus als auch gesondert für die Sicht im Web Browser (Viewer). Änderungen sind hierbei direkt nach dem Speichern verfügbar, d.h. ein neuer Assettyp kann unmittelbar nach dem Speichern zur Erfassung von Assets dieser Art verwendet werden.

Neben den Assettypen können auch die Beziehungen zwischen Assets konfiguriert werden (Abb. 7), Kategorienarten inkl. verschachtelter Kategorien erzeugt werden, Wertelisten gepflegt werden und vieles mehr.

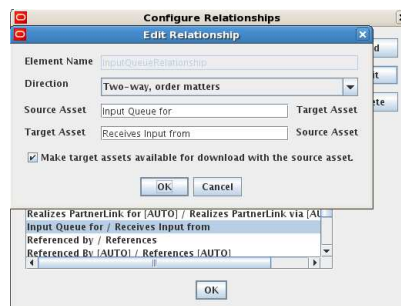


Abb. 7: Dialog zur Konfiguration der Assetbeziehungen

REX – Der Weg zum eigenen Harvester

OER stellt mit dem Repository Extensibility Framework (REX) eine sehr mächtige, auf Web Services basierende, Integrationsschicht zur Verfügung (Abb. 8).

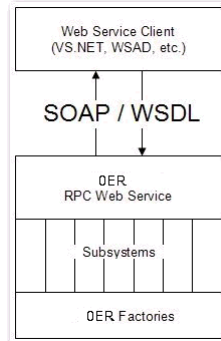


Abb. 8: REX Architektur

Diese Schnittstelle muss einmalig durch einen OER Administrator freigegeben werden. Anschließend kann über die URL <http://<oerip:oerport>/oer/services/FlashlineRegistry?WSDL> auf die REX WSDL zugegriffen werden.

REX ermöglicht den Zugriff auf eine Vielzahl von OER Subsystemen. Diese Subsysteme gliedern sich grob nach den in OER verfügbaren Entitäten. Die bereitgestellten Subsysteme umfassen:

- acceptableValue
- asset
- assetType
- authToken
- categorization
- categorizationType
- department
- extraction
- import/export
- project
- relationship
- role
- user
- vendor

Für die Subsysteme stellt REX Dienste nach der CRUD-Q Konvention zur Verfügung:

- **C**reate
- **R**ead
- **U**ppdate
- **D**eleate
- **Q**uery

Wobei für den Dienstenamen die folgende Namenskonvention verwendet wird:

<subsystem><Operation>

So wird die Operation, welche ein neues Asset im Repository erzeugt, mit `assetCreate(...)` aufgerufen. Analog gibt es `assetRead(...)`, `assetUpdate(...)`, `assetDelete(...)` und `assetQuery(...)` Operationen.

Einige Subsysteme können hierbei auch noch weitere Operationen jenseits von CRUD-Q anbieten, andere eventuell auch weniger. So kann ein User z.B. nicht gelöscht sondern nur inaktiviert werden. Eine detaillierte Liste zusammen mit vielen Beispielen zur Nutzung der REX Schnittstelle, findet sich im Oracle® Fusion Middleware Integration Guide for Oracle Enterprise Repository unter http://docs.oracle.com/cd/E28280_01/doc.1111/e15754/partpage3.htm.

Die Lösung

Für die weiteren Ausführungen wird angenommen, dass ein Oracle Enterprise Repository unter <http://vbox:7101/oer> zur Verfügung steht. Dieses wurde so konfiguriert, dass REX erlaubt ist.

Die REX WSDL kann dann unter <http://vbox:7101/oer/services/FlashlineRegistry?WSDL> heruntergeladen werden.

Der Harvester für die oben beschriebene Herausforderung wurde in Java implementiert. Um von Java aus auf die REX API zugreifen zu können, wurden mit AXIS WSDL2java (siehe auch <http://tinyurl.com/o9mw2bw>) Java Stubs generiert:

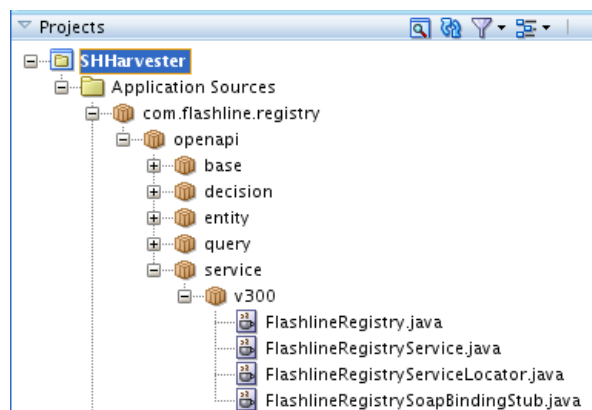


Abb. 9: Die Java Stubs für den REX Zugriff im JDeveloper

Hiermit steht nun alles bereit, um ein klassisches Javaprogramm, bestehend aus

- **Eingabe** der Daten aus den XML Dateien und einer csv Datei mit den Zusatzinformationen
- **Verarbeitung** und Kombination dieser Daten
- **Ausgabe** der aufbereiteten Daten in das Oracle Enterprise Repository

zu schreiben.

Die Hauptklasse `JMSHarvester` setzt diesen Ablauf in der Methode `harvest` um (die von der `main`-Methode aus aufgerufen wird).

Hier der schematische Ablauf:

- Verarbeite die Übergabeparameter (u.a. URL zum Repository, login, Verzeichnis mit den Konfigurationsdateien)
- Stelle die Verbindung zum Oracle Enterprise Repository her
- Lies die JMS Konfigurationsdatei(en) ein
- Lies die Datei mit den zusätzlichen Informationen ein

- Finde für jede Queue aus der JMS Konfiguration die zugehörigen Informationen anhand des JNDI Namens. Verarbeite nur die Queues mit zusätzlichen Informationen und logge sowohl Queues ohne Informationen als auch Informationen ohne Queues
- Für jede Queue lade die angereicherten Daten in das Repository.

Weitere Details zur Implementierung werden im Vortrag präsentiert.

Der so implementierte Harvester benötigt in OER noch

- zwei Assettypen,
 - DOAG: Queue
 - DOAG: Applikation
- drei Beziehungen und
 - InputQueueRelationship (Input Queue for/Receives Input from)
 - OutputQueueRelationship (Output Queue for/ Output Messages to)
 - OwnsRelationship (Owned by/owns)
- zwei Kategorienarten
 - Environment
 - Application

Diese wurden mit den weiter oben dargestellten OER Werkzeugen erzeugt und konfiguriert. Bei Bedarf können sie über die OER Export/Import Schnittstelle von einer Umgebung in die eine andere transportiert werden.

Das Ergebnis

Führt man den JMSharvester mit den in **Die Herausforderung** dargestellten Dateien aus, erzeugt dieser den nachfolgenden Output, an dem man die einzelnen Verarbeitungsschritte ablesen kann:

```
<2013-10-15 17:20><Info><JMSharvester> Loglevel set to Info
```

```
<2013-10-15 17:20><Debug><OERConnection> Retrieved Asset Type 'DOAG: Queue' with ID 50101
```

```
<2013-10-15 17:20><Debug><OERConnection> Retrieved Asset Type 'DOAG: Application' with ID 50100
```

```
<2013-10-15 17:20><Info><JMSharvester> Processing all files in directory /home/oracle/jdeveloper/SHHarvester/Demodaten/jms
```

```
<2013-10-15 17:20><Info><JMSharvester> Parsing WebLogic JMS configuration file /home/oracle/jdeveloper/SHHarvester/Demodaten/jms/doagjmsmodule-jms.xml
```

```
<2013-10-15 17:20><Info><JMSharvester> Processing Infos from /home/oracle/jdeveloper/SHHarvester/Demodaten/MessagingAssets.csv
```

```
<2013-10-15 17:20><Info><JMSharvester> No additional information for JMS Queue 'Testqueue' found
```

```
<2013-10-15 17:20><Info><JMSharvester> JMS Queue 'Testqueue' will not be harvested!
```

```
<2013-10-15 17:20><Info><JMSharvester> Info File Report (Lines for which NO JMS Queue was found):
```

<2013-10-15 17:20><Info><JMSharvester> Line 9:
com.company.mediator.msg.APP2.env.Event_IQ;INT;Owner2;com.company.mediator.
msg.APP2.int.Event_IQ;DOAGJMSModule;INT;N

<2013-10-15 17:20><Info><JMSharvester> Start harvesting JMS Queue(s):

<2013-10-15 17:20><Info><OERConnection> Application categorization 'APP1'
with ID '50000' created in OER

<2013-10-15 17:20><Info><OERConnection> Environment categorization 'INT'
with ID '50001' created in OER

<2013-10-15 17:20><Info><OERConnection> Asset of type 'DOAG: Queue' created
for JMS Queue 'APP1 Event_IQ - INT' in OER

<2013-10-15 17:20><Info><OERConnection> Value 'Owner1' with ID '50518'
created for Acceptable Value List 'Organizational Ownership' in OER

<2013-10-15 17:20><Info><OERConnection> Asset of Type 'DOAG: Application'
created for Application 'APP1' in OER

<2013-10-15 17:20><Info><OERConnection> Relationship 'Owns / Owned by'
created between 'APP1' and 'APP1 Event_IQ - INT'

<2013-10-15 17:20><Info><OERConnection> Relationship 'Receives Input from /
Input Queue for' created between 'APP1' and 'APP1 Event_IQ - INT'

<2013-10-15 17:20><Info><OERConnection> Environment categorization 'FAT'
with ID '50002' created in OER

<2013-10-15 17:20><Info><OERConnection> Asset of type 'DOAG: Queue' created
for JMS Queue 'APP1 Event_IQ - FAT' in OER

<2013-10-15 17:20><Info><OERConnection> Asset of Type 'DOAG: Application'
with Name 'APP1 (1.0)' already exists.

<2013-10-15 17:20><Info><OERConnection> Relationship 'Owns / Owned by'
created between 'APP1' and 'APP1 Event_IQ - FAT'

<2013-10-15 17:20><Info><OERConnection> Relationship 'Receives Input from /
Input Queue for' created between 'APP1' and 'APP1 Event_IQ - FAT'

<2013-10-15 17:20><Info><OERConnection> Environment categorization 'PROD'
with ID '50003' created in OER

<2013-10-15 17:20><Info><OERConnection> Asset of type 'DOAG: Queue' created
for JMS Queue 'App1 Event_IQ - PROD' in OER

...

<2013-10-15 17:20><Info><JMSharvester> JMSharvester successfully finished

<2013-10-15 17:20><Info><JMSharvester> Processing Time: 17144 ms

Im Oracle Enterprise Repository finden sich nun die geharvesteten Assets (Abb. 10).

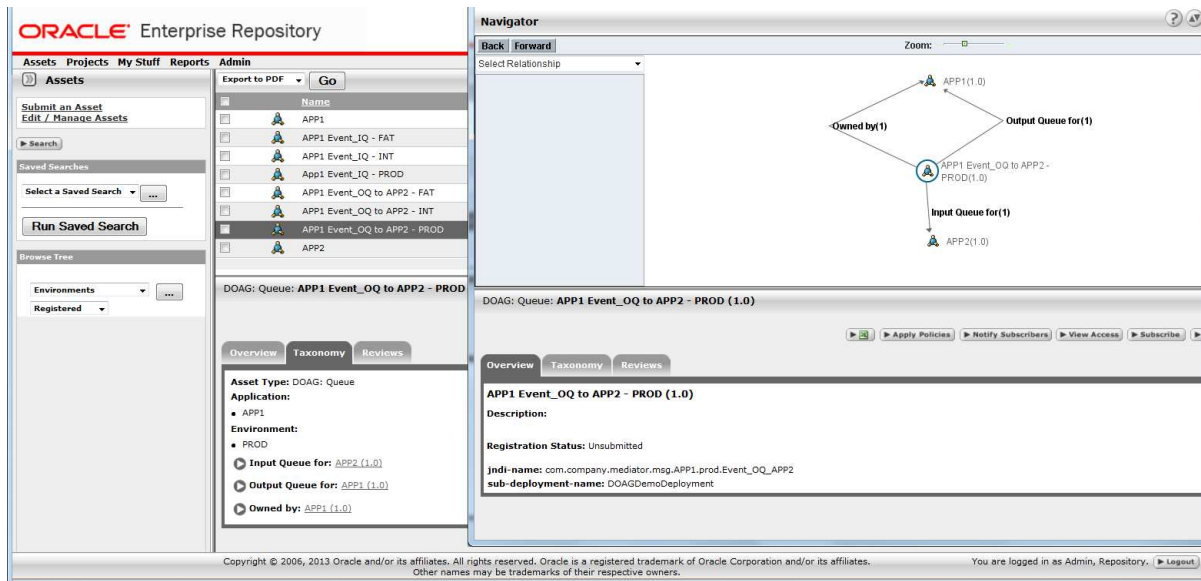


Abb. 10: Die geharvesteten Assets in OER

Weitere Details zu diesem Artikel können dem zugehörigen Foliensatz entnommen oder gerne auch direkt beim Autor angefragt werden.

Kontaktadresse:

Sören Halter
Oracle B.V. & Co. KG
Robert-Bosch-Str. 5

D- 65503 Dreieich

Telefon: +49 (0) 261 6679441
Mobil: +49 (0) 174 3443752
E-Mail: soeren.halter@oracle.com
Internet: www.oracle.de