**Error**

<detail>ORA-20001: Oracle error -20001: ORA-20001: SQL\_PLSQL\_ERROR: N, ROUTINE, MO\_GLOBAL.INIT, N, ERRNO, -20001, N, REASON, ORA-20001: SQL\_PLSQL\_ERROR: N, ROUTINE, MO\_GLOBAL.SET\_ORG\_ACCESS, N, ERRNO, -20001, N, REASON, ORA-20001: SQL\_PLSQL\_ERROR: N, ROUTINE, MO\_GLOBAL.DELETE\_ORGS, N, ERRNO, -1031, N, REASON, ORA-01031: insufficient privileges has been detected in fnd\_global.initialize[fnd\_init\_sql]. ORA-06512: at "APPS.XYZ\_PK", line 1602 ORA-06512: at "APPS.XYZ\_PK", line 1511 ORA-06512: at "XYZ.XYZ\_PROC", line 9 ORA-06512: at line 1 </detail>

**Solution**

The issue with the EBS.

**How to make your File Adapter pick only one file at a time from a location**

In SOA 11g, you use File adapter to read files from the given location.

With this read operation it picks all the files at time.

You want to configure File Adapters that it should pick one file at time from the given location with given polling interval.

**Solution:**

You set the "SingleThreadModel" and "MaxRaiseSize" properties for your file adapter.

Edit the adapter's jca file and add the following properties:

property name="SingleThreadModel" value="true"

property name="MaxRaiseSize" value="1"

**DBAdapter - Merge operation**

If you use merge operation for updating records using DBAdapter, make sure property DetectOmission is set to false as by default it's true. Or else values coming as Null will not get udpated in database records - causing updates buggy.

**Transaction Handling in the same composite**

There are scenarios when we want to complete commit or complete rollback. This is known as transaction handling.

To demonstrate this scenario, I am taking following use case.

Use Case: Read a file and insert data into Header and Line tables. Data must be inserted into both the tables or it should be complete roll back in case of any exception.

For handling transactions, you must use a XA datasource. You can create a XA datasource using admin console.

There are two cases for transaction handling.

First Case :

If you are not using catch all, transaction will automatically rollback in case of any exception. It will throw an exception and complete transaction will be rollback.

Second Case :

If you are using a catch all, then transaction will not be rolled back automatically in case of any exception. Add a catch all activity in the BPEL service.

To resolve this issue, you need to use a rethrow activity at the end of your catch all block.

**Encode to base64 binary.**

<extensionActivity>

<bpelx:exec name="JavaEmbeddingToDecode">

<![CDATA[try{

String input = (String) getVariableData("VarMessage");

oracle.soa.common.util.Base64Encoder encoder = new oracle.soa.common.util.Base64Encoder();

java.lang.String encoded = null;

encoded = encoder.encode(input);

setVariableData("EncodedStrVar",encoded);

} catch(Exception e){

e.printStackTrace();

addAuditTrailEntry(e);

}]]>

</bpelx:exec>

</extensionActivity>

**Decode bsae64 binary**

<extensionActivity>

<bpelx:exec name="JavaEmbeddingToDecode">

<![CDATA[try{

String encodeddata = (String)getVariableData("Base64Var");

oracle.soa.common.util.Base64Decoder decoder = new oracle.soa.common.util.Base64Decoder();

String decoded = null;

decoded = decoder.decode(encodeddata);

setVariableData("DecodedStrVar",decoded);

} catch(Exception e){

e.printStackTrace();

addAuditTrailEntry(e);

}]]>

</bpelx:exec>

</extensionActivity>