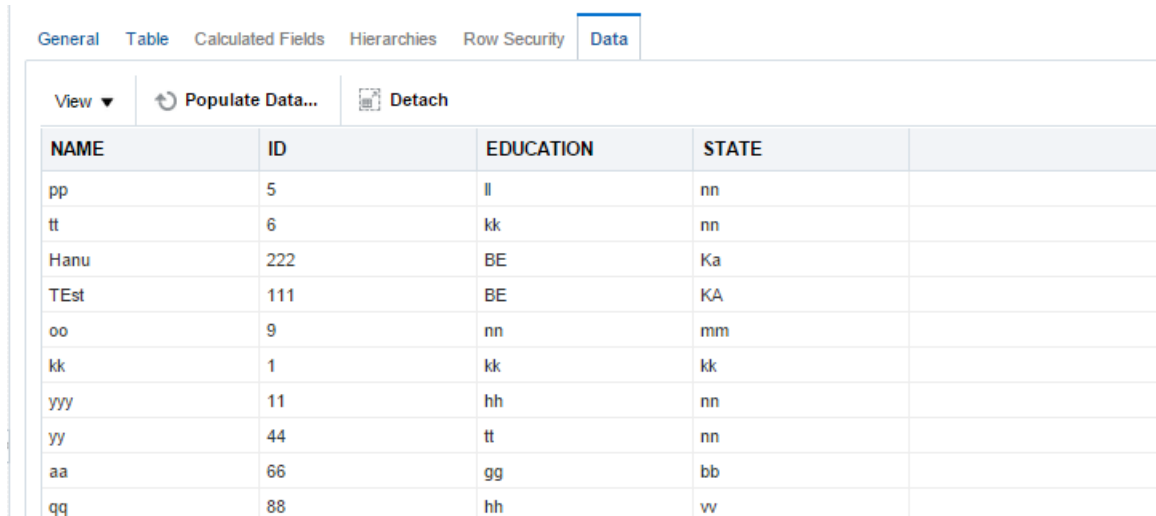


BAM 12c Custom Alert Action To Populate Data to External DIM Data object

Overview:

Currently user can populate the data to external dimensional data object manually by clicking on “populate Data”

User has to go to external Data Object data tab and need to click on “Populate Data” to fetch data from an external table



NAME	ID	EDUCATION	STATE
pp	5	ll	nn
tt	6	kk	nn
Hanu	222	BE	Ka
TEst	111	BE	KA
oo	9	nn	mm
kk	1	kk	kk
yyy	11	hh	nn
yy	44	tt	nn
aa	66	gg	bb
qq	88	hh	vv

We need feature that will populate the data in external dimensional data object periodically.

Using custom alert action, user can create custom alert action which can trigger periodically and populate external data object.

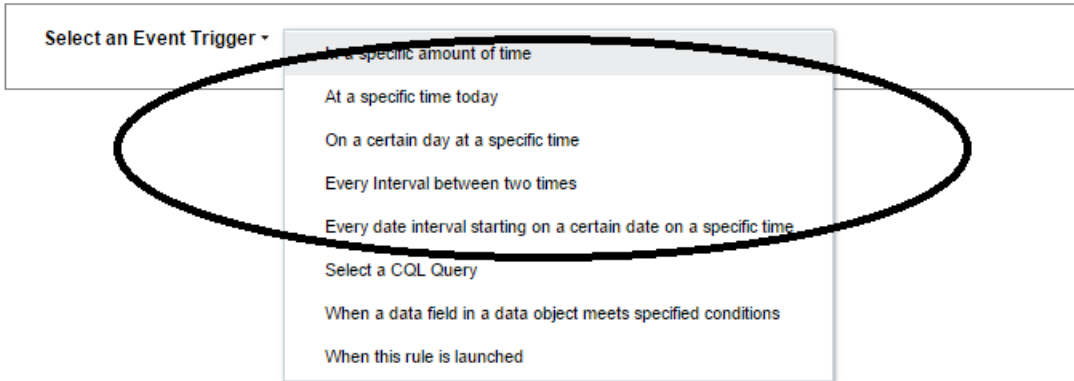
We have created custom action library (jar) which can be used to create alert action.

User needs to copy the jar (I.e... populate-external-do-customaction.jar) to the server environment and need to specify full path in custom action.

Create Custom Alert Action:

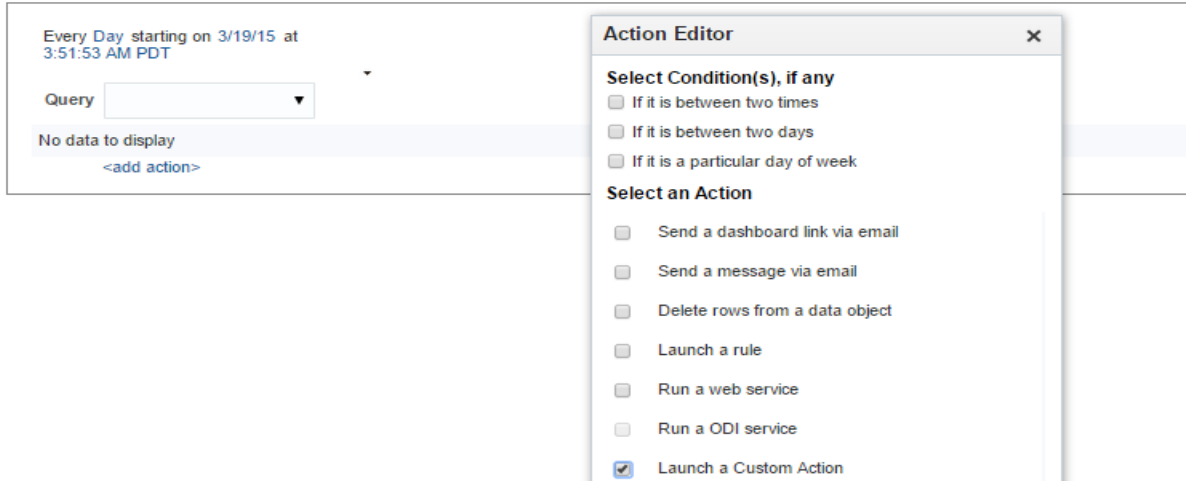
- 1) Go to designer page, click on create Alert(give alert name & display name)
- 2) You can create custom action for any time related Event triggers

Alert Display Name : PopulateExternalDO / Alert : PopulateExternalDO

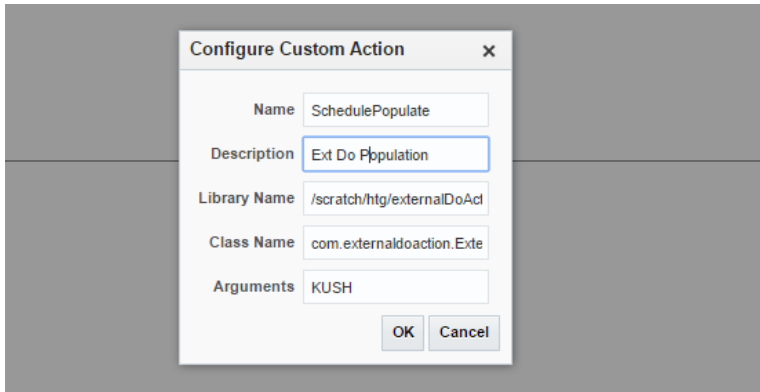


- 3) To populate data periodically create alert with event “Every date interval starting on certain date on a specific time”
- 4) Specify date & time and click on add action and choose “Launch a Custom Action”

Alert Display Name : PopulateExternalDO / Alert : PopulateExternalDO



- 5) Click on custom action and configure custom action



Specify following details

- ✓ Name: Name for custom action
- ✓ Description: Description
- ✓ Library Path: Specify full path of copied jar

For Ex: /scratch/htg/externalDoAction/deploy/populate-external-do-customaction.jar

- ✓ Class Name: Class name of custom action
i.e.. com.externaldoaction.ExternalDoPopulateAction

- ✓ Arguments: <DO NAME> Specify the external data object name to be populated

6) Save the Alert

Custom Alert created with above options will periodically trigger the action and populate data to external data object.

Check the data tab of external data object to verify the data is getting populated on the time intervals specified in the alert action.

Details of Custom Alert action Library:

Custom Alert action library as the following custom alert action class which uses config & common query API's to populate the external data object

```
package com.externaldoaction;
import java.util.HashMap;
import javax.naming.InitialContext;
import oracle.beam.common.alertsengine.externalactions.IAlertExternalAction;
import oracle.beam.common.client.service.composer.BeamComposerClient;
import oracle.beam.commonquery.CommonQueryService;
import oracle.beam.commonquery.ExternalDOQueryService;
import oracle.beam.config.common.ConfigSession;
```

```

import oracle.beam.config.model.dataobject.ExternalDataObject;

public class ExternalDoPopulateAction implements IAlertExternalAction {
    public ExternalDoPopulateAction() {
        super();
    }

    @Override
    public void initialize(String param) {
        try{
            ExternalDataObject dataObject = getDataObject(param);
            populateExternalDataObject(dataObject);
        }catch(Exception ex){
            ex.printStackTrace();
        }
    }

    @Override
    public void takeAction(HashMap hashMap, String param1, HashMap<String, Object> hashMap2,
        HashMap<String, String> hashMap3) throws Exception {
    }

    private ExternalDataObject getDataObject(String name) throws Exception {
        BeamComposerClient oComposerClient = new BeamComposerClient();
        ConfigSession configSession = oComposerClient.getEjbSession();
        ExternalDataObject dataobject = (ExternalDataObject)
configSession.findDataObjectByName(name);
        return dataobject;
    }

    private void populateExternalDataObject(ExternalDataObject doName) throws Exception{
        InitialContext oInitCtx = new InitialContext();
        CommonQueryService oQueryService =
(CommonQueryService)oInitCtx.lookup("OracleBeam#oracle.beam.commonquery.CommonQueryService");
        //Call the External DO Service to copy the data from External DO to Simple DO.
        ExternalDOQueryService oExternalDOQueryService =
oQueryService.getNewExternalDOQueryService();
        oExternalDOQueryService.populateExternalDimDO(doName);
    }
}

```