

Fehleranalyse und -behebung im Oracle Warehouse Builder während der Entwicklung und im Betrieb

Marco Lepper
virtual7 GmbH
Karlsruhe

Schlüsselworte

Oracle Warehouse Builder, OWB, Debugging, Fehleranalyse, Fehlerbehebung

Einleitung

Der Oracle Warehouse Builder findet als etablierte ETL-Lösung weit verbreitete Anwendung in den Datawarehouse-Landschaften deutscher Unternehmen. Bei der Umsetzung und dem Betrieb von einfachen ETL-Prozessen bis hin zu komplexen mehrstufigen Ladestrecken bleiben Probleme vielfältigster Natur selten aus. So sind bestimmte Fehler wie solche, die die Bereitstellung von Mappings oder deren Ausführung verhindern, zwar nicht zu übersehen, aber nicht immer führen die Logausgaben des OWB auf die richtige Spur zur Korrektur. Warnungen der Validierungsfunktion scheinen zuerst von geringer Bedeutung zu sein und werden gerne ignoriert, verstecken dann aber nicht selten handfeste Probleme. Und auch wenn die Umsetzung der Ladestrecken handwerklich sauber erscheint, können sich Fehler in die Logik eingeschlichen haben, was sich im schlimmsten Fall erst im Produktivbetrieb durch fehlerhafte Daten im Warehouse bemerkbar macht. Im Folgenden werden Herangehensweisen zur Analyse und Behebung der unterschiedlichen Fehlerquellen, typische Fallstricke bei der Implementierung und Best Practices zur proaktiven Fehlervermeidung vorgestellt. Aufgrund des Umfangs des Themas kann dabei nur ein Auszug behandelt werden.

Übersicht

Der Entwicklungsprozess im OWB läuft in der Regel folgendermaßen ab:

1. Erstellung der Mappings und Workflows
2. Deployment in eine Testumgebung
3. Funktionstest
4. Übernahme in die Produktivumgebung
5. Produktivbetrieb

Bei diesen Schritten können Probleme unterschiedlicher Art auftreten:

- Probleme bei der Arbeit mit dem OWB Design Center
- Objekte können nicht generiert werden
- Generierte Objekt können nicht bereitgestellt werden
- Bereitgestellte Mappings oder Workflows können nicht erfolgreich ausgeführt werden
- Die Ausführung liefert nicht das gewünschte Ergebnis
- Metadatenimport bei der Übernahme auf ein anderes System schlägt fehl

Wie lassen sich die Ursachen dieser Probleme analysieren und beheben?

Probleme bei der Arbeit mit dem OWB Design Center

Beispiel 1:

Beim Speichern von Änderungen werden diese zwar korrekt festgeschrieben, aber es werden folgende oder ähnliche Fehlermeldungen angezeigt:

- Preference.properties (Access is denied)
- Error while trying to save layout :
MappingEditorLayout.xml(Access is denied).

Ursache:

Die Berechtigungen auf dem Installationsordner des OWB-Clients erlauben keinen schreibenden Zugriff auf Konfigurationsdateien.

Lösung:

Die benötigten Berechtigungen vergeben. Wenn organisatorisch nichts dagegen spricht, kann der Einfachheit halber Schreibrechte auf den kompletten OWB-Ordner und alle darin enthaltenen Objekte vergeben werden.

Beispiel 2:

Das Speichern von Änderungen am Projekt schlägt mit einer Exception fehl.

Vorgehen:

Zuerst sollte geprüft werden, ob die Exception Hinweise auf die Ursache des Problems liefert und dieses damit behoben werden kann.

Ist dies nicht der Fall, hilft als Workaround eventuell ein Neustart des Design Centers.

Tritt das Problem wiederholt auf, sollte geprüft werden, ob der Patch-Stand auf Datenbank und/oder OWB-Client veraltet sind. Dies bezieht sich nicht nur auf die Versionsnummer (z.B. 11.2.0.3) sondern auch auf die installierten Interim-Patches. Zumindest der neueste verfügbare kumulative OWB-Patch sollte installiert werden, weiterhin sollten die nachfolgenden Einzel-Patches auf Relevanz geprüft und gegebenenfalls ebenfalls installiert werden. Zu beachten ist auch, dass auf dem Server und allen Clients zwingend dieselben OWB-Patches installiert werden müssen. Alleine durch die in den verfügbaren Patches enthaltenen zahlreichen Bugfixes lassen sich bereits viele Probleme vermeiden.

Eine weitere mögliche Fehlerquelle sind Probleme im OWB-Repository. Durch Bugs im OWB, ob bekannte oder unbekannt, könnten sich dort Inkonsistenzen eingeschlichen haben. Oracle stellt das sogenannte Health-Check-Skript bereit, mit dem sich bestimmte Fehler im Repository finden lassen. Das Skript ermöglicht dann entweder direkt eine Korrektur oder gibt Hinweise, wie sich die gefundenen Probleme beheben lassen.

Unglücklicherweise gehen bei diesem Fehlerbild alle Änderungen seit dem letzten erfolgreichen Speichern verloren. Um den potentiellen Verlust gering zu halten, empfiehlt es sich daher, Änderungen regelmäßig und in kurzen Abständen zu speichern.

Fehler beim Generieren eines Mappings

Dieses Problem tritt typischerweise auf, wenn im Mapping ein Implementierungsfehler vorliegt.

Zur Analyse sollte zuerst ein Blick in das Generierungsprotokoll geworfen werden. Je nach Ansicht wird dort aber nicht unbedingt eine Fehlermeldung angezeigt. Dies ändert sich manchmal, wenn das Mapping zur Bearbeitung geöffnet und dann aus dieser Sicht heraus die Generierung gestartet wird. Sofern es dann eine Fehlermeldung gibt, weist diese nicht zwingend klar genug auf das zugrunde liegende Problem hin. In diesem Fall sollten auch die Warnungen genauer betrachtet und geprüft werden, ob sich dahinter nicht Fehler in der Implementierung verstecken.

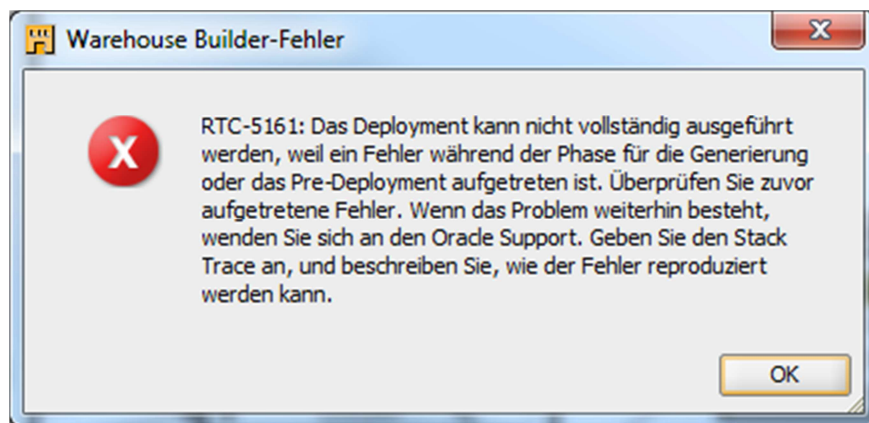
Führt dieses Vorgehen nicht zur Beseitigung des Problems, bietet es sich an, eine Kopie des Mappings zu erstellen und dieses durch die Herausnahme von Operatoren immer weiter zu vereinfachen, bis eine Generierung möglich ist. Die Problemursache ist dann in den zuletzt entfernten Operatoren zu finden.

Beispiele für Fehler, die zum Scheitern des Generierungsschritts führen können, sind fehlende Verknüpfungen zwischen Operatoren oder fehlerhafte Join-Bedingungen.

Das Mapping kann nicht bereitgestellt werden

Kann ein Mapping nicht bereitgestellt werden, kann dies mehrere Ursachen haben. Dies beginnt bei einem fehlerhaften Setup des OWB-Workspaces, in dem das Zieldatenbankschema nicht als OWB-Benutzer aufgenommen wurde. Dies stellt sicher, dass das Schema die für die Bereitstellung benötigten Berechtigungen erhält

Weitere mögliche Ursachen sind Fehler in der vorangehenden Validierung oder Generierung.



Dieser Fehler deutet daraufhin, dass die Generierung des Mappings nicht möglich ist, also schon der Schritt vor der Bereitstellung scheitert.

Auch hier empfiehlt sich als Vorgehen, das Bereitstellungsprotokoll zu prüfen und alle Fehler und/oder Warnmeldungen zu beheben und gegebenenfalls sich durch eine Vereinfachung des Mappings an die Fehlerquelle heranzutasten.

Es ist zwingend notwendig, dass der Bereitstellungsschritt ohne Fehler und ohne Warnung abgeschlossen wird. Bei Warnungen im Validierungsschritt kann ein einfach gebautes Mapping in

vielen Fällen trotzdem erfolgreich ausgeführt werden, bei Warnungen im Bereitstellungsschritt ist davon auszugehen, dass die Ausführung scheitert.

Bei Bereitstellung eines Mappings als PL/SQL-Package kommt es oft aufgrund folgender Probleme zu Warnungen:

- Verwendete Objekte wie Tabellen, Views, Functions, Procedures etc existieren in der Datenbank nicht bzw. unterscheiden sich in der Schreibweise oder sind anders aufgebaut als sie im Mapping verwendet werden
- Es fehlen Grants für den Zugriff auf Objekte eines anderen Schemas.

Ein Mapping kann nicht erfolgreich ausgeführt werden

Bei PL/SQL-Mappings liegt in diesem Fall in der Regel ein Fehler im generierten Code vor. Sofern die Fehlermeldung keinen entscheidenden Hinweis liefert, kann das PL/SQL-Package in der Datenbank geprüft werden. Lässt sich dieses nicht kompilieren, ist die Fehlersuche dort anzusetzen. Entweder kann der PL/SQL-Code direkt geprüft werden, oder die Generierungsfunktion des OWBs wird genutzt. Hierüber lassen sich auch SQL-Statements für einzelne Schritte im Mapping generieren, die dann testweise im Zielschema ausgeführt werden können. Hierüber lässt sich die Fehlerquelle dann relativ schnell lokalisieren.

Fehlerhafte Ausführungsergebnisse

Auch wenn ein Mapping ohne Fehlermeldung ausgeführt wird, ist dies kein Garant dafür, dass die Daten wie gewünscht geladen werden können. Vielleicht liegt generell noch ein Fehler in der Logik vor, vielleicht versteckt sich auch nur ein Implementierungsfehler, durch den z.B. zu viele Datensätze herausgefiltert oder aber Daten durch einen kartesischen Join unerwünscht multipliziert werden.

Hier bietet das Generierungswerkzeug mit der Möglichkeit zur Erzeugung von SQL-Code für einzelne Schritte im Mapping wieder hervorragende Möglichkeiten zur Analyse.

Probleme beim Import der OWB-Metadaten

Diesbezüglich wurden von Oracle in der Vergangenheit bereits zahlreiche Bugs behoben. Sollte ein Importvorgang mit Fehlern abbrechen, gibt es verschiedene Möglichkeiten.

Es sollte geprüft werden, ob auf Oracle Support ein Patch bereitsteht, der dieses Problem betrifft, und wenn ja, ob dieser Patch auch in die bestehende Umgebung eingespielt werden kann.

Im Zielrepository sollte ein Health-Check durchgeführt und gefundene Probleme behoben werden.

Scheitert der Import bei Objekten, die beim Import überschrieben werden sollen, hilft es teilweise, diese Objekte im Zielrepository vor dem Import zu löschen.

Sind in der Importdatei viele Objekte enthalten, hilft es teilweise, den Import in mehreren Schritten jeweils nur mit einer kleineren Anzahl von Objekten durchzuführen.

Fallstricke

Datenbankobjekte werden geändert, diese Änderung aber nicht oder nicht richtig in die Mappings übernommen. Geänderte Objekte, die im OWB Verwendung finden, müssen erneut importiert werden und alle Operatoren, die diese Objekte referenzieren, anschließend synchronisiert werden.

Unterschiedliche Patch-Stände (inklusive Interim-Patches) zwischen Server und Client kann zu Problemen führen.

Expressions werden bei der Validierung eines Mappings nicht geprüft. Auf einen grünen Haken im Validierungsschritt ist daher leider nur wenig Verlass.

Best Practices

Die Nutzung der aktuellsten Version des OWBs inklusive der passenden Interim-Patches vermeiden lästige Probleme durch Bugs, die von Oracle schon längst behoben wurden.

Bei intensiven Entwicklungsarbeiten empfiehlt sich eine regelmäßige Durchführung des Metadaten-Repository-Health-Checks. Auch auf dem aktuellsten Patch-Stand kommt es leider weiterhin regelmäßig zu Inkonsistenzen, wenn auch bei weitem nicht mehr in dem Umfang wie zu früheren Versionen.

Warnungen in der Validierung, der Generierung oder der Bereitstellung sollten niemals ignoriert werden. Auch wenn dies erheblichen Aufwand bedeuten kann, sollten Warnungen ausnahmslos beseitigt werden. Neben einer besseren Code-Qualität bringt dies auch den Vorteil, dass als Warnung versteckte schwerwiegende Probleme in der Masse der scheinbar trivialen Warnungen z.B. bezüglich fehlender explizite Datentypkonvertierungen nicht mehr so einfach übersehen werden können.

Für jeden Tabellenzieloperatoren können Fehlertabellen konfiguriert werden. In Kombination mit einer konfigurierten höheren Anzahl von tolerierten Fehlern bei Ausführung der Mappings lassen sich im Entwicklungssystem Probleme mit Daten gut analysieren, da die betroffenen Datensätze inklusive der ORA-Meldung in den Fehlertabellen abgelegt werden.

Expressions können bei der Bearbeitung einzeln validiert werden, diese Möglichkeit sollte bei komplexeren Expressions unbedingt genutzt werden.

Kontaktadresse:

Marco Lepper

virtual7 GmbH

Zeppelinstraße 2

D-76185 Karlsruhe

Telefon: +49 (0) 721-619 017 39

Fax: +49 (0) 721-619 017 29

E-Mail lepper@virtual7.de

Internet: www.virtual7.de