Tech Note: Oracle BAM – ODI Integration

Supported BAM Version: Oracle BAM 12.1.3

Supported ODI Version: ODI 11g or 12c

Objectives:

- This document explains integration form ODI to BAM.
- We are using Oracle as Source Technology and JMS Topic XML as target technology from ODI
- This document briefly explains the steps to integrate ODI with BAM using some sample tables.

Note:

We have used following versions to create above examples.

Oracle Data Integrator 11g (11.1.1) Build ODI_11.1.1.6.0_GENERIC_111219.1055

If you are using a later version of ODI, “Interface” is replaced with “Mapping”, create “Mapping” accordingly.

1. Table of Content

1. Table of Content 1
2. Abstract 3
3. Creates Target Data Store for JMS 5
3.1. Create Physical Architecture
3.2. Create Logical Architecture
3.3. Create Model
4. Create Source Data Store for Oracle 16
4.1. Create Physical Architecture
4.2. Create Logical Architecture
4.3. Create Model
5. Create Project 23
5.1. Create New Project
5.2. Import Knowledge Modules
5.3. Single Row per Message
5.4. Batch Rows per Message
6. CDC 46
6.1. Customize IKM
6.2. Modify Source Data Store for CDC
6.3. Creating a Package to Set Up CDC
6.4. Creating the Integration Flows to Consume the Changes
6.5. Construct Package For CDC with Interfaces
7. BEAM Configuration 63
7.1. Create Data Objects
7.2. Configure EMS in BEAM

2. Abstract

Integration form ODI to BEAM:

We are using Oracle as Source Technology and JMS Topic XML as target technology from ODI.

We have sample tables DEPARTMENT and EMPLOYEE tables from source data store Oracle DB.

Create sample tables:

```
Create Table DEPARTMENT(deptid varchar2(30),dname varchar2(30),PRIMARY KEY (deptid));

CREATE TABLE Employee(id varchar2(30),name varchar2(30),address varchar2(30),deptid varchar2(30), PRIMARY KEY (id),
Foreign Key (deptid) references DEPARTMENT(deptid) );
```

XML schema should be prepared corresponding to above tables which represent JMS message. Table1 represents DEPARTMENT and Table2 represents EMPLOYEE.

ODI_DEMO.xsd :

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<xs:schema version="1.0" targetNamespace="http://xmlns.oracle.com/bam" xmlns:tns="http://xmlns.oracle.com/bam"
xmlns:xs="http://www.w3.org/2001/XMLSchema">
 <xs:element name='Root'>
<xs:complexType>
      <xs:choice minOccurs="0" maxOccurs="unbounded">
              <xs:element name="Table1" type="tns:Table1" minOccurs="0" maxOccurs="unbounded" />
                                    <xs:element name="Table2" type="tns:Table2" minOccurs="0" maxOccurs="unbounded" />
      </xs:choice>
    </xs:complexType>
</xs:element>

<xs:complexType  name='Table1'>
  <xs:sequence>
                         <xs:element name="ID" type="xs:string" minOccurs="0" maxOccurs="1"/>
                         <xs:element name="DNAME" type="xs:string" minOccurs="0" maxOccurs="1"/>
  </xs:sequence>
  <xs:attributeGroup ref="attributeGroup"/>
</xs:complexType>

<xs:complexType  name='Table2'>
  <xs:sequence>
                         <xs:element name="ID" type="xs:string" minOccurs="0" maxOccurs="1"/>
                         <xs:element name="NAME" type="xs:string" minOccurs="0" maxOccurs="1"/>
                         <xs:element name="ADDRESS" type="xs:string" minOccurs="0" maxOccurs="1"/>
                         <xs:element name="DEPTID" type="xs:string" minOccurs="0" maxOccurs="1"/>
  </xs:sequence>
  <xs:attributeGroup ref="attributeGroup"/>
```

```
</xs:complexType>

<xs:simpleType name="OperationNameType">
  <xs:restriction base="xs:string">
            <xs:enumeration value="INSERT"/>
            <xs:enumeration value="DELETE"/>
            <xs:enumeration value="UPDATE"/>
            <xs:enumeration value="UPSERT"/>
  </xs:restriction>
</xs:simpleType>

<xs:attributeGroup name="attributeGroup">
    <xs:attribute name="operationType" type="tns:OperationNameType" use="required" />
            <xs:attribute name="dataObjectName" type="xs:string" use="required" />
            <xs:attribute name="keys" type="xs:string" />
</xs:attributeGroup>

 </xs:schema>
```

We will be integrating DEPARTMENT and EMPLOYEE from Source Data Store to BAM using JMS TOPIC XML technology through ODI.

Prerequisites:

1. XML Schema definition should be prepared for corresponding target tables like ODI_DEMO.xsd.
2. Create topic (eg: jms/odiTopic) and connection factory (eg: jms/odiFactory) on beam server.
3. ODI repository should be created.

Connect to ODI repository by providing user name and password

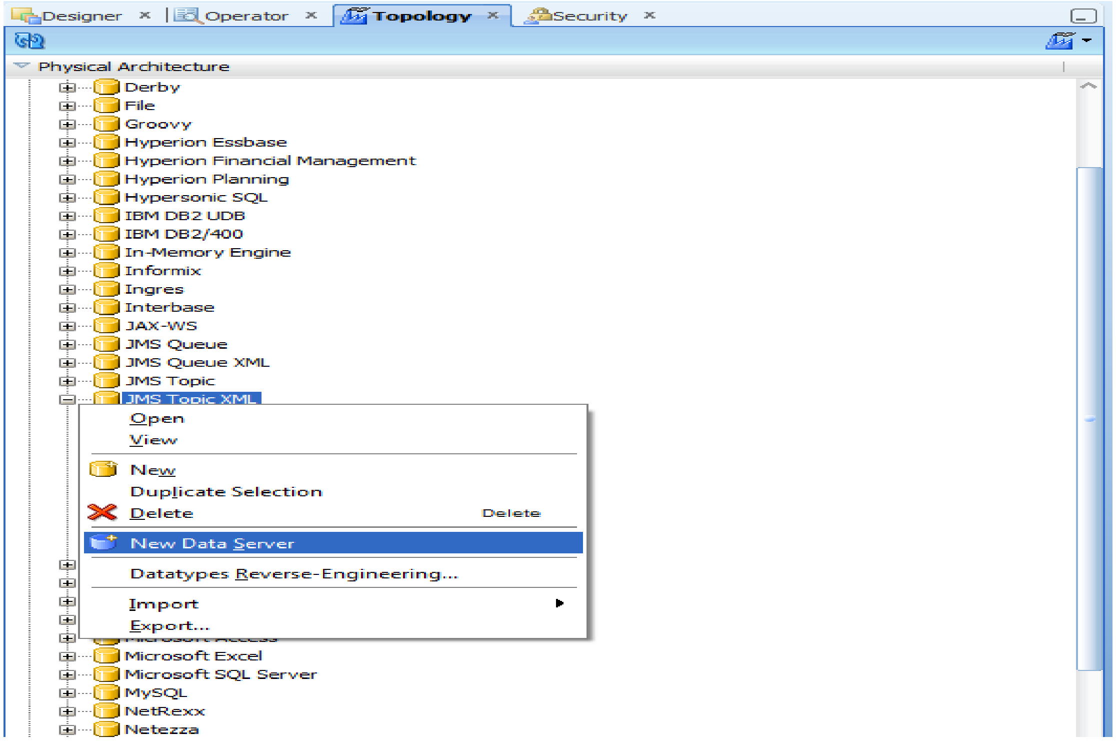
3. Creates Target Data Store for JMS

3.1 Create Physical Architecture:

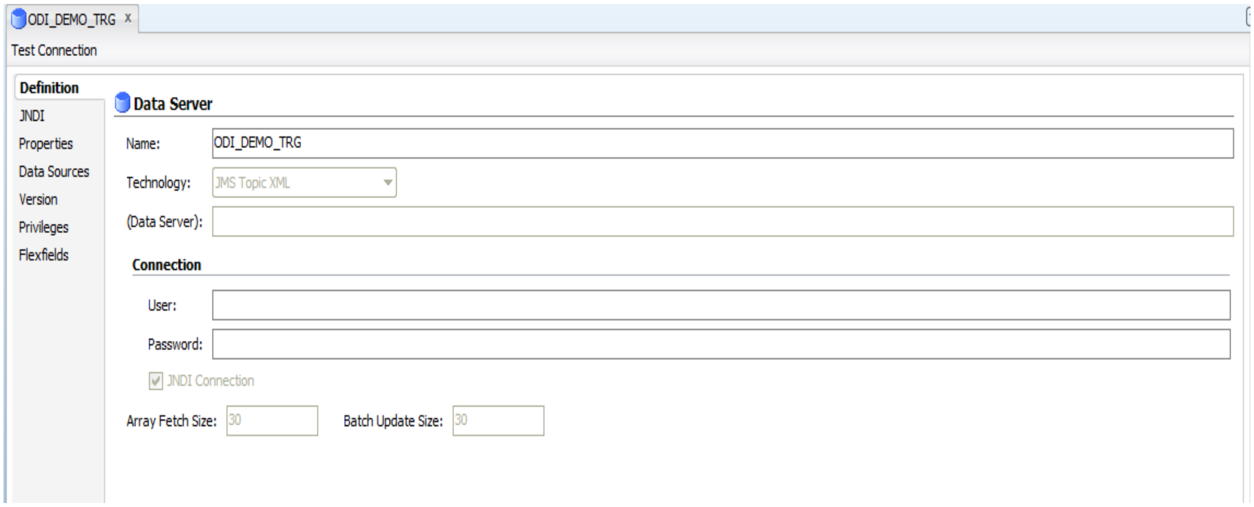
The physical architecture defines the different elements of the information system. A *technology* handles formatted data. Therefore, each technology is associated with one or more data types that allow Oracle Data Integrator to generate data handling scripts.

The physical components that store and expose structured data are defined as *data servers*. A data server is always linked to a single technology. A data server stores information according to a specific technical logic which is declared into *physical schemas* attached to this data server. Every database server, JMS message file, group of flat files, and so forth that is used in Oracle Data Integrator must be declared as a data server. Every schema, database, JMS Topic, etc., used in Oracle Data Integrator, must be declared as a physical schema.

Go to Topology →Physical Architecture → Technologies→JMS Topic XML then right click and select New Data Server.



1. Provide name for Data Server



Name: ODI_DEMO_TRG

Technology: JMS Topic XML

(Data Server):

User:

Password:

Array Fetch Size: 30

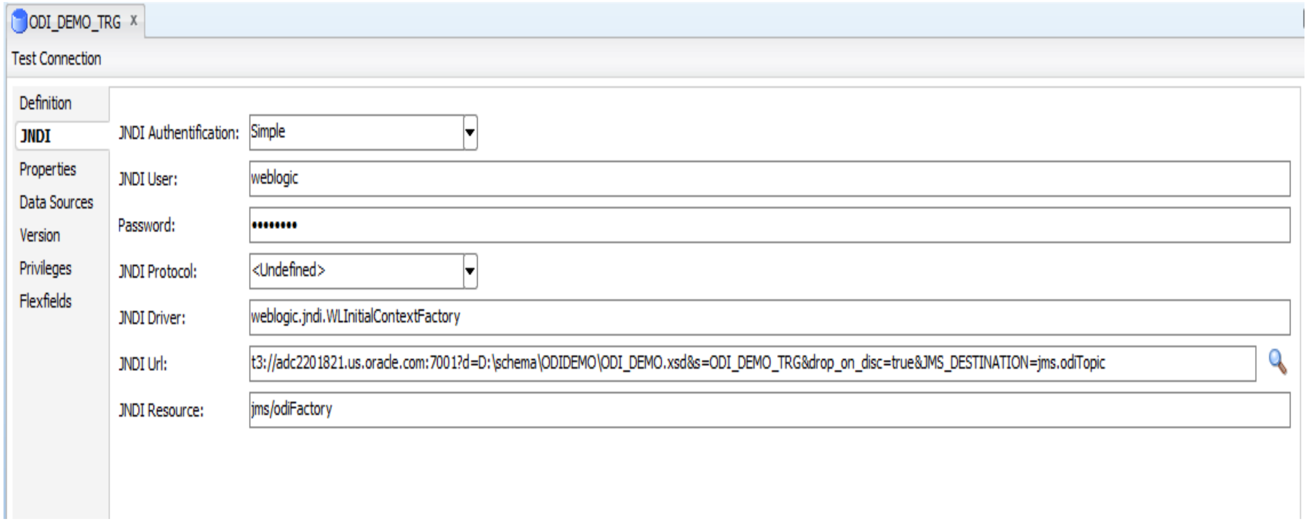
Batch Update Size: 30

2. Select JNDI tab

   a. JNDI Authentication: From the list, select the authentication mode.
   b. JNDI User: Enter the username to connect to the JNDI directory (not mandatory).
   c. Password: This user's password (not mandatory).
   d. JNDI Protocol: From the list, select the JNDI protocol (not mandatory).
   e. JNDI Driver: Name of the initial context factory java class to connect to the JNDI provider
   f. JNDI URL: `<JMS_RESOURCE>?d=<DTD_FILE>&s=<SCHEMA>&JMS_DESTINATION=<JMS_DESTINATION_NAME>`.
   g. JNDI Resource: Logical name of the JNDI resource corresponding to your JMS Queue (or Topic) connection factory.

Parameter	Value	Notes
d	<DTD File location>	DTD File location (relative or absolute) in UNC format. Use slash "/" in the path name and not backslash "\" in the file path. This parameter is mandatory.
re	<Root element>	Name of the element to take as the root table of the schema. This value is case sensitive. This parameter can be used for reverse-engineering a specific message definition from a WSDL file, or when several possible root elements exist in a XSD file.
ro	true | false	If true, the XML file is opened in read only mode.
s	<schema name>	Name of the relational schema where the XML file will be loaded.This value must match the one set for the physical schema attached to this data server. This parameter is mandatory.
cs	true | false	Load the XML file in case sensitive or insensitive mode. For case insensitive mode, all element names in the DTD file should be distinct (Ex: Abc and abc in the same file are banned). The case sensitive parameter is a permanent parameter for the schema. It CANNOT be changed after schema creation. Please note that when opening the XML file in insensitive mode, case will be preserved for the XML file.
JMSXML_ROWSEPARATOR	5B23245D	Hexadecimal code of the string used as a line separator (line break) for different XML contents. Default value is 5B23245D which corresponds to the string [#$].
JMS_DESTINATION	JNDI Queue name or Topic name	JNDI Name of the JMS Queue or Topic. This parameter is mandatory.
tna	boolean (true|false	Transform Non Ascii. Set to false to keep non-ascii characters. Default is true. This parameter is not mandatory.

JNDI URL Properties Table:



JNDI Authentification: Simple

JNDI User: weblogic

Password: ••••••••

JNDI Protocol: <Undefined>

JNDI Driver: weblogic.jndi.WLInitialContextFactory

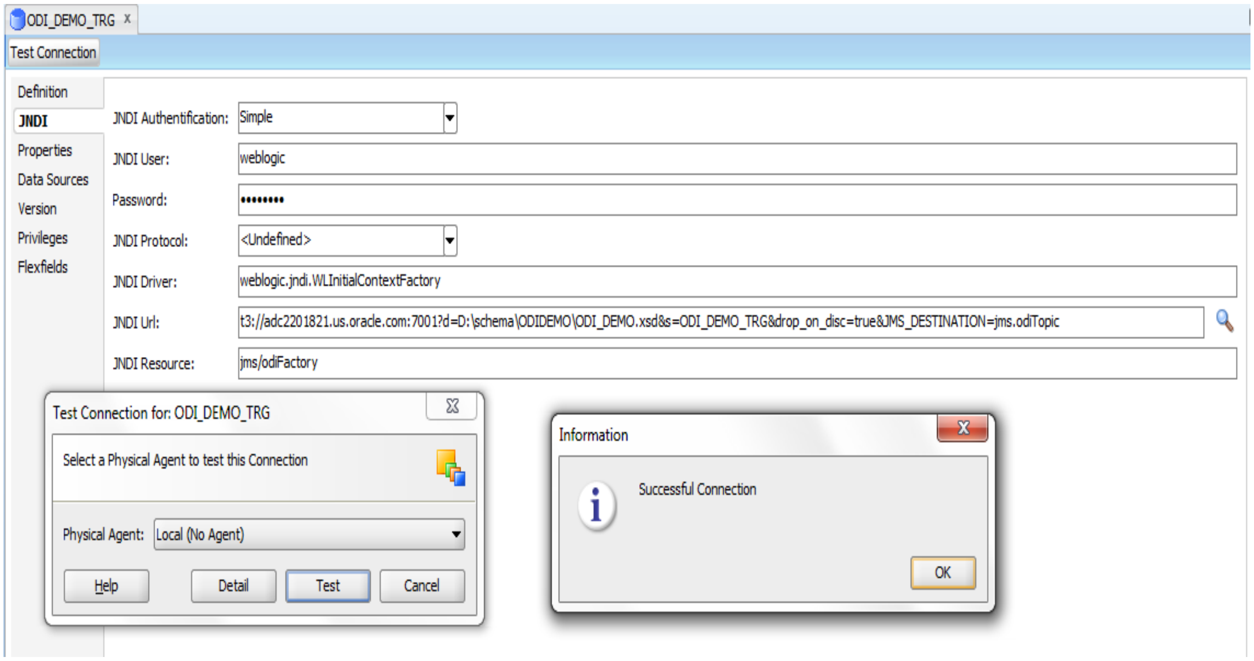
JNDI Url: t3://adc2201821.us.oracle.com:7001?d=D:\schema\ODIDEMO\ODI_DEMO.xsd&s=ODI_DEMO_TRG&drop_on_disc=true&JMS_DESTINATION=jms.odiTopic

JNDI Resource: jms/odiFactory

Following XSD represents target tables /JMS message.

a. JNDI Authentication: Simple.
b. JNDI User: weblogic
c. Password: weblogic
d. JNDI Protocol: not mandatory.
e. JNDI Driver: weblogic.jndi.WLInitialContextFactoryprovider
f. JNDI URL: t3://adc2201821.us.oracle.com:7001?d=D:\schema\ODIDEMO\ODI_DEMO.xsd&s=ODI_DEMO_TRG&drop_on_disc=true&JMS_DESTINATION=jms.odiTopic.
g. JNDI Resource: Logical name of the JNDI resource corresponding to your JMS Queue (or Topic) connection factory: jms/odiFactory

3. Test Connection



JNDI Authentification: Simple
JNDI User: weblogic
Password: ••••••••
JNDI Protocol: <Undefined>
JNDI Driver: weblogic.jndi.WLInitialContextFactory
JNDI Url: t3://adc2201821.us.oracle.com:7001?d=D:\schema\ODIDEMO\ODI_DEMO.xsd&s=ODI_DEMO_TRG&drop_on_disc=true&JMS_DESTINATION=jms.odiTopic
JNDI Resource: jms/odiFactory

Test Connection for: ODI_DEMO_TRG

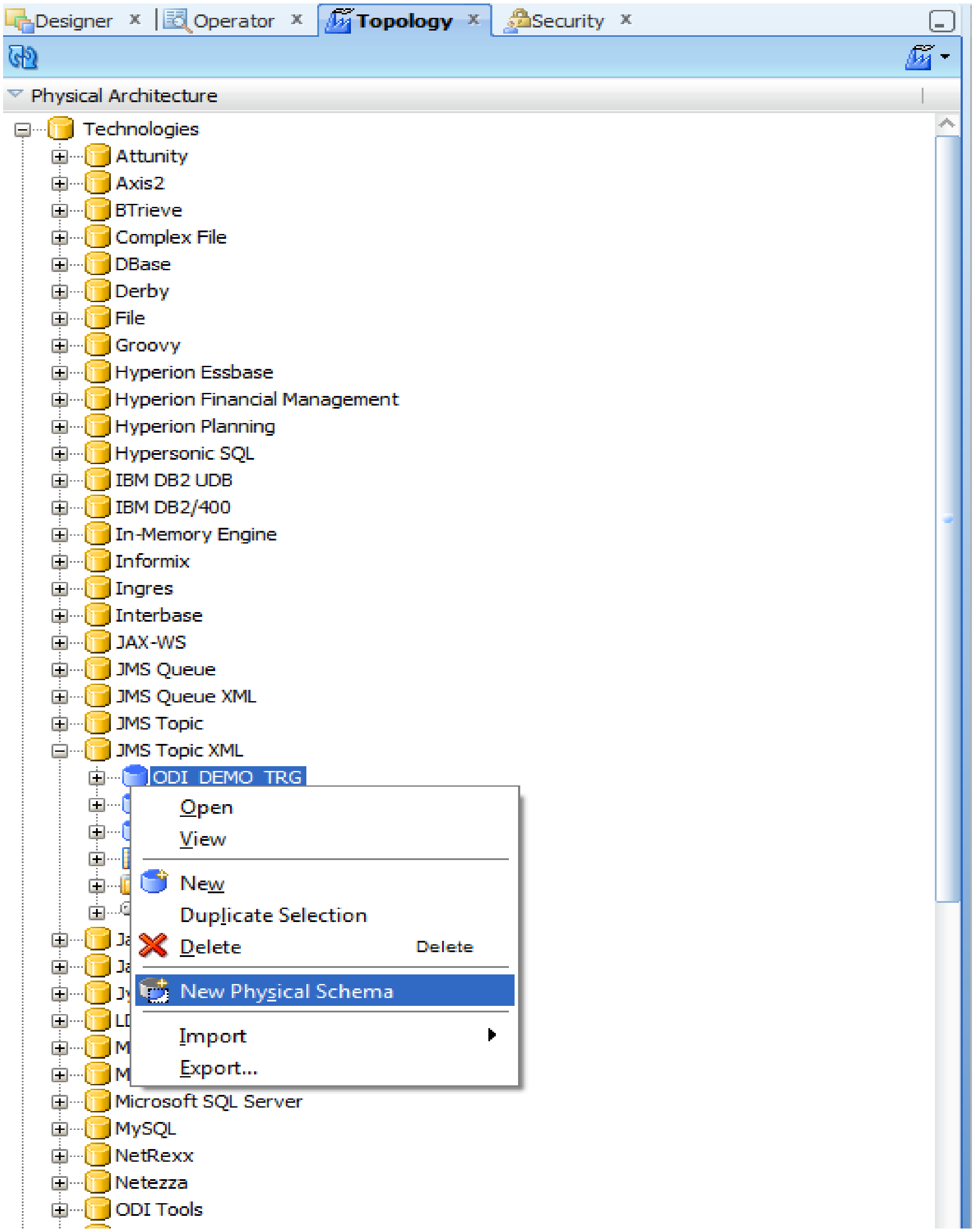
Select a Physical Agent to test this Connection

Physical Agent: Local (No Agent)

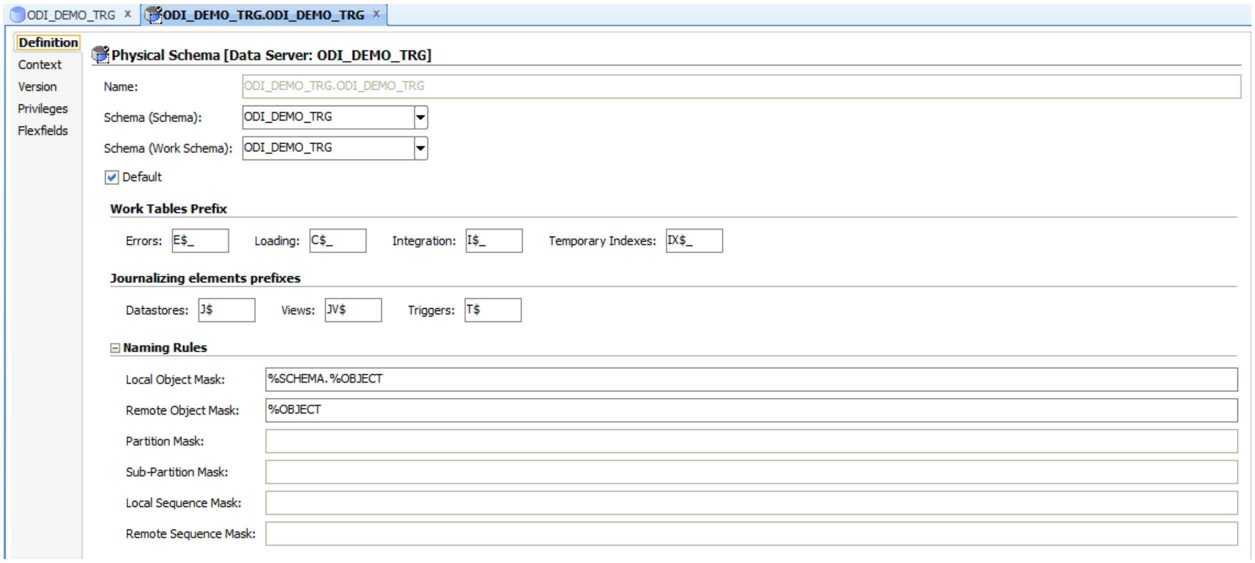
Information

Successful Connection

4. Create New Physical Schema
    1. Select New Physical Schema



1. Select Schema which has given in the JNDI URL i.e ODI_DEMO_TRG



Physical Schema [Data Server: ODI_DEMO_TRG]

Name: ODI_DEMO_TRG.ODI_DEMO_TRG

Schema (Schema): ODI_DEMO_TRG

Schema (Work Schema): ODI_DEMO_TRG

☑ Default

Work Tables Prefix

Errors: E$_ Loading: C$_ Integration: I$_ Temporary Indexes: IX$_

Journalizing elements prefixes

Datastores: J$ Views: JV$ Triggers: T$

Naming Rules

Local Object Mask: %SCHEMA.%OBJECT

Remote Object Mask: %OBJECT

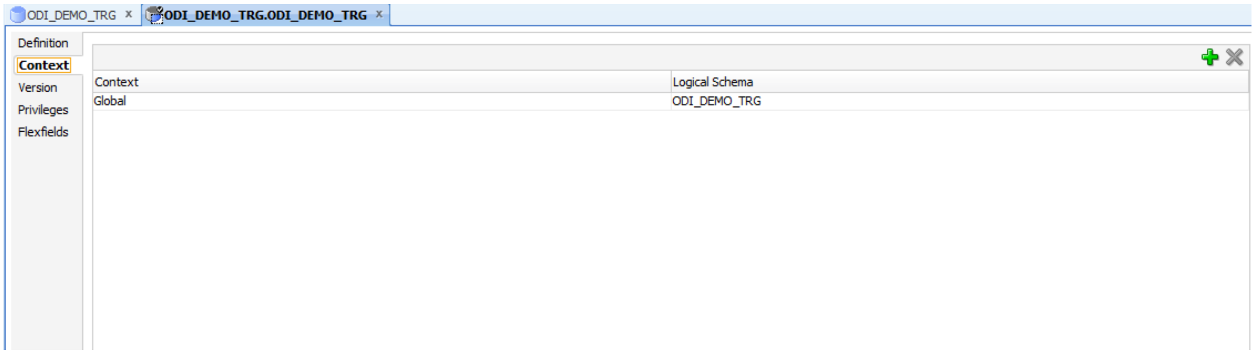
Partition Mask:

Sub-Partition Mask:

Local Sequence Mask:

Remote Sequence Mask:

2. Select Context Tab and Select Logical Schema.
   Note: Create Logical Architecture with undefined physical schema then link physical to logical here.



Context	Logical Schema
Global	ODI_DEMO_TRG

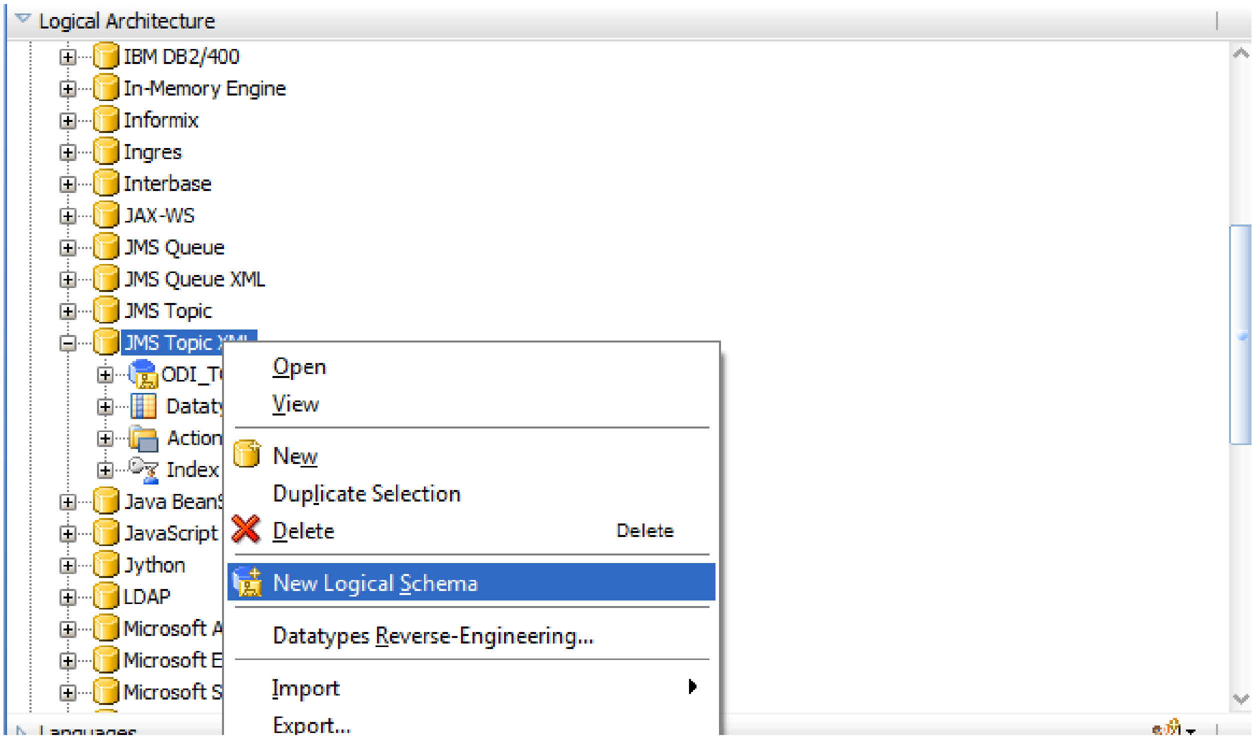
3.2 Create Logical Architecture:

The logical architecture allows a user to identify as a single Logical Schema a group of similar physical schemas - that is containing data stores that are structurally identical - but located in different physical locations. Logical Schemas, like their physical counterpart, are attached to a technology.

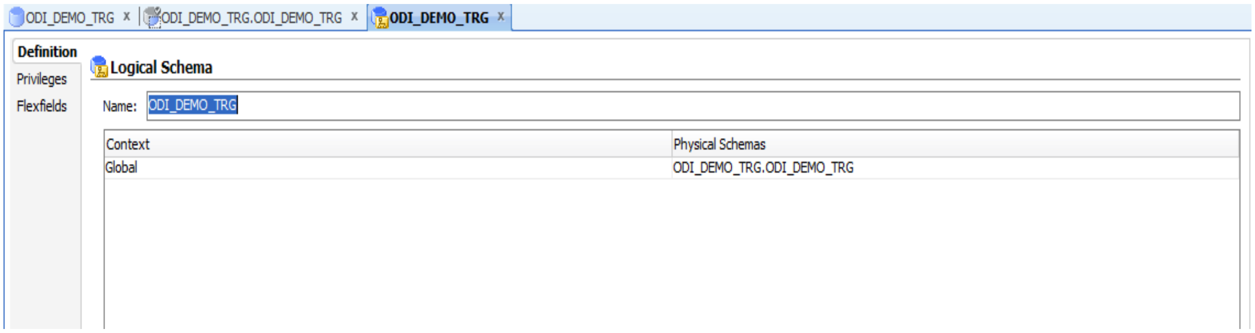
Context allows resolving logical schemas into physical schemas. In a given context, one logical schema resolves in a single physical schema.

Go to Topology →Logical Architecture → Technologies→JMS Topic XML then right click

1. Select New Logical Schema.



2. Provide name for Logical Schema.



Name: ODI_DEMO_TRG

Context	Physical Schemas
Global	ODI_DEMO_TRG.ODI_DEMO_TRG

3.3 Create Model:

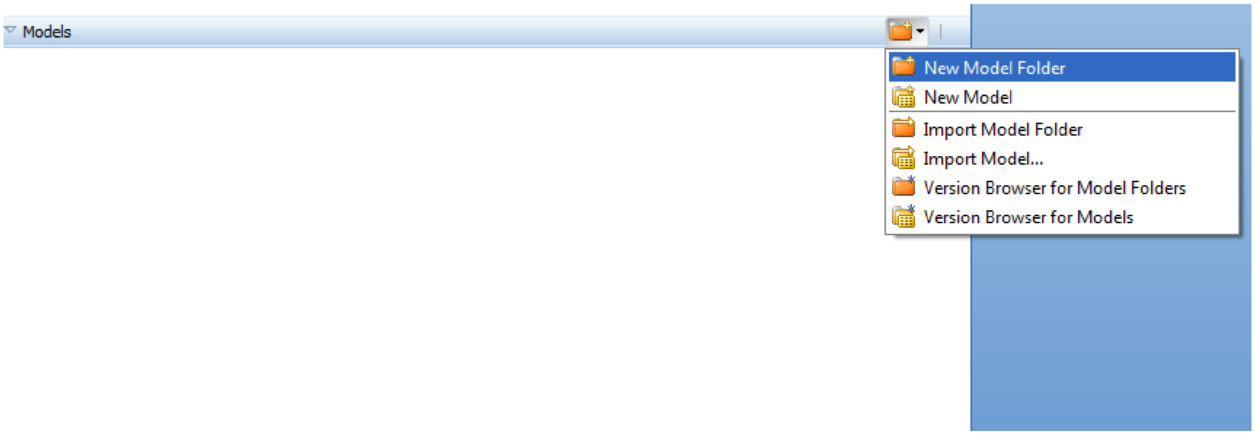
A Model is the description of a set of data stores. It corresponds to a group of tabular data structures stored in a data server. A model is based on a Logical Schema defined in the topology. In a given Context, this Logical Schema is mapped to a Physical Schema. The Data Schema of this Physical

Schema contains physical data structure: tables, files, JMS messages, elements from an XML file, that are represented as data stores.

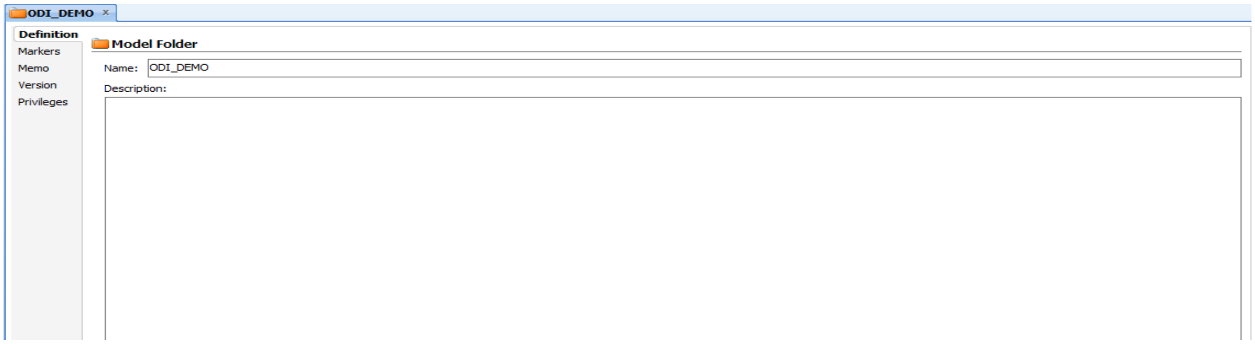
Models as well as all their components are based on the relational paradigm (table, columns, keys, etc.). Models in Data Integrator only contain *Meta data*, that is the description of the data structures. They do not contain a copy of the actual data.

Go to Designer →Models

1. Select New Folder



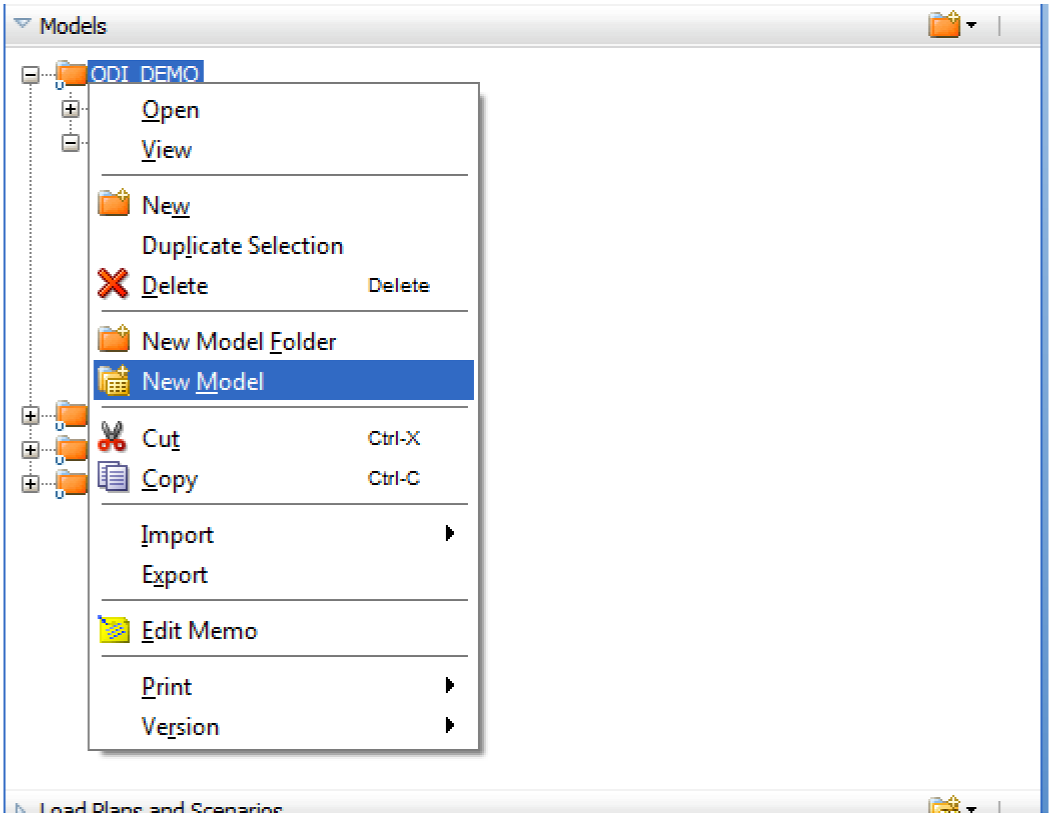
2. Provide name for folder



Name: ODI_DEMO

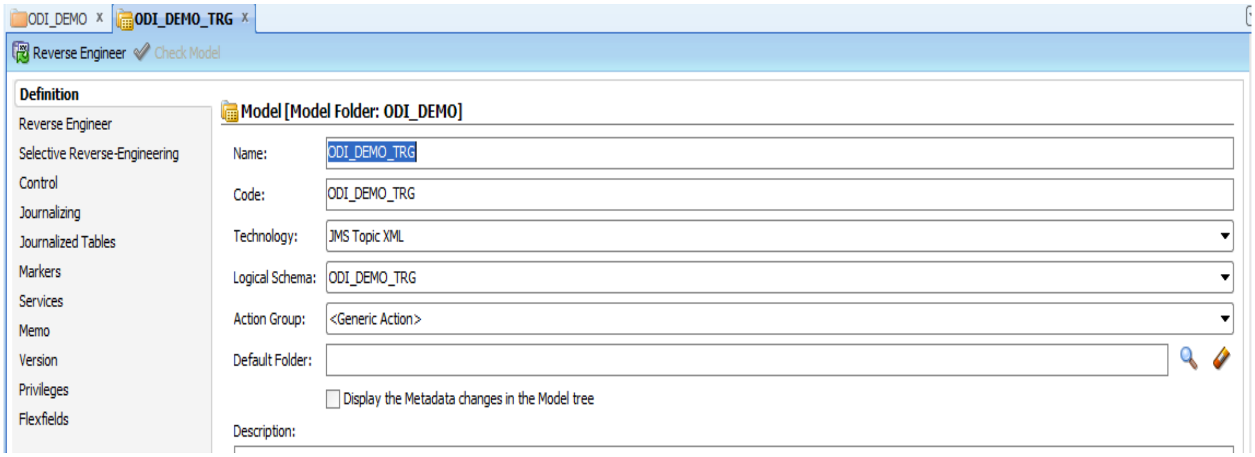
Description:

3. Select New Model



4. Provide Model Information

   a. **Name**: Name of the model used in the user interface.
   b. **Technology**: Select JMS TOPIC XML.
   c. **Logical Schema**: Select the Logical Schema.



Model [Model Folder: ODI_DEMO]

Name: ODI_DEMO_TRG

Code: ODI_DEMO_TRG

Technology: JMS Topic XML

Logical Schema: ODI_DEMO_TRG

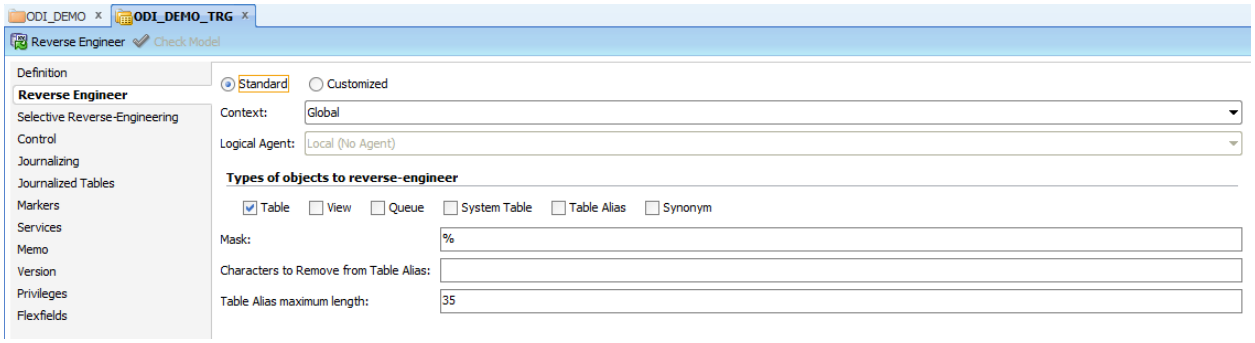
Action Group: <Generic Action>

Default Folder:

☐ Display the Metadata changes in the Model tree

Description:

5. Select Reverse Engineer



● Standard ○ Customized

Context: Global

Logical Agent: Local (No Agent)

Types of objects to reverse-engineer

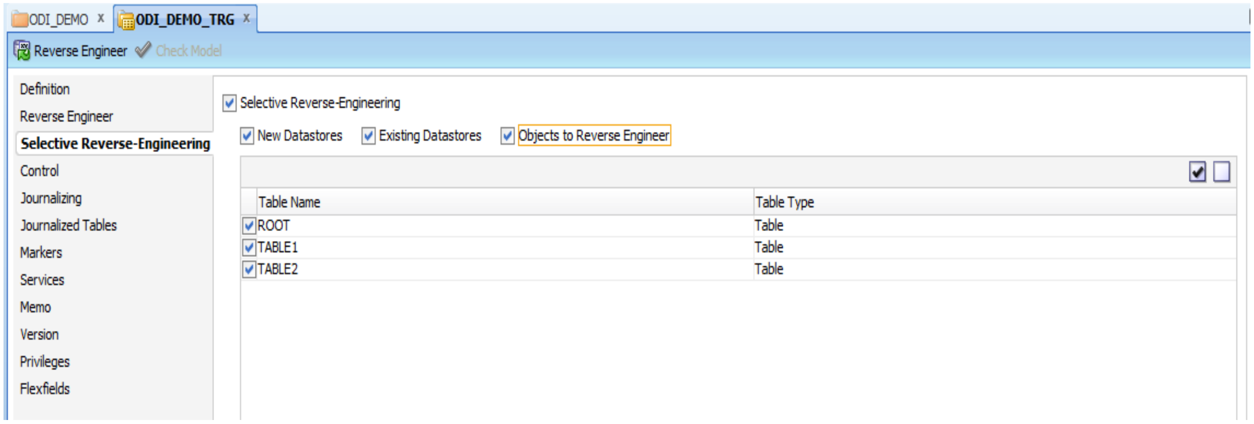
☑ Table ☐ View ☐ Queue ☐ System Table ☐ Table Alias ☐ Synonym

Mask: %

Characters to Remove from Table Alias:

Table Alias maximum length: 35

6. Selective Reverse Engineering and Click on Reverse Engineer.

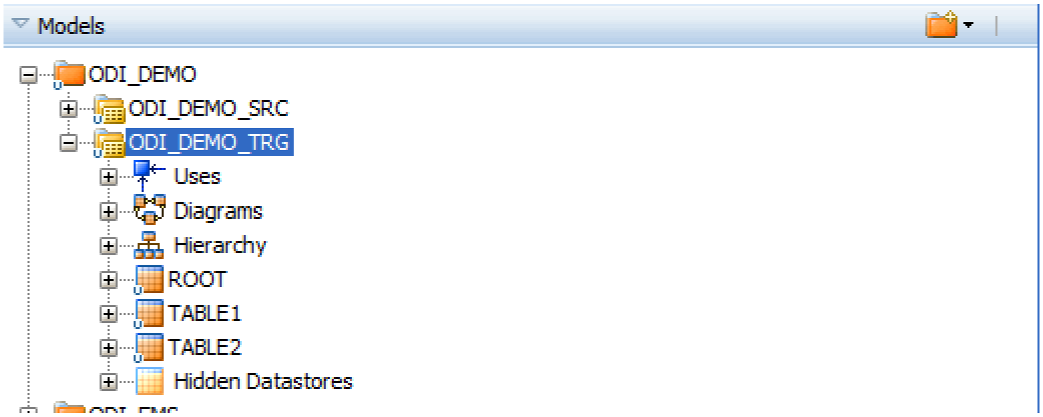


☑ Selective Reverse-Engineering

☑ New Datastores ☑ Existing Datastores ☑ Objects to Reverse Engineer

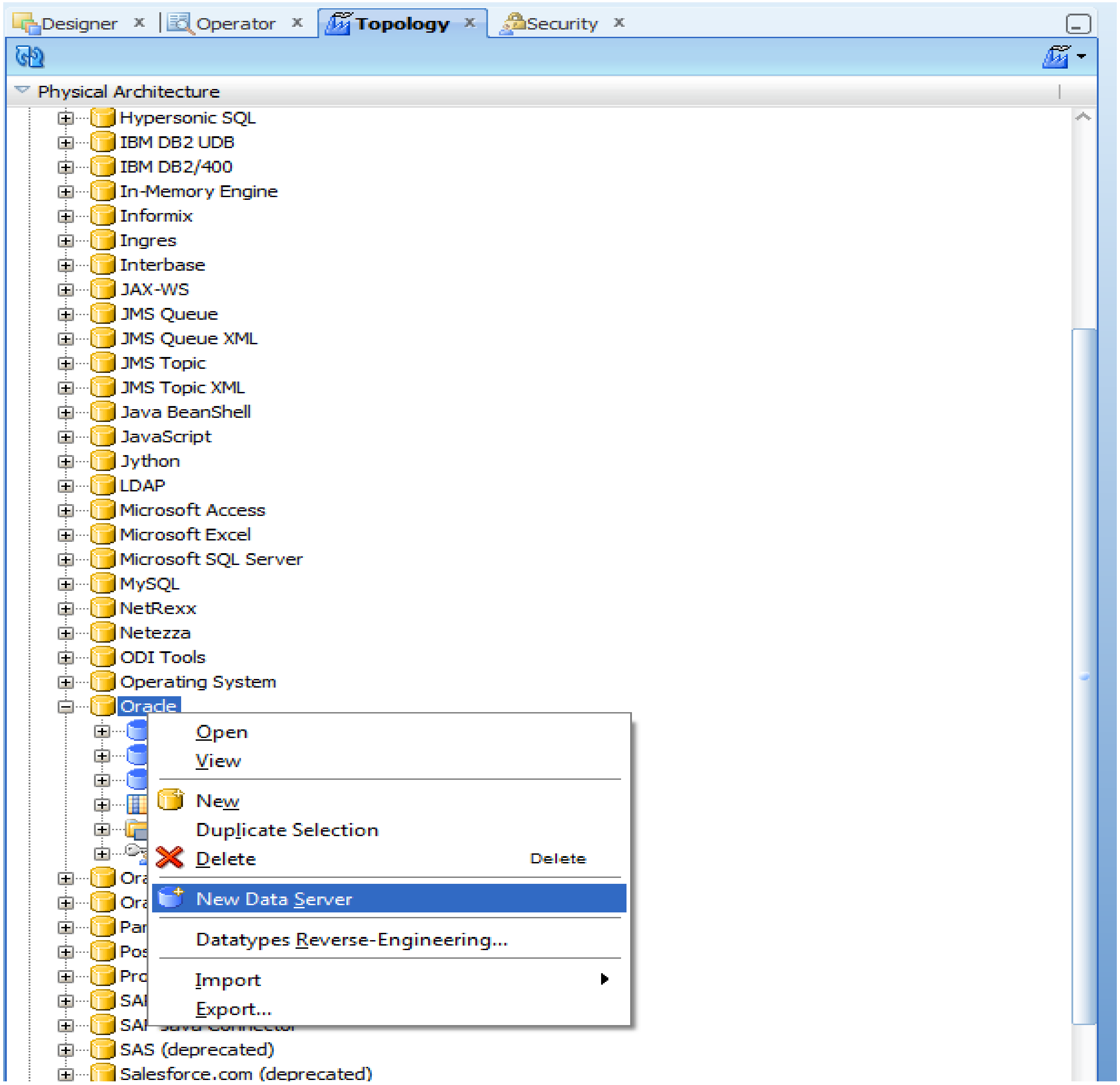
Table Name	Table Type
☑ ROOT	Table
☑ TABLE1	Table
☑ TABLE2	Table

7. After Reverse Engineering Selected Data Stores shown like below

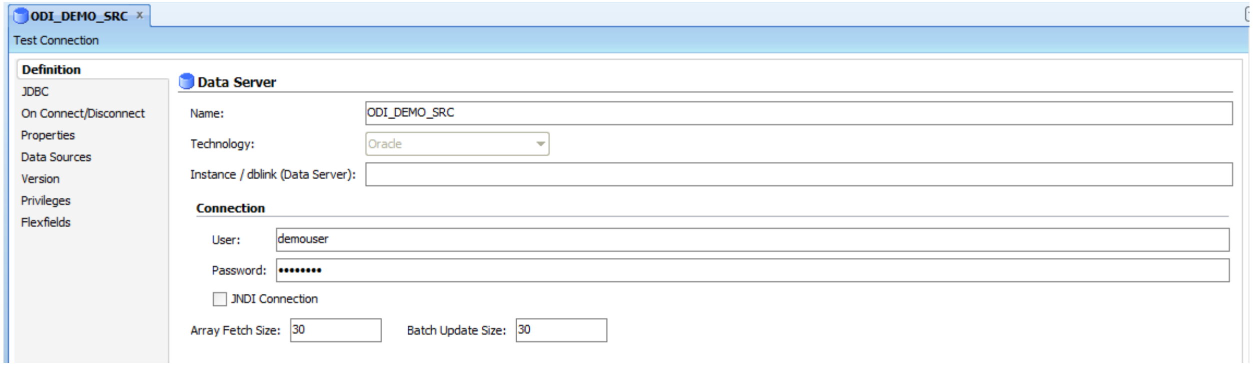


4. Create Source Data Store for Oracle

4.1 Create Physical Architecture: Go to Topology →Physical Architecture → Technologies→Oracle then right click and select New Data Server.



1. Provide name for Data Server

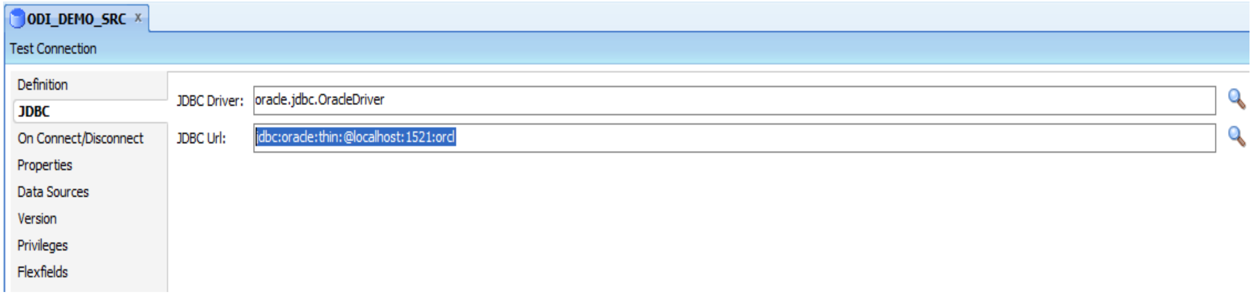


Name: ODI_DEMO_SRC
Technology: Oracle
Instance / dblink (Data Server):
Connection
User: demouser
Password: ••••••••
□ JNDI Connection
Array Fetch Size: 30
Batch Update Size: 30

2. Select JDBC tab

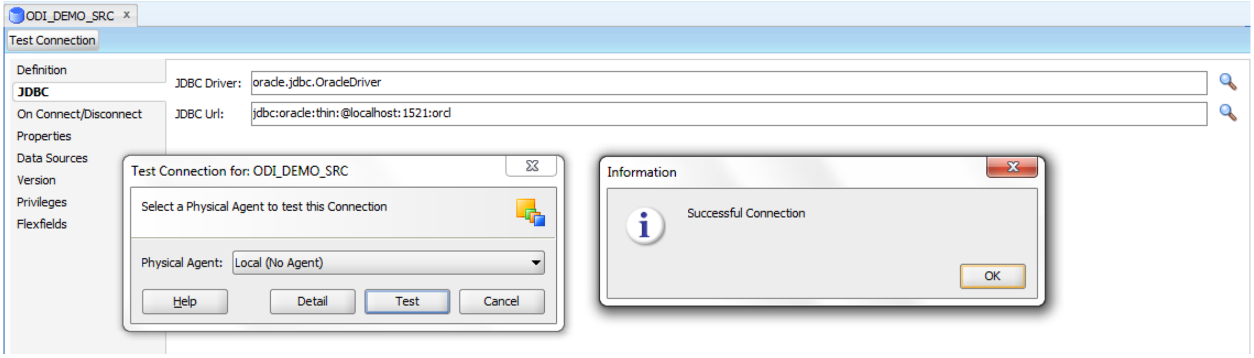
JDBC Driver: oracle.jdbc.OracleDriver

JDBC URL: jdbc:oracle:thin:@localhost:1521:orcl



JDBC Driver: oracle.jdbc.OracleDriver
JDBC Url: jdbc:oracle:thin:@localhost:1521:orcl

3. Test Connection

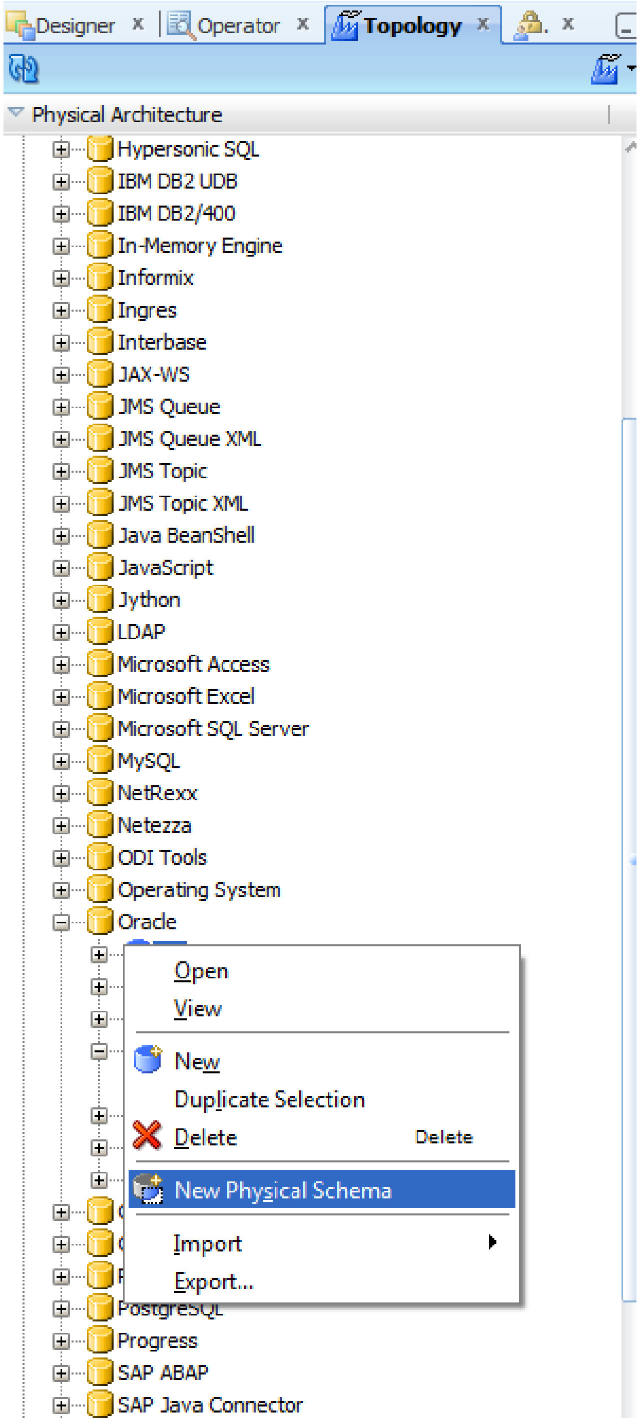


JDBC Driver: oracle.jdbc.OracleDriver
JDBC Url: jdbc:oracle:thin:@localhost:1521:orcl

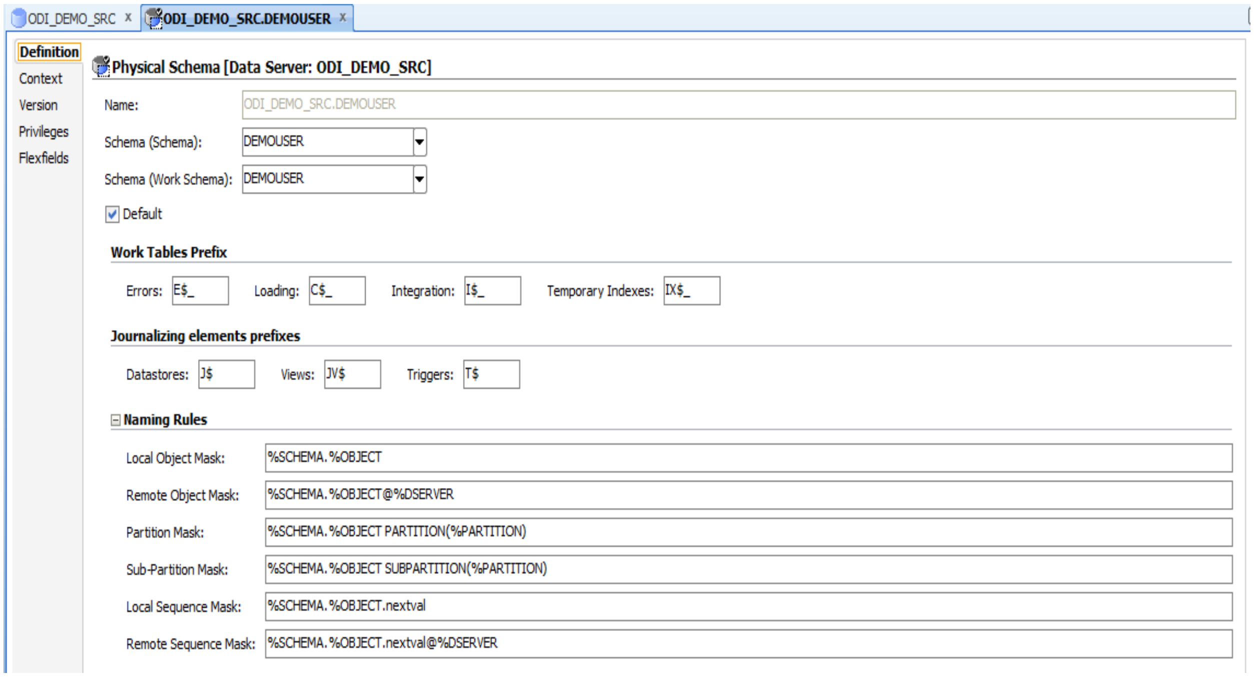
Test Connection for: ODI_DEMO_SRC
Select a Physical Agent to test this Connection
Physical Agent: Local (No Agent)

Successful Connection

4. Create New Physical Schema
    1. Select New Physical Schema



2. Select Schema i.e DEMOUSER



Physical Schema [Data Server: ODI_DEMO_SRC]

Name: ODI_DEMO_SRC.DEMOUSER

Schema (Schema): DEMOUSER

Schema (Work Schema): DEMOUSER

☑ Default

Work Tables Prefix

Errors: E$_ Loading: C$_ Integration: I$_ Temporary Indexes: IX$_

Journalizing elements prefixes

Datastores: J$ Views: JV$ Triggers: T$

Naming Rules

Local Object Mask: %SCHEMA.%OBJECT

Remote Object Mask: %SCHEMA.%OBJECT@%DSERVER

Partition Mask: %SCHEMA.%OBJECT PARTITION(%PARTITION)

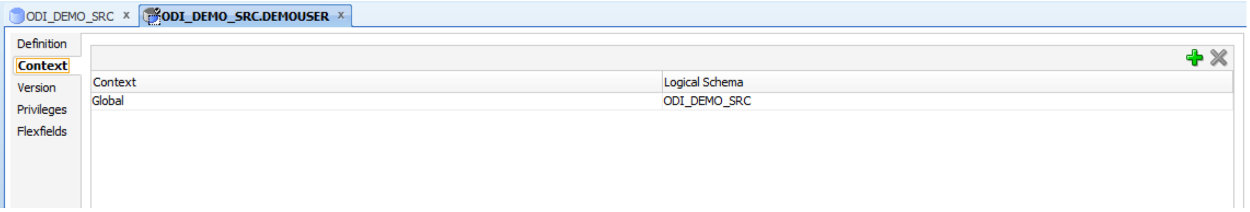
Sub-Partition Mask: %SCHEMA.%OBJECT SUBPARTITION(%PARTITION)

Local Sequence Mask: %SCHEMA.%OBJECT.nextval

Remote Sequence Mask: %SCHEMA.%OBJECT.nextval@%DSERVER

3. Select Context Tab and Select Logical Schema.

Note: Create Logical Architecture with undefined physical schema then link physical to logical here.

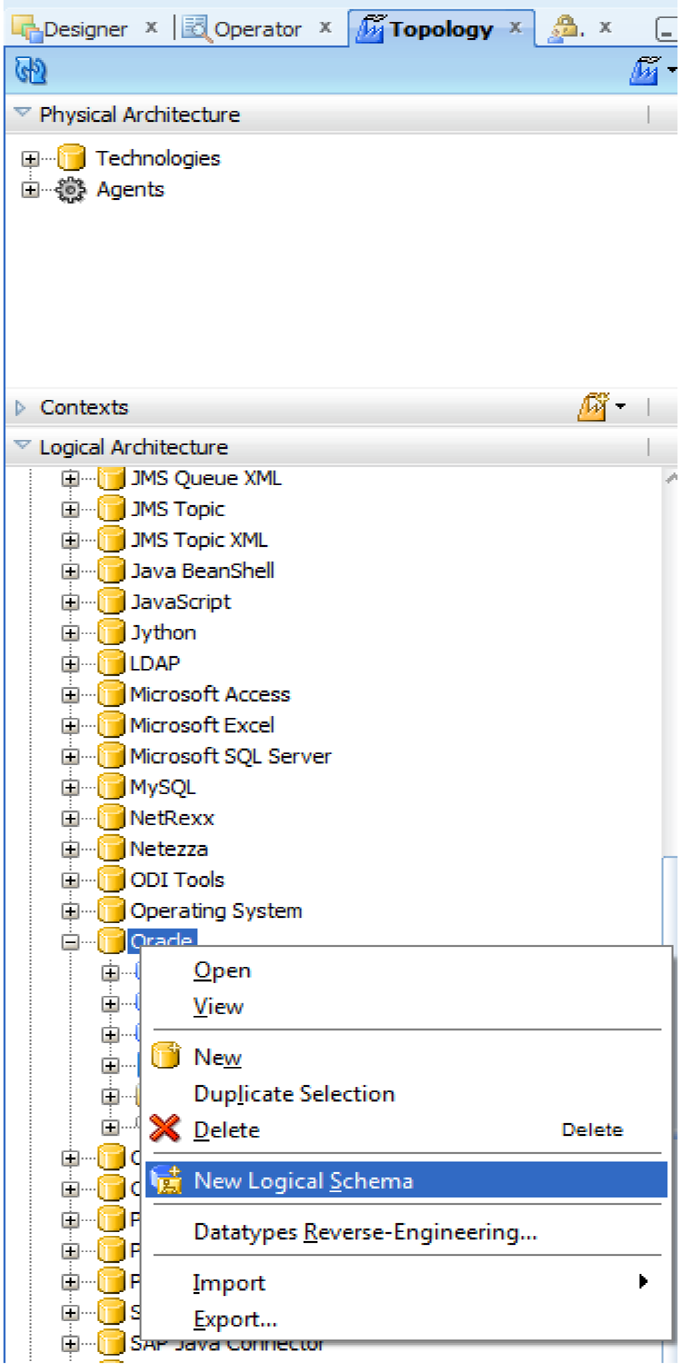


Context	Logical Schema
Global	ODI_DEMO_SRC

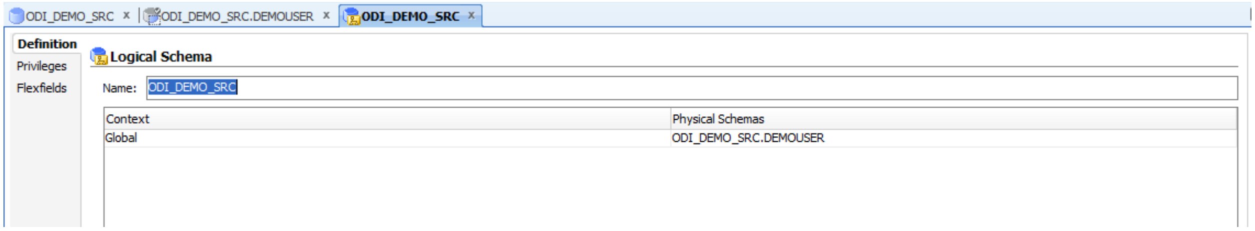
4.2 Create Logical Architecture:

Go to Topology →Logical Architecture → Technologies→Oracle then right click

1. Select New Logical Schema.



2. Provide name for Logical Schema.



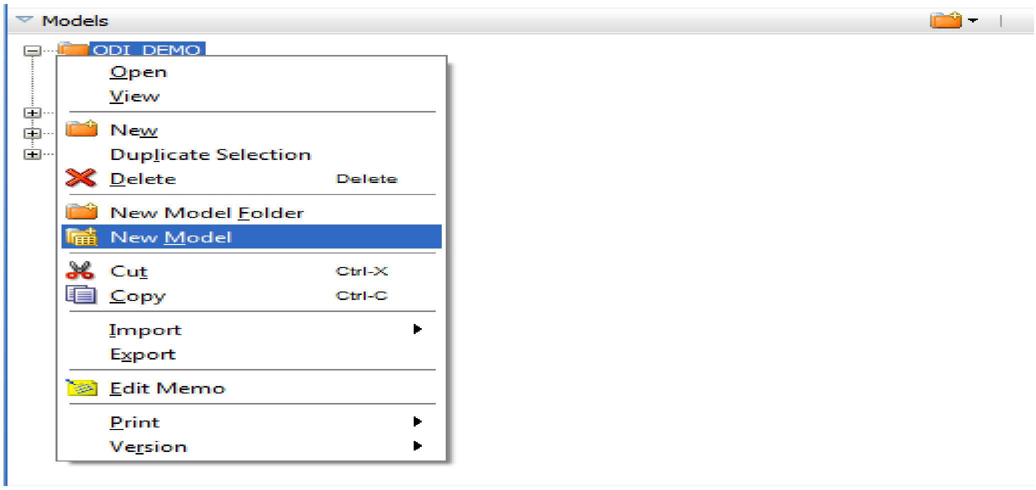
Name: ODI_DEMO_SRC

Context	Physical Schemas
Global	ODI_DEMO_SRC.DEMOUSER

4.3 Create Model:

Go to Designer →Models

1. Create New Folder and Select New Model

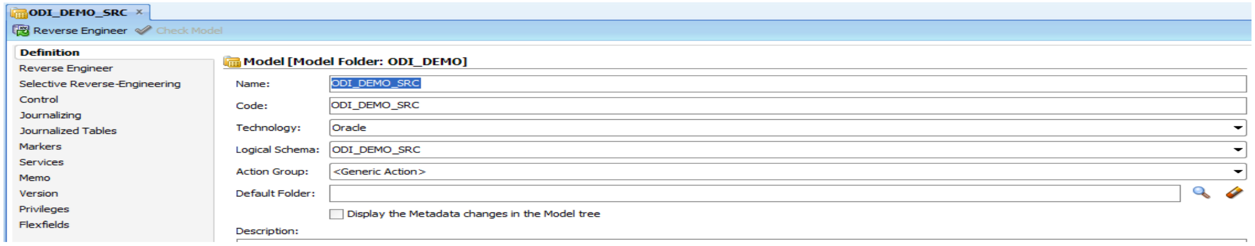


2. Provide Model Information

Name: Name of the model used in the user interface.

Technology: Select ORACLE.

Logical Schema: Select the Logical Schema.



Model [Model Folder: ODI_DEMO]

Name: ODI_DEMO_SRC

Code: ODI_DEMO_SRC

Technology: Oracle

Logical Schema: ODI_DEMO_SRC

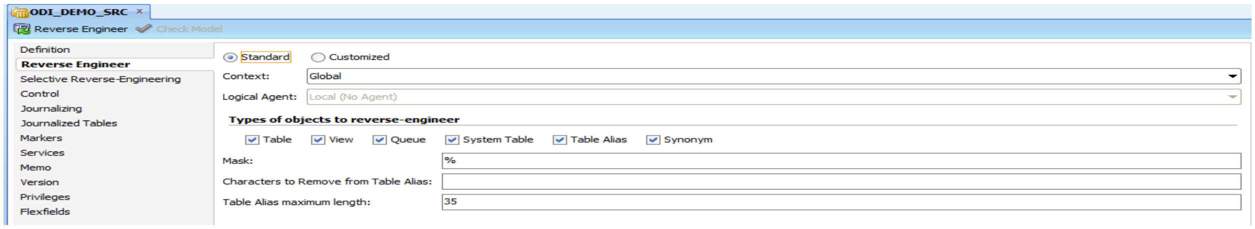
Action Group: <Generic Action>

Default Folder:

☐ Display the Metadata changes in the Model tree

Description:

3. Select Reverse Engineer



● Standard ○ Customized

Context: Global

Logical Agent: Local (No Agent)

Types of objects to reverse-engineer

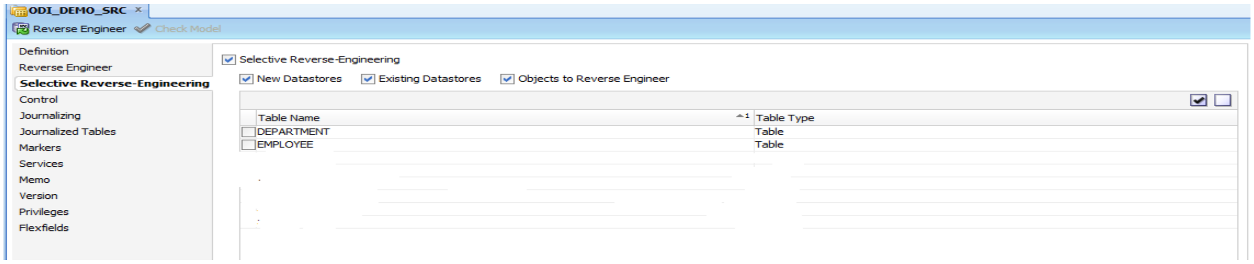
☑ Table ☑ View ☑ Queue ☑ System Table ☑ Table Alias ☑ Synonym

Mask: %

Characters to Remove from Table Alias:

Table Alias maximum length: 35

4. Selective Reverse Engineering and Click on Reverse Engineer.

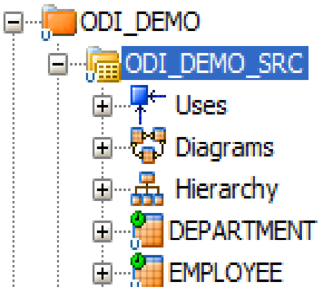


☑ Selective Reverse-Engineering

☑ New Datastores ☑ Existing Datastores ☑ Objects to Reverse Engineer

	Table Name	Table Type
□	DEPARTMENT	Table
□	EMPLOYEE	Table

5. After Reverse Engineering Selected Data Stores shown like below



5. Create Project

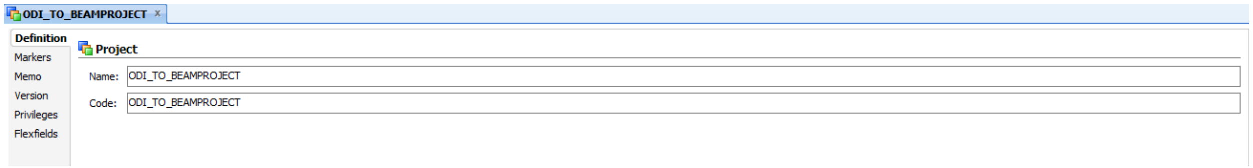
Now Source (ORACLE) and Target (JMS) are available. Now create Project, Interfaces and Packages.

We have two example package implementations

1. Single Row Per Message by using variables and filter
2. Batch Rows per Message by using Temp Interfaces and Filters.

5.1 Create New Project

Designer Navigator, click **New Project** in the toolbar of the **Projects**. Enter the **Name** of the project

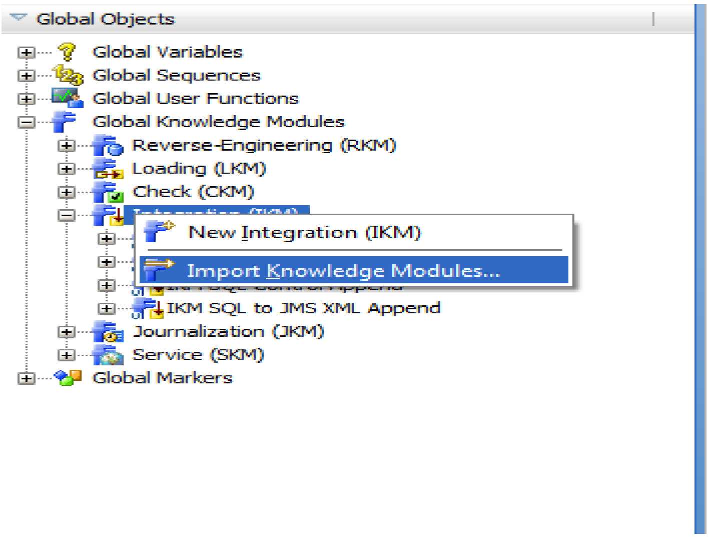


Name: ODI_TO_BEAMPROJECT

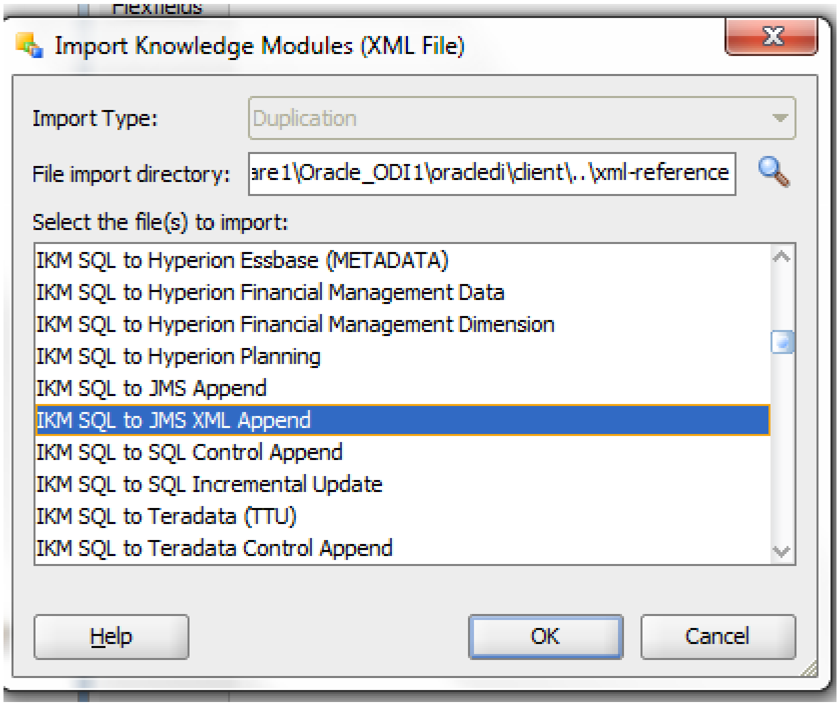
Code: ODI_TO_BEAMPROJECT

5.2 Import Knowledge modules

1. Go to Global Objects in Designer then Import Knowledge Modules.



2. Import IKM SQL to JMS XML Append



Import Type: Duplication

File import directory: are1\Oracle_ODI1\oracledi\client\..\xml-reference

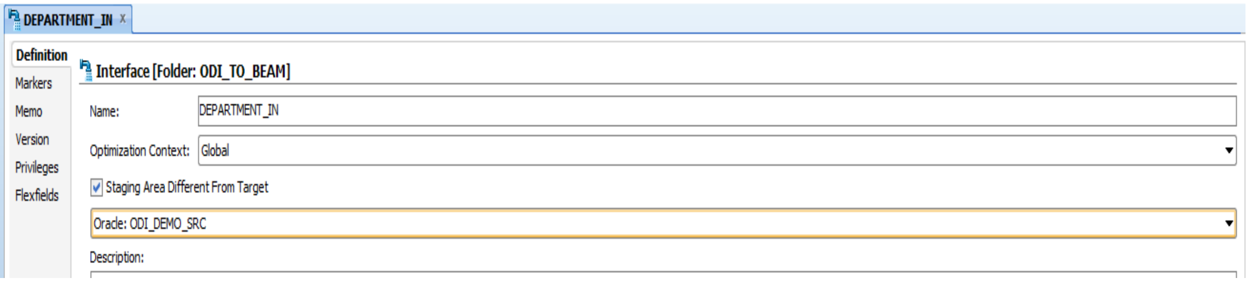
Select the file(s) to import:

IKM SQL to Hyperion Essbase (METADATA)
IKM SQL to Hyperion Financial Management Data
IKM SQL to Hyperion Financial Management Dimension
IKM SQL to Hyperion Planning
IKM SQL to JMS Append
IKM SQL to JMS XML Append
IKM SQL to SQL Control Append
IKM SQL to SQL Incremental Update
IKM SQL to Teradata (TTU)
IKM SQL to Teradata Control Append

5.3 Single Row Per Message

5.3.1 Create Interfaces

1. Crete Folder then Create Interface DEPARTMENT_IN.



Name: DEPARTMENT_IN

Optimization Context: Global

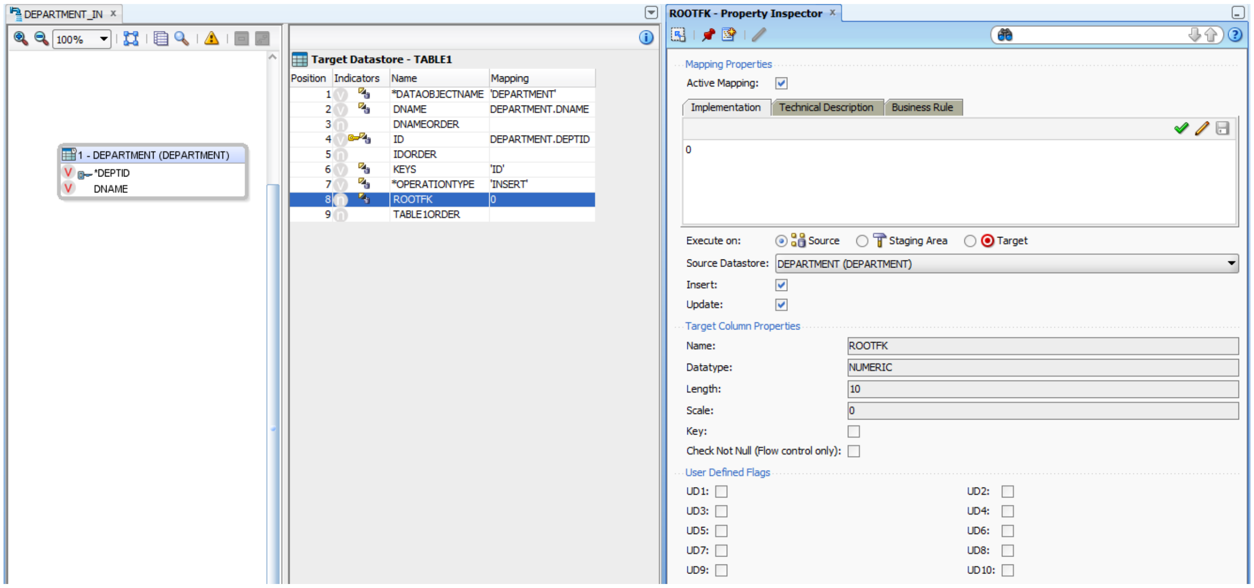
☑ Staging Area Different From Target

Oracle: ODI_DEMO_SRC

Description:

2. Go to Mappings Tab then Drag and Drop Department table into target form ODI_DEMO_SRC model and Department table form Oracle to Source. Columns with same name will get automatically mapped and other columns can be mapped by drag and drop from source to target.

Assign ROOTFK, OPERATION TYPE, KEYS and DATAOBJECT NAME to 0, ‘INSERT’, ‘ID’ and ‘DEPARTMENT’ respectively..



DEPARTMENT_IN

1 - DEPARTMENT (DEPARTMENT)
*DEPTID
DNAME

Target Datastore - TABLE1

Position	Indicators	Name	Mapping
1		*DATAOBJECTNAME	'DEPARTMENT'
2		DNAME	DEPARTMENT.DNAME
3		DNAMEORDER	
4		ID	DEPARTMENT.DEPTID
5		IDORDER	
6		KEYS	'ID'
7		*OPERATIONTYPE	'INSERT'
8		ROOTFK	0
9		TABLE1ORDER	

ROOTFK - Property Inspector

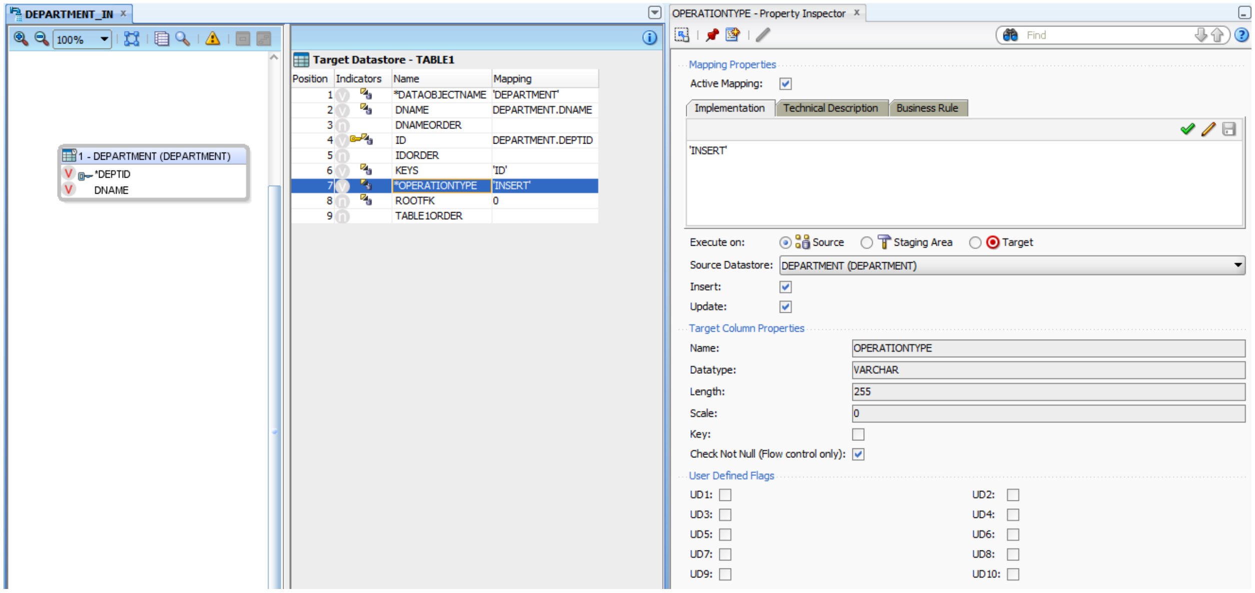
Mapping Properties
Active Mapping: ☑
Implementation | Technical Description | Business Rule
0

Execute on: ● Source ○ Staging Area ○ Target
Source Datastore: DEPARTMENT (DEPARTMENT)
Insert: ☑
Update: ☑

Target Column Properties
Name: ROOTFK
Datatype: NUMERIC
Length: 10
Scale: 0
Key: □
Check Not Null (Flow control only): □

User Defined Flags
UD1: □ UD2: □
UD3: □ UD4: □
UD5: □ UD6: □
UD7: □ UD8: □
UD9: □ UD10: □

2. Assign OPERATIONTYPE to 'INSERT'



DEPARTMENT_IN

1 - DEPARTMENT (DEPARTMENT)
*DEPTID
DNAME

Target Datastore - TABLE1

Position	Indicators	Name	Mapping
1		*DATAOBJECTNAME	'DEPARTMENT'
2		DNAME	DEPARTMENT.DNAME
3		DNAMEORDER	
4		ID	DEPARTMENT.DEPTID
5		IDORDER	
6		KEYS	'ID'
7		*OPERATIONTYPE	'INSERT'
8		ROOTFK	0
9		TABLE1ORDER	

OPERATIONTYPE - Property Inspector

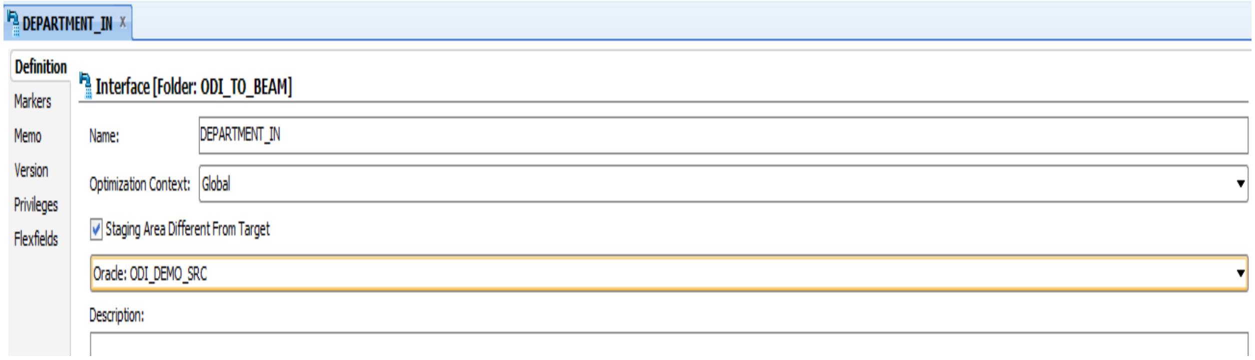
Mapping Properties
Active Mapping: ☑
Implementation | Technical Description | Business Rule
'INSERT'

Execute on: ● Source ○ Staging Area ○ Target
Source Datastore: DEPARTMENT (DEPARTMENT)
Insert: ☑
Update: ☑

Target Column Properties
Name: OPERATIONTYPE
Datatype: VARCHAR
Length: 255
Scale: 0
Key: □
Check Not Null (Flow control only): ☑

User Defined Flags
UD1: □ UD2: □
UD3: □ UD4: □
UD5: □ UD6: □
UD7: □ UD8: □
UD9: □ UD10: □

3. Go to Overview tab and Select check box 'Staging area different form Target' and then select Oracle Source from Drop Down.



Interface [Folder: ODI_TO_BEAM]

Name: DEPARTMENT_IN

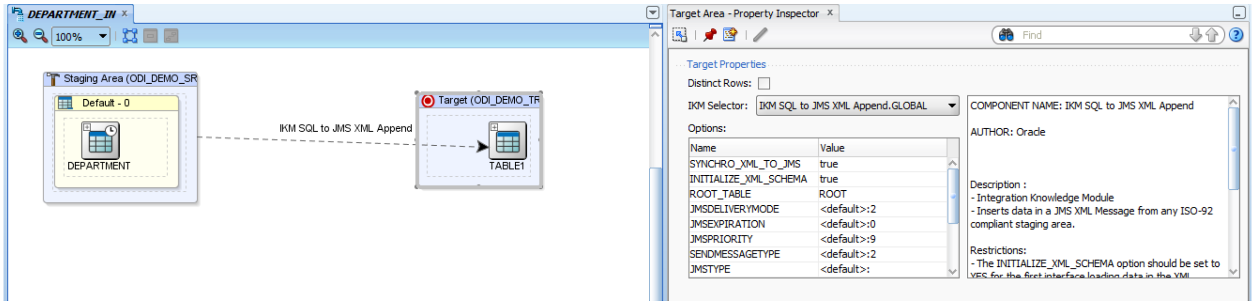
Optimization Context: Global

☑ Staging Area Different From Target

Oracle: ODI_DEMO_SRC

Description:

4. Go to Flow tab select target then it will show Property Inspector
   Select IKM – IKM SQL to JMS XML APPEND
   Options:
   SYNCHRO_XM_TO_JMS to true
   INITIALIZE_XML_SCHEMA to true
   ROOT_TABLE to ROOT.


DEPARTMENT_IN
100%
Staging Area (ODI_DEMO_SR
Default - 0
DEPARTMENT
IKM SQL to JMS XML Append
Target (ODI_DEMO_TR
TABLE1
Target Area - Property Inspector
Find

Target Properties

Distinct Rows: □

IKM Selector: IKM SQL to JMS XML Append.GLOBAL

Options:

Name	Value
SYNCHRO_XML_TO_JMS	true
INITIALIZE_XML_SCHEMA	true
ROOT_TABLE	ROOT
JMSDELIVERYMODE	<default>:2
JMSEXPIRATION	<default>:0
JMSPRIORITY	<default>:9
SENDMESSAGETYPE	<default>:2
JMSTYPE	<default>:

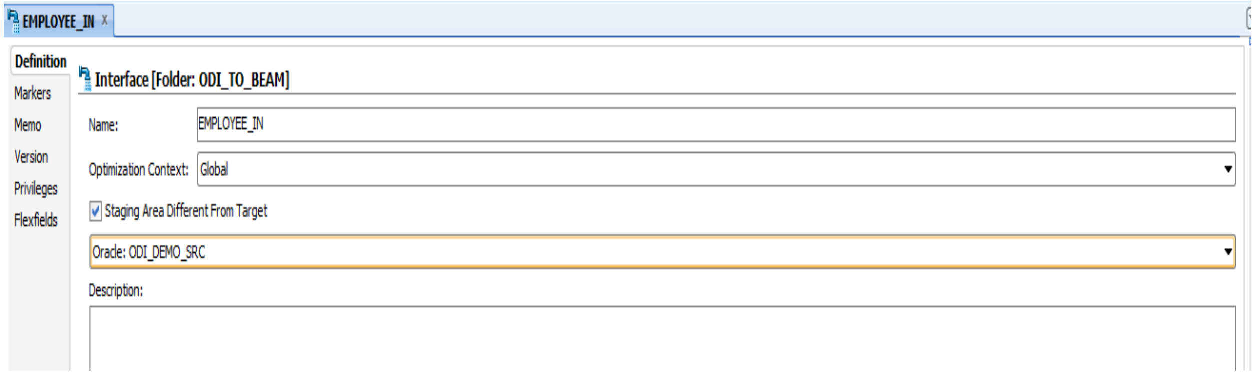
COMPONENT NAME: IKM SQL to JMS XML Append

AUTHOR: Oracle

Description :
- Integration Knowledge Module
- Inserts data in a JMS XML Message from any ISO-92 compliant staging area.

Restrictions:
- The INITIALIZE_XML_SCHEMA option should be set to YES for the first interface loading data in the XML

5. Create Interface EMPLOYEE_IN



Interface [Folder: ODI_TO_BEAM]

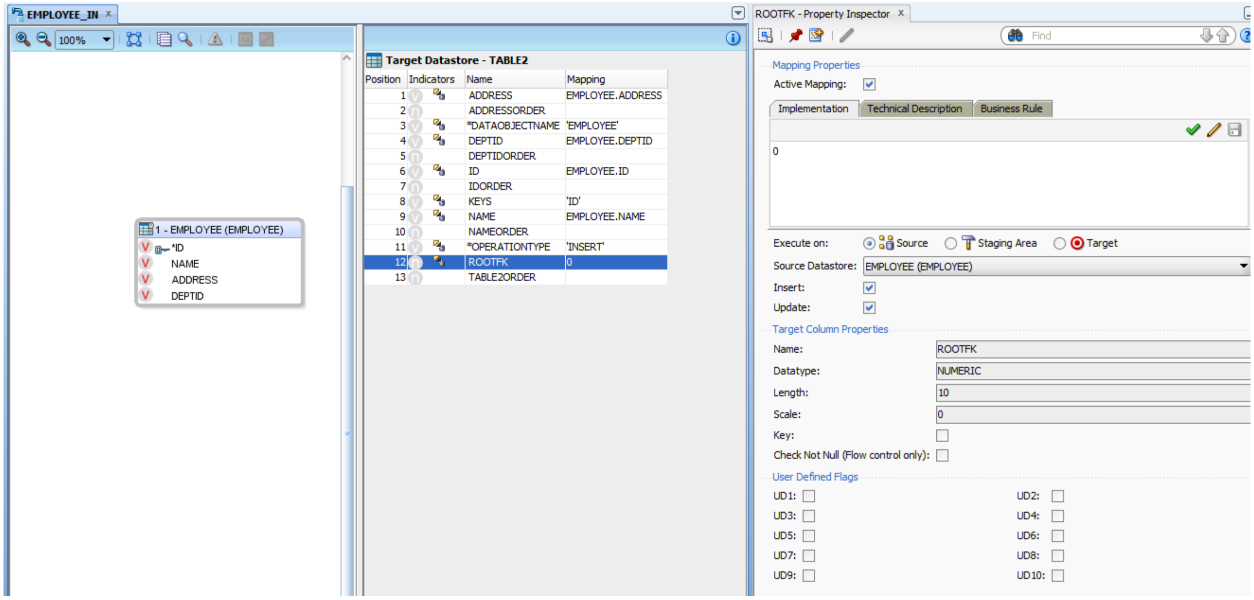
Name: EMPLOYEE_IN

Optimization Context: Global

☑ Staging Area Different From Target

Oracle: ODI_DEMO_SRC

Description:



Target Datastore - TABLE2

Position	Indicators	Name	Mapping
1	V	ADDRESS	EMPLOYEE.ADDRESS
2	n	ADDRESSORDER	
3	V	*DATAOBJECTNAME	'EMPLOYEE'
4	V	DEPTID	EMPLOYEE.DEPTID
5	n	DEPTIDORDER	
6	V	ID	EMPLOYEE.ID
7	n	IDORDER	
8	V	KEYS	'ID'
9	V	NAME	EMPLOYEE.NAME
10	n	NAMEORDER	
11	V	*OPERATIONTYPE	'INSERT'
12	n	ROOTFK	0
13	n	TABLE2ORDER	

ROOTFK - Property Inspector

Mapping Properties

Active Mapping: ☑

Implementation | Technical Description | Business Rule

0

Execute on: ● Source ○ Staging Area ○ Target

Source Datastore: EMPLOYEE (EMPLOYEE)

Insert: ☑

Update: ☑

Target Column Properties

Name: ROOTFK

Datatype: NUMERIC

Length: 10

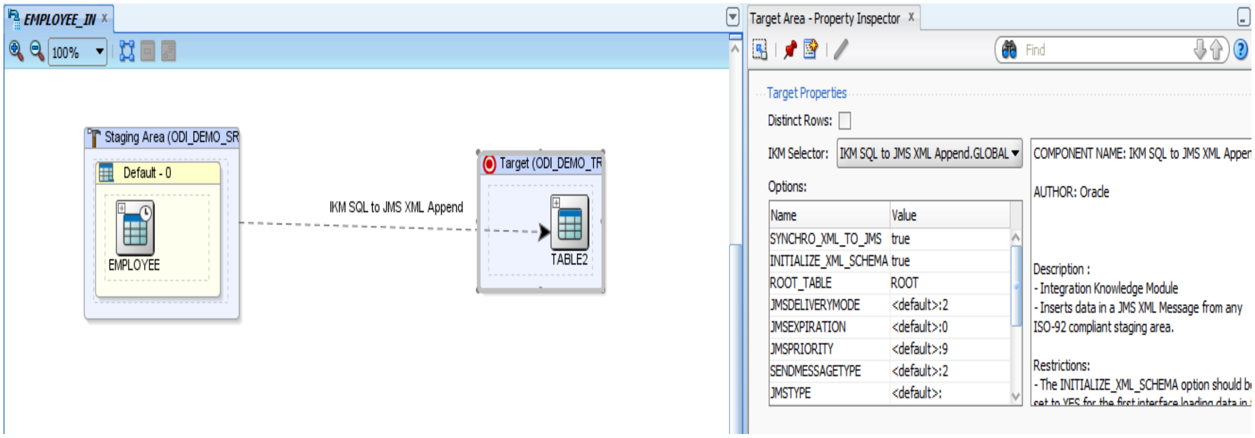
Scale: 0

Key: □

Check Not Null (Flow control only): □

User Defined Flags

UD1: □ UD2: □
UD3: □ UD4: □
UD5: □ UD6: □
UD7: □ UD8: □
UD9: □ UD10: □


EMPLOYEE_IN
100%
Staging Area (ODI_DEMO_SR
Default - 0
EMPLOYEE
IKM SQL to JMS XML Append
Target (ODI_DEMO_TR
TABLE2
Target Area - Property Inspector
Find

Target Properties

Distinct Rows: □

IKM Selector: IKM SQL to JMS XML Append.GLOBAL

Options:

Name	Value
SYNCHRO_XML_TO_JMS	true
INITIALIZE_XML_SCHEMA	true
ROOT_TABLE	ROOT
JMSDELIVERYMODE	<default>:2
JMSEXPIRATION	<default>:0
JMSPRIORITY	<default>:9
SENDMESSAGETYPE	<default>:2
JMSTYPE	<default>:

COMPONENT NAME: IKM SQL to JMS XML Appen

AUTHOR: Oracle

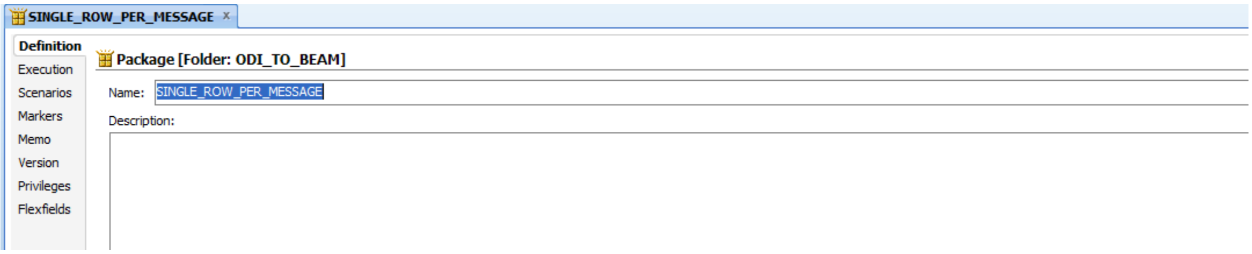
Description :
- Integration Knowledge Module
- Inserts data in a JMS XML Message from any ISO-92 compliant staging area.

Restrictions:
- The INITIALIZE_XML_SCHEMA option should b
set to YES for the first interface loading data in

6. Create Folder as ODI_TO_BEAM and Package as SINGLE_ROW_PER_MESSAGE

Now we are implementing this package which delivers one row per JMS message. Following steps needs to be performed.

1. Create required variables
2. Update Interface with filters
3. Create Package flow with variables and filters



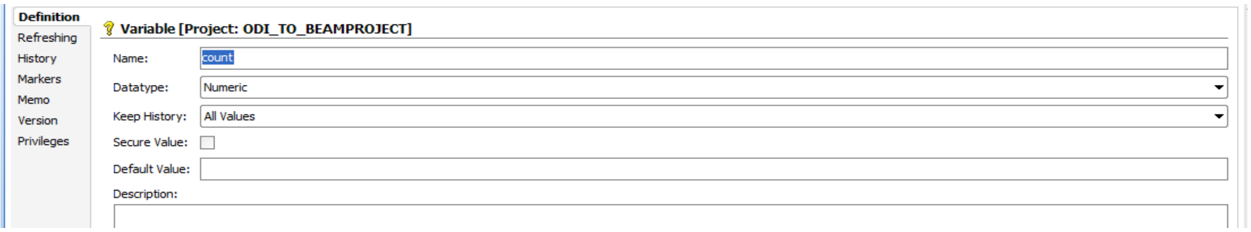
Package [Folder: ODI_TO_BEAM]

Name: SINGLE_ROW_PER_MESSAGE

Description:

5.3.2 Create Variables

1. Create count variable



Variable [Project: ODI_TO_BEAMPROJECT]

Name: count

Datatype: Numeric

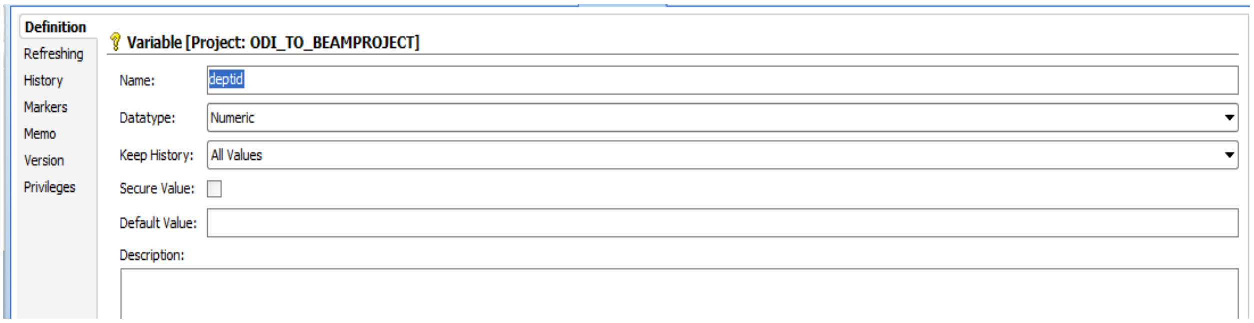
Keep History: All Values

Secure Value: □

Default Value:

Description:

2. Create deptid variable



Variable [Project: ODI_TO_BEAMPROJECT]

Name: deptid

Datatype: Numeric

Keep History: All Values

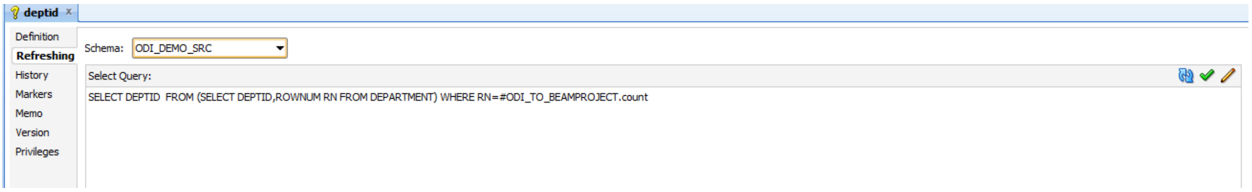
Secure Value: □

Default Value:

Description:

Go to refreshing tab and provide the following query.

SELECT DEPTID FROM (SELECT DEPTID, ROWNUM RN FROM DEPARTMENT) WHERE RN=#ODI_TO_BEAMPROJECT.count

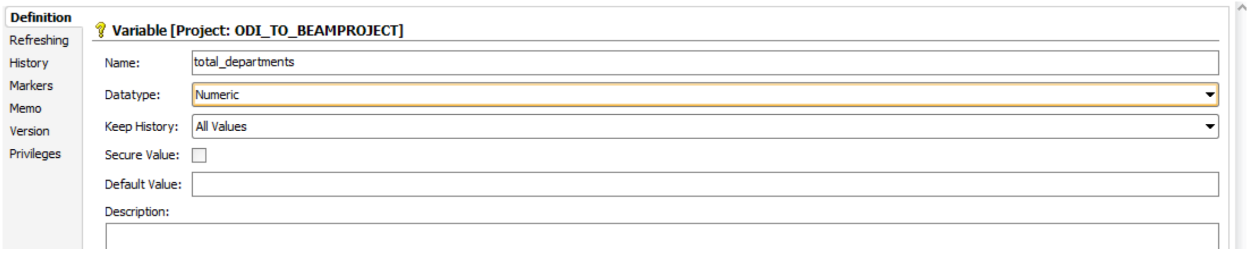


Schema: ODI_DEMO_SRC

Select Query:

```
SELECT DEPTID  FROM (SELECT DEPTID,ROWNUM RN FROM DEPARTMENT) WHERE RN=#ODI_TO_BEAMPROJECT.count
```

3. Create total_departments



Variable [Project: ODI_TO_BEAMPROJECT]

Name: total_departments

Datatype: Numeric

Keep History: All Values

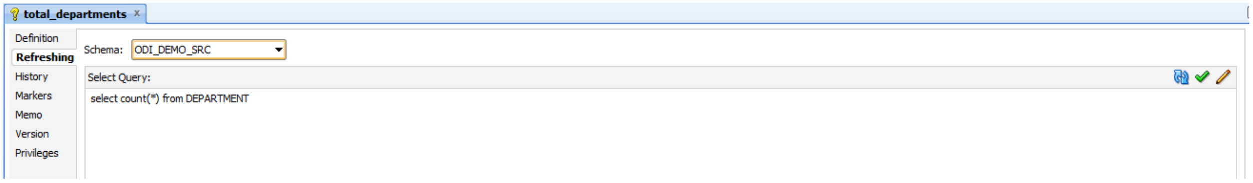
Secure Value: □

Default Value:

Description:

Go to refreshing tab and provide the following query.

select count(*) from DEPARTMENT

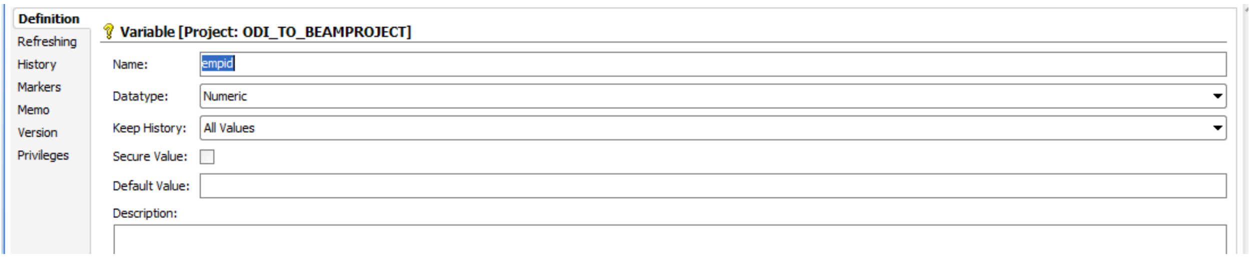


Schema: ODI_DEMO_SRC

Select Query:

```
select count(*) from DEPARTMENT
```

4. Create variable empid



Variable [Project: ODI_TO_BEAMPROJECT]

Name: empid

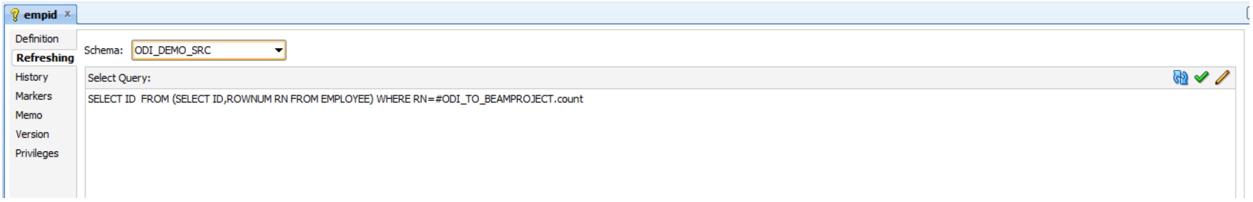
Datatype: Numeric

Keep History: All Values

Secure Value: □

Default Value:

Description:

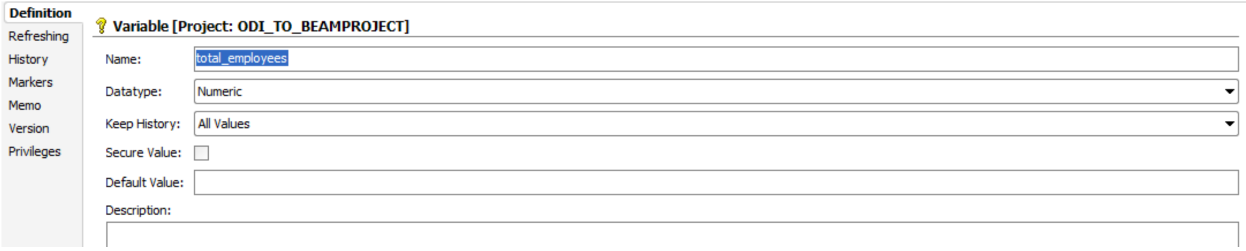


Schema: ODI_DEMO_SRC

Select Query:

```
SELECT ID  FROM (SELECT ID,ROWNUM RN FROM EMPLOYEE) WHERE RN=#ODI_TO_BEAMPROJECT.count
```

5. Create variable total_employees



Variable [Project: ODI_TO_BEAMPROJECT]

Name: total_employees

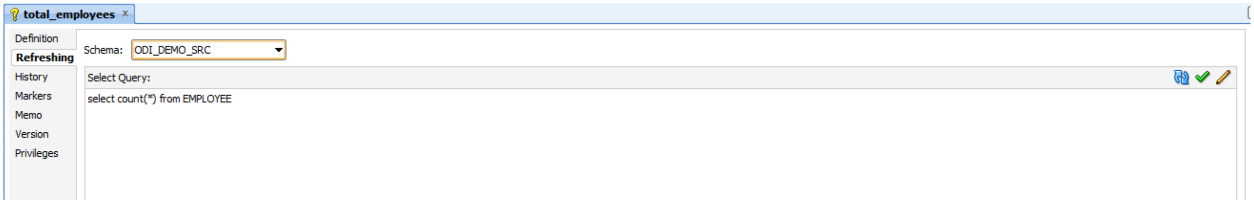
Datatype: Numeric

Keep History: All Values

Secure Value: □

Default Value:

Description:



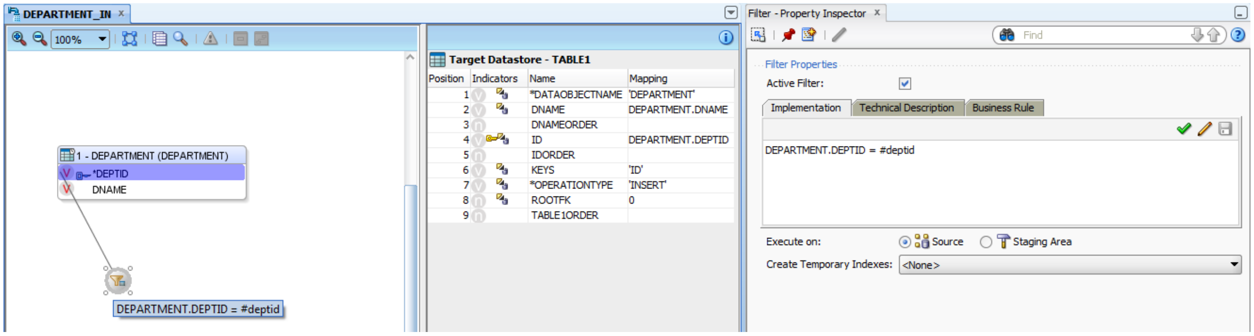
Schema: ODI_DEMO_SRC

Select Query:

```
select count(*) from EMPLOYEE
```

Define Filter on interfaces:

Update Interfaces with filters: Drag DEPTID from Source to blank area then filter shows up like below. Update the expression with above created variable.


DEPARTMENT_IN
100%
1 - DEPARTMENT (DEPARTMENT)
*DEPTID
DNAME
DEPARTMENT.DEPTID = #deptid
Filter - Property Inspector
Find
Implementation
Technical Description
Business Rule

Target Datastore - TABLE1

Position	Indicators	Name	Mapping
1		*DATAOBJECTNAME	'DEPARTMENT'
2		DNAME	DEPARTMENT.DNAME
3		DNAMEORDER	
4		ID	DEPARTMENT.DEPTID
5		IDORDER	
6		KEYS	'ID'
7		*OPERATIONTYPE	'INSERT'
8		ROOTFK	0
9		TABLE1ORDER	

Filter Properties

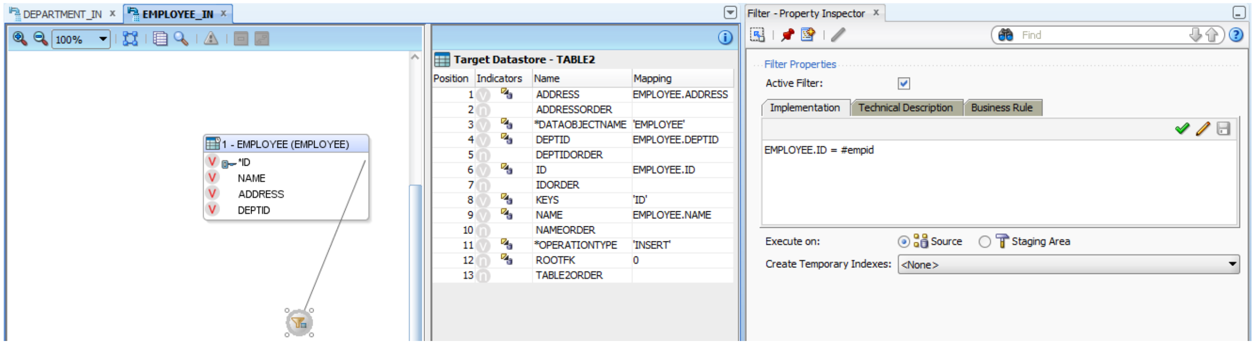
Active Filter: ☑

DEPARTMENT.DEPTID = #deptid

Execute on: ● Source ○ Staging Area

Create Temporary Indexes: <None>

Drag ID from Source to blank area then filter shows up like below. Update the expression with above created variable.



Target Datastore - TABLE2

Position	Indicators	Name	Mapping
1		ADDRESS	EMPLOYEE.ADDRESS
2		ADDRESSORDER	
3		*DATAOBJECTNAME	'EMPLOYEE'
4		DEPTID	EMPLOYEE.DEPTID
5		DEPTIDORDER	
6		ID	EMPLOYEE.ID
7		IDORDER	
8		KEYS	'ID'
9		NAME	EMPLOYEE.NAME
10		NAMEORDER	
11		*OPERATIONTYPE	'INSERT'
12		ROOTFK	0
13		TABLE2ORDER	

Filter Properties

Active Filter: ☑

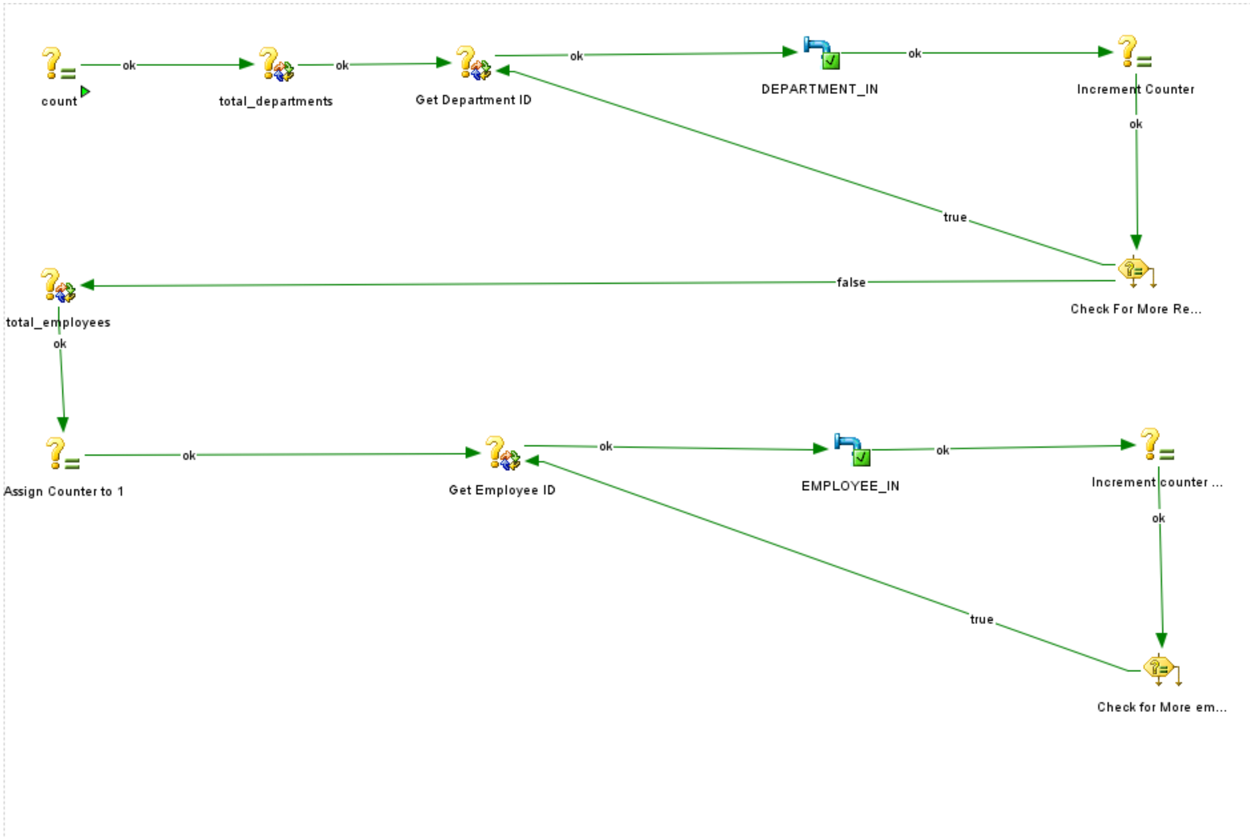
EMPLOYEE.ID = #empid

Execute on: ● Source ○ Staging Area

Create Temporary Indexes: <None>

5.3.3 Create Flow in package using variables and interfaces:

1. Assign count value to 1
2. Refresh the total_departments.
3. Refresh deptid which gets deptid
4. Add interface
5. Increment count by 1
6. Repeat step 3 if count <= total_departments
7. Refresh total_employees
8. Assign count value to 1
9. Refresh empid which gets id
10. Add interface
11. Increment count by 1
12. Repeat step 9 if count <= total_employees



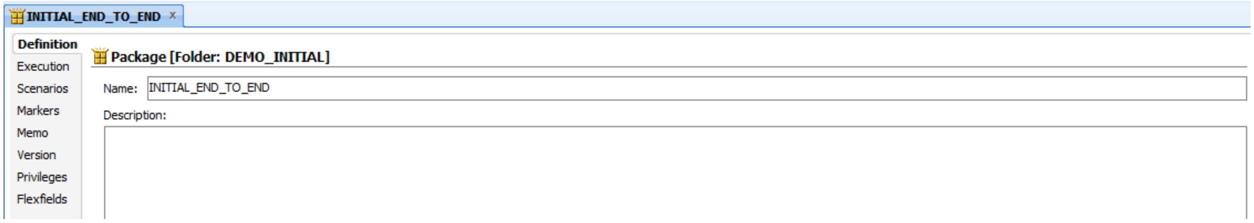
13. Execute Package: Execute package by clicking on ▶ and review the results in Session List of the Operator tab.

5.4 Batch Rows per Message

Now we are implementing this package which delivers defined number of rows per JMS message. Following steps needs to be performed.

1. Create required variables
2. Create Temp Interface with rownum
3. Create Interface which holds Temp Interface and add filter on rownum
4. Create Package flow with variables and interfaces

Create Project ODI_DEMO and Package INITIAL_END_TO_END



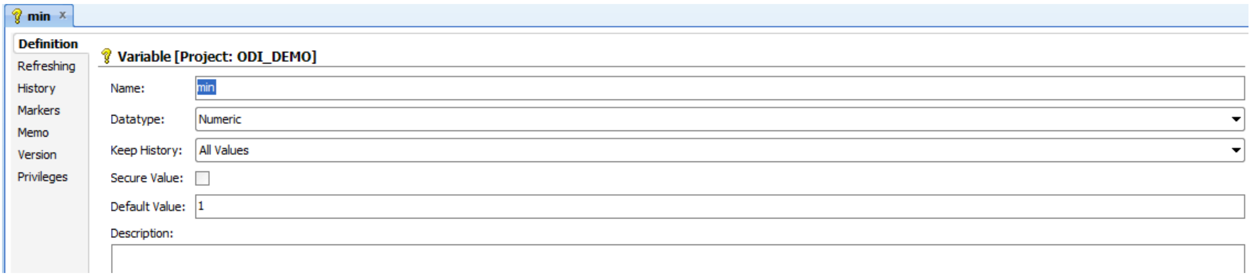
Package [Folder: DEMO_INITIAL]

Name: INITIAL_END_TO_END

Description:

5.4.1 Create Variables

1. Create variable min



Variable [Project: ODI_DEMO]

Name: min

Datatype: Numeric

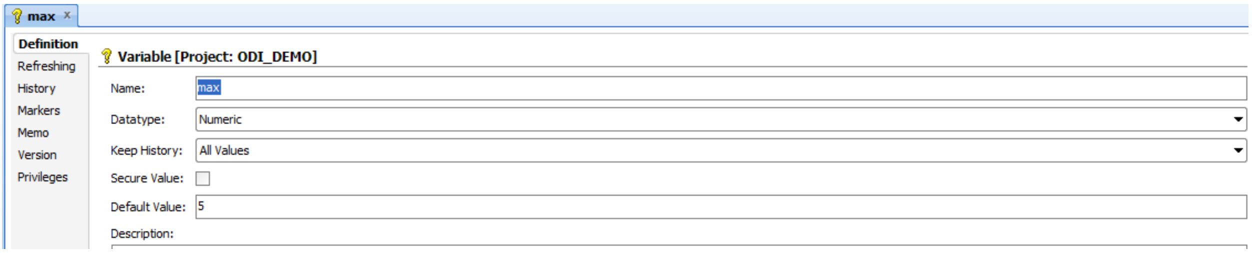
Keep History: All Values

Secure Value: □

Default Value: 1

Description:

2. Create variable max



Variable [Project: ODI_DEMO]

Name: max

Datatype: Numeric

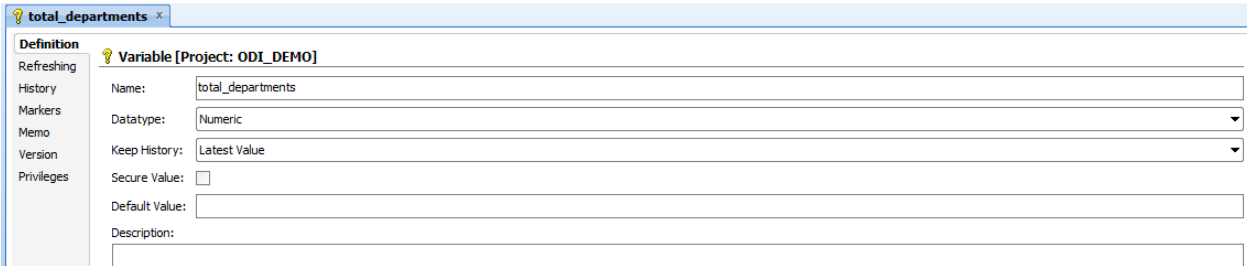
Keep History: All Values

Secure Value: □

Default Value: 5

Description:

3. Create Variable total_departments



Variable [Project: ODI_DEMO]

Name: total_departments

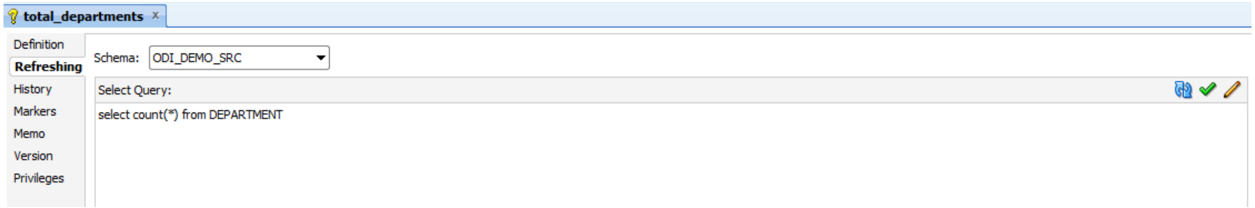
Datatype: Numeric

Keep History: Latest Value

Secure Value: □

Default Value:

Description:

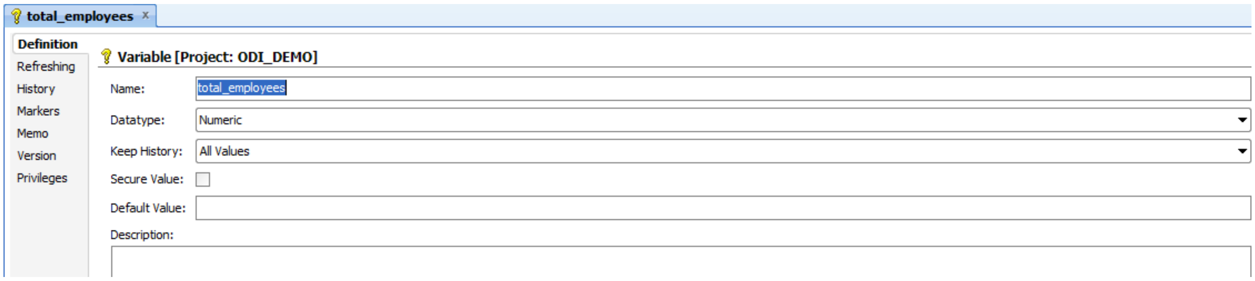


Schema: ODI_DEMO_SRC

Select Query:

```
select count(*) from DEPARTMENT
```

4. Create variable total_employees



Variable [Project: ODI_DEMO]

Name: total_employees

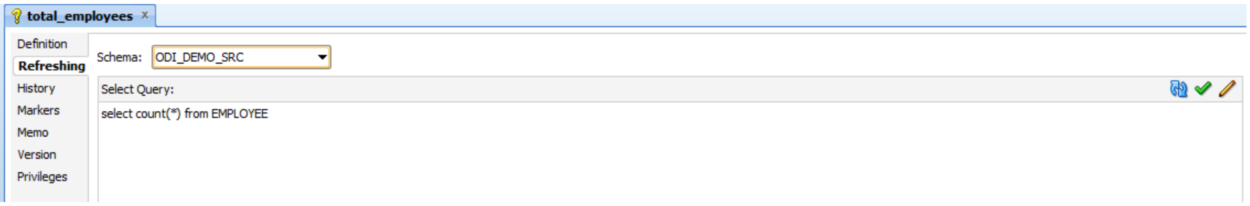
Datatype: Numeric

Keep History: All Values

Secure Value: □

Default Value:

Description:



Schema: ODI_DEMO_SRC

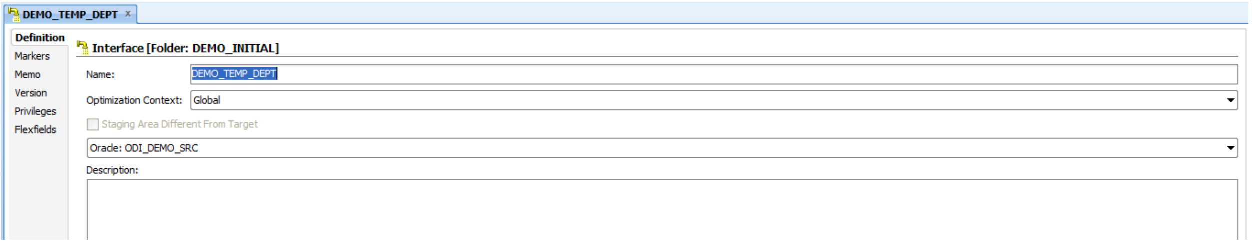
Select Query:

```
select count(*) from EMPLOYEE
```

5.4.2 Create Interfaces:

Department Interface:

1. Create Temp Interface for DEPARTMENT
   Name: 'DEMO_TMP_DEPT'
   Schema: Select source



Interface [Folder: DEMO_INITIAL]

Name: DEMO_TEMP_DEPT

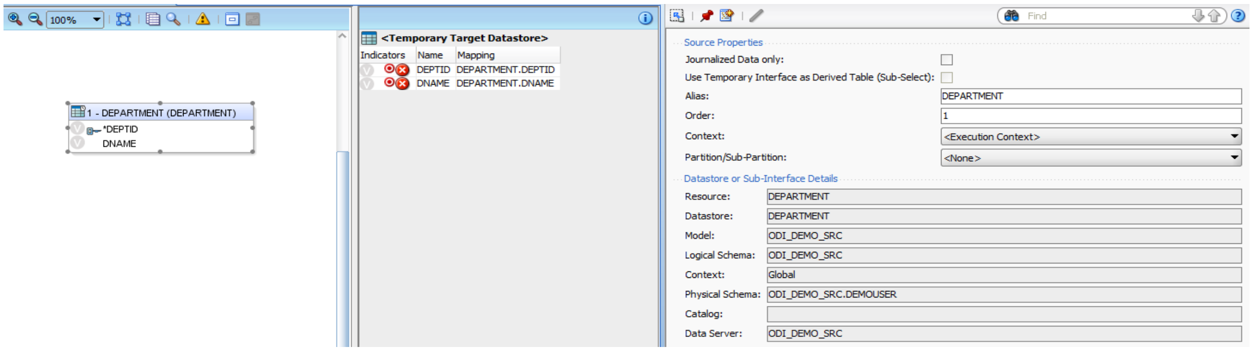
Optimization Context: Global

□ Staging Area Different From Target

Oracle: ODI_DEMO_SRC

Description:

2. Go To mapping page Drag DEPARTMENT into source then select all the columns from source then drop them into target.



<Temporary Target Datastore>

Indicators	Name	Mapping
	DEPTID	DEPARTMENT.DEPTID
	DNAME	DEPARTMENT.DNAME

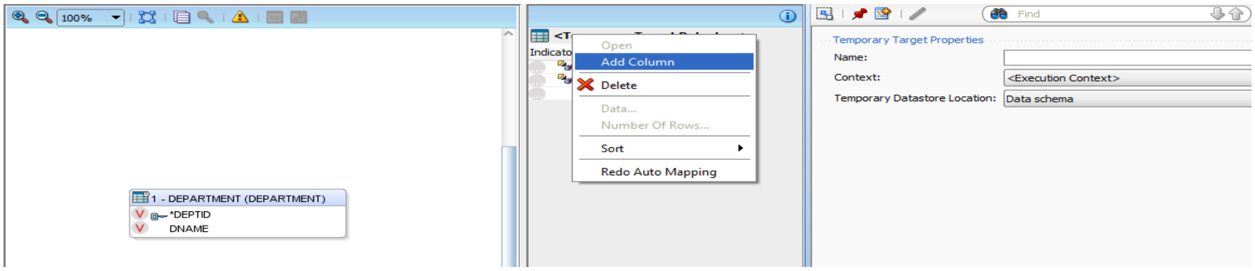
Source Properties

Journalized Data only:	□
Use Temporary Interface as Derived Table (Sub-Select):	□
Alias:	DEPARTMENT
Order:	1
Context:	<Execution Context>
Partition/Sub-Partition:	<None>

Datastore or Sub-Interface Details

Resource:	DEPARTMENT
Datastore:	DEPARTMENT
Model:	ODI_DEMO_SRC
Logical Schema:	ODI_DEMO_SRC
Context:	Global
Physical Schema:	ODI_DEMO_SRC.DEMOUSER
Catalog:	
Data Server:	ODI_DEMO_SRC

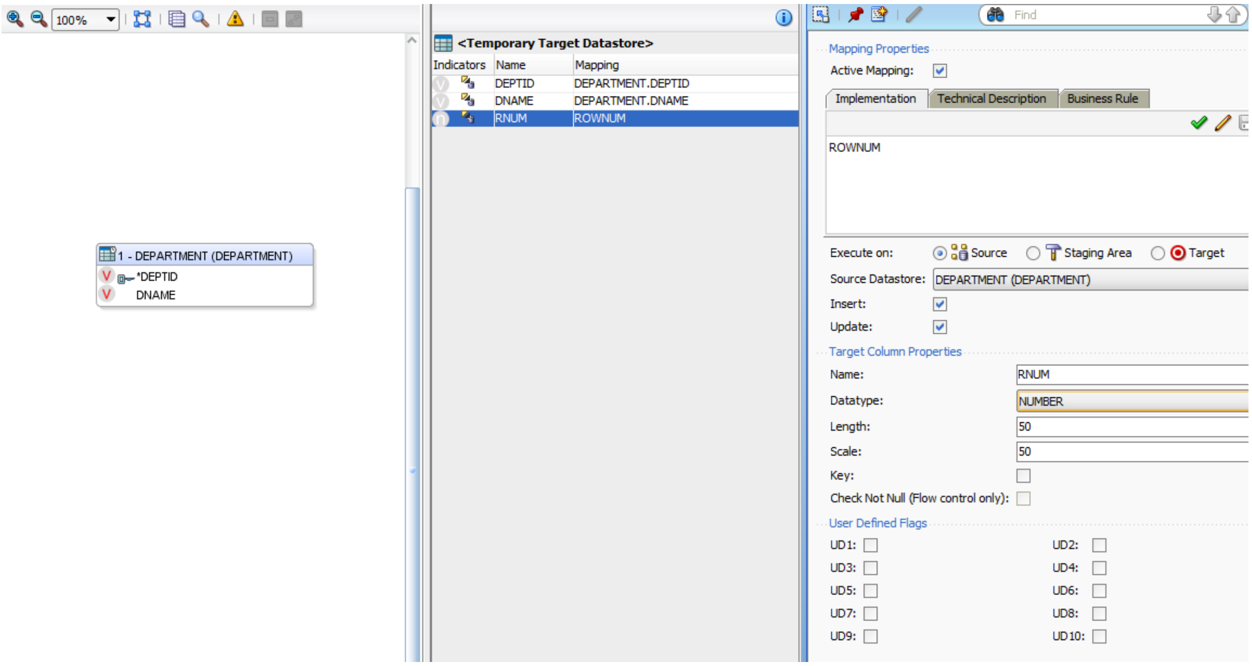
3. Select Target Data Store and Right Click then click on add column



Temporary Target Properties

Name:	
Context:	<Execution Context>
Temporary Datastore Location:	Data schema

4. Select the new column then set name, type and assign value ROWNUM to it



<Temporary Target Datastore>

Indicators	Name	Mapping
	DEPTID	DEPARTMENT.DEPTID
	DNAME	DEPARTMENT.DNAME
	RNUM	ROWNUM

Mapping Properties

Active Mapping: ☑

ROWNUM

Execute on: ○ Source ○ Staging Area ● Target

Source Datastore: DEPARTMENT (DEPARTMENT)

Insert: ☑

Update: ☑

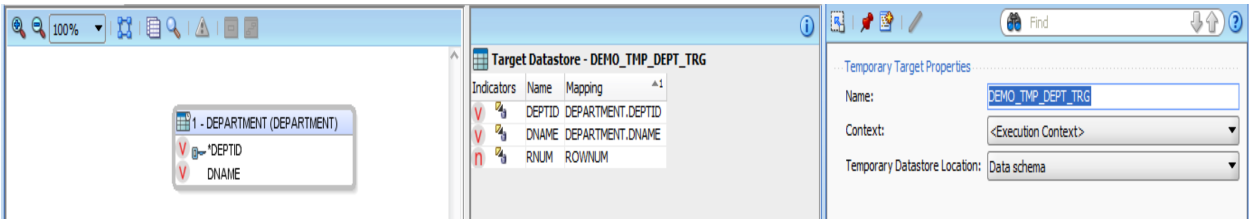
Target Column Properties

Name:	RNUM
Datatype:	NUMBER
Length:	50
Scale:	50
Key:	□
Check Not Null (Flow control only):	□

User Defined Flags

UD1:	□	UD2:	□
UD3:	□	UD4:	□
UD5:	□	UD6:	□
UD7:	□	UD8:	□
UD9:	□	UD10:	□

5. Select Target Data Store and set name DEMO_TMP_ DEPT_TRG for it



Target Datastore - DEMO_TMP_DEPT_TRG

Indicators	Name	Mapping
V	DEPTID	DEPARTMENT.DEPTID
V	DNAME	DEPARTMENT.DNAME
n	RNUM	ROWNUM

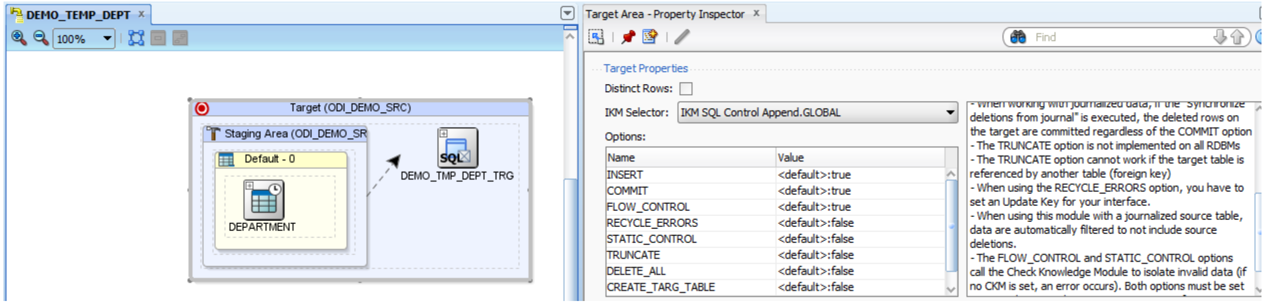
Temporary Target Properties

Name: DEMO_TMP_DEPT_TRG

Context: <Execution Context>

Temporary Datastore Location: Data schema

6. Go to flow tab and select IKM



Target Area - Property Inspector

Target Properties

Distinct Rows: □

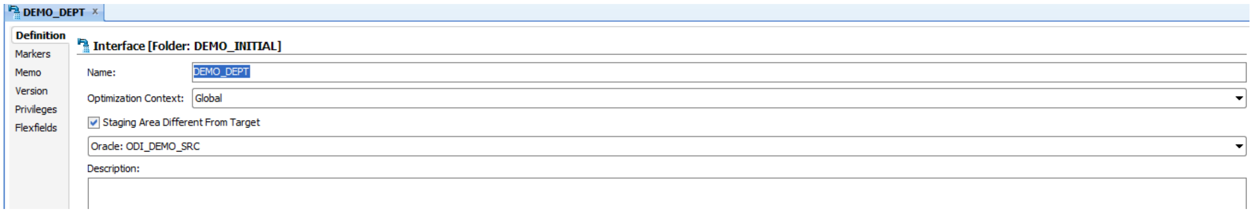
IKM Selector: IKM SQL Control Append.GLOBAL

Options:

Name	Value
INSERT	<default>:true
COMMIT	<default>:true
FLOW_CONTROL	<default>:true
RECYCLE_ERRORS	<default>:false
STATIC_CONTROL	<default>:false
TRUNCATE	<default>:false
DELETE_ALL	<default>:false
CREATE_TARG_TABLE	<default>:false

- When working with journalized data, if the "Synchronize deletions from journal" is executed, the deleted rows on the target are committed regardless of the COMMIT option
- The TRUNCATE option is not implemented on all RDBMs
- The TRUNCATE option cannot work if the target table is referenced by another table (foreign key)
- When using the RECYCLE_ERRORS option, you have to set an Update Key for your interface.
- When using this module with a journalized source table, data are automatically filtered to not include source deletions.
- The FLOW_CONTROL and STATIC_CONTROL options call the Check Knowledge Module to isolate invalid data (if no CKM is set, an error occurs). Both options must be set

7. Create interface DEMO_DEPT



Interface [Folder: DEMO_INITIAL]

Name: DEMO_DEPT

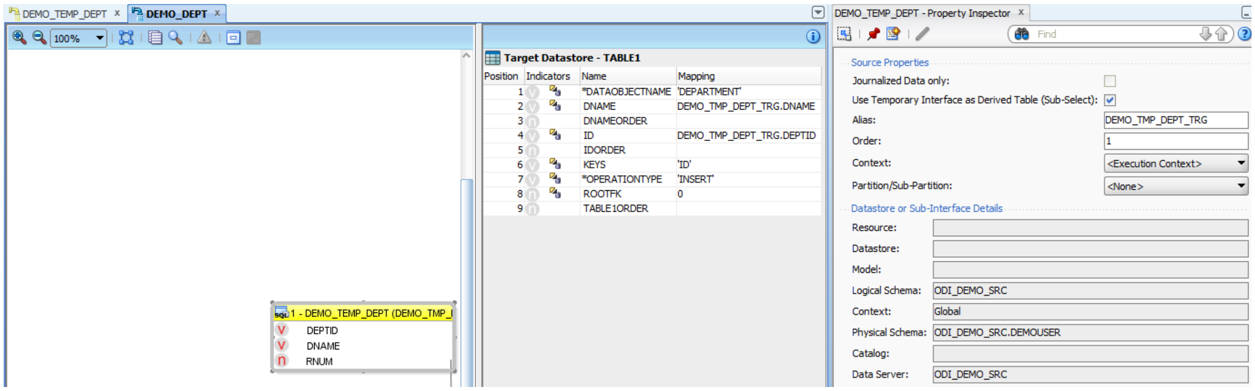
Optimization Context: Global

☑ Staging Area Different From Target

Oracle: ODI_DEMO_SRC

Description:

8. Go to mappings tab then drag the temp interface DEMO_TEMP_DEPT into source



Target Datastore - TABLE1

Position	Indicators	Name	Mapping
1	V	*DATAOBJECTNAME	'DEPARTMENT'
2	V	DNAME	DEMO_TMP_DEPT_TRG.DNAME
3	n	DNAMEORDER	
4	V	ID	DEMO_TMP_DEPT_TRG.DEPTID
5	n	IDORDER	
6	V	KEYS	'ID'
7	V	*OPERATIONTYPE	'INSERT'
8	n	ROOTFK	0
9	n	TABLE1ORDER	

DEMO_TEMP_DEPT - Property Inspector

Source Properties

Journalized Data only: □

Use Temporary Interface as Derived Table (Sub-Select): ☑

Alias: DEMO_TMP_DEPT_TRG

Order: 1

Context: <Execution Context>

Partition/Sub-Partition: <None>

Datastore or Sub-Interface Details

Resource:

Datastore:

Model:

Logical Schema: ODI_DEMO_SRC

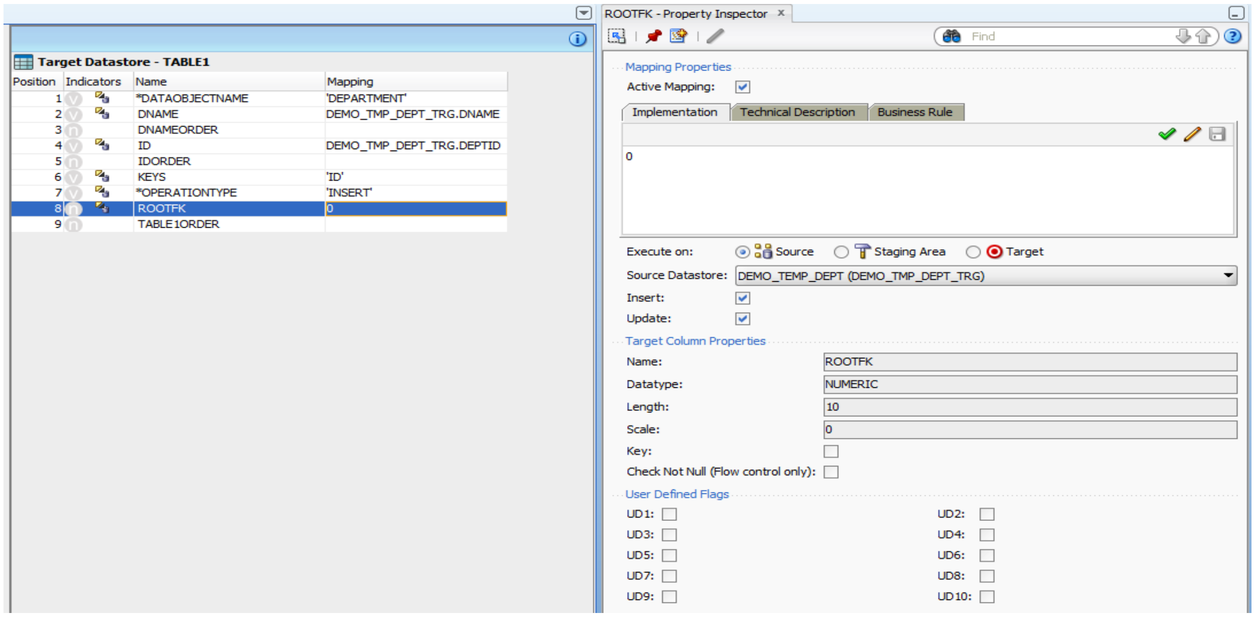
Context: Global

Physical Schema: ODI_DEMO_SRC.DEMOUSER

Catalog:

Data Server: ODI_DEMO_SRC

9. Select ROOTFK, OPERATION TYPE, KEYS and DATAOBJECT NAME on target and assign 0 and ’INSERT’, ‘ID’ and ‘DEPARTMENT’ respectively.



Target Datastore - TABLE1

Position	Indicators	Name	Mapping
1		*DATAOBJECTNAME	'DEPARTMENT'
2		DNAME	DEMO_TMP_DEPT_TRG.DNAME
3		DNAMEORDER	
4		ID	DEMO_TMP_DEPT_TRG.DEPTID
5		IDORDER	
6		KEYS	'ID'
7		*OPERATIONTYPE	'INSERT'
8		ROOTFK	0
9		TABLE1ORDER	

ROOTFK - Property Inspector

Mapping Properties

Active Mapping: ☑

Implementation | Technical Description | Business Rule

0

Execute on: ○ Source ○ Staging Area ● Target

Source Datastore: DEMO_TEMP_DEPT (DEMO_TMP_DEPT_TRG)

Insert: ☑

Update: ☑

Target Column Properties

Name: ROOTFK

Datatype: NUMERIC

Length: 10

Scale: 0

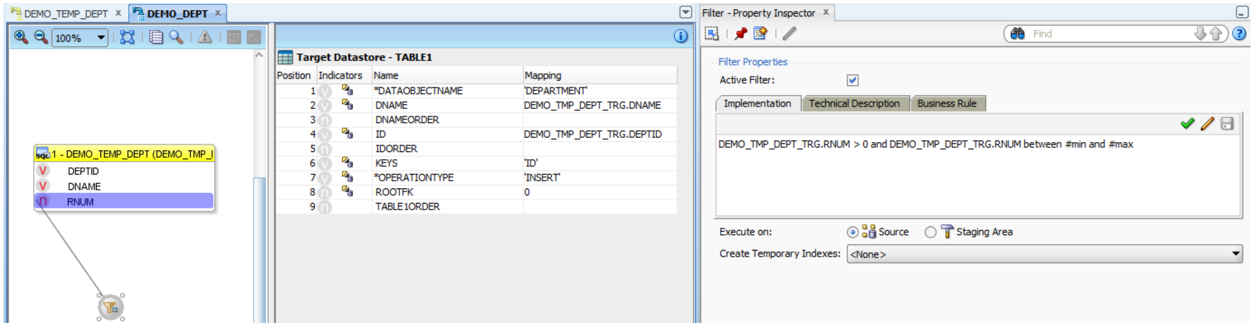
Key: □

Check Not Null (Flow control only): □

User Defined Flags

UD1: □ UD2: □
UD3: □ UD4: □
UD5: □ UD6: □
UD7: □ UD8: □
UD9: □ UD10: □

10. Create Filter on ROWNUM and update expression as DEMO_TMP_DEPT_TRG.RNUM > 0 and DEMO_TMP_DEPT_TRG.RNUM between #min and #max



Target Datastore - TABLE1

Position	Indicators	Name	Mapping
1		*DATAOBJECTNAME	'DEPARTMENT'
2		DNAME	DEMO_TMP_DEPT_TRG.DNAME
3		DNAMEORDER	
4		ID	DEMO_TMP_DEPT_TRG.DEPTID
5		IDORDER	
6		KEYS	'ID'
7		*OPERATIONTYPE	'INSERT'
8		ROOTFK	0
9		TABLE1ORDER	

Filter - Property Inspector

Filter Properties

Active Filter: ☑

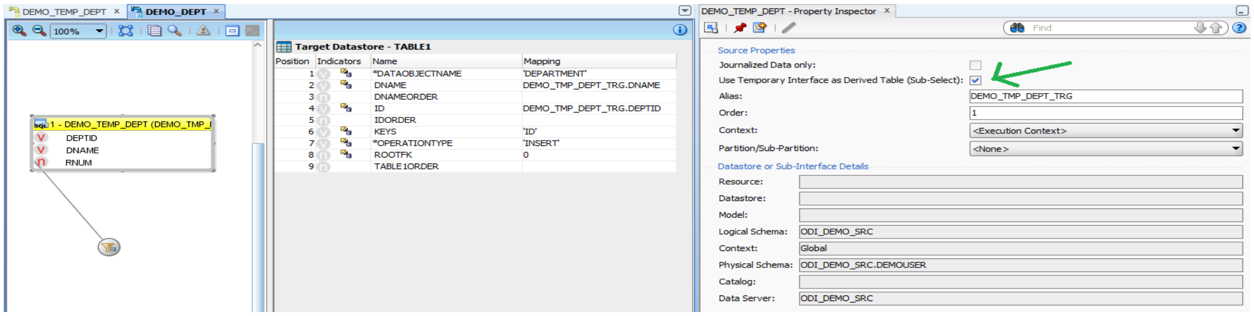
Implementation | Technical Description | Business Rule

DEMO_TMP_DEPT_TRG.RNUM > 0 and DEMO_TMP_DEPT_TRG.RNUM between #min and #max

Execute on: ● Source ○ Staging Area

Create Temporary Indexes: <None>

11. Select TEMP_DEPARTMENT from source then check option use temp interface as Derived Table



Target Datastore - TABLE1

Position	Indicators	Name	Mapping
1		*DATAOBJECTNAME	'DEPARTMENT'
2		DNAME	DEMO_TMP_DEPT_TRG.DNAME
3		DNAMEORDER	
4		ID	DEMO_TMP_DEPT_TRG.DEPTID
5		IDORDER	
6		KEYS	'ID'
7		*OPERATIONTYPE	'INSERT'
8		ROOTFK	0
9		TABLE1ORDER	

DEMO_TEMP_DEPT - Property Inspector

Source Properties

Journalized Data only: □

Use Temporary Interface as Derived Table (Sub-Select): ☑

Alias: DEMO_TMP_DEPT_TRG

Order: 1

Context: <Execution Context>

Partition/Sub-Partition: <None>

Datastore or Sub-Interface Details

Resource:

Datastore:

Model:

Logical Schema: ODI_DEMO_SRC

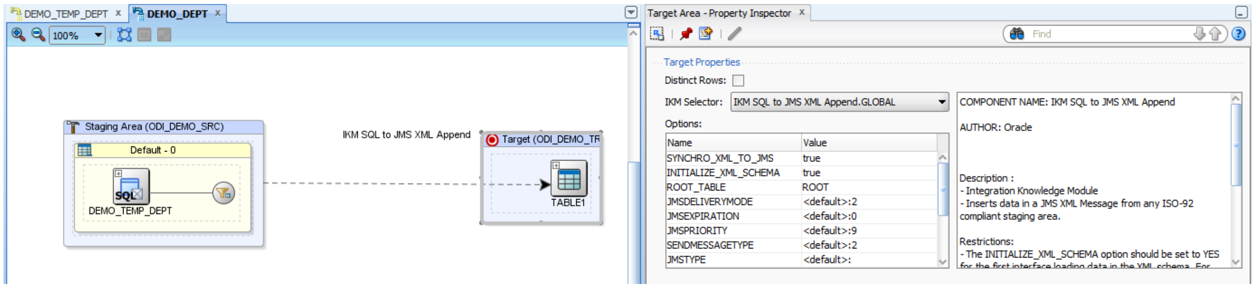
Context: Global

Physical Schema: ODI_DEMO_SRC.DEMOUSER

Catalog:

Data Server: ODI_DEMO_SRC

12. Go To Flow tab then Select IKM, update SYNCHRO_XML_TO_JMS,INITIALIZE_XML_SCHEMA to true and ROOT_TABLE to ROOT.



Target Area - Property Inspector

Target Properties

Distinct Rows: □

IKM Selector: IKM SQL to JMS XML Append.GLOBAL

Options:

Name	Value
SYNCHRO_XML_TO_JMS	true
INITIALIZE_XML_SCHEMA	true
ROOT_TABLE	ROOT
JMSDELIVERYMODE	<default>:2
JMSEXPIRATION	<default>:0
JMSPRIORITY	<default>:9
SENDMESSAGETYPE	<default>:2
JMSTYPE	<default>:

COMPONENT NAME: IKM SQL to JMS XML Append

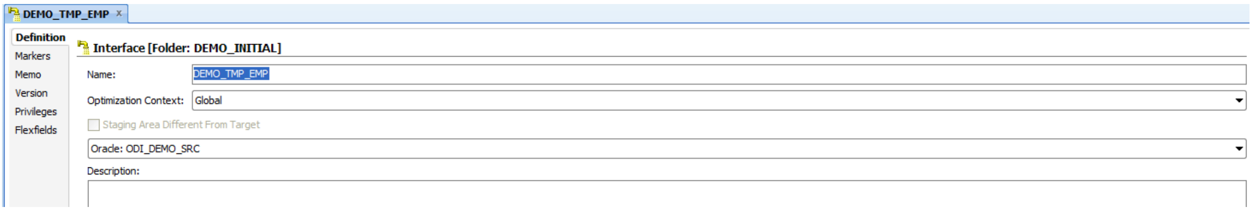
AUTHOR: Oracle

Description :
- Integration Knowledge Module
- Inserts data in a JMS XML Message from any ISO-92 compliant staging area.

Restrictions:
- The INITIALIZE_XML_SCHEMA option should be set to YES for the first interface loading data in the XML schema. For

EMPLOYEE interface:

1. Create Temp interface for EMPLOYEE
   Name: DEMO_TMP_EMP



Interface [Folder: DEMO_INITIAL]

Name: DEMO_TMP_EMP

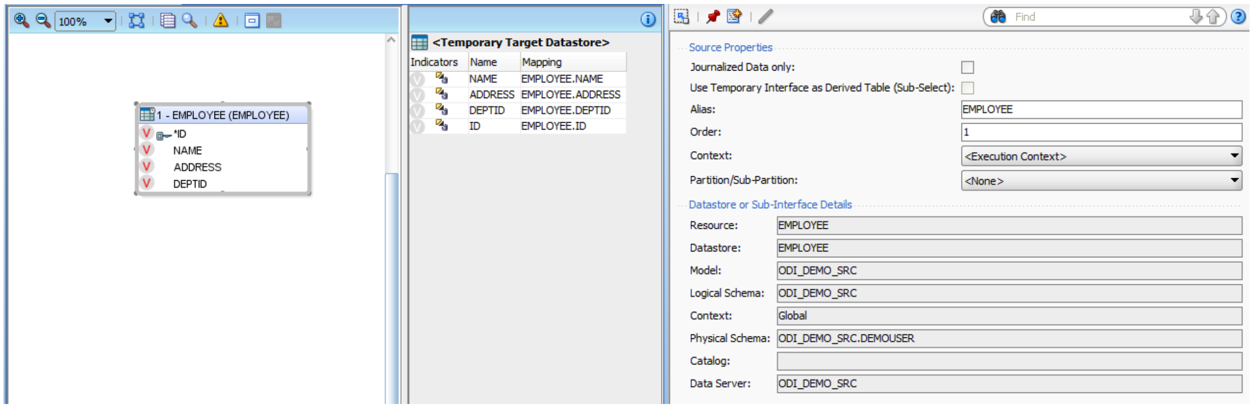
Optimization Context: Global

□ Staging Area Different From Target

Oracle: ODI_DEMO_SRC

Description:

2. Go to mappings tab drag EMPLOYEE to source then select all the columns from source then drop them into target.



<Temporary Target Datastore>

Indicators	Name	Mapping
	NAME	EMPLOYEE.NAME
	ADDRESS	EMPLOYEE.ADDRESS
	DEPTID	EMPLOYEE.DEPTID
	ID	EMPLOYEE.ID

Source Properties

Journalized Data only: □

Use Temporary Interface as Derived Table (Sub-Select): □

Alias: EMPLOYEE

Order: 1

Context: <Execution Context>

Partition/Sub-Partition: <None>

Datastore or Sub-Interface Details

Resource: EMPLOYEE

Datastore: EMPLOYEE

Model: ODI_DEMO_SRC

Logical Schema: ODI_DEMO_SRC

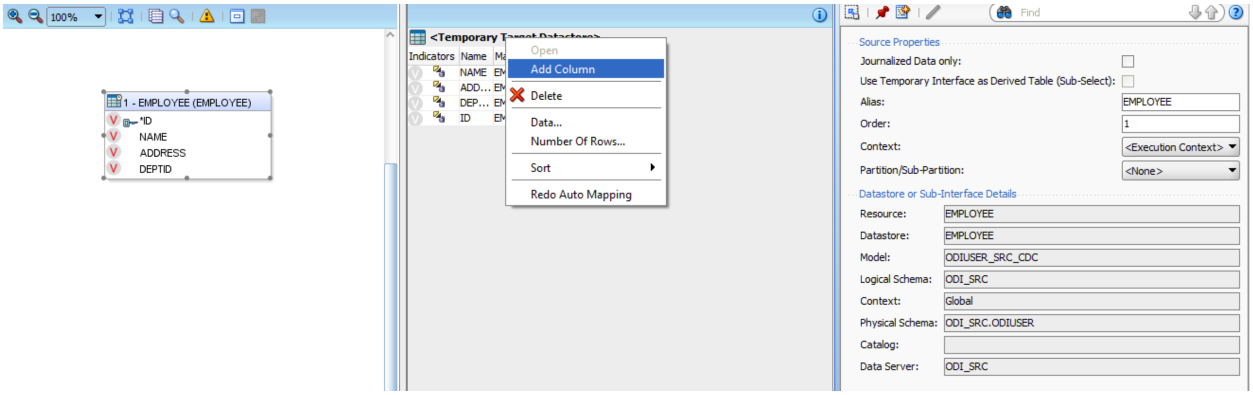
Context: Global

Physical Schema: ODI_DEMO_SRC.DEMOUSER

Catalog:

Data Server: ODI_DEMO_SRC

3. Select Target Data Store and Right Click then click on add column



Source Properties

Journalized Data only: □

Use Temporary Interface as Derived Table (Sub-Select): □

Alias: EMPLOYEE

Order: 1

Context: <Execution Context>

Partition/Sub-Partition: <None>

Datastore or Sub-Interface Details

Resource: EMPLOYEE

Datastore: EMPLOYEE

Model: ODIUSER_SRC_CDC

Logical Schema: ODI_SRC

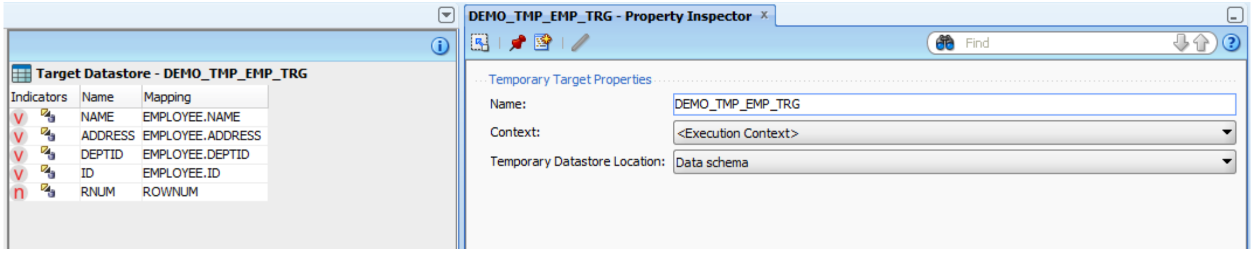
Context: Global

Physical Schema: ODI_SRC.ODIUSER

Catalog:

Data Server: ODI_SRC

4. Select Target Data Store and set name DEMO_TMP_ DEPT_TRG for it



Target Datastore - DEMO_TMP_EMP_TRG

Indicators	Name	Mapping
V	NAME	EMPLOYEE.NAME
V	ADDRESS	EMPLOYEE.ADDRESS
V	DEPTID	EMPLOYEE.DEPTID
V	ID	EMPLOYEE.ID
n	RNUM	ROWNUM

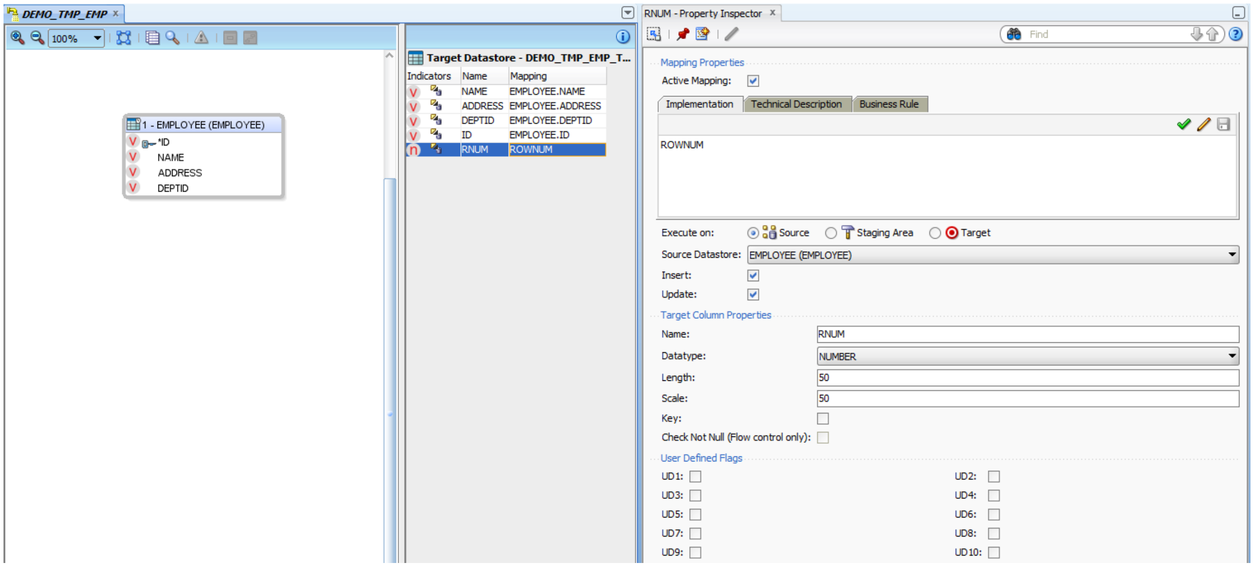
Temporary Target Properties

Name: DEMO_TMP_EMP_TRG

Context: <Execution Context>

Temporary Datastore Location: Data schema

5. Select the new column then set name, type and assign value ROWNUM to it



Target Datastore - DEMO_TMP_EMP_T...

Indicators	Name	Mapping
V	NAME	EMPLOYEE.NAME
V	ADDRESS	EMPLOYEE.ADDRESS
V	DEPTID	EMPLOYEE.DEPTID
V	ID	EMPLOYEE.ID
n	RNUM	ROWNUM

Mapping Properties

Active Mapping: ☑

ROWNUM

Execute on: ● Source ○ Staging Area ○ Target

Source Datastore: EMPLOYEE (EMPLOYEE)

Insert: ☑

Update: ☑

Target Column Properties

Name: RNUM

Datatype: NUMBER

Length: 50

Scale: 50

Key: □

Check Not Null (Flow control only): □

User Defined Flags

UD1: □ UD2: □

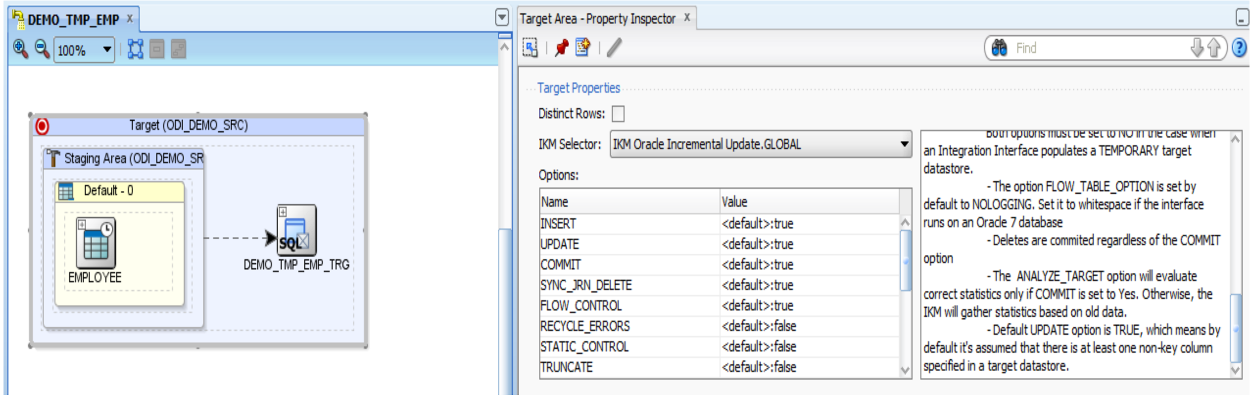
UD3: □ UD4: □

UD5: □ UD6: □

UD7: □ UD8: □

UD9: □ UD10: □

6. Go to Flow tab and Select IKM



Target Properties

Distinct Rows: □

IKM Selector: IKM Oracle Incremental Update.GLOBAL

Options:

Name	Value
INSERT	<default>:true
UPDATE	<default>:true
COMMIT	<default>:true
SYNC_JRN_DELETE	<default>:true
FLOW_CONTROL	<default>:true
RECYCLE_ERRORS	<default>:false
STATIC_CONTROL	<default>:false
TRUNCATE	<default>:false

Both options must be set to NO in the case when an Integration Interface populates a TEMPORARY target datastore.

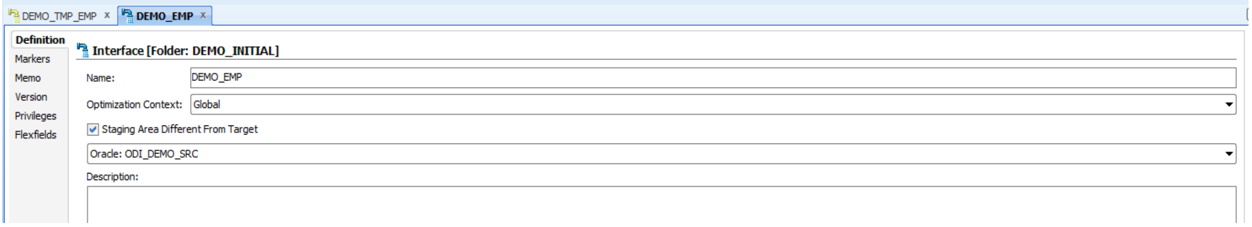
- The option FLOW_TABLE_OPTION is set by default to NOLOGGING. Set it to whitespace if the interface runs on an Oracle 7 database

- Deletes are commited regardless of the COMMIT option

- The ANALYZE_TARGET option will evaluate correct statistics only if COMMIT is set to Yes. Otherwise, the IKM will gather statistics based on old data.

- Default UPDATE option is TRUE, which means by default it's assumed that there is at least one non-key column specified in a target datastore.

7. Create Interface DEMP_EMP



Interface [Folder: DEMO_INITIAL]

Name: DEMO_EMP

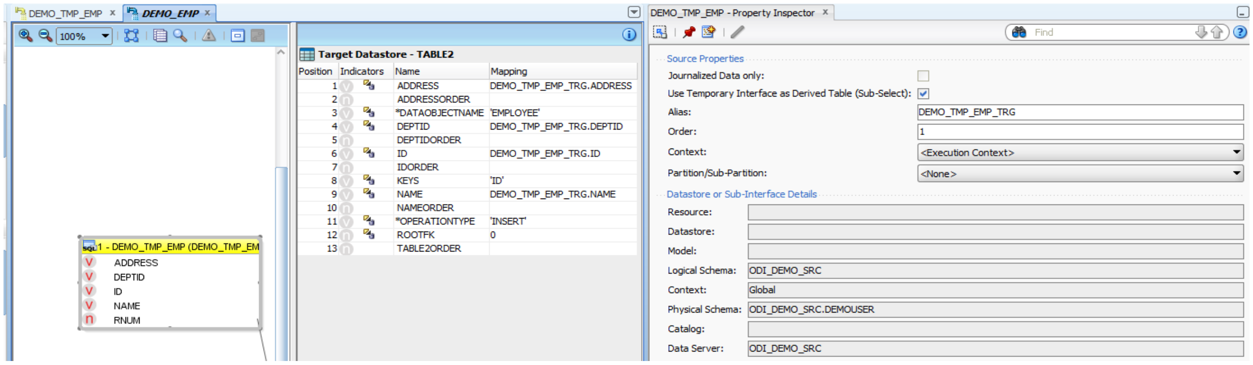
Optimization Context: Global

☑ Staging Area Different From Target

Oracle: ODI_DEMO_SRC

Description:

8. Go to mappings tab drag DEMO_TMP_EMP in to source


DEMO_TMP_EMP
DEMO_EMP
100%
1 - DEMO_TMP_EMP (DEMO_TMP_EM
ADDRESS
DEPTID
ID
NAME
RNUM
DEMO_TMP_EMP - Property Inspector
Find

Target Datastore - TABLE2

Position	Indicators	Name	Mapping
1	V	ADDRESS	DEMO_TMP_EMP_TRG.ADDRESS
2	n	ADDRESSORDER	
3	V	*DATAOBJECTNAME	'EMPLOYEE'
4	V	DEPTID	DEMO_TMP_EMP_TRG.DEPTID
5	n	DEPTIDORDER	
6	V	ID	DEMO_TMP_EMP_TRG.ID
7	n	IDORDER	
8	V	KEYS	'ID'
9	V	NAME	DEMO_TMP_EMP_TRG.NAME
10	n	NAMEORDER	
11	V	*OPERATIONTYPE	'INSERT'
12	n	ROOTFK	0
13	n	TABLE2ORDER	

Source Properties

Journalized Data only: □

Use Temporary Interface as Derived Table (Sub-Select): ☑

Alias: DEMO_TMP_EMP_TRG

Order: 1

Context: <Execution Context>

Partition/Sub-Partition: <None>

Datastore or Sub-Interface Details

Resource:

Datastore:

Model:

Logical Schema: ODI_DEMO_SRC

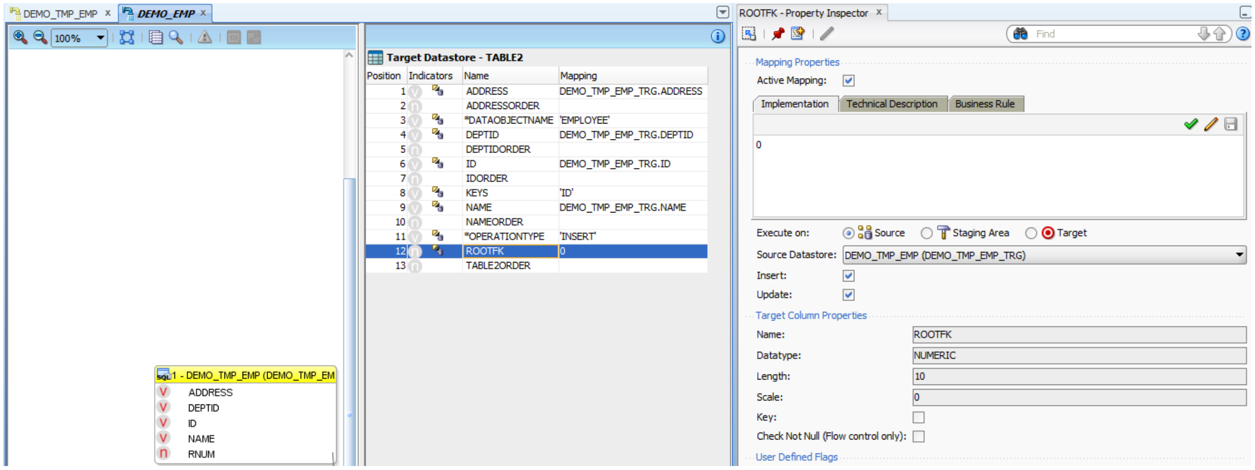
Context: Global

Physical Schema: ODI_DEMO_SRC.DEMOUSER

Catalog:

Data Server: ODI_DEMO_SRC

9. Select ROOTFK, OPERATION TYPE,DATAOBJECT NAME and KEYS on target and assign 0 and ’INSERT’, ‘EMPLOYEE’ and ’ID’ respectively



Target Datastore - TABLE2

Position	Indicators	Name	Mapping
1	V	ADDRESS	DEMO_TMP_EMP_TRG.ADDRESS
2	n	ADDRESSORDER	
3	V	*DATAOBJECTNAME	'EMPLOYEE'
4	V	DEPTID	DEMO_TMP_EMP_TRG.DEPTID
5	n	DEPTIDORDER	
6	V	ID	DEMO_TMP_EMP_TRG.ID
7	n	IDORDER	
8	V	KEYS	'ID'
9	V	NAME	DEMO_TMP_EMP_TRG.NAME
10	n	NAMEORDER	
11	V	*OPERATIONTYPE	'INSERT'
12	n	ROOTFK	0
13	n	TABLE2ORDER	

ROOTFK - Property Inspector

Mapping Properties

Active Mapping: ☑

Implementation | Technical Description | Business Rule

0

Execute on: ○ Source ○ Staging Area ● Target

Source Datastore: DEMO_TMP_EMP (DEMO_TMP_EMP_TRG)

Insert: ☑

Update: ☑

Target Column Properties

Name: ROOTFK

Datatype: NUMERIC

Length: 10

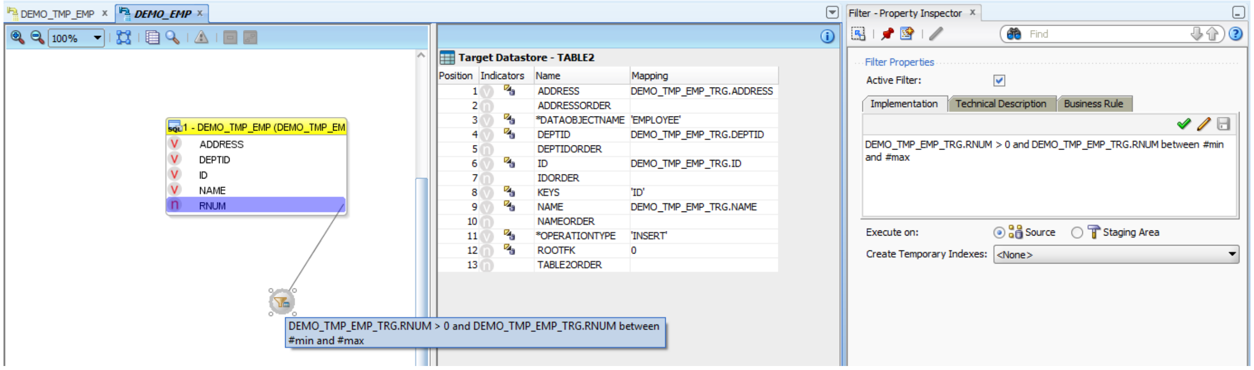
Scale: 0

Key: □

Check Not Null (Flow control only): □

User Defined Flags

10. Create Filter on ROWNUM and update expression as DEMO_TMP_EMP_TRG.RNUM > 0 and DEMO_TMP_EMP_TRG.RNUM between #min and #max


DEMO_TMP_EMP
DEMO_EMP
100%
1 - DEMO_TMP_EMP (DEMO_TMP_EM
ADDRESS
DEPTID
ID
NAME
RNUM
DEMO_TMP_EMP_TRG.RNUM > 0 and DEMO_TMP_EMP_TRG.RNUM between #min and #max
Find

Target Datastore - TABLE2

Position	Indicators	Name	Mapping
1	V	ADDRESS	DEMO_TMP_EMP_TRG.ADDRESS
2	n	ADDRESSORDER	
3	V	*DATAOBJECTNAME	'EMPLOYEE'
4	V	DEPTID	DEMO_TMP_EMP_TRG.DEPTID
5	n	DEPTIDORDER	
6	V	ID	DEMO_TMP_EMP_TRG.ID
7	n	IDORDER	
8	V	KEYS	'ID'
9	V	NAME	DEMO_TMP_EMP_TRG.NAME
10	n	NAMEORDER	
11	V	*OPERATIONTYPE	'INSERT'
12	n	ROOTFK	0
13	n	TABLE2ORDER	

Filter - Property Inspector

Filter Properties

Active Filter: ☑

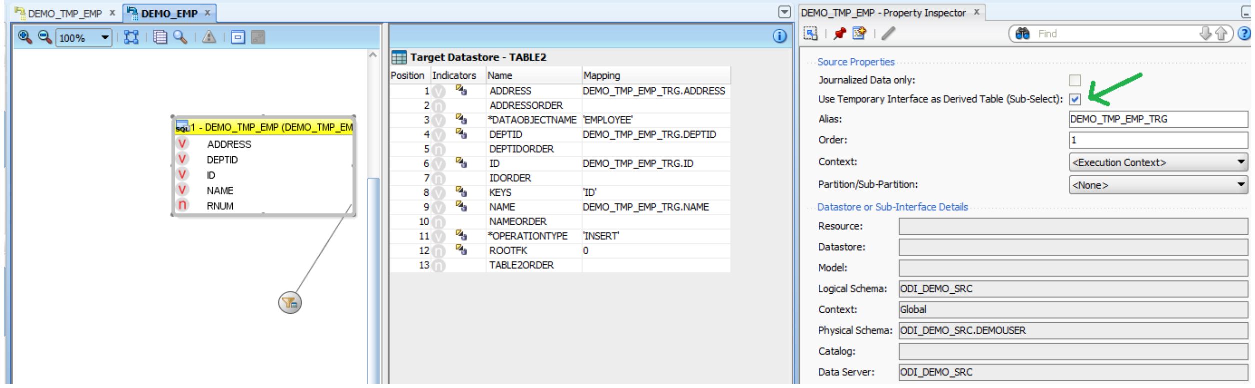
Implementation | Technical Description | Business Rule

DEMO_TMP_EMP_TRG.RNUM > 0 and DEMO_TMP_EMP_TRG.RNUM between #min and #max

Execute on: ● Source ○ Staging Area

Create Temporary Indexes: <None>

11. Select TEMP_EMPLOYEE from source then check option use temp interface as Derived Table



Target Datastore - TABLE2

Position	Indicators	Name	Mapping
1	V	ADDRESS	DEMO_TMP_EMP_TRG.ADDRESS
2	n	ADDRESSORDER	
3	V	*DATAOBJECTNAME	'EMPLOYEE'
4	V	DEPTID	DEMO_TMP_EMP_TRG.DEPTID
5	n	DEPTIDORDER	
6	V	ID	DEMO_TMP_EMP_TRG.ID
7	n	IDORDER	
8	V	KEYS	'ID'
9	V	NAME	DEMO_TMP_EMP_TRG.NAME
10	n	NAMEORDER	
11	V	*OPERATIONTYPE	'INSERT'
12	n	ROOTFK	0
13	n	TABLE2ORDER	

DEMO_TMP_EMP - Property Inspector

Source Properties

Journalized Data only: □

Use Temporary Interface as Derived Table (Sub-Select): ☑

Alias: DEMO_TMP_EMP_TRG

Order: 1

Context: <Execution Context>

Partition/Sub-Partition: <None>

Datastore or Sub-Interface Details

Resource:

Datastore:

Model:

Logical Schema: ODI_DEMO_SRC

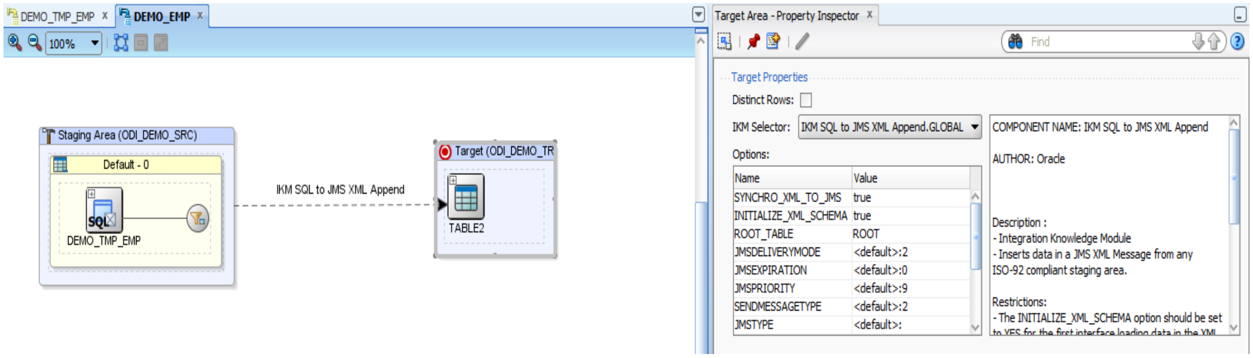
Context: Global

Physical Schema: ODI_DEMO_SRC.DEMOUSER

Catalog:

Data Server: ODI_DEMO_SRC

12. Go To Flow tab then Select IKM, update SYNCHRO_XML_TO_JMS,INITIALIZE_XML_SCHEMA to true and ROOT_TABLE to ROOT.


DEMO_TMP_EMP
DEMO_EMP
100%
Staging Area (ODI_DEMO_SRC)
Default - 0
SQL
DEMO_TMP_EMP
IKM SQL to JMS XML Append
Target (ODI_DEMO_TR
TABLE2
Target Area - Property Inspector
Find

Target Properties

Distinct Rows: □

IKM Selector: IKM SQL to JMS XML Append.GLOBAL

Options:

Name	Value
SYNCHRO_XML_TO_JMS	true
INITIALIZE_XML_SCHEMA	true
ROOT_TABLE	ROOT
JMSDELIVERYMODE	<default>:2
JMSEXPIRATION	<default>:0
JMSPRIORITY	<default>:9
SENDMESSAGETYPE	<default>:2
JMSTYPE	<default>:

COMPONENT NAME: IKM SQL to JMS XML Append

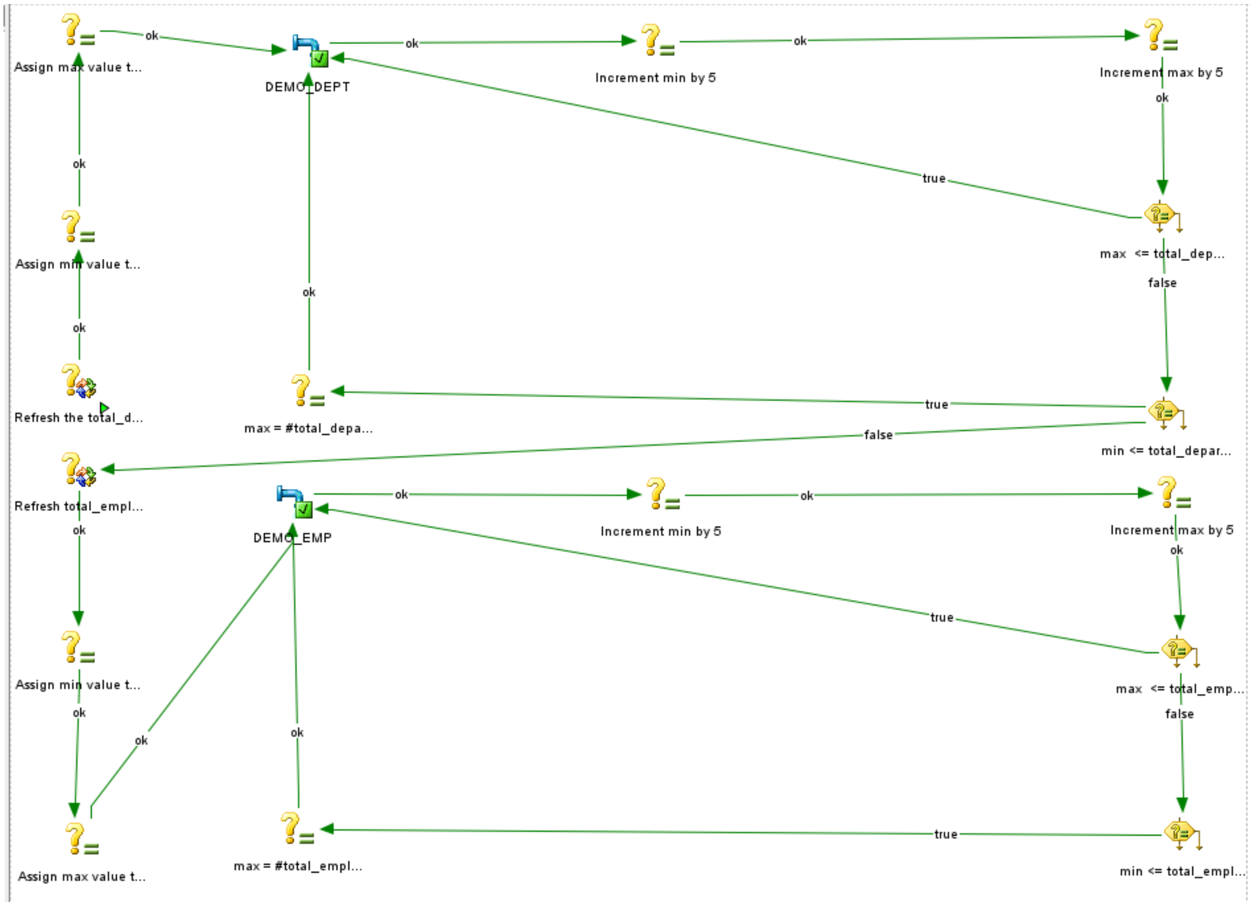
AUTHOR: Oracle

Description :
- Integration Knowledge Module
- Inserts data in a JMS XML Message from any ISO-92 compliant staging area.

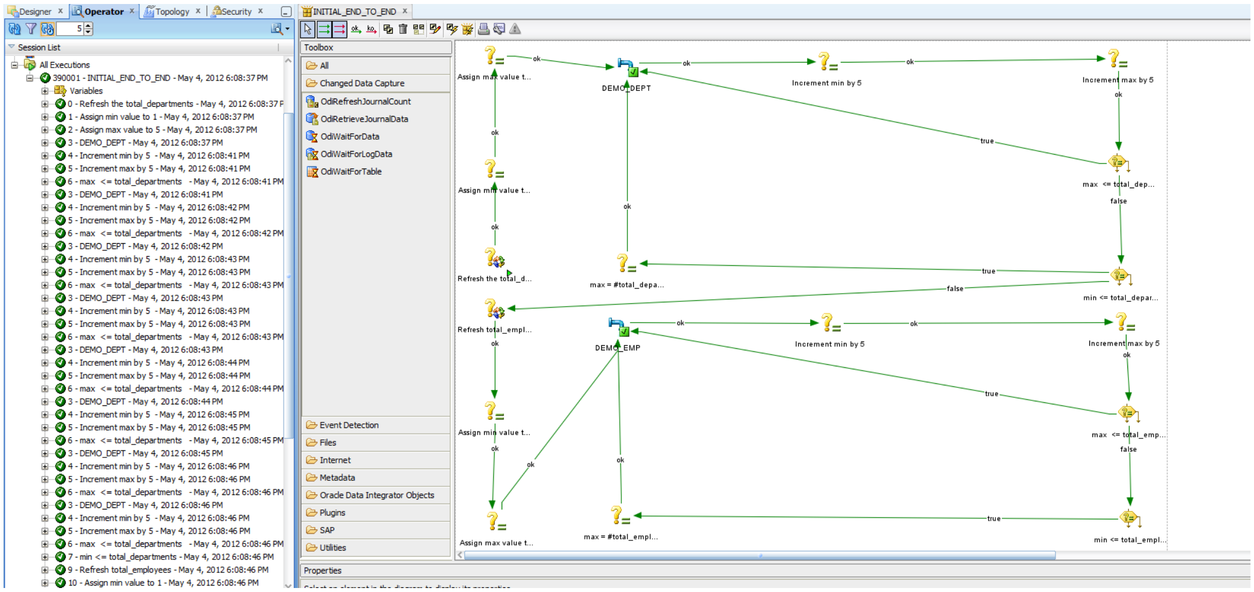
Restrictions:
- The INITIALIZE_XML_SCHEMA option should be set to YES for the first interface loading data in the XML

5.4.3 Create Package flow with defined variables and interfaces

1. Refresh the total_departments
2. Assign min value to 1
3. Assign max value to 5 {batch size}
4. Add interface DEMO_DEPT
5. Increment min by 5 {batch size}
6. Increment max by 5 {batch size}
7. Repeat step 4 if max <= total_departments
8. Check if min <= total_departments then assign max = #total_departments then Repeat step 4
9. Refresh total_employees
10. Assign min value to 1
11. Assign max value to 5 {batch size}
12. Add Interface DEMO_EMP
13. Increment min by 5 {batch size}
14. Increment max by 5 {batch size}
15. Repeat step 4 if max <= total_employees
16. Check if min <= total_employees then assign max = #total_employees then Repeat step 4

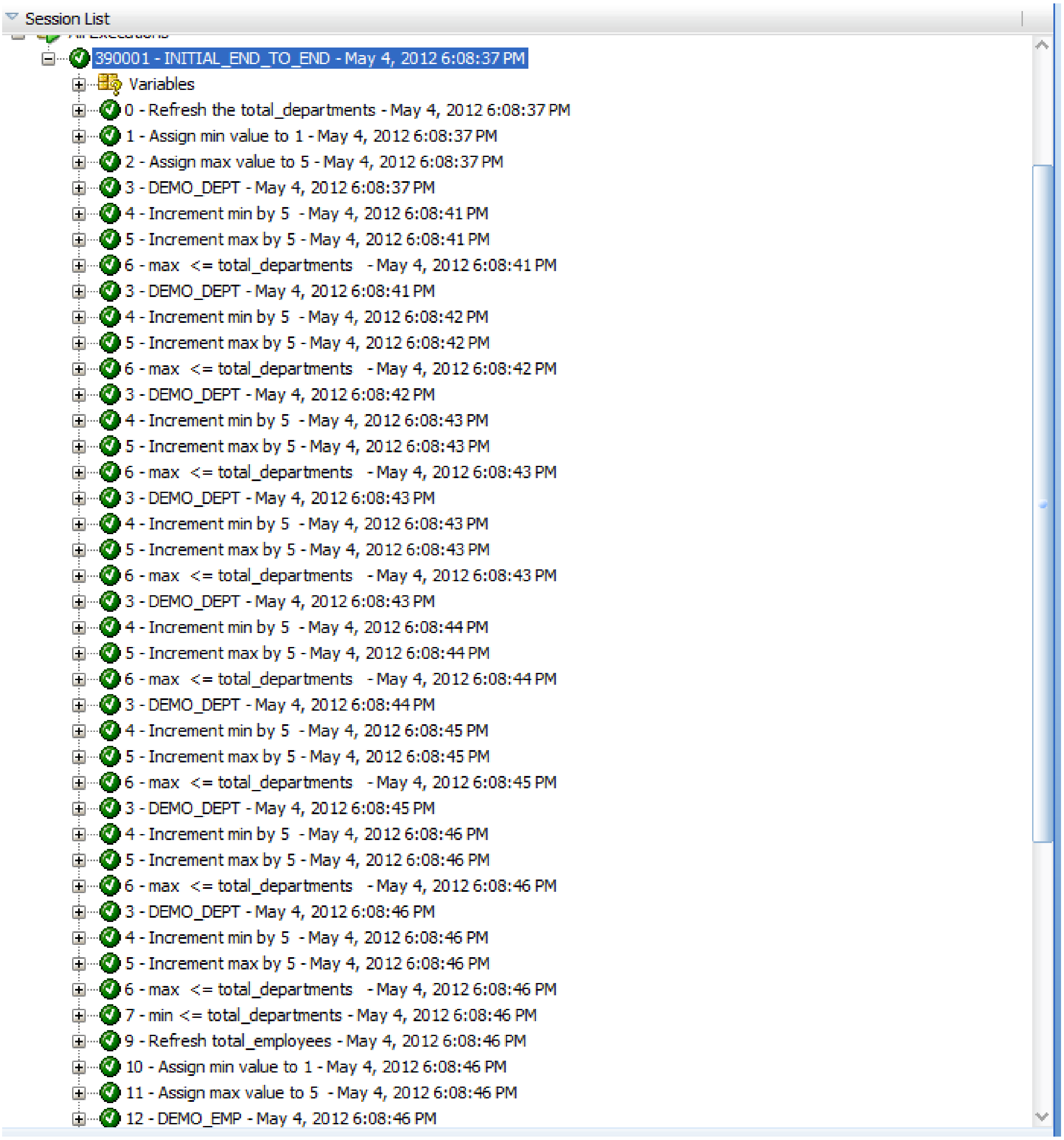


Execute Package: Execute package by clicking on ▶



```
All Executions
  390001 - INITIAL_END_TO_END - May 4, 2012 6:08:37 PM
    Variables
    0 - Refresh the total_departments - May 4, 2012 6:08:37 P
    1 - Assign min value to 1 - May 4, 2012 6:08:37 PM
    2 - Assign max value to 5 - May 4, 2012 6:08:37 PM
    3 - DEMO_DEPT - May 4, 2012 6:08:37 PM
    4 - Increment min by 5  - May 4, 2012 6:08:41 PM
    5 - Increment max by 5 - May 4, 2012 6:08:41 PM
    6 - max  <= total_departments   - May 4, 2012 6:08:41 PM
    3 - DEMO_DEPT - May 4, 2012 6:08:41 PM
    4 - Increment min by 5  - May 4, 2012 6:08:42 PM
    5 - Increment max by 5 - May 4, 2012 6:08:42 PM
    6 - max  <= total_departments   - May 4, 2012 6:08:42 PM
    3 - DEMO_DEPT - May 4, 2012 6:08:42 PM
    4 - Increment min by 5  - May 4, 2012 6:08:43 PM
    5 - Increment max by 5 - May 4, 2012 6:08:43 PM
    6 - max  <= total_departments   - May 4, 2012 6:08:43 PM
    3 - DEMO_DEPT - May 4, 2012 6:08:43 PM
    4 - Increment min by 5  - May 4, 2012 6:08:43 PM
    5 - Increment max by 5 - May 4, 2012 6:08:43 PM
    6 - max  <= total_departments   - May 4, 2012 6:08:43 PM
    3 - DEMO_DEPT - May 4, 2012 6:08:43 PM
    4 - Increment min by 5  - May 4, 2012 6:08:44 PM
    5 - Increment max by 5 - May 4, 2012 6:08:44 PM
    6 - max  <= total_departments   - May 4, 2012 6:08:44 PM
    3 - DEMO_DEPT - May 4, 2012 6:08:44 PM
    4 - Increment min by 5  - May 4, 2012 6:08:45 PM
    5 - Increment max by 5 - May 4, 2012 6:08:45 PM
    6 - max  <= total_departments   - May 4, 2012 6:08:45 PM
    3 - DEMO_DEPT - May 4, 2012 6:08:45 PM
    4 - Increment min by 5  - May 4, 2012 6:08:46 PM
    5 - Increment max by 5 - May 4, 2012 6:08:46 PM
    6 - max  <= total_departments   - May 4, 2012 6:08:46 PM
    3 - DEMO_DEPT - May 4, 2012 6:08:46 PM
    4 - Increment min by 5  - May 4, 2012 6:08:46 PM
    5 - Increment max by 5 - May 4, 2012 6:08:46 PM
    6 - max  <= total_departments   - May 4, 2012 6:08:46 PM
    7 - min <= total_departments - May 4, 2012 6:08:46 PM
    9 - Refresh total_employees - May 4, 2012 6:08:46 PM
    10 - Assign min value to 1 - May 4, 2012 6:08:46 PM
```

We can see the execution results in Operator tab like below



```
390001 - INITIAL_END_TO_END - May 4, 2012 6:08:37 PM
    Variables
    0 - Refresh the total_departments - May 4, 2012 6:08:37 PM
    1 - Assign min value to 1 - May 4, 2012 6:08:37 PM
    2 - Assign max value to 5 - May 4, 2012 6:08:37 PM
    3 - DEMO_DEPT - May 4, 2012 6:08:37 PM
    4 - Increment min by 5  - May 4, 2012 6:08:41 PM
    5 - Increment max by 5 - May 4, 2012 6:08:41 PM
    6 - max  <= total_departments   - May 4, 2012 6:08:41 PM
    3 - DEMO_DEPT - May 4, 2012 6:08:41 PM
    4 - Increment min by 5  - May 4, 2012 6:08:42 PM
    5 - Increment max by 5 - May 4, 2012 6:08:42 PM
    6 - max  <= total_departments   - May 4, 2012 6:08:42 PM
    3 - DEMO_DEPT - May 4, 2012 6:08:42 PM
    4 - Increment min by 5  - May 4, 2012 6:08:43 PM
    5 - Increment max by 5 - May 4, 2012 6:08:43 PM
    6 - max  <= total_departments   - May 4, 2012 6:08:43 PM
    3 - DEMO_DEPT - May 4, 2012 6:08:43 PM
    4 - Increment min by 5  - May 4, 2012 6:08:43 PM
    5 - Increment max by 5 - May 4, 2012 6:08:43 PM
    6 - max  <= total_departments   - May 4, 2012 6:08:43 PM
    3 - DEMO_DEPT - May 4, 2012 6:08:43 PM
    4 - Increment min by 5  - May 4, 2012 6:08:44 PM
    5 - Increment max by 5 - May 4, 2012 6:08:44 PM
    6 - max  <= total_departments   - May 4, 2012 6:08:44 PM
    3 - DEMO_DEPT - May 4, 2012 6:08:44 PM
    4 - Increment min by 5  - May 4, 2012 6:08:45 PM
    5 - Increment max by 5 - May 4, 2012 6:08:45 PM
    6 - max  <= total_departments   - May 4, 2012 6:08:45 PM
    3 - DEMO_DEPT - May 4, 2012 6:08:45 PM
    4 - Increment min by 5  - May 4, 2012 6:08:46 PM
    5 - Increment max by 5 - May 4, 2012 6:08:46 PM
    6 - max  <= total_departments   - May 4, 2012 6:08:46 PM
    3 - DEMO_DEPT - May 4, 2012 6:08:46 PM
    4 - Increment min by 5  - May 4, 2012 6:08:46 PM
    5 - Increment max by 5 - May 4, 2012 6:08:46 PM
    6 - max  <= total_departments   - May 4, 2012 6:08:46 PM
    7 - min <= total_departments - May 4, 2012 6:08:46 PM
    9 - Refresh total_employees - May 4, 2012 6:08:46 PM
    10 - Assign min value to 1 - May 4, 2012 6:08:46 PM
    11 - Assign max value to 5  - May 4, 2012 6:08:46 PM
    12 - DEMO_EMP - May 4, 2012 6:08:46 PM
```

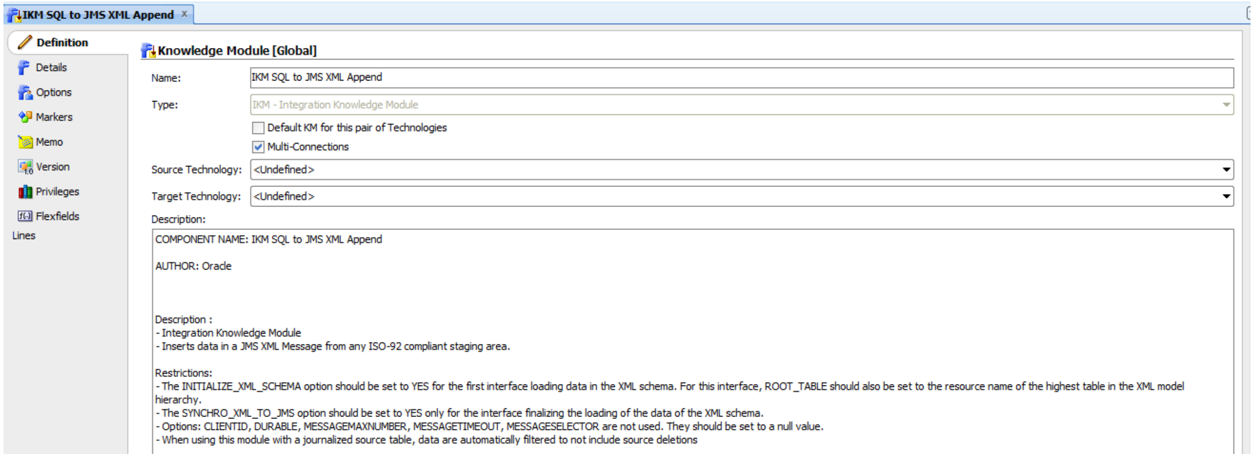
Please refer to below link for ODI Documentation for more information

http://docs.oracle.com/cd/E23943_01/integrate.1111/e12644/jms_xml.htm#CIHDCIFC

6. CDC

6.1 Customize IKM

Copy and Customize IKM to notify Target Data Source about Deletes: Go to Global Objects and then Global Knowledge Modules → Integration KM then click on IKM SQL to JMS XML Append .



Name: IKM SQL to JMS XML Append

Type: IKM - Integration Knowledge Module

□ Default KM for this pair of Technologies

☑ Multi-Connections

Source Technology: <Undefined>

Target Technology: <Undefined>

Description:

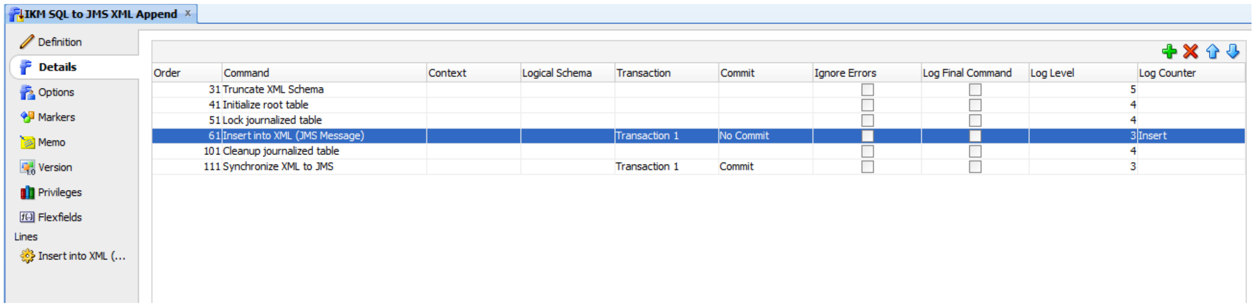
COMPONENT NAME: IKM SQL to JMS XML Append

AUTHOR: Oracle

Description :
- Integration Knowledge Module
- Inserts data in a JMS XML Message from any ISO-92 compliant staging area.

Restrictions:
- The INITIALIZE_XML_SCHEMA option should be set to YES for the first interface loading data in the XML schema. For this interface, ROOT_TABLE should also be set to the resource name of the highest table in the XML model hierarchy.
- The SYNCHRO_XML_TO_JMS option should be set to YES only for the interface finalizing the loading of the data of the XML schema.
- Options: CLIENTID, DURABLE, MESSAGEMAXNUMBER, MESSAGETIMEOUT, MESSAGESELECTOR are not used. They should be set to a null value.
- When using this module with a journalized source table, data are automatically filtered to not include source deletions

Go to Details tab then select Insert into XML

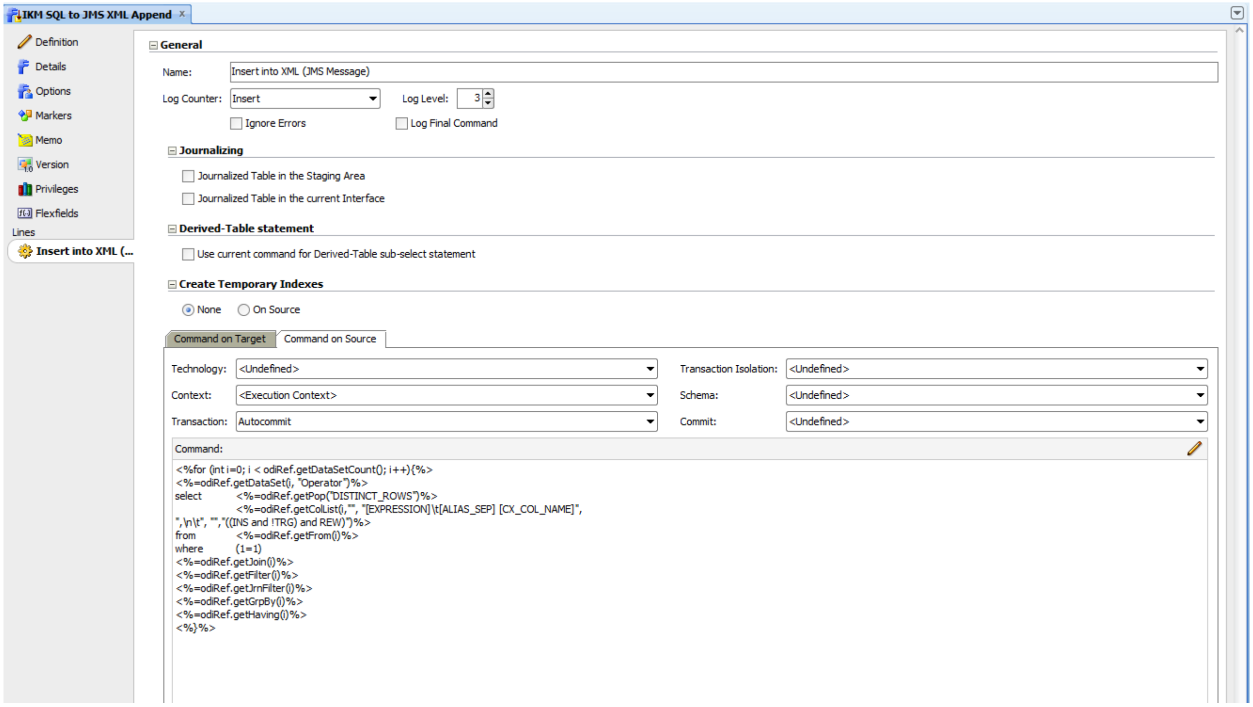


Order	Command	Context	Logical Schema	Transaction	Commit	Ignore Errors	Log Final Command	Log Level	Log Counter
31	Truncate XML Schema					□	□	5	
41	Initialize root table					□	□	4	
51	Lock journalized table					□	□	4	
61	Insert into XML (JMS Message)			Transaction 1	No Commit	□	□	3	Insert
101	Cleanup journalized table					□	□	4	
111	Synchronize XML to JMS			Transaction 1	Commit	□	□	3	

Modify the code on Source.

Remove following code from where clause and use this IKM in all the corresponding interfaces.

```
<% if (odiRef.getDataSet(i, "HAS_JRN").equals("1")) { %>

            JRN_FLAG <> 'D'

            <%} else {%>
```



General

Name: Insert into XML (JMS Message)

Log Counter: Insert

Log Level: 3

□ Ignore Errors □ Log Final Command

Journalizing

□ Journalized Table in the Staging Area

□ Journalized Table in the current Interface

Derived-Table statement

□ Use current command for Derived-Table sub-select statement

Create Temporary Indexes

● None ○ On Source

Command on Target | Command on Source

Technology: <Undefined>

Transaction Isolation: <Undefined>

Context: <Execution Context>

Schema: <Undefined>

Transaction: Autocommit

Commit: <Undefined>

Command:

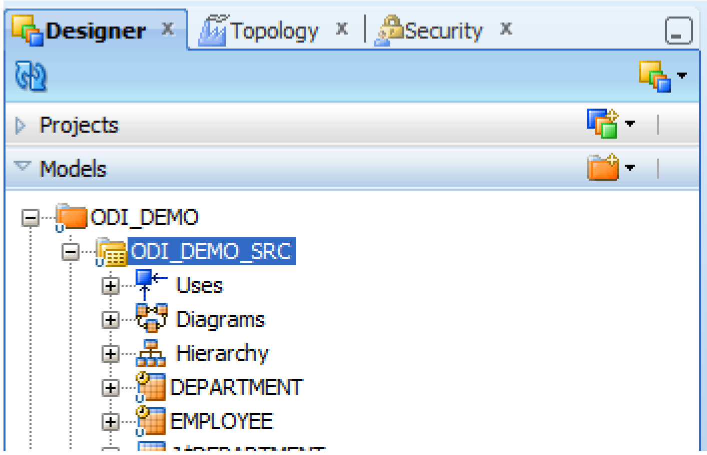
```
<%for (int i=0; i < odiRef.getDataSetCount(); i++){%>
<%=odiRef.getDataSet(i, "Operator")%>
select          <%=odiRef.getPop("DISTINCT_ROWS")%>
                <%=odiRef.getColList(i,"", "[EXPRESSION]\t[ALIAS_SEP] [CX_COL_NAME]",
",\n\t", "","((INS and !TRG) and REW)")%>
from            <%=odiRef.getFrom(i)%>
where           (1=1)
<%=odiRef.getJoin(i)%>
<%=odiRef.getFilter(i)%>
<%=odiRef.getJrnFilter(i)%>
<%=odiRef.getGrpBy(i)%>
<%=odiRef.getHaving(i)%>
<%}%>
```

6.2 Modify Source Data Store for CDC

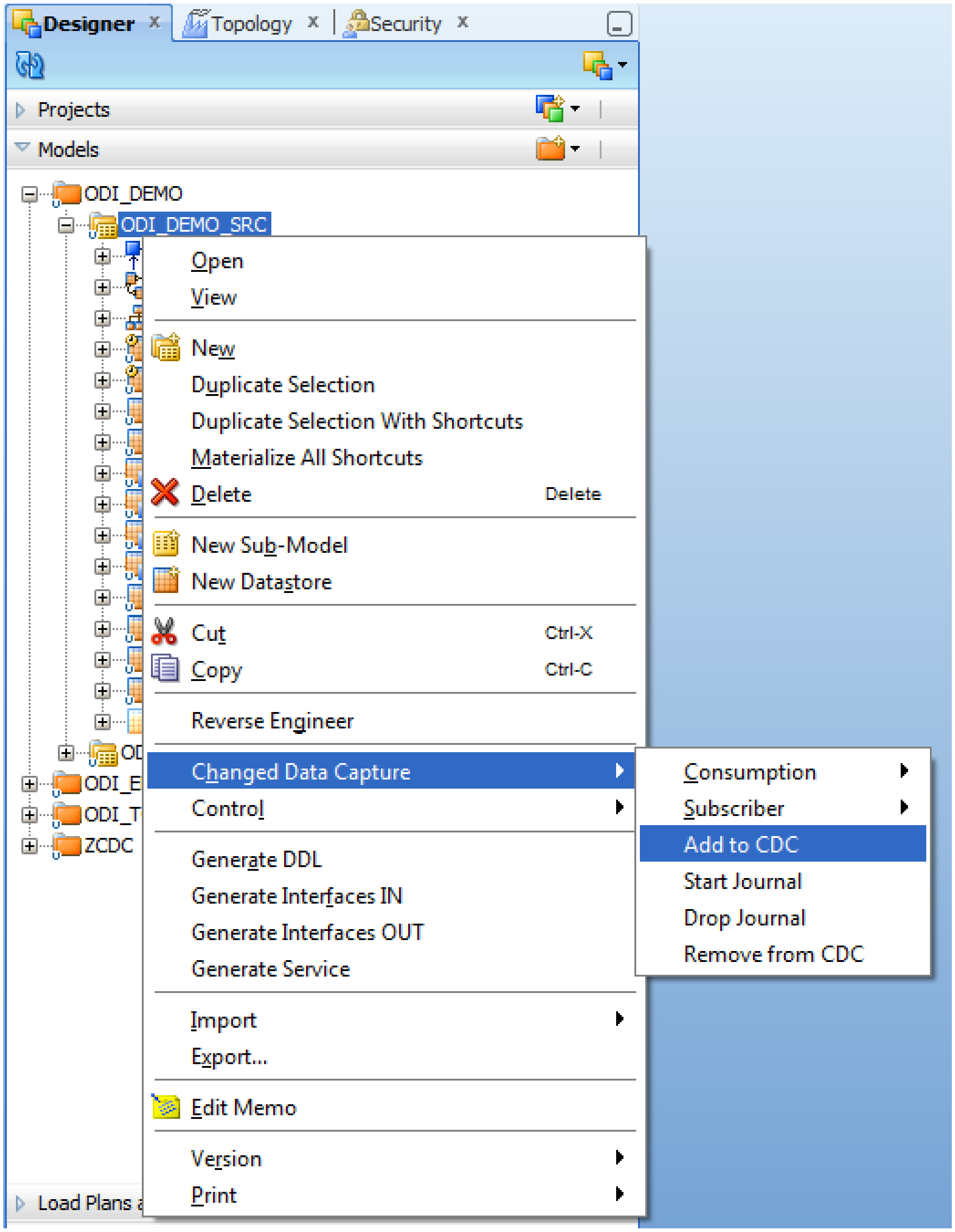
We Have Source Oracle (ODI_DEMO_SRC) and JMS Target (ODI_DEMO_TRG)

6.2.1 Model Updates

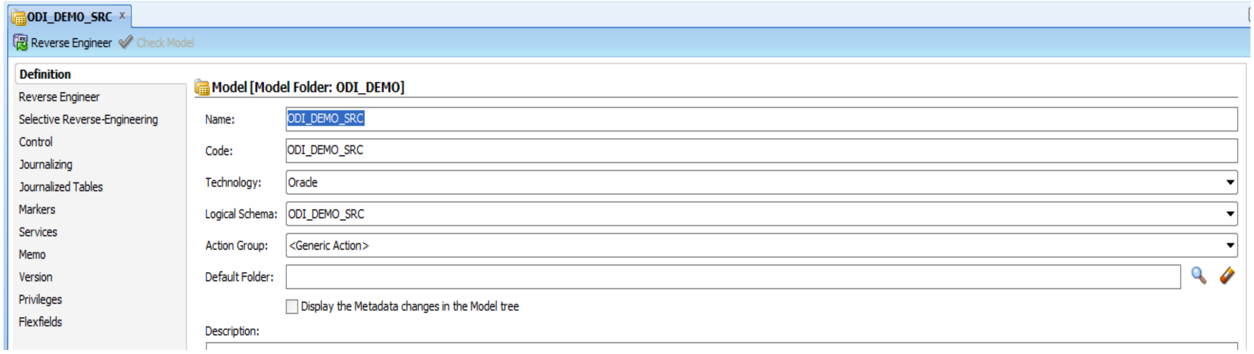
Open the Models accordion in the Designer navigator by clicking on the label



6.2.2 Right-click on the ODI_DEMO_SRC model and select Changed Data Capture > Add to CDC. Click yes to add all the tables to the CDC.


Designer
Topology
Security
Projects
Models
ODI_DEMO
ODI_DEMO_SRC
OD
ODI_E
ODI_T
ZCDC
Load Plans a
Open
View
New
Duplicate Selection
Duplicate Selection With Shortcuts
Materialize All Shortcuts
Delete
Delete
New Sub-Model
New Datastore
Cut
Ctrl-X
Copy
Ctrl-C
Reverse Engineer
Changed Data Capture
Control
Generate DDL
Generate Interfaces IN
Generate Interfaces OUT
Generate Service
Import
Export...
Edit Memo
Version
Print
Consumption
Subscriber
Add to CDC
Start Journal
Drop Journal
Remove from CDC

6.2.3 Double-click on the Oracle CDC Source model. The editor for this model opens



Model [Model Folder: ODI_DEMO]

Name: ODI_DEMO_SRC

Code: ODI_DEMO_SRC

Technology: Oracle

Logical Schema: ODI_DEMO_SRC

Action Group: <Generic Action>

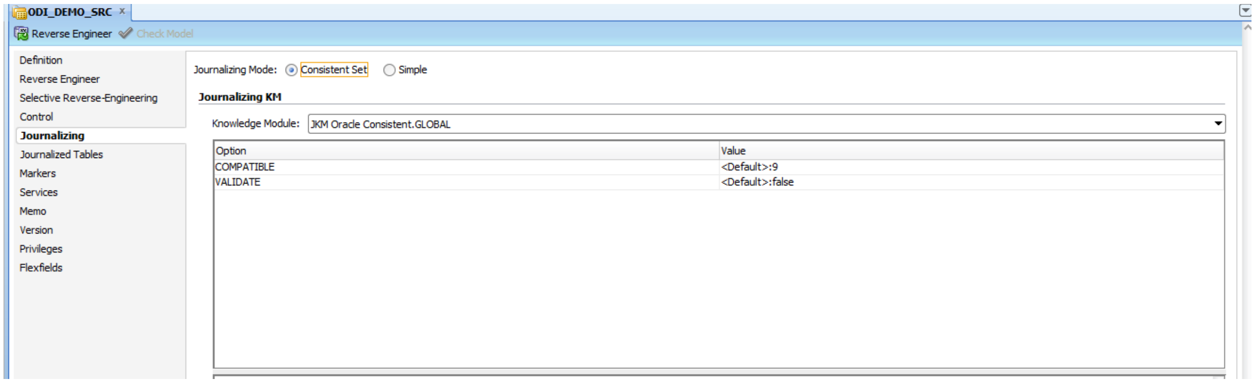
Default Folder:

□ Display the Metadata changes in the Model tree

Description:

6.2.4 Go to the Journalizing tab.

Select the **Consistent Set** journalizing mode, click **OK** to close the popup window then the **JKM Oracle Consistent** knowledge module



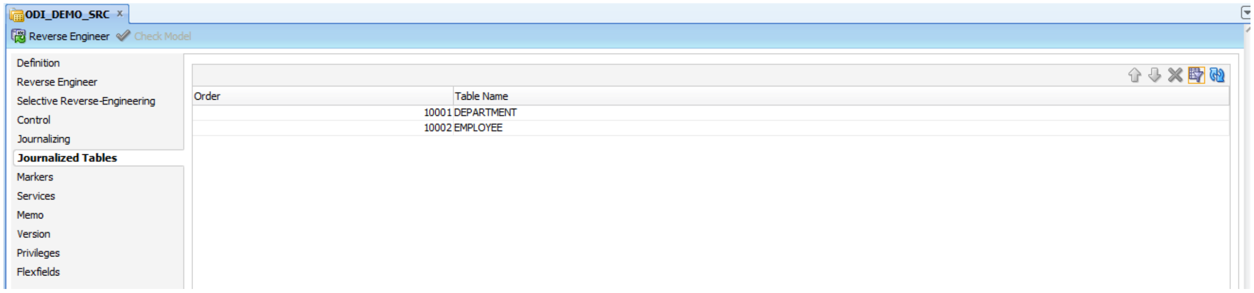
Journalizing Mode: ● Consistent Set ○ Simple

Journalizing KM

Knowledge Module: JKM Oracle Consistent.GLOBAL

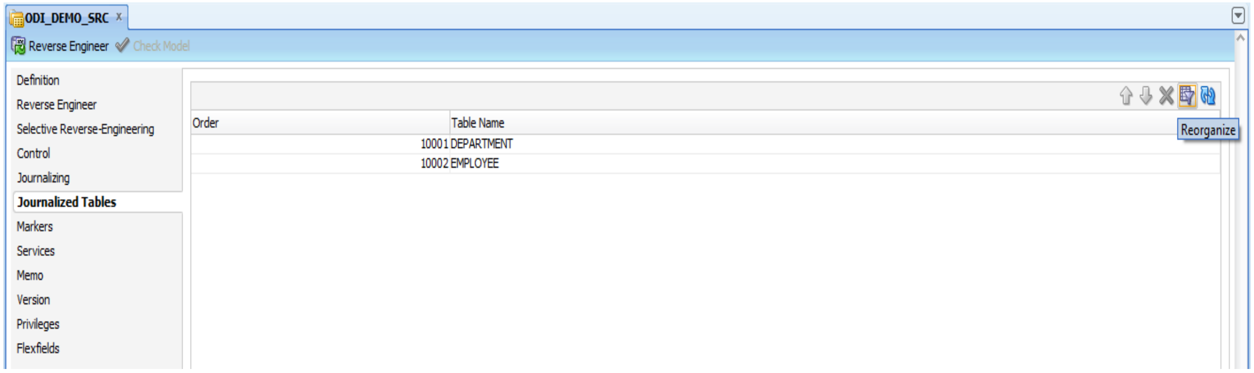
Option	Value
COMPATIBLE	<Default>:9
VALIDATE	<Default>:false

6.2.5 Go to the Journalized Tables tab, select the tables and click on the Remove from CDC button if we want to exclude some of the tables from CDC.



Order	Table Name
10001	DEPARTMENT
10002	EMPLOYEE

6.2.6 Click on the Reorganize button to order the tables according to their foreign-keys. Press yes on all confirmation dialogs. The data stores are automatically organized as shown below

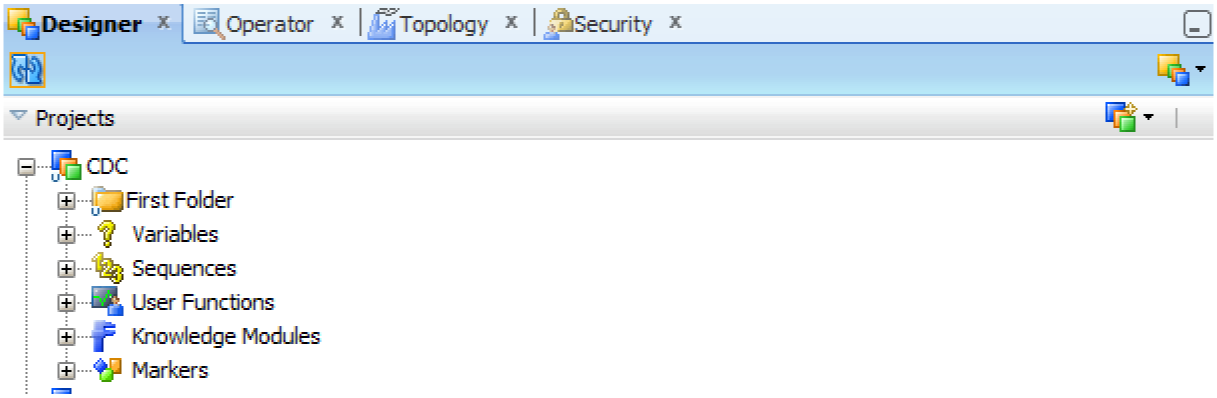


Order	Table Name
10001	DEPARTMENT
10002	EMPLOYEE

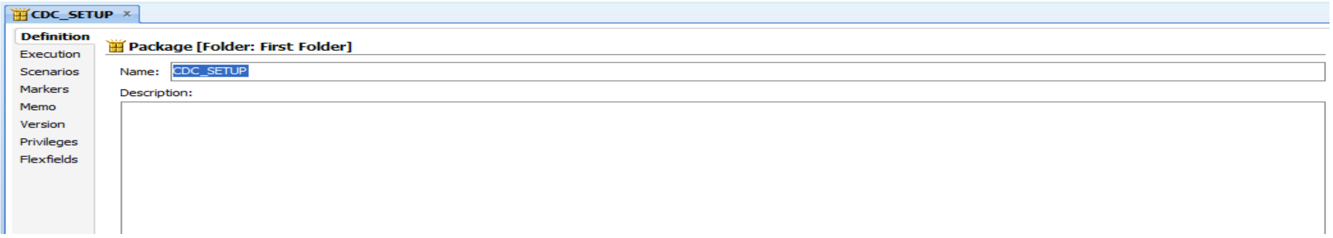
6.2.6 Click Save( ) on the toolbar to save the Model

6.3 Creating a Package to Set Up CDC

6.3.1 We will create a package that sets up the CDC infrastructure in a given context. In the Designer navigator, Click on the Projects accordion.



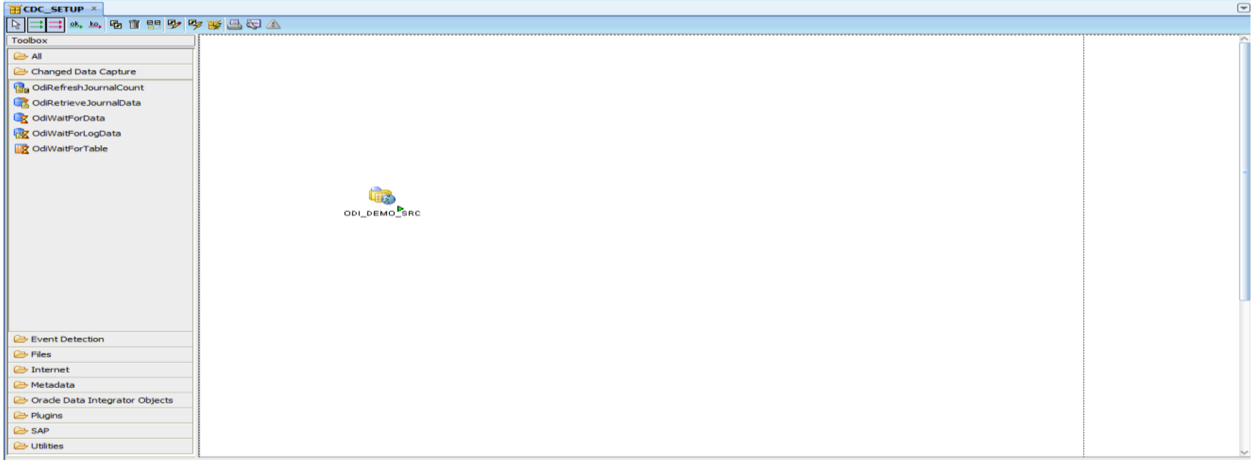
6.3.2 Expand the CDC project and then expand the First Folder folder Select Package and Create New Package with name CDC SETUP



Name: CDC_SETUP

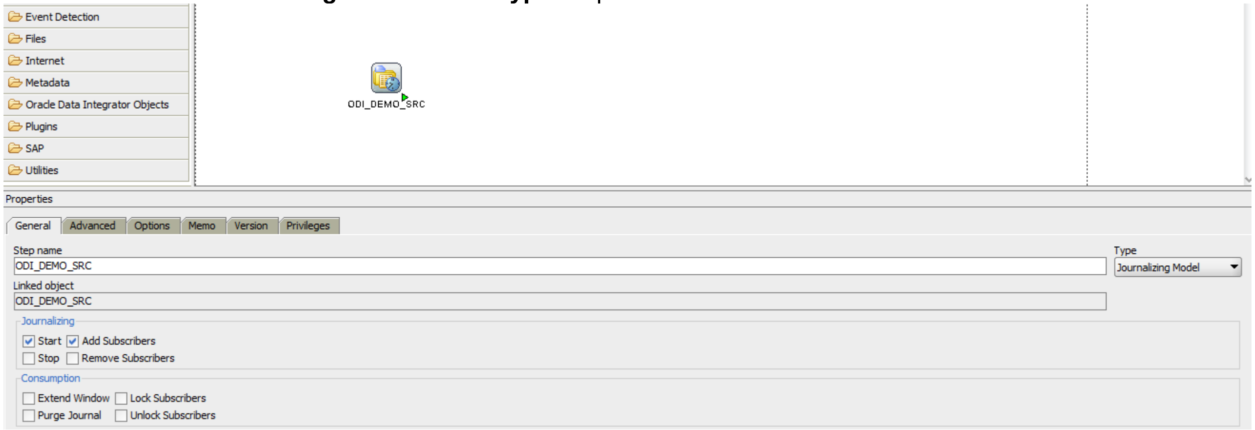
Description:

6.3.3 Go to the Diagram tab. Drag and drop into the diagram the ODI_DEMO_SRC data model from the Designer's Models tree view. A new step appears in the diagram, named after your data model



6.3.4 Click on this step. In the Properties panel:

- Select **Journalizing Model** in the **Type** drop-down list



Step name: ODI_DEMO_SRC

Type: Journalizing Model

Linked object: ODI_DEMO_SRC

Journalizing

☑ Start ☑ Add Subscribers

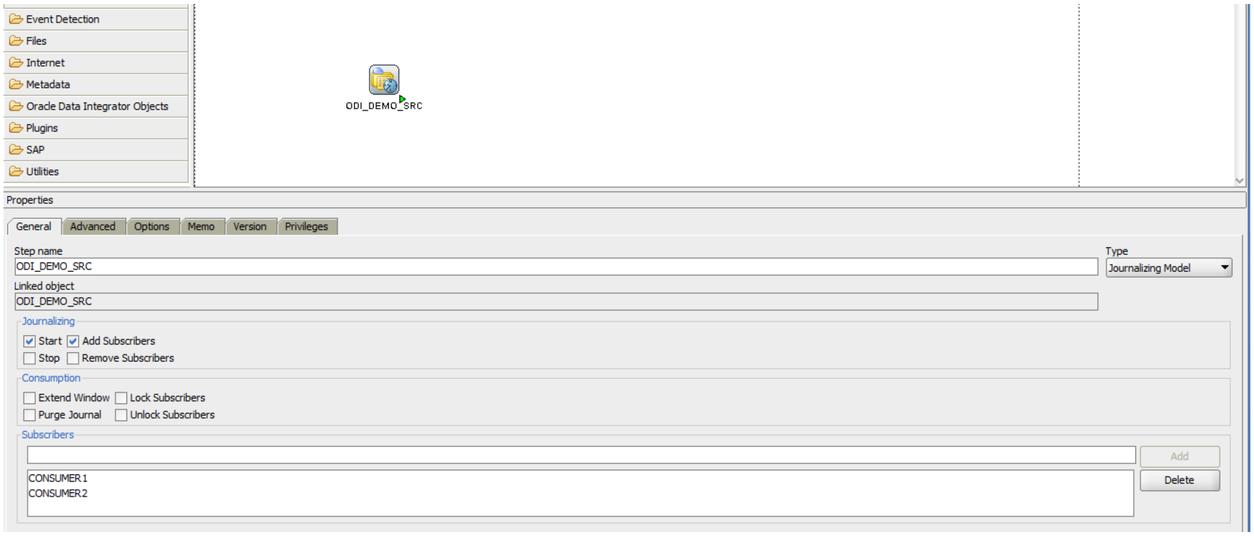
☐ Stop ☐ Remove Subscribers

Consumption

☐ Extend Window ☐ Lock Subscribers

☐ Purge Journal ☐ Unlock Subscribers

- Click the **Start** and **Add Subscribers** checkboxes.
- Enter CONSUMER1 in the **Subscribers** field, and then click **Add**.
- Enter CONSUMER2 in the **Subscribers** field, and then click **Add**.



Step name: ODI_DEMO_SRC

Type: Journalizing Model

Linked object: ODI_DEMO_SRC

Journalizing

☑ Start ☑ Add Subscribers

☐ Stop ☐ Remove Subscribers

Consumption

☐ Extend Window ☐ Lock Subscribers

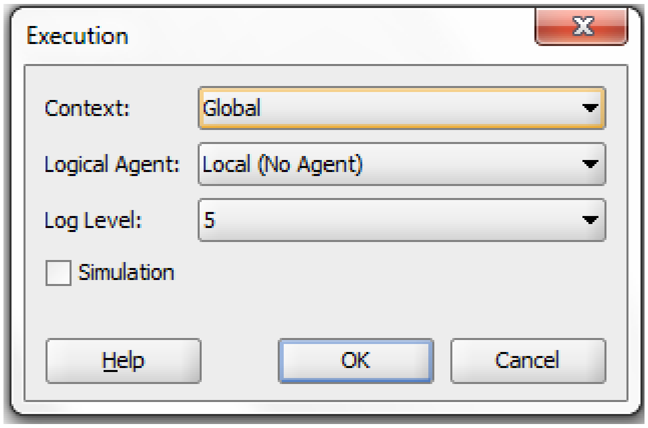
☐ Purge Journal ☐ Unlock Subscribers

Subscribers

CONSUMER1
CONSUMER2

6.3.5 Click Save( ) on the toolbar to save the package.

6.3.6 **Click on the Execute( ) button on the toolbar.**



Context: Global

Logical Agent: Local (No Agent)

Log Level: 5

☐ Simulation

6.3.7 Press OK when the **Session Started** window appears.

6.3.8 Open the **Operator** navigator. In the Operator, select the **Session List** tab, and expand the **All Executions** node.

6.3.9 Check that the last session ran correctly. You can review the steps and tasks that have activated the CDC process.

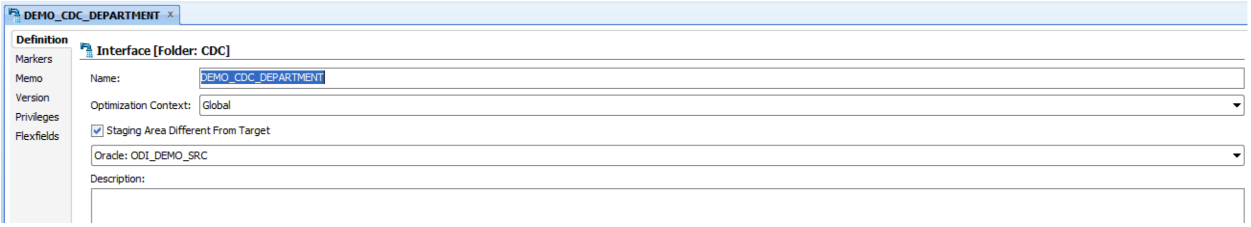
6.4 Creating the Integration Flows to Consume the Changes

In this exercise you will create the integration flows that will consume the changes captured using CDC.

1. In the Designer navigator, open the **Projects** accordion**.**

2. Expand the **ODI_DEMO_CDC** project and then expand the **First Folder** then Create Interfaces.

1. Create Interface DEMO_CDC_DEPARTMENT



DEMO_CDC_DEPARTMENT ×

Definition
Markers
Memo
Version
Privileges
Flexfields

Interface [Folder: CDC]

Name: DEMO_CDC_DEPARTMENT

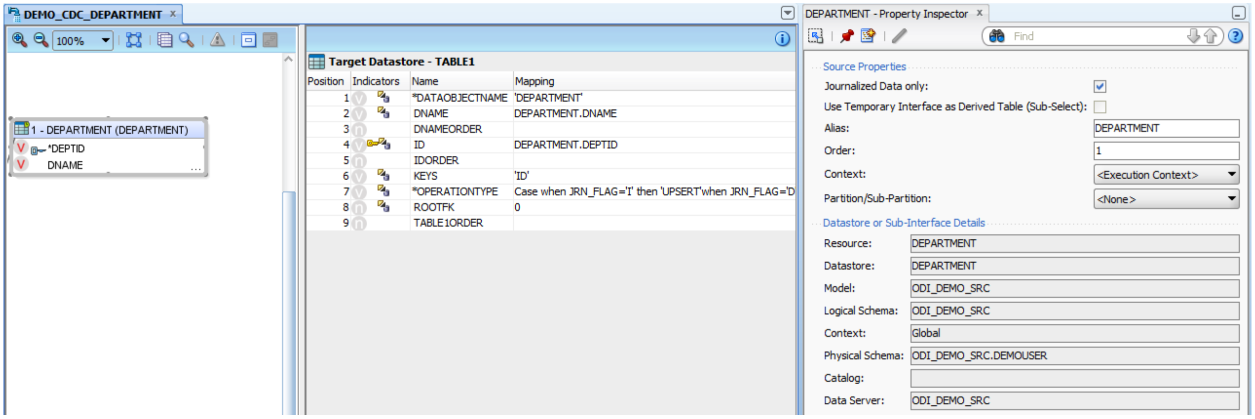
Optimization Context: Global

☑ Staging Area Different From Target

Oracle: ODI_DEMO_SRC

Description:

2. Go To mappings tab the Drag ODI_DEMO_SRC.DEPARTMENT to source and JMS.Table1 to target



DEMO_CDC_DEPARTMENT ×

1 - DEPARTMENT (DEPARTMENT)
*DEPTID
DNAME

Target Datastore - TABLE1

Position	Indicators	Name	Mapping
1	V	*DATAOBJECTNAME	'DEPARTMENT'
2	V	DNAME	DEPARTMENT.DNAME
3	n	DNAMEORDER	
4	V	ID	DEPARTMENT.DEPTID
5	n	IDORDER	
6	V	KEYS	'ID'
7	V	*OPERATIONTYPE	Case when JRN_FLAG='I' then 'UPSERT'when JRN_FLAG='D
8	n	ROOTFK	0
9	n	TABLE1ORDER	

DEPARTMENT - Property Inspector ×

Source Properties

Journalized Data only: ☑

Use Temporary Interface as Derived Table (Sub-Select): □

Alias: DEPARTMENT

Order: 1

Context: <Execution Context>

Partition/Sub-Partition: <None>

Datastore or Sub-Interface Details

Resource: DEPARTMENT

Datastore: DEPARTMENT

Model: ODI_DEMO_SRC

Logical Schema: ODI_DEMO_SRC

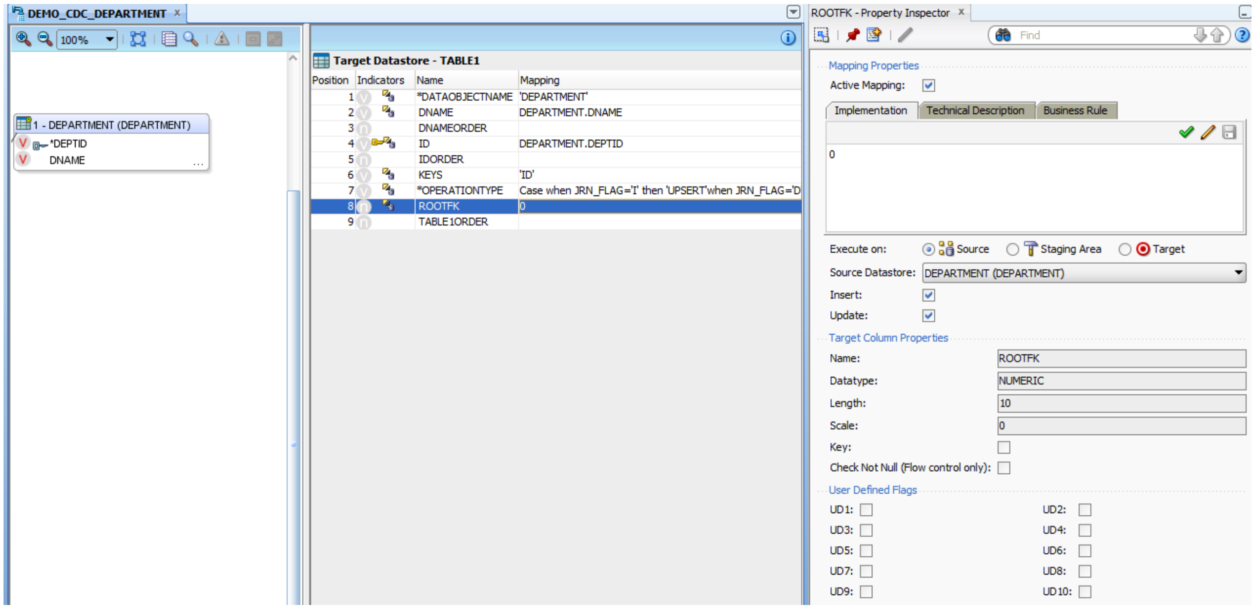
Context: Global

Physical Schema: ODI_DEMO_SRC.DEMOUSER

Catalog:

Data Server: ODI_DEMO_SRC

3. Select ROOTFK and update its value to 0



DEMO_CDC_DEPARTMENT ×

1 - DEPARTMENT (DEPARTMENT)
*DEPTID
DNAME

Target Datastore - TABLE1

Position	Indicators	Name	Mapping
1	V	*DATAOBJECTNAME	'DEPARTMENT'
2	V	DNAME	DEPARTMENT.DNAME
3	n	DNAMEORDER	
4	V	ID	DEPARTMENT.DEPTID
5	n	IDORDER	
6	V	KEYS	'ID'
7	V	*OPERATIONTYPE	Case when JRN_FLAG='I' then 'UPSERT'when JRN_FLAG='D
8	n	ROOTFK	0
9	n	TABLE1ORDER	

ROOTFK - Property Inspector ×

Mapping Properties

Active Mapping: ☑

Implementation | Technical Description | Business Rule

0

Execute on: ○ Source ○ Staging Area ● Target

Source Datastore: DEPARTMENT (DEPARTMENT)

Insert: ☑

Update: ☑

Target Column Properties

Name: ROOTFK

Datatype: NUMERIC

Length: 10

Scale: 0

Key: □

Check Not Null (Flow control only): □

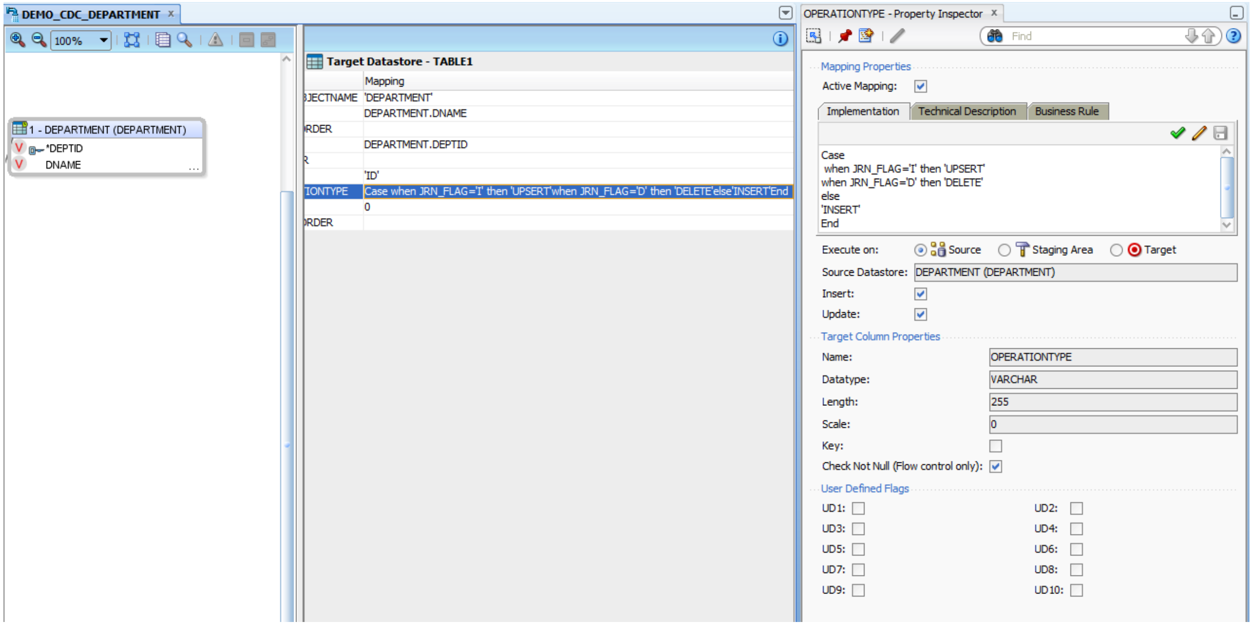
User Defined Flags

UD1: □ UD2: □
UD3: □ UD4: □
UD5: □ UD6: □
UD7: □ UD8: □
UD9: □ UD10: □

4. Select OPERATION TYPE and update its value with following code

```
Case
 when JRN_FLAG='I' then 'UPSERT'
when JRN_FLAG='D' then 'DELETE'
else
'INSERT'
End
```

And Assign KEYS and DATAOBJECT NAME to ‘ID’ and ‘DEPARTMENT’ respectively..



Target Datastore - TABLE1

	Mapping
JECTNAME	'DEPARTMENT'
	DEPARTMENT.DNAME
RDER	
	DEPARTMENT.DEPTID
R	
	'ID'
IONTYPE	Case when JRN_FLAG='I' then 'UPSERT'when JRN_FLAG='D' then 'DELETE'else'INSERT'End
	0
RDER	

Mapping Properties

Active Mapping: ☑

```
Case
 when JRN_FLAG='I' then 'UPSERT'
when JRN_FLAG='D' then 'DELETE'
else
'INSERT'
End
```

Execute on: ○ Source ○ Staging Area ● Target

Source Datastore: DEPARTMENT (DEPARTMENT)

Insert: ☑

Update: ☑

Target Column Properties

Name: OPERATIONTYPE

Datatype: VARCHAR

Length: 255

Scale: 0

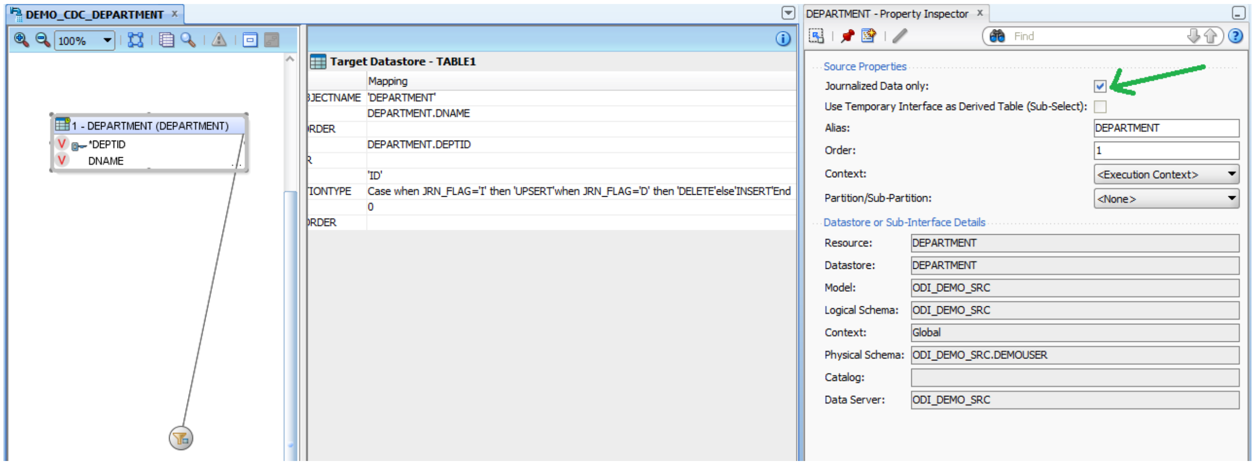
Key: □

Check Not Null (Flow control only): ☑

User Defined Flags

UD1: □ UD2: □
UD3: □ UD4: □
UD5: □ UD6: □
UD7: □ UD8: □
UD9: □ UD10: □

5. Select DEPARTMENT form Source and go to properties then check Journalized Data Only.



Target Datastore - TABLE1

	Mapping
JECTNAME	'DEPARTMENT'
	DEPARTMENT.DNAME
RDER	
	DEPARTMENT.DEPTID
R	
	'ID'
IONTYPE	Case when JRN_FLAG='I' then 'UPSERT'when JRN_FLAG='D' then 'DELETE'else'INSERT'End
	0
RDER	

Source Properties

Journalized Data only: ☑

Use Temporary Interface as Derived Table (Sub-Select): □

Alias: DEPARTMENT

Order: 1

Context: <Execution Context>

Partition/Sub-Partition: <None>

Datastore or Sub-Interface Details

Resource: DEPARTMENT

Datastore: DEPARTMENT

Model: ODI_DEMO_SRC

Logical Schema: ODI_DEMO_SRC

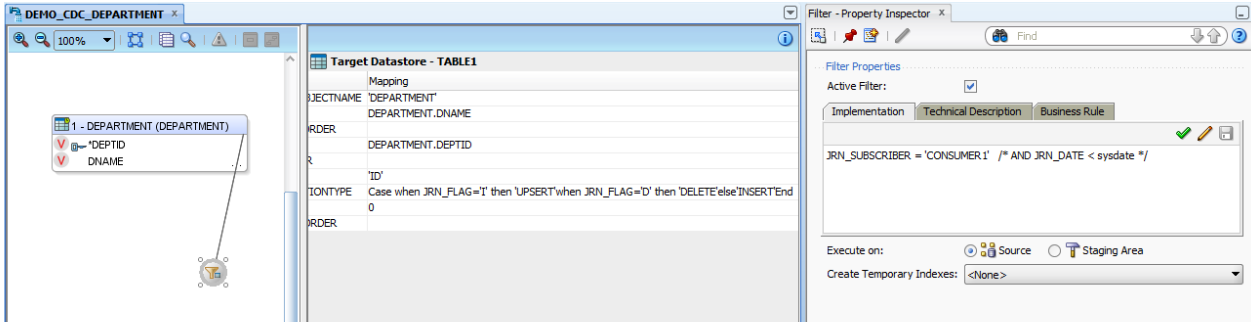
Context: Global

Physical Schema: ODI_DEMO_SRC.DEMOUSER

Catalog:

Data Server: ODI_DEMO_SRC

6. Select filter and update it’s with appropriate subscriber



Target Datastore - TABLE1

	Mapping
JECTNAME	'DEPARTMENT'
	DEPARTMENT.DNAME
RDER	
	DEPARTMENT.DEPTID
R	
	'ID'
TIONTYPE	Case when JRN_FLAG='I' then 'UPSERT'when JRN_FLAG='D' then 'DELETE'else'INSERT'End
	0
RDER	

Filter Properties

Active Filter: ☑

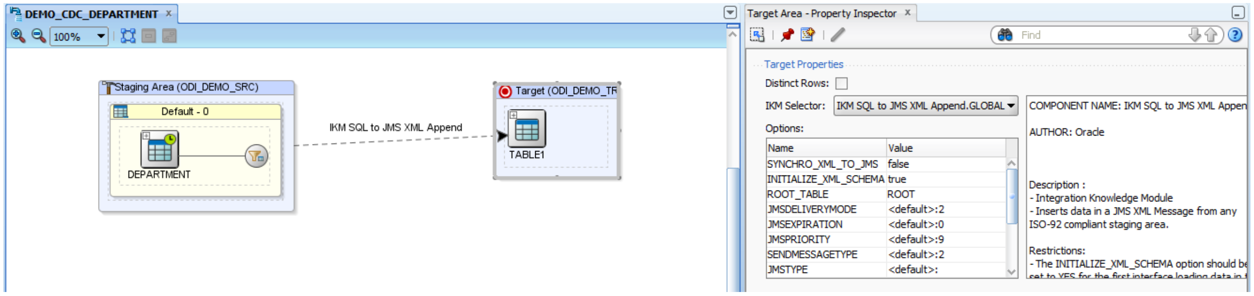
Implementation | Technical Description | Business Rule

```
JRN_SUBSCRIBER = 'CONSUMER1'   /* AND JRN_DATE < sysdate */
```

Execute on: ● Source ○ Staging Area

Create Temporary Indexes: <None>

7. Go to Flow tab and Select target then update IKM properties
   SYNCHRO_XM_TO_JMS to false {It should be true when we want send separate JMS message for this data source}
   INITIALIZE_XML_SCHEMA to true
   ROOT_TABLE to ROOT.


DEMO_CDC_DEPARTMENT
100%
Staging Area (ODI_DEMO_SRC)
Default - 0
DEPARTMENT
IKM SQL to JMS XML Append
Target (ODI_DEMO_TR
TABLE1
Target Area - Property Inspector
Find

Target Properties

Distinct Rows: □

IKM Selector: IKM SQL to JMS XML Append.GLOBAL

Options:

Name	Value
SYNCHRO_XML_TO_JMS	false
INITIALIZE_XML_SCHEMA	true
ROOT_TABLE	ROOT
JMSDELIVERYMODE	<default>:2
JMSEXPIRATION	<default>:0
JMSPRIORITY	<default>:9
SENDMESSAGETYPE	<default>:2
JMSTYPE	<default>:

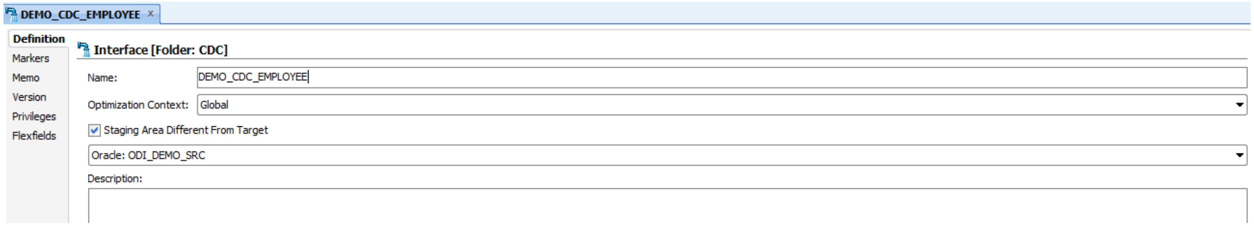
COMPONENT NAME: IKM SQL to JMS XML Appen

AUTHOR: Oracle

Description :
- Integration Knowledge Module
- Inserts data in a JMS XML Message from any ISO-92 compliant staging area.

Restrictions:
- The INITIALIZE_XML_SCHEMA option should be set to YES for the first interface loading data in t

3. Create Interface For Employee
   1. Create   with name DEMO_CDC_EMPLOYEE



Interface [Folder: CDC]

Name: DEMO_CDC_EMPLOYEE

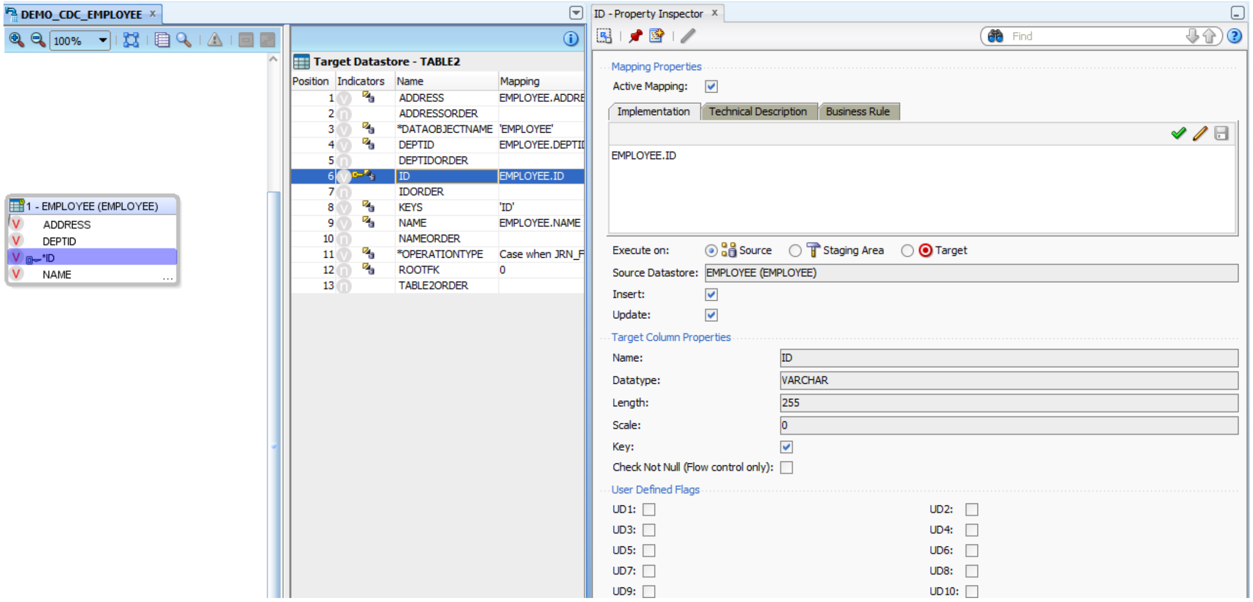
Optimization Context: Global

☑ Staging Area Different From Target

Oracle: ODI_DEMO_SRC

Description:

2. Go To mappings tab the Drag ODI_DEMO_SRC.EMPLOYEE to source and JMS.Table2 to target



DEMO_CDC_EMPLOYEE

1 - EMPLOYEE (EMPLOYEE)
- ADDRESS
- DEPTID
- *ID
- NAME

Target Datastore - TABLE2

Position	Indicators	Name	Mapping
1	V	ADDRESS	EMPLOYEE.ADDRE
2	n	ADDRESSORDER	
3	V	*DATAOBJECTNAME	'EMPLOYEE'
4	V	DEPTID	EMPLOYEE.DEPTII
5	n	DEPTIDORDER	
6	V	ID	EMPLOYEE.ID
7	n	IDORDER	
8	V	KEYS	'ID'
9	V	NAME	EMPLOYEE.NAME
10	n	NAMEORDER	
11	V	*OPERATIONTYPE	Case when JRN_F
12	n	ROOTFK	0
13	n	TABLE2ORDER	

ID - Property Inspector

Mapping Properties

Active Mapping: ☑

Implementation | Technical Description | Business Rule

EMPLOYEE.ID

Execute on: ○ Source ○ Staging Area ● Target

Source Datastore: EMPLOYEE (EMPLOYEE)

Insert: ☑

Update: ☑

Target Column Properties

Name: ID

Datatype: VARCHAR

Length: 255

Scale: 0

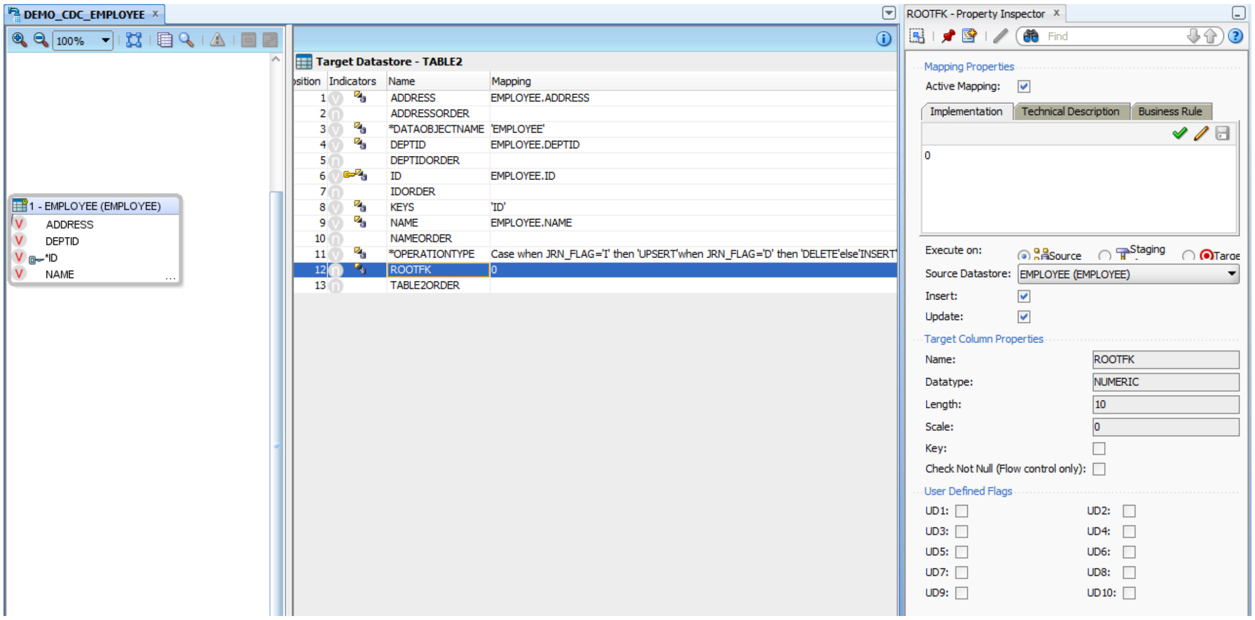
Key: ☑

Check Not Null (Flow control only): ☐

User Defined Flags

UD1: ☐ UD2: ☐
UD3: ☐ UD4: ☐
UD5: ☐ UD6: ☐
UD7: ☐ UD8: ☐
UD9: ☐ UD10: ☐

3. Select ROOTFK and update its value to 0



DEMO_CDC_EMPLOYEE

1 - EMPLOYEE (EMPLOYEE)
- ADDRESS
- DEPTID
- *ID
- NAME

Target Datastore - TABLE2

osition	Indicators	Name	Mapping
1	V	ADDRESS	EMPLOYEE.ADDRESS
2	n	ADDRESSORDER	
3	V	*DATAOBJECTNAME	'EMPLOYEE'
4	V	DEPTID	EMPLOYEE.DEPTID
5	n	DEPTIDORDER	
6	V	ID	EMPLOYEE.ID
7	n	IDORDER	
8	V	KEYS	'ID'
9	V	NAME	EMPLOYEE.NAME
10	n	NAMEORDER	
11	V	*OPERATIONTYPE	Case when JRN_FLAG='I' then 'UPSERT'when JRN_FLAG='D' then 'DELETE'else'INSERT'
12	n	ROOTFK	0
13	n	TABLE2ORDER	

ROOTFK - Property Inspector

Mapping Properties

Active Mapping: ☑

Implementation | Technical Description | Business Rule

0

Execute on: ○ Source ○ Staging ● Targe

Source Datastore: EMPLOYEE (EMPLOYEE)

Insert: ☑

Update: ☑

Target Column Properties

Name: ROOTFK

Datatype: NUMERIC

Length: 10

Scale: 0

Key: ☐

Check Not Null (Flow control only): ☐

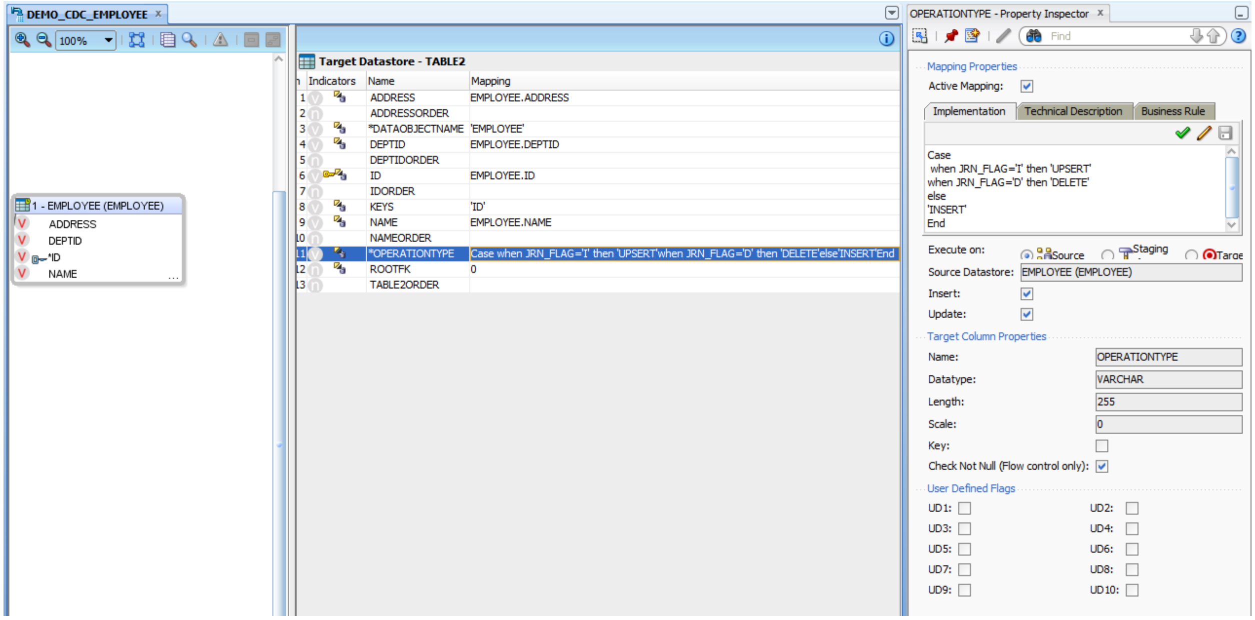
User Defined Flags

UD1: ☐ UD2: ☐
UD3: ☐ UD4: ☐
UD5: ☐ UD6: ☐
UD7: ☐ UD8: ☐
UD9: ☐ UD10: ☐

4. Select OPERATION NAME and update its value with following code

```
Case
 when JRN_FLAG='I' then 'UPSERT'
when JRN_FLAG='D' then 'DELETE'
else
'INSERT'
End
```

and Assign KEYS and DATAOBJECT NAME to ‘ID’ and ‘EMPLOYEE’ respectively..



Target Datastore - TABLE2

	Indicators	Name	Mapping
1		ADDRESS	EMPLOYEE.ADDRESS
2		ADDRESSORDER	
3		*DATAOBJECTNAME	'EMPLOYEE'
4		DEPTID	EMPLOYEE.DEPTID
5		DEPTIDORDER	
6		ID	EMPLOYEE.ID
7		IDORDER	
8		KEYS	'ID'
9		NAME	EMPLOYEE.NAME
10		NAMEORDER	
11		*OPERATIONTYPE	Case when JRN_FLAG='I' then 'UPSERT'when JRN_FLAG='D' then 'DELETE'else'INSERT'End
12		ROOTFK	0
13		TABLE2ORDER	

OPERATIONTYPE - Property Inspector

Mapping Properties

Active Mapping: ☑

Implementation | Technical Description | Business Rule

```
Case
 when JRN_FLAG='I' then 'UPSERT'
when JRN_FLAG='D' then 'DELETE'
else
'INSERT'
End
```

Execute on: ● Source ○ Staging ○ Targe

Source Datastore: EMPLOYEE (EMPLOYEE)

Insert: ☑

Update: ☑

Target Column Properties

Name: OPERATIONTYPE

Datatype: VARCHAR

Length: 255

Scale: 0

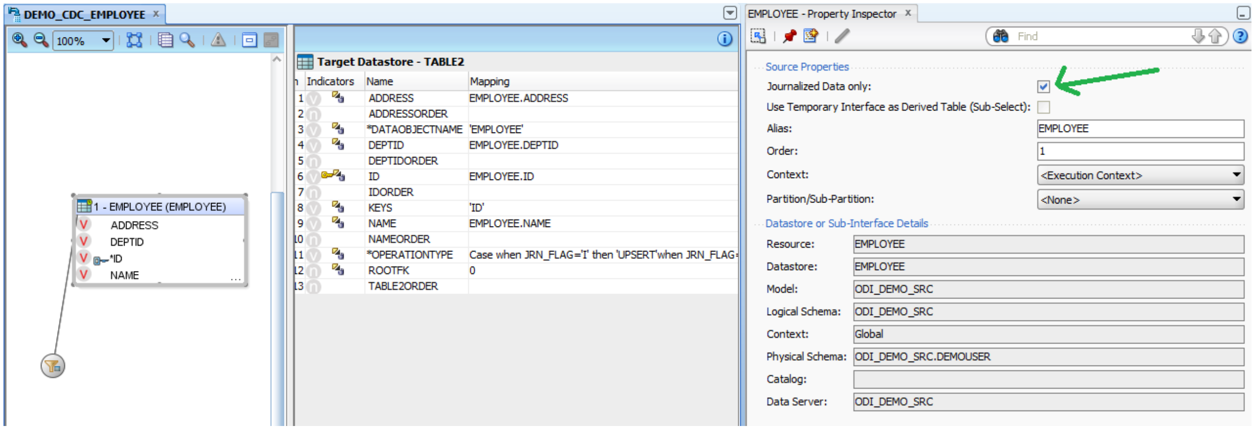
Key: □

Check Not Null (Flow control only): ☑

User Defined Flags

UD1: □ UD2: □
UD3: □ UD4: □
UD5: □ UD6: □
UD7: □ UD8: □
UD9: □ UD10: □

5. Select EMPLOYEE form Source and go to properties then check Journalized Data Only.



Target Datastore - TABLE2

	Indicators	Name	Mapping
1		ADDRESS	EMPLOYEE.ADDRESS
2		ADDRESSORDER	
3		*DATAOBJECTNAME	'EMPLOYEE'
4		DEPTID	EMPLOYEE.DEPTID
5		DEPTIDORDER	
6		ID	EMPLOYEE.ID
7		IDORDER	
8		KEYS	'ID'
9		NAME	EMPLOYEE.NAME
10		NAMEORDER	
11		*OPERATIONTYPE	Case when JRN_FLAG='I' then 'UPSERT'when JRN_FLAG=
12		ROOTFK	0
13		TABLE2ORDER	

EMPLOYEE - Property Inspector

Source Properties

Journalized Data only: ☑

Use Temporary Interface as Derived Table (Sub-Select): □

Alias: EMPLOYEE

Order: 1

Context: <Execution Context>

Partition/Sub-Partition: <None>

Datastore or Sub-Interface Details

Resource: EMPLOYEE

Datastore: EMPLOYEE

Model: ODI_DEMO_SRC

Logical Schema: ODI_DEMO_SRC

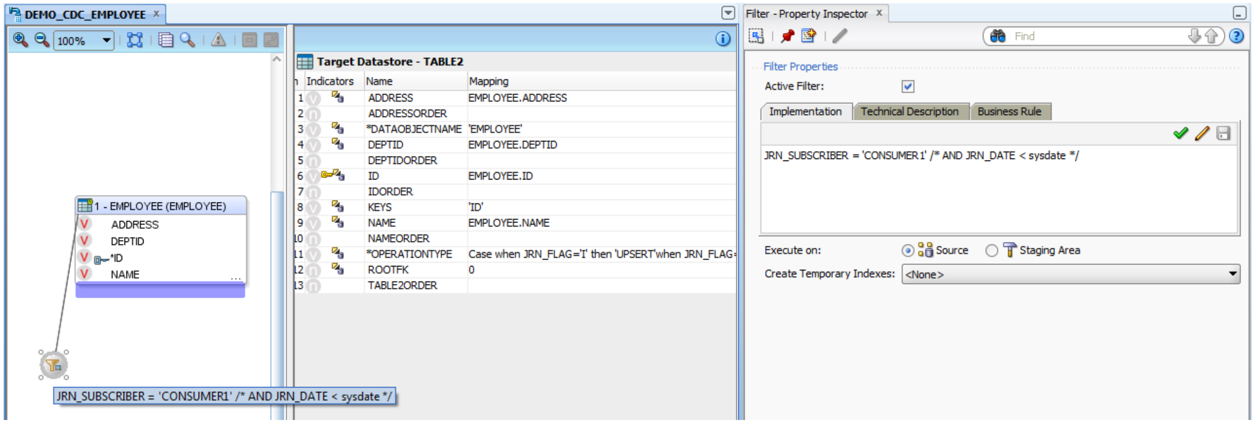
Context: Global

Physical Schema: ODI_DEMO_SRC.DEMOUSER

Catalog:

Data Server: ODI_DEMO_SRC

6. Select filter and update it's with appropriate subscriber



DEMO_CDC_EMPLOYEE

1 - EMPLOYEE (EMPLOYEE)
- ADDRESS
- DEPTID
- *ID
- NAME

JRN_SUBSCRIBER = 'CONSUMER1' /* AND JRN_DATE < sysdate */

Target Datastore - TABLE2

	Indicators	Name	Mapping
1	V	ADDRESS	EMPLOYEE.ADDRESS
2	n	ADDRESSORDER	
3	V	*DATAOBJECTNAME	'EMPLOYEE'
4	V	DEPTID	EMPLOYEE.DEPTID
5	n	DEPTIDORDER	
6	V	ID	EMPLOYEE.ID
7	n	IDORDER	
8	V	KEYS	'ID'
9	V	NAME	EMPLOYEE.NAME
10	n	NAMEORDER	
11	V	*OPERATIONTYPE	Case when JRN_FLAG='I' then 'UPSERT'when JRN_FLAG=
12	n	ROOTFK	0
13	n	TABLE2ORDER	

Filter - Property Inspector

Filter Properties

Active Filter: ☑

Implementation | Technical Description | Business Rule

JRN_SUBSCRIBER = 'CONSUMER1' /* AND JRN_DATE < sysdate */

Execute on: ● Source ○ Staging Area

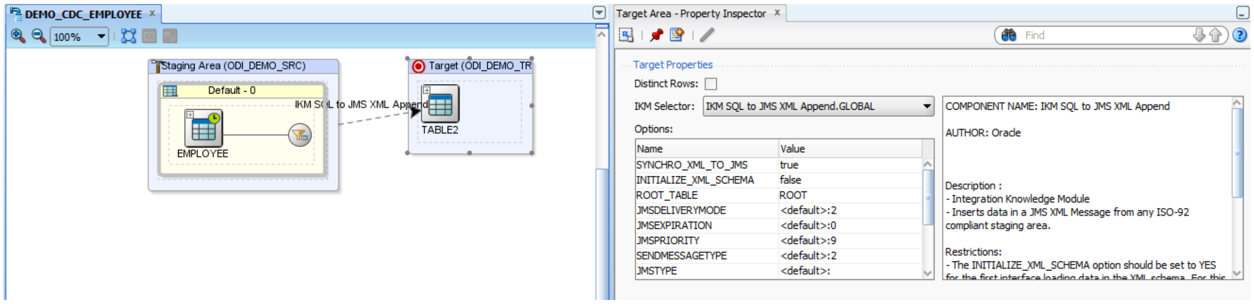
Create Temporary Indexes: <None>

7. Go to Flow tab and Select target then update IKM properties

SYNCHRO_XM_TO_JMS to true

INITIALIZE_XML_SCHEMA to false

ROOT_TABLE to ROOT.



DEMO_CDC_EMPLOYEE

Target Area - Property Inspector

Target Properties

Distinct Rows: □

IKM Selector: IKM SQL to JMS XML Append.GLOBAL

Options:

Name	Value
SYNCHRO_XML_TO_JMS	true
INITIALIZE_XML_SCHEMA	false
ROOT_TABLE	ROOT
JMSDELIVERYMODE	<default>:2
JMSEXPIRATION	<default>:0
JMSPRIORITY	<default>:9
SENDMESSAGETYPE	<default>:2
JMSTYPE	<default>:

COMPONENT NAME: IKM SQL to JMS XML Append

AUTHOR: Oracle

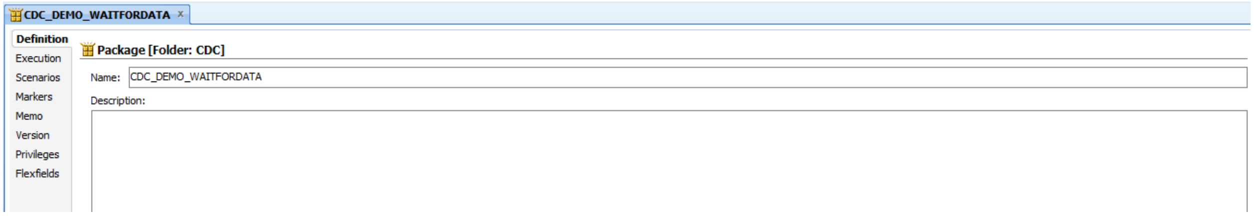
Description :
- Integration Knowledge Module
- Inserts data in a JMS XML Message from any ISO-92 compliant staging area.

Restrictions:
- The INITIALIZE_XML_SCHEMA option should be set to YES for the first interface loading data in the XML schema. For this

6.5 Construct Package For CDC with Interfaces

Expand the **CDC** project and then expand the **First Folder** folder. Select the **Packages** node, right-click and select **New Package.**

1. Enter the following package name: CDC_DEMO_WAITFORDATA



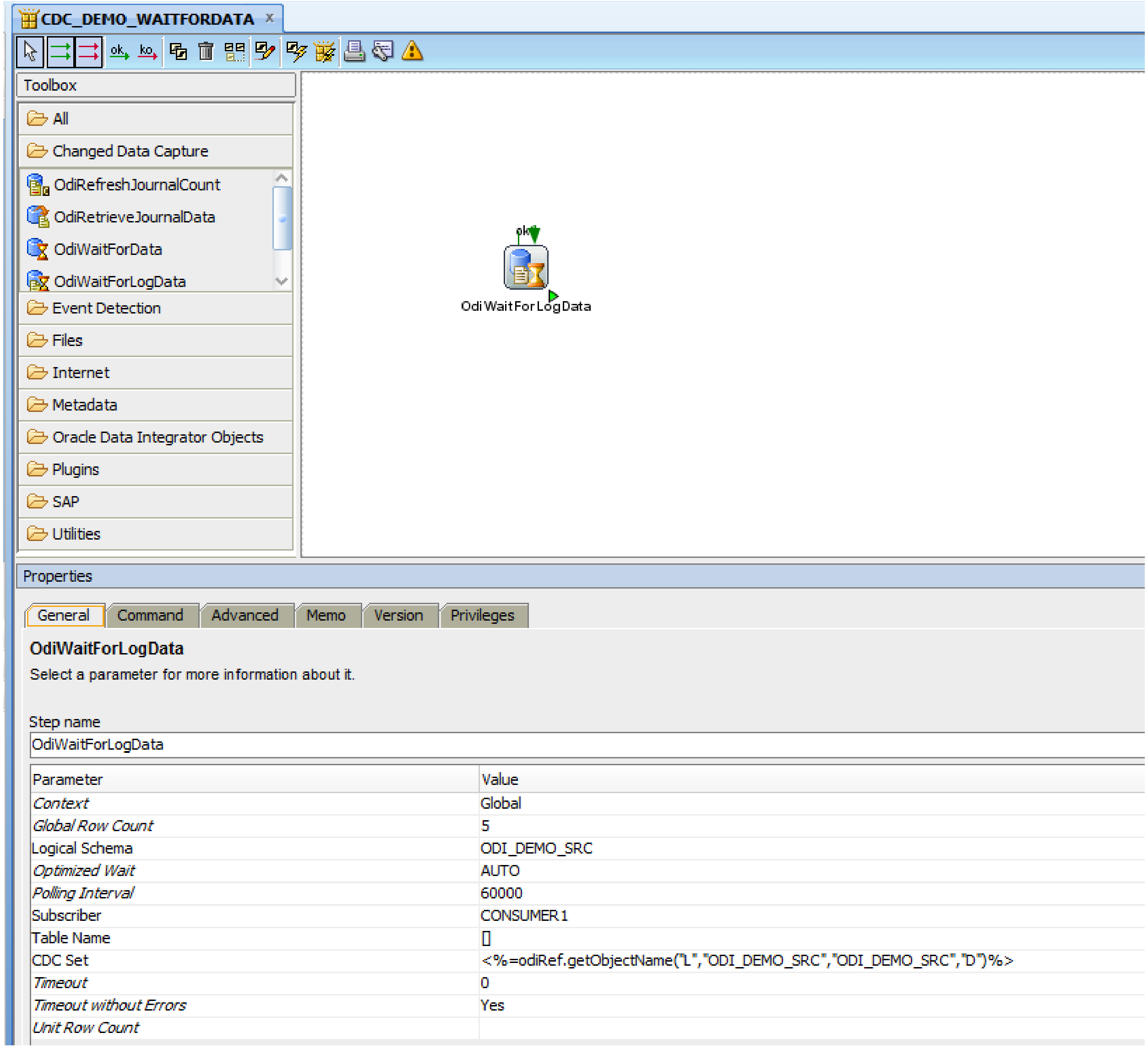
CDC_DEMO_WAITFORDATA

Package [Folder: CDC]

Name: CDC_DEMO_WAITFORDATA

Description:

1. Go to the **Diagram** tab and Go to Tool box and select ODIWaitForLogData. This step helps us to wait for defined number of changes happened to Data Source. We have defined Global Row Count to 5.



OdiWaitForLogData

Select a parameter for more information about it.

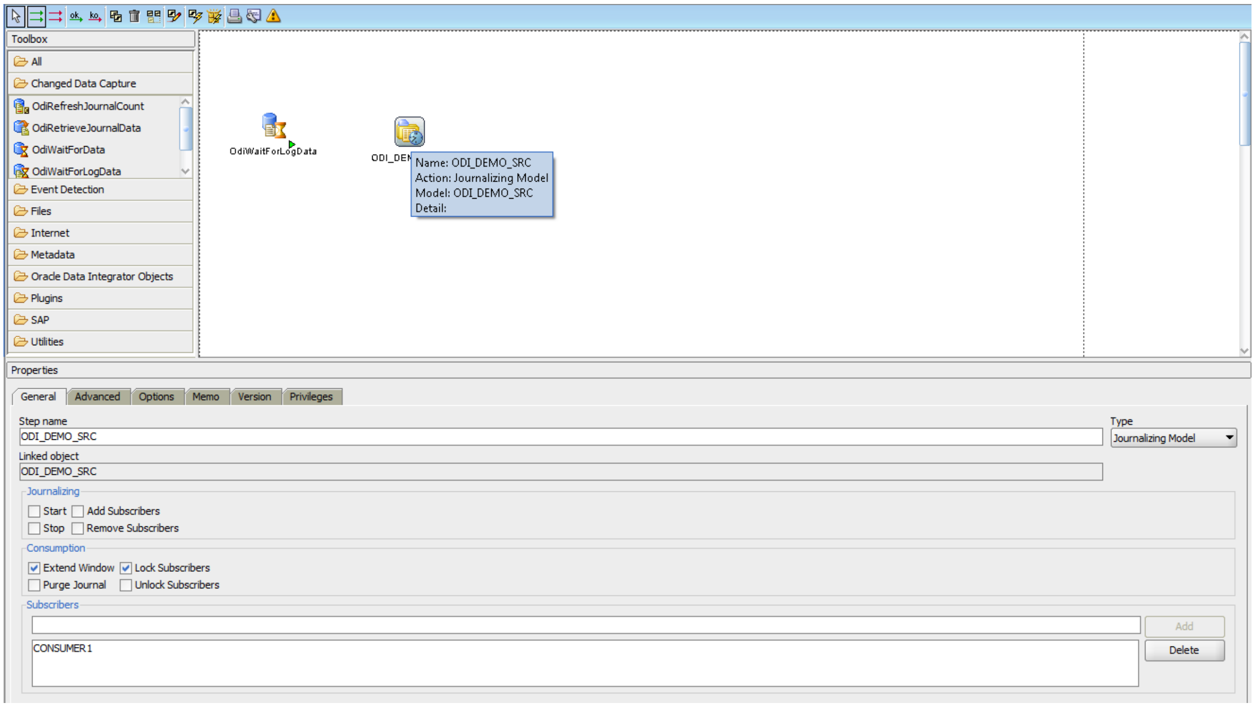
Step name

OdiWaitForLogData

Parameter	Value
Context	Global
Global Row Count	5
Logical Schema	ODI_DEMO_SRC
Optimized Wait	AUTO
Polling Interval	60000
Subscriber	CONSUMER1
Table Name	[]
CDC Set	<%=odiRef.getObjectName("L","ODI_DEMO_SRC","ODI_DEMO_SRC","D")%>
Timeout	0
Timeout without Errors	Yes
Unit Row Count	

2. Drag and drop into the diagram the **ODI_DEMO_SRC** data model from the Designer's Models tree view. A new step appears in the diagram, named after your data model.

   Select **Journalizing Model** in the **Type** drop-down list
   Click the **Extend Window** and **Lock Subscribers** checkboxes.
   Enter CONSUMER1 in the **Subscribers** field, and then click **Add**.



Name: ODI_DEMO_SRC
Action: Journalizing Model
Model: ODI_DEMO_SRC
Detail:

Step name: ODI_DEMO_SRC
Type: Journalizing Model
Linked object: ODI_DEMO_SRC

Journalizing
☐ Start ☐ Add Subscribers
☐ Stop ☐ Remove Subscribers

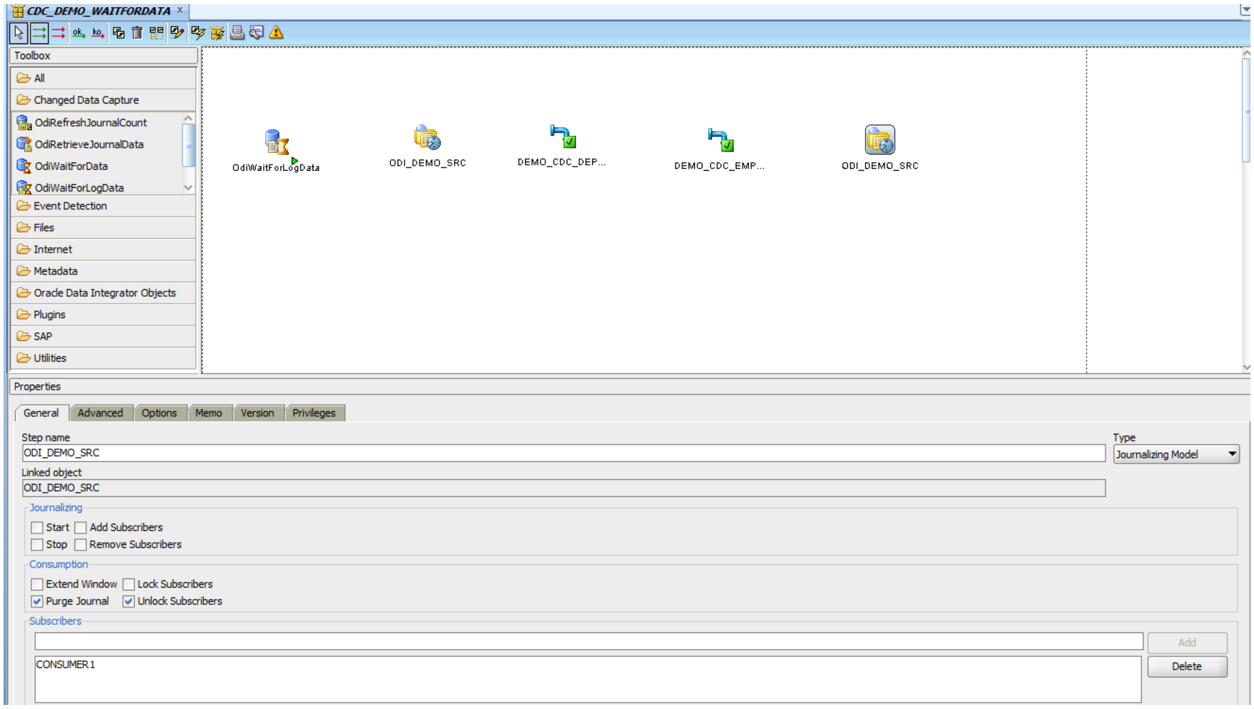
Consumption
☑ Extend Window ☑ Lock Subscribers
☐ Purge Journal ☐ Unlock Subscribers

Subscribers
CONSUMER1

3. Drag and drop the **DEMO_CDC_DEPARTMENT** and **DEMO_CDC_EMPLOYEE** interfaces from the Designer's Projects tree view.

4. Drag and drop into the diagram the **Oracle CDC Source** data model from the Designer's Models tree view. A new step appears in the diagram, named after your data model.

   Click this step. In the Properties panel:
   - Select **Journalizing Model** in the **Type** drop-down list.
   - Click the **Purge Journal** and **Unlock Subscribers** checkboxes
   - Enter CONSUMER1 in the **Subscribers** field, and then click **Add**.



Step name: ODI_DEMO_SRC

Type: Journalizing Model

Linked object: ODI_DEMO_SRC

Journalizing

□ Start □ Add Subscribers

□ Stop □ Remove Subscribers

Consumption

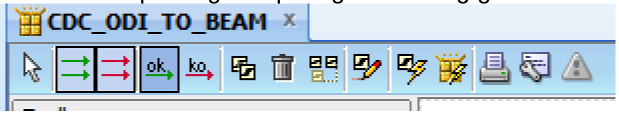
□ Extend Window □ Lock Subscribers

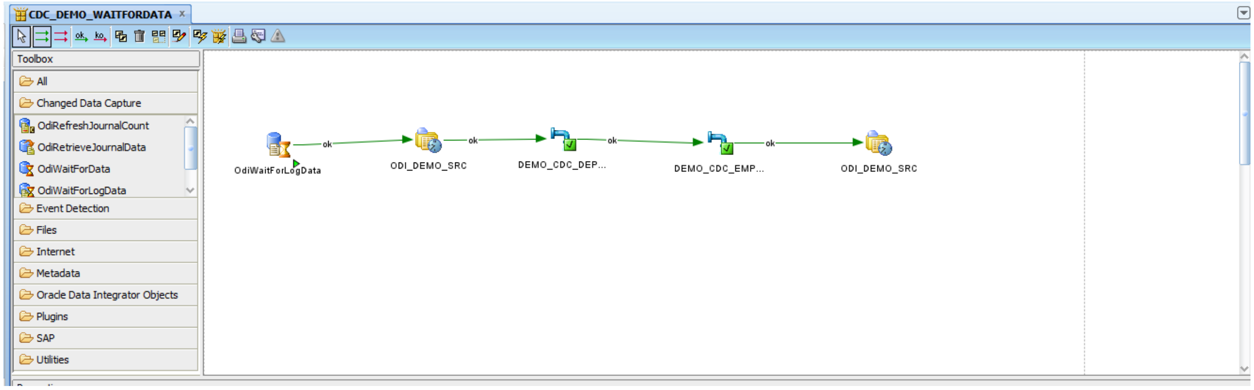
☑ Purge Journal ☑ Unlock Subscribers

Subscribers

CONSUMER1

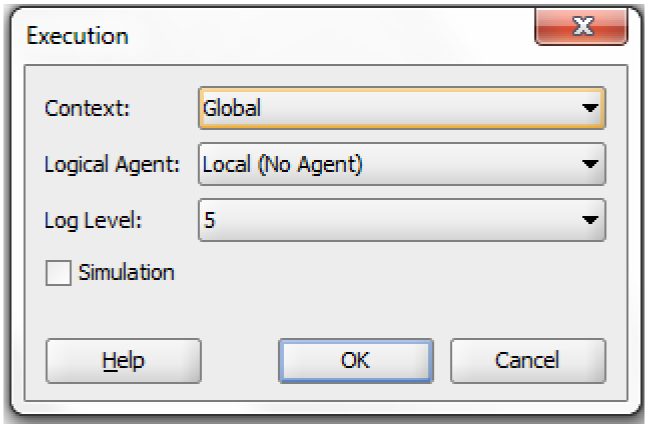
5. Link the package steps together using green arrows.





6. Click **Save( )** on the toolbar to save the package.

7. Click on the **Execute(▶)** button on the toolbar.



Execution

Context: Global

Logical Agent: Local (No Agent)

Log Level: 5

□ Simulation

8. Press OK when the **Session Started** window appears.
9. Open the **Operator** navigator. In the Operator, select the **Session List** tab, and expand the **All Executions** node.
10. Check that the last session ran correctly. You can review the steps and tasks that have Activated the CDC process

Please refer to below link for ODI Documentation for more information

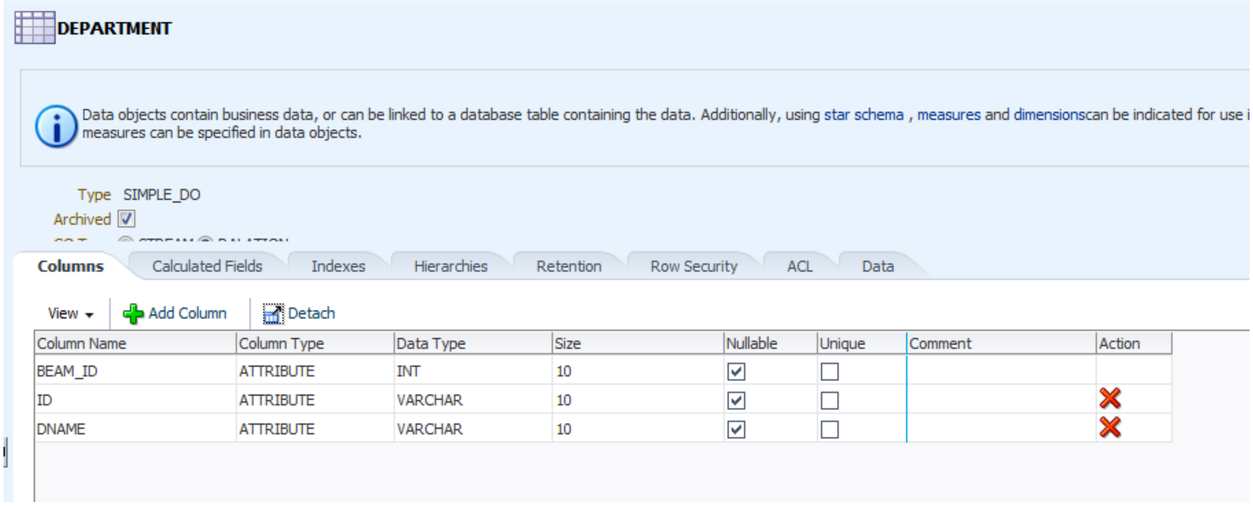
http://docs.oracle.com/cd/E14571_01/integrate.1111/e12643/data_capture.htm

7. BEAM Configuration

7.1 Create Data Objects

Create DO’s corresponding to Source tables

Create DEPARTMENT DO with following columns



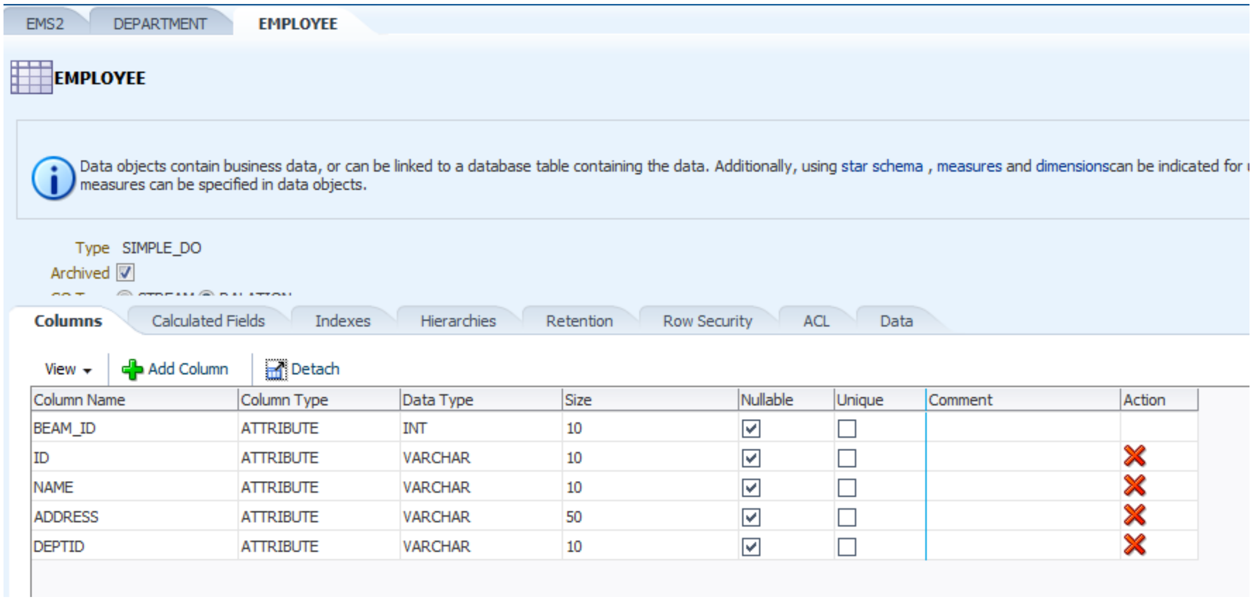
DEPARTMENT

Data objects contain business data, or can be linked to a database table containing the data. Additionally, using star schema , measures and dimensionscan be indicated for use i
measures can be specified in data objects.

Type SIMPLE_DO
Archived ☑

Column Name	Column Type	Data Type	Size	Nullable	Unique	Comment	Action
BEAM_ID	ATTRIBUTE	INT	10	☑	□		
ID	ATTRIBUTE	VARCHAR	10	☑	□		✗
DNAME	ATTRIBUTE	VARCHAR	10	☑	□		✗

Create EMPLOYEE DO with following columns



EMPLOYEE

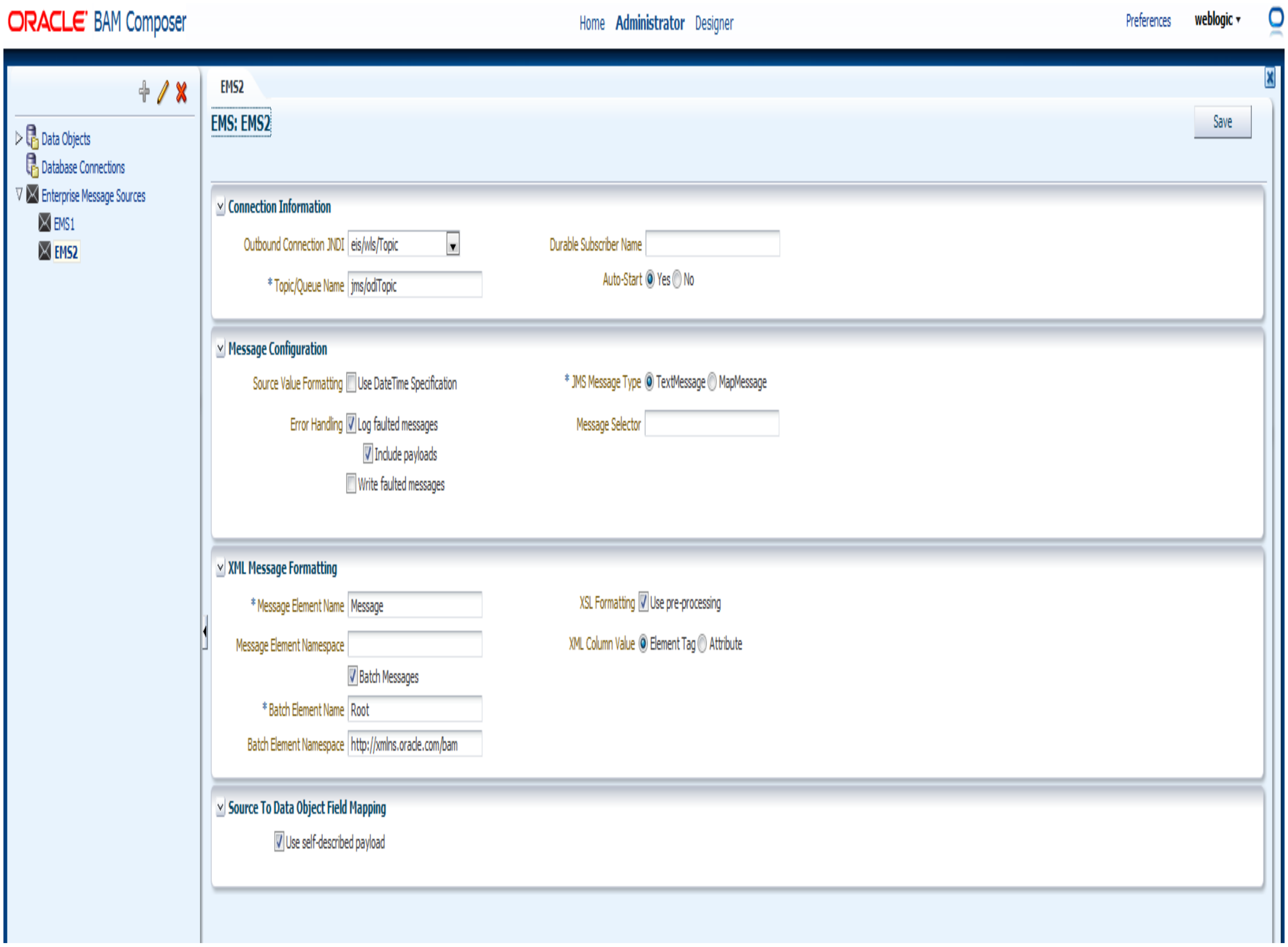
Data objects contain business data, or can be linked to a database table containing the data. Additionally, using star schema , measures and dimensionscan be indicated for
measures can be specified in data objects.

Type SIMPLE_DO
Archived ☑

Column Name	Column Type	Data Type	Size	Nullable	Unique	Comment	Action
BEAM_ID	ATTRIBUTE	INT	10	☑	□		
ID	ATTRIBUTE	VARCHAR	10	☑	□		✗
NAME	ATTRIBUTE	VARCHAR	10	☑	□		✗
ADDRESS	ATTRIBUTE	VARCHAR	50	☑	□		✗
DEPTID	ATTRIBUTE	VARCHAR	10	☑	□		✗

7.2 Configure EMS in BEAM

1. **Go to Administrator tab in BEAM Home page click on Enterprise Message Sources**



EMS: EMS2

Connection Information

Outbound Connection JNDI: eis/wls/Topic

Durable Subscriber Name:

* Topic/Queue Name: jms/odiTopic

Auto-Start ● Yes ○ No

Message Configuration

Source Value Formatting ☐ Use DateTime Specification

* JMS Message Type ● TextMessage ○ MapMessage

Error Handling ☑ Log faulted messages

☑ Include payloads

☐ Write faulted messages

Message Selector:

XML Message Formatting

* Message Element Name: Message

XSL Formatting ☑ Use pre-processing

Message Element Namespace:

XML Column Value ● Element Tag ○ Attribute

☑ Batch Messages

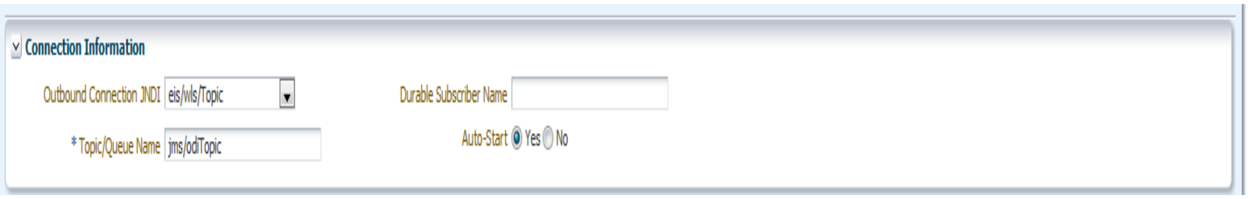
* Batch Element Name: Root

Batch Element Namespace: http://xmlns.oracle.com/bam

Source To Data Object Field Mapping

☑ Use self-described payload

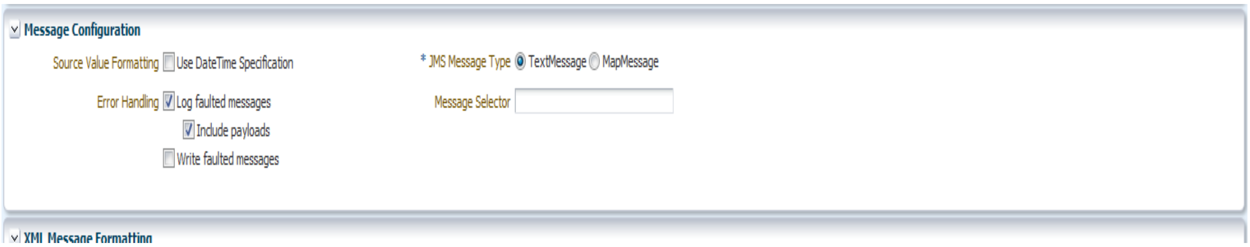
2. **Select following options then save it.**
   Connection Information:
   Outbound Connection JNDI: eis/wls/Topic
   Topic /Queue name : jms/odiTopic



Connection Information

Outbound Connection JNDI: eis/wls/Topic ▾
Durable Subscriber Name:
* Topic/Queue Name: jms/odiTopic
Auto-Start ● Yes ○ No

Message Configuration:
Error Handling→ select log faulted messages and include payloads

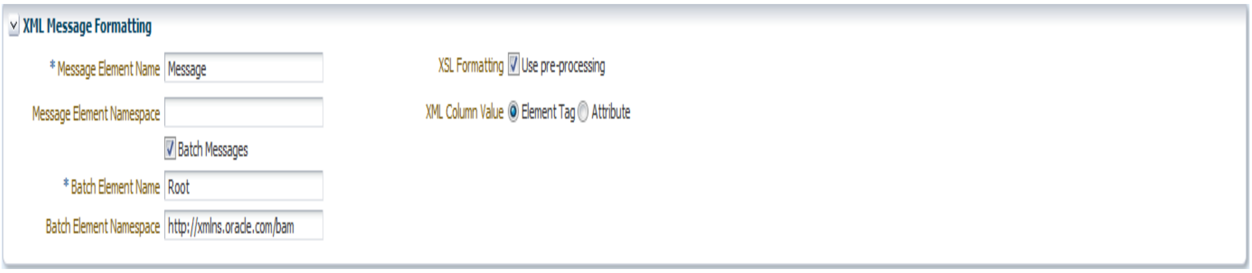


Message Configuration

Source Value Formatting □ Use DateTime Specification
* JMS Message Type ● TextMessage ○ MapMessage
Error Handling ☑ Log faulted messages
☑ Include payloads
□ Write faulted messages
Message Selector:

XML Message Formatting

XML Message Processing:
Message Element Name: Message
Batch Messages: checked
Batch Element Name: Root
Batch Element Name Space: http://xmlns.oracle.com/bam

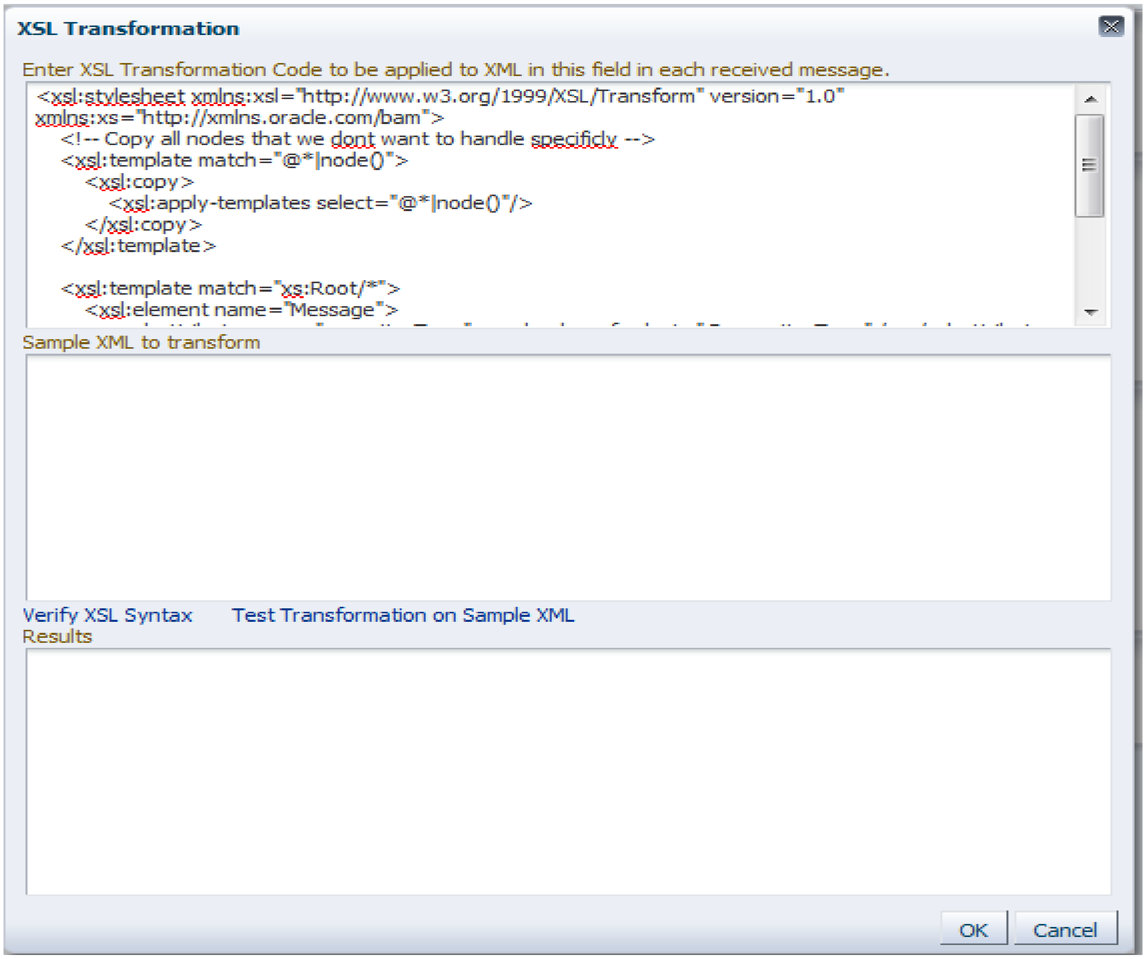


XML Message Formatting

* Message Element Name: Message
Message Element Namespace:
☑ Batch Messages
* Batch Element Name: Root
Batch Element Namespace: http://xmlns.oracle.com/bam
XSL Formatting ☑ Use pre-processing
XML Column Value ● Element Tag ○ Attribute

XSL Formatting → select use preprocessing and add following code in the popup

```
<xsl:stylesheet xmlns:xsl="http://www.w3.org/1999/XSL/Transform" version="1.0" xmlns:xs="http://xmlns.oracle.com/bam">
  <!-- Copy all nodes that we dont want to handle specificly -->
  <xsl:template match="@*|node()">
    <xsl:copy>
      <xsl:apply-templates select="@*|node()"/>
    </xsl:copy>
  </xsl:template>

  <xsl:template match="xs:Root/*">
    <xsl:element name="Message">
      <xsl:attribute name="operationType"><xsl:value-of select="@operationType" /></xsl:attribute>
      <xsl:attribute name="keys"><xsl:value-of select="@keys"/></xsl:attribute>
      <xsl:attribute name="dataObjectName"><xsl:value-of select="@dataObjectName"/></xsl:attribute>
      <xsl:apply-templates />
    </xsl:element>
  </xsl:template>
</xsl:stylesheet>
```



XSL Transformation

Enter XSL Transformation Code to be applied to XML in this field in each received message.

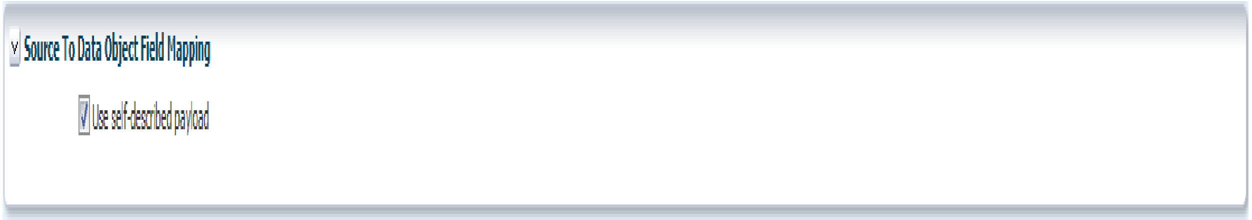
```
<xsl:stylesheet xmlns:xsl="http://www.w3.org/1999/XSL/Transform" version="1.0"
xmlns:xs="http://xmlns.oracle.com/bam">
  <!-- Copy all nodes that we dont want to handle specificly -->
  <xsl:template match="@*|node()">
    <xsl:copy>
      <xsl:apply-templates select="@*|node()"/>
    </xsl:copy>
  </xsl:template>

  <xsl:template match="xs:Root/*">
    <xsl:element name="Message">
```

Sample XML to transform

Results

Source to Data Object Field Mapping:

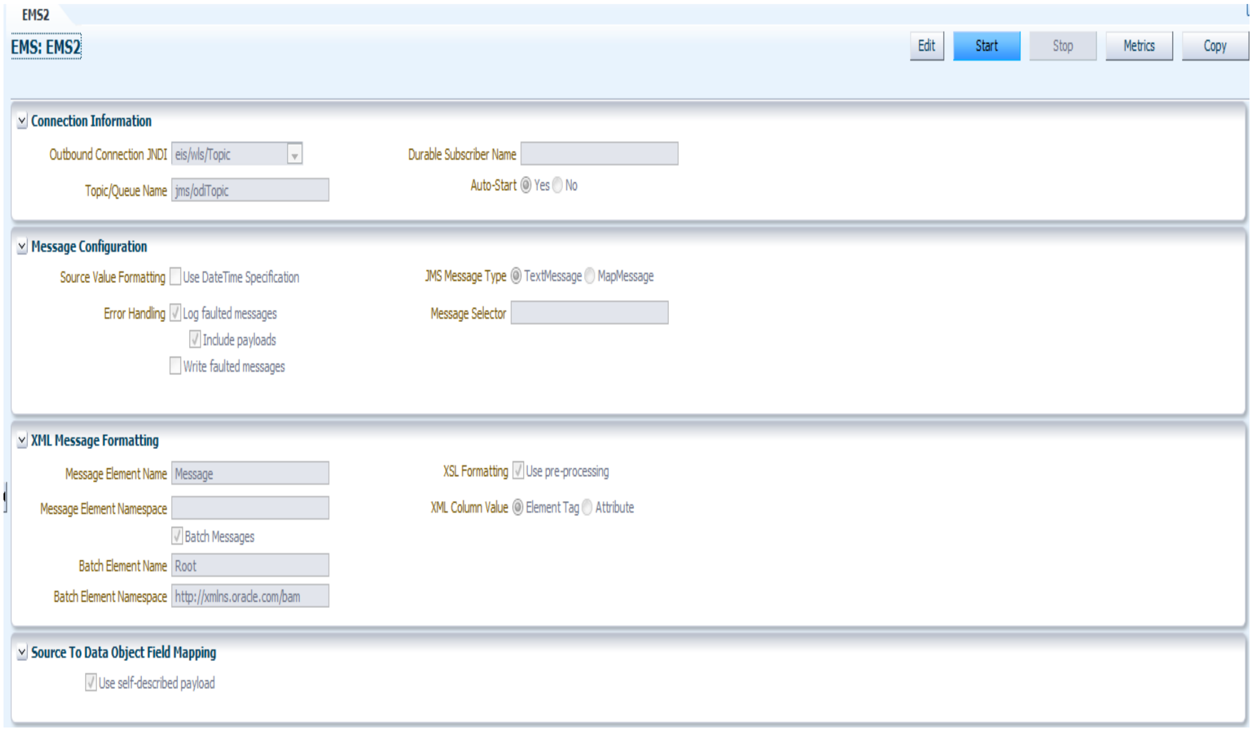
Use self described payload: checked



Source To Data Object Field Mapping

☑ Use self-described payload

3. Start EMS



EMS: EMS2

Connection Information

Outbound Connection JNDI: eis/wls/Topic

Topic/Queue Name: jms/odiTopic

Durable Subscriber Name:

Auto-Start ● Yes ○ No

Message Configuration

Source Value Formatting □ Use DateTime Specification

Error Handling ☑ Log faulted messages

☑ Include payloads

□ Write faulted messages

JMS Message Type ● TextMessage ○ MapMessage

Message Selector:

XML Message Formatting

Message Element Name: Message

Message Element Namespace:

☑ Batch Messages

Batch Element Name: Root

Batch Element Namespace: http://xmlns.oracle.com/bam

XSL Formatting ☑ Use pre-processing

XML Column Value ● Element Tag ○ Attribute

Source To Data Object Field Mapping

☑ Use self-described payload