

Table of Contents

1. Report master creation	1
1.1. Introduction	1
1.2. Pre-requisite	1
1.3. Report Master based on an entity/view	1
1.4. Report Master based on a service	2
1.5. Entities diagram	4

1. Report master creation

1.1. Introduction



this feature will be officially available with the R17.12 branch first release

A report master is an OFBiz content which allows a user to generate data reports. It defines data connexion, and a general filtering form for data. It can be based on an entity, a dedicated service, or in a wider sense on any shape a data connexion can take and return back a map.

1.2. Pre-requisite

- OFBiz
- Birt plugin

1.3. Report Master based on an entity/view

1. Create or choice a database entity or view
2. Create the general filtering form within the file `plugins/birt/widget/birt/BirtMasterForms.xml`. The only informations to be changed are entity-name and form name.

```
<form name="CTNT_MASTER_EXAMPLE" type="single"
extends="AbstractFlexibleReportSearchForm">
  <auto-fields-entity entity-name="Example" default-field-type="find"/>
</form>
```

3. Add the informations about this Master in the database using the file `plugins/birt/data/BirtMasterData.xml`

```
<DataResource dataResourceId="DR_MASTER_EXAMPLE"
dataResourceTypeId="ELECTRONIC_TEXT" dataTemplateTypeId="FORM_COMBINED" />
<ElectronicText dataResourceId="DR_MASTER_EXAMPLE">
  <textData><![CDATA[<!--default domain form-->
  <form name="{masterContentId}_{contentId}" type="single"
extends="{masterContentId}" extends-
resource="component://birt/widget/birt/BirtMasterForms.xml">
    </form>]]>
  </textData>
</ElectronicText>
<Content contentId="CTNT_MASTER_EXAMPLE" contentTypeId="REPORT_MASTER"
dataResourceId="DR_MASTER_EXAMPLE" statusId="CTNT_PUBLISHED" contentName="Example"
description="Master Content for Example" />
<!-- Data retrieval will be done using perform find on entity Example-->
<ContentAttribute contentId="CTNT_MASTER_EXAMPLE" attrName="Entity"
attrValue="Example"/>
```

The form in the database is the form that will allow users to change form parameters. You can add any field you desire. Some field names are though reserved: reportContentId, overrideFilters, entityViewName, birtContentType.

4. Add in the file content/config/contentEntityLabels.xml the Property that will allow translation for your report master description.

```
<property key="Content.description.CTNT_MASTER_EXAMPLE">
  <value xml:lang="en">Example</value>
  <value xml:lang="fr">Exemple</value>
</property>
```

Your Report Master is created ! You can now create reports using it.

1.4. Report Master based on a service

In plugins/birt/src/main/java/org/apache/ofbiz/birt/flexible/BirtMasterReportServices.java there is 2 sets of dedicated services (see examples there)

1. In each set the first service, will return 4 items:
 - an object of type Map<String, String> called **dataMap**.
 - **Keys:** data field names.
 - **Values:** data types (OFBiz types).
 - an object of type Map<String, String> called **fieldDisplayLabels**.
 - **Keys:** data field names.
 - **Values:** the names displayed to the user.

This output is optional, should it be missing, the keys will be displayed.

- an object of type Map<String, String> called **filterMap**.
 - **Keys:** data filtering field names (exact names used for the form fields).
 - **Values:** data type (OFBiz type).

This output is optional, if missing, filters can not be displayed on the report.

- an object of type Map<String, String> called **filterDisplayLabels**.
 - **Keys:** data filtering field names (exact names used for the form fields).
 - **Values:** names to be displayed to the user.

This output is optional, should it be missing, the keys will be displayed.

2. The second service will actually get the data. It receives an object (Object type) called reportContext.

From this object, you can obtain the map parameters using the following code:

```
Map<String, Object> parameters =  
UtilGenerics.checkMap(reportContext.getParameterValue("parameters"));
```

This Map will give access fields of the filtering form.

This service will return a list of type List<GenericValue>, containing the data.

A Map<String, Object> would also do.

Then,

1. create the parent form in the file plugins/birt/widget/birt/BirtMasterForms.xml. Field names created here must be the names used on the Map parameters of the previous service, and also corresponding to the map filterMap.

```
<form name="CTNT_MASTER_TURNOVER" type="single"  
extends="AbstractFlexibleReportSearchForm">  
  <field name="fromDate"><date-time type="date"/></field>  
  <field name="thruDate"><date-time type="date"/></field>  
  <field name="productCategoryId"><lookup target-form-  
name="LookupProductCategory"/></field>  
  <field name="productStoreId"><lookup target-form-  
name="LookupProductStore"/></field>  
</form>
```

2. Create the master in database following.

```

    <CustomMethod customMethodId="CM_FB_TURNOVER"
    customMethodTypeId="FLEXIBLE_BIRT" customMethodName="flexibleReportTurnOver"
    description="service to resolve invoice for turnover report domain"/>
    <DataResource dataResourceId="DR_MASTER_TURNOVER"
    dataResourceTypeId="ELECTRONIC_TEXT" dataTemplateTypeId="FORM_COMBINED" />
    <ElectronicText dataResourceId="DR_MASTER_TURNOVER">
        <textData><![CDATA[<!--default domain form-->
            <form name="{masterContentId}_{contentId}" type="single"
            extends="{masterContentId}" extends-
            resource="component://birt/widget/birt/BirtMasterForms.xml">
                </form>]]>
        </textData>
    </ElectronicText>
    <Content contentId="CTNT_MASTER_TURNOVER" customMethodId="CM_FB_TURNOVER"
    contentTypeId="REPORT_MASTER" dataResourceId="DR_MASTER_TURNOVER"
    statusId="CTNT_PUBLISHED" contentName="Turnover" description="Master Content for
    TURNOVER domain" />
    <!-- Data retrieval will be done using two service calls. First the
    contentAttribute Service gives the service that will define which data and label
    will be retrieved,
        and which filter and label are supported by the report design (default value
    will call the second service with "prepareField" suffix).
        Second, the custom method gives the service to retrieve all data in the report
    design.
        Here : flexibleReportTurnOverPrepareFields (customMethodName + "prepareFields")
    then flexibleReportTurnOver-->
    <ContentAttribute contentId="CTNT_MASTER_TURNOVER" attrName="Service"
    attrValue="default"/>

```

The form in the database is the form that will allow users to change form parameters. You can add any field you desire. Some field names are reserved: reportContentId, overrideFilters, entityViewName, birtContentType.

3. Import these data in the base using Webtools XML import (or the longer "gradlew 'ofbiz -l readers=seed,ext' command).
4. Add in the file content/config/contentEntityLabels.xml the Property which will translate your report Master description.

```

<property key="Content.description.CTNT_MASTER_TURNOVER">
    <value xml:lang="en">Turnover (product)</value>
    <value xml:lang="fr">Rotation (des stocks)</value>
</property>

```

1.5. Entities diagram

The following diagram shows the Entities linked with Content to store report_master/report.

[Report Master] | *Report-Master.png*