Releasemanagement mit ODI und Apache Subversion

Dr. Holger Dresing
Oracle Deutschland B.V. & Co. KG

Schlüsselworte
Oracle Data Integrator – ODI, Apache, Subversion, Releasemanagement, Lifecycle Management, ETL, Data Integration, Version Control System, VCS

Einleitung
Vorherige Releases von ODI haben eine eigene Oracle spezifische Lifecycle Management Lösung enthalten. Seit ODI 12.2.1 ist Apache Subversion in ODI integriert. Damit ist es möglich, Tags eines Projektes zu erstellen, Branches zu verwalten, um parallel in Teams entwickeln zu können und Deployment Archive anzulegen, um ODI Objekte aus der Entwicklungs- in die Produktionsumgebung zu überführen. Wichtig ist, auch die Grundlagen von Subversion zu verstehen: Ein „Trunk“ stellt die Hauptlinie in der Entwicklung dar, Branches sind Verzweigungen der Hauptentwicklungslinie, um beispielsweise Versionen oder Fehlerbehebungen zu erstellen und „Tags“ beschreiben den aktuellen Zustand.
Die hier dargestellten Möglichkeiten skizzieren die im Data Integration Blog von Oracle ausführlich dargestellten Zusammenhänge [1]. ODI Studio wird ebenso als bekannt vorausgesetzt.

Architektur
ODI Studio ist weiterhin das Benutzerinterface, mit dem die Versionsoperationen durchgeführt werden. Beispielsweise wird ein Objekt weiterhin in eine XML Datei exportiert, die anschließend in das Version Control System (VCS) übernommen wird. Ebenso wird beim Import zunächst die XML Datei ausgecheckt und anschließend das Objekt in das ODI Repository importiert.

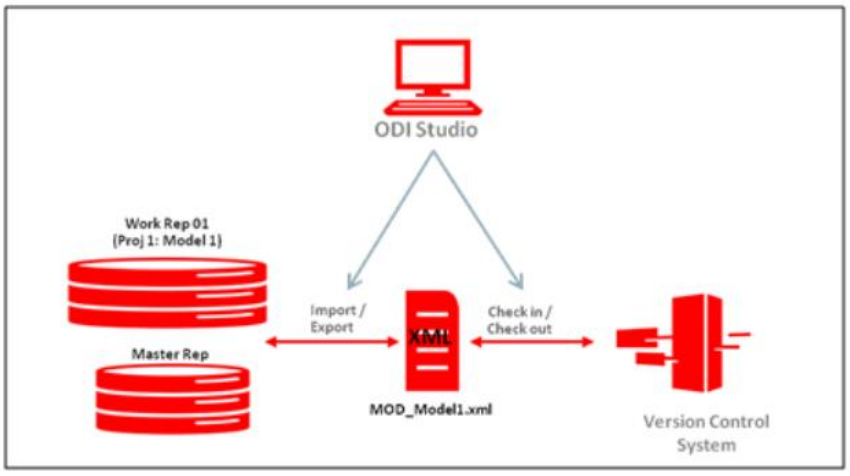

Abb. 1: Vorgehen bei Subversion

Apache Subversion muß zunächst unabhängig von ODI installiert werden. Danach muß Apache Subversion konfiguriert werden im Menü „Team/Subversion“. Dort wird die URL und ein Subversion

[1] Siehe Abschnitt „Weiterführende Hinweise“

User angegeben. Zusätzlich wird hier definiert, ob es schon Trunks und Branches gibt und wie Versionen generiert werden sollen: bei jedem Speichern oder manuelles Erstellen.
Im ODI Studio gibt es zwei Menüs zum Arbeiten mit VCS:

- Im Menü „Tree/Subversion“ sind alle objektübergreifenden Kommandos zusammengefaßt.
- Im Menü „Datei/Version/Subversion“ werden alle objektspezifischen Kommandos verfügbar.

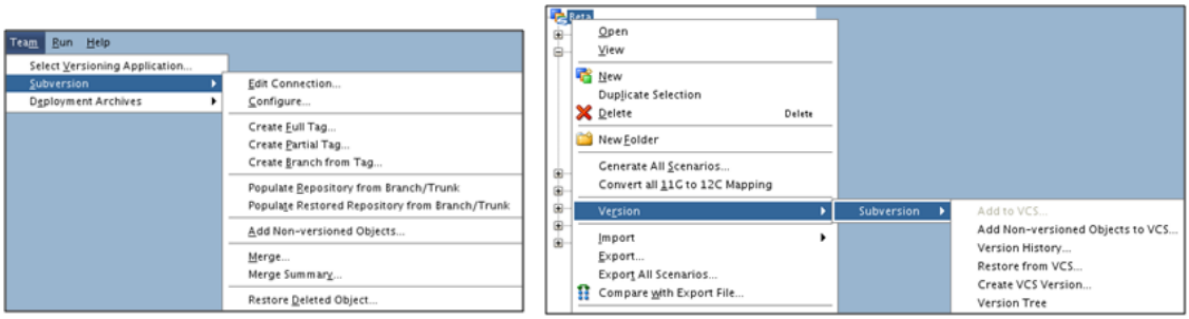

Abb. 2: Apache Subversion Kommandos im ODI Studio

Wenn ODI Objekte durch Apache Subversion verwaltet werden sollen, können sie als Einzelobjekte oder als Gruppe von Objekten hinzugefügt werden. Wenn ein Objekt hinzugefügt wird („Add to VCS“), werden auch alle hierarchisch abhängigen Objekte hinzugefügt. Der Entwickler muss dafür keine zusätzlichen Schritte durchführen, er kann jedoch nachträglich die abhängigen Objekte auswählen. Mit „Add Non-versioned Objects to VCS“ werden zunächst alle nicht versionierten Objekte als Liste angezeigt und der Anwender kann entscheiden, welche Objekte hinzugenommen werden sollen. Im Projektbaum von ODI Studio werden die versionierten Objekte graphisch markiert (siehe Folder „Version Control“ in Abb. 3).

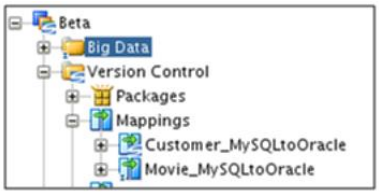

Abb. 3: Das versionierte Objekt „Version Control“ im ODI Projektbaum

Neue Objekt Versionen werden im Context Menü „Create VCS Version“ erstellt. Wenn eine neue Version erstellt wird, werden auch die nachfolgenden Kommandos auf das Container Objekt angewandt:

- Es werden auch Versionen für alle abhängigen Objekte erstellt, die seit der letzten Version verändert. Das betrifft jedoch nicht die nicht-versionierten Objekte.
- Alle Löschungen, Umbenennungen und Veränderungen werden auch an Subversion weitergegeben.

Die Objekt Historie kann auch im ODI Studio betrachtet werden: im Tabellenformat oder im Baumformat. Auf alle Formate können Such- und Filteroperationen angewandt werden.

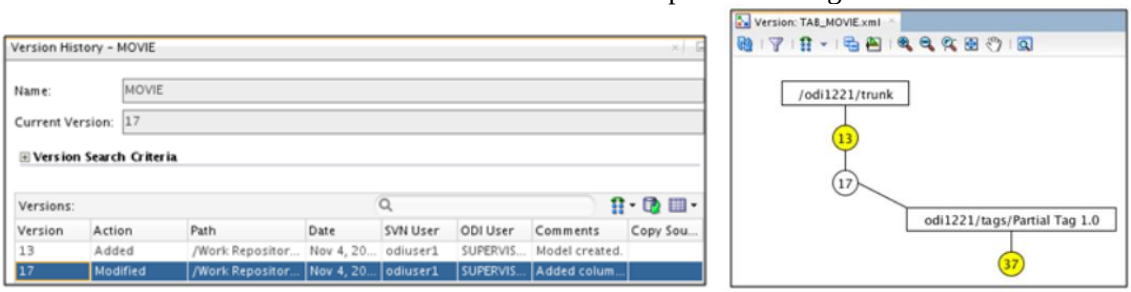

Abb. 4: Historie von Objekten nachvollziehen

Darauf basierend können Versionen verglichen werden. Auch hierfür gibt es ein besonderes Fenster im ODI Studio (siehe Abb. 5).

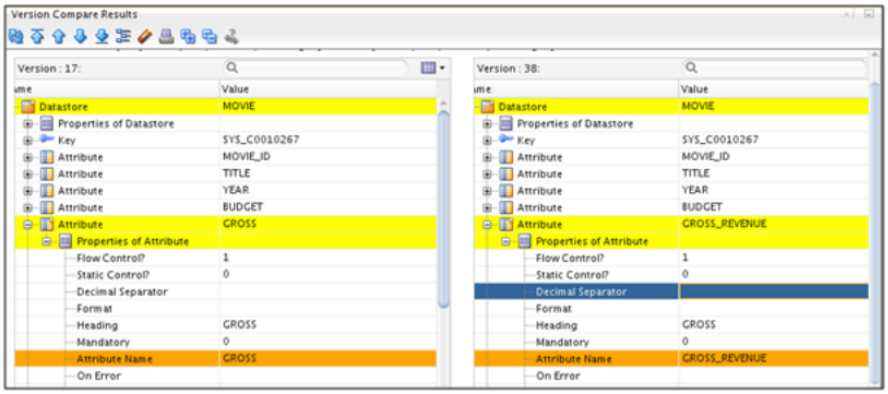

Abb. 5: Vergleich von Versionen

In ODI gelöschte Objekte, die in Subversion noch vorhanden sind, können wiederhergestellt werden. Dafür gibt es das Kommando „Team/Subversion/Restore Deleted Object“. Die zweite Möglichkeit ist das Wiederherstellen einer älteren Version im Context Menü. Dort ist es auch möglich, Änderungen zu überschreiben bzw. zusammen zu fassen.

Gleichzeitiges Entwickeln mit mehreren Subversion Branches

Wie in Abb. 6 dargestellt, kann ODI mit einem oder mehreren Master/Work Repositories arbeiten. In diesem Beispiel wird der Subversion Trunk von User Set 1 als Hauptentwicklungslinie gesehen und enthält alle ODI Objekte. Branch 01 ist für das User Set 2 konfiguriert, das mit einem Subset von ODI Objekten arbeitet. Das User Set 3 arbeitet auf einem Release Branch und behebt Fehler in der Release Code Line. Alle 3 User Sets arbeiten gleichzeitig (parallel) und können ihre Subversion Branches wieder zusammenführen.

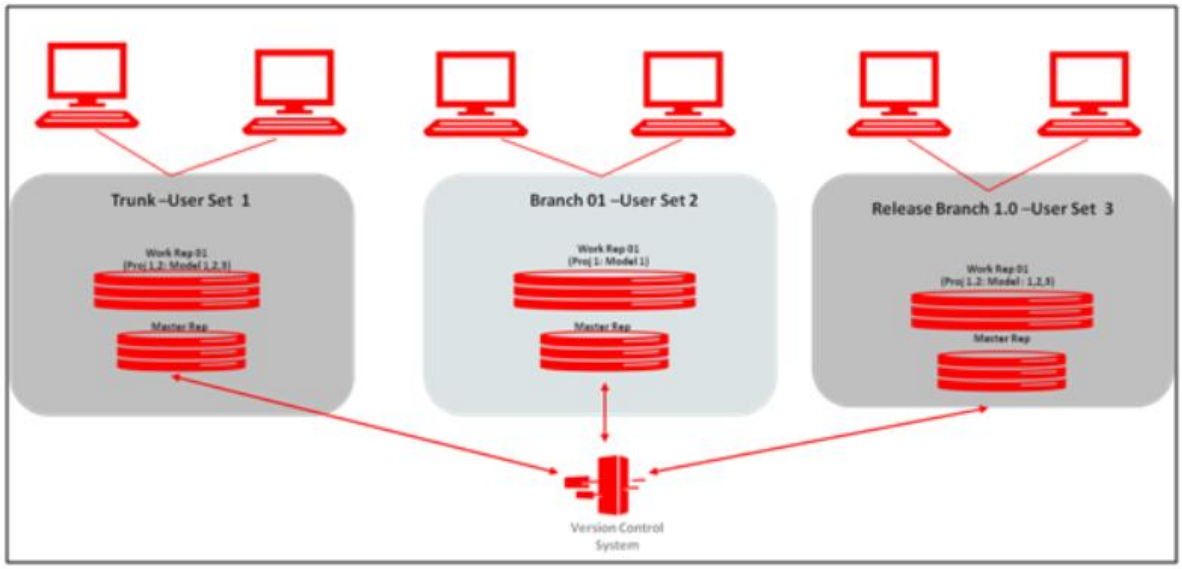

Abb. 6: Beispiele für verschiedene ODI Repositories mit Subversion

Ein „Tag“ in Subversion ist eine Bezeichnung für eine aktuelle Version eines Sets von Objekten, um damit wie bei einem Schnappschuss seinen aktuellen Zustand festzuhalten. Es friert den aktuellen Zustand der Objekte ein und man kann auf einen alten Zustand in der Zukunft zurückgehen. Ein Tag kann direkt in ODI Studio erstellt werden. Er umfasst alle oder eine Auswahl von Objekten im Repository.
ODI führt immer die folgenden Aktionen aus, um die Konsistenz eines Tags zu gewährleisten:

- Fügt Objekte zu Subversion hinzu, falls sie nicht versioniert sind.
- Erstellt Versionen von Objekten falls die letzten Änderungen nicht versioniert sind.
- Automatisch werden alle Abhängigkeiten der ausgewählten Objekte hinzugefügt, so dass alle relevanten Objekte beim Zurücksetzen vorhanden sind.

Es gibt zwei Arten von Tags, die von ODI erzeugt werden:

- Full Tag – umfasst alle Objekte in einem Repository, alle Objekte werden mit Subversion synchronisiert.
- Partial Tag – umfasst eine Auswahl von Objekten inklusive aller Abhängigkeiten.

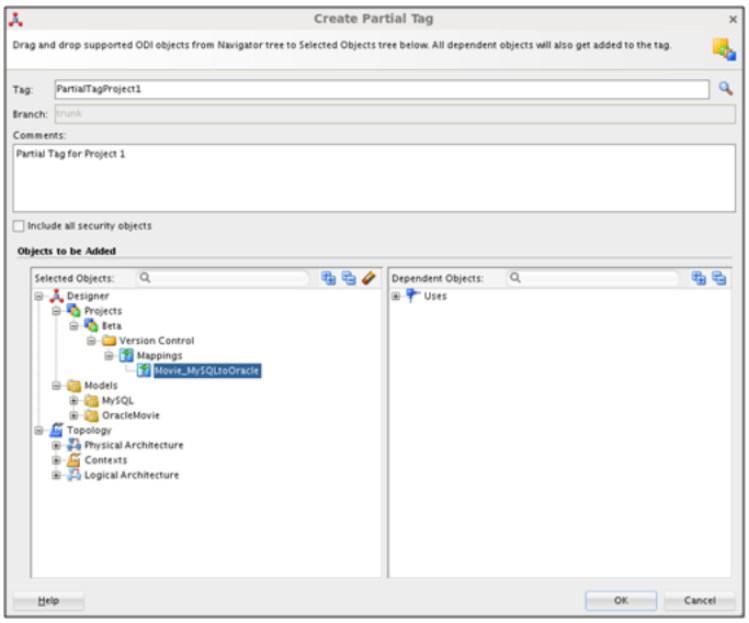

Abb. 7: Partialer Tag

Aus einem „Tag“ kann ein „Branch“ erstellt werden. Die Objekt-Versionen eines ausgewählten Tags sind die Branches, die nicht unbedingt die letzte Version eines Objektes darstellen müssen.
Für die Tags und Branches gibt es im ODI Studio verschiedene Kommandos. Mit „Populating Repository“ wird ein neues ODI Repository angelegt und mit einem Branch verknüpft. Mit „Populating Restored Repository“ können korrupte Repositories mit einem älteren Backup wieder hergestellt werden.
Das Zusammenfassung von Entwicklungs-Branches ist eine wesentlicher Bestandteil paralleler Entwicklung. ODI Studio ermöglich, Branches, Trunks und Tags zusammenzufassen.

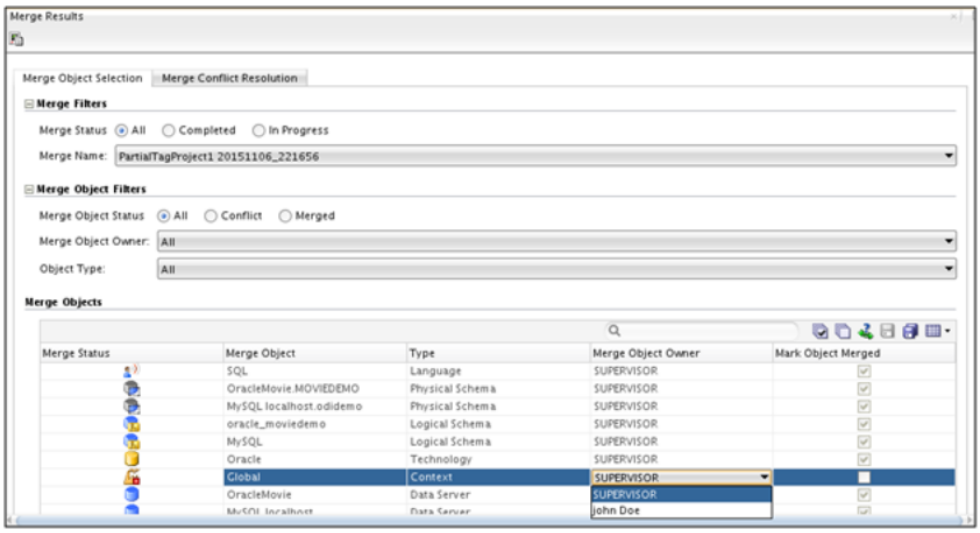

Abb. 8: Merge Ergebnisse

Der „Merge“-Prozesse fasst automatisch alle Objektveränderungen zusammen, die keine Konflikte beinhalten. Dafür gibt es eine Oberfläche in ODI Studio, die beim Zusammenfassen dieser Konflikte hilft. Es gibt verschiedene Möglichkeiten , um Konflikte zu behandeln:

- Konflikte können als gelöst markiert werden.
- Eine spezielles GUI für die Konflikt-Auflösung kann geöffnet werden.
- Ein Konflikt kann einem speziellen Benutzer zugewiesen werden, der den Konflikt bearbeiten und auflösen kann. Vorkonfiguriert ist dafür der Benutzer, der die am Konflikt beteiligten Objekte zuletzt bearbeitet hat.
-

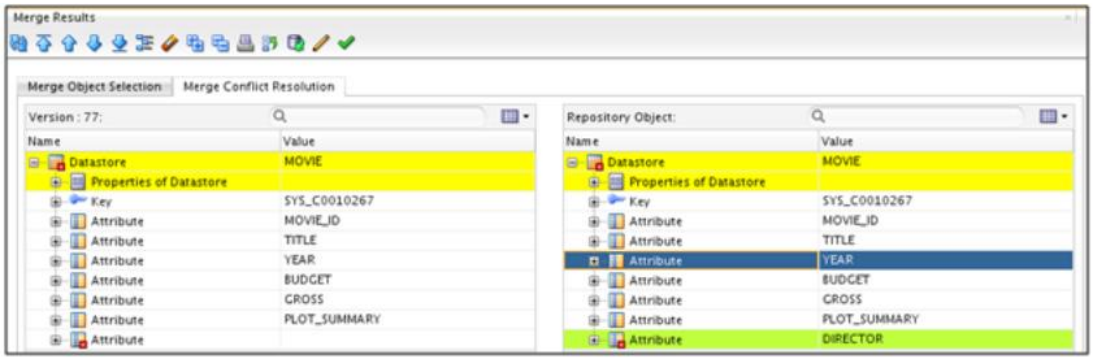

Abb. 9: Auflösen von Konflikten

Deployment Archiv

Es gibt vier Typen von Deployment Archiven:

1. Inital Deployment Archiv – verbunden mit einen neuen ODI Repository, das alle Objekte des Deployment Projektes enthält.
2. Patch Deployment Archive – um Patches mit Bug Fixes oder Erweiterungen in ein initiales Deployment Archive einzuspielen. Dabei werden i. a. nur die Objekte mit den Änderungen und deren Abhängigkeiten eingespielt.
3. Rollback Deployment Archive – wird benutzt, um zu einem Zustand vor einem Patch zurückzugehen. Es wird erstellt, um zu einem Deployment Archive zurückzugehen und enthält alle notwendigen Artifakte, die von dem Patch betroffen waren.

4. Execution Deployment Archive – hält nur die ausführbaren Objekte und wird vom Initialen oder vom Patch Deloyment Archiv erzeugt.

Die Deployment Archive können vom ODI Studio unter „Team/Deployment Archives“ bearbeitet werden. Dazu ist der Name und der Pfad des Archives anzugeben sowie optional ein Schlüssel und zusätzliche Topolgy Informationen für ODI.

Deployment Archive können einzelne Objekte, die über ein VCS Label (Tag) erstellt werden, oder alle Objekte eines Repository enthalten. Zusätzlich gibt es die Möglichkeit, die Objekte für ein Deployment Archive auswählen. Auch in diesem Fall werden abhängige Objekte automatisch hinzugefügt.

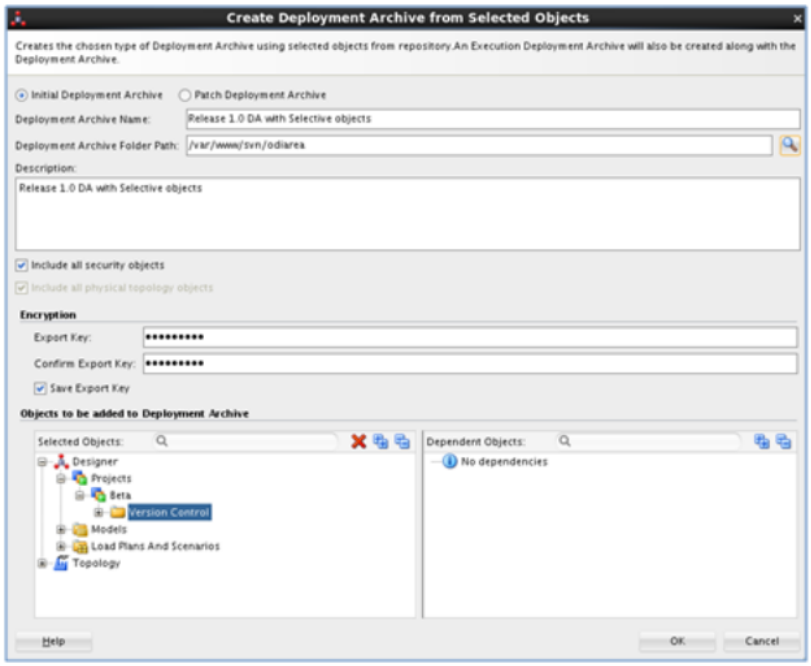

Abb. 10: Erstellen eines Deployment Archives

Im ODI Studio gibt es Übersichten, um sich alle Deployment Archive anzusehen. Mit einem Doppelklick kann man auch die Anhalte ansehen.

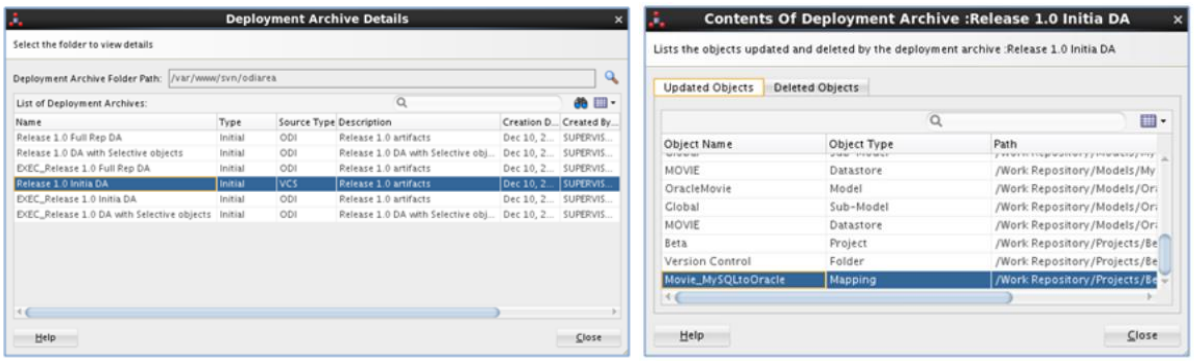

Abb. 11: Deployment Archive in der Übersicht

Ein leeres Repositoy kann mit einem Initaial Deployment Archive gefüllt werden. Anschließend kann auch Deployment Archive importiert werden. Danach ist es auch möglich, Rollback Archive einzuspielen. Die in der nachfolgenden Abbildung gezeigten Übersichten können angepaßt werden, so dass z. B. der Typ des Deployments Archivs, Zeitpunkte des Erstellens oder die Anzahl der Objekte mit angezeigt werden.

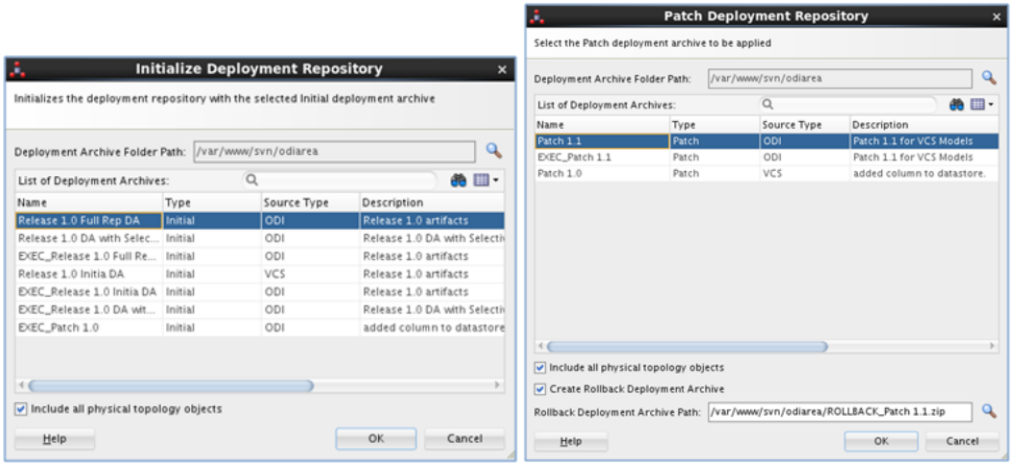

Abb.12: Import von Deployment Archiven

Mit Rollback Archiven können die gleichen Operationen durchgeführt werden. Dabei ist jedoch zu berücksichtigen, dass alle Bedingungen dafür vollständig erfüllt sein müssen.
Es ist möglich, alle Release Management Funktionen über das ODI SDK auszuführen, um die hier gezeigten Vorgehensweisen in ODI zu automatisieren.

Ausblick
Mit ODI 12.2.1 ist es möglich, Lifecycle Management Anforderungen mit einem Standard-Werkzeug umzusetzen. Bei älteren Werkzeugen wie dem Oracle Warehouse Builder wurde diese Funktionalität immer wieder gefordert, mit ODI wurde sie jetzt auch umgesetzt.

Weiterführende Hinweise
https://blogs.oracle.com/dataintegration/entry/oracle_data_integrator_12_21, Zugriff am 17.10.2016
https://blogs.oracle.com/dataintegration/entry/oracle_data_integrator_12_22, Zugriff am 17.10.2016
https://blogs.oracle.com/dataintegration/entry/oracle_data_integrator_12_23, Zugriff am 17.10.2016

Kontaktadresse:

Dr. Holger Dresing
Oracle Deutschland B.V. & Co. KG
Thurnithistraße 2
D-30519 Hannover

Telefon: +49 (0) 95787-118
Fax: +49 (0) 95787-118
E-Mail holger.dresing@oracle.com
Internet: www.oracle.de