



SAP DEVELOPERS NETWORK

EXCHANGE USEFUL CODE, GET HELP FROM COLLEAGUES

Categories

HOME | SAP HANA | SAP Hybris | SAP ABAP | SAP Security|SuccessFactors|S/4 HANA|SAP JOBS|SAP FREELANCE

SAP JOBS

[Full time]
консультант FI at
TeamIdea

2 days ago

[Full time]
Consultant
MM\SD\PP at
Adecco Russia

3rd Apr 2020

[Full time] ABAP
разраотчик at
Adecco Russia

3rd Apr 2020

Wednesday, 08 April 2020 10:49

Implementing DCL with CDS Views and Roles

Written by <https://blogs.sap.com/2020/04/09/implementing-dcl-with-cds-views-and-roles/>

Font size | Print | Email | Be the first to comment!

Rate this item

(0 votes)

Hi Readers,

I am going to write here about how to use the concept of DCL with CDS views and using roles to provide row based authorizations in UI.

So,

DCL stands for Data Control Language. It provides an access control mechanism to restrict the results returned by the CDS view from the database according to conditions. With the help of roles (PFCG), we can also use the same DCL and restrict the results for different categories of users.

Let us look into the process, with the help of an example.

Suppose I have the below database table:

SAP Dictionary: Display Table

Transparent Table: **ZARJ_DCL_TEST** Active

Short Description: DCL Test For UI

Attributes Delivery and Maintenance Fields Input Help/Check Currency/Quantity Fields Indexes

Field	Key	Initia...	Data element	Data Type	Length	Decima...	Short Description
<input type="checkbox"/> MANDT	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	MANDT	CLNT	3		0 Client
<input type="checkbox"/> PID	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	CHAR_80	CHAR	80		0 Field of length 80
<input type="checkbox"/> KEY1	<input type="checkbox"/>	<input type="checkbox"/>	ZAR_CN_CO	CHAR	40		0 DCL Country Code Test
<input type="checkbox"/> VALUE	<input type="checkbox"/>	<input type="checkbox"/>	CHAR_80	CHAR	80		0 Field of length 80
<input type="checkbox"/> NAME	<input type="checkbox"/>	<input type="checkbox"/>	CHAR_80	CHAR	80		0 Field of length 80

ZARJ_DCL_TEST – The table with the fields MANDT, PID, KEY1, VALUE and NAME.

Now, lets check the data entered for the above table:

Table: ZARJ_DCL_TEST

Displayed Fields: 5 of 5 Fixed Columns: 2 List Width: 0600

	MANDT	PID	KEY1	VALUE
<input type="checkbox"/> 300		01E29DB7-7FCE-484D-98CA-CE63C8A2773C	IND	BAN
<input type="checkbox"/> 300		0D77A695-E779-4F55-A8F5-70CC69C72300	IND	TRI
<input type="checkbox"/> 300		0EDBFDC-7907-4572-8E10-57055F5C6320	IND	KOL
<input type="checkbox"/> 300		207FD7F7-02BB-44CC-99C2-144819AC946D	RSA	LST
<input type="checkbox"/> 300		4C55E041-B8D2-46B3-AD34-9A5634FF24E2	RSA	CCT
<input type="checkbox"/> 300		6119CA4C-E74E-4B80-A42C-D2B6A04DB507	IND	MAS
<input type="checkbox"/> 300		63BDFEF4-3800-48D3-A52E-FD030BA19614	RSA	PTR
<input type="checkbox"/> 300		7A3ABE50-AA44-45F6-A76B-68A53C3DEC3D	RSA	JHB
<input type="checkbox"/> 300		842CF695-FF86-4842-858C-123FF3E07F81	USA	NYC
<input type="checkbox"/> 300		AC0DFBFC-CD24-45EC-8AB3-AB638D726A01	USA	CAL
<input type="checkbox"/> 300		B40748E3-8C79-4490-AA13-2D724CFC72E9	USA	PSV
<input type="checkbox"/> 300		E1634C3F-A098-4D9E-9959-C6B49E7D131E	IND	MUM
<input type="checkbox"/> 300		E816948A-CDDF-49A4-9800-F4F79A2B457D	IND	BHO
<input type="checkbox"/> 300		F0ECA04A-DE7C-4D14-998F-F56C5BB7B4F1	IND	DEL

Here in the KEY1 field we can see that we have three country codes, RSA, IND and USA. Now, our aim is to display records with only one or two of these country codes to the user from UI.

So, in the next step, we can create an Implementation CDS View (ZDCL_I_CO) from this table.

@AbapCatalog.sqlViewName: 'ZAR_I_DCL' @AbapCatalog.compiler.compareFilter: true @AbapCatalog.preserveKey: t

Now, consume this implementation view in our next consumption view (ZDCL_C_CO):

```
1 @AbapCatalog.sqlViewName: 'ZAR_C_DCL'
2 @AbapCatalog.compiler.compareFilter: true
3 @AbapCatalog.preserveKey: true
4 @AccessControl.authorizationCheck: #CHECK
5 @EndUserText.label: 'Consumption DCL'
6 @VDM.viewType: #CONSUMPTION
7 @OData.publish: true
8 @Search.searchable: true
9 define view ZDCL_C_CO as select from ZDCL_I_CO {
10     key CityUUID as CityUUID,
11     @Search.defaultSearchElement: true
12     @UI.identification: [{position: 10}]
13     @UI.selectionField: {position: 10}
14     @UI.lineItem: { position: 10}
15     @EndUserText.label: 'Country Code'
16     KeyCounCode as KeyCounCode,
17     @Consumption.filter: {selectionType: #SINGLE, multipleSelections: false}
18     @UI.selectionField: {position: 20}
19     @UI.identification: [{position: 20}]
20     @UI.lineItem: { position: 20}
21     @EndUserText.label: 'City Code'
22     CityCode as CityCode,
23     @UI.lineItem: { position: 30}
24     @EndUserText.label: 'City Name'
25     CityName as CityName
26 }
27
```

Here is the code:

@AbapCatalog.sqlViewName: 'ZAR_C_DCL' @AbapCatalog.compiler.compareFilter: true @AbapCatalog.preserveKey: t

Now, lets check the output by viewing it in the data preview in eclipse:



► ZDCL_C_CO ►

Raw Data Show Log Max. Rows: 100

Filter pattern ☒ 11 rows retrieved - 18 ms SQL Console Data Aging 11 Number of Entries Select Columns Add filter

CityUUID	KeyCounCode	CityCode	CityName
01E29DB7-7FCE-484D-98CA-CE63C8A2773C	IND	BAN	BANGALORE
0D77A695-E779-4F55-A8F5-70CC69C72300	IND	TRI	TRIVANDRUM
0EDBFDCC-7907-4572-8E10-57055F5C6320	IND	KOL	KOLKATA
207FD7F7-02BB-44CC-99C2-144819AC946D	RSA	LST	LESHOTO
4C55E041-B8D2-46B3-AD34-9A5634FF24E2	RSA	CCT	CAPE TOWN
6119CA4C-E74E-4B80-A42C-D2B6A04DB507	IND	MAS	CHENNAI
63BDFEF4-3800-48D3-A52E-FD030BA19614	RSA	PTR	PRETORIA
7A3ABE50-AA44-45F6-A76B-68A53C3DEC3D	RSA	JHB	JOHANNESBURG
E1634C3F-A098-4D9E-9959-C6B49E7D131E	IND	MUM	MUMBAI
E816948A-CDDF-49A4-9800-F4F79A2B457D	IND	BHO	BHOPAL
FOECA04A-DE7C-4D14-998F-F56C5BB7B4F1	IND	DEL	DELHI

Now from here we need to follow the below steps to create a Access Control (DCL)

- Create a Authorization Field and Authorization Object
- Create a Access Control in Eclipse for the desired CDS view
- Create a PFCG role and assign it to the user with authorizations
- Create a Fiori App to preview the output (Optional as the output can be viewed in the CDS level itself)

Creating an OData Service and a Fiori Worklist App

As I have mentioned before this step is optional. We can activate the OData Service generated from the above consumption view (due to the @OData.publish: true annotation in line 7). To do this, go to the TCode /n/iwfnd/maint_service and click on the add service button. The below screen will come up.



Get Services More ▾ Exit

Filter

System Alias: ☐ Co-Deployed

Technical Service Name:

Version:

External Service Name: External Mapping ID:

Select Backend Services

☐ Add Selected Services

Type	Technical Service Name	Version	Service Description	External Service Name	Namespace

Here enter the system alias as "LOCAL" and enter the CDS view name with both preceding and succeeding stars in the Technical Service Name and click on get services. Our CDS generated service should come in the Select Backend Services table below.

Select the checkbox beside the service in the table and click on the add selected services button to add the service. It would prompt a dialog for selecting the transport request for the MPC and DPC classes. Select an transport request or click local object and save it.

Now go to the previous screen and select the filter and enter the service name added on the previous screen and click enter. In this case, it should display the service as below:

Service Catalog

☐ Add Service ☐ Delete Service ☐ Service Details ☐ Load Metadata ☐ Error Log ☐ Request Statistics

☐ Refresh Catalog ☐ OAuth ☐ Soft State ☐ Processing Mode ☐ Add to Transport

Type	Technical Service Name	Version	Service Description	External Service Name	Namespace	OAuth	Soft State	Status	Pr
BEP	<input checked="" type="checkbox"/> ZDCL_C_CO_CDS	1	Consumption DCL	ZDCL_C_CO_CDS		<input type="checkbox"/>	Not Supported	Ro	

ICF Nodes

☐ ICF Node ☐ Call Browser ☐ SAP Gateway Client

Status	ICF Node	Session Time-out	Soft State	Description
○○	ODATA	00:00:00		Standard Mode

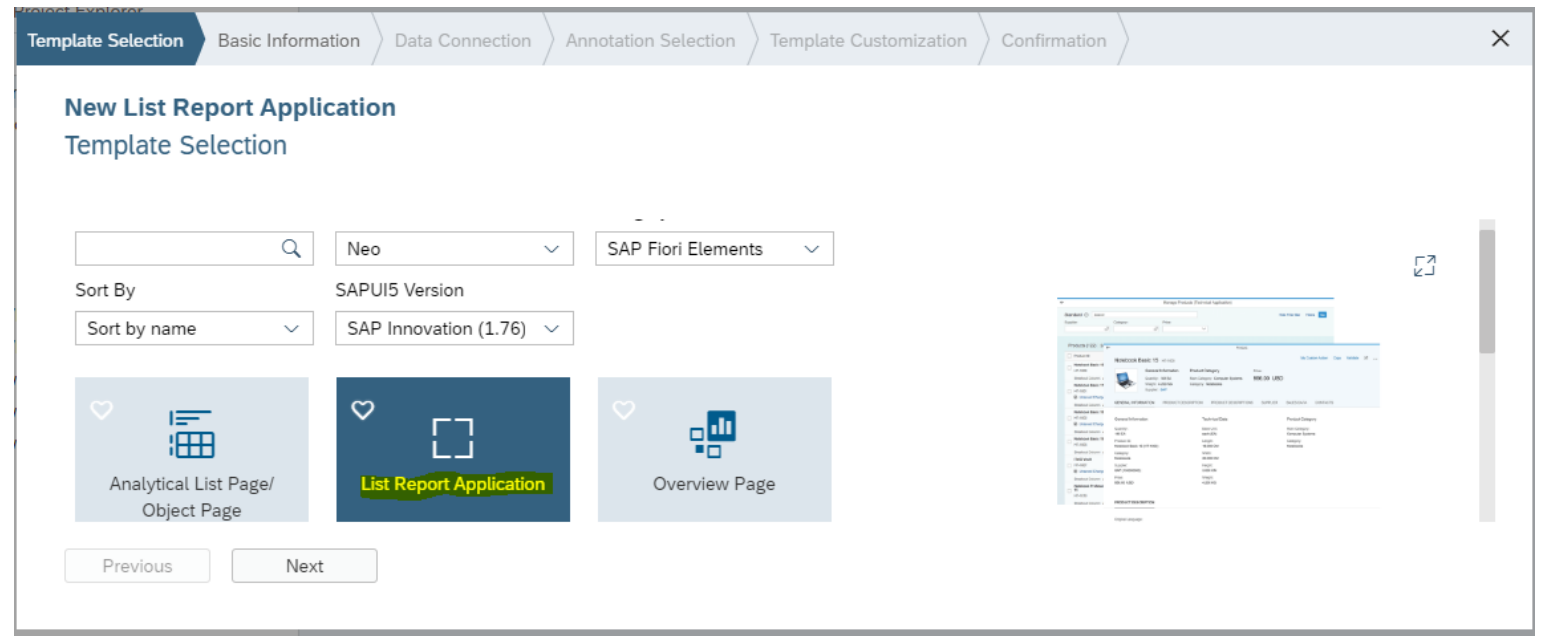
System Aliases

☐ Add System Alias ☐ Remove System Alias ☐ Customizing ☐ Service Implementation

SAP System Alias	Description	Default System	Me
LOCAL	Local System Alias	<input checked="" type="checkbox"/>	

Now in our WebIDE, we can create a List report application with this OData service, to look at the data that is being fetched:

Select the List Report Application in the list of templates-



Select the proper OData service & in the next screen select all the annotations-



Template Selection
Basic Information
Data Connection
Annotation Selection
Template Customization
Confirmation

New List Report Application

Data Connection

Choose a service from one of the sources listed below.

Workspace
File System
Service URL
SAP API Business ...

Services

Service	Description	System
> <input type="radio"/> ZDCL_C_CO_CDS	Consumption DCL	BMC_HA4_I

Select the header collection entity set (This is responsible for the data that is being populated in the first view table)-

Template Selection
Basic Information
Data Connection
Annotation Selection
Template Customization
Confirmation

New List Report Application

Template Customization

Data Binding

OData Collection*

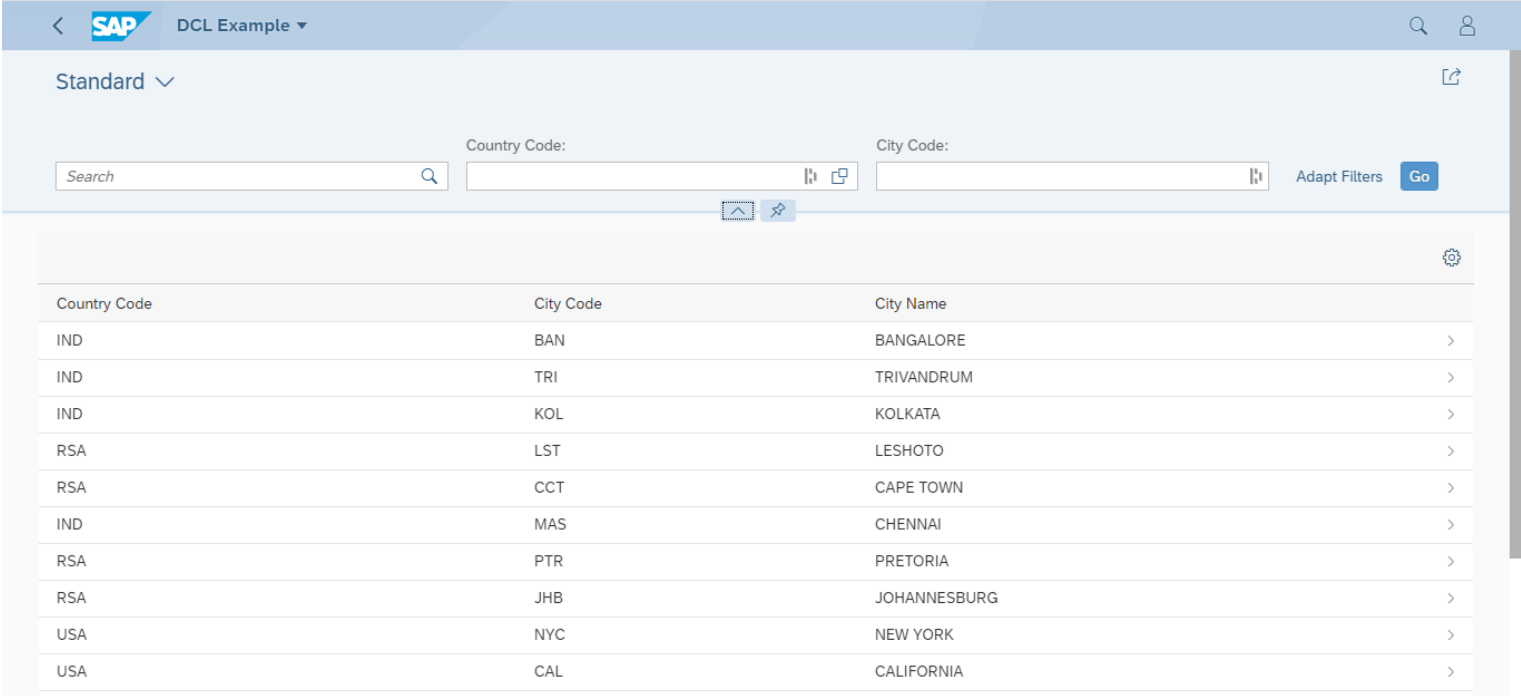
OData Navigation

OData Sub Navigation

☒ Smart Variant Management for List Report

☐ Flexible Column Layout

Now, click on finish and run the application with proper run configurations (preferably run the application from the component instead of the index.html). The below screen should appear with both the smart filters and the smart table.



The screenshot shows a SAP Fiori application interface. At the top, there is a header bar with the SAP logo, a dropdown menu labeled 'DCL Example', and search and user icons. Below the header, there is a section labeled 'Standard' with a dropdown arrow. Underneath, there are two input fields: 'Country Code' and 'City Code', each with a search icon and a 'Go' button. To the right of these fields is an 'Adapt Filters' button. Below the input fields is a table with three columns: 'Country Code', 'City Code', and 'City Name'. The table contains 10 rows of data. Each row has a right arrow icon in the 'City Name' column.

Country Code	City Code	City Name
IND	BAN	BANGALORE
IND	TRI	TRIVANDRUM
IND	KOL	KOLKATA
RSA	LST	LESHOTO
RSA	CCT	CAPE TOWN
IND	MAS	CHENNAI
RSA	PTR	PRETORIA
RSA	JHB	JOHANNESBURG
USA	NYC	NEW YORK
USA	CAL	CALIFORNIA

We can see that all the country codes and all the data is visible in the UI.

Creating a Authorization Object and Authorization Field

To do this, we make use of the TCodes SU21 and SU20 respectively.

Authorization Field: The *authorization fields* contain data element fields that are in the form of single values or range value and this value sets which are used for authorizations.

To create an authorization field, we go to the TCode SU20. Click on the Authorization Field button on the top menu bar.



< **SAP** Initial Screen: Authorization Field Maintenance

Save as Variant... Sync. Field Buffer **Authorization Field** Display Application Help More ▾ Exit

Standard Selection

Authorization Field Name: to:

Data element:

Authorization Object:

Only Select OrgLevel Fields: ☐

Other Restrictions (Object Directory)

Package:

Software Component:

Application Component:

Display Options

☐ Display Field Selection List with Technical Properties

Execute

In the next screen, enter the desired authorization field name (ZAR_DLC_CC), and the name of the data element you wish to provide authorization for. In our case the data element is ZAR_CN_CO, for the field KEY1 in our se11 table.

< **SAP** Create Authorization Field

Display <-> Change Authorization field Information More ▾ Exit

Authorization Field Header Data

Authorization Fld:

ABAP Dictionary

Data Element:

Domain:

Output Length: 0

Convers. Routine:

Provide Search Help in Standard Maintenance Dialog

☒ Using Check Table (Authorization Fld)

☐ Using Search Help of Data Element

Screenshot of our se11 table field for which we wish to add authorization to:

Attributes Delivery and Maintenance Fields Input Help/Check Currency/Quantity Fields Indexes							
<div> </div> <div> </div> <div> <input type="text" value="Search"/> <input type="text" value="Built-In Type"/> 5 </div>							
Field	Key	Initia...	Data element	Data Type	Length	Decima...	Short Description
<input type="checkbox"/> MANDT	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	MANDT	CLNT	3	0	Client
<input type="checkbox"/> PID	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	CHAR_80	CHAR	80	0	Field of length 80
<input type="checkbox"/> KEY1	<input type="checkbox"/>	<input type="checkbox"/>	ZAR_CN_CO	CHAR	40	0	DCL Country Code Test
<input type="checkbox"/> VALUE	<input type="checkbox"/>	<input type="checkbox"/>	CHAR_80	CHAR	80	0	Field of length 80
<input type="checkbox"/> NAME	<input type="checkbox"/>	<input type="checkbox"/>	CHAR_80	CHAR	80	0	Field of length 80
<input type="checkbox"/>							
<input type="checkbox"/>							
<input type="checkbox"/>							

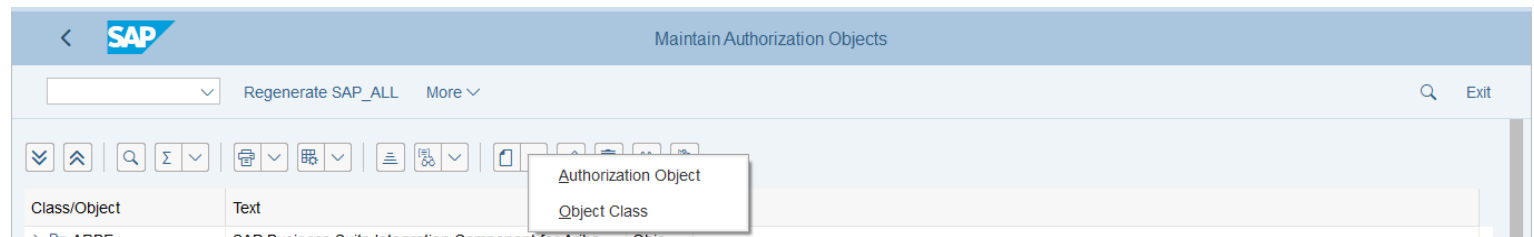
Authorization Class/Object: The authorization class is an group of one or more authorization obje
 Authorization object is a place where the configurations regarding authorizations and permissions are

set up and initialized against the authorization fields.

We do not need to create the authorization class every time we create the authorization object. It can be done only once.

To create the Authorization Class and object, we go to the tcode SU21. All our DCL based authorizations can be collectively placed in an authorization class. We can name it as ZDCL.

So, click on the new button (paper icon) and click on Object Class.



Enter the details in the below screen and create a class.

Now, under the class, create an authorization object (ZCOUNCDE). Click on the authorization object button in the above screen. Enter the authorization fields and click on save.



Display Authorization Object

Object:

ZCOUNCDE

Text:

Country Code Authorization

Class:

ZDCL

Test DCL Authorizations

Author:

Authorization fields

Authorization Field	Short Description
ACTVT	Activity
ZAR_DLC_CC	DCL Country Code Test

Authorization Object Documentation

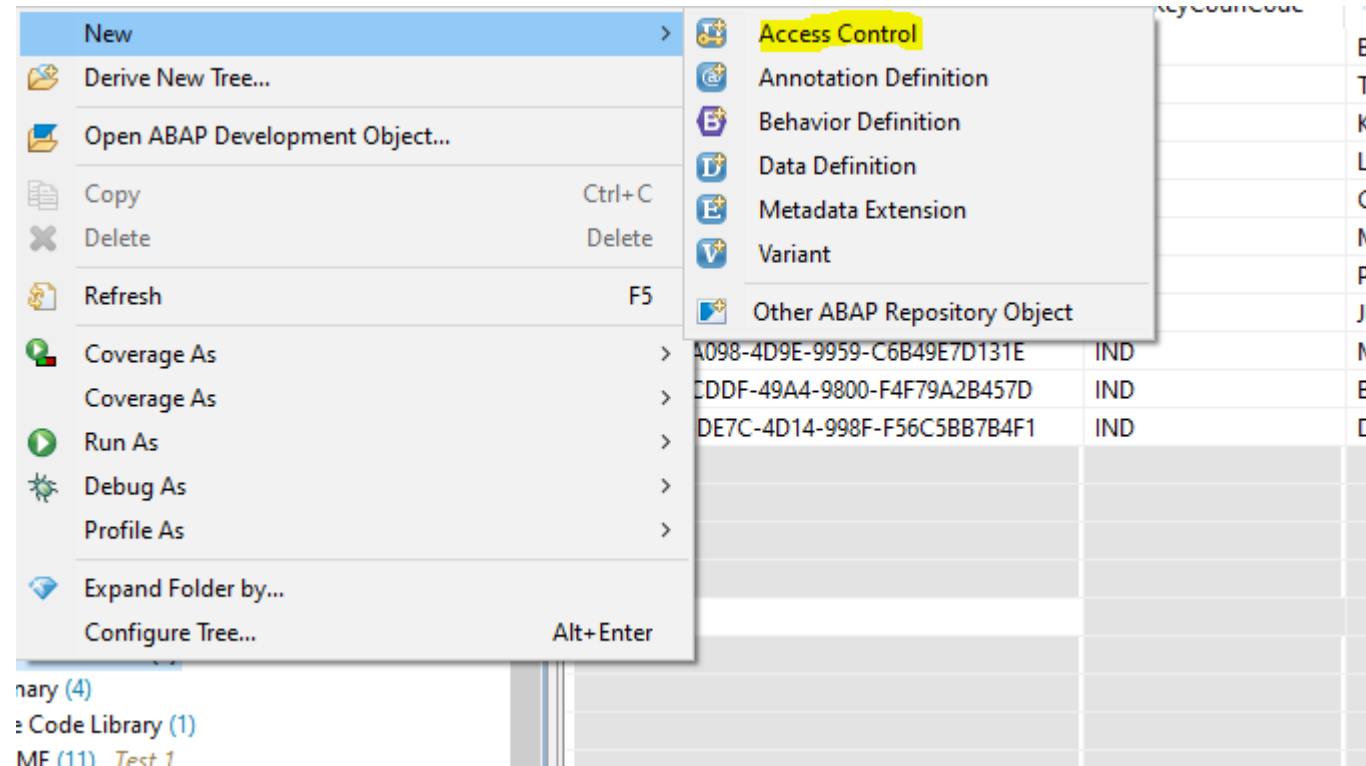
On clicking save, the below screen should appear:



✓ ZDCL	Test DCL Authorizations	Object Class
✎ ZCOUNCDE	Country Code Authorization	Authorization Object

Creating an Access Control with CDS

Now that we have an Authorization Object and a Authorization Field, we need to connect them with our CDS view to make it recognize the DCL. To do this, go to eclipse, right click on the package, click new and select Access Control



In the next dialog, Enter the name (ZDCL_C_CO) and description, click on next, in the next screen select an transport request, then click on next.



New Access Control

Access Control

Specify a value for field 'Name'

Project: * HA4_300_arj_en Browse...

Package: * ZARJ_DCL Browse...

☐ Add to favorite packages

Name: *

Description: *

Original Language: EN

? < Back Next > Finish Cancel

Now, in the below screen, for this example, we shall use an "Define Role with Simple Conditions" template. Select this and click on finish.

New Access Control

Templates
Select one of the available templates.

☒ Use the selected template

<ul style="list-style-type: none"> Define Role with Simple Conditions Define Role with PFCG Aspect Define Role with Inherited Conditions Define Role with Generic Aspect Define Generic Aspect Define Role with Unrestricted Access 	Defines a role that grants instance-specific access to a CDS entity through literal conditions or current user name.
---	--

```

@EndUserText.label: '${dcl_source_description}'
@MappingRole: true
define role ${dcl_source_name} {
  grant
  select
  on
    ${cds_entity}
  where
    ${entity_element_1} = '${literal_value}'
    or ${entity_element_2} = aspect user;
  ${cursor}
}

```

Now add the below code in the newly generated access control:

```
@EndUserText.label: 'Access Control for ZDCL_C_CO CDS View' @MappingRole: true define role ZDCL_C_CO { grar
```

Here the authorization object and authorization field should be same as the one which was defined in the TCodes SU20 and SU21.

```

1 @EndUserText.label: 'Access Control for ZDCL_C_CO CDS View'
2 @MappingRole: true
3 define role ZDCL_C_CO {
4     grant
5         select
6         on
7             ZDCL_C_CO
8             where ( KeyCounCode ) = aspect pfcg_auth( ZCOUNCDE, ZAR_DLC_CC );
9
10 }

```

Now save and activate the Access Control.

Creating a role with the authorization object and assigning it to the user

To create a role, we use the TCode PFCG. Open PFCG and enter the desired role name and click on single role-

Role:

Short Description:

In the next screen enter the short description and go to the users tab. Click on edit and assign the user ids which are going to test and use this application.



Role

Role: ☐ Obsolete

Description:

Target System: ☒ No destination

[Description](#) [Menu](#) [Workflow](#) [Authorizations](#) [User](#) [MiniApps](#) [Personalization](#)

[Organizational Mgmt](#) [User Comparison](#) [Info](#)

User Assignments

User ID	User Name	From	to	I...
<input type="checkbox"/>				
<input type="checkbox"/>				
<input type="checkbox"/>				

Now go to the authorizations tab and click on display authorization data:

Role

Role: ☐ Obsolete

Description:

Target System: ☒ No destination

[Description](#) [Menu](#) [Workflow](#) [Authorizations](#) [User](#) [MiniApps](#) [Personalization](#)

Profile Name:

Profile Text:

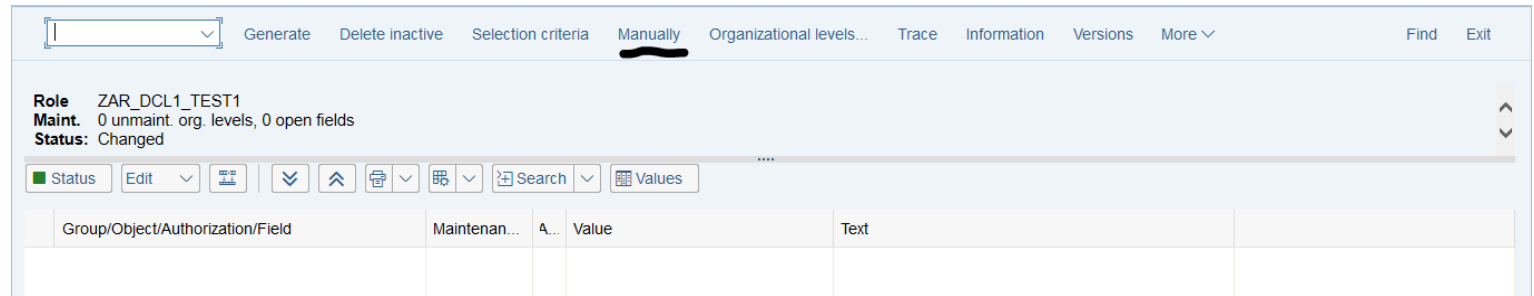
Status: Authorization profile is current

Edit Authorization Data and Generate Profiles

[Display Authorization Data](#)

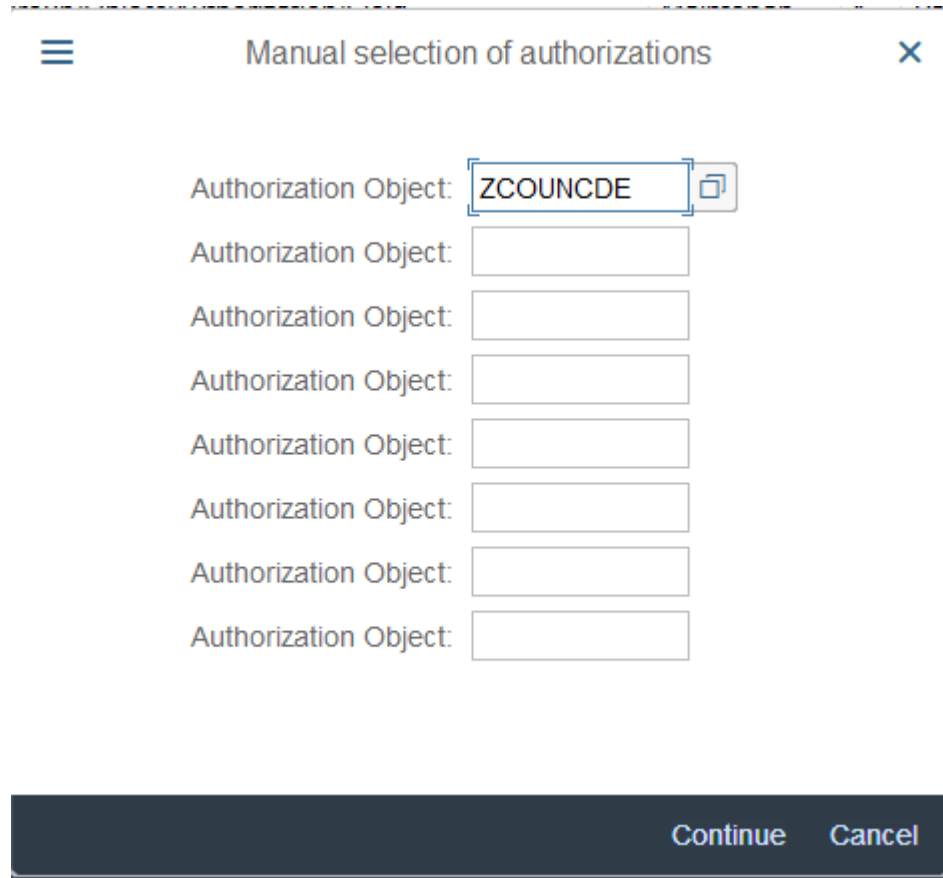
In the next screen, we are going to add our authorization object and assign what values we require displayed for the users having this role.

So, in the next screen, if there is any popup requesting for any template click on “Do not select any template”, the in the menu bar, click on the manually icon. This helps us in manually adding the authorization objects of our choice.







The screenshot shows the SAP authorization maintenance interface. The menu bar at the top includes options like 'Generate', 'Delete inactive', 'Selection criteria', 'Manually' (which is underlined), 'Organizational levels...', 'Trace', 'Information', 'Versions', and 'More'. Below the menu bar, the role 'ZAR_DCL1_TEST1' is displayed with maintenance statistics: 'Maint. 0 unmain. org. levels, 0 open fields' and 'Status: Changed'. A toolbar contains buttons for 'Status', 'Edit', and various icons for selection and search. Below the toolbar is a table with columns: 'Group/Object/Authorization/Field', 'Maintenan...', 'A...', 'Value', and 'Text'.

It will open a popup. There add the authorization object that we have created before and click on continue.



The screenshot shows a dialog box titled 'Manual selection of authorizations'. It contains a list of input fields for 'Authorization Object:'. The first field is populated with the text 'ZCOUNCDE' and has a small icon to its right. The other seven fields are empty. At the bottom of the dialog, there are two buttons: 'Continue' and 'Cancel'.

Now, expand the sub tree, and edit the desired authorization field to add restrictions.

	Group/Object/Authorization/Field	Maintenan...	A...	Value	Text	
<input type="checkbox"/>	Object Class ZDCL	Manual			Test DCL Authorizations	
<input checked="" type="checkbox"/>	Authorization Object ZCOUNCDE	Manual			Country Code Authorization	
<input type="checkbox"/>	Authorization T-H489011800	Manual			Country Code Authorization	
<input type="checkbox"/>	ACTVT	Manual			Activity	
<input type="checkbox"/>	ZAR_DLC_CC	Manual			DCL Country Code Test	

To add full authorization enter * or click on full authorization. Else, we can manually enter values or ranges that we require to only appear for the particular user.



Field values








Object: ZCOUNCDE Country Code Authorization

Field Name: ZAR_DLC_CC DCL Country Code Test

Full authorization

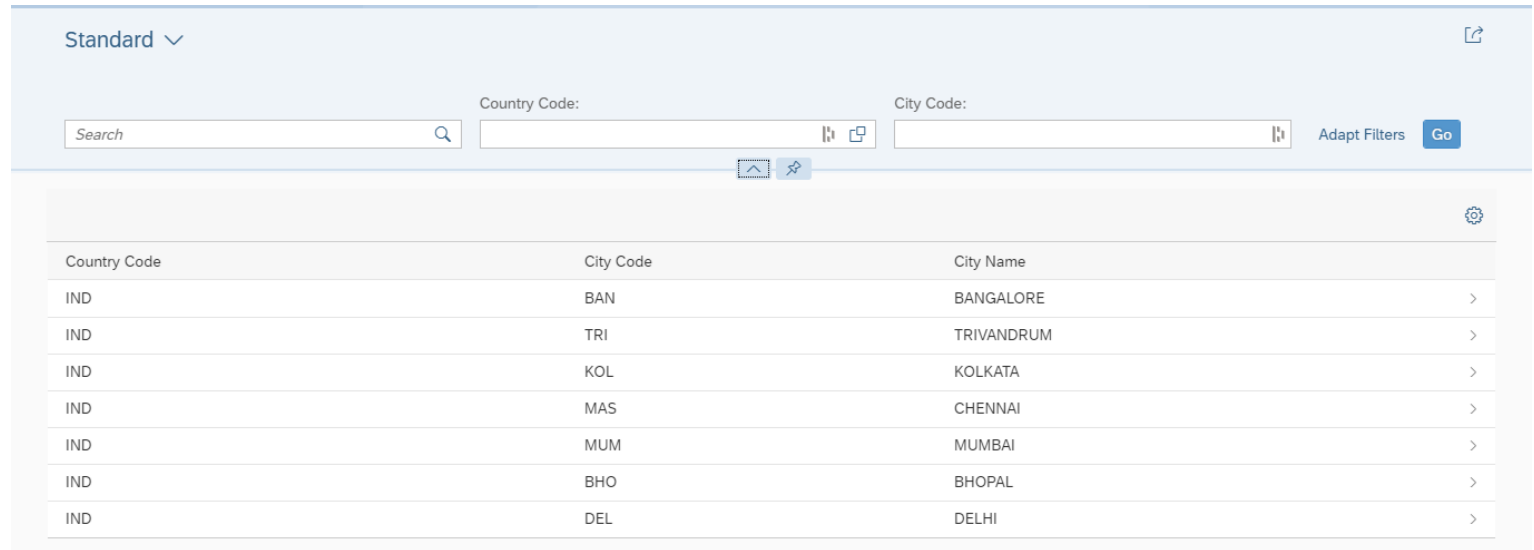
Value Intrvl

	'From'	'To'	
<input checked="" type="checkbox"/>	IND		
<input type="checkbox"/>			
<input type="checkbox"/>			
<input type="checkbox"/>			
<input type="checkbox"/>			
<input type="checkbox"/>			
<input type="checkbox"/>			
<input type="checkbox"/>			
<input type="checkbox"/>			
<input type="checkbox"/>			



In our case, we need to display the records with country code "IND". So, we can enter the same in the from column and click on save. Now, save the authorization data and click on generate.

Also, now we can go to the browser and refresh the FIORI application:



The screenshot shows the SAP Fiori application interface. At the top, there is a header bar with the text 'Standard' and a dropdown arrow. Below the header, there is a search bar with the placeholder text 'Search' and a magnifying glass icon. To the right of the search bar, there are two input fields: 'Country Code:' and 'City Code:'. The 'Country Code:' field contains the value 'IND'. To the right of these fields, there are icons for 'Adapt Filters' and a 'Go' button. Below the input fields, there is a table with three columns: 'Country Code', 'City Code', and 'City Name'. The table contains eight rows of data, all with 'IND' in the 'Country Code' column. The 'City Code' values are BAN, TRI, KOL, MAS, MUM, BHO, and DEL. The 'City Name' values are BANGALORE, TRIVANDRUM, KOLKATA, CHENNAI, MUMBAI, BHOPAL, and DELHI. Each row has a right arrow icon in the 'City Name' column.

Country Code	City Code	City Name
IND	BAN	BANGALORE
IND	TRI	TRIVANDRUM
IND	KOL	KOLKATA
IND	MAS	CHENNAI
IND	MUM	MUMBAI
IND	BHO	BHOPAL
IND	DEL	DELHI

We can see that in both the cases, we only get the records having country code as IND. Thus our DCL is working and adding the authorizations.

Note:

Through this method, we can add authorizations for the display functionality only. To add authorizations for other functionalities such as create, update and delete, we can place our authorization logic in the BOPF determination.

Please feel free to ask any questions in the comments.

Regards.

Read **18** times

Tweet

Published in SAP HANA

More in this category: « SAP HANA Interview Questions And Answers
Dynamic Images in SAP Analytics Cloud Story »

Part 1: How to Create



Leave a comment

Make sure you enter all the required information, indicated by an asterisk (*). HTML code is not allowed.

Message *

enter your message here...

Name *

enter your name...

Email *

enter your e-mail address...

Website URL

enter your site URL...



I'm not a robot

reCAPTCHA
[Privacy](#) - [Terms](#)

Submit comment

Join to SAPBAZAR.COM

I'm a SAP Freelancer

I'm a Customer



When it comes to Insurance,
why Window Sho

1 Crore Life Cover ·

50 Lakh Life Cover Option for
Better Half @ ₹367/month



BUY NOW

*T&C Apply

```
'; jQuery('body').append(panel); setTimeout(showPanel, 1200); function showPanel() {  
jQuery('#easycookieinfo').slideDown('slow'); } jQuery('.accept a').on('click', function() { var exp_date = new Date();
```

EasyCookieInfo



Accepted

```
exp_date.setTime(exp_date.getTime() + (1000*3600*24*365)); document.cookie =  
'cookie_32af4b2b4053a2018154d37f74abb95f=accepted;expires='+exp_date.toUTCString()+';path=/';  
jQuery('#easycookieinfo').slideUp('slow'); return false; }); });
```

Useful Links

[About Us](#)

[Terms & Conditions](#)

[Disclaimer](#)

[Support](#)

[Contact Us](#)

[Follow us on](#)

