Dynamic Authorization Concept and Role Assignment in BI

Applies to:
This applies to SAP BI 3.X or SAP BI 7.X. For more details, visit the Business Intelligence homepage.

Summary
The document describes the procedure and set up for Dynamic Authorization and Role Assignment in a step-by-step manner.

Authors: Merlin Alwyn and Sandhya Mohan
Company: Infosys Technologies Ltd.
Created on: 10 March 2010

Author Bio
Merlin Alwyn is working as a Senior Systems Engineer with Infosys Technologies Ltd. Has a good experience in the SAP BI technology and have been involved in the implementation and execution of BW/BI Projects.

Sandhya Mohan is currently working as a Senior Software Engineer with Infosys Technologies Ltd. She is working in SAP BI 7.0 and is involved in implementation and execution of BI Project’s.
Table of Contents
What is Dynamic Authorization?..................................................................................................................3
Advantages:..............................................................................................................................................3
Scenario.......................................................................................................................................................3
  Step-by-step procedure to be followed .....................................................................................................3
  Steps for Role Creation and Assignment:................................................................................................6
CMOD Code Snippet:.................................................................................................................................11
Related Content..........................................................................................................................................13
Disclaimer and Liability Notice..................................................................................................................14
What is Dynamic Authorization?

Dynamic authorization concept is used to maintain Single roles and profiles for different end users. The key parameters for dynamic authorization can be any of the following: DSO, Master data table or a customized table.

Further this is used in the reports by using a customer exit variable which works based on the authorization details loaded in any of the above mentioned objects.

Advantages:

Security is well maintained by way of dynamic authorization.

Reduce the effort of the developers by having a single role for all users in an application.

Easy maintenance and future enhancements can be done.

Performance can be tuned by way of an optimized ABAP code.

Scenario

In an organization there may be different level of hierarchies maintained. For E.g.: Unit Head → Subunit Head → Customer.

In this hierarchy some users may have a single level of authorization or a multiple levels of authorization. Person A may have Org head authorization (i.e. full access to the entire data of the org.) and person B can have multiple authorization like subunit head for X and Customer access for A, B, C.

Step-by-step procedure to be followed

1. Mark the required info object as Authorization relevant.

In the above scenario we need to enable authorization for all the 3 levels namely Customer, Subunit head and Unit head. The authorization check should be enabled in the info objects’ → Business Explorer tab as shown below.

2. Create an authorization object for the required infoobject (E.g.: Customer) in transaction RSSM as shown below.
3. Select the object for which the authorization variable has been created. E.g. 0Customer.

```
Maintain Authorization Fields
```

```
Authorization object  Z_CUST  Test Customer
Created by  INF116622

Selected InfoObjects

<table>
<thead>
<tr>
<th>Fld</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0CUSTOMER</td>
<td>Customer number</td>
</tr>
</tbody>
</table>

Authorization relevant Objects

<table>
<thead>
<tr>
<th>Fld</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0BILLTOPRT</td>
<td>Bill-to-party</td>
</tr>
<tr>
<td>0CCOMPVIEW</td>
<td>Cost Component View</td>
</tr>
</tbody>
</table>
```

4. Enable authorization for the corresponding Multiprovider/Cube/DSO which is used for reporting by editing the authorization object with Check for info cubes option.

   In this case we enabled authorization for the cube Z_ABC at the customer level.

```
Switch On/Off InfoProvider Check
```

```
Authorization Object  Z_CUST  Test Customer

InfoCube Directory

<table>
<thead>
<tr>
<th>InfoProvider</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Z_ABC</td>
<td>z_abc</td>
</tr>
</tbody>
</table>
```

5. The next step is to create a DSO/Master data table/customized table to store the authorization relevant data. The mandatory fields for the DSO/table should be user name, level of authorization (E.g. Unit head, subunit head etc.) and the relevant info objects for which the authorization is maintained (E.g. customer, unit, and subunit).

6. The data can be loaded either through a flat file or an extractor which extracts data from the source-system.

7. The data load has to be done regularly in order to avoid any security mishaps that might happen due to some changes in the authorization levels.

8. The next step is to add the authorization object in the query. Create a characteristic variable for the customer info object.

   The features of the variable should be as shown below.

   - Type of variable: Characteristic value
   - Processing By: Customer Exit
NB: Generally in Analysis authorization only we enable processing type as Authorization.

9. Then add the created variable to the 0Customer Info object in the global filter of the query designer, so that the data fetched is restricted based on the individuals authorization.

10. The next step is to assign the authorization object (in this case Z_CUST) to a role that will be assigned to the end user.

11. For the variable Z_CUSTOMER a user exit in ABAP should be written in CMOD to fetch the data from the database table/DSO. A sample code snippet for the same is attached at the end of this document.

12. For the authorization concept, in the CMOD code the i_step value should be equal to ‘0’.

13. If the i_step = 1, then Call is made directly before variable entry.
Steps for Role Creation and Assignment:

1. The role can be created through the transaction PFCG.
2. In the role maintenance page give the name of the role that has to be created and click create single role or composite role button as per requirement.

<table>
<thead>
<tr>
<th>Role Maintenance</th>
</tr>
</thead>
<tbody>
<tr>
<td>Role</td>
</tr>
<tr>
<td>Name</td>
</tr>
</tbody>
</table>

3. On clicking the create button it navigates to the next page Change Roles.
4. In the Authorizations tab we need to either provide the profile name or the system can propose the profile name through the below shown icon.

<table>
<thead>
<tr>
<th>Change Roles</th>
</tr>
</thead>
<tbody>
<tr>
<td>Role</td>
</tr>
<tr>
<td>Description</td>
</tr>
</tbody>
</table>

5. The system generated profile name would be like as shown below. Once the profile name is generated save the role.

<table>
<thead>
<tr>
<th>Information About Authorization Profile</th>
</tr>
</thead>
<tbody>
<tr>
<td>Profile Name</td>
</tr>
<tr>
<td>Profile Text</td>
</tr>
<tr>
<td>Status</td>
</tr>
</tbody>
</table>

6. In the same tab we have Maintain Authorization Data and Generate Profiles. Click on the Change authorization Data icon.
7. We get a set of templates from which we can select the respective templates which will suite our requirement or we can ignore the templates.

8. If we don’t want to follow the template then we can click on the selection criteria button and choose the necessary authorizations from that page.

<table>
<thead>
<tr>
<th>Change role: Authorizations</th>
</tr>
</thead>
<tbody>
<tr>
<td>Z_CUST_ROLE</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Change role: Insert authorizations</th>
</tr>
</thead>
<tbody>
<tr>
<td>SAP Business Information Warehouse - Reporting</td>
</tr>
<tr>
<td>Test Customer</td>
</tr>
</tbody>
</table>

9. In this case we select the above 2 highlighted authorizations.
   a. Business Information Warehouse
   b. SAP Business Information Warehouse – Reporting

10. Expand the SAP Business Information Warehouse – Reporting tree and there we can find the authorization object (i.e. Z_CUST) that has been created. Click on the Not selected icon to then click on the insert chosen button.

11. Similarly for inserting authorization for the infoprovider choose Data Warehousing Workbench - Infocube under Business Information Warehouse tree and click on the Insert Chosen button.
12. Once the insertion of authorization is done the page looks like this.

13. Select the infocube on which the report has been built so that the authorization is enabled on the infoprovider. Also select the infoarea and activities for which all the authorization has to be enabled.

14. Following are the list of activities that can be assigned to an infocube. In general we choose activity 03 which is used for display.
15. The relevant objects are given for infocube and it appears like below. We have given full access for infocube sub object.

![Dynamic Authorization and Role Assignment in BI](image)

16. Next the authorization variable (Z_CUSTOMER) has to be inserted for SAP Business Information Warehouse – Reporting as shown below.

![Authorization Variable Insertion](image)

17. Once all the required objects have been added then click on the generate button and save the role.

18. Next in the User tab give the list of users to whom this role should be assigned and save.

![User Tab](image)

19. Click on the User comparison tab to compare the record with master data. We get the following window, select complete comparison to finish the validation process.

![User Comparison](image)
20. Now the role is assigned to the corresponding end users.

21. The above procedure explained is for one level of authorization i.e. customer. Similarly it has to be done for Unit head and Subunit head.
CMOD Code Snippet:

IF i_step = 0 .
CASE i_vnam .

*Authorization Variable for Unit, Subunit, Customer.
WHEN 'Z_SUBUNIT' OR 'Z_CUSTOMER' OR 'Z_UNIT'.

IF sy-subrc = 0 .

1_s_range-low = '**'.
1_s_range-sign = 'I'.
1_s_range-opt = 'CP'.

APPEND 1_s_range TO e_t_range .

ENDIF .
ENDCASE .
ENDIF .

*Before execution of i/p variable and also Auth check for a customer .
If i_step = 1 .
WHEN 'Z_CUSTOMER'.

CLEAR gi_itab_temp .
CLEAR gi_itab_auth_all .
LOOP AT gi_itab_auth INTO wa_itab_auth.
SELECT
/bic/unit
/bic/sunit
/bic/customer
FROM /bic/afiar_o1000
INTO TABLE gi_itab_temp
WHERE /bic/ic_user = sy-uname
   AND /bic/customer = '**'.
SORT gi_itab_temp BY unit .
IF wa_itab_auth-cus EQ '**'.
SELECT /bic/unit
   /bic/sunit
   /bic/customer FROM /bic/mcustomer
      INTO TABLE gi_itab_auth_all
WHERE
   /bic/unit = wa_itab_auth-unit .
SORT gi_itab_auth_all BY unit .

***UNIT HEAD Access ******
IF wa_itab_auth-cus EQ '*' AND wa_itab_auth-sunit EQ '**'.
LOOP AT gi_itab_temp INTO wa_itab_temp .
READ TABLE gi_itab_auth_all INTO wa_itab_auth_all
   WITH KEY unit = wa_itab_auth-unit .
LOOP AT gi_itab_auth_all INTO wa_itab_auth_all .
1_s_range-low = wa_itab_auth_all-cus .
1_s_range-sign = 'I'.
1_s_range-opt = 'EQ'.
APPEND 1_s_range TO e_t_range .
ENDLOOP .
ENDLOOP .

****Subunit Head access
ELSEIF wa_itab_auth-cus EQ '*' AND wa_itab_auth-sunit NE '*' .
  CLEAR wa_itab_temp .
  CLEAR wa_itab_auth_all .
  READ TABLE gi_itab_temp INTO wa_itab_temp
    WITH KEY unit = wa_itab_auth-unit
        sunit = wa_itab_auth-sunit .
  LOOP AT gi_itab_temp INTO wa_itab_temp WHERE sunit = wa_itab_auth-sunit .
    READ TABLE gi_itab_auth_all INTO wa_itab_auth_all WITH KEY
      sunit = wa_itab_auth-sunit
        unit = wa_itab_auth-unit .
    LOOP AT gi_itab_auth_all INTO wa_itab_auth_all
      WHERE unit = wa_itab_auth-unit
          AND sunit = wa_itab_auth-sunit .
      l_s_range-low    = wa_itab_auth_all-cus .
      l_s_range-sign   = 'I'.
      l_s_range-opt    = 'EQ'.
      APPEND l_s_range TO e_t_range .
    ENDLOOP .
  ENDLOOP .
ENDIF .
ENDLOOP .
Related Content

www.sdn.sap.com
help.sap.com

For more information, visit the Business Intelligence homepage.
Disclaimer and Liability Notice

This document may discuss sample coding or other information that does not include SAP official interfaces and therefore is not supported by SAP. Changes made based on this information are not supported and can be overwritten during an upgrade.

SAP will not be held liable for any damages caused by using or misusing the information, code or methods suggested in this document, and anyone using these methods does so at his/her own risk.

SAP offers no guarantees and assumes no responsibility or liability of any type with respect to the content of this technical article or code sample, including any liability resulting from incompatibility between the content within this document and the materials and services offered by SAP. You agree that you will not hold, or seek to hold, SAP responsible or liable with respect to the content of this document.