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?

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

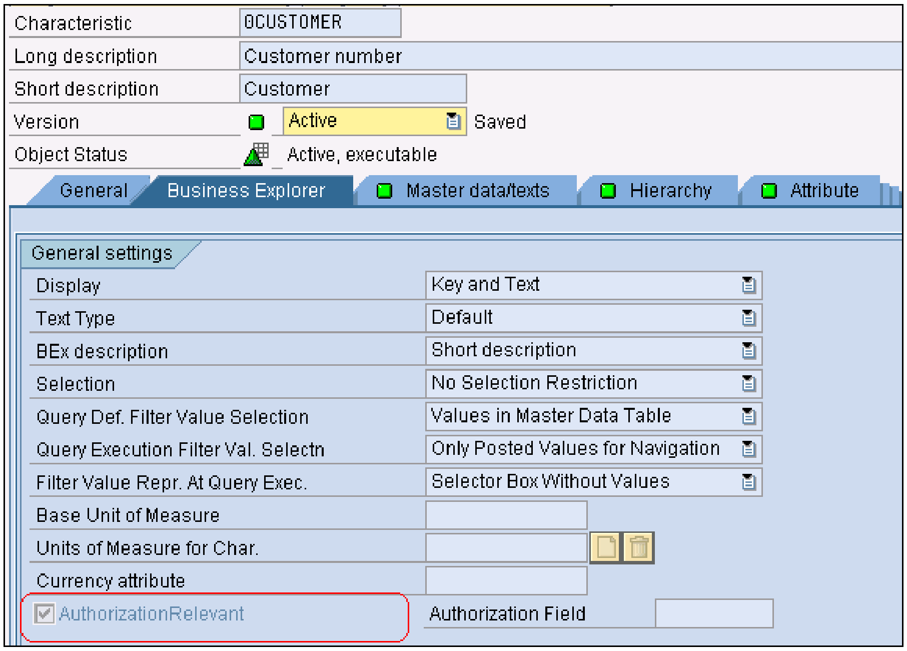
In an organization there may be different level of hierarchies maintained . For E.g.: Unit Head → Subunit Head → Customer.

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.

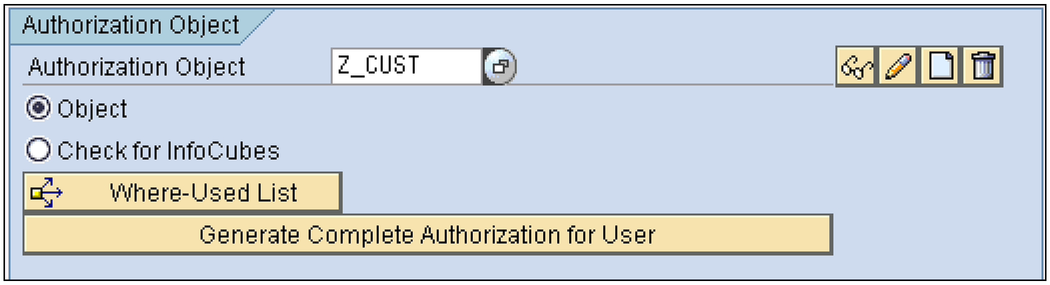


Characteristic	0CUSTOMER
Long description	Customer number
Short description	Customer
Version	🟩 Active Saved
Object Status	Active, executable

General settings

Display	Key and Text
Text Type	Default
BEx description	Short description
Selection	No Selection Restriction
Query Def. Filter Value Selection	Values in Master Data Table
Query Execution Filter Val. Selectn	Only Posted Values for Navigation
Filter Value Repr. At Query Exec.	Selector Box Without Values
Base Unit of Measure	
Units of Measure for Char.	
Currency attribute	
☑ AuthorizationRelevant	Authorization Field

2. Create an authorization object for the required infoobject (E.g.: Customer) in transaction RSSM as shown below.

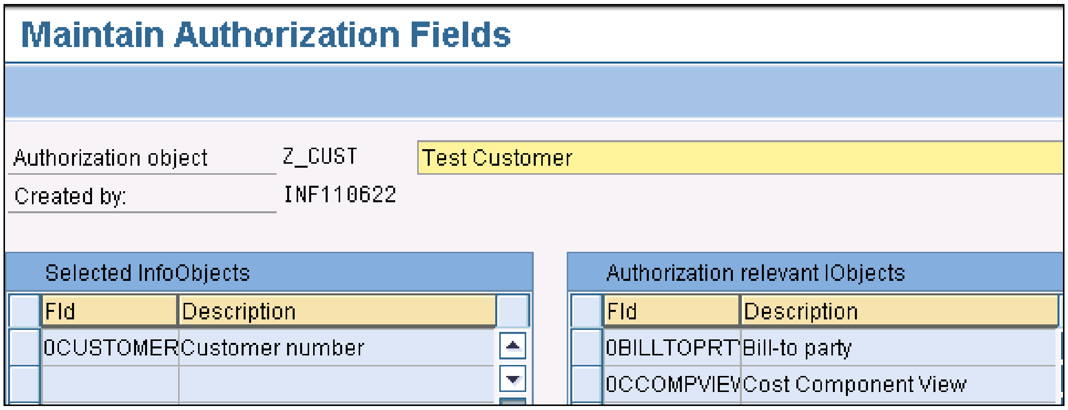


Authorization Object: Z_CUST

● Object

○ Check for InfoCubes

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: INF110622

Selected InfoObjects

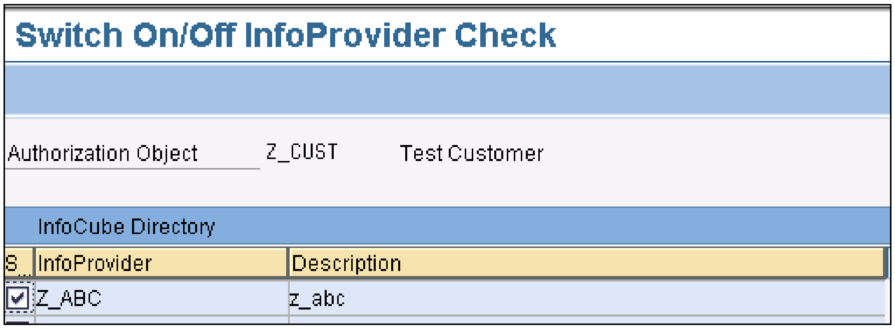
Fld	Description
0CUSTOMER	Customer number

Authorization relevant IObjects

Fld	Description
0BILLTOPRT	Bill-to party
0CCOMPVIEV	Cost Component View

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

S	InfoProvider	Description
☑	Z_ABC	z_abc

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.

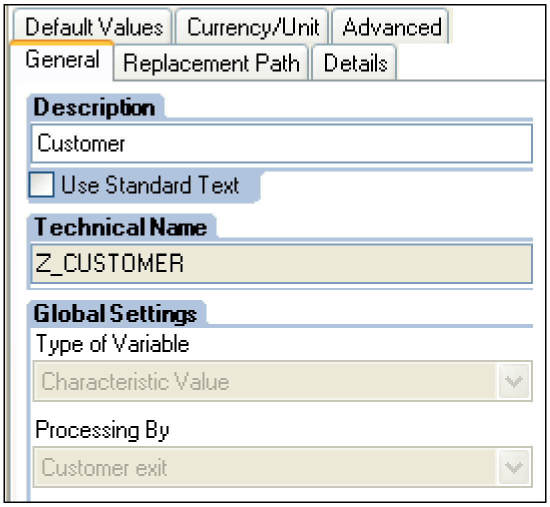
7. The data load has to be done regularly in order to avoid any security mishaps that might happen due to some changes in the authorization levels.

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



Description

Customer

□ Use Standard Text

Technical Name

Z_CUSTOMER

Global Settings

Type of Variable

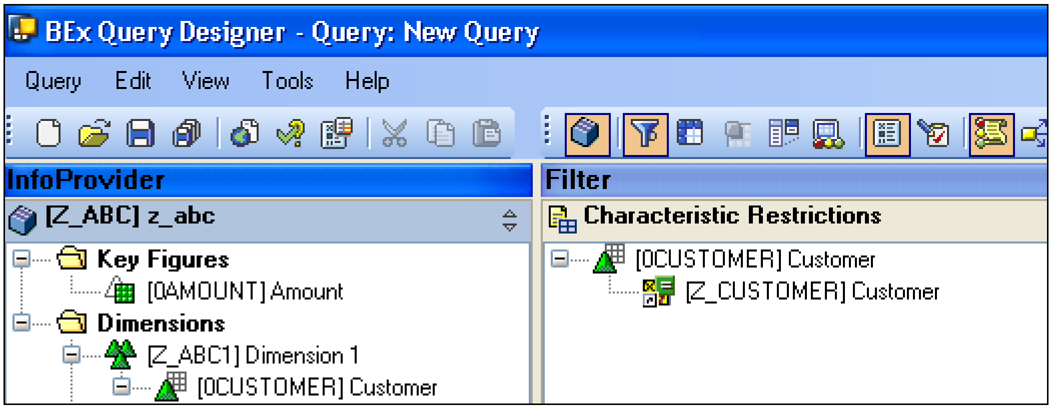
Characteristic Value

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.



[Z_ABC] z_abc

Key Figures

[0AMOUNT] Amount

Dimensions

[Z_ABC1] Dimension 1

[0CUSTOMER] Customer

Characteristic Restrictions

[0CUSTOMER] Customer

[Z_CUSTOMER] Customer

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.

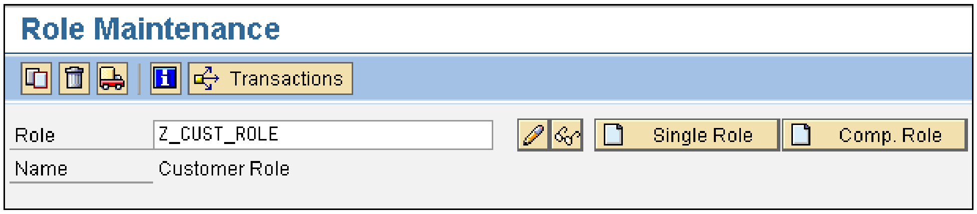
11. For the variable Z_CUSTOMER a user exit in ABAP should be written in CMOD to fetch the data from the database table/DSO. A sample code snippet for the same is attached at the end of this document.

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.

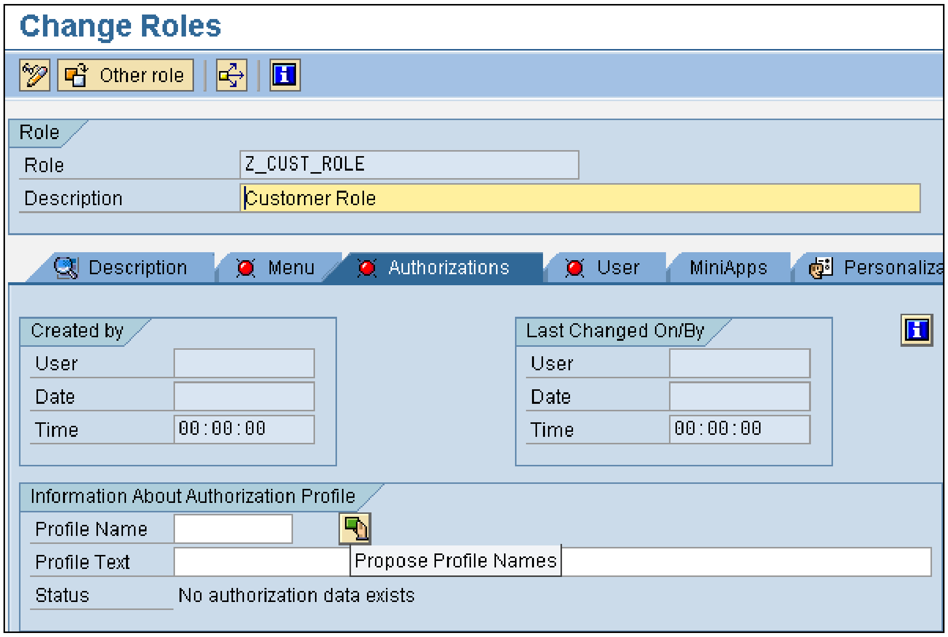


Role Maintenance

Role: Z_CUST_ROLE

Name: Customer Role

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.



Change Roles

Role

Role: Z_CUST_ROLE

Description: Customer Role

Created by

User:

Date:

Time: 00:00:00

Last Changed On/By

User:

Date:

Time: 00:00:00

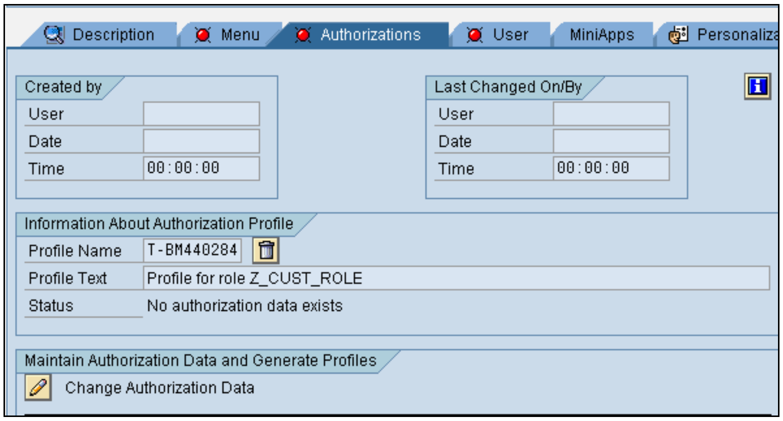
Information About Authorization Profile

Profile Name:

Profile Text:

Status: No authorization data exists

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



Created by

User:

Date:

Time: 00:00:00

Last Changed On/By

User:

Date:

Time: 00:00:00

Information About Authorization Profile

Profile Name: T-BM440284

Profile Text: Profile for role Z_CUST_ROLE

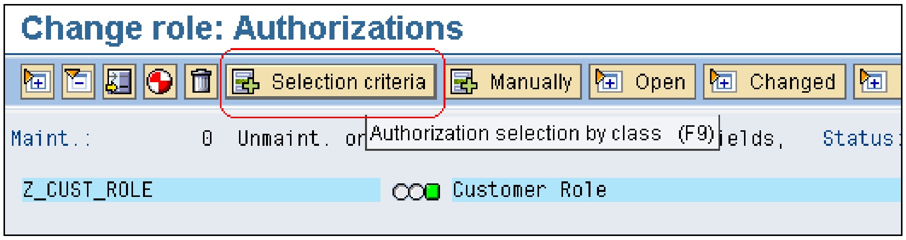
Status: No authorization data exists

Maintain Authorization Data and Generate Profiles

6. In the same tab we have Maintain Authorization Data and Generate Profiles. Click on the Change authorization Data icon.

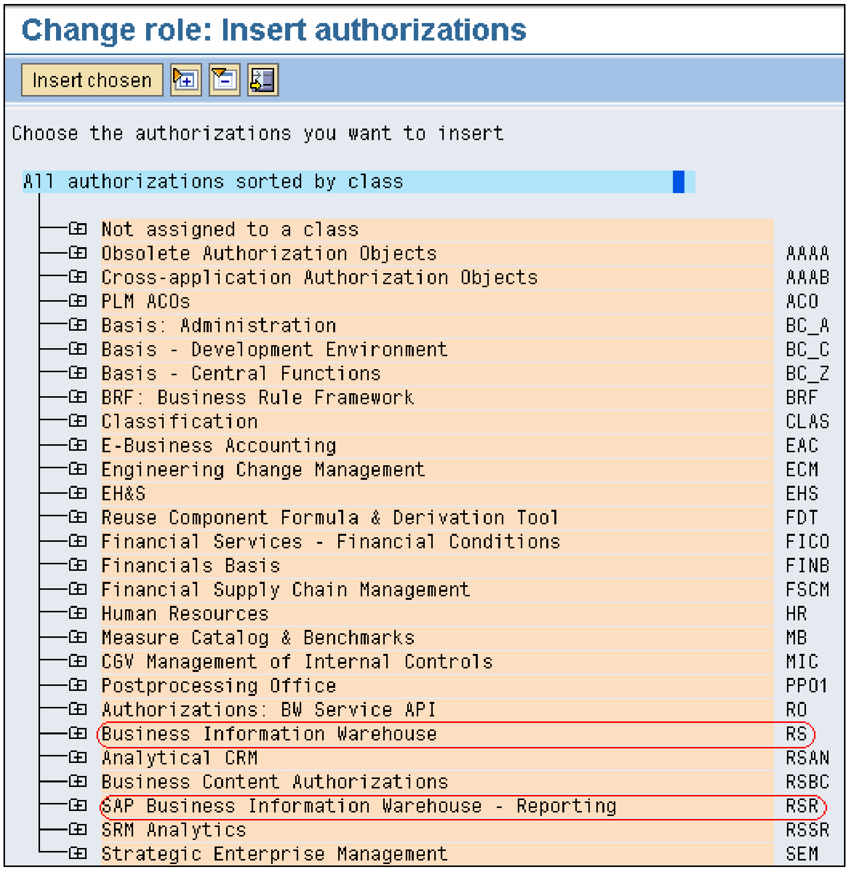
7. We get a set of templates from which we can select the respective templates which will suite our requirement or we can ignore the templates.

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.



Change role: Authorizations

```
Maint.:          0  Unmaint. or                          elds,   Status:
Z_CUST_ROLE                     Customer Role
```



Change role: Insert authorizations

Choose the authorizations you want to insert

All authorizations sorted by class

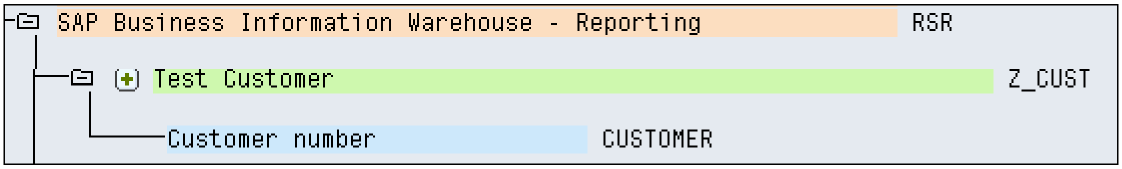
Not assigned to a class	
Obsolete Authorization Objects	AAAA
Cross-application Authorization Objects	AAAB
PLM ACOs	ACO
Basis: Administration	BC_A
Basis - Development Environment	BC_C
Basis - Central Functions	BC_Z
BRF: Business Rule Framework	BRF
Classification	CLAS
E-Business Accounting	EAC
Engineering Change Management	ECM
EH&S	EHS
Reuse Component Formula & Derivation Tool	FDT
Financial Services - Financial Conditions	FICO
Financials Basis	FINB
Financial Supply Chain Management	FSCM
Human Resources	HR
Measure Catalog & Benchmarks	MB
CGV Management of Internal Controls	MIC
Postprocessing Office	PP01
Authorizations: BW Service API	RO
Business Information Warehouse	RS
Analytical CRM	RSAN
Business Content Authorizations	RSBC
SAP Business Information Warehouse - Reporting	RSR
SRM Analytics	RSSR
Strategic Enterprise Management	SEM

9. In this case we select the above 2 highlighted authorizations.

   a. Business Information Warehouse

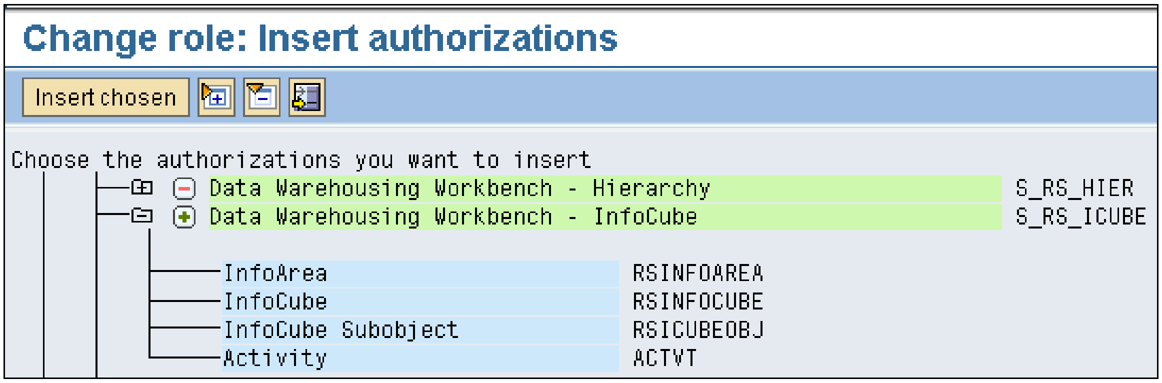
   b. SAP Business Information Warehouse – Reporting

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.



```
SAP Business Information Warehouse - Reporting      RSR
    Test Customer                                      Z_CUST
        Customer number              CUSTOMER
```

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.

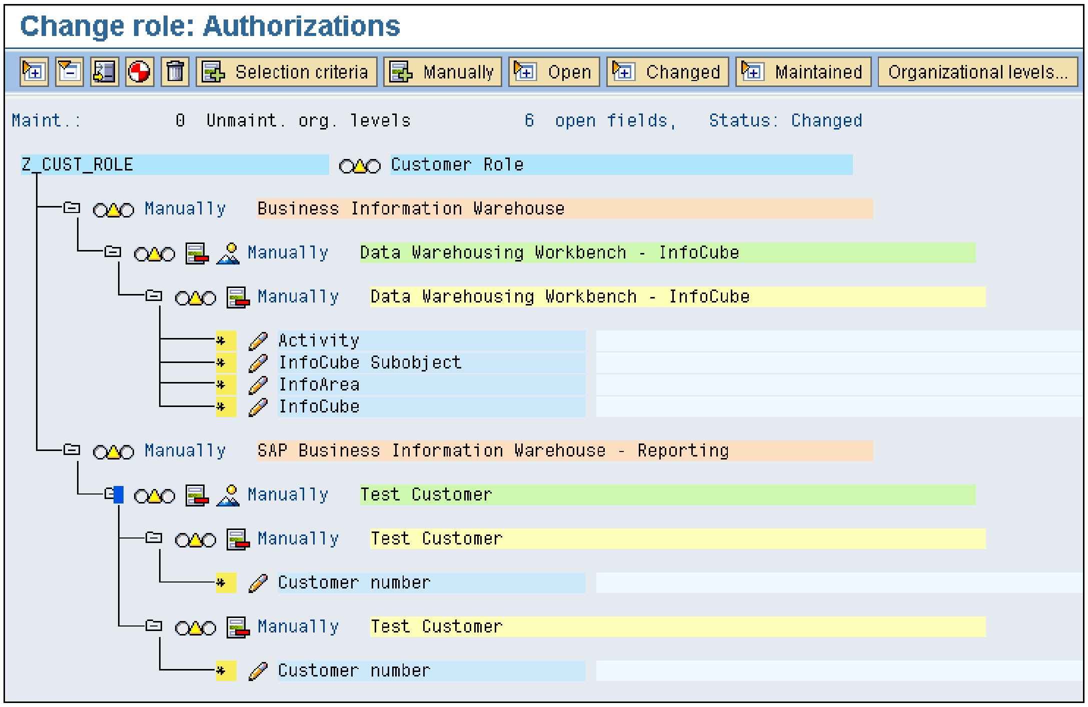


Change role: Insert authorizations

```
Choose the authorizations you want to insert
   Data Warehousing Workbench - Hierarchy          S_RS_HIER
   Data Warehousing Workbench - InfoCube           S_RS_ICUBE

      InfoArea                  RSINFOAREA
      InfoCube                  RSINFOCUBE
      InfoCube Subobject        RSICUBEOBJ
      Activity                  ACTVT
```

12. Once the insertion of authorization is done the page looks like this.



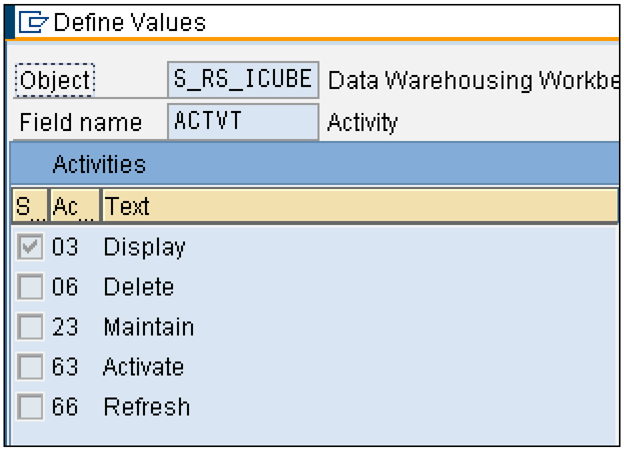
Change role: Authorizations

```
Maint.:        0  Unmaint. org. levels        6  open fields,   Status: Changed

Z_CUST_ROLE                  Customer Role
   Manually   Business Information Warehouse
      Manually   Data Warehousing Workbench - InfoCube
         Manually   Data Warehousing Workbench - InfoCube
            Activity
            InfoCube Subobject
            InfoArea
            InfoCube
   Manually   SAP Business Information Warehouse - Reporting
      Manually   Test Customer
         Manually   Test Customer
            Customer number
         Manually   Test Customer
            Customer number
```

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.



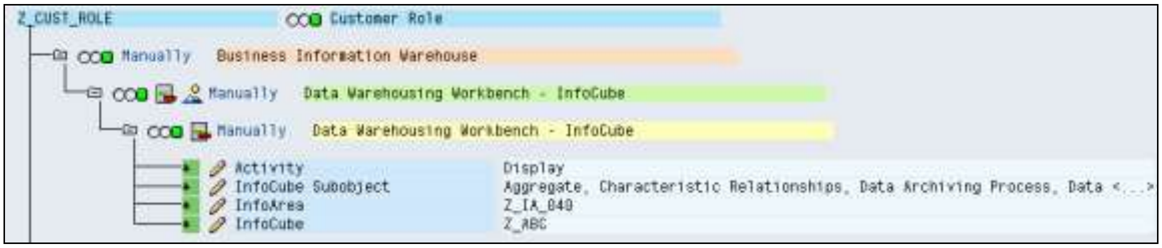
Define Values

Object	S_RS_ICUBE	Data Warehousing Workbe
Field name	ACTVT	Activity

Activities

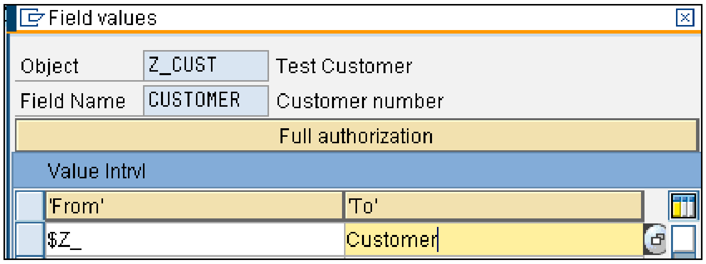
S...	Ac...	Text
☑	03	Display
□	06	Delete
□	23	Maintain
□	63	Activate
□	66	Refresh

15. The relevant objects are given for infocube and it appears like below. We have given full access for infocube sub object.



```
Z_CUST_ROLE                      Customer Role
  Manually   Business Information Warehouse
    Manually   Data Warehousing Workbench - InfoCube
      Manually   Data Warehousing Workbench - InfoCube
        Activity             Display
        InfoCube Subobject   Aggregate, Characteristic Relationships, Data Archiving Process, Data <...>
        InfoArea             Z_IA_040
        InfoCube             Z_ABC
```

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



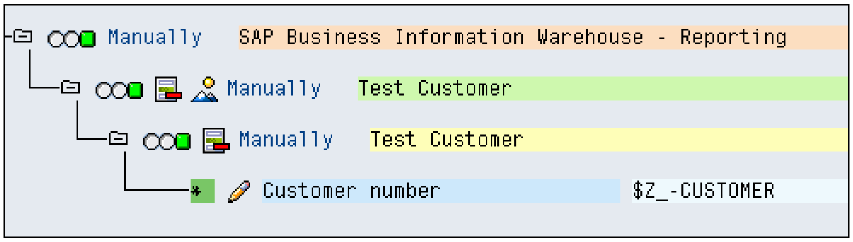
Field values

Object	Z_CUST	Test Customer
Field Name	CUSTOMER	Customer number

Full authorization

Value Intrvl

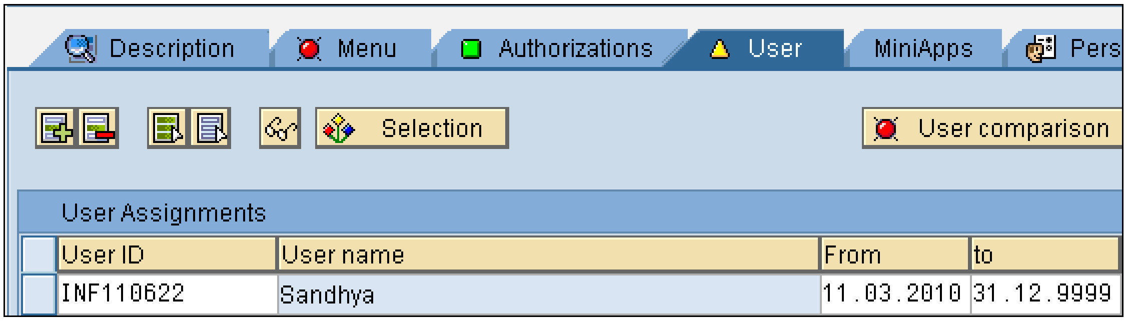
'From'	'To'
$Z_	Customer



```
Manually   SAP Business Information Warehouse - Reporting
  Manually   Test Customer
    Manually   Test Customer
      Customer number        $Z_-CUSTOMER
```

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

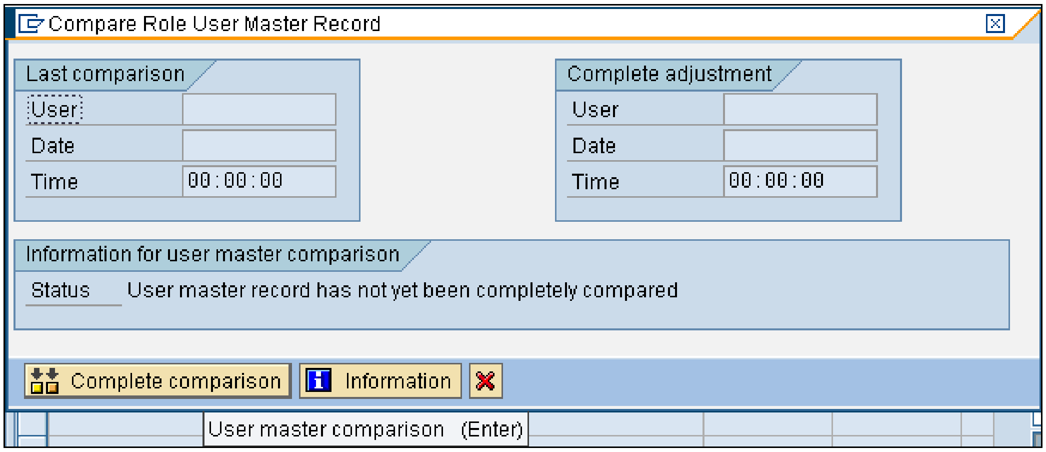
18. Next in the User tab give the list of users to whom this role should be assigned and save.



User Assignments

User ID	User name	From	to
INF110622	Sandhya	11.03.2010	31.12.9999

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.



Compare Role User Master Record

Last comparison

User	
Date	
Time	00:00:00

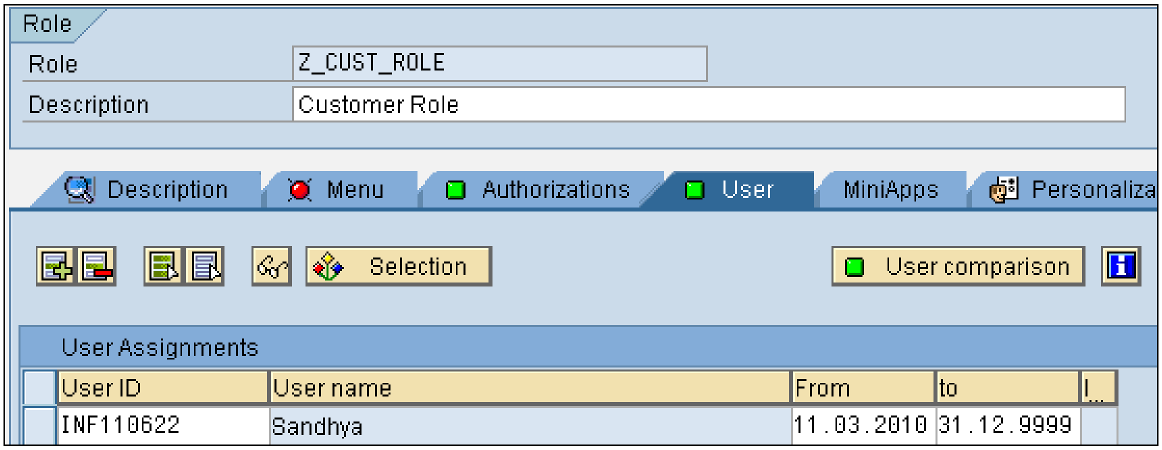
Complete adjustment

User	
Date	
Time	00:00:00

Information for user master comparison

Status User master record has not yet been completely compared

20. Now the role is assigned to the corresponding end users.



Role

Role	Z_CUST_ROLE
Description	Customer Role

User Assignments

User ID	User name	From	to	I...
INF110622	Sandhya	11.03.2010	31.12.9999	

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 .

        l_s_range-low    = '*'.
        l_s_range-sign   = 'I'.
        l_s_range-opt    = 'CP'.

        APPEND l_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 .
                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 .
****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 .
        ELSEIF wa_itab_auth-cus NE  '*'.
          l_s_range-low    = wa_itab_auth-cus.
          l_s_range-sign   = 'I'.
          l_s_range-opt    = 'EQ'.
          APPEND l_s_range TO e_t_range.
        ENDIF.
      ENDLOOP.
ENDIF.
```

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.