

Anbindung von SAP ERP als stammdatenhaltendes System

Andreas Ballenthin, OPITZ CONSULTING Deutschland GmbH

Die Nutzung von Daten aus einem SAP-ERP-System bildet eine wichtige Grundlage für unterschiedliche Datenintegrations-Szenarien. Ein wesentlicher Bestandteil ist dabei die Integration von Stammdaten in ein MDM-System. Hier stellt die Anbindung von SAP-ERP-Daten eine wesentliche Komponente dar. Dieser Artikel zeigt, wie IT-Verantwortliche den SAP-ERP-Adapter aus dem Funktionsumfang des Oracle Data Integrator im Zusammenhang mit der Integration in MDM-Systeme nutzen können.

Master Data Management (MDM), auf Deutsch „Stammdaten-Management“, ist ein Managementsystem zur Sicherstellung der Qualität der Stammdaten und verfolgt den Zweck, die Eignung der Stammdaten-Objekte bei ihrer Verwendung in allen wertschöpfenden Prozessen des Unternehmens sicherzustellen.

Um eine qualitätsgesicherte Definition herbeizuführen sowie die Pflege und Verwaltung der Stammdatenobjekte zu gewährleisten, setzt MDM alle hierzu notwendigen operativen und steuernden Prozesse auf. Auch die IT-Komponenten zur Abbildung dieser Prozesse werden zur Verfügung gestellt. MDM ist somit eine IT-lastige Geschäftstransformation (siehe Abbildung 1).

Ein MDM-System sorgt insbesondere dafür, dass bilaterale Stammdaten-Schnittstellen zwischen Unternehmenssystemen durch multilaterale Schnittstellen ersetzt werden. Dies reduziert unter anderem die Anzahl der insgesamt notwendigen Schnittstellen.

Ein MDM-System folgt wie die Core-Schicht eines Data Warehouse der Idee, einen Single Point of Truth (SpOT) herzustellen (siehe Abbildung 2). Insbesondere in Verbindung mit datenqualitätssichernden Maßnahmen arbeitet das System gemäß dem Ansatz des Konsolidierungsknotens. Die hier verwendeten Datenintegrations-Strategien und -Werkzeuge sind typischerweise ETL-Werkzeuge sowie klassische ETL- und DQ-Prozesse. In der Regel werden bei diesem Architektur-Ansatz keine Anstrengungen unternommen, die Datenqualität

in den unterlagerten operativen Systemen im Rahmen des Datenqualitäts-Managements zu verbessern.

Anbindung des MDM-Systems mit dem ODI an SAP ERP

Unternehmen in Deutschland hinterlegen viele Stammdaten, die unternehmensweit interessant sind, in SAP-ERP-Systemen. Deshalb ist es notwendig, diese effizient an das MDM-System anzubinden (siehe Abbildung 3).

Innerhalb des Oracle Data Integrator (ODI) kann die Anbindung von SAP ERP über den SAP-ABAP-ERP-Adapter erfolgen. Der aus mehreren Knowledge-Modulen bestehende Adapter ermöglicht diese Anbindung unter der Nutzung von SAP JCo Libraries und beinhaltet die folgenden Funktionen:

- Lesen von SAP-Daten aus einem SAP-ERP-System
- Laden dieser Daten in eine Oracle-Umgebung
- Reverse Engineering der SAP-Metadaten

Der Adapter wird dem ODI über eine Java-EE-Applikation als Open Tool bekannt gemacht. Er besteht aus zwei Knowledge-Modulen:

• RKM SAP ERP

Die Hauptaufgabe dieses Moduls besteht darin, im ODI-Datenspeicher ein maßgeschneidertes Reverse Engineering der SAP-Tabellen durchzuführen. Dieses schafft eine Verbindung zum SAP-System, das die Metadaten der SAP-Tabellen

vorhält, und bewerkstelligt deren Transformation in das ODI-Repository.

• LKM SAP ERP to Oracle (SQLLDR)

Dieses Modul verbindet sich zum SAP-System und holt die Daten aus den im SAP-Modell definierten Tabellen ab. Es wird in den einzelnen ODI-Interfaces verwendet und dient dem Laden der Quelldaten in die Oracle-Umgebung.

Durch die nahtlose Eingliederung in die SAP-Transportprozesse erfüllt der SAP-ABAP-ERP-Adapter die rechtlichen Rahmenbedingungen und auch die Erfordernisse der Revisionsicherheit.

Installation und Konfiguration

Vor der Installation des SAP-ABAP-ERP-Adapters sollten mit der Installation betraute Mitarbeiter einige wichtige Vorbedingungen und administrative Voraussetzungen beachten:

- Die Anleitung „Getting Started with SAP ABAP ERP Adapter“ lesen
- Für alle Betriebssysteme, in denen ODI-Agenten laufen, den passende SAP-Java-Connector (SAP JCo) installieren
- Agenten-lokal einen Oracle-Client installieren, um den SQL-Loader auszuführen
- Auf dem Client im Verzeichnis „\$ODI-HOME\oracledi\client\odi\bin\odi.conf“ die Referenzen auf „odi-sap.jar“ und „sapjco3.jar“ aufnehmen, um das GUI-basierte Reverse Engineering funktionsfähig zu bekommen (siehe Listing 1)

Wir liefern das Know-how für Ihren Erfolg

PROJEKTE

Keine „fixen Ideen“, sondern Projekte mit klarer Ziel- und Ablaufplanung.

PROFILE

Logistikmanager berichten. Direkt, authentisch und unverschnörkelt.

PROZESSE

Der Blick auf die Supply Chain – von der Beschaffung über Produktion und Distribution bis zur Entsorgung.

PRODUKTE

Neuheiten für die Praxis. Was sagen Anwender über Produkte und logistische Lösungen?

PROGNOSEN

Die Rubrik für Theorie und Strategie, neue Ideen, Trends und Märkte.

EXTRA

Die Spezialrubrik für den vertiefenden Blick auf besondere Themen.

NACHGEFRAGT

Die Rückschau auf Entwicklungen und Erfahrungen, Erfolge und Enttäuschungen interessanter Projekte.



ONLINE-ANGEBOT

Unter logistik-heute.de finden Abonnenten u.a. tagesaktuelle News, Marktübersichten, Studien, ein Karriere-Portal und haben **kostenfrei**en Zugriff auf die gesamte Artikeldatenbank.

Noch mehr Logistik-IT?



Jetzt kostenlos
Heft bestellen!

www.logistik-heute.de
Tel. +49 (0)89/32391-314



**LOGISTIK
HEUTE**

Neben diesen technischen Voraussetzungen ist es ratsam, den SAP-Basis-Administrator mit in die Pläne und Vorgänge einzubeziehen, denn eine reibungslose Kooperation mit ihm ist für ein erfolgreiches Projekt von unschätzbarem Nutzen. Je einfacher SAP-Transportprozesse später initiiert werden können, desto reibungsloser funktioniert ein Deployment von der SAP-Entwicklungsumgebung in weitere SAP-Instanzen.

Für die Client-Installation empfiehlt der Autor die Variante mit einer 32-Bit-Windows-Virtual-Machine mit installiertem ODI- und Oracle-Client. Die Installation auf einem Windows 2008 Server (64-Bit) ist mit einigen Schwierigkeiten und Anpassungen verbunden, da die ODI-GUI ein 32-Bit-Java-Development-Kit (JDK) benötigt.

Die Server-Installation gestaltet sich recht einfach, wenn der SAP-ERP-Connector auf einem nicht lokalen Agenten laufen soll und kein Reverse Engineering geplant ist; das Open Tool benötigt die ODI-GUI und somit JDK-32-Bit. Die Konfigurationsbeschreibung basiert auf einem Praxisfall, bei dem das Projekt-Team mit einem Linux-User arbeitete, der nur über „Others“-Rechte für die ODI-Verzeichnisstruktur verfügte. Zusätzlich kam ein separater Agent namens „ODIWR3“ zum Einsatz. Für die Server-Installation nahm das Team einige Anpassungen vor:

- Erweiterung des Personal Initialization File „\$HOME/.bash_profile“ um folgende Einträge (siehe Listing 2)
- Setzen von „ORACLE_HOME“ für die Ausführung des SQL-Loader
- Ablage der notwendigen JARs sowie der JCO (unter Windows als Dynamic Link Library (.dll), unter Linux/Unix als Shared Object (.so) in der „LD_LIBRARY_PATH“)

Darüber hinaus benötigt der SAP-Connector einige weiterführende Topologie-Definitionen. Die Definition des anzubindenden SAP-ERP-Systems erfolgte über verschiedene Parameter. Die IP-Adresse des SAP-ERP-Systems beziehungsweise der Hostname entsprechen dem technischen Namen des Host

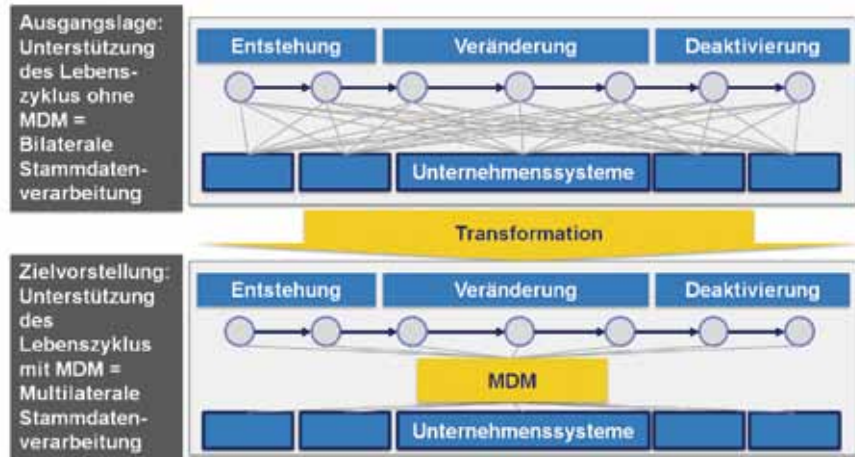


Abbildung 1: MDM ist eine geschäftslastige IT-Transformation

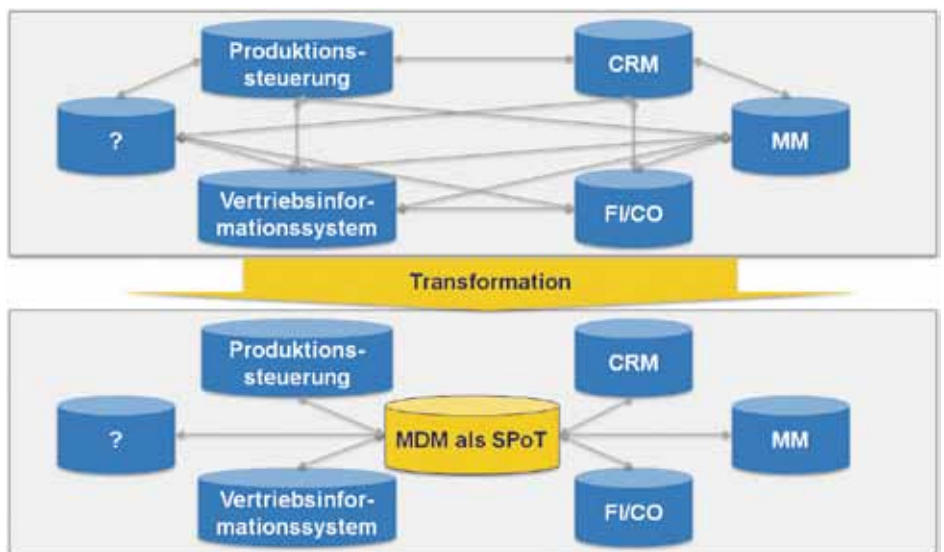


Abbildung 2: MDM-System als SPoT ist eine geschäftslastige IT-Transformation

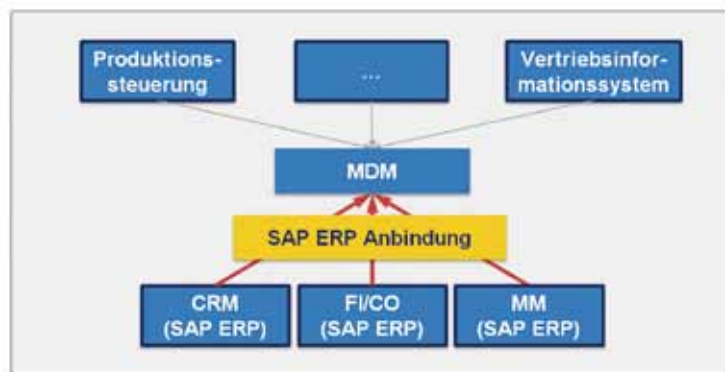


Abbildung 3: Unternehmenssysteme, insbesondere SAP ERP

```
AddJavaLibFile ../../jdev/extensions/oracle.odi.navigators/lib/odi-sap.jar
AddJavaLibFile ../../jdev/extensions/oracle.odi.navigators/lib/sapjco3.jar
```

Listing 1

auf dem SAP-System. „User“ ist der eindeutige Username, mit dem sich der Anwender nach Eingabe des Kennworts (case sensitive) auf dem SAP-System anmelden kann.

Im Abschnitt „Flexfields“ müssen weitere systemspezifische Einstellungen vorgenommen werden:

- Die „SAP System Number“ ist eine aus zwei Ziffern bestehende Nummer, die einer bestimmten SAP-Instanz zugewiesen ist, auch Web Application Server (WAS).
- Die „SAP Client Number“ ist eine aus drei Ziffern bestehende Nummer, die einer abgeschlossenen Einheit, dem sogenannten „Client“ in SAP, zugewiesen ist. Ein Client wiederum kann ein Trainings-, Test-, Entwicklungs- oder Produktions-Client sein beziehungsweise verschiedene Bereiche innerhalb einer größeren Unternehmensorganisation beschreiben.
- Die „SAP System ID“ ist eine dreistellige, eindeutige Kennung eines SAP-Systems innerhalb einer Systemlandschaft.
- Der „SAP Router String“ und die „SAP SNC Connection Properties“ sind optionale Angaben zur Erhöhung der Sicherheit durch SAP-Router und SNC, sofern diese Sicherheiten installiert sind.
- Die „SAP Language“ beschreibt die Sprache, die beim Login verwendet wird – „en“ für „Englisch“ und „de“ für „Deutsch“.
- Die Felder für die „SAP Version“ beschreiben die jeweiligen Versionen der SAP-Applikationen (ERP, ABAP, BW).
- Das „SAP Character Set“ ist nur dann notwendig, wenn das SAP-System keinen Unicode verwendet. Bei Unicode-Systemen lautet die Angabe „UTF8“.

Zudem muss die Konfiguration des SAP-FTP-Servers erfolgen. Auf diesem FTP-Server legen die Funktionsbausteine, die der SAP-ERP-Connector erzeugt hat, ihre Daten-Files ab; in einem weiteren Schritt werden die Dateien von dort abgeholt. Im Anschluss an diese Konfigurationseinstellungen kann der SAP-Connector seine Arbeit aufnehmen.

Reverse Engineering mit dem SAP-ABAP-ERP-Adapter

Im Allgemeinen legt ein ETL-Entwickler vor dem Reverse Engineering ein „Data Model“

an. Dieses muss auf der SAP-ABAP-Technologie sowie auf einem im Vorfeld zu definierenden logischen Schema basieren.

Das Reverse Engineering nutzt das Reverse-Engineering-Knowledge-Modul „RKM SAP ERP“ und bietet einen direkten Import von Tabellen aus dem SAP-Directory. Dazu muss das RKM im Reiter „Reverse“ des „SAP ERP Data Model“ ausgewählt werden. Je nach Vorliebe kann sich das Development-Team für die Nutzung der GUI („SAP Metadata Browser“) entscheiden oder die Tabellen direkt im Knowledge-Modul parametrieren. Die Nutzung der GUI wird über die Option „USE_GUI_KM“ („true“ oder „false“) gesteuert. Der Einsatz der GUI empfiehlt sich allerdings nur für den Fall, dass man genau weiß, wo sich die gesuchte Tabelle in der SAP-Hierarchie wiederfindet.

In der Regel ist es praktikabler, direkt den Namen der SAP-Tabelle anzusprechen und die GUI nicht zu nutzen. Nach dem Import baut SAP seine eigene Tabellen-Hierarchie im ODI auf, einschließlich aller Kommentare innerhalb des Dictionary.

Extraktion von Daten aus SAP ERP

Abbildung 4 zeigt einen schematischen Überblick über die Funktionsweise des Load-Knowledge-Moduls „LKM SAP ERP to Oracle“ (SQLLDR). Es wird durch eine Kombination von Aufrufen aus Java BeanShell, Jython, dem ODI und Oracle ausgeführt. Diese Schritte verdeutlichen die Funktionsweise innerhalb der Abfolge des Knowledge-Moduls:

- In Schritt 180 lädt der SAP-ERP-Connector ABAP-Code in das SAP-ERP-System. Dieser wurde in früheren Schritten des Knowledge-Moduls generiert. Voraussetzung für das Laden des Codes ist, dass die Option „UPLOAD_ABAP_CODE“ gesetzt ist. Sofern die Option „EXECUTE_ABAP_CODE“ aktiviert wurde, führt der Connector den Code aus. Dieser generiert ein durch Zeichen getrenntes Text-File, wobei das Trennzeichen im Rahmen der Option „FIELD_SEPARATOR“ innerhalb des Interface bestimmt wird. Dieses Textfile wird zum FTP-Server transferiert, der als „SAPFTP“ im SAP-System festgelegt ist.
- Schritt 190 veranlasst den ODI-Agenten, das generierte Textfile vom SAPF-

TP-Server unter Berücksichtigung der definierten Parameter in das temporäre Verzeichnis „System.getProperty(„java.io.tmpdir““ zu transferieren. Somit wird gewährleistet, dass das Daten-File physisch im lokalen File-System des ODI-Agenten gespeichert ist.

- Schritt 200 erzeugt ein „Oracle SQL Loader Control“-File im temporären Verzeichnis des Daten-Files.
- In Schritt 210 wird der Oracle-Loader unter Berücksichtigung des zuvor erstellten Files aufgerufen, um die Daten in die C\$-Tabelle des Physical-Work-Schemas auf dem Ziel-Data-Server zu laden.

Für jedes Source-Set im ODI-Interface kann der ETL-Entwickler über die Option „UPLOAD_ABAP_CODE“ entscheiden, ob er einen neuen ABAP-Code generieren und in den SAP-Application-Server hochladen oder lediglich einen bereits hochgeladenen ABAP-Code ausführen möchte (siehe Abbildung 5).

Den ABAP-Code kann er ausschließlich in ein SAP-Entwicklungssystem hochladen. Um ihn in einem Test- oder Produktiv-System auszuführen, muss ihn der SAP-Basis-Administrator zunächst über das SAP-Transportsystem „deployen“. In diesem Fall kommt das im Vorfeld angesprochene gute Kooperationsverhältnis zum SAP-Basis-Administrator zum Tragen. Aufgrund der schnelleren Verifizierung ist es bei eventuell auftretenden Syntax-Fehlern von Vorteil, den ABAP-Code zunächst direkt im SAP-System zu korrigieren. Erst im Anschluss sollte der ETL-Entwickler die Behebung des Fehlers im ODI-Interface vornehmen. Typischerweise verwendet man in einem ODI-Interface nur ein Source Set, im weniger häufigen Fall können mehrere Source-Sets über eine Mengenoperation miteinander verbunden werden.

Praktische Erfahrungen aus dem Betrieb

In der Praxis ließen sich aus der Sicht eines SQL-Entwicklers bislang die folgenden Eigenheiten beobachten:

- SAP behandelt Strings und numerische Werte in Filtern identisch. Dazu ein Beispiel: „BSEG.BEWAR“ ist ein Character-Feld. Eine Filterbedingung „BSEG.

BEWAR = «#VAR_BEWEGUNGSART_1» funktioniert nicht, „BSEG.BEWAR = #VAR_BEWEGUNGSART_1“ hingegen schon.

- „IN-Clauses“ in Filtern funktionieren nicht. Hier müssen mehrere „OR“-Bedingungen verwendet werden.
- Filter und Joins müssen in Großbuchstaben (Upper Case) formuliert sein, „or“ funktioniert im Gegensatz zu „OR“ nicht.
- Fehlende Leerzeichen in Filtern/Joins führen zum Fehler: „BSEG.GJAHR=2011“ funktioniert nicht, „BSEG.GJAHR = 2011“ funktioniert.
- Das Filtern mehrerer Tabellen über die gleiche Variable funktioniert nicht. Hier braucht man pro Tabelle eine Variable. Will man den ODI-Join von „BSEG“ und „BKPF“ auf das Geschäftsjahr filtern, benötigt man zwei Variablen: „BSEG.GJAHR = #VAR_GESCHAEFTSJAH“ und „BKPF.GJAHR = #VAR_GESCHAEFTSJAH_BKPF“. Hinweis: Der in ODI formulierte Join wird in ABAP als Loop mit Unterabfragen abgebildet, beide Filter sind somit nötig und nicht redundant.

Zu Beginn musste der Code des Knowledge-Moduls in Schritt 210 angepasst werden, um die Möglichkeit zu gewährleisten, Sonderzeichen in Passwörtern sauber zu übergeben (siehe Listing 3, Änderungen rot markiert).

Die Ergebnisse des Load-Knowledge-Moduls (der befüllten C\$-Tabellen) können mit einem beliebigen Integration-Knowledge-Modul (IKM) weiterverarbeitet werden, beispielsweise mit IKM SQL Control Append (Truncate/Insert) und IKM Oracle Incremental Update (Merge).

Fazit

Bei einer zentralisierten Architektur im Zusammenspiel mit einer SAP-ERP-Anbindung hat sich der Oracle Data Integrator für das Master Data Management in der Praxis als zuverlässiges und wertvolles Werkzeug bewährt. Als Hilfsmittel kann der Autor den ODI in diesem Zusammenhang nur empfehlen.

Andreas Ballenthin

andreas.ballenthin@opitz-consulting.com

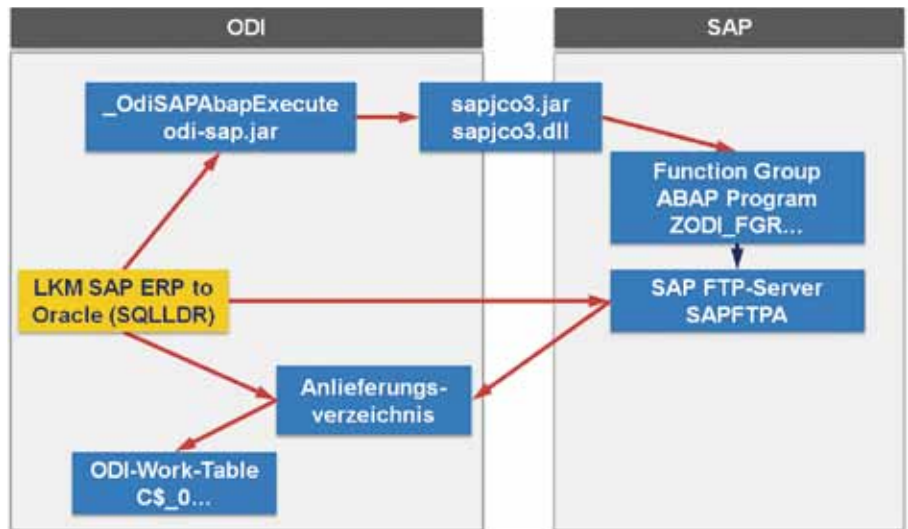


Abbildung 4: Funktionsweise von LKM SAP ERP to Oracle (SQLDR)

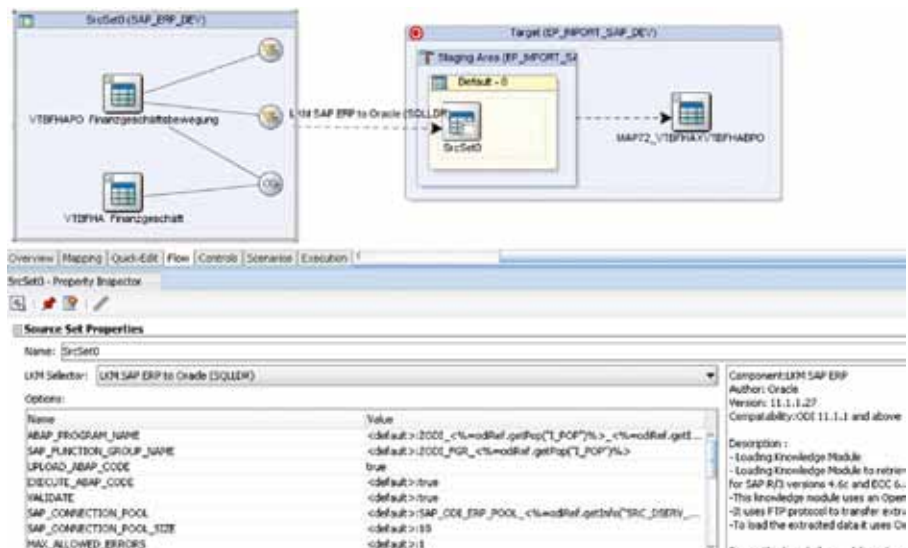


Abbildung 5: Beispielkonfiguration des LKM SAP ERP to Oracle (SQLDR)

```

PATH=$PATH:$HOME/bin:/app/oracle/OracleHomes/db11g/bin
ORACLE_HOME=/app/oracle/OracleHomes/db11g
CLASSPATH=$CLASSPATH:/opt/odi/Oracle/Middleware/Oracle_ODI1/oracledi/drivers
LD_LIBRARY_PATH=$LD_LIBRARY_PATH:/opt/odi/Oracle/Middleware/Oracle_ODI1/oracledi/agent/drivers
export ORACLE_HOME PATH CLASSPATH LD_LIBRARY_PATH

```

Listing 2

```

loadcmd = r"sqlldr ' <%=odiRef.getInfo("DEST_USER_NAME")%> / <%=odiRef.getInfo("DEST_PASS")%> @ <%=odiRef.getInfo("DEST_DSERV_NAME")%> ' control="%s" log="%s" > "%s" "" % (ctlfile, logfile, outfile)

```

Listing 3