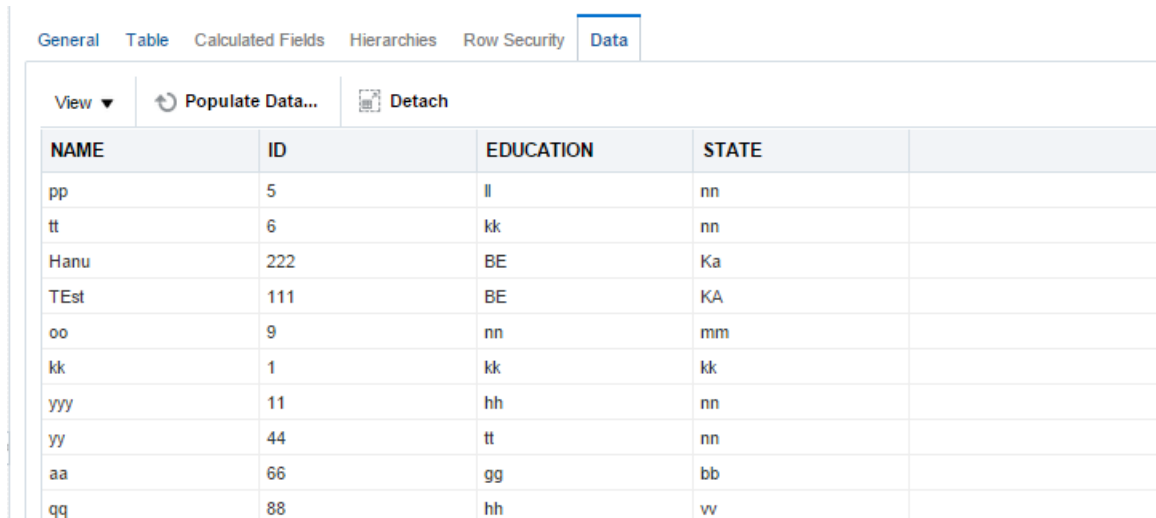


## Oracle Business Activity Monitoring (Oracle BAM)12cのカスタム・アラート・アクションを使用した外部ディメンション・データ・オブジェクトへのデータ移入

概要：

現在、外部のディメンション・データ・オブジェクトへのデータ移入は、「Populate Data」をクリックして手動で行います。

ユーザーは、外部データ・オブジェクトのDataタブに移動し、「Populate Data」をクリックして外部表からデータを取得する必要があります。



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

外部ディメンション・データ・オブジェクトに定期的にデータを移入する機能が必要です。

このような場合は、カスタム・アラート・アクションを使用します。定期的にトリガーされて外部データ・オブジェクトにデータを移入するカスタム・アラート・アクションを作成することができます。

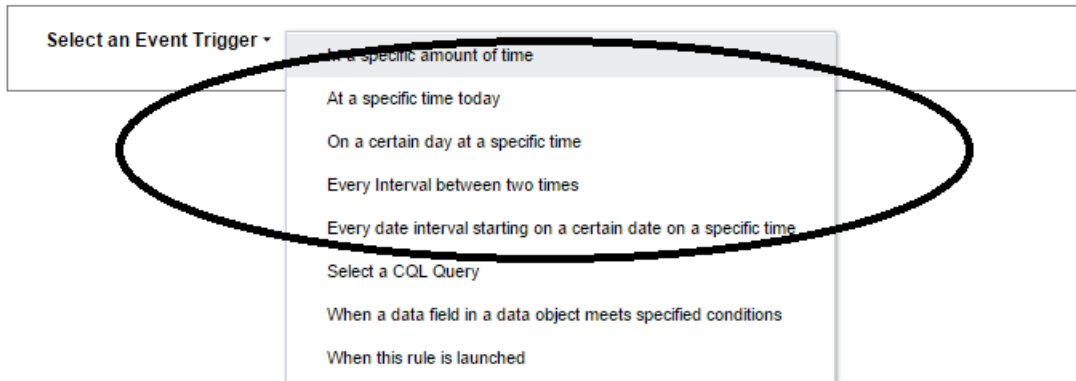
オラクルは、アラート・アクションを作成する際に利用できるカスタム・アクション・ライブラリ (jar) を作成しました。

ユーザーは、このjar (populate-external-do-customaction.jar) をサーバー環境にコピーし、カスタム・アクションでフルパスを指定する必要があります。

### カスタム・アラート・アクションの作成

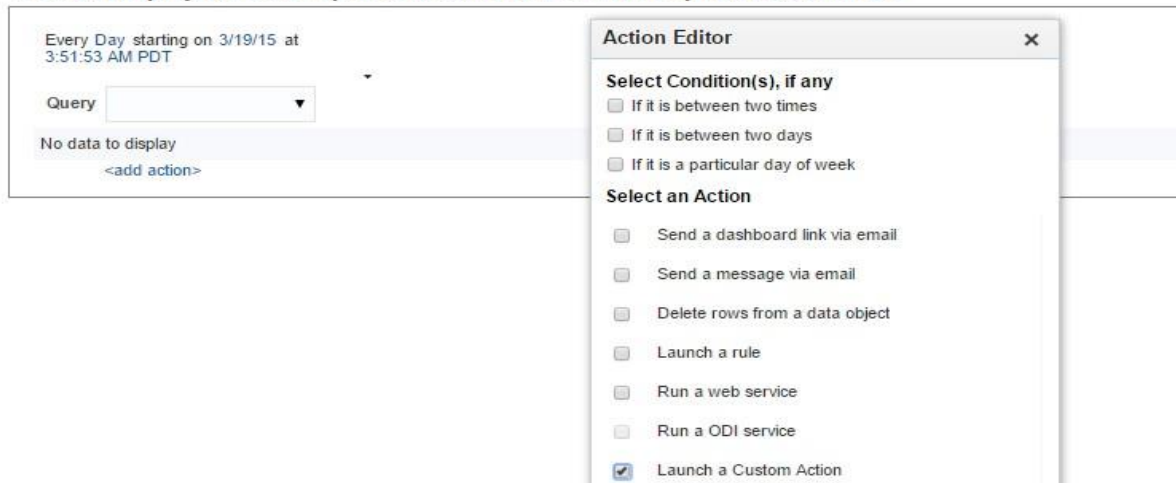
- 1) Designerページに移動し、「Create Alert」をクリックして、アラート名と表示名を指定します。
- 2) 時間に関連するすべてのイベント・トリガーにカスタム・アクションを作成できます。

☑ Alert Display Name : PopulateExternalDO / Alert : PopulateExternalDO

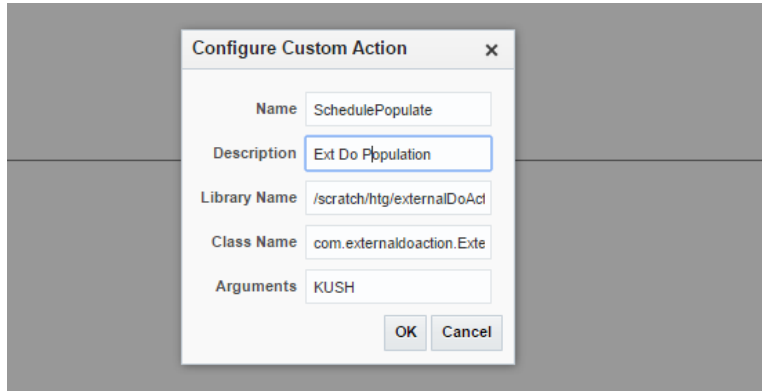


- 3) データを定期的に移入するには、"Every date interval starting on certain date on a specific time"イベントでアラートを作成します。
- 4) 日付と時刻を指定して「add action」をクリックし、「Launch a Custom Action」を選択します。

☑ Alert Display Name : PopulateExternalDO / Alert : PopulateExternalDO



- 5) 「custom action」をクリックし、カスタム・アクションを構成します。



以下の詳細情報を指定します。

- ✓ Name : カスタム・アクションの名前
- ✓ Description : 説明
- ✓ Library Name : コピーしたjarのフルパス

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

- ✓ Class Name : カスタム・アクションのクラス名

指定するクラス：com.externaldoaction.ExternalDoPopulateAction

- ✓ Arguments : データの移入先となる外部データ・オブジェクトの名前

#### 6) アラートを保存します。

上記のオプションで作成したカスタム・アラートによって、定期的アクションがトリガーされ、外部データ・オブジェクトにデータが移入されるようになります。

アラート・アクションに指定した時間間隔でデータが移入されていることを、外部データ・オブジェクトのDataタブで確認します。

### **カスタム・アラート・アクション・ライブラリの詳細**

カスタム・アラート・アクション・ライブラリには、構成用共通問合せAPIを使用して外部データ・オブジェクトにデータを移入する以下のカスタム・アラート・アクション・クラスが含まれています。

```
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);
    }
}

```