Create a Simple ABAP CDS View in ADT

Beginner 20 min. ABAP Development, Beginner, SAP NetWeaver, Tutorial

You will learn how to create a CDS (Core Data Services) view using ABAP Development Tools (ADT).

You will learn

- How to use the new Core Data Services (CDS) tools in ABAP Development Tools for Eclipse (ADT).
- How to use the following ABAP and SQL elements in a CDS view:
- SELECT statement
- CASE statement
- WHERE clause

1 Step 1: Create a CDS view
2 Step 2: Enter the data source
3 Step 3: Use an existing CDS association
4 Step 4: Add fields from existing associations
5 Step 5: Add a CASE statement
6 Step 6: Add a WHERE clause
7 Step 7: Check your code and view your changes
8 Step 8: Test yourself

Back to Top


Julie Plummer September 18, 2023

Created by  March 9, 2023 Contributors 

Prerequisites

- You have a valid instance of an on-premise AS ABAP server, version 7.51 or higher (some ABAP Development Tools may not be available in earlier versions)
- You have run the transaction `SEPM_DG_OIA_NEW` or transaction `STC01 -> tasklist SAP_BASIS_EPM_OIA_CONFIG`. (If you do not, your CDS view will display empty.)
- **Tutorial**: **Create an ABAP Project in ABAP Development Tools (ADT)**
- **Tutorial**: **Create an ABAP Package**

> In this tutorial, you will create an ABAP Dictionary-based CDS view. As of ABAP AS 7.57, such views are deprecated. This tutorial is available for compatibility purposes only.For an short, up-to-date tutorial on CDS View Entities, see:**Tutorial**:**Create an ABAP Core Data Services (CDS) View in ABAP On-Premise**

DS is an extension of the ABAP Dictionary that allows you to define semantically rich ata models in the database and to use these data models in your ABAP programs. DS is a central part of enabling code push-down in ABAP applications.

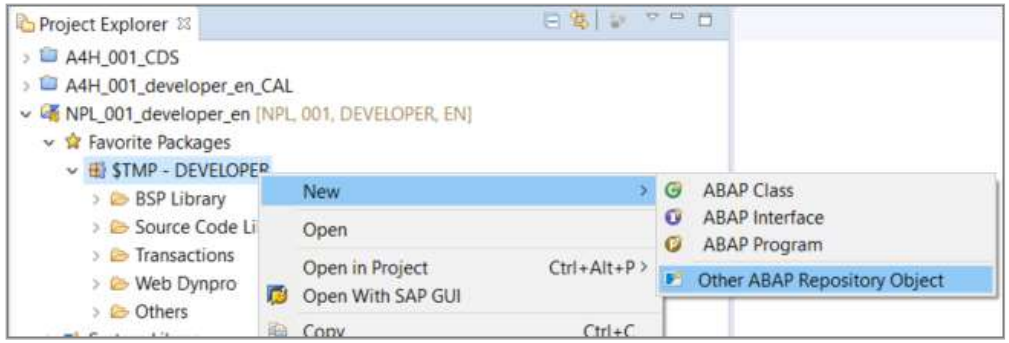
ou can find more information about these deprecated CDS Views here:- **ABAP eyword documentation, version 7.51: CDS Views**

ou can find more information about CDS View Entities here:- **ttps://help.sap.com/doc/abapdocu_latest_index_htm/latest/en-US/index.htm?le=abencds_v2_views.htm**- **SAP Community**.-Throughout this tutorial, objects ame include the suffix `xxx`. Always replace this with your group number or initials.

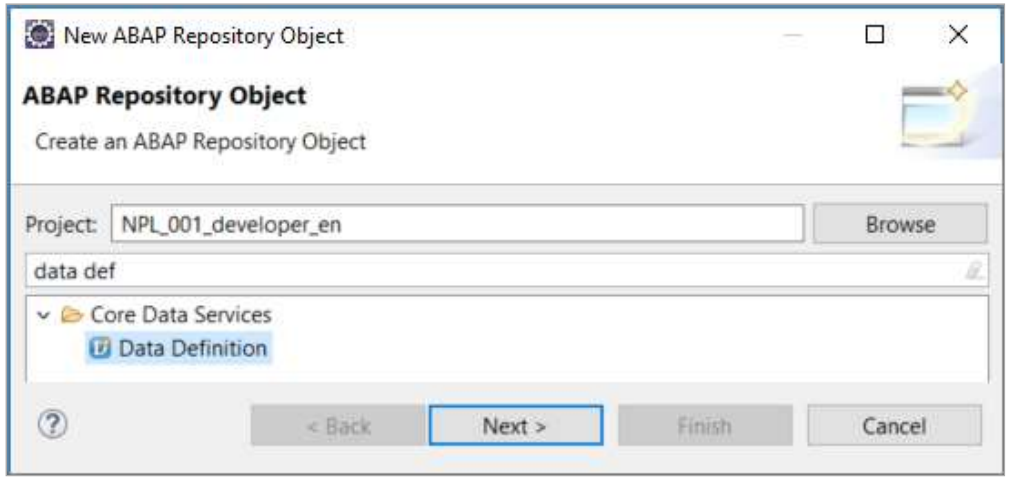
STEP 1

Create a CDS view

1. In the context menu of your package choose **New** and then choose **Other ABAP Repository Object**.



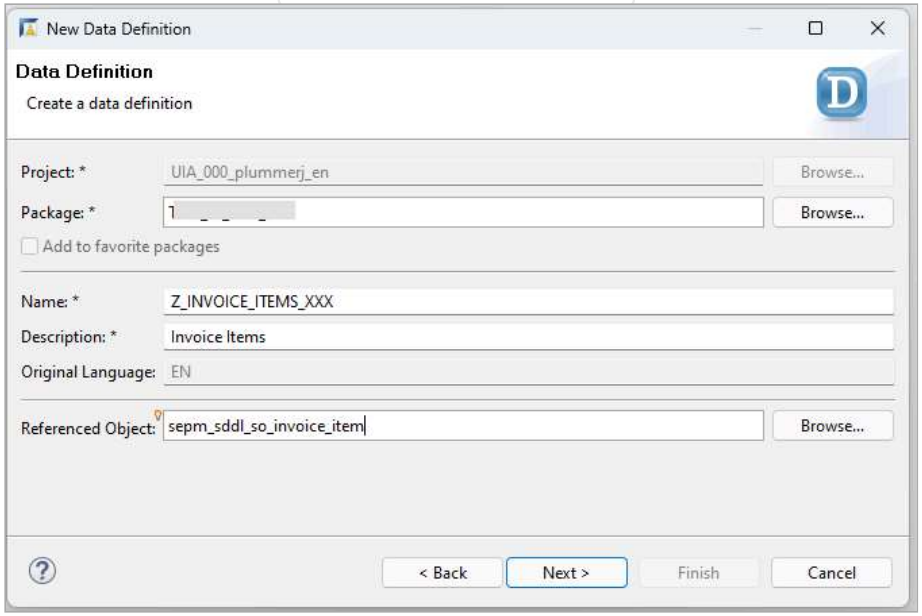
1. Select **Data Definition**, then choose **Next**.



1. Enter the following values, then choose **Next**:

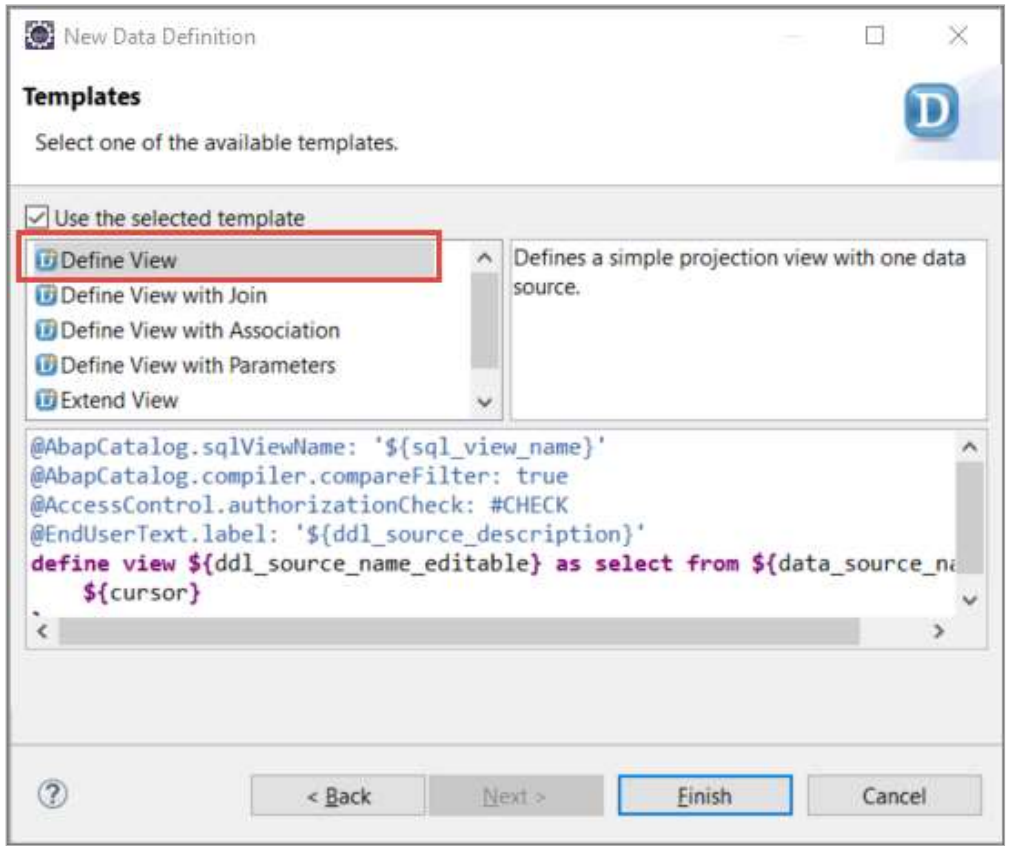
- Name = `Z_INVOICE_ITEMS_XXX`
- Description = **Invoice Items**

- Referenced Object: `sepm_sddl_so_invoice_item`



1. Accept the default transport request (local) by simply choosing **Next** again.

2. Select the entry **Define View**, then choose **Finish**



Log in to complete tutorial Done

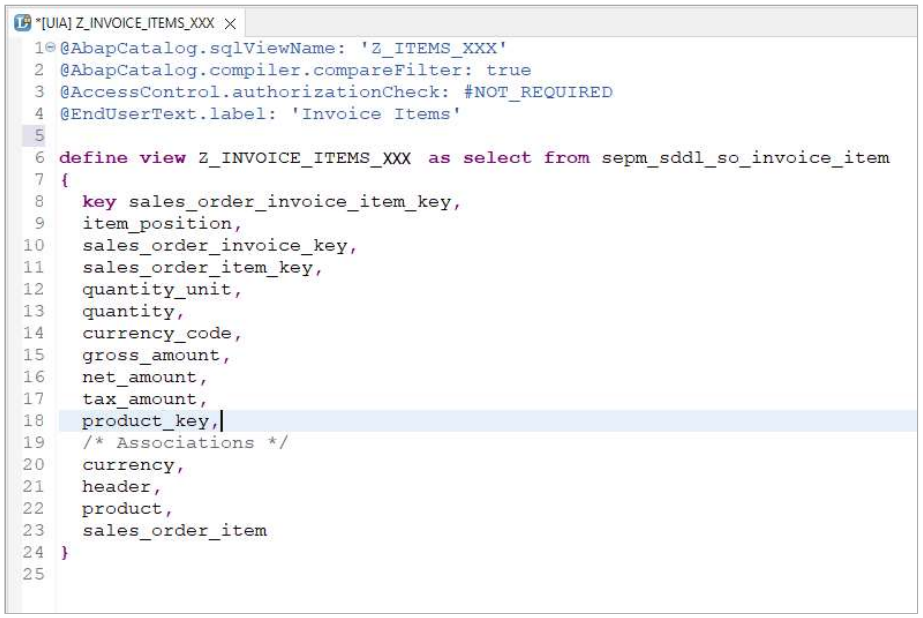
STEP 2

Enter the data source

The new view appears in an editor, with the fields from the referenced object, `sepm_sddl_so_invoice_item`. In this editor, enter the following values:

1. Enter `Z_ITEMS_XXX` as the SQL view name.

   Your CDS view should now look like this:



   > The SQL view name is the internal/technical name of the view which will be created in the database. `Z_Invoice_Items` is the name of the CDS view which provides enhanced view-building capabilities in ABAP. You should always use the CDS view name in your ABAP applications.

2. Delete all the fields except:

```
ABAP key sales_order_invoice_item_key, currency_code, gross_amount
```

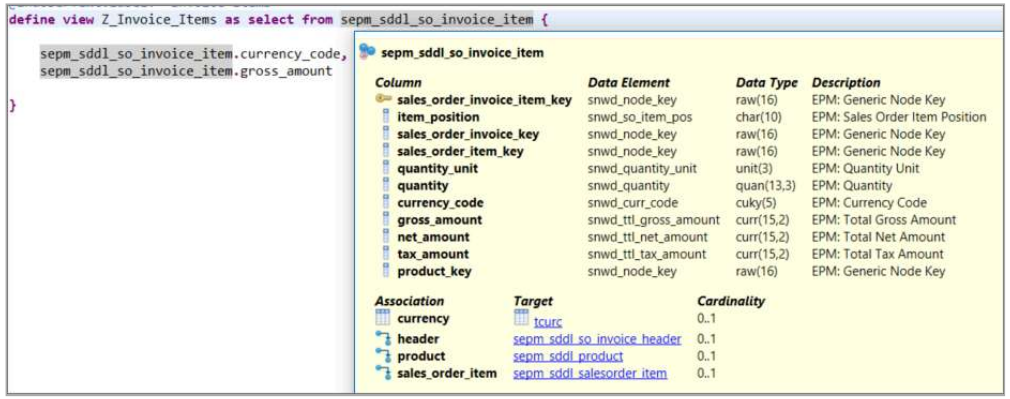
Log in to complete tutorial Done

STEP 3

Use an existing CDS association

You will now model the relationships between data sources by using some existing CDS associations. You can use associations in path expressions to access elements (fields and associations) in related data sources without specifying JOIN conditions. You can now display the element info by positioning the cursor on the data source name `sepm_sddl_so_invoice_item` and choosing **F2**.

To see the related data sources that can be accessed using associations, scroll down.To see details about the target data source of the association header, choose the hyperlink `sepm_sddl_so_invoice_header`.



Log in to complete tutorial Done

STEP 4

Add fields from existing associations

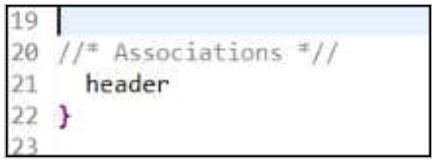
You will now add fields of related data sources to the SELECT list of `Z_Invoice_Items`, using the associations in path expressions. Each element in the path expression must be separated by a period.

1. Add the association `header` to your selection list, preferably with a comment, by adding the following the code. do not forget to add a comma after the previous item, `gross_amount`:

ABAP Copy

```
//      * Associations *//
        header

```



2. You will get an error, “Field header must be included in the selection list together with field `SEPM_SDDL_SO_INVOICE_ITEM.SALES_ORDER_INVOICE_KEY`”. Resolve this by adding the field `sepm_sddl_so_invoice_item.sales_order_invoice_key` to the Select statement.

3. Add the `company_name` of the business partner to the SELECT list using the associations **header** and **buyer** in a path expression

4. Add the `payment_status` from the invoice header to the SELECT list using the association **header**

ABAP Copy

```
header.payment_status
```

```
define view Z_Invoice_Items_3
  as select from sepm_sddl_so_invoice_item

{
  header.buyer.company_name,
  sepm_sddl_so_invoice_item.sales_order_invoice_key,
  sepm_sddl_so_invoice_item.currency_code,
  sepm_sddl_so_invoice_item.gross_amount,
  header.payment_status,

//* Assoiations *//
  header
}
```

Log in to complete tutorial Done

STEP 5

Add a CASE statement

If the invoice has been paid, you want to set the `payment_status` to X (true). Do this by implementing a CASE expression, assigning the alias `payment_status` to the CASE expression.

Remove the existing declaration, `header.payment_status,` and replace it with the following code. Do not forget to separate the new, calculated field `paid` and the association `header` with a comma.

ABAP Copy

```
case header.payment_status
    when 'P' then 'X'
    else ' '
end as paid,

```

```
        case header.payment_status
            when 'P' then 'X'
            else ' '
        end as paid,

        //      * Associations *//
        header
}
```

You can check your code below.

STEP 6

Add a WHERE clause

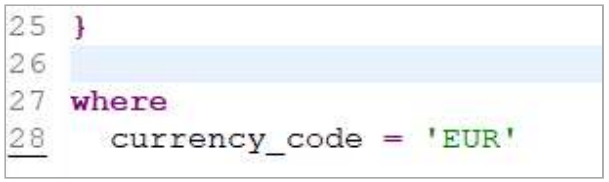
You will now filter the results so that only invoice items with `currency_code = 'EUR'` are retrieved.

1. Add a WHERE clause:

ABAP Copy

```
WHERE currency_code = 'EUR'

```



2. Save and activate the data definition by choosing **Save** (`Ctrl+S`) and **Activate** (`Ctrl+F3`).

Log in to complete tutorial Done

STEP 7

Check your code and view your changes

STEP 8

Test yourself

Which of the following is not a valid association path? Choose one answer only.

- sepm_sddl_so_invoice_item.product.changed_by
- sepm_sddl_so_invoice_item.product.dim_unit
- sepm_sddl_so_invoice_item.product.supplier
- sepm_sddl_so_invoice_item.product.storage_bins

Log in to complete tutorial Check answer

ext Steps

TUTORIAL

Display a CDS View Using ALV with IDA

15 min.

Find us on