

Das Metadatenrepository sinnvoll einsetzen

Marcel Amende, ORACLE Deutschland B.V. & Co. KG, Düsseldorf

Schlüsselworte

MDS, SOA, Dienste, Referenzen, Artefakte, WSDL

Einleitung & Problemstellung

Entwicklern stellt sich bei der Benutzung der SOA Suite und des JDevelopers als *dem* zentralen Entwicklungswerkzeug für alle Aspekte einer SOA wiederholt dieselbe Frage: Sollen WSDLs und abhängige Artefakte beim Erzeugen neuer Dienste, Referenzen und Komponenten lokal in das Projekt *kopiert* werden oder auf dem Server *referenziert* werden?

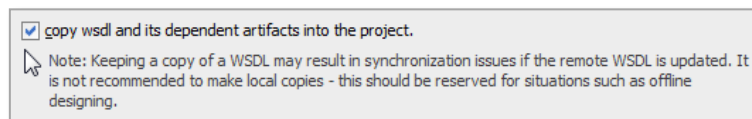


Abb. 1: Auswahlmöglichkeit im JDeveloper, Artefakte kopieren oder referenzieren?

In der Praxis bemerkt man schnell, dass man diese Frage nicht richtig beantworten kann. Die Auswahlmöglichkeiten entsprechen eher der Wahl zwischen „Tod“ oder „Teufel“:

Kopiert man die Artefakte lokal in das Projekt, kann man zwar auch ohne Zugriff auf die (entfernten) Ressourcen und Endpunkte kompilieren, bezahlt dies aber mit einer wachsenden Unübersichtlichkeit durch die flache Projektstruktur und Synchronisationsprobleme bei Änderungen an den Diensten und zugehörigen Artefakten.

Referenziert man die Artefakte hingegen, ist das Projekt ohne Zugriff auf die (entfernten) Ressourcen und Endpunkte nicht mehr kompilierbar. Ohne (Online-) Zugriff auf entfernte, referenzierte Dienste ist man nur eingeschränkt arbeitsfähig. Zudem muss man nun eventuelle Abhängigkeiten in der Ausbringungsreihenfolge einzelner Projekte betrachten: Der Dienst muss vor dem referenzierenden Konsumenten ausgebracht werden. Referenzieren zwei Projekte wechselseitig Dienste ist man sogar in einer Henne/Ei-Problematik gefangen.

SOA „Separation of Concerns“

Wenn keine der angebotenen Optionen passt, wie macht man es denn dann besser? In einer SOA bietet es sich an, das Grundprinzip der Aufgabenteilung anzuwenden, in der Literatur oft unter dem Stichwort „*Separation of Concerns*“ zu finden: Dies bedeutet insbesondere, dass Artefakte unabhängig von den konsumierenden Diensten entworfen und implementiert werden, die diese in abstrakter Form (d.h. ohne Implementierungsdetails) referenzieren. So erreicht man eine Unabhängigkeit vom Dienstanbieter und den physikalischen Endpunkten.

SOA Meta Data Store (MDS)

Für die Entwicklungs-, als auch die Laufzeit wird eine zentrale, zuverlässige Ablage für Artefakte und Metadaten benötigt. In beiden Fällen bietet sich der *Oracle Meta Data Store (MDS)* an, allerdings in zwei Varianten: Für die Entwicklung kann die dateisystembasierte Variante genutzt werden, die über beliebige Quellkontrollsysteme versioniert, verteilt und gesichert wird, zur Laufzeit die datenbankbasierte und somit hochverfügbare und clusterfähige Variante.

Eine Vielzahl von verschiedenen Artefakten eignet sich für die zentrale Ablage im MDS: Neben WSDLs und XML Schemata sind dies z.B. Transformationen, Rule Dictionaries, Domain Value

Maps, XRef-Definitionen und JCA-Konfigurationen – kurzum: Alles, was in irgendweiner Form Projektübergreifend wiederverwendet werden kann und soll.

Einsatz des MDS in SOA Projekten – Vorbereitung

Ein SOA-Projekt im JDeveloper kann in wenigen einfachen Schritten für die MDS-Nutzung vorbereitet werden:

1. Für die zentral zu verwaltenen Artefakte wird ein eigenes Projekt (z.B. „SharedArtifacts“) im JDeveloper angelegt.
2. Im datenbankbasierten MDS werden alle vom Anwender selbst verwalteten Artefakte in einem Ordner „apps“ abgelegt. Alle anderen Bereiche im MDS sind für die Laufzeitkonfigurationen der verschiedenen Oracle Produkte reserviert und nicht von außen zugänglich und nutzbar. Diese Struktur muss im dateibasierten MDS händisch nachgebildet werden, indem man im neuen Projekt einen neuen Ordner Namens „apps“ für eigene Artefakte anlegt.
3. Der Applikationsdeskriptor „adf-config.xml“ verweist auf das im Rahmen einer JDeveloper Applikation zur Entwicklungszeit verwendete Metadatenrepository. Dies ist auf das neu erstellte Projekt (hier: „SharedArtifacts“) anzupassen:

```
<metadata-store class-name="oracle.mds.persistence.stores.file.FileMetadataStore">
  <property name="metadata-path" value="C:\...\<SOA App. Folder>" />
  <property name="partition-name" value="SharedArtifacts" />
</metadata-store>
```

4. Gemeinsam zu verwendende Artefakte werden nun in den Ordner „apps“ kopiert. Die Organisation kann über Unterordner erfolgen, die einer Taxonomie folgen und die Artefakte versionieren. Insbesondere WSDLs sind dabei in ihrer abstrakten Form, d.h. ohne Implementierungsdetails, abzulegen. Diese kann bei Endpunkten in der Oracle SOA Suite über die URL-Endung „.wsdl“ abgerufen werden, während die Endung „?wsdl“ zu den konkreten WSDLs führt.
5. Damit die Artefakte zur Laufzeit zur Verfügung stehen, müssen diese in das datenbankbasierte MDS ausgebracht werden. Dafür steht im JDeveloper der *Metadata Archive (MAR) Deployment Descriptor* zur Verfügung. Alternativ kann der Inhalt des „apps“-Ordners auch in eine „.jar“-Datei gepackt werden, die wiederum in eine “.zip“ Datei verpackt alleine oder mit anderen SOA Projekten als *SOA Budle* ausgebracht werden kann.

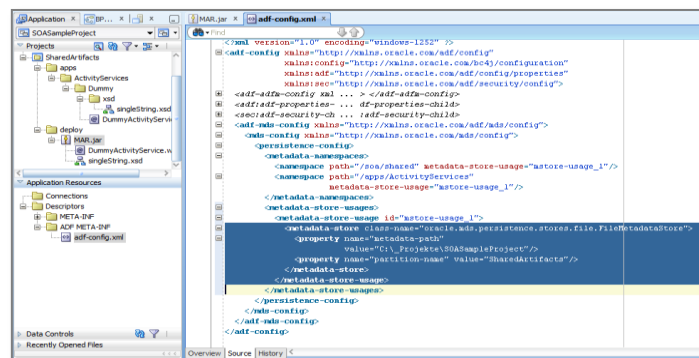


Abb. 2: Konfiguration für dateibasiertes MDS im JDeveloper



Einsatz des MDS in SOA Projekten – Nutzung

Am Beispiel einer einzubindenden Referenz auf einen Dienst erfolgt die Nutzung des MDS folgendermaßen:

1. Für die Verwendung der Komponentenpalette können Verbindungen sowohl für das dateibasierte MDS, als auch für das datenbankbasierte MDS angelegt werden.
2. Ressourcen im MDS adressiert man nun über den Protokollprefix „oramds:“. Dieser zeigt lokal im JDeveloper auf das dateibasierte MDS, also den „apps“-Ordner im „SharedArtifacts“ Projekt. Zur Laufzeit zeigt er auf das datenbankbasierte MDS. Ein Verweis auf eine WSDL sieht dann folgendermaßen aus:

```
oramds:/apps/../../../../Service.wsdl
```

Die Warnung des JDevelopers bezüglich des Fehlens der „konkreten Bindings“ kann bestätigt werden. Dieses wird bei der Ausbringung über Konfigurationspläne hinzugefügt.

3. Läuft der einzubindende Dienst ebenfalls in der SOA Suite, können *Binding* „Port“ und „Location“ im Fusion Middleware Control nachgeschlagen und in der SCA-Konfiguration („composite.xml“) eingefügt werden (  |):

```
<binding.ws port="..." location="..." />
```

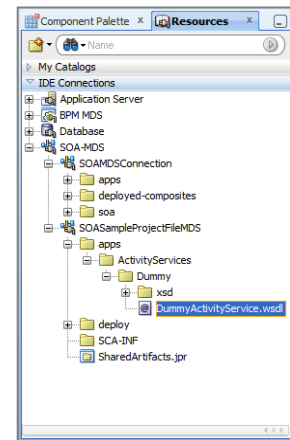


Abb. 3: MDS Connections

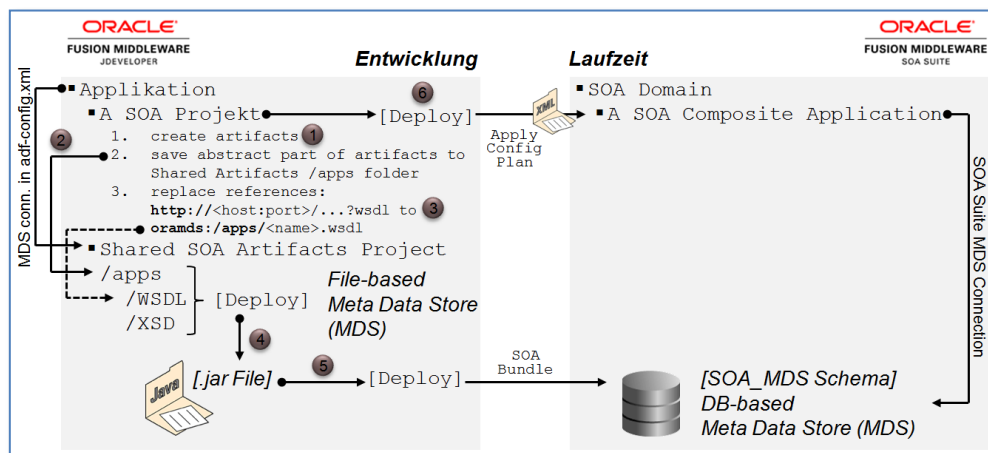


Abb. 4: Schematischer Ablauf, Nutzung des MDS zur Entwicklungs- und Laufzeit

Fazit

In wenigen Schritten können SOA Projekte für die Verwendung des MDS für alle Arten von gemeinsam zu nutzenden Artefakten konfiguriert werden. Ein Aufwand, der sich immer lohnt, wird man doch mit einer wesentlich übersichtlicheren Struktur (z.B. flexible Unterordner), besseren Ordnung (durch separates Projekt) und universell leichter Handhabbarkeit (Online/Offline-Problematik, Entkopplung von Dienstanbieter und Konsument) belohnt. Konsequenterweise wird daher dieses Vorgehen als Standardarbeitsweise im Umgang mit Artefakten in der nächsten Version der Oracle SOA Suite etabliert.

Kontaktadresse:

Marcel Amende
Senior Leitender Systemberater
ORACLE Deutschland B.V. & Co. KG
Hamborner Str. 51
D-40472 Düsseldorf
Telefon: +49 (0) 211-74839 539
E-Mail: Marcel.Amende@oracle.com
Internet: www.oracle.de