



**Former Member**

September 9, 2017 5 minute read

## All About Data Control Language (DCLs)

[Follow](#)

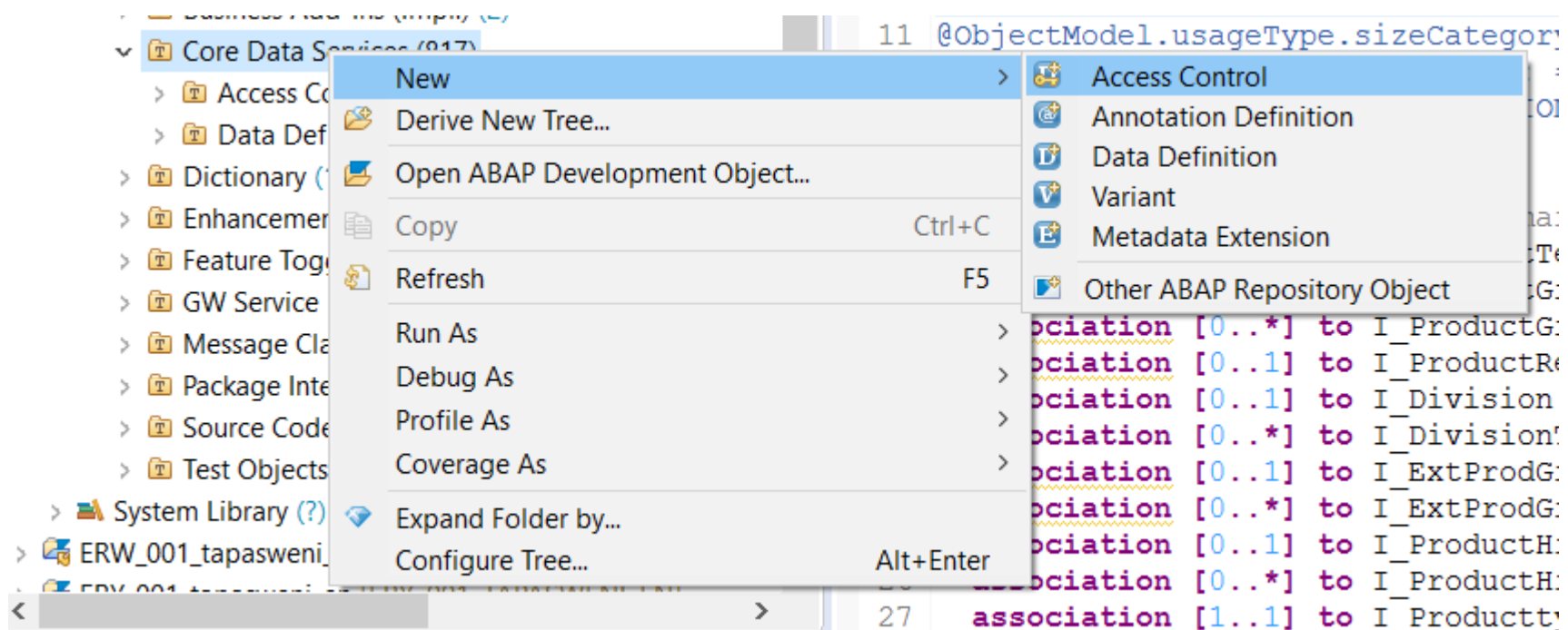
[RSS feed](#)

[Like](#)

4 Likes 8,088 Views 1 Comment

In this blog post I will be covering the features of Access Controls (Data Control Language) and how they can be used to enable row level authorization restriction on CDS views.

### Creating a new access control



Different types of DCLs

## 1. DCL in which relevant authorization objects are included on relevant fields

```
@EndUserText.label: '${dcl_source_description}'
@MappingRole: true
define role ${dcl_source_name} {
  grant
  select
  on
    ${cds_entity}
  where
    ${condition};
    // -- Example WHERE condition
    // -- Two-field mapping to PFCG authorization with filter on read authorization
    // ( SalesOrderID, OrgID ) = aspect pfcg_auth( S_ACM_DEMO, SACMTSOID, SACMORGUID, ACTVT = '03' )
    // and
    // -- Equals-or-initial operator
    // ( CustomerCountry ) ?= aspect pfcg_auth( S_ACM_DEMO, SACMCNTRY )
    // or
    // -- Reference to the logged on users name
    // CreatedBy = aspect User
    // or
    // -- Literal condition
    // isPublic = 'X';
}
```

Result set of a CDS view will be filtered by the restriction applied on fields included in the DCL of a CDS view. The relevant annotation included in CDS view is

```
@AccessControl.authorizationCheck: #CHECK
```

Multiple authorization objects can be clubbed together with

```
and
```

or

```
or
```

operator. Rules of evaluating the pfcg\_auth are mentioned after 4th type.

2. DCL which inherit the authorization from another DCL

```
@EndUserText.label: '${dcl_source_description}'
@MappingRole: true
define role ${dcl_source_name} {
  grant
    select
      on
        ${cds_entity}
        inherit
          ${super_role_name};
}
```

The authorization

objects included in the

`${cds_entity}`

will be the ones which are included in

`${super_role_name}`

Usage of such DCL is encouraged in connected CDS view. For example in the hierarchies like,

consumption view on

|– transactional view on

|–basic view on table

The DCL of basic CDS view will have relevant authorization object and transactional CDS view and Consumption CDS view will inherit the lower DCLs.

The relevant annotation included in CDS view is

```
@AccessControl.authorizationCheck:#CHECK
```

3. DCL with unrestricted access

```
@EndUserText.label: '${dcl_source_description}'
@MappingRole: true
define role ${dcl_source_name} {
    grant
        select
            on
                ${cds_entity};
}
```

Such a DCL is used when unrestricted access is

given to the user of the CDS view. Creating such a DCL on a CDS view declares that the data from the CDS view should be extracted without any restriction. This is very specific scenario based. The relevant annotation included in CDS view is

```
@AccessControl.authorizationCheck:#CHECK
```

4. DCLO which completely blocks the data from a CDS view

```
@EndUserText.label: '${dcl_source_description}'
@MappingRole: true
define role ${dcl_source_name} {
    grant
        select
            on
                ${cds_entity}
            where
                // Condition that is always false
                ${key_field} is null and ${key_field} is not null;
}
```

A DCLO or a access

inhibiting DCL is used when the data should be blocked completely from a CDS view entity. The data can be accessed using specific annotation in the CDS view which is trying to access data from a CDS view protected by a DCLO. The annotation included in CDS view is

```
@AccessControl.authorizationCheck:#PRIVILEGED_ONLY
```

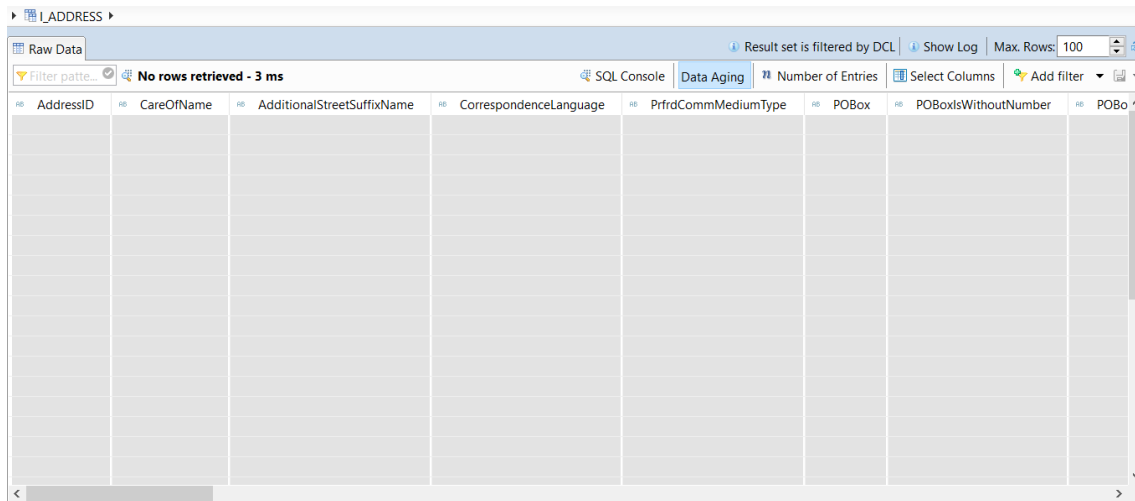
## Hierarchy of the evaluation of a PFCG condition

The following rule applies with respect to the hierarchy of the evaluation of a PFCG condition:

- If multiple authorizations are evaluated, the resulting conditions are joined using a logical “or”.
- In the conditions of each authorization used, the values for the authorization fields in question are joined using a logical “and”.
- If there are multiple values for an authorization field, they are joined using a logical “or”.

## Data Preview after creating a DCL/DCL0

In ABAP development tools in data preview the result set is always filtered by a DCL after having annotation as described above.



AddressID	CareOfName	AdditionalStreetSuffixName	CorrespondenceLanguage	PrfrdCommMediumType	POBox	POBoxIsWithoutNumber	POBox
-----------	------------	----------------------------	------------------------	---------------------	-------	----------------------	-------

If the annotation included in CDS view is

```
@AccessControl.authorizationCheck:#NOT_REQUIRED
```

and even if a DCL exist for such a CDS view the result set will not be filtered by the DCL.

## Inside a DCL

1. **Define Role** Provide a role name here

2. **Grant Select on**

This is the CDS View on which data restriction are to be applied through a DCL. DDL source name should be used and not the SQL view of the DDL source.

3. **Where** This is the field which is to be restricted. If there is an alias in the CDS view this, the alias should be used and not the technical name.

4. **Aspect pfcg\_auth** This is the place to include the authorization object and the fields on which it is applied with the ACTVT permitted activities. The possible value of ACTVT for that authorization object can be seen in TCODE SU21 as shown below. As most CDS views are used for reading, 03 is used which is for display.

5. **@MappingRole** This annotation must have the value true, so this role is assigned to all users in the system. The value false is not supported.

## Define role operators in a DCL

The following operators can be used in the where clause while defining a role. The operator compares a left side and a right side. The left side is always an element of the CDS entity to which the rule applies. The right side is represented using a literal value. The result of the expression is true or false.

operator	True if
=	The value of the left side is equal to the value of the right side.
<>	The value of the left side is not equal to the value of the right side.
<	The value of the left side is less than the value of the right side.
>	The value of the left side is greater than the value of the right side.
<=	The value of the left side is less than or equal to the value of the right side.
>=	The value of the left side is greater than or equal to the value of the right side.
?=	The value of left side can be blank or filled equal to the value on the right side.
IS [NOT] NULL	The value on the left side is (not) the <a href="#">null value</a> .

## Some Examples

- Multiple fields of a CDS view can be mapped to different authorization fields of an authorization object as shown below. The ACTVT fields described above decides the permitted activities allowed when the CDS view is accessed.

```
@MappingRole: 'true'
DEFINE ROLE demo_role {
  grant SELECT ON entity WHERE
    (cdsfield_1, cdsfield_2 ) = ASPECT pfcg_auth
      ( object,
        authfield_1,
        authfield_2,
        ACTVT = '03' );
}
```

If `?=` instead of `=` is used in the example above, the access condition is expanded as follows:

```
...
authfield_1 = 'A' OR
( ( cdsfield_1 IS NULL or cdsfield_1 = '' ) AND
  ( cdsfield_2 IS NULL or cdsfield_2 = '' ) ) )
```

- It is also possible to not map any field from the CDS view to the authorization object field. This means CDS access control prevents data from being read in full if the current user does not have at least an authorization for the authorization object object with the activity "03".



```

@MappingRole: true
DEFINE ROLE demo_role {
  GRANT SELECT ON entity WHERE
    ( ) = ASPECT pfcg_auth( authobject, ACTVT = '03' ); }

```

- Using the following code only those authorizations are used that contain all the permitted activities defined plus the authorization field country with the value "IN". Only the rows satisfying this condition from the CDS view are fetched matching the authorization field values.

```

@MappingRole: true
DEFINE ROLE demo_role {
  GRANT SELECT ON entity WHERE
    (cdsfield_1) = ASPECT pfcg_auth( object,
                                     authfield_1,
                                     ACTVT = '02',
                                     ACTVT = '03',
                                     country = 'IN' );
}

```

I hope this blog post was informative and helps you to secure the data of a CDS view using data control language (DCL).

Also details about **Restriction Type for Authorization fields used as part of DCLs**

1. The authorization object for example: X\_XX\_OBJ added in DCLs become part of PFCG role appearing in authorization tab from the relevant OData service SU22 data.

2. If you find that the added authorization object needs a restriction type for CLOUD from its View cluster (go to transaction SM34->give APS\_IAM\_VC as View cluster, then click on Display)  
Then follow steps as mentioned in [wiki](#) to add it to your Business catalog restriction OR create new Restriction type if required as per [wiki](#).
3. After creation or addition of restriction type check for XPRA creation with central IAM you can use [this](#) wiki.

You can contact my colleague [VENKAT BHARGAV A S.](#)

#### Alert Moderator

---

#### Assigned tags

[SAP S/4HANA Cloud](#) | [SAP S/4HANA](#) | [Security](#) | [DCL](#) |

---

#### Related Blog Posts

[How to maintain language for the Employee-Business Partner in S/4HANA](#)

By **Suresh Honnappanavar** , Sep 24, 2018

[Changing your Language Settings in SAP S/4HANA Cloud](#)

By **Anand Kapadia** , Oct 18, 2019

[Machine Learning, Manufacturing and Production and the OODA Loop](#)

By **Sven Denecken** , Dec 16, 2017

## Related Questions

---

[Error while reading REST API data in Custom Business Object in S4C](#)

By **Taranamjit Kaur Dhindsa** , Apr 08, 2020

[Fetch SAP S/4 HANA data using JDBC](#)

By **Vishal Garg** , Jun 17, 2019

[kba 2841820 - Restriction of Business Role SAP\\_BR\\_PROJ\\_FIN\\_CONTROLLER](#)

By **Guillaume Binot** , Sep 24, 2019

## 1 Comment

You must be [Logged on](#) to comment or reply to a post.

---



**Jacques-Antoine Ollier**

September 27, 2018 at 6:13 pm

Hello,

This is a good blog on the different type of DCL we can implement.

However, I am not sure this is new or not, but it seems that the type 3, Unrestricted access does not work like that anymore.

If we go with the syntax you mentioned, we obtain an error message mentioning:

**MAPPINGROLE is only supported in combination with ASPECT PFCG\_AUTH**

It seems the MappingRole is now waiting for a Where clause.

Maybe in order to do unrestricted access the new way is to mention @AccessControl.authorizationCheck:#NOT\_ALLOWED: to prevent any authorization check.

Thank you for this blog!

Best

Jacques-Antoine

Like (0)

Find us on

Privacy	Terms of Use
Legal Disclosure	Copyright
Trademark	Cookie Preferences
Newsletter	Support