

Features zugeteilt werden. Anhand der Ergebnisse reduziert sich die Anzahl der Hersteller weiter, sodass die finalen Kandidaten zu einem intensiven, mehrtägigen „Proof of Technology“-Workshop eingeladen werden können. In jedem dieser Schritte gilt die Prämisse: „Keine Angst, Kandidaten auszusortieren.“

Im „Proof of Technology“-Workshop werden final die gestellten Anforderungen nochmals live geprüft. Es empfiehlt sich, während der gesamten Dauer Spezialisten aus dem eigenen Unternehmen den Workshop begleiten zu lassen. Dies

stellt sicher, dass auch im Nachgang bereits das Wissen darüber existiert, wie die gestellten Anforderungen umgesetzt werden können.

Fazit

Der Artikel zeigt anhand von Beispielen die verschiedenen Möglichkeiten des Umstiegs auf ein anderes ETL-Tool. Da jedes Tool seine Vor- und Nachteile hat, ist es umso wichtiger, die Anforderungen, individuell auf das eigene Umfeld betrachtet, festzulegen und in einem Tool-Auswahlprozess zu bewerten.



Michael Klose
michael.klose@cgi.com

OWB ohne OWB: Wie rette ich meine ETL-Sourcen nach 12c R2?

Sven Bosinger, its-people GmbH

Oracle hat angekündigt, den Oracle Warehouse Builder (OWB) ab der Datenbank-Version 12c R2 nicht mehr zu unterstützen. Dies hat zur Folge, dass alle ETL-Prozesse, die mit dem OWB erstellt wurden, in ein neues Werkzeug migriert werden müssen. Oracle empfiehlt hier den Oracle Data Integrator (ODI). Doch ist eine Migration immer notwendig?

Der Artikel stellt eine Möglichkeit vor, die mit dem OWB erzeugten ETLs ohne den OWB und damit potenziell auch unter 12c R2 weiter zu nutzen. Dazu wurde ein ETL-Workframe geschaffen, der die fehlenden OWB-Runtime-Komponenten ersetzt und so eine Lauffähigkeit der Sourcen ohne Installation des OWB ermöglicht.

Motivation

Im Januar 2010 und später noch einmal konkretisiert im Oktober 2013 hat Oracle folgendes Statement of Direction veröffentlicht: „No major enhancements are planned for Oracle Warehouse Builder beyond the OWB 11.2 release. OWB 11.2 continues to be available and supported by Oracle, and patches and bug fixes will continue to be offered at regular intervals. Oracle will conti-

nue to support OWB 11.2 for the full lifetime of Oracle Database 11g in accordance with Oracle's Lifetime Support Policies for Oracle Database releases. Future database releases beyond Oracle Database 12c Release 1 will not be certified with OWB 11.2.“ Daraus ergeben sich für die Nutzer des OWB folgende Konsequenzen:

- Wer OWB einsetzt, muss sich zeitnah Gedanken über eine Migrationsstrategie machen.
- Die Datenbank 12c R2 ist für Ende 2015 / Anfang 2016 angekündigt. Wer diese einsetzen möchte, muss zuvor alle OWB-Mappings migrieren.
- Der Premium-Support für 12c R1 endet im Juli 2018, der Extended Support im Juli 2021. Ein Betrieb über den Juli 2018

hinaus zieht zusätzliche Support-Kosten nach sich. Nach dem Juli 2021 ist ein Support in der Regel nicht mehr möglich.

- Das von Oracle präferierte Nachfolgeprodukt des OWB ist der Oracle Data Integrator (ODI). Eine Migration dorthin ist aufwändig und zieht Kosten nach sich. Der ODI deckt nicht alle Funktionalitäten des OWB ab. Es ist davon auszugehen, dass bei einer durchschnittlichen DWH-Datenbewirtschaftung maximal 90 Prozent der ETL-Sourcen „1:1“ migriert werden können. Für die restlichen 10 Prozent ist eine zum Teil aufwändige Neuentwicklung notwendig (Erfahrungswerte).
- Der ODI ist nicht wie der OWB (Standard Edition) in den Lizenzkosten der Datenbank enthalten, sondern muss zusätzlich lizenziert werden.

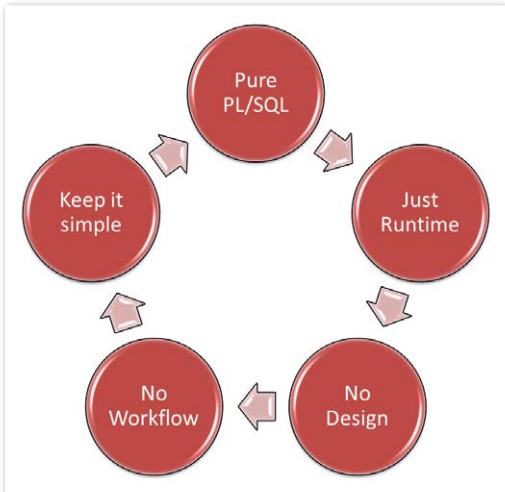


Abbildung 1: Design-Prinzipien des ETL-Workframes

- Da eine „1:1“-Migration der ETL-Sourcen hin zum ODI nicht immer möglich ist, muss für die Änderungen eine gegebenenfalls aufwändige Testphase eingeplant werden.

Die genannten Punkte bedeuten in Summe, dass heutige OWB-Nutzer massiven Handlungsdruck haben und damit einhergehende Migrationskosten für die nächsten Jahre planen müssen. Betrachtet man diese durch Oracle gesetzten Rahmenbedingungen, lohnt es sich, in diesem Zusammenhang etwas genauer hinzuschauen, worum es sich beim OWB technologisch gesehen eigentlich handelt.

OWB ist ein ETL-Werkzeug, das ausschließlich Tabellen in einer Oracle-Datenbank befüllen kann. Es besteht aus zwei Modulen – der Design- und der Runtime-Komponente. In der Design-Komponente wird mithilfe einer grafischen Oberfläche ein Mapping modelliert, also ein Datentransfer zwischen einer Quell- und einer Ziel-Tabelle. Um einen lauffähigen Code zu erhalten, muss man das Mapping nach abgeschlossener Design-Phase generieren. Das Ergebnis ist ein PL/SQL-Package, das in der Lage ist, den Datentransfer durchzuführen. Es wird dann über die Runtime-Komponente aufgerufen. Diese stellt zudem ein Repository zur Verfügung, in dem die Laufzeitinformationen der Packages gespeichert sind.

Es gibt keinerlei Anzeichen dafür, dass PL/SQL über 12c R2 hinaus nicht mehr zur Verfügung steht. Somit müsste theoretisch ein einmal generiertes Package auch weiterhin in einer Umgebung, die PL/SQL unterstützt, lauffähig sein. Das Runtime-

Repository, das OWB bereitstellt, müsste durch eine Eigenentwicklung substituierbar sein. Der im Weiteren vorgestellte ETL-Workframe basiert auf diesen Annahmen und stellt eine technische Umsetzung dieser Ideen dar. Es handelt sich hierbei nicht um ein Produkt, sondern um einen Consulting-Softframe.

Der ETL-Workframe

Der durch die Firma its-people entwickelte ETL-Workframe stellt eine Möglichkeit dar, mit dem OWB generierte Mappings in einer OWB-freien Umgebung lauffähig zu halten. Mit diesem Ansatz werden folgende Ziele verfolgt:

- Der zeitliche Druck, schon bei der Migration auf Oracle Database 12c R2 alle ETLs, die mit dem OWB erzeugt worden sind, auf ein neues Produkt umgestellt zu haben, soll entschärft werden.
- Fehlende Funktionen (wie Premap- und Postmap-Trigger) des neuen ETL-Werkzeugs (etwa ODI) können durch partiellen Einsatz der alten OWB-Mappings ausgeglichen werden.
- Die Migrationskosten sinken, indem komplett vorhandene ETL-Strecken (Mappings) in das neue zu beschaffende ETL-Werkzeug integriert werden, sofern dieses in der Lage ist, PL/SQL-Blöcke zu starten.
- Aufwändige Tests der „1:1“ übernommenen ETL-Sourcen entfallen, da diese beim Einsatz des ETL-Workframes nicht verändert und somit nicht erneut getestet werden müssen.

Der ETL-Workframe selbst ist eine reine PL/SQL-Entwicklung. Somit ist eine prinzipielle Unabhängigkeit gegenüber dedizierten Datenbank-Releases gegeben. Der ETL-Workframe und die durch ihn gekapselten Mappings können also auch jenseits von 12c R1 betrieben werden (siehe Abbildung 1).

Der ETL-Workframe deckt lediglich die Runtime-Komponente des OWB ab, da es hier nicht um die Entwicklung eines neuen ETL-Werkzeugs geht, sondern nur um die Lauffähigkeit der generierten Mappings. Ebenso werden hierbei Workflow-Themen nicht betrachtet.

Anwendungsfall

Zur Verdeutlichung der Funktionsweise des ETL-Workframes wird im Folgenden auf ein einfaches Beispiel anhand eines OWB-Mappings mit Namen „MAP_KUNDEN“ Bezug genommen (siehe Abbildung 2). Die Inhalte der Stage-Tabelle „STG_KUNDEN“ werden in die Tabelle „DWH_KUNDE“ integriert, dabei ermittelt eine PL/SQL-Funktion eine Bonitäts-Kennzahl und formatiert über eine Expression diverse Spalten um. Durch einen Premap-Trigger wird in einer Protokoll-Tabelle ein Eintrag gesetzt. Zusätzlich erzeugt eine Sequenz ein Surrogate-Key.

Betrachtet man den PL/SQL-Code des generierten Mappings, kann man leicht die Schnittstellen zum OWB-Runtime-Repository erkennen. Durch das Mapping werden drei durch den OWB bereitgestellte Packages aufgerufen:

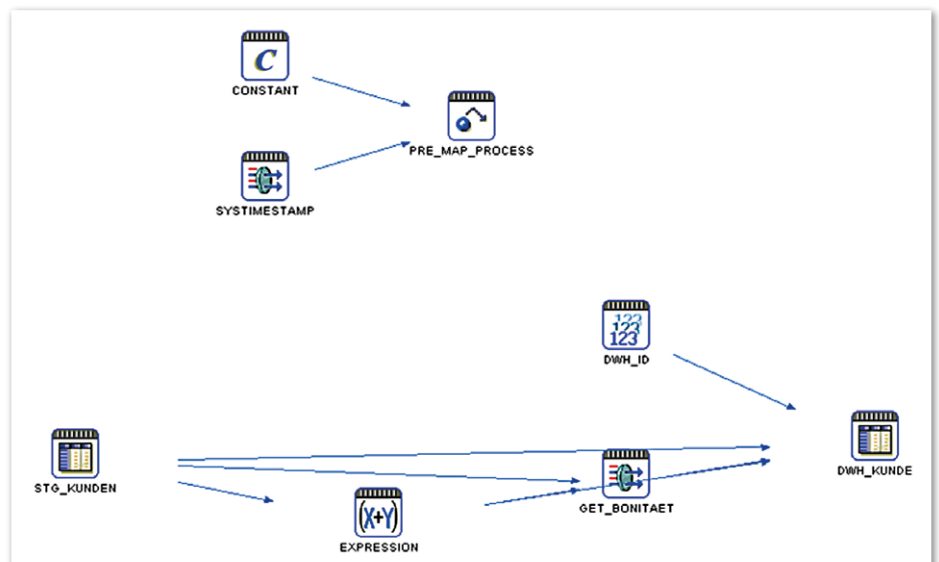


Abbildung 2: Einfaches OWB-Mapping

- WB_RT_MAPAUDIT
- WB_RT_MAPAUDIT_UTIL
- WB_RT_MAPAUDIT_UTIL_INVOKER

Alle drei Packages befinden sich im Schema „OWBSYS“. Da eine Veränderung des Source-Codes der Mappings nicht vorgenommen werden soll, ist es notwendig, in der OWB-freien Datenbank ein Schema „OWBSYS“ anzulegen und dort die drei genannten Packages durch eine Eigenentwicklung nachzubilden. Dazu gilt es zu betrachten, welche Funktionalitäten diese bereitstellen.

Im Package „WB_RT_MAPAUDIT“ sind alle Protokollierungen hinsichtlich des Runtime-Repository gebündelt. Insbesondere werden folgende Prozeduren/Funktionen benötigt:

- AUDITD_BEGIN
- AUDIT_BEGIN
- REGISTER_AUDIT_DETAIL_TYPE
- ERROR

- AUDITD_END
- AUDIT_FAIL
- ERROR_SOURCE
- REGISTER_FEEDBACK
- REGISTER_FEEDBACK_BULK
- AUDITD_PROGRESS
- LOG_FILE

Die Runtime-Repository-Tabellen des OWB sind ein einfaches Datenmodell, das über eine View-Schicht angesprochen wird. Der Aufbau kann der OWB-Dokumentation entnommen werden. Baut man das Package „WB_RT_MAPAUDIT“ nach, indem man seine eigenen Runtime-Tabellen entwirft und die entsprechenden OWBSYS-Calls auf das eigene Repository umleitet, so kann man einfach eine eigene Protokollierung implementieren. Es ist dabei nicht zwingend notwendig, sich an den Aufbau des OWB-Repository zu halten. Lediglich die Package-Calls aus dem Mapping müssen über entsprechende

Wrapper-Prozeduren/-Funktionen weiterhin unterstützt werden.

Im Package „WB_RT_MAPAUDIT_UTIL“ sind alle Hilfsfunktionen hinsichtlich des Mapping-Ablaufs gebündelt. Insbesondere werden folgende Prozeduren/Funktionen benötigt:

- SUPPORTSDESIGNCLIENT
- RESOLVE_NAME
- VALIDATE_RUNTIME_PARAMETER
- PREMAP
- POSTMAP
- REGISTER_SYS_PARAM
- SHOW_RUN_RESULTS
- SET_SCHEMA_WORKSPACE

Die Funktionalität, die sich hinter den einzelnen Prozeduren/Funktionen versteckt, ist aus ihrem Namen schon ersichtlich. So prüft beispielsweise die Funktion „SUPPORTSDESIGNCLIENT“, ob die zu den Mappings passende Runtime-Umgebung

Sie wollen wissen. Sie wissen was. Wir wissen das.

dbi InSite
Workshops

Insider-Wissen von IT-Experten: Unsere massgeschneiderten Workshops für Oracle, SQL Server, MySQL, Linux & mehr.

Phone +41 32 422 96 00 · BaselArea · Lausanne · Zürich

dbi-services.com/InSite



Infrastructure at your Service.

dbi services

installiert wurde. Diese Prozeduren/Funktionen müssen durch eine Eigenentwicklung entsprechend bereitgestellt sein.

Das Package „WB_RT_MAPAUDIT_UTIL_INVOKER“ kapselt Aufrufe der durch die Datenbank selbst bereitgestellten Funktionen. Besonders herauszuheben ist dabei die Prozedur „GATHER_TABLE_STATS“, die in der Regel nach einem erfolgreichen Ablauf des Mappings für neue Statistiken sorgt. Die Kapselung der durch den OWB verwendeten Funktionen ist ebenfalls durch ein eigenes Package vorzunehmen.

Der wachsende ETL-Workframe

Ersetzt man die drei beschriebenen Packages durch Eigenentwicklungen und stellt eigene Protokoll-Tabellen zur Verfügung, ist man in der Lage, das Mapping „MAP_KUNDEN“ aus dem obigen Anwendungsfall zu starten und zu überwachen. Damit ist eine Migrationsfähigkeit dieses einfachen Mappings in eine OWB-freie Umgebung gegeben. Auch komplexere Mappings basieren hauptsächlich auf den Funktionen der drei Packages. In Einzelfällen können natürlich noch weitere von OWB bereitgestellte Funktionen notwendig sein, die dann entsprechend nachgebildet werden müssen.

Der ETL-Workframe ist so gestaltet, das man ihn jederzeit erweitern kann. Die im vorliegenden Fall aufgeführten OWB-Objekte sind nicht alle, die potenziell von einem Mapping genutzt werden können. Je nach Funktionalität eines Mappings sind weitere möglich. Daher muss im Einzelfall eine Erweiterung erfolgen. So kann der ETL-Workframe im Laufe der Zeit um weitere Funktionen ergänzt werden (siehe Abbildung 3).

Migration

Hat man durch Nachbildung der Runtime-Komponente dafür gesorgt, dass Mappings auch in einer OWB-freien Umgebung lauffähig sind, kann eine Migration der einzelnen Mappings erfolgen. Um ein einzelnes Mapping in eine OWB-freie Umgebung zu migrieren, sind folgende Schritte notwendig:

- Das zu migrierende Mapping wird hinsichtlich noch nicht vorhandener OWB-Calls untersucht
- Sind zusätzliche Funktionen erforderlich, müssen sie erstellt und in die Funktionsbibliothek im Schema „OWBSYS“ aufgenommen werden

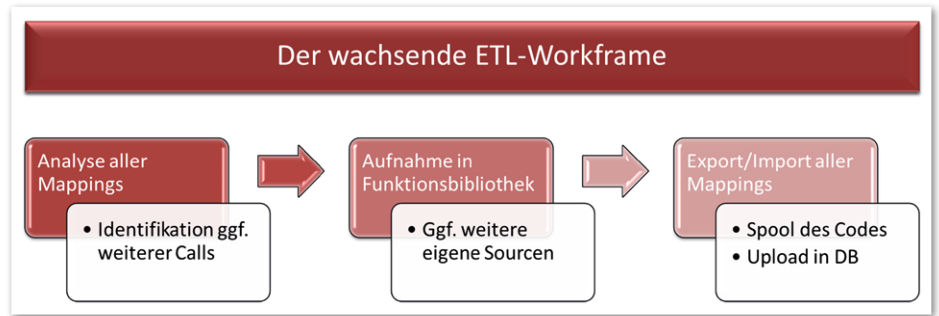


Abbildung 3: Wachsender ETL-Workframe

- Der Source-Code des Mappings wird aus der OWB-Umgebung exportiert; ein einfache Spool der Sourcen reicht aus, um diese in die OWB-freie Ziel-Datenbank zu importieren

Mit diesen drei Schritten kann nahezu jedes Mapping in eine OWB-freie Umgebung migriert und dort lauffähig gehalten werden. Die künstliche Grenze, die durch die Oracle-Datenbank 12c R2 gezogen wird, lässt sich somit zumindest hinsichtlich der Runtime-Komponente auflösen.

Fazit und Abgrenzung

Ziel des ETL-Workframes ist es, den Migrationsdruck hinsichtlich der Datenbewirtschaftung mit dem OWB zu mindern. Dies geschieht dadurch, dass die aus dem OWB heraus generierten Mappings auch über die Version 12c R1 hinaus lauffähig sind. Dabei ist allerdings zu beachten, dass der ETL-Workframe nur die Runtime-Komponente des OWB und nicht die Design-Komponente abbildet. Er ist weder ein vollwertiger ETL-Werkzeug-Ersatz, noch kann er unabhängig von einer zuvor stattgefundenen OWB-Entwicklung gesehen werden. Er leistet den Betrieb eines oder mehrerer generierter Mappings in einer OWB-freien Umgebung. Deshalb ist eine regelmäßige Datenbewirtschaftung durchzuführen. Dies eröffnet heutigen OWB-Nutzern, die vor der Migrationsentscheidung stehen, verschiedene Perspektiven.

Szenario 1: Man setzt zukünftig den ODI oder ein beliebiges anderes ETL-Werkzeug ein. Alle vorhandenen Mappings, die sich einfach „1:1“ migrieren lassen, werden migriert. Alle anderen werden durch den ETL-Workframe gekapselt und aus dem neuen ETL-Werkzeug heraus aufgerufen. Nachteil dieser Strategie ist, dass zusätzlich zu den Migrationskosten noch Lizenzgebühren für das ETL-Werkzeug anfallen.

Szenario 2: Man entscheidet sich für eine Open-Source-Strategie und setzt auch im ETL-Bereich darauf. Dann kann man wie in Szenario 1 verfahren und spart die Lizenzgebühren für das ETL-Werkzeug.

Szenario 3: Man betreibt ohne Support die Design-Komponente des OWB 11.2 auf einer Datenbank 12c R1. Dort erfolgen Wartung und Weiterentwicklung. Für den produktiven Einsatz der Mappings im eigentlichen DWH wird der ETL-Workframe eingesetzt. Dies kann natürlich nur ein Übergangsszenario sein, da auch der Extended Support Lizenzgebühren kostet und Juli 2021 für 12c R1 endet.

Szenario 4: Man verzichtet auf ein ETL-Werkzeug. Jegliche Wartung und Weiterentwicklung wird durch PL/SQL-Programmierung vorgenommen. Die vorhandenen Mappings werden durch den ETL-Workframe lauffähig gehalten. Was Lizenzkosten betrifft, ist das die günstigste Variante. Allerdings verzichtet man langfristig auf alle Vorteile, die ein ETL-Werkzeug bietet.

Welche dieser Szenarien im jeweiligen Fall zum Einsatz kommen, ist im Einzelfall zu klären. Allerdings eröffnet der ETL-Workframe Strategien jenseits der Komplett-Migration in ein anderes ETL-Werkzeug.



Sven Bosinger
sven.bosinger@its-people.de