

# Migration von OWB nach ODI

## Aktueller Stand und neue Entwicklungen

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

**Schlüsselworte: Oracle Data Integrator – ODI, Oracle Warehouse Builder – OWB, Migration, ETL, Datenmigration**

### 1. Einleitung

Schon Anfang 2009 hat Oracle angekündigt, den Oracle Data Integrator (ODI) und den Oracle Warehouse Builder (OWB) zu einem Produkt zusammenzufassen. Im Statement of Direction (SoD) wird von einem ODI 12c gesprochen, der alle Funktionen sowohl vom ODI als auch vom OWB abdeckt und der den OWB ersetzt. Dieses SoD wurde Ende März 2011 nochmal verfeinert. Seitdem ist klar, dass der OWB 11.2 der letzte OWB und gegen eine Oracle Datenbank 12.1 lauffähig ist.

Beide Werkzeuge decken typische Integrationsaufgaben wie z. B. ETL-Prozesse ab. Es gibt jedoch grundsätzliche Unterschiede: Der OWB bietet eine graphische Benutzeroberfläche, um typische Aufgaben beim Entwickeln von ETL-Prozessen in einer Oracle Datenbank zu implementieren und er nutzt dazu PL/SQL, um den Code für diese Prozesse zu generieren. Der ODI arbeitet ebenso Datenbank-basiert mit einer graphischen Benutzeroberfläche, aber er generiert SQL-Skripte, die zur Laufzeit ausgeführt werden und die der Entwickler durch Code Templates/Knowledge Module beeinflussen kann. Dadurch ist er heterogen auch mit Technologie (Datenbanken) anderer Hersteller ohne zusätzliche Softwarekomponenten wie den Oracle Datenbank-Gateways einsetzbar. Er lässt sich anwendungsbezogen sehr leicht erweitern und die Erfahrungen zeigen, dass sich Datenintegrationsprojekte sehr viel schneller umsetzen lassen und dadurch erheblich geringere Kosten verursachen.

Nichtsdestotrotz war der Oracle Warehouse Builder ein sehr erfolgreiches Produkt, das viele Oracle Anwender für Ihre ETL-Prozesse eingesetzt und in welches sie erhebliche Investitionen getätigt haben. Für die meisten dieser Anwender stellt sich die Frage, wie der Umstieg am schnellsten und kostengünstigsten zu bewerkstelligen ist. Neben der Migration, also dem Umstellen der ETL Prozesse von dem „Werkzeug OWB“ in das „Werkzeug ODI“, gibt es noch weitere Fragen, die an dieser Stelle aber nicht weiter betrachtet werden sollen. Dazu gehören die Unterschiede zwischen den Werkzeugen hinsichtlich Funktionalität, Architektur, Hardwareanforderungen, Administration und Umschulung der Mitarbeiter.

Oracle bietet für die Migration zwei Wege an, die an dieser Stelle betrachtet werden:

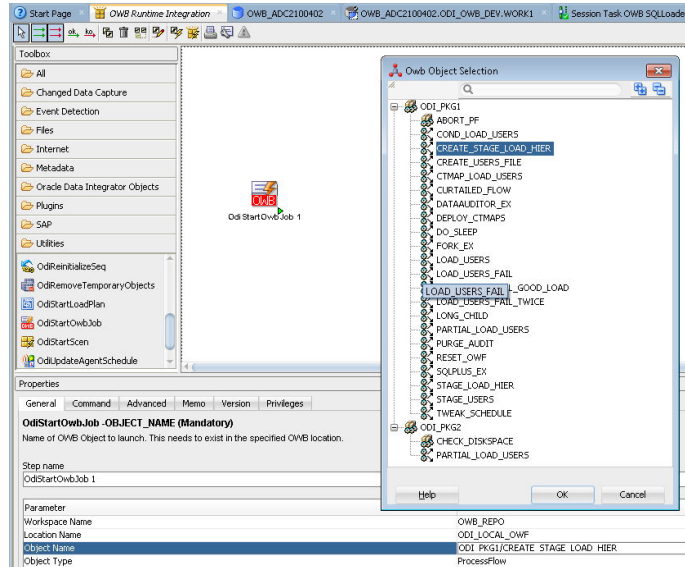
- Eine schrittweise Migration: ODI 12c unterstützt die Ausführung und Verwaltung von OWB 11gR2 Jobs direkt im ODI-Studio und der ODI-Konsole, und bietet so eine gemeinsame Orchestrierungs- und Monitoring-Lösung. Diese Funktion ermöglicht, OWB-Objekte über einen längeren Zeitraum zu migrieren.
- Eine vollständige Migration: Der ODI 12c unterstützt eine einfache Zuordnung von OWB 11gR2 Konzepten und Objekten zu denen des ODI. Zusätzlich wird ein Migrationswerkzeug angeboten, welches automatisch viele OWB-Objekte und -Zuordnungen in ODI-Äquivalente übersetzen kann.

Aus Sicht von Oracle ist auch diese Anmerkung wichtig: Niemand ist gezwungen, eine Migration durchzuführen. Der Anwender kann mit dem alten OWB weiter arbeiten solange er das möchte.

## 2. Schrittweise Migration

Im ODI gibt es eine neue Technologie mit der Bezeichnung OWB Repository (*Connection Manager*<sup>1</sup>). Mit dieser Technology wird eine Verbindung zum Runtime Repository im OWB aufgebaut. In Packages (*Business Flows*) können „deployed“ Mappings und Business Flow aufgerufen und genutzt werden. Dazu gibt es im ODI ein Open Tool (*Operator*), das wie alle andere in einem Workflow verwendet werden kann. Um diese zu pflegen oder anzupassen, ist weiterhin eine OWB Umgebung erforderlich. Aufrufen und Administrieren kann man sie über ODI.

Die dahinterliegende Vorstellung ist, dass neue ETL-Projekte mit ODI erstellt werden. Spätestens mit dem produktiven Einsatz dieses Projektes wird eine ODI Laufzeitumgebung benötigt. Der Anwender ist auch in der Lage, Mappings, die im OWB erstellt wurden, in der neuen ODI Umgebung auszuführen. Damit könnten die vorhanden ETL-Projekte, die mit dem OWB erstellt wurden, schrittweise nach ODI überführt werden. Die Erfahrung zeigt, dass fortlaufend Änderungen an den Quell- und Zielstrukturen erfolgreicher Data Warehouses durchgeführt werden und somit häufig auch eine schrittweise Migration möglich ist.



## 3. Vollständige Migration

Das Migrationswerkzeug ist ein Kommandozeilenprogramm, mit dem die Metadaten aus dem OWB Design Repository in das ODI Repository migriert werden können. Die Laufzeitdaten und die Objekte in der Datenbank werden nicht migriert.

### 3.1 Voraussetzungen, Installation und Aufruf des Migrationswerkzeugs

Um die Migration durchführen zu können, wird der OWB 11.2.0.3 oder 11.2.0.4 benötigt. Migriert wird nach ODI 12c (12.1.2.0.0). Das Migrationswerkzeug arbeitet im „line mode“ und nutzt das OWB SDK (auch OMB Plus Entwicklungsumgebung genannt) und das ODI SDK. Dabei werden die Metadaten direkt vom OWB Repository in das ODI Repository geschrieben. Für die Planung kann man diesen Vorgang auch simulieren, um zunächst prüfen zu können, inwieweit eine automatische Migration möglich ist.

Das Migrationswerkzeug besteht aus zwei Patches. Um es zu installieren, müssen beim OWB der Patch #17830453 und beim ODI der Patch #17053768 eingespielt werden. Das Migrationswerkzeug

<sup>1</sup> Im Text werden immer zuerst die Begriffe genannt, die der ODI verwendet. Um es OWB Anwendern zu erleichtern, diese Begriffe einzuordnen, werden in kursiver Schrift die vergleichbaren Begriffe aus dem OWB ergänzt. Die Begriffe passen nicht immer vollständig, aber sie geben einen Hinweis.

steht unter Linux und Windows 64 bereit. Danach gibt es unter Linux das Shellskript für die Migration, das wie folgt aufgerufen wird:

```
>$OWBHOME/bin/unix/migration.sh <odi_master_repos_pwd> <odi_user_pwd>
<owb_wksp_owner_pwd> <config_file>
mit den Parametern: <odi_master_repos_pwd> ODI Master Repository Passwort
                   <odi_user_pwd>         ODI User Passwort
                   <owb_wksp_owner_pwd>   OWB Workspace Owner Passwort
                   <config_file>         Konfigurationsdatei
```

Die Konfigurationsdatei enthält zusätzliche Informationen, z. B. die Verbindungsinformationen zum OWB und ODI sowie die zu migrierenden Objekte:

Konfigurationsdatei	Erläuterungen
ODI_MASTER_USER=ODIREP	Schemaname für das ODI Master Repository
ODI_MASTER_URL=jdbc:oracle:thin:@localhost:1521:machine ODI_MASTER_DRIVER=oracle.jdbc.OracleDriver	JDBC-Verbindung zum ODI Master Repository
ODI_USERNAME=SUPERVISOR	Administrator für die ODI Umgebung
ODI_WORK_REPOSITORY_NAME=WORK0	Name des ODI Work Repository
OWB_WORKSPACE_OWNER=rep_0	OWB Workspace Owner
OWB_URL=localhost:1521:machine.oracle.com	JDBC-Verbindung zum OWB Workspace
OWB_WORKSPACE_NAME=REP_0_WS_0	Name des OWB Workspaces
MIGRATION_LOG_FILE=/tmp/migration.log	Pfad für die Report Dateien, die das Ergebnis der Migration dokumentieren
MIGRATION_REPORT_INCLUDE=ALL	Die Report Dateien können alle (ALL), nur die fehlerhaften (FAILED) oder die erfolgreich (PASSED) migrierten Objekte enthalten.
MIGRATION_MODE=RUN	Im Modus FAST_CHECK wird die Migration nur geprüft, im Modus DRY_RUN werden die Migration ohne Commit durchgeführt und im Modus RUN wird die Migration durchgeführt.
MIGRATE_DEPENDENCIES=TRUE	Im Fall TRUE werden auch alle abhängigen Objekt migriert, im Fall FALSE werden nur die angegebenen Objekte migriert.
STOP_ON_ERROR=FALSE	Im Fehlerfall wird die Migration nicht abgebrochen (FALSE), alternative: TRUE
SPLIT_JOIN_FOR_ANSI_SYNTAX=TRUE	Es wird die ANSI Syntax für einen Join verwendet (TRUE), alternative: FALSE
MIGRATE_UNBOUND_OPERATOR=TRUE	Legt fest, ob UNBOUND Operatoren mit migriert werden sollen (TRUE). UNBOUND Operatoren enthalten z. B. external Tables, materialized Views oder pluggable Mappings, alternative: FALSE
MIGRATION_OBJECTS=PROJECT.MY_PROJECT	Hier werden die zu migrierenden Objekte angegeben, z. B. Projekte, Tabellen oder Mappings. Ein ,*' steht für alle Objekte.
FLUSH_BATCH_SIZE=50	Gibt die Anzahl der parallel abzuarbeitenden Objekte bei der Migration an.

### 3.2 Migrierbare und nicht migrierbare Objekte

Grundsätzlich können alle Datenobjekte und Mappings migriert werden. Aber wie immer gibt es eine Reihe von Besonderheiten, die hier aus Platzgründen nicht aufgeführt werden konnten.<sup>2</sup> Nicht migrierbar sind die Workflows oder Business Flows aus dem OWB. Zum Ausführen verwendet der

<sup>2</sup> Siehe Dokumentation „Oracle Fusion Middleware: Migration vom Oracle Warehouse Builder to Oracle Data Integrator 12c (12.1.2), E50186-06, Januar 2014

OWB die Oracle Workflow Option, die nur noch für den OWB in der Datenbank zertifiziert war. Ebenso werden zur Zeit noch keine multidimensionalen Metadaten im ODI verwaltet, diese werden daher auch noch nicht migriert. Daher ergibt sich auch der Punkt, dass die Weitergabe von multidimensionalen Strukturen an den Oracle Discoverer oder Oracle Business Intelligence nicht unterstützt ist. Die Funktionalität der OWB Data Quality Option wird heute durch Oracle Enterprise Data Quality abgedeckt. Dafür gibt es keine Migrationsunterstützung. In PL/SQL erstellte Routinen im OWB werden nicht nach ODI überführt. Diese Verwaltung des Source Codes unterliegt dem Anwender. Die Ausrufe z. B. in Expressions werden jedoch migriert. Die Funktionalität der OMB Plus Skripte und der Experts werden im ODI durch das ODI SDK abgedeckt. Die Migration muß der Anwender manuell durchführen.

<b>Migrierbare Objekte</b>	
<ul style="list-style-type: none"> <li>▪ Module and Location</li> <li>▪ Data Objects               <ul style="list-style-type: none"> <li>▪ Table (Columns, Keys, Indexes)</li> <li>▪ View (Columns, Keys)</li> <li>▪ Materialized View (Columns, Keys, Indexes)</li> <li>▪ External Table (Columns)</li> <li>▪ File (Records, Fields)</li> <li>▪ Sequence</li> </ul> </li> </ul>	<ul style="list-style-type: none"> <li>▪ Mappings               <ul style="list-style-type: none"> <li>▪ Classic Mappings, Code Template Mappings, Pluggable Mappings</li> <li>▪ Most Mapping Operators                   <ul style="list-style-type: none"> <li>▪ Aggregator, Constant, Deduplicator, Expression, External Table, Filter, Flat File, Joiner, Lookup, Input parameters, Materialized View, Pluggable Mapping, Pivot, Pre/Post Mapping, Sequence, Set, Sorter, Splitter, Subquery, Table, Table Function, Transform-Function, Unpivot, View</li> </ul> </li> </ul> </li> </ul>

### 3.3 Ergebnisse der Migration

Jeder Aufruf des Migrationswerkzeugs erzeugt zwei Report Dateien, in denen die Ergebnisse der Migration gespeichert sind: Die Log Datei mit den migrierten Objekten und evtl. den aufgetretenen Fehlern und dem Exclusion Record mit einer statistischen Übersicht aller migrierten Objekte und den Begründungen, falls Objekte nicht migriert werden konnten.

Beispiel:

```

*****
* Oracle Warehouse Builder - Migration Utility - Summary Report
* Created: 9/30/13 10:17 AM
* Migration Report Style - RUN
*
* OWB Release:11.2.0.4.0 - OWB Repository:
OWB_REPO_MIG/machine.oracle.com:1521:orcl11204 - OWB Workspace:
OWB_REPO_MIG.OWB_REPO_WKSPI
*
* ODI Release:12.1.1.2 - ODI Master Repository:
mig12c/jdbc:oracle:thin:@machine:1521:orcl11203 - ODI User/Work Repository:
SUPERVISOR/WORKREP1
*
* Report File: /scratch/jsmith/Migration/migration.report
*****
Migration started at 9/30/13 10:17 AM Pacific Standard Time
Statistics
-----
Total Projects Migrated: 2
*****
PROJECT: PUBLIC_PROJECT
Object      Types Migrated Not-Migrated
-----
LOCATION:                1          0
*****
PROJECT: MY_PROJECT
Object Types Migrated Not-Migrated
-----
TABLE:                  2          0
MAPPING_MODULE:        1          0
MODULE:                 1          0
MAPPING:                1          0
Details
-----
*****
PROJECT: PUBLIC_PROJECT
Object Types      Status
-----

```

```

LOCATION
ORACLE_WH LOCATION SUCCESS
*****
PROJECT: MY_PROJECT
Object Types Status
-----
TABLE
TAB1          SUCCESS
TAB2          SUCCESS
MAPPING_MODULE
AA_UNBOUND_TEST SUCCESS
MODULE
AA_UNBOUND_TEST SUCCESS
MAPPING
MAP_UO_CASE2  SUCCESS
MAPPING
UNBOUND_LOOKUP_MAP [ERROR][Migration][MU-5011]Unable to find bound table for
output attribute group OUTGRP1 in Lookup DEPT LOOKUP.
Migration ended at 9/30/13 10:17 AM Pacific Standard Time
Total migration time (hh:mm:ss): 00:00:07

```

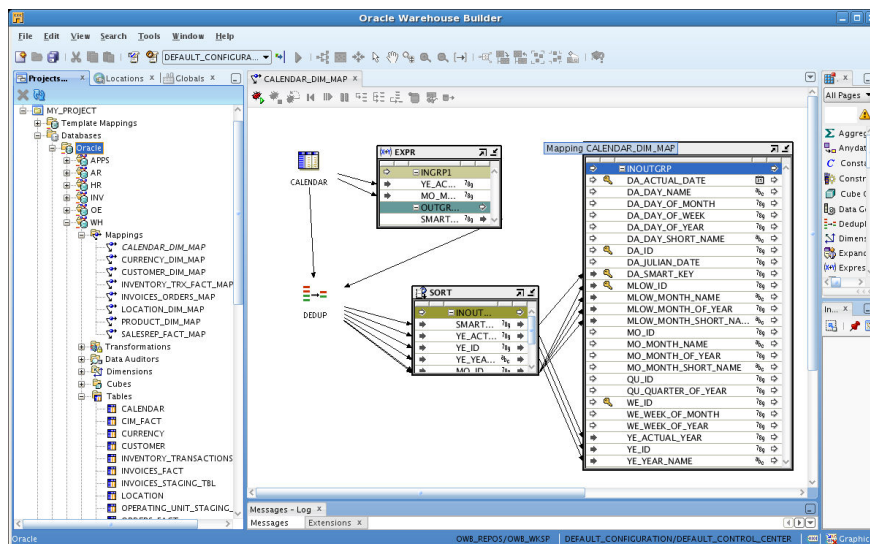
Während der Migration werden keine weiteren Zwischendateien oder Export-Dateien geschrieben, die ein Anwender abgreifen kann.

Das Migrationswerkzeug kann wiederholt aufgerufen werden. Es benötigt im Modus FAST\_CHECK nur wenige Minuten. Ein Mapping wird nur migriert, wenn alle Objekte migrierbar sind. Sollten also beispielsweise einzelne Expressions nicht migrierbar sein, muß man sich Strategien überlegen, um möglichst viele Teile migrieren zu können. Eine Möglichkeit ist, solche nicht migrierbaren Expressions mit OMB Plus herauszufiltern, dann die Migration soweit möglich auszuführen, um anschließend nur noch die herausgefilterten Teile wieder manuell nachzupflegen.

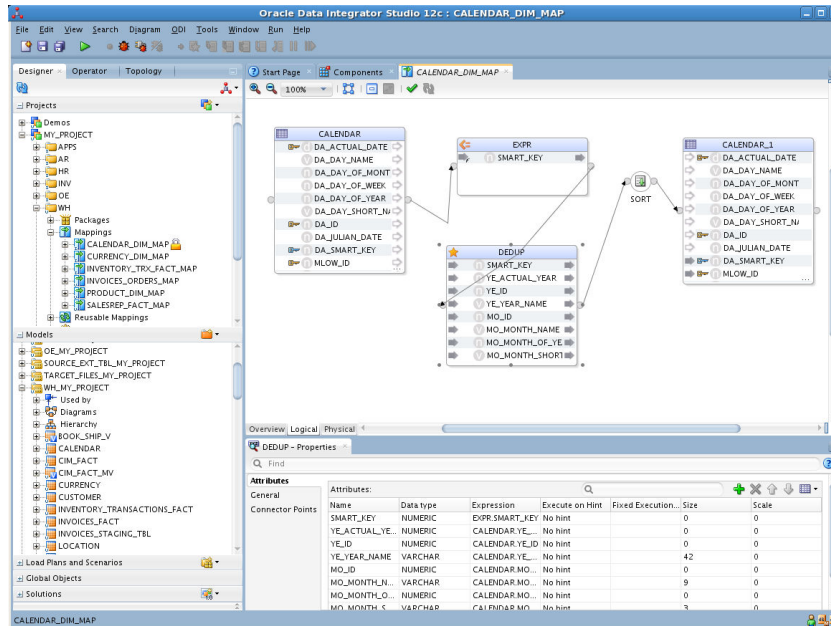
Die Erfahrungen zeigen, dass die Ergebnisse für einen OWB Entwickler auch im ODI sehr gut nachvollziehbar sind.

### 3.4 Beispiel für eine Migration

In der Oracle OWB Standard Demo gibt es ein Projekt MY\_PROJECT mit einem Modul WH und einem Mapping CALENDAR\_DIM\_MAP. Dieses Mapping sieht wie folgt aus:



Wenn dieses Mapping einschließlich der Locations, Tabellen und Mappings nach ODI mit dem Migrationswerkzeug basierend auf der hier vorgestellten Konfigurationsdatei ausgeführt wird, erhält man im ODI das Mapping, wie es im Screenshot auf der nächsten Seiten zu sehen ist. Ebenso kann man die Topology im ODI untersuchen und stellt fest, dass auch die OWB Location umgesetzt wurden.



#### 4. Fazit

Es läßt sich schnell erkennen, dass sich Datenobjekte und Mappings im OWB weitgehend automatisiert umsetzen lassen, zumindest solange nur Basic ETL Features benutzt wurden. Viele OWB Entwickler fragen

- (1) nach Pre-/Post-Mappings,
- (2) nach Expressions- und Transformations,
- (3) nach Tabellenobjekten, oder
- (4) Tabellenobjekte, die sowohl als Quelle als auch als Ziel benutzt werden.

Für diese Fälle gibt es unterschiedliche Lösungen:

- (1) Pre-/Postmappings werden zu PL/SQL-Aufrufen, die über einen Knowledge Modul Operator aufrufbar werden,
- (2) in ODI wird nicht mehr nach Expressions und Transformationen unterschieden,
- (3) den Operator Tabellenobjekt gibt es im ODI nicht, wohl aber die Möglichkeit, in den Modellen neue Tabellen anzulegen, in Mappings zu nutzen und beim ersten Durchlauf des Mappings die Objekte anzulegen, oder
- (4) Sonderfälle, die nicht automatisch migrierbar sind bzw. erst so bearbeitet werden müssen, dass sie migrierbar werden (hier: die Tabelle mit einem Alias ein zweites Mal in OWB einfügen).

Wenn die OWB Anwender das Werkzeug in großer Zahl erfolgreich nutzen und die Schwachstellen dokumentieren, kann erwartet werden, dass Oracle in der Zukunft weitere Features einbaut. Es kann aber auch für Consulting Dienstleister die Möglichkeit sein, sich erfolgreich in Migrationsprojekte einzubringen und somit die Migration der Anwender von OWB nach ODI noch einfacher und leistungsfähiger zu gestalten.

Kontaktadresse:

Dr. Holger Dresing  
 Oracle Deutschland B.V. & Co. KG  
 Data Integration Solutions DIS - EMEA  
 Thurnithstr. 2  
 D-30519 Hannover

Telefon/Fax: +49 511-95787 118  
 E-Mail: holger.dresing@oracle.com  
 Internet: www.oracle.com