OBIA 12c: How to Configure OBIA Load Plan with Oracle Autonomous Database (ADB) Preconfigured Database Services

Table of Contents

Introduction:

Oracle Autonomous Database comes with preconfigured database services like, HIGH, MEDIUM, LOW etc... These services control the priority of the sessions when the system is under resource pressure and some extent control the parallel degree used.

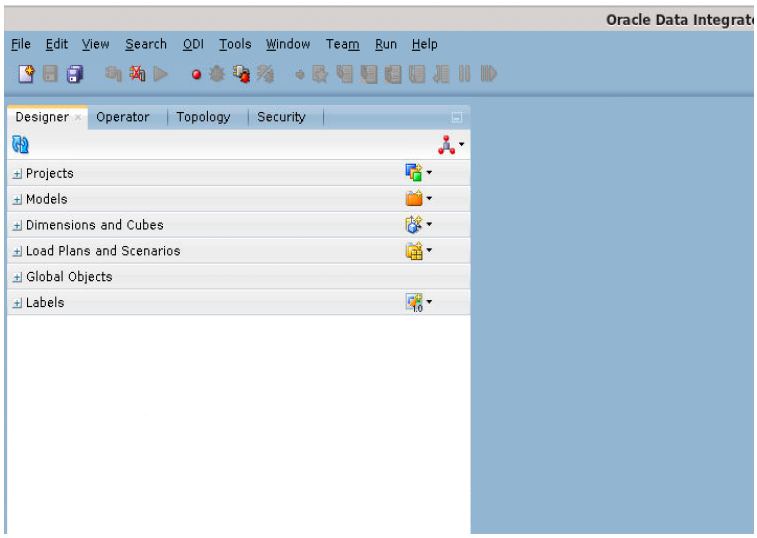
Out-of-the-box, Oracle Business Intelligence Applications 11.1.1.10.3 Patch Set 3 support ONLY Oracle Autonomous Database (ADB) preconfigured database service "LOW". However, if you would want to use other preconfigured database services other than "LOW" for a specific OBIA load plan scenario / group of OBIA load plan scenarios then this article would provide you an overview on how to implement & configure OBIA load plan scenario to use various Oracle ADB preconfigured database service while OBIA Load plan execution takes place.

Implementation steps:

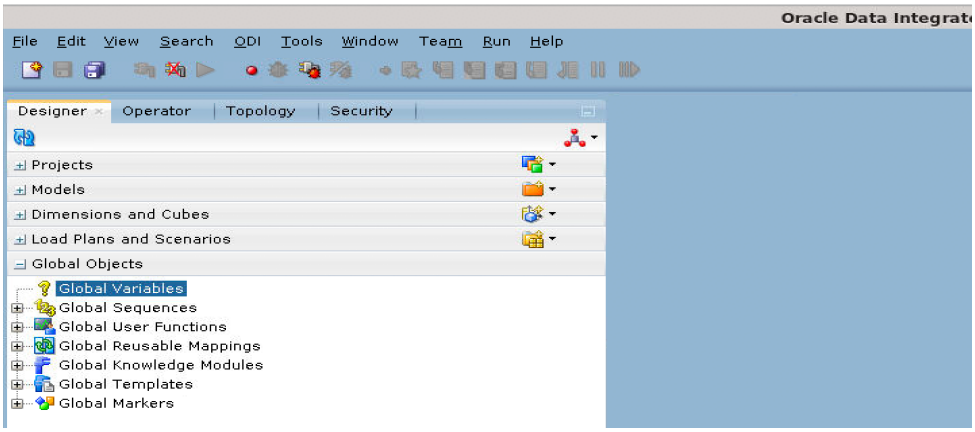
Step 1: Define a ODI variable which would hold the value of preconfigured database service
Step 2: Identify impacted ODI artifacts and make necessary changes
Step 3: Modify ODI Topology to make ADB service as configurable
Step 4: Customize the load plan to configure ADB service as configurable
Step 5: Validate the Load Plan execution results

Detailed Technical Implementation Instructions:

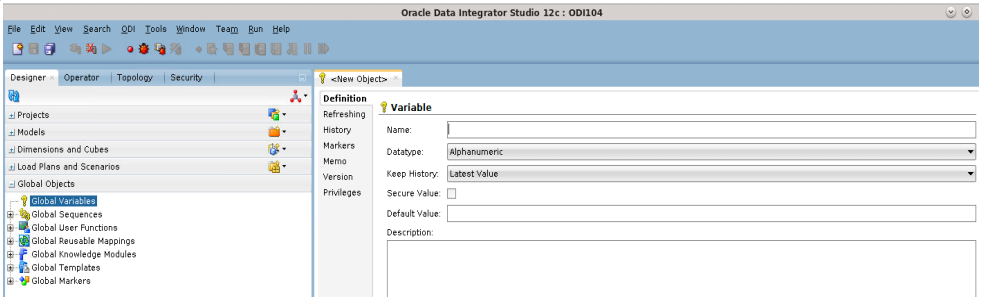
1. Create ADW_SERVCE global variable
   Login to ODI and Go to Designer



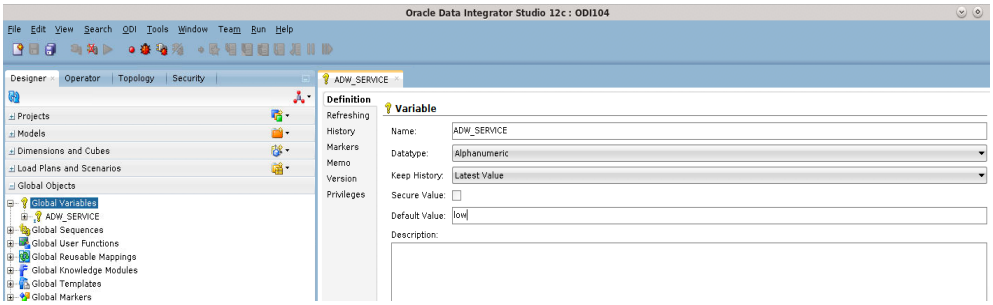
- Expand the "Global Objects"



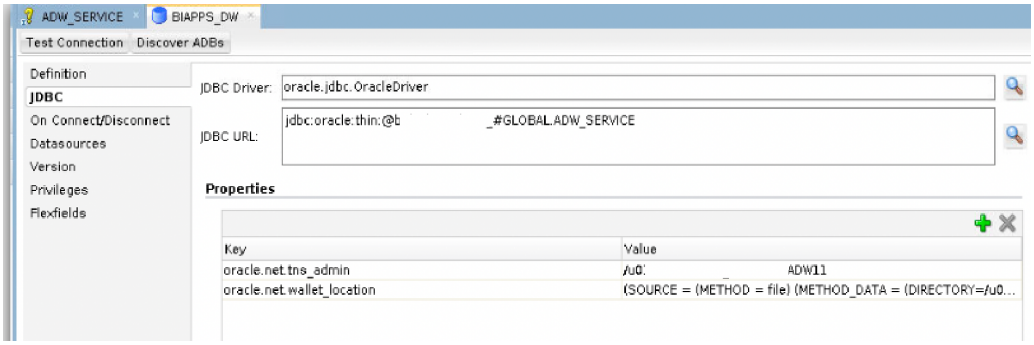
- Right click on Global Variables and then click on New Variable



- Type the name as “ADW_SERVICE” and add the default value as “low” and then click on save button



- Now add **ADW_SERVICE** variable to the JDBC url of the ODI Data server as shown below.

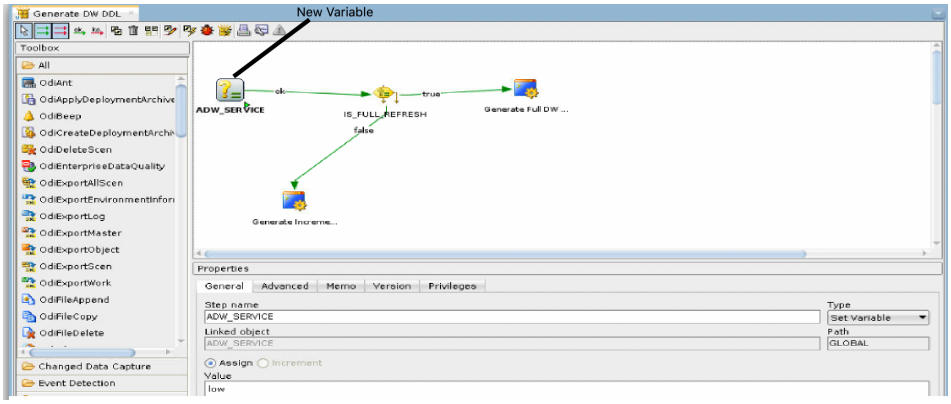


- Add ADW_SERVICE variable as set variable to the below ODI procedures. Once after making changes, regenerate the corresponding Procedure Package scenario.

- Generate DW DDL
- Generate SDS DDL
- Log ETL Run Status
- GENERATE_FAILED_TASKS_REPORT
- Load Datastore Flex Metadata
- UTILITIES_CLEAN_WORK_AND_FLOW_TABLES
- Cloud Connector Invoke Web Service
- Diagnostics Cloud Connector
- Load and Control Healthcheck
- TABLE_MAINT_PROC

Follow the below sample process to add variable to all the above procedures.

Go to Designer- > Expand the BI Apps Project -> Expand the Components -> Expand the DW -> Expand the Oracle -> Expand the Generate DW DDL -> Double click on Generate DW DDL Package -> Drag and drop the ADW service variable -> Set the value as "low" -> Mark ADW_SERVICE as "First Step" -> Then save and regenerate the scenario of the Procedure.



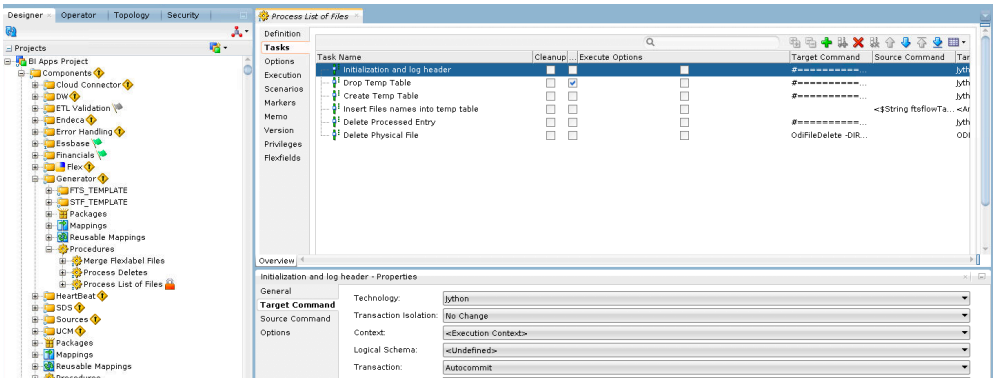
Note: Repeat the same process for all the above procedures.

2. Changes that are needed for Fusion Adaptor (For an Example)

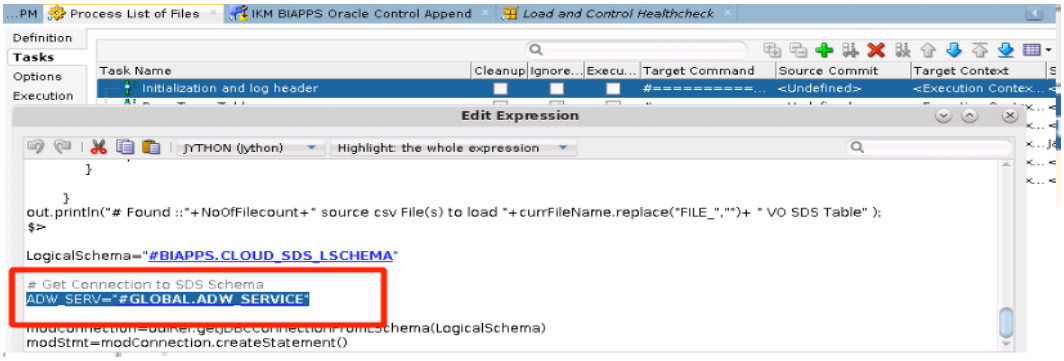
Note: Step No. 2 is applicable only for Fusion Adaptor. Ignore this if you are implementing the solution for Non-Fusion Adaptor such as EBS, PSFT.

Follow the below step to modify the procedure.

- Go to Designer- > Expand the BI Apps Project -> Expand the Components ->Expand the Generator -> Expand the procedures -> Double click on Process List of Files -> Go to Tasks



- Open the task name “Initialization and log header” and add the ADW_SERVICE variable as shown below. Then click on ok and save.



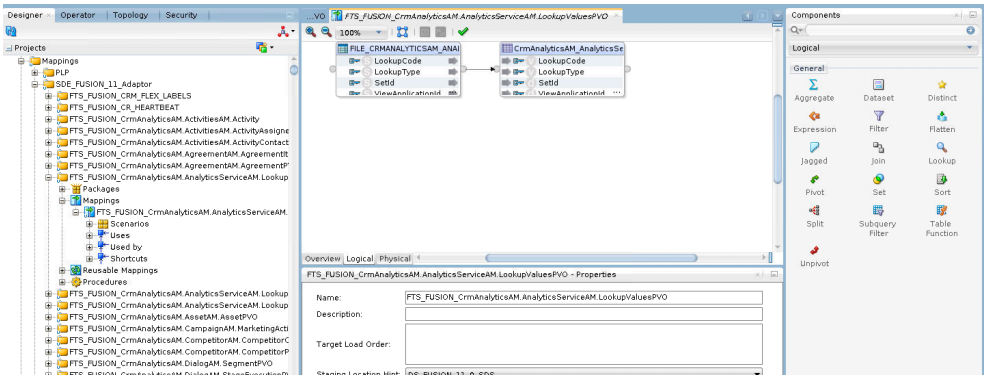
3. Changes that are needed for ODI Knowledge Modules

   To execute a scenario in HIGH or MEDIUM service, first you would need to identify the IKM, which is used in the mapping and then do the following changes.

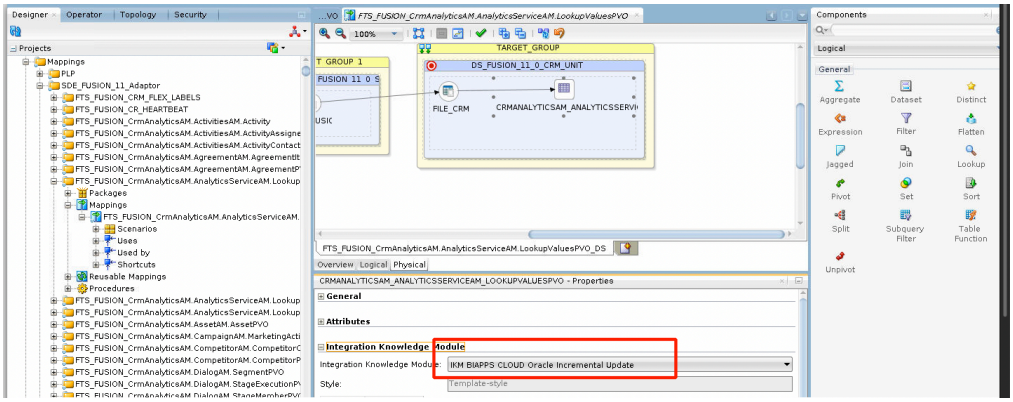
   Suppose you want to use high/medium service for the one of the FTS Map (FTS_FUSION_CrmAnalyticsAM.AnalyticsServiceAM.LookupValuesPVO)

   Follow the steps to modify the map to achieve high/medium service.

   - Go to the Designer-> Expand the BI Apps Project - > Expand the Mappings -> Expand the SDE_FUSION_11_Adaptor -> Expand the FTS_FUSION_CrmAnalyticsAM.AnalyticsServiceAM.LookupValuesPVO Folder-> Expand the Mappings -> Double click on Mapping (FTS_FUSION_CrmAnalyticsAM.AnalyticsServiceAM.LookupValuesPVO)



- Click on Physical -> Click on the Target_GROUP -> Get the IKM Name as highlighted below.

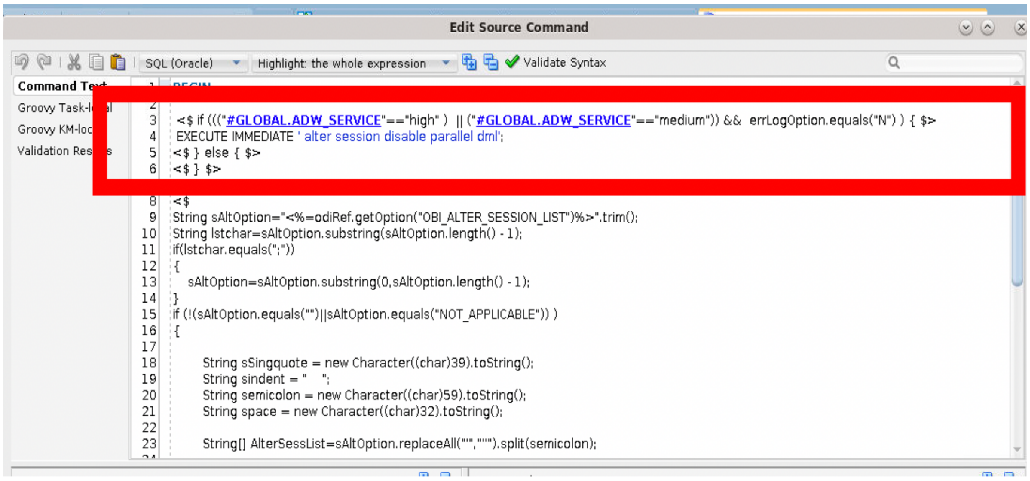


- Once you get the KM name from the mapping which you want to run using high or medium ADW service, modify the IKM as shown in below steps. Please make sure you follow the OBIA 11.1.1.10.3 customization guidelines to modify any existing or new ODI Objects.

- Create the Copy of the IKM and rename the KM from IKM BIAPPS CLOUD Oracle Incremental Update to CUSTOM_IKM BIAPPS CLOUD Oracle Incremental Update
- Go to Run Alter Session1 Commands Task of the KM and add the below logic. Then click on OK and save.

  Code snippet:

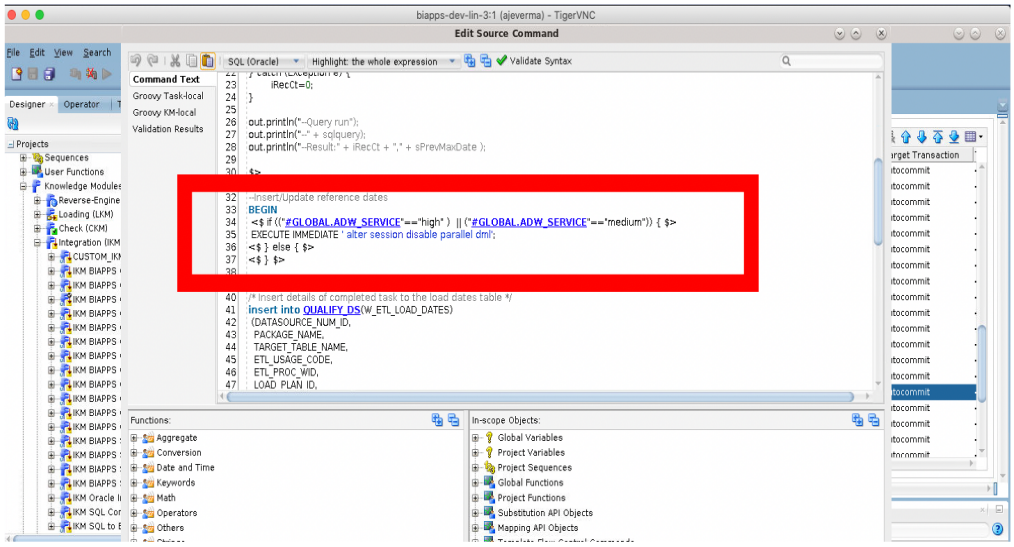
```
BEGIN
<$ if (("#GLOBAL.ADW_SERVICE"=="high" )  ||
("#GLOBAL.ADW_SERVICE"=="medium")) { $>
 EXECUTE IMMEDIATE ' alter session disable parallel
dml';
<$ } else { $>
<$ } $>
```



- Go to step Insert reference dates and add the additional logic as shown in below screen

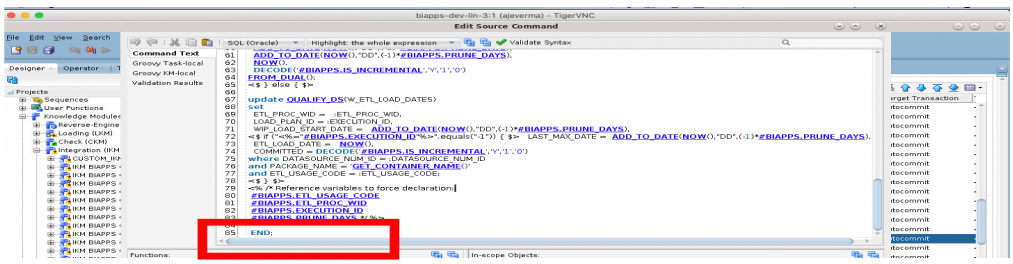
Code snippet:

```
BEGIN
<$ if (("#GLOBAL.ADW_SERVICE"=="high" )  ||
("#GLOBAL.ADW_SERVICE"=="medium")) { $>
 EXECUTE IMMEDIATE ' alter session disable parallel
dml';
<$ } else { $>
<$ } $>
```



Code snippet:

```
END;
```

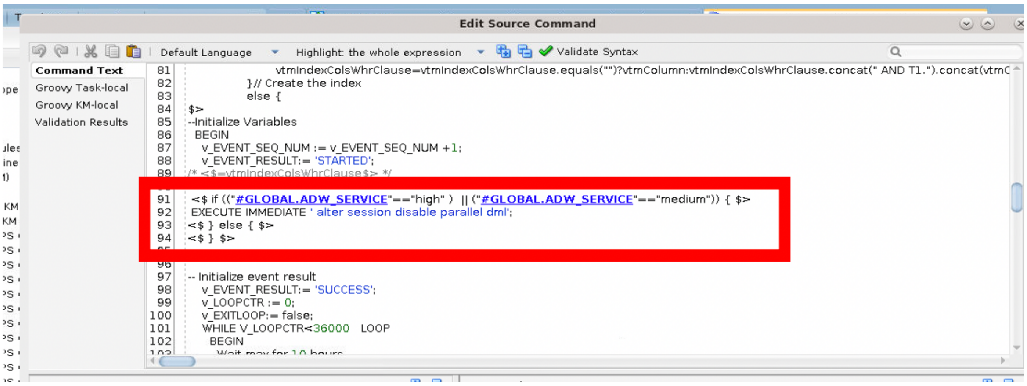


Also add semicolon (;) at the end as highlighted below

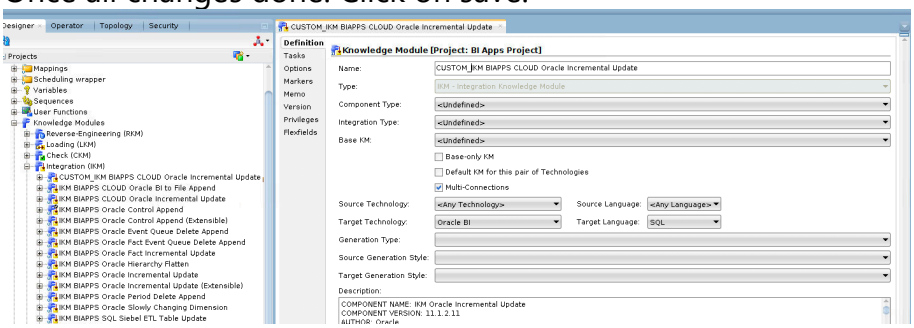
- Go to step Diagnostics - Find Dupes in Full Load and Autocorrect and Add disable parallel dml logic as show in below screen

Code snippet:

```
BEGIN
<$ if (("#GLOBAL.ADW_SERVICE"=="high" )  ||
("#GLOBAL.ADW_SERVICE"=="medium")) { $>
 EXECUTE IMMEDIATE ' alter session disable parallel
dml';
<$ } else { $>
<$ } $>
```



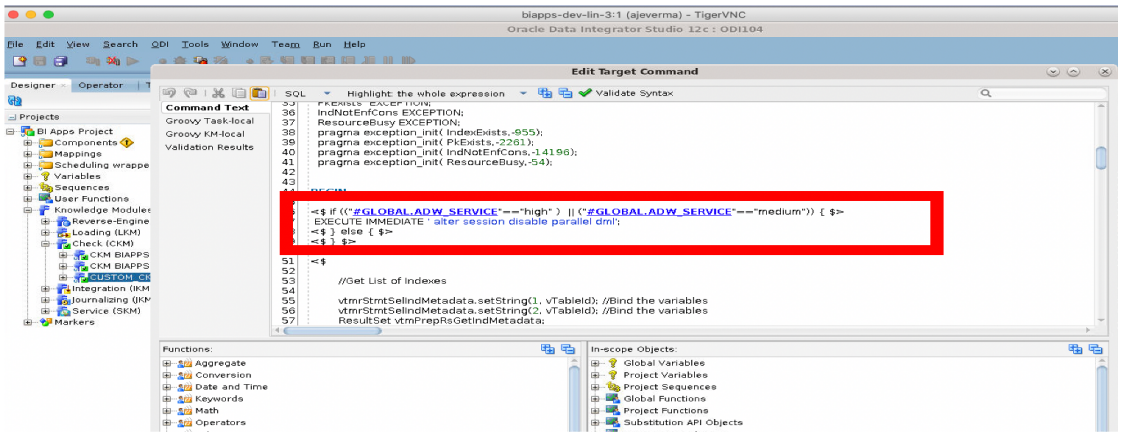
- Once all changes done. Click on save.



- Similar steps must be followed incase if you want to implement high or medium for any of the mapping.

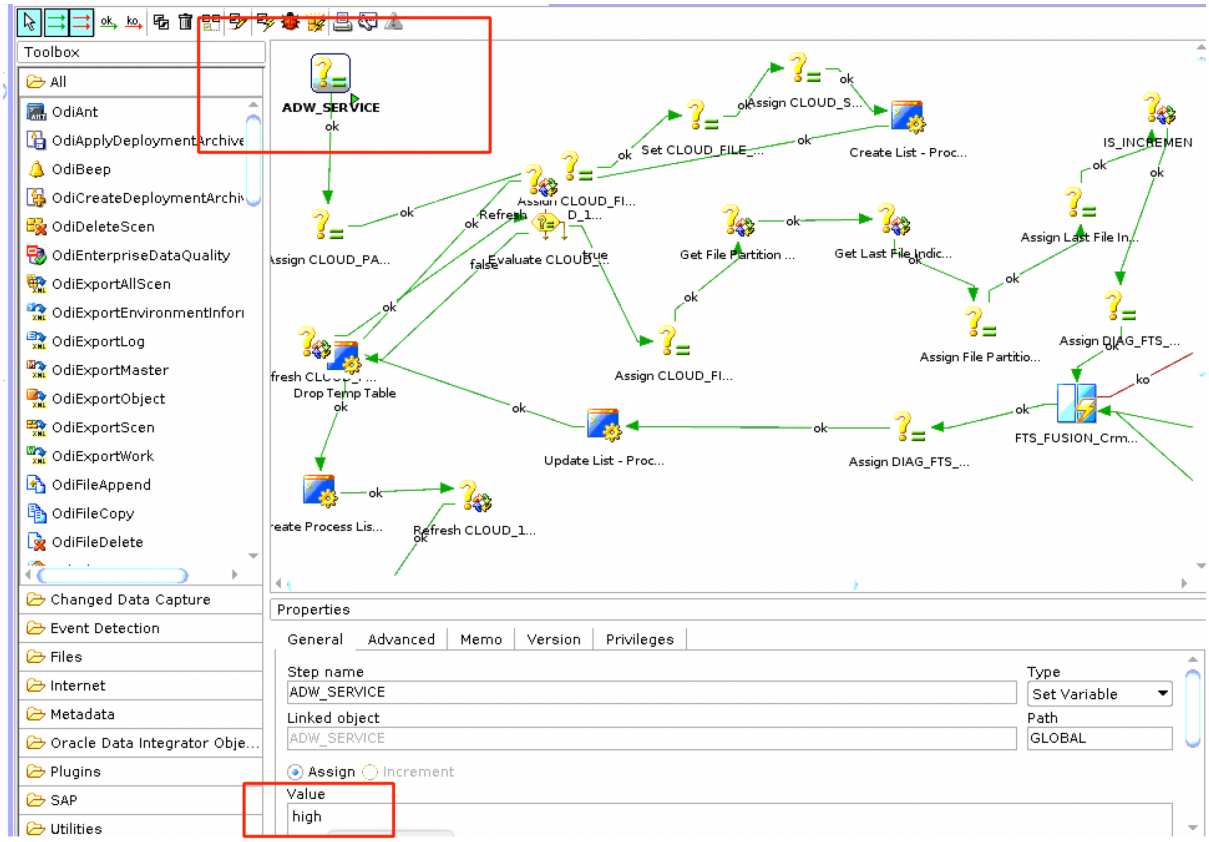
➢ Once KM IKM changes done, now go and modify the CKM. Please follow the customization guidelines as per OBIA 10.3.

- Create copy of the CKM (CKM BIAPPS Oracle)
- CUSTOM_CKM BIAPPS Oracle
  Add the logic to step "*Diagnostics - Find Dupes,Cnstrnt Vltn rows on I$*" of CKM.
- Save the changes.



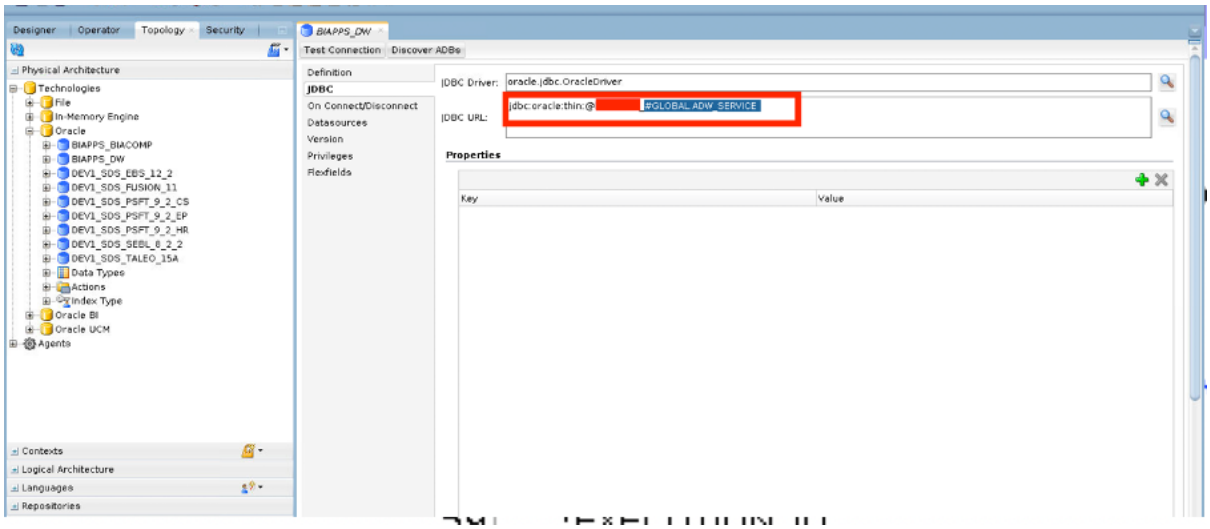
➢ Add the new variable to respective mapping package

- Add ADW_SERVICE variable to the package for which you want to perform the load using high/medium service.
- Save and regenerate the Package.
- Repeat the Step3 for any other maps of BIA adaptor.



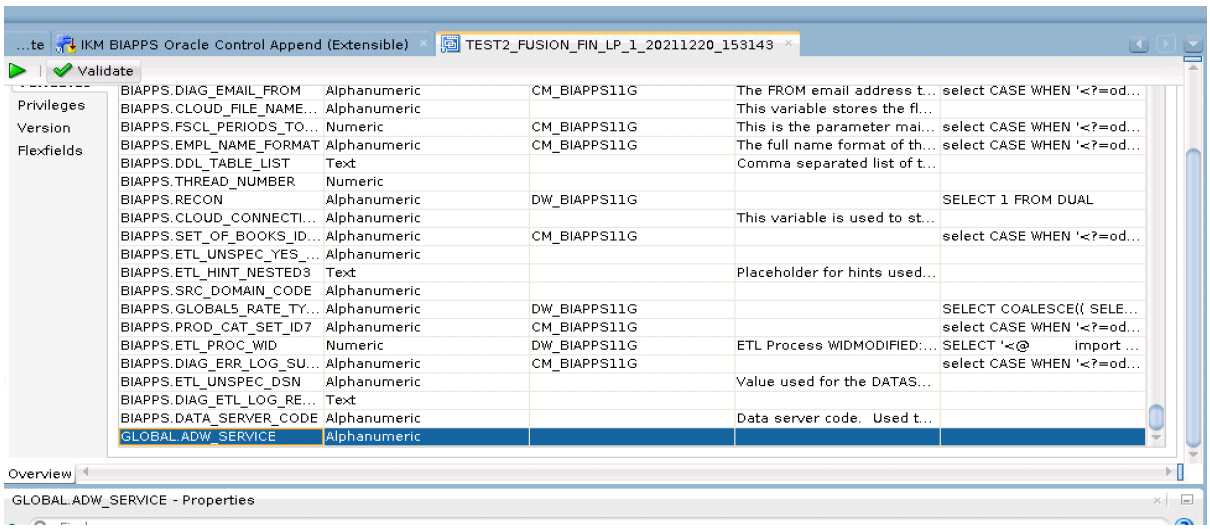
Note: Incase you are implementing this solution for Fusion Maps then you need to regenerate all FTS maps scenario.
For Non-Fusion Maps, No need to regenerate all the maps, only the impacted one should be regenerated.

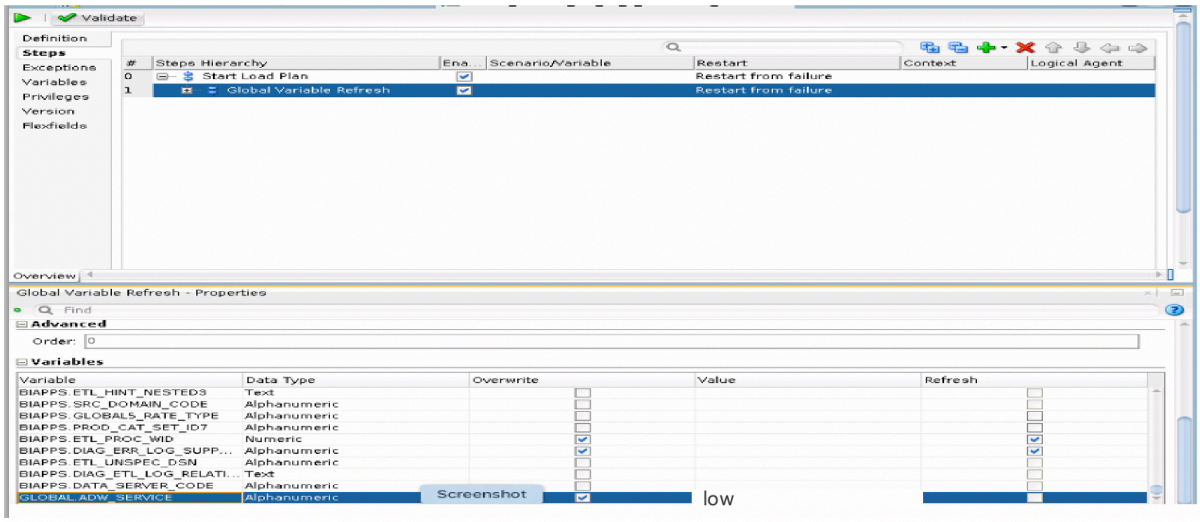
4. Add the new variable to DW/SDS Physical Topology
In order to pass the value dynamically use ADW_SERVICE variable in JDBC url properties.



5. Customize the Load Plan

Add “ADW_SERVICE” variable to the load plan and assign the value as “high” or “medium” or “low”.

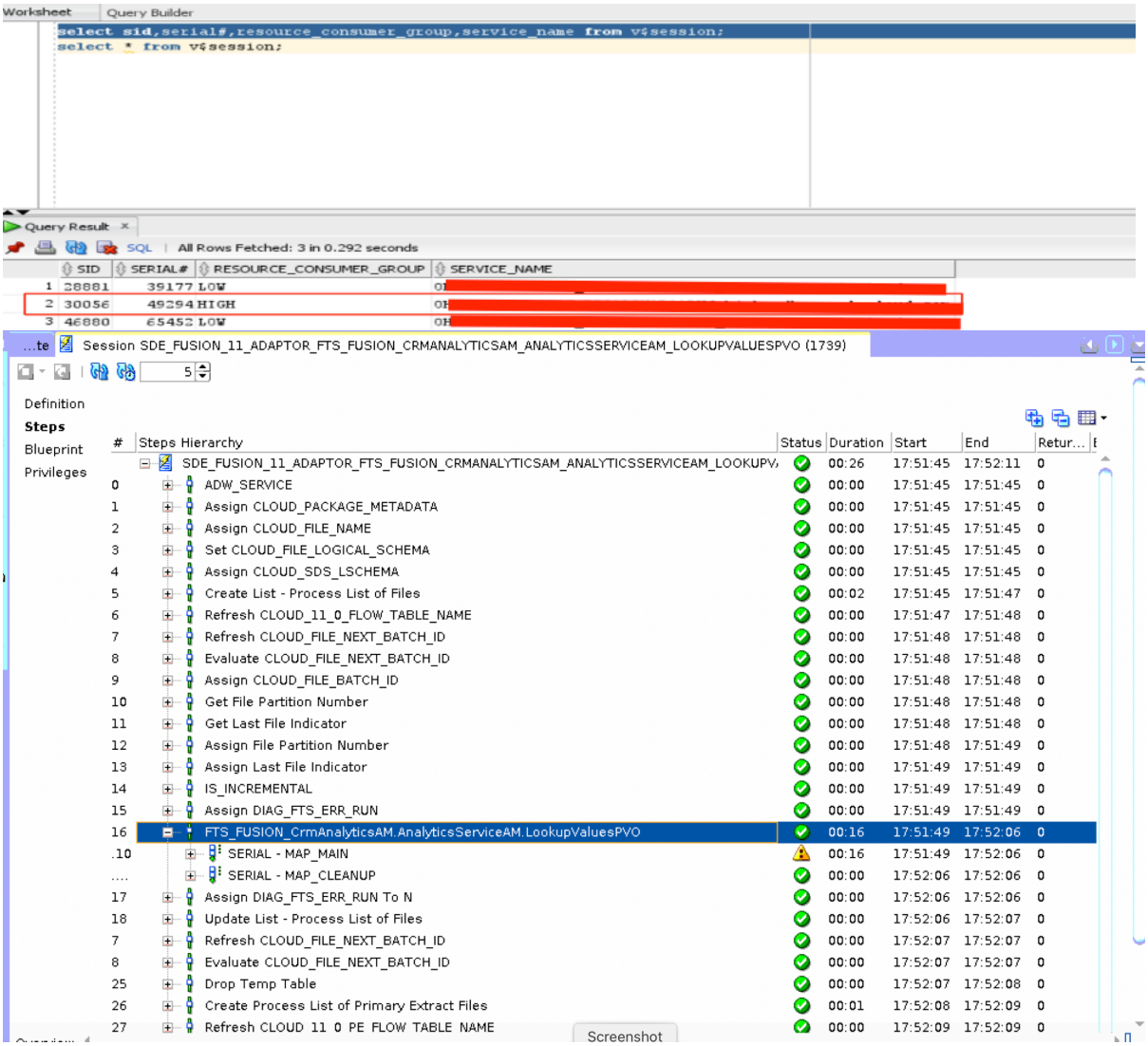




Click on save and run the Load Plan.

6. Verify the loads.
   Please execute the below command to see if the load is running in high/medium/low service.

```
Select * from v$session;
```



References

- https://docs.oracle.com/en/cloud/paas/autonomous-database/adbsa/service-names-data-warehouse.html#GUID-80E464A7-8ED4-45BB-A7D6-E201DD4107B7
- https://docs.oracle.com/en/cloud/paas/autonomous-database/adbsa/manage-service-concurrency.html#GUID-C286FAFA-7FFE-4722-A7E5-DC779224E7A4