S4Explorer – User Manual

Key Contacts

Name	Area of Expertise	Email	Phone
Ralf Slofstra	Business Intelligence	Ralf_Slofstra@epam.com	
Mitchell Beekink	Business Intelligence	Mitchell_Beekink@epam.com	
Roland Bouman	Business Intelligence	Roland_Bouman@epam.com	

Document Review

Date	Version	Name	Role	Comments

Document History

Version	Reason for change	Date
0.1	Initial version	22-10-2021
0.2	Revision	04-02-2022

Contents

1 Overview

In this chapter you will find general information about the S4Explorer Application.

1.1 Running the App

S4Explorer is a browser-based application or webapp. To run it, you simply open a URL in your internet browser.

1.1.1 Prerequisites

To run S4Explorer you need the following:

- A modern web browser, like Google Chrome, Mozilla Firefox, or Microsoft Edge.
- You need to have user access to S/4 HANA system that serves S4Explorer
- You need to know the hostname and port for any endpoints that serve the S4Explorer app. These will typically be S/4 HANA servers, S/4 HANA application servers, or Servers running a SAP Web dispatcher that re-routes application requests to a S/4 HANA server.

S4Explorer will in theory run on any device that runs a web browser.

1.1.1.1 Device Types

In practice, a desktop computer will offer the best experience. A tablet may still work reasonably well. You can try to run S4Explorer on a mobile phone but currently such a device is not recommended.

1.1.2 Starting the App

S4Explorer is a web application. You can run it simply by navigating to the appropriate S4Explorer URL using your web browser. The URL will typically look something like this:

https://<host>:<port>/sap/bc/ui5_ui5/sap/zs4explorer/index.html

The appropriate value for <host> and <port> depends on which S/4 HANA system you want to inspect. Typically, S4Explorer is served directly from a S/4 HANA server, and that instance of S4Explorer can be used to work with the S/4 HANA instance that hosts it.

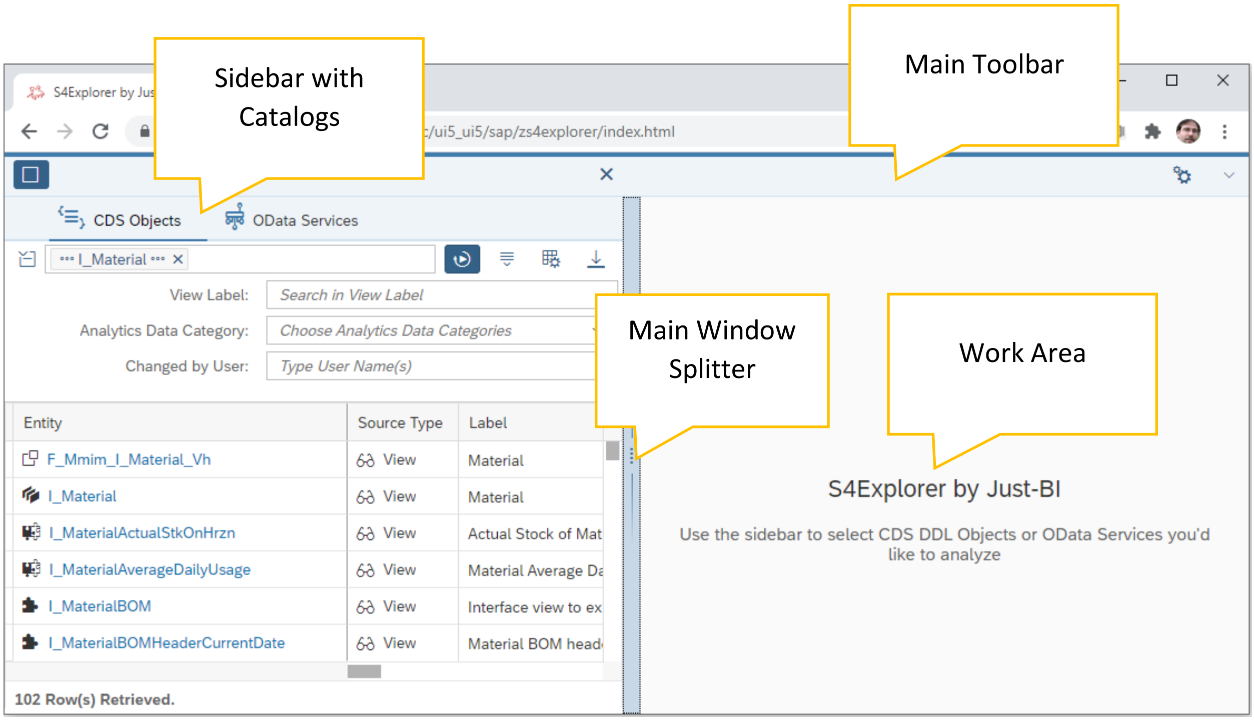
1.1.3 Logging on

Typically, S/4 HANA instances are configured with Single Sign On (SSO). A window may popup prompting you to select a particular certificate.

Alternatively, if SSO is not enabled, a Basic Authentication dialog may appear, prompting you to enter your credentials. If this is the case, enter your SAP username as User, and the user’s SAP password as password.

1.2 Main Screen

Below is a screenshot of the S4Explorer main screen:



View Label: Search in View Label

Analytics Data Category: Choose Analytics Data Categories

Changed by User: Type User Name(s)

Entity	Source Type	Label
F_Mmim_I_Material_Vh	View	Material
I_Material	View	Material
I_MaterialActualStkOnHrzn	View	Actual Stock of Mat
I_MaterialAverageDailyUsage	View	Material Average Da
I_MaterialBOM	View	Interface view to ex
I_MaterialBOMHeaderCurrentDate	View	Material BOM head

102 Row(s) Retrieved.

S4Explorer by Just-BI

Use the sidebar to select CDS DDL Objects or OData Services you'd like to analyze

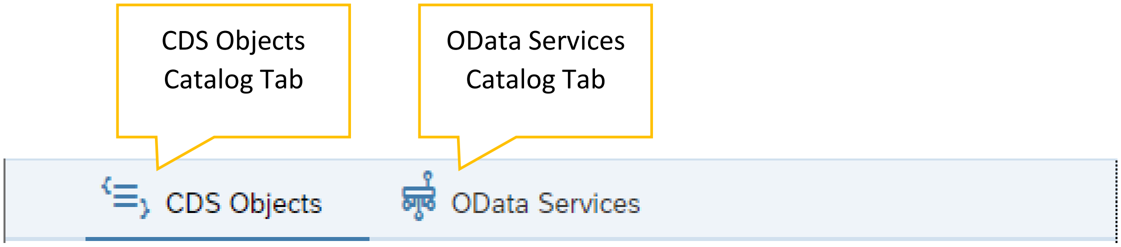
1.2.1 Sidebar

The *Sidebar* is on the left-hand side in the application window. It contains Object Catalogs that enable you to find objects that you can then explore in more detail.

Currently S4Explorer provides 2 catalogs:

- **Error! Reference source not found.**: use this catalog to work with S/4 HANA CDS objects, like views, table functions and so on.
- The OData Services Catalog: use this catalog to work with OData services.

The sidebar is essentially a tab container, presenting each catalog as a separate tab item:



1.2.2 Work Area

The Work Area is on the right side of the application window. When you select objects from a Object Catalog (in the sidebar), this area is used to show the details of those objects.

1.2.2.1 Object Detail Page

Objects in the Work Area are presented as tab pages which show you the details of that particular object. You can open as many objects as you like; each will be available in its own tab.

1.2.2.1.1 Object Detail Header Panel

The Object Detail Page has a header which contains general information about the object.

1.2.2.1.2 Object Detail Tabs

1.2.3 Main Toolbar

The Main Toolbar is at the top of the application window. It contains a few buttons and a menu that are applicable to the application as a whole.

1.2.3.1 Expand/Collapse sidebar button

This toggle button is on the far left side of the toolbar and lets you minimize or restore the width of the **Error! Reference source not found.**.

- ■ By default, it is depressed, and the sidebar is expanded (i.e. has a non-zero width)
- ◫ When it is not depressed, the sidebar is collapsed (i.e., has a zero-width); thereby maximizing the size of the Work Area.

1.2.3.2 Close All Button

✕ This button lets you close all tab pages that are currently loaded in the **Error! Reference source not found.**.

1.2.3.3 Settings Dialog Button

⚙ Pressing this button will open the Settings Dialog.

1.2.3.4 Links menu button

⌄ Clicking this will open the Links menu. The links menu lets you visit various webpages and resources related to S4Explorer. It also lets you open the About Dialog which contains information about the current S4Explorer version.

1.2.4 Main Window Splitter

The Main Window Splitter is a vertical divider that separates the **Error! Reference source not found.** and the **Error! Reference source not found.**. The Splitter can be dragged vertically to control the size of the Sidebar and Work Area.

1.3 General Workflow

S4Explorer support many different use cases, but the general workflow is always the same:

1. Choose an appropriate Object Catalog, depending upon the kind of objects you want investigate.
2. Use the Catalog Search Form to define search criteria.
3. Browse the Catalog Results Table and use the Links to drill down on objects of interest to open then in the **Error! Reference source not found.**

4. Explore the object by selecting any of the Object's views, which summarizes a particular aspect of the object. Some of these views show the object in a Diagrams to support visual exploration, whereas other types of information are more conveniently displayed in Result Tables. In many cases, the diagrams and tables contain links that allow you to further drill down and through to other, related objects.
5. Copy or download your exploration results for further processing, sharing or collaboration.

1.3.1 Choosing a Catalog

A S4Explorer Object Catalog is a searchable and browsable collection of objects of a particular type. Currently S4Explorer supports the following catalogs:

- **Error! Reference source not found.**: use this catalog if you want to work with SAP CDS objects, like CDS Views, CDS View Extensions or Table Functions
- The OData Services Catalog: use this catalog to find OData services.

All Object catalogs are accessible through the **Error! Reference source not found.**, where each catalog is available as a tab page. You can select a particular catalog simply by clicking its tab. After selecting a particular catalog tab, its search form and results area are revealed in the sidebar.

Each catalog consists of

- Catalog Search Form: you can find this search form in the top of each catalog. The search form lets you specify criteria to find the objects you need.
- Catalog Results Table: The bottom half of the catalog is where the search results of the most recent query are displayed in a data grid.

1.3.2 Define Search Criteria

At the top of the Object catalog is the Catalog Search Form. You can use the Catalog Search Form to specify search criteria.

Depending on the type of catalog, you can choose from multiple fields corresponding to searchable attributes appropriate for the type objects in that catalog.

By default, entering or modifying search criteria will automatically trigger execution of the query and update the Catalog Results Table.

1.3.3 Review Search Results

Search results appear in the Catalog Results Table. The columns in the search results table represent attributes belonging to the kind of objects contained in the catalog.

Typically, you will define your search criteria so that the query yields a relatively small list of results. By scrolling through the list and looking at the various attributes of the objects in the search result, you decide which objects you want to inspect more closely.

1.3.4 Inspect Object Details

Catalog objects always have an attribute that represents the object name which lets you identify specific individual objects. In the results table, this identifier is represented as a link. Clicking the link will open a new Object Detail Page in the **Error! Reference source not found.** which shows you the details of that particular Object.

1.3.5 Export Analysis Results

While S4Explorer lets you find objects and gain insight in how they are built and related to objects, the journey typically does not stop there. In general you will want to communicate your insights and share them with your co-workers.

Various elements like result tables or diagrams can be exported and downloaded and shared as is, or embedded in other applications.

S4Explorer also enables you to easily copy results to the clipboard so you can easily import them into other applications and office documents.

2 Common UI Elements

2.1 Toolbar

Toolbars are horizontal bars that appear usually at the top of a page or tab. They usually contain a set of toolbar buttons which let you apply some action to the container below the toolbar.

2.2 Links

Links appear throughout S4Explorer:

- In cells of Result Tables
- In informational headers of Object Detail Pages
- In the links menu in various diagrams

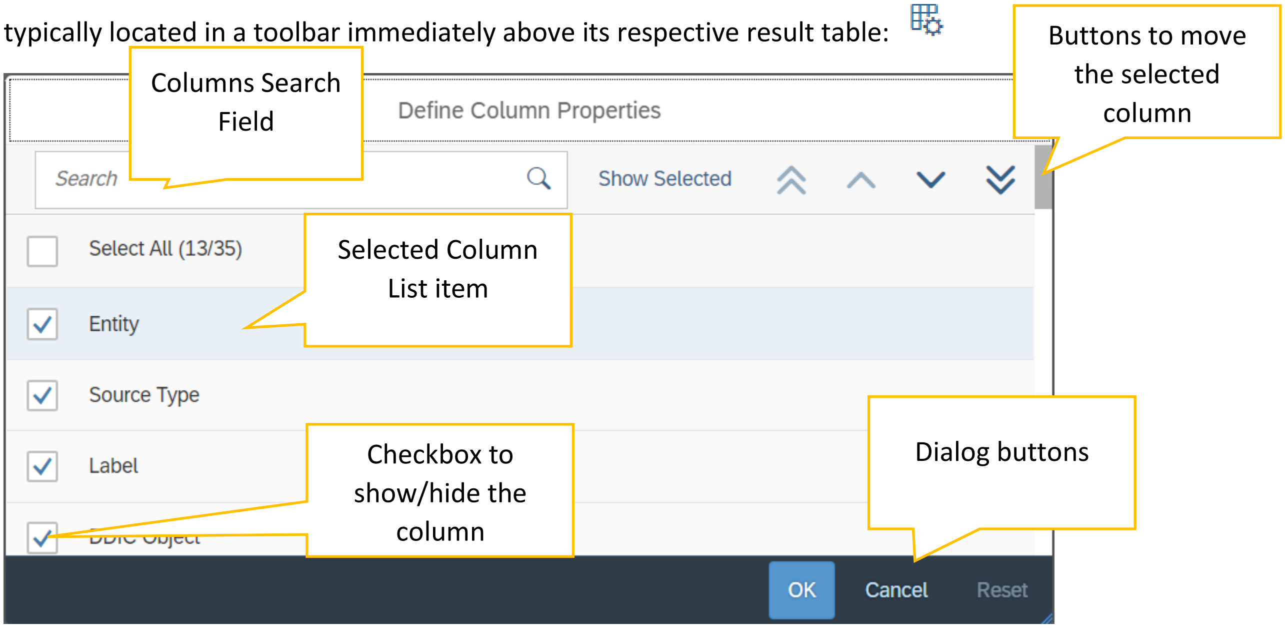
2.3 Result Table

Result tables or data grids appear throughout S4Explorer to summarize information in a tabular fashion, organized in rows and columns.

2.3.1 Table Settings Dialog

Each S4Explorer result table has an associated Table settings Dialog that allows you to control the visibility and order of the table's columns.

The Table Settings dialog for a particular table can be opened using the table settings dialog button, is typically located in a toolbar immediately above its respective result table:



□ Select All (13/35)
☑ Entity
☑ Source Type
☑ Label
☑ DDIC Object

Columns Search Field

Buttons to move the selected column

Selected Column List item

Checkbox to show/hide the column

Dialog buttons

The table settings dialog has a toolbar containing a number of controls, a list of columns and dialog buttons to confirm or cancel the dialog.

2.3.1.1 Table Settings Dialog Columns List

The main contents of the table settings dialog is the columns list. Each column of the result table is available here as a list item.

The column list items have a checkbox which can be used to control whether the corresponding column will be visible in the result table.

You can select items in the list by clicking on them. You can then use the buttons in the Table Settings Dialog Toolbar to control at the position of the column inside the result table.

2.3.1.2 Table Settings Dialog Toolbar

The Table Settings Dialog toolbar contains the following controls

- A Search field that lets you find columns by name. Simply type a part of the name of the column you want to find, and the column list will be automatically filtered to show only those columns that match the entered search term.
- Buttons to move the currently selected column:
    - ︽ Make this the first column
    - ︿ Move the column one position forward
    - ﹀ Move the column one position backward
    - ︾ Make this the last column

2.3.1.3 Result Table Toolbar

Typically, S4Explorer result tables have a toolbar at the top, and any controls in the toolbar are intended to work with the table. Generic controls that appear in multiple results tables are:

- ⚙ The Table Settings Dialog button. Clicking this will open the Table Settings Dialog for this results table.
- ⤓ Download Button. Clicking this will export the result table to a downloadable Excel workbook.

2.4 Exporting and Sharing

The content of S4Explorer Result Tables and Diagrams can be shared and used outside S4Explorer. There are 2 different ways to do this:

- Copy S4Explorer data to the clipboard. After copying data to the clipboard it may be pasted in other applications
- Download S4Explorer content to a local file. Once downloaded, the file may be shared with other users, or embedded into office documents.

2.4.1 Copying to the Clipboard

S4Explorer provides specific copy actions to offer a copy experience that is more appropriate than the browser’s generic copy functionality:

- Quick Copy Icons
- Copy toolbar button
- Result table context menu copy actions

2.4.2 Downloading

In various sections of S4Explorer you will have the option to download images, tables or other objects to view and edit on your local machine.

- Download Button. This indicates that an object is available for download.

2.5 Object Catalog

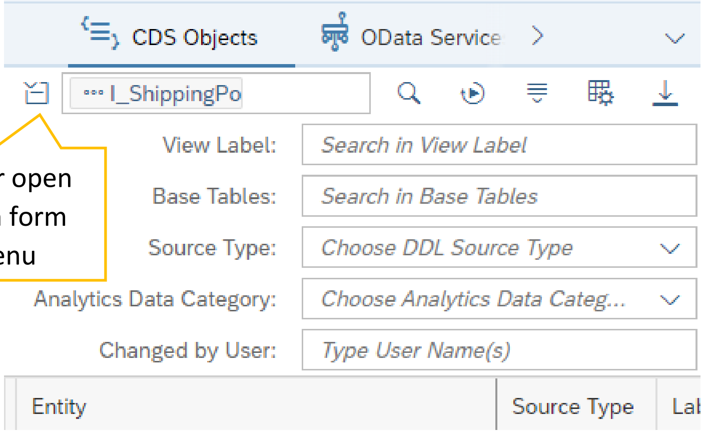
An Object catalog is a browsable and searchable collection of objects of a particular type.

2.5.1.1 Catalog Search Form

The catalog search form collectively allows the user to search for, compare and select entities that he / she wants to analyze.

2.5.1.2 Search Form Field Menu

The search form field menu is located in the sidebar and allows the user to enter values used to filter and find entities from the catalogs.



Collapse or open the search form field menu

- The **Search** button executes the search using the provided input.
- The **Auto-Query** button can enable or disable auto-query. When enabled, S4Explorer will monitor the filter fields for change, and automatically execute the query to refresh the results as needed. This eliminates the need to manually search for items.

- The **Filter Menu** button pops-out the available search fields. Here, the user can select and / or deselect any fields that should be displayed in the search menu.
- The **Table Settings** button opens a dialog window. Here, the user can select and / or deselect any fields that should be displayed in the results table.

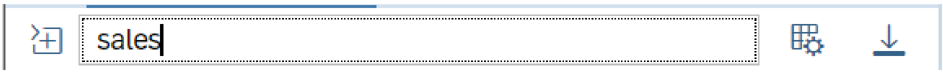
2.5.1.3 Smart Search Fields

Using the columns search field the user can perform a variety of searches on both the CDS Objects Catalog as well as OData Services. The following section will briefly demonstrate various search functionalities to enhance search performance.

2.5.1.3.1 Basic Search by Name

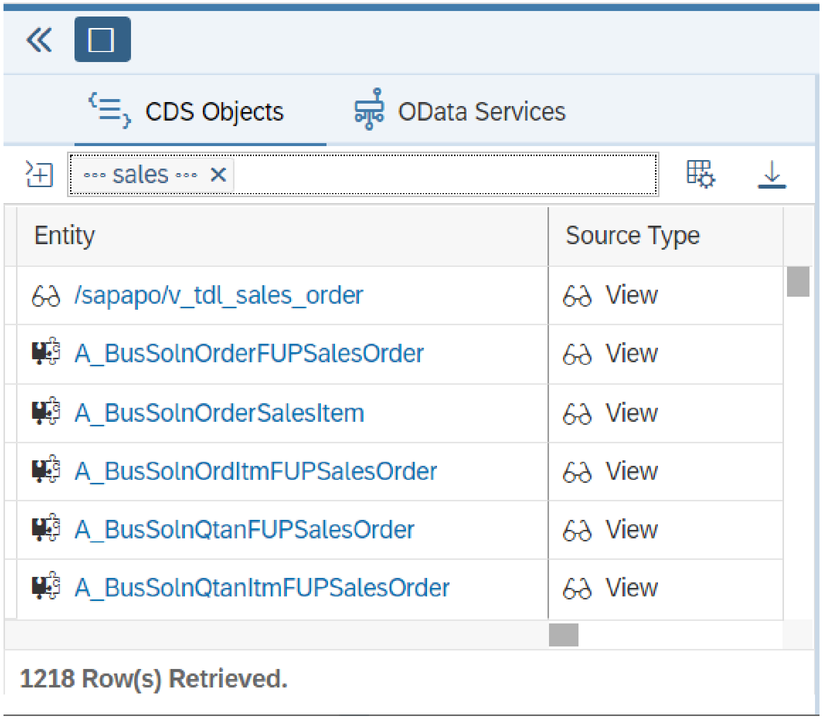
The Object Catalog Toolbar provides an input field for quick and basic search by object name. In case of CDS objects, this basic search functionality lets you find objects by entity name.

To start searching for particular objects, focus the search field and simply type a search term. In this example, we entered the search term sales:



sales

To actually initiate the search, hit “Enter” after you typed your search term. By default, S4Explorer monitors changes in the search field, and executes the query automatically when needed. In this case, the search term matched 1218 results:



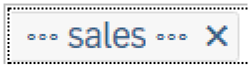
··· sales ··· ✕

Entity	Source Type
/sapapo/v_tdl_sales_order	View
A_BusSolnOrderFUPSalesOrder	View
A_BusSolnOrderSalesItem	View
A_BusSolnOrdItmFUPSalesOrder	View
A_BusSolnQtanFUPSalesOrder	View
A_BusSolnQtanItmFUPSalesOrder	View

1218 Row(s) Retrieved.

2.5.1.3.2 Search Tokens

After hitting Enter, the search term is processed and turned into a display element called a “Token”, which appears as a light grey box containing a few icons and the search term text:



sales

The ellipsis or “…” icons that are visible on both sides of the search term text indicate that S4Explorer searches for objects where the search term appears anywhere within the object name. So, the ellipsis indicates the variable part of the matched items. For example, in the search results, you will find entities like /sapapo/v_tdl_**sales**_order and A_BusSolnOrderFUP**Sales**Order and so on.

By clicking on the “decline” icon (the X that appears at the far-right side of the search token) you can remove the search term. You can also remove the token by pressing the Backspace key on the keyboard. That lets you remove whatever search token appears right before the position of the cursor.

2.5.1.3.3 Edit and Refine Your Search

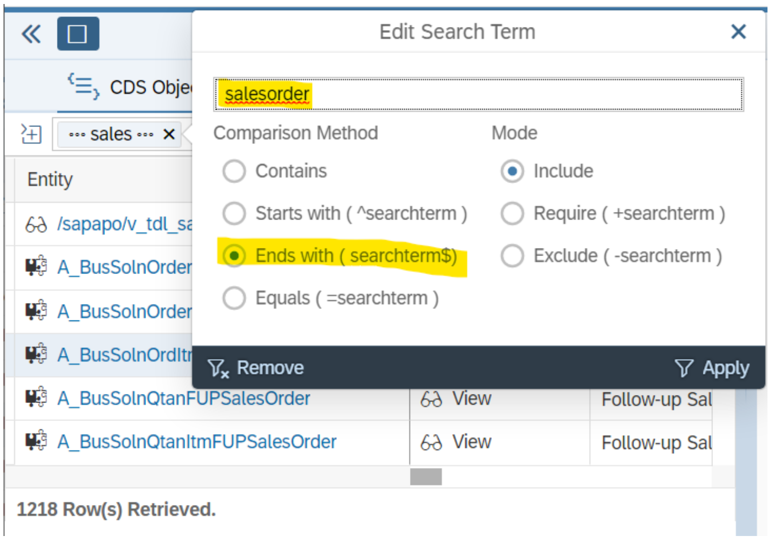
The basic “Contains” search behaviour is often sufficient. But sometimes, you need something more specific and precise.

For example, our initial example search term Sales gave us a number of matches ending with Sales Order, like A_BusSolnOrderFUPSalesOrder and A_BusSolnOrdItmFUPSalesOrder. Suppose that we now want to find only those objects that end with the term “Sales Order”.

A single click on the search token reveals the Edit Search Term popup dialog that lets you edit the search term.

2.5.1.3.4 Comparison Method: contains, starts with, ends with

In the popup window, we can modify the search term text, and we can also choose the comparison method. The screenshot bellow shows we changed the search term to salesorder and chose “Ends with” as comparison method.



Edit Search Term

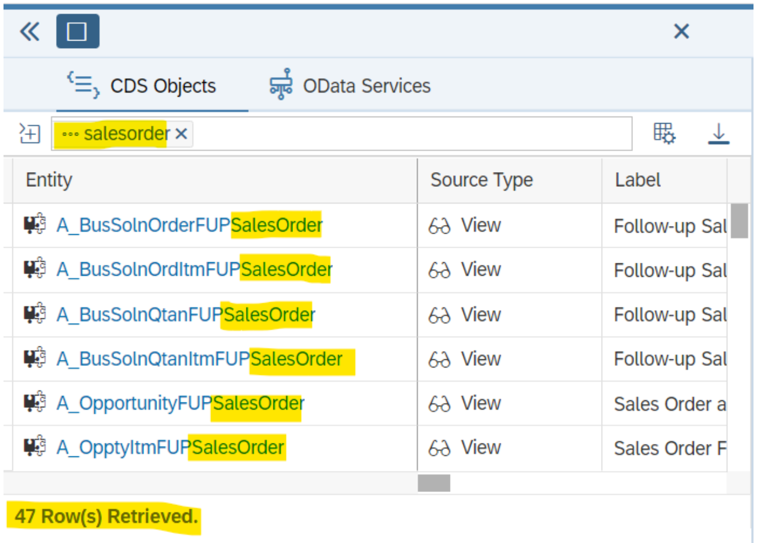
salesorder

Comparison Method	Mode
○ Contains	● Include
○ Starts with (^searchterm)	○ Require (+searchterm)
● Ends with (searchterm$)	○ Exclude (-searchterm)
○ Equals (=searchterm)	

Entity		
/sapapo/v_tdl_sa		
A_BusSolnOrder		
A_BusSolnOrder		
A_BusSolnOrdItr		
A_BusSolnQtanFUPSalesOrder	View	Follow-up Sal
A_BusSolnQtanItmFUPSalesOrder	View	Follow-up Sal

1218 Row(s) Retrieved.

When we confirm the edits to the search term by clicking the apply button, the popup closes, and the search token is updated. The new query is executed automatically in response to the updated search token:



Entity	Source Type	Label
A_BusSolnOrderFUPSalesOrder	View	Follow-up Sal
A_BusSolnOrdItmFUPSalesOrder	View	Follow-up Sal
A_BusSolnQtanFUPSalesOrder	View	Follow-up Sal
A_BusSolnQtanItmFUPSalesOrder	View	Follow-up Sal
A_OpportunityFUPSalesOrder	View	Sales Order a
A_OpptyItmFUPSalesOrder	View	Sales Order F

47 Row(s) Retrieved.

As you can see in the screenshot, the ellipsis icon in the search token now only appears before the search term text. This indicates that items will be matched only if they end with the search term. The now yields 47 results, and each indeed ends with SalesOrder.

2.5.1.3.5 Multiple Search Tokens

After entering a search term, the search field maintains its focus, and the cursor is positioned immediately after the search token. This is convenient as it lets you type multiple search tokens.

When multiple search tokens are present, S4Explorer's default behaviour is to apply all search tokens in an inclusive manner (logic OR). For example, suppose you're interested in finding views about materials and products. You could enter the search terms product and material, and this will find you all CDS objects whose name contain either “product” or “material” (or both).

2.5.1.3.6 Comparison Mode: include and require

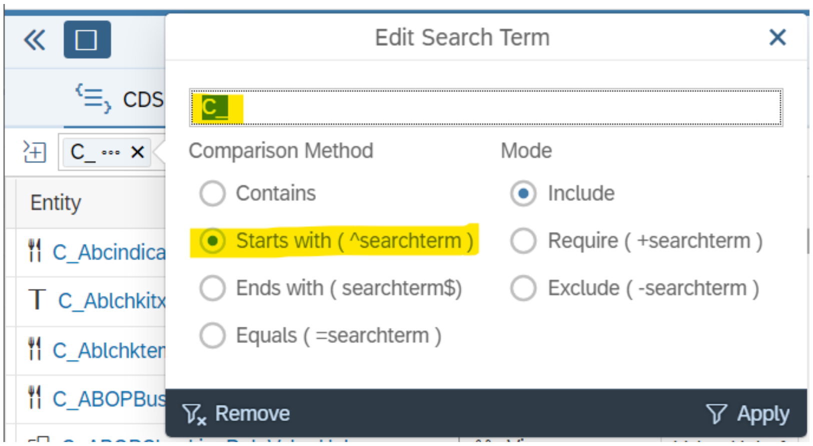
For yet more advanced search capabilities you can control how multiple search terms affect the query.

We just mentioned that by default, multiple search terms are applied inclusively, meaning that items that match either of the search terms (or both!) are included in the result (logic OR). But sometimes you want to consider only those items that contain all of the search terms.

For example, suppose we’re interested in objects whose name contains the text SalesOrder, but which are also prefixed with a certain string.

Searching by prefix can be very useful as standard SAP objects adhere to a naming convention where the prefix conveys information about the intended use of the view. For example, the prefix I_ indicates that this is an intermediate object, not intended for direct consumption, whereas a prefix C_ indicates the object is likely suitable for consumption by a report or application.

Suppose we are interested in views prefixed with C_, which also contain the search term SalesOrder. We can achieve this by first entering a search term for C_ and choosing the “Starts with” comparison method:



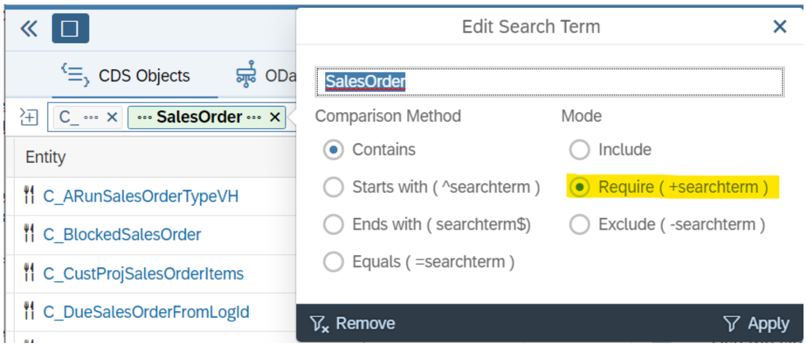
Edit Search Term

C_

Comparison Method
○ Contains
● Starts with (^searchterm)
○ Ends with (searchterm$)
○ Equals (=searchterm)

Mode
● Include
○ Require (+searchterm)
○ Exclude (-searchterm)

As second search term, we would then enter SalesOrder, using the default contains comparison method. But because we want to find objects that both start with C_ AND contain SalesOrder, we need to choose the “require” mode:



Entity
- C_ARunSalesOrderTypeVH
- C_BlockedSalesOrder
- C_CustProjSalesOrderItems
- C_DueSalesOrderFromLogId

Edit Search Term

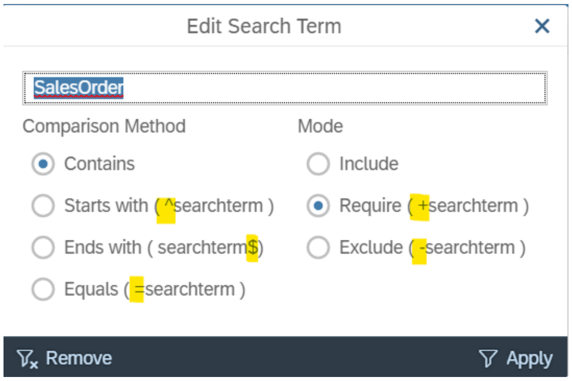
SalesOrder

Comparison Method	Mode
● Contains	○ Include
○ Starts with (^searchterm)	● Require (+searchterm)
○ Ends with (searchterm$)	○ Exclude (-searchterm)
○ Equals (=searchterm)	

You might notice that the Search Token for SalesOrder is rendered in light green hue. This indicates that this token uses the "require" mode.

This combination of search terms will now return only objects that start with C_ and contain the search term SalesOrder.

2.5.1.3.7 Keyboard Shortcuts



Edit Search Term

SalesOrder

Comparison Method	Mode
● Contains	○ Include
○ Starts with (^searchterm)	● Require (+searchterm)
○ Ends with (searchterm$)	○ Exclude (-searchterm)
○ Equals (=searchterm)	

Smart Search was designed to fully support editing with only the keyboard. Once you enter a search term, the search field remains focused, so you can immediately enter another search term.

Using the arrow keys, you can navigate the search tokens, and select a particular one. The Edit Search Term popup dialog of the selected search token may be opened by pressing the spacebar. You can also remove the selected search token using only the keyboard by pressing the Backspace key.

In addition, both the Comparison method and the Search Mode can be specified directly as you enter a search term by using special operator symbols. You may have noticed this by looking at the Edit Search Term popup:

For example:

- to specify objects starting with prefix C_, use the circumflex, like so: ^C_
- to specify objects ending with postfix _Test, use the dollar sign, like so: _Test$
- to specify objects must match SalesOrder, use the plus sign, like so: +salesorder
- to specify objects must not match item, use the minus sign (dash), like so: -item

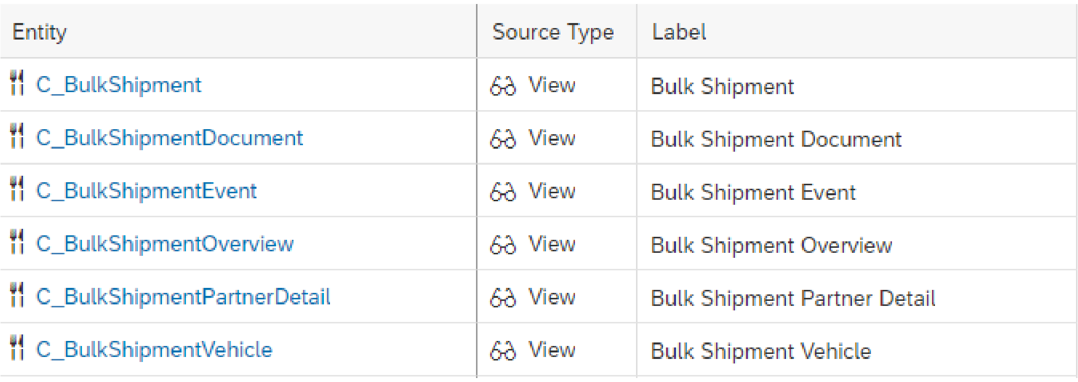
2.5.1.3.8 SAP-style "asterisk" Wildcards

As bonus, S4Explorer also supports SAP-style "asterisk" wildcards.

For example, upon entering C_*SalesOrder, two search tokens are created, one to require the results match the prefix C_, and another to require that results match the postfix SalesOrder. This makes it even quicker to enter multiple search tokens!

2.5.2 Catalog Results Table

Below the search form S4Explorer will provide a tabular overview of all of the entities that match the input search criteria. The user can compare, analyze and open entities by clicking on their entity name:



Entity	Source Type	Label
C_BulkShipment	View	Bulk Shipment
C_BulkShipmentDocument	View	Bulk Shipment Document
C_BulkShipmentEvent	View	Bulk Shipment Event
C_BulkShipmentOverview	View	Bulk Shipment Overview
C_BulkShipmentPartnerDetail	View	Bulk Shipment Partner Detail
C_BulkShipmentVehicle	View	Bulk Shipment Vehicle

The table can be customized using the **Table Settings** button to open a dialog window. Here, the user can select and / or deselect any fields that should be displayed in the results table.

2.5.3 Object Detail Page

Object detail pages display information about one particular object found in a particular Object Catalog. They appear in the **Error! Reference source not found.** as tabpages.

Object Detail Pages can be opened directly by clicking Links that appear the object Catalog Results Table. In addition, Object pages themselves may contain other tables of related objects as well as Diagrams, which will also typically have a link which allow you to open the Object page for the related object.

2.6 Dialogs

2.6.1 Settings Dialog

The settings dialog provides access to various settings and defaults that determine S4Explorer's behavior. You can open the settings dialog by clicking the Settings Dialog Button from the **Error! Reference source not found.**.

2.6.2 About Dialog

The about dialog provides general information about the current S4Explorer version. You can open the about dialog by clicking the from the **Error! Reference source not found.**.

2.7 Diagrams

S4Explorer appear in several places in Object Detail Pages to illustrate a particular aspect of the current object.

For CDS objects, the following diagrams are available:

- CDS Structure Diagram
- Dependency Tree Diagram
- Column Lineage Diagram

For OData services the following diagram is available:

- The OData structure diagram

2.7.1 General Diagram User Interface

The different types of diagrams in S4Explorer are designed to visualize different aspects to provide insights appropriate for different use cases. Regardless, the user interface to work with diagrams is to a large extent generic, and independent of a particular type of diagram.

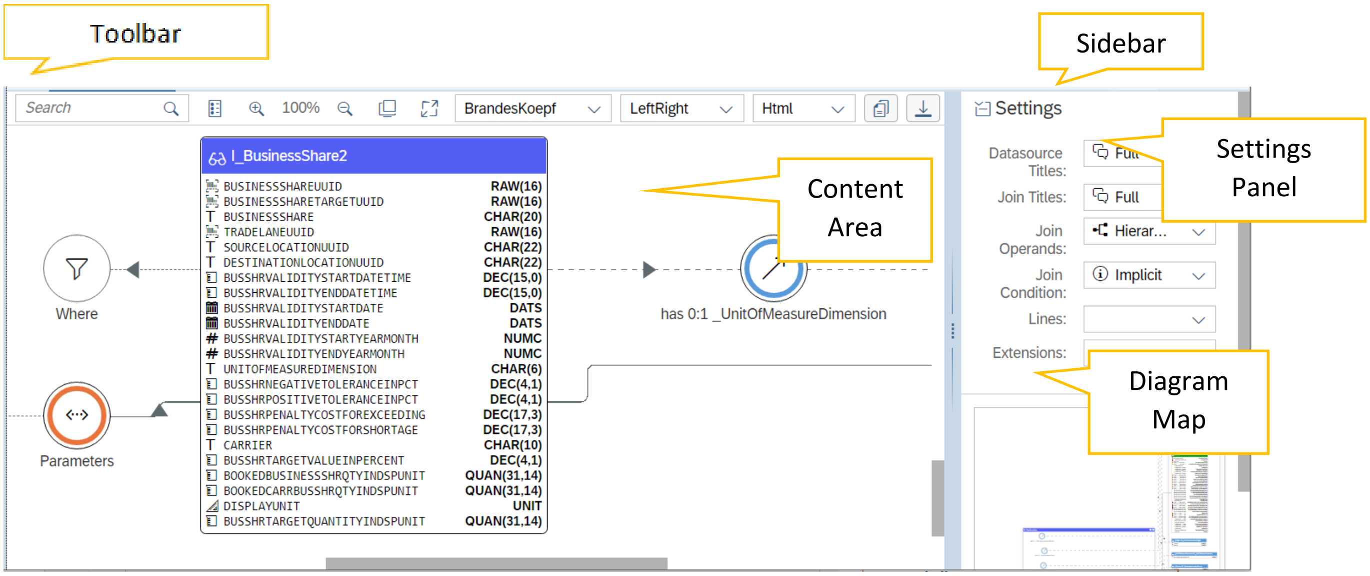
Diagrams feature only three different kinds of objects:

- Nodes. These are elements on a diagram that are used to represent a kind of thing. The types of things that are represented on diagrams depend on the kind of diagram, and is indicated by the particular shape and color. Nodes are rendered either as rectangles or circles. Nodes usually have a title, an Icon, and may have a collection attributes. Nodes may also have a number of associated actions to reveal more detailed information about the node, and/or to offer functionality to drill down onto the node.
- Lines/Connections. Nodes may be connected by lines. Lines represent some sort of relationship between the things that are represented by the nodes on either end of the line. Lines have a direction, and a style (dashed, solid etc), and may have additional attributes that define the details of the relationship
- Groups. Nodes may be organized in groups. Groups appear as rectangular areas that contain one or more nodes.

The common UI elements in a S4Explorer diagram are:

- Diagram Toolbar
- Diagram Content Area
- Diagram Sidebar
- Diagram Settings Panel
- Diagram Map

The following screenshot is of a S4Explorer Structrure diagram, illustrating the common user interface elements:



I_BusinessShare2

BUSINESSSHAREUUID	RAW(16)
BUSINESSSHARETARGETUUID	RAW(16)
BUSINESSSHARE	CHAR(20)
TRADELANEUUID	RAW(16)
SOURCELOCATIONUUID	CHAR(22)
DESTINATIONLOCATIONUUID	CHAR(22)
BUSSHRVALIDITYSTARTDATETIME	DEC(15,0)
BUSSHRVALIDITYENDDATETIME	DEC(15,0)
BUSSHRVALIDITYSTARTDATE	DATS
BUSSHRVALIDITYENDDATE	DATS
BUSSHRVALIDITYSTARTYEARMONTH	NUMC
BUSSHRVALIDITYENDYEARMONTH	NUMC
UNITOFMEASUREDIMENSION	CHAR(6)
BUSSHRNEGATIVETOLERANCEINPCT	DEC(4,1)
BUSSHRPOSITIVETOLERANCEINPCT	DEC(4,1)
BUSSHRPENALTYCOSTFOREXCEEDING	DEC(17,3)
BUSSHRPENALTYCOSTFORSHORTAGE	DEC(17,3)
CARRIER	CHAR(10)
BUSSHRTARGETVALUEINPERCENT	DEC(4,1)
BOOKEDBUSINESSSHRQTYINDSPUNIT	QUAN(31,14)
BOOKEDCARRBUSSHRQTYINDSPUNIT	QUAN(31,14)
DISPLAYUNIT	UNIT
BUSSHRTARGETQUANTITYINDSPUNIT	QUAN(31,14)

Datasource Titles: Full
Join Titles: Full
Join Operands: Hierar...
Join Condition: Implicit
Lines:
Extensions:

2.7.1.1 Diagram Toolbar

The Toolbar provides a number of controls to work with the diagram

- SearchField. The searchfield lets you search for a particular element appearing on the diagram.
- Legend Button. With this button you can toggle the visibility of the Diagram Legend.
- Zoom controls. Using these controls you can z oom in or zoom out the content area of the diagram. Available controls are:
  - Zoom in
  - Zoom out
  - Zoom to fit
  - Enter full screen
- Node Placement Selector. With this selector you can control the layouting algorithm that decides where nodes are placed and lines are routed. This option is provided to allow you to experiment and choose the most optimal rendition of your diagram. What is most optimal is to some extent a matter of preference and may also depend on the complexity and size of the diagram.
  The following options are available:
  - BrandesKroepf: Minimizes the number of edge bends at the expense of the graph size
  - LinearSegments: Calculates the most optimal layout balance
  - Simple: Minimizes the area taken by the graph at the expense of everything else.
- Orientation Selector. With this selector you can control to direction of the layout. Like node placement, this lets you customize the diagram layout and choose whatever option is most appropriate for your use case. The available options are:
  - BottomTop: The diagram “starts at” the bottom of the content area, and grows upward
  - LeftRight: The diagram “starts at” the left side of the content area and flows to the right, like a data flow diagram
  - RightLeft: The diagram starts at the right side of the content area and flow to the left
  - TopBottom: The diagram starts at the top of the content area, and flows downward, like an organization chart.

- Rendering Selector: This lets you choose the technique to draw the diagram. The options are:
    - Html: the diagram is rendered using only Hypertext Markup Language. This is appropriate in most cases, and especially useful if you want to export your diagram for use on a static local intranet or website, as the diagram content will be readily usable in this format.
    - Svg: the diagram is rendered using Scaleable Vector Graphics. This is appropriate when you want to export the diagram as an image that you want to share in office documents, or if you want to prepare a version that is suitable for printing. Vector graphics can be scaled up and down to whatever size is appropriate without loss of quality
- Copy Button: The copy button lets you make a quick copy of the diagram so you can easily exchange it with other applications (like MS Teams chats, emails, and so on). The exact behavior depends on the current selection in the Rendering Selector:
    - Html: the HTML markup is copied to the clipboard as text. This option is suitable for copying the diagram to a webpage
    - Svg: the diagram is copied to the clipboard as PNG image data. This lets you paste the diagram as a static image
- Export Button: The Export button is currently only available on the Column Lineage Diagram. This option exports the low level graph data used to render the diagram to an excel workbook as structured data.
- Download Button: This lets you download the diagram to a file on your local computer which you then may share as needed. The exact results will depend on the Chosen rendering mode:
    - Html: this will download the diagram as a standalone, static HTML file. This is ideal for including S4Explorer diagrams on static websites or a knowledge base.
    - Svg: the diagram will be downloaded as a standalone SVG image file. This is ideal for including S4Explorer diagrams in other documents, like office documents, documentation systems, or also to include the image on a web page. SVG images can also be used to high-quality printouts or large format plotting

2.7.1.2 Diagram Content Area

The Diagram Content Area is the diagram proper – the canvas where the nodes, lines and groups are layed out and visualized.

2.7.1.3 Diagram Sidebar

The Sidebar appears on the right side of the Diagram, and Is separated from it by a splitter bar, which may be dragged to resize it (and consequently resize the diagram content area).

The sidebar contains the Diagram Settings Panel and the Diagram Map

2.7.1.4 Diagram Settings Panel

The Diagram Setting panel sits In the top of the Diagram Sidebar. Its contents depend on the type of diagram. These are usually controls that let you control some particular aspect of how the Diagram content is rendered.

For example, in the column lineage diagram you'll find a control that lets you choose the level of detail that will be shown, and in the CDS Dependencies Tree Diagram you'll find controls that let you control the grouping of the entities.

2.7.1.5 Diagram Map

The diagram map is located in the bottom of the Diagram sidebar.

It appears as a mini-version of the diagram. It offers an overview of the entire diagram, and also offers interactivity that lets you use the map to quickly navigate to the corresponding area in the actual Diagram content area.

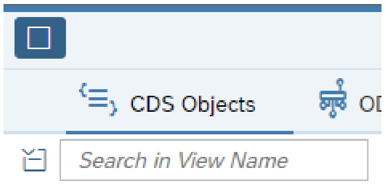
For example, if you click an item in the Diagram Map, the Content Area of the diagram will scroll automaticaly to that particular element in the diagram and focus on it.

3 Working with CDS Objects

One of S4Explorer’s key functionalities is to enable you to analyze, understand and work with CDS objects more efficiently than ever before. The following chapter will elaborate on the CDS objects catalog, the details that can be analyzed per CDS view and the features and / or options available in each of the functionalities while working with CDS objects.

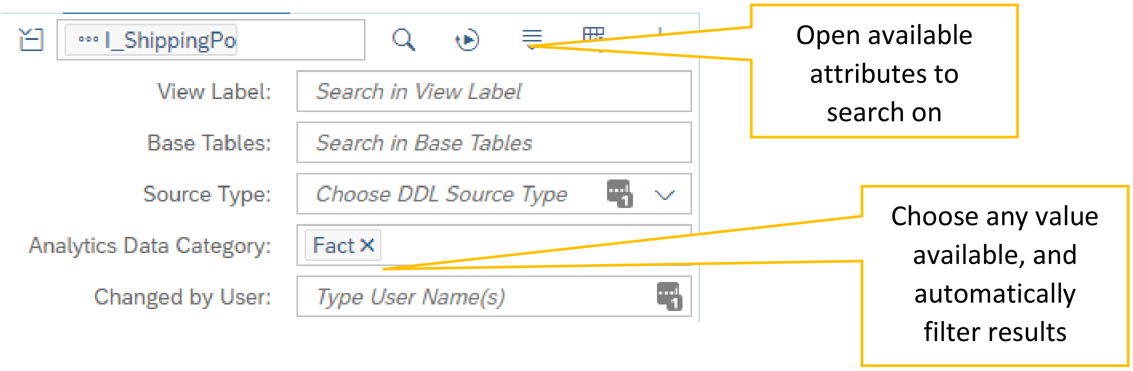
3.1 The CDS Objects Catalog

The CDS Objects Catalog is the collection of CDS views available in your SAP S4HANA build, you can use this catalog to work with S/4 HANA CDS objects such as views, tables, functions etc. The sidebar can be used for the CDS Objects Catalog by selecting CDS Objects in the top-left corner.



3.1.1 CDS Catalog Field Reference

The CDS Catalog Field Reference is a collection of attributes available in the CDS catalog. Many of these attributes can be used to find and identify the correct CDS view quickly. For example, by searching on the Analytics Data Category ‘Fact’ we can quickly identify all CDS views that are categorized as fact.



I_ShippingPo

View Label: Search in View Label

Base Tables: Search in Base Tables

Source Type: Choose DDL Source Type

Analytics Data Category: Fact ✕

Changed by User: Type User Name(s)

Open available attributes to search on

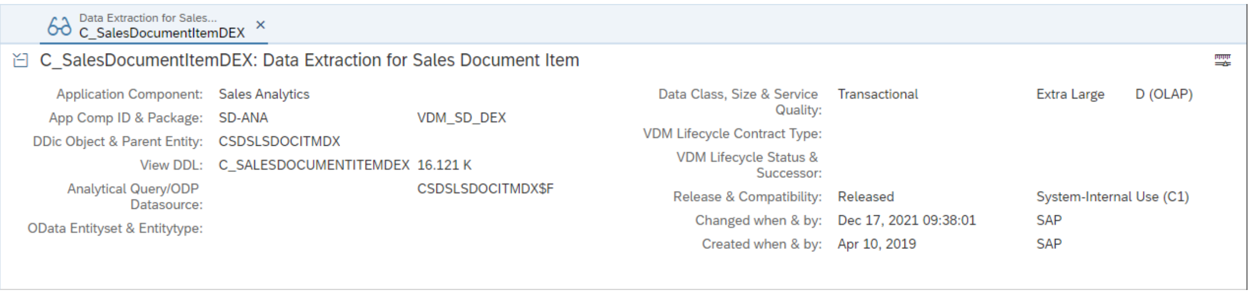
Choose any value available, and automatically filter results

3.2 CDS Object Detail Page

Selecting a CDS object from the CDS object catalog will opens its object detail page in the work area. The following sections will elaborate on the available functionalities S4Explorer offers to analyze and understand the CDS object.

3.2.1 CDS Object Detail Page Header

The page header provides an overview of a pre-defined set of key view characteristic. The page header can be minimized / maximized by clicking on the button left to the CDS view name. Below you can find an example of the detail page header for the C_SalesDocumentItemDEX CDS view:



C_SalesDocumentItemDEX: Data Extraction for Sales Document Item

Application Component: Sales Analytics
App Comp ID & Package: SD-ANA VDM_SD_DEX
DDic Object & Parent Entity: CSDSLSDOCITMDX
View DDL: C_SALESDOCUMENTITEMDEX 16.121 K
Analytical Query/ODP Datasource: CSDSLSDOCITMDX$F
OData Entityset & Entitytype:

Data Class, Size & Service Quality: Transactional Extra Large D (OLAP)
VDM Lifecycle Contract Type:
VDM Lifecycle Status & Successor:
Release & Compatibility: Released System-Internal Use (C1)
Changed when & by: Dec 17, 2021 09:38:01 SAP
Created when & by: Apr 10, 2019 SAP

The following fields can be found in the page header:

- The **Application Component** refers to the modelling construct to which the CDS Object is linked to.
- The **App Comp ID** refers to the previously mentioned application component abbreviation.
- The **Package** is the name of the package to which the object is assigned.
- The **DDic Object** defines the data dictionary object that backs the CDS view. This identifier is identical to the value specified for the `AbapCatalog.sqlViewName` annotation.
- **Parent Entity** If this CDS object is an extension view, then the Parent Entity is the name of the CDS view of which it is an extension.
- The **View DDL** field lists the name of the data definition (DDL) source object. The actual DDL source is shown in the CDS Source Code detail tab. In addition, the size of the DDL is indicated.
- The **Analytical Query** provides the analytical query name which can be used in reporting and analysis.
- The **ODP DataSource** provides the ODP data source which can be used for extracting data through the ODP framework. It consists of the name followed by a suffix, which defines the type of ODP data source:

Suffix	Explanation
$F	Transactional data / facts
$P	Master data / attributes
$Q	Time-dependent data / attributes
$T	Texts
$H	Hierarchies

- The **OData Entity set** refers to a collection of entity types.
- The **Entity type** refer to the sets and types of structured records that consist of names and typed properties. For example: customer, employee etc.
- The **Data Class** refers to the definitions of the CDS views for these categories. The data class can be:
    - Master data
    - Transaction data
    - Organizational data.

- The **Size** is measured in t-shirt sizes and refers to the SAP S4HANA sizing.
- The **Service Quality** defines the type of view, as well as the type of processing (OLAP, OLTP):

Indicator		Explanation
F		Fact tables
D		Dimension tables
S		SID tables
X OR Y		Attribute tables

- The **VDM Lifecycle Contract Type** describes the intended publication or usage of the view. This can be

Value	Explanation
Empty	Not subjected to any contract, default value for private and consumption views.
PUBLIC_LOCAL_API	Local system API contract, default for basic and composite views.
SAP_INTERNAL_API	Marks the view for SAP internal use only, default for transactional views.
PUBLIC_REMOTE_API	Annotated CDS view is used to define a remote API.

- The **VDM Lifecycle Status** can indicate if a CDS view is deprecated.
- The **VDM Successor** If a CDS view is deprecated this field will indicate the successor *(If available).*
- The **Release Status** indicates the release status of the CDS view.
- The **Compatibility Status** indicates the definition between C1 *(use system internally)* and C2 *(Use as a remote API)*.
- The **Changed when & by** provides an overview of the latest activated change date time, as well as the user responsible for these changes.
- The **Created when & by** provides an overview of the creation date time, as well as the user responsible for the creation of the view.

3.2.2 CDS Structure Diagram

CDS Structure diagrams present a graphical schematic representation of the CDS source code. The visual representation makes it easy to see the flow of data through CDS views, and help you to quickly grasp key features of a view, like its data foundation, data sources, joins, its associations, its filters and so on.

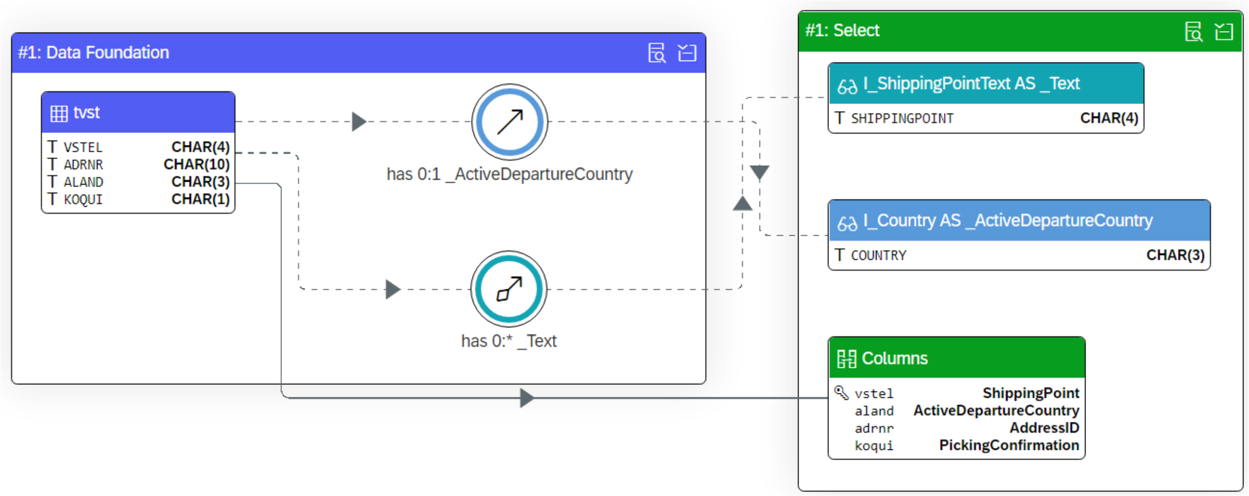
CDS Structure diagrams capture an amazing amount of information about a CDS object and present it in a compact way. They are a great tool to quickly understand the essence of the functionality of the CDS object. Structure Diagrams also allow you to quickly identify aspects of further interest, which can then be investigated further on a more detailed (code) level. CDS Structure diagrams also support visual exploration, by letting you navigate or drill down to related CDS objects.

By way of example, let's consider the CDS view `I_ShippingPoint`. Below is the (simplified) CDS code that defines the view:

```
01  define view I_ShippingPoint as
02
03  select from tvst
04
05  association [0..1] to I_Country as _ActiveDepartureCountry
06  on $projection.ActiveDepartureCountry = _ActiveDepartureCountry.Country
07
08  association [0..*] to I_ShippingPointText as _Text
09  on $projection.ShippingPoint = _Text.ShippingPoint
10
11  {
12    key vstel   as ShippingPoint,
13        aland   as ActiveDepartureCountry,
14        adrnr   as AddressID,
15        koqui   as PickingConfirmation,
16        _Text,
17        _ActiveDepartureCountry
18  }
```

Note that this view is built on a single select statement, spanning lines `03..18`.(A CDS view may have multiple select-clauses, each connected to the next through a `union` operator.)

Below is what this view might look like in the S4Explorer Structure Diagram:



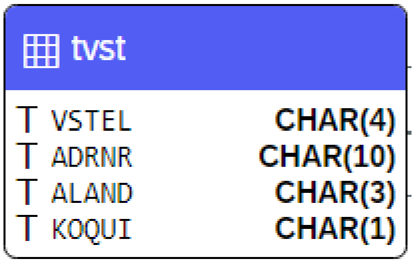
3.2.2.1 The Data Foundation

The Data foundation is the part of the select statement that lists the various data sources (tables, views, and table functions) upon which the view is defined, and how these are combined using joins and associations.

In the CDS code example, the data foundation runs from lines `03..09`.

In the Structure Diagram, the data foundation is visible as a group with a bright blue title labeled “#1 Data Foundation”.

Inside the data foundation, we find a `from` clause (line `03`) that lists the first data source, in this case the base table `tvst`. In the Structure diagram, this datasource is visualized as a rectangular node, which has the same bright blue header color as the data foundation group:



tvst

T	VSTEL	CHAR(4)
T	ADRNR	CHAR(10)
T	ALAND	CHAR(3)
T	KOQUI	CHAR(1)

In the title of the node, the icon indicates that this data source is a base table.

Inside the node we see the list of columns from this table that are used somewhere in the view definition. Note that the `tvst` table has many more columns than the ones that are shown here, but the Structure Diagram only shows those columns that are actually used by the current view.

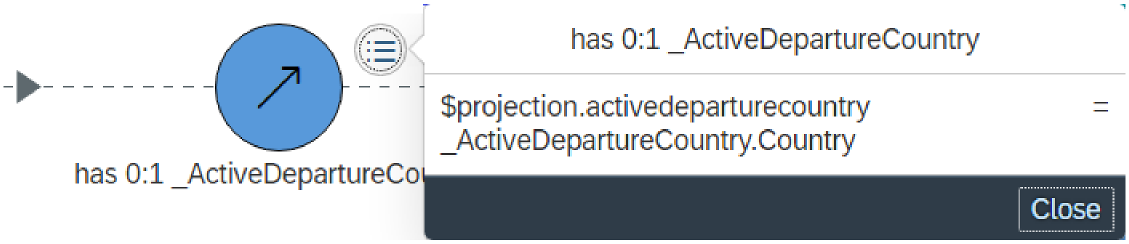
Each column line begins with an Icon that indicates the data type of the column, followed by the column name, followed by its data type.

3.2.2.2 Associations

Apart from the `tvst` table, the data foundation of the `I_ShippingPoint` view defines two associations (lines `05..09`).

The first one appears on lines `05..06` and is a `0:1` (zero to one) association to `I_Country`, meaning that for each row that comes out of the view, there may optionally exist at most 1 related row in the view `I_Country`.

In the structure diagram, this association appears as a blue-ish circular node containing an diagonal arrow pointing upward:



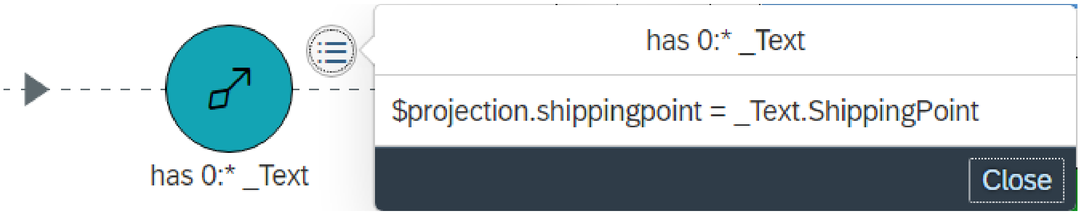
```
$projection.activedeparturecountry =
_ActiveDepartureCountry.Country
```

Note that there is a dashed line connecting the data source to the association node, and another dashed line from the association node to the node that represents the associated data source. The lines are dashed because the minimum cardinality is defined as 0 (optional)

If you select the node, a details button appears. When clicking the button, a popup appears that reveals the condition that binds the associated `I_Country` view to the `tvst` table.

The second association appears on lines `08..09`. This is a `0:*` (zero to many) association to the text table `I_ShippingpointText`, meaning that for a single row from the view there could be multiple corresponding rows in the associated table.

You might notice this association has a greenish color and icon depicting diagonal arrow with a diamond at the bottom end. Both the color and the icon indicate a “to many” relationship, making it easy to distinguish them from any “to one” associations.



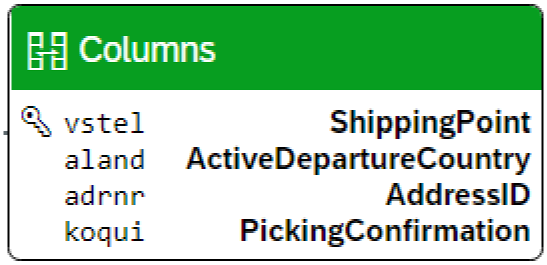
```
$projection.shippingpoint = _Text.ShippingPoint
```

3.2.2.3 Select-list

The Select list is the part of the select statement that defines the elements that are exposed and available when the view is used. In the code sample, this appears on lines `11..18`.

In the structure diagram, the Select list appears as a group with a green title.

Inside the group, the column expressions that this view produces are listed in a rectangular node with the title Columns, which lists each separate scalar expression produced but the select statement:

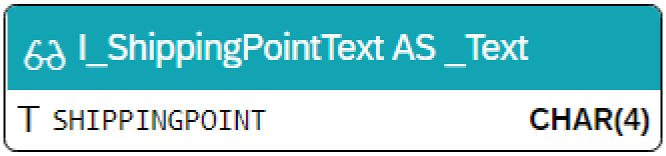


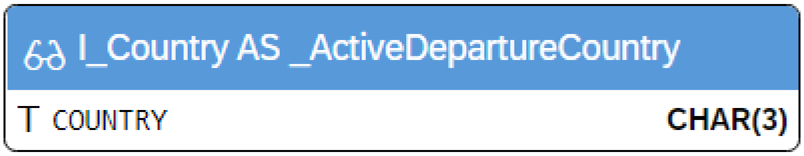
Columns	
vstel	ShippingPoint
aland	ActiveDepartureCountry
adrnr	AddressID
koqui	PickingConfirmation

Each line inside the node represents a scalar column expression, which is made up of the expression that defines the column, followed by the alias that is used to name the expression. Each of these are available as output columns of the view that is being defined.

Expressions in the select list may have an icon indicating some special feature of that column expression. In the example, we see a key icon for the expression `vstel`, which is exposed as output column `ShippingPoint`. That's because this expression was defined as key column for this view.

Apart from the Columns node, we also see two nodes corresponding to the associations that were defined in the data foundation. These indicate the fact that these associations are not only defined but also exposed by the view. This corresponds to lines `16` and `17` in the code of the sample view:





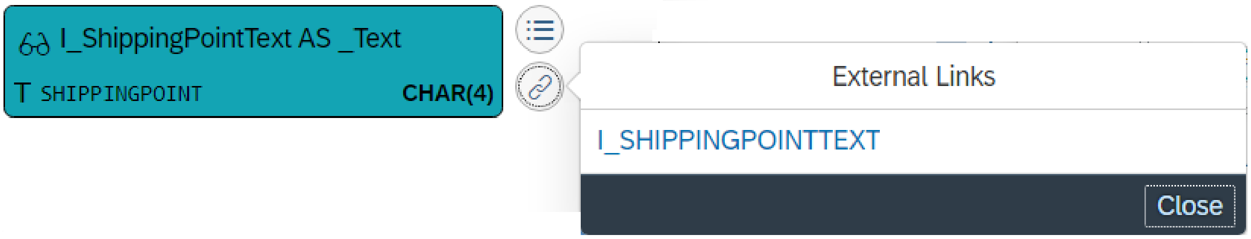
Because these associations are exposed in this way, it means that someone that uses the view may access any columns available in the associated views.

(A CDS view may define an association and not expose it in its entirety in the select list. If that is the case, then these associations would not appear in the group that represents the select list.)

3.2.2.4 Visual exploration

Any node that represents a CDS object that appears in the structure diagram may be used for visual exploration. Simply select the respective node so its action buttons are revealed. Then click the link button to reveal a list of links.

In the example below, the `I_ShippingPointText` node was selected, revealing the links menu. Clicking the link would open a new object detail page for the respective view.



3.2.2.5 Parameters

CDS objects like views may have parameters and these are visualized in their own distinct manner in Structure Diagrams. Consider the CDS view `P_APAGINGANALYSIS`. The first couple of lines this view's CDS code are shown below:

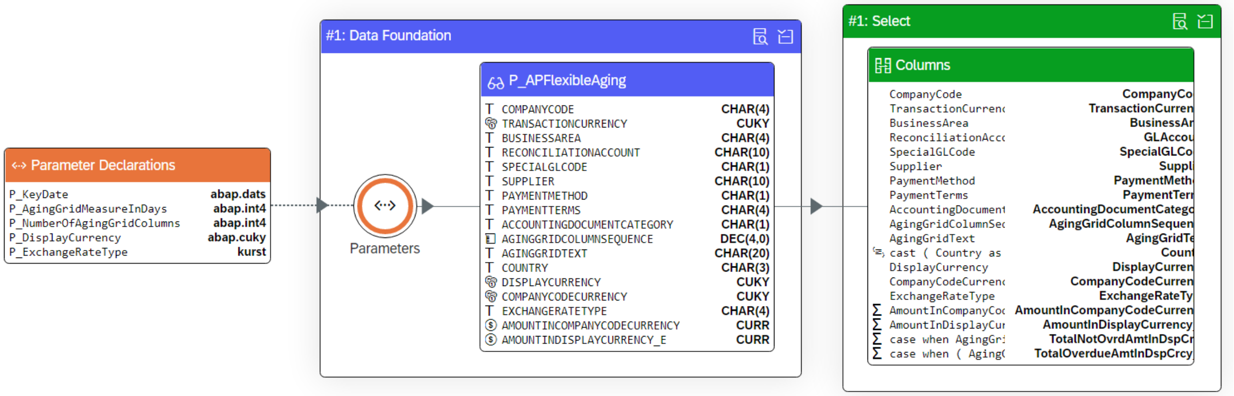
```
define view P_APAGINGANALYSIS
```

```
02  with parameters
03      P_KeyDate : abap.dats ,
04      P_AgingGridMeasureInDays : abap.int4,
05      P_NumberOfAgingGridColumns : abap.int4,
06      P_DisplayCurrency : abap.cuky,
07      P_ExchangeRateType : kurst
08  as select
09  from P_APFlexibleAging(
10      P_KeyDate : :P_KeyDate,
11      P_AgingGridMeasureInDays : :P_AgingGridMeasureInDays,
12      P_NumberOfAgingGridColumns : :P_NumberOfAgingGridColumns,
13      P_DisplayCurrency : :P_DisplayCurrency,
14      P_ExchangeRateType: :P_ExchangeRateType
15  )
16  // rest of the view definition omitted
```

In line `02`, the keywords with parameters initiate the declaration of the parameter list. Lines `03..07` each declare a distinct parameter.

Line `08` starts the select statement, and on lines `09..15` we see a `from`-clause on the underlying view `P_APFlexibleAging`. The `P_APFlexibleAging` view itself also declares a parameter list, and each of these parameters need to be passed a value. This happens on lines `10..14`. In this case, the parameter values of the parameters defined by the current `P_APAGINGANALYSIS` view are all passed to a corresponding parameter declared by the `P_APFlexibleAging` view used in the `from`-clause.

This is what the S4Explorer Structure diagram looks like:

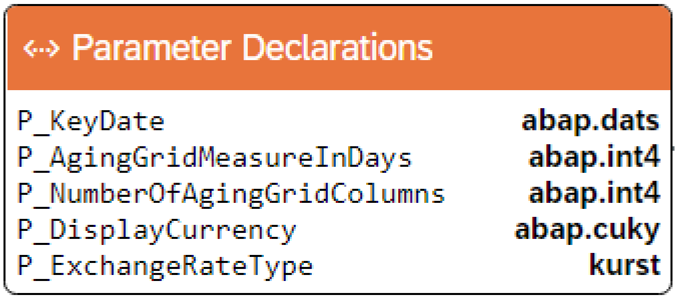


P_KeyDate	abap.dats
P_AgingGridMeasureInDays	abap.int4
P_NumberOfAgingGridColumns	abap.int4
P_DisplayCurrency	abap.cuky
P_ExchangeRateType	kurst

COMPANYCODE	CHAR(4)
TRANSACTIONCURRENCY	CUKY
BUSINESSAREA	CHAR(4)
RECONCILIATIONACCOUNT	CHAR(10)
SPECIALGLCODE	CHAR(1)
SUPPLIER	CHAR(10)
PAYMENTMETHOD	CHAR(1)
PAYMENTTERMS	CHAR(4)
ACCOUNTINGDOCUMENTCATEGORY	CHAR(1)
AGINGGRIDCOLUMNSEQUENCE	DEC(4,0)
AGINGGRIDTEXT	CHAR(20)
COUNTRY	CHAR(3)
DISPLAYCURRENCY	CUKY
COMPANYCODECURRENCY	CUKY
EXCHANGERATETYPE	CHAR(4)
AMOUNTINCOMPANYCODECURRENCY	CURR
AMOUNTINDISPLAYCURRENCY_E	CURR

CompanyCode	CompanyCo
TransactionCurren	TransactionCurren
BusinessArea	BusinessAr
ReconciliationAcc	GLAccou
SpecialGLCode	SpecialGLCo
Supplier	Suppl
PaymentMethod	PaymentMeth
PaymentTerms	PaymentTerı
AccountingDocument	AccountingDocumentCatego
AgingGridColumnSe	AgingGridColumnSequen
AgingGridText	AgingGridTe
cast (Country as	Count
DisplayCurrency	DisplayCurren
CompanyCodeCurren	CompanyCodeCurren
ExchangeRateType	ExchangeRateTy
Σ AmountInCompanyCo	AmountInCompanyCodeCurren
Σ AmountInDisplayCur	AmountInDisplayCurrency
Σ case when AgingGr	TotalNotOvrdAmtInDspCr
Σ case when (Aging	TotalOverdueAmtInDspCrcy

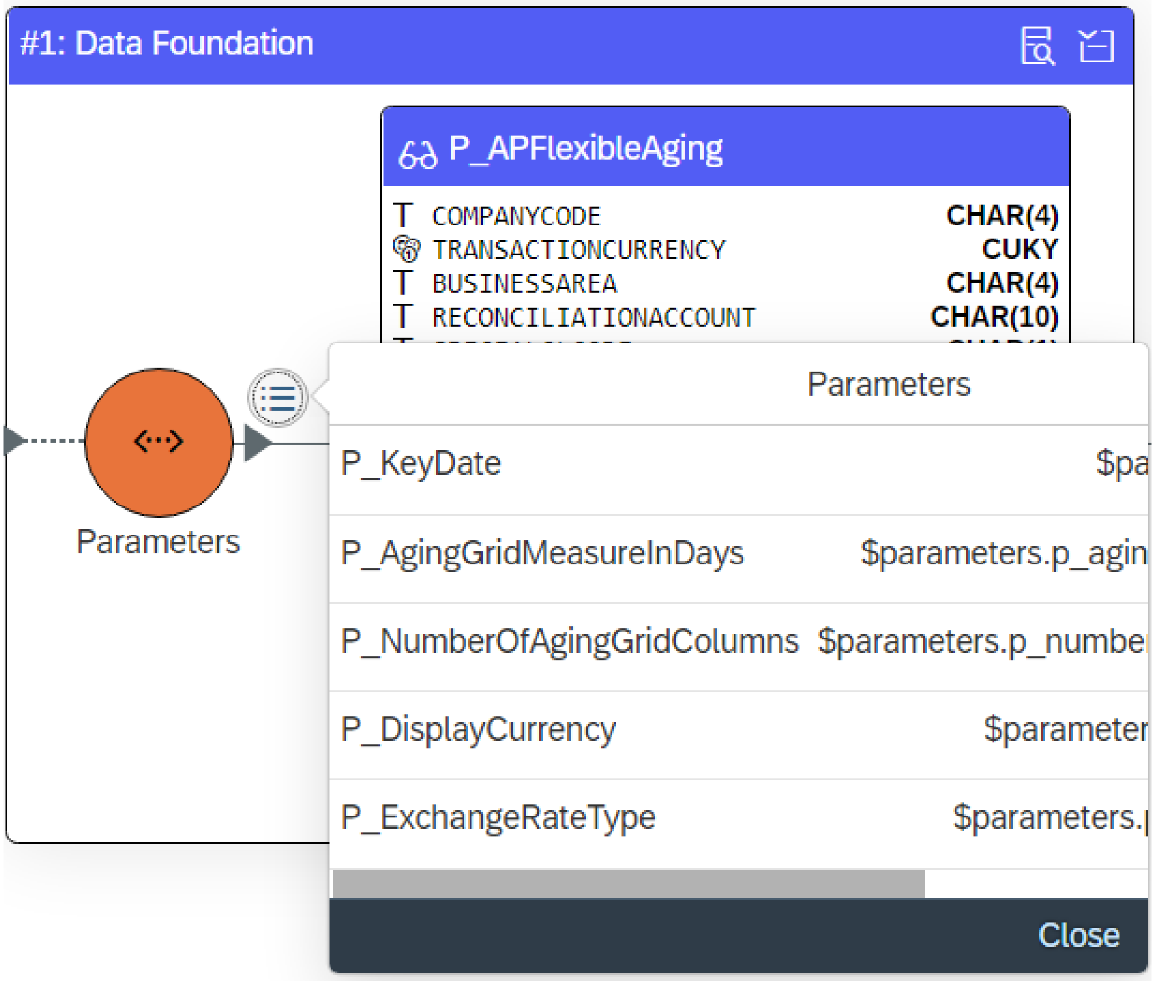
The parameter declarations are rendered as separate rectangular node having an orange header labeled “Parameter Declarations”. The attributes inside the Parameter Declarations node each represent the declaration of a single parameter, listing both its name and its declarared datatype:



P_KeyDate	abap.dats
P_AgingGridMeasureInDays	abap.int4
P_NumberOfAgingGridColumns	abap.int4
P_DisplayCurrency	abap.cuky
P_ExchangeRateType	kurst

(Sometimes, parameters are declared with a default value, and if that's the case, this will also be visible in the Structure Diagram.)

The assignment of `P_APFlexibleAging`'s parameters appears as a circular node labeled "Parameters", which appears in between the node that represents `P_APAGINGANALYSIS`'s parameter list and the node that represents the `P_APFlexibleAging` view in the from-clause. When selecting the parameters node, the details action button appears. Clicking it will open a popup that reveals the assignment details:



T COMPANYCODE	CHAR(4)
TRANSACTIONCURRENCY	CUKY
T BUSINESSAREA	CHAR(4)
T RECONCILIATIONACCOUNT	CHAR(10)

Parameters

P_KeyDate	$pa
P_AgingGridMeasureInDays	$parameters.p_agin
P_NumberOfAgingGridColumns	$parameters.p_numbe
P_DisplayCurrency	$parameter
P_ExchangeRateType	$parameters.

Note that any parameter declared by the view can be referenced in any expression elsewhere in the view. Passing references to declared parameters on to data sources used in the view is a common use case, but by no means the only one. Checkout the section on Filter conditions to see an example of parameters being used in the `where`-clause of a CDS view.

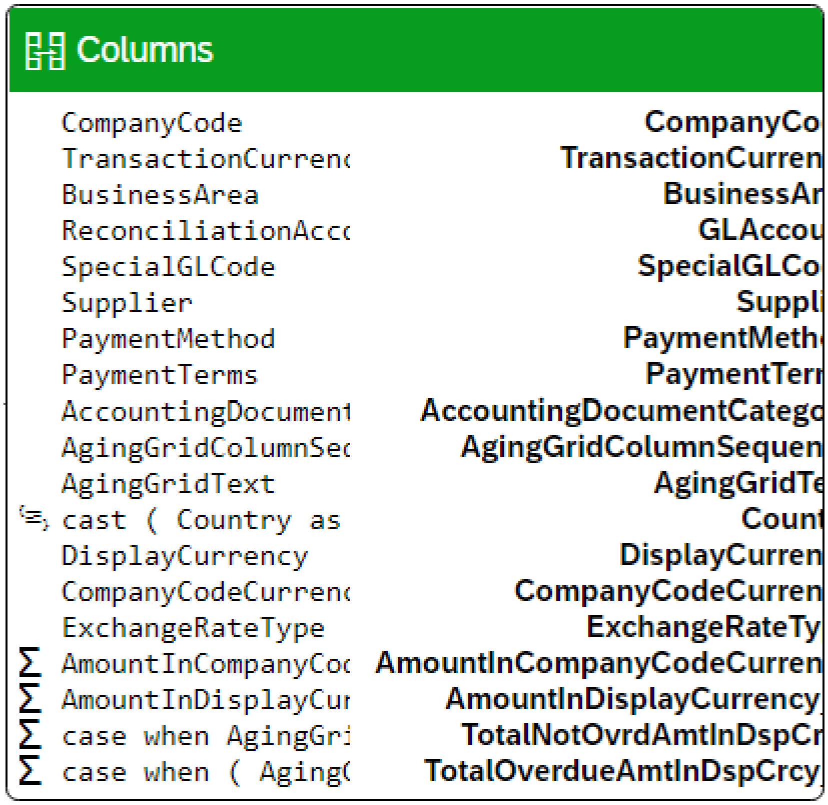
3.2.2.6 Expressions and Default Aggregations in the Select-list

In the `I_ShippingPoint` example, we mentioned that columns in the select list may have an icon to signify some special meaning. The only case in that example was the key column.

The `P_APAGINGANALYSIS` view does not declare any key columns, but that example illustrates other uses of icons for a number of columns.

For example, the declaration of the `Country` column is an expression, which is indicated by a ≡-icon.

The columns `AmountInCompanyCodeCurrency`, `AmountInDisplayCurrency_E`, `TotalNotOvrdAmtInDspCrcy` and `TotalOverdueAmtInDspCrcy_E` have a `@DefaultAggregation` annotation, indicating that on these columns, an aggregate function will be applied. In all these cases, the value of the annotation is `#SUM`, indicating summation. In the structure diagram this is indicated by the Σ-icon.



Columns

	CompanyCode	CompanyCo
	TransactionCurren	TransactionCurren
	BusinessArea	BusinessAr
	ReconciliationAcc	GLAccou
	SpecialGLCode	SpecialGLCo
	Supplier	Suppl
	PaymentMethod	PaymentMeth
	PaymentTerms	PaymentTer
	AccountingDocument	AccountingDocumentCatego
	AgingGridColumnSec	AgingGridColumnSequen
	AgingGridText	AgingGridTe
≡	cast (Country as	Count
	DisplayCurrency	DisplayCurren
	CompanyCodeCurrenc	CompanyCodeCurren
	ExchangeRateType	ExchangeRateTy
Σ	AmountInCompanyCoc	AmountInCompanyCodeCurren
Σ	AmountInDisplayCur	AmountInDisplayCurrency
Σ	case when AgingGri	TotalNotOvrdAmtInDspCr
Σ	case when (Aging(	TotalOverdueAmtInDspCrcy

Note that the columns `TotalNotOvrdAmtInDspCrcy` and `TotalOverdueAmtInDspCrcy_E` are also based on an expression. However, only one icon is shown, and in this case, the icon for the aggregation takes precedence.

To see the details for all columns, you can always select the Columns node to reveal its Details action button. Clicking it reveals a popup with the full definition of each column.

3.2.2.7 Join Operations

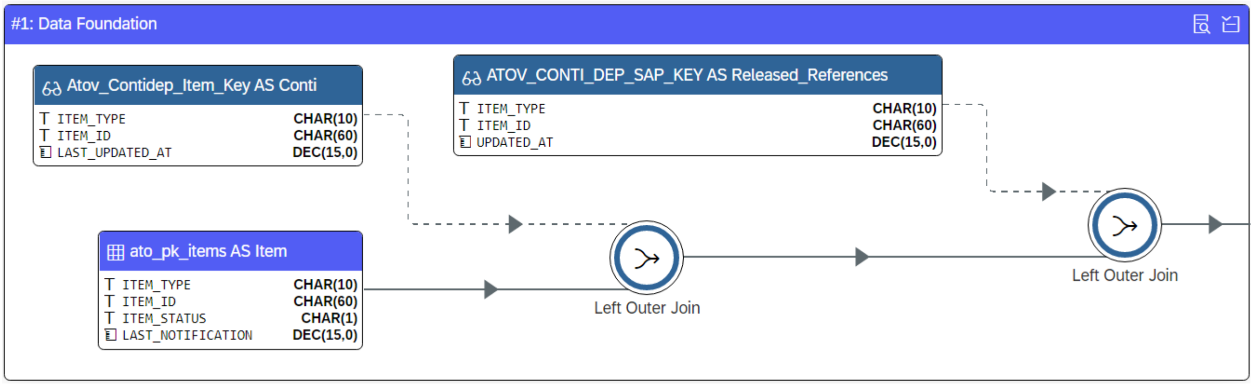
Joins are rendered as circular nodes, which are connected to its join operands (data sources) by lines. This is best illustrated with an example.

Consider the CDS view `Atov_Pk_Item_ContiDep_Join`. The CDS code for its data foundation is listed below:

```
select
from             ato_pk_items          as Item

left outer join Atov_Contidep_Item_Key as Conti
```

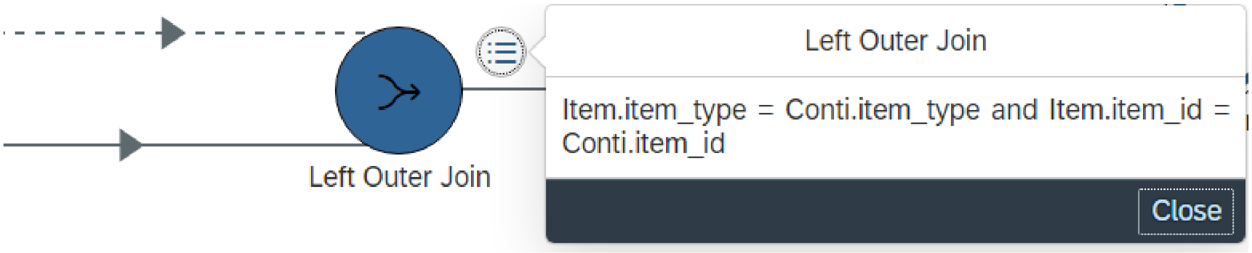
```
on  Item.item_type = Conti.item_type
and Item.item_id = Conti.item_id

left outer join ATOV_CONTI_DEP_SAP_KEY as Released_References
on  Item.item_type = Released_References.item_type
and Item.item_id   = Released_References.item_id
```

This is what the data foundation looks like in the structure diagram:



You may notice that the first data source has a bright blue header, just like the data foundation group itself. This may be regarded as “the start” of the flow of data inside the data foundation.

The join operations are represented by the circular nodes, having two incoming lines for the join operands and one outgoing line which represents the join result.



Left Outer Join

Item.item_type = Conti.item_type and Item.item_id = Conti.item_id

You can select the join node and press its details action button to inspect the join condition, which is shown in a popup.

In the example, a left outer join is used. This is reflected in the style of lines of the join operands. The data source that acts as the left operand has a solid line. The line incoming from the right join operand is dashed, indicating that corresponding rows from that table may or may not be present.

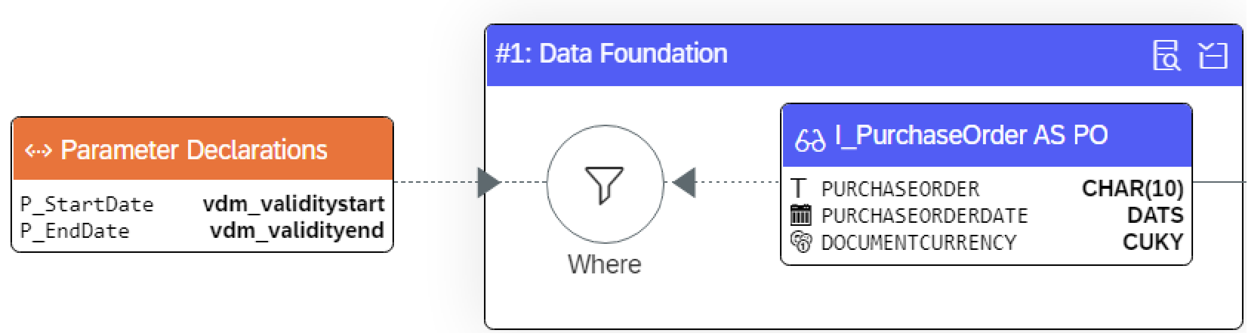
The Settings panel offers a number of controls that let you control how joins are rendered in the structure diagram. These options are discussed in the section about the Structure Diagram Settings panel.

3.2.2.8 Filter Conditions

CDS select statements my include a `where`-clause, which has the effect of filtering results based on criteria. For example, consider the `P_PurchOrderFiltered` view. Its `where`-clause is listed below:

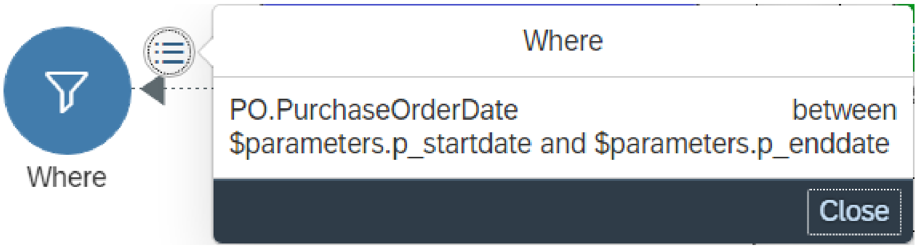
```
where PO.PurchaseOrderDate between $parameters.P_StartDate
                           and     $parameters.P_EndDate
```

This is what it looks like in the Structure Diagram:



The `where`-clause is rendered as a circular node with a filter Icon, and labeled "Where". It has lines incoming from both the Parameter Declarations node, as well as from the node representing the `I_PurchaseOrder` data source. The lines symbolize the fact that the columns form the data source and the values passed through the parameters "come together" in the where-clause, where they are compared so the parameter values can be used to restrict or filter the data coming from the data source.

You can inspect the condition by selecting the filter node, and pressing its details action button to reveal a popup containing the full condition code:



```
PO.PurchaseOrderDate between
$parameters.p_startdate and $parameters.p_enddate
```

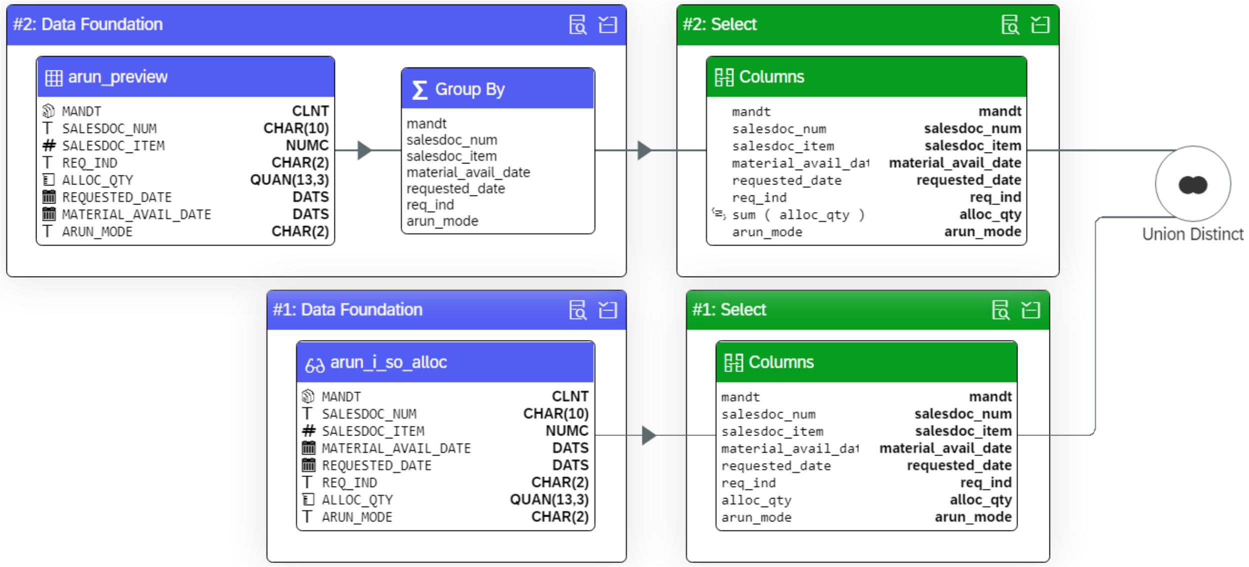
3.2.2.9 Group By-list

Select statements in a CDS view may have a `group by`-clause. If present, the group by-clause is rendered as a rectangular node inside the data foundation. The attributes that appear inside it represent the expressions listed in the group-by clause.

3.2.2.10 Union operations

So far, we've only seen examples of views containing a single select-statement. CDS views may use one or more union operations to combine results of multiple select statements.

Consider theCDS view `ARUN_UNION_ALLOC`. The following screenshot illustrates how union operations are visualized by the CDS structure diagram:



#2: Data Foundation

arun_preview

MANDT	CLNT
SALESDOC_NUM	CHAR(10)
SALESDOC_ITEM	NUMC
REQ_IND	CHAR(2)
ALLOC_QTY	QUAN(13,3)
REQUESTED_DATE	DATS
MATERIAL_AVAIL_DATE	DATS
ARUN_MODE	CHAR(2)

Group By

mandt
salesdoc_num
salesdoc_item
material_avail_date
requested_date
req_ind
arun_mode

#2: Select

Columns

mandt	mandt
salesdoc_num	salesdoc_num
salesdoc_item	salesdoc_item
material_avail_dat	material_avail_date
requested_date	requested_date
req_ind	req_ind
sum (alloc_qty)	alloc_qty
arun_mode	arun_mode

Union Distinct

#1: Data Foundation

arun_i_so_alloc

MANDT	CLNT
SALESDOC_NUM	CHAR(10)
SALESDOC_ITEM	NUMC
MATERIAL_AVAIL_DATE	DATS
REQUESTED_DATE	DATS
REQ_IND	CHAR(2)
ALLOC_QTY	QUAN(13,3)
ARUN_MODE	CHAR(2)

#1: Select

Columns

mandt	mandt
salesdoc_num	salesdoc_num
salesdoc_item	salesdoc_item
material_avail_dat	material_avail_date
requested_date	requested_date
req_ind	req_ind
alloc_qty	alloc_qty
arun_mode	arun_mode

The union operation is visible as a circular node having an icon that resembles a Venn-diagram of a mathematical union. The union operation is labeled "Union Distinct", to indicate that this union operation will one yield unique occurrences of rows from either table. In case of an explicit `union all` operation, the label will read "Union All".

3.2.2.11 View Extensions (Append Views)

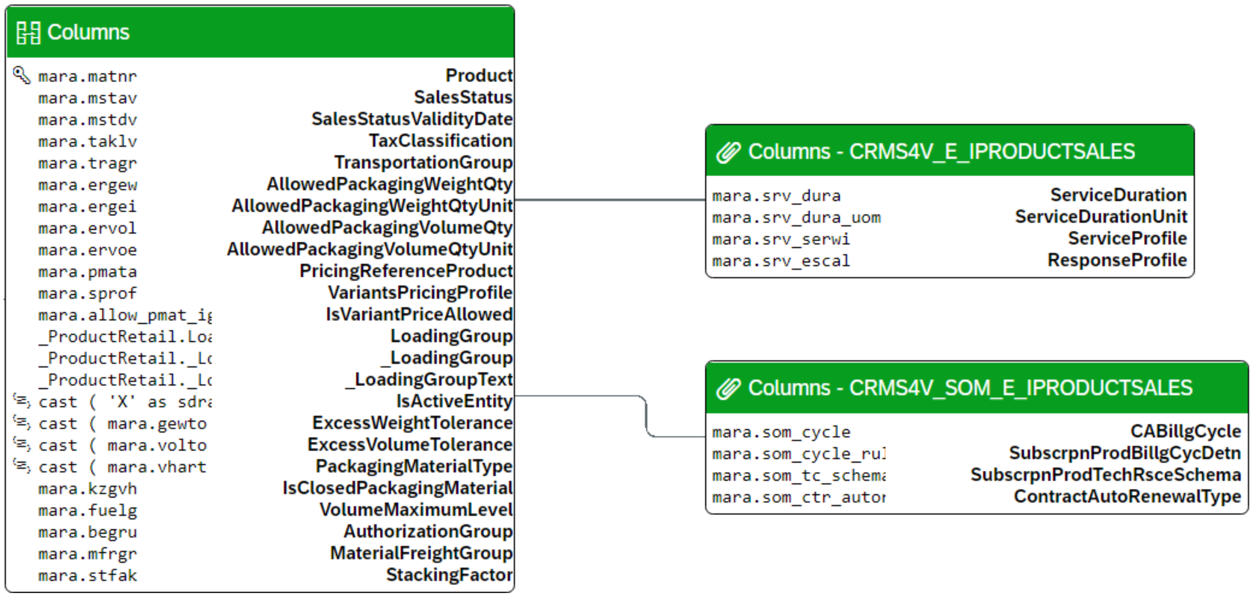
CDS Views may be enhanced by append views, also known as a view extension. Extensions typically add new columns to existing view. View Extensions may also add new associations to existing views. View Extensions are defined in their own DDL file, but do not create new, independent new entities. Rather, it's more as if the extension's DDL is included in that of the view that is extended.

Extensions are a powerful tool to manage custom developments, it can be a challenge to maintain them. In particular, it can be quite confusing to work with a view and notice columns or associations which appear not to be defined anywhere in the view's actual soure code. This is due to the fact that a view extension's DDL refers to the view it extends, but not vice versa: by looking at the CDS source code of a particular view, you simply cannot know if the view has been extended, and if so, how many extensions of the view exist.

S4Explorer has many features including the Structure diagram to facilitate working with view extensions. Consider for example the CDS View `I_ProductSales`.

3.2.2.11.1 Columns defined by View Extensions

Below is a partial screenshot of the Structure diagram of `I_ProductSales`:



Columns

mara.matnr	Product
mara.mstav	SalesStatus
mara.mstdv	SalesStatusValidityDate
mara.taklv	TaxClassification
mara.tragr	TransportationGroup
mara.ergew	AllowedPackagingWeightQty
mara.ergei	AllowedPackagingWeightQtyUnit
mara.ervol	AllowedPackagingVolumeQty
mara.ervoe	AllowedPackagingVolumeQtyUnit
mara.pmata	PricingReferenceProduct
mara.sprof	VariantsPricingProfile
mara.allow_pmat_ig	IsVariantPriceAllowed
_ProductRetail.Loa	LoadingGroup
_ProductRetail._Lc	_LoadingGroup
_ProductRetail._Lc	_LoadingGroupText
cast ('X' as sdra	IsActiveEntity
cast (mara.gewto	ExcessWeightTolerance
cast (mara.volto	ExcessVolumeTolerance
cast (mara.vhart	PackagingMaterialType
mara.kzgvh	IsClosedPackagingMaterial
mara.fuelg	VolumeMaximumLevel
mara.begru	AuthorizationGroup
mara.mfrgr	MaterialFreightGroup
mara.stfak	StackingFactor

Columns - CRMS4V_E_IPRODUCTSALES

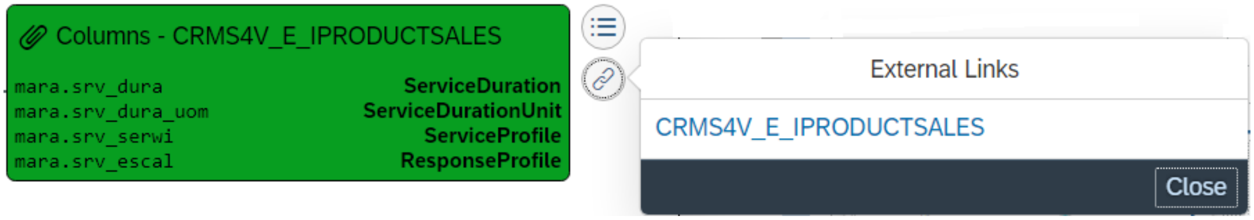
mara.srv_dura	ServiceDuration
mara.srv_dura_uom	ServiceDurationUnit
mara.srv_serwi	ServiceProfile
mara.srv_escal	ResponseProfile

Columns - CRMS4V_SOM_E_IPRODUCTSALES

mara.som_cycle	CABillgCycle
mara.som_cycle_rul	SubscrpnProdBillgCycDetn
mara.som_tc_schema	SubscrpnProdTechRsceSchema
mara.som_ctr_autor	ContractAutoRenewalType

Apart from the familiar Columns list itself, you may notice it has 2 attached nodes that look exactly like column lists. These correspond to its extensions CRMS4V_E_IPRODUCTSALES and CRMS4V_SOM_E_IPRODUCTSALES. These extension nodes have the same green color as the column list but are recognizable as extensions by their 📎-icon.

You can navigate to the actual extension by selecting the node, and then clicking the links button:



Columns - CRMS4V_E_IPRODUCTSALES

mara.srv_dura	ServiceDuration
mara.srv_dura_uom	ServiceDurationUnit
mara.srv_serwi	ServiceProfile
mara.srv_escal	ResponseProfile

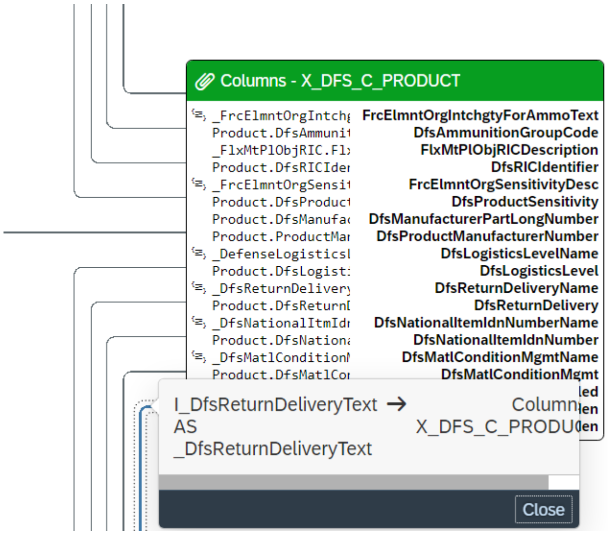
External Links

CRMS4V_E_IPRODUCTSALES

The link in the popup will open a new Object Detail page for the Extension.

3.2.2.11.2 Associations defined by View Extensions

Any associations defined by extensions are also incorporated into the Structure Diagram. These associations and the associated data sources look just like ordinary associations, however, they are connected by a line to the node representing the extension that defines the association. For example, consider the C_Product view, and examine its X_DFS_C_PRODUCT extension:

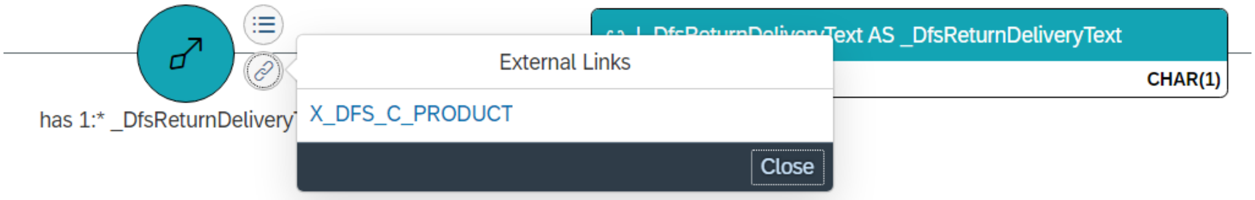


Columns - X_DFS_C_PRODUCT

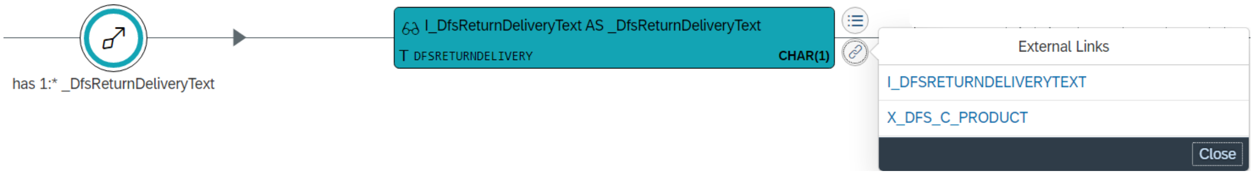
_FrcElmntOrgIntchg	FrcElmntOrgIntchgtyForAmmoText
Product.DfsAmmunit	DfsAmmunitionGroupCode
_FlxMtPlObjRIC.Flx	FlxMtPlObjRICDescription
Product.DfsRICIder	DfsRICIdentifier
_FrcElmntOrgSensit	FrcElmntOrgSensitivityDesc
Product.DfsProduct	DfsProductSensitivity
Product.DfsManufac	DfsManufacturerPartLongNumber
Product.ProductMar	DfsProductManufacturerNumber
_DefenseLogisticsl	DfsLogisticsLevelName
Product.DfsLogist:	DfsLogisticsLevel
_DfsReturnDelivery	DfsReturnDeliveryName
Product.DfsReturnI	DfsReturnDelivery
_DfsNationalItmIdr	DfsNationalItemIdnNumberName
Product.DfsNationa	DfsNationalItemIdnNumber
_DfsMatlConditionM	DfsMatlConditionMgmtName
Product.DfsMatlCor	DfsMatlConditionMgmt
	led
	len
	len

I_DfsReturnDeliveryText → AS _DfsReturnDeliveryText Column X_DFS_C_PRODU

If an association is defined in a view extension, then a link to the view extension is available on the node that represents the association:


has 1:* _DfsReturnDelivery
I_DfsReturnDeliveryText AS _DfsReturnDeliveryText
CHAR(1)
External Links
X_DFS_C_PRODUCT
Close

Such a link is also available on the associated datasource itself (along with the link to the data source):



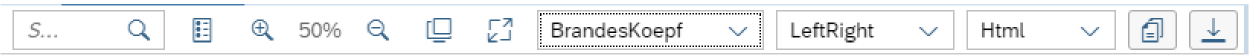
3.2.2.11.3 Select which extensions to display

You can control which extensions are to be displayed by using the extension selector in the settings panel. The extension selector is a multi-combobox that lets you select some or all extensions you would like the see in the Structure Diagram.

If the Object detail page is opened on a view, then by default all of its extensions are loaded and selected. If the Object detail page is opened on an extension itself, then all extensions are loaded, but only the current one is selected.

3.2.2.12 *Structure Diagram Toolbar*

The toolbar allows the user to customize the structure diagram to their particular needs. The following functions are included in the toolbar *(explained from left to right)*:

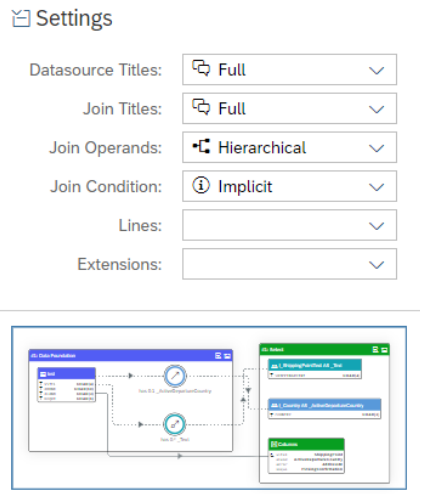


The following functions are included in the toolbar *(explained from left to right)*:

- The **Search Bar** allows you to choose from a list of nodes, associations and entities that are visualized in the diagram. By selecting any of the entities the visualization will automatically jump to and highlight the selected entity.
- The **Legend Button** is an on/off toggle to display or remove the display of the legend in the visualization of the structure.
- The **Zoom Controller** allows you to increase or decrease the zoom level of the visualization. This can also be done by using the scroll wheel on the mouse for a more precise approach.
- The **Zoom to Fit button** automatically applies a zoom level on which the entire visualization is visible. This is particularly useful for more complex and larger CDS views.
- The **Enter Fullscreen button** fills the visualization and the toolbar to the full browser size. By selecting this button again, the user is able to return to the previous viewing mode.
- The **Visualization Definition Dropdown Box** allows you to change the format of the visualization to suit their particular needs. The user can choose from Brandes Koepf, Linear Segments or Simple visualizations. The default value is Brandes Koepf.
- The **Flow Direction Dropdown Box** allows you to define the direction of the data flow in the visualization. The user can choose from left to right, right to left, top to bottom or bottom to top. The default value is from left to right.
- The **Format Dropdown Box** allows you to define the format of the visualization, which can be HTML of SVG. The default value is HTML.
- The **Copy Button** can be used to copy the chosen format to the clipboard.
- The **Download Button** can be used to download the visualization to your local machine.

3.2.2.13 Structure Diagram Settings

On the right-hand side of the work area, you can find the settings page. This section provides the possibility to define settings for customizing the visualization. You can minimize / maximize this tab by clicking on the button to the left of the title.



Settings

Datasource Titles: Full

Join Titles: Full

Join Operands: Hierarchical

Join Condition: Implicit

Lines:

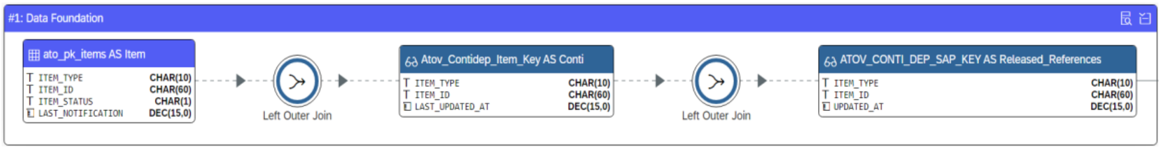
Extensions:

The following functions are included in the settings *(explained from top to bottom)*:

- The **Data Source Titles** option allows you to define the length of the data source titles in the visualization. The options are:
  - Full: Data sources are shown as `<ObjectName> as <alias>`.
  - Short: only the name (or, if present, the alias) is shown in the title.
- The **Join Titles** option allows you to define the length of the join definitions in the visualization. The options are:
  - Full: join titles are shown as `<left datasource> <join type> <right datasource>`, where `<join type>` indicates the type of join operation (Left Outer Join, Right Outer Join, Inner Join, Cross Join)

   - Short: only the join type is shown. In this case. This means you need to follow the lines that are connected to the join node to figure out
- The **Join Operands** option allows you to define how joins are visualized. The options are:
   - Sequential: In this layout, join nodes appear in between the joined data sources.
   - Hierarchical: In this layout, each join appears as separate operator that accepts the data sources as operands.

  By default, joins are rendered using the Hierarchical option. This was already shown in a prior example on the view `Atov_Pk_Item_ContiDep_Join`.
  This is what the data foundation looks like in the Structure diagram using the Sequential option:



  While the sequential layout may seem simpler than the hierarchical option, it has a drawback in that its graphical representation obscures the actual flow of data.We can understand this when we ask ourselves what the data flow looks right after each node in the graph.

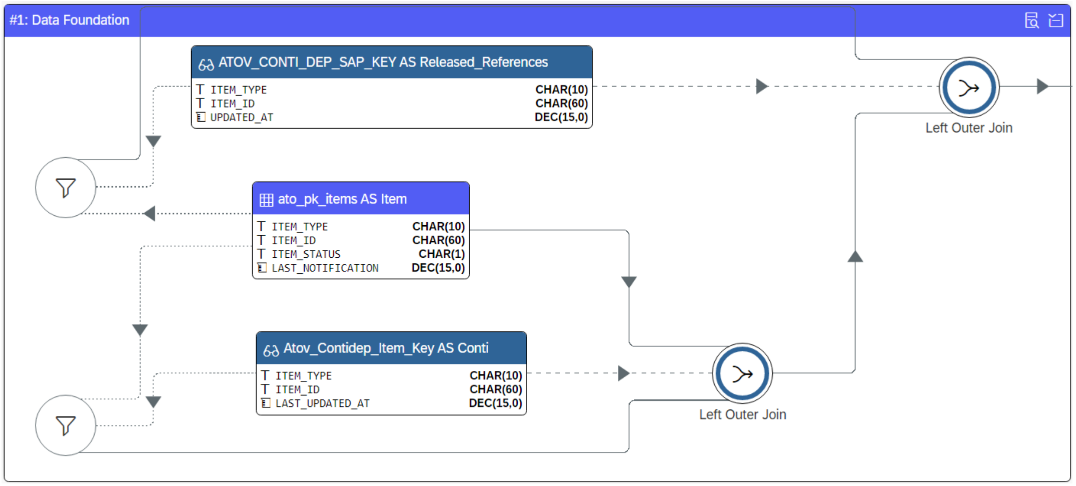
  Conceptually, a join is an operation that combines the rows and columns of its operand data sources. So, right after each join node, we should have all columns from both its operands. But in the sequential layout, the right operand appears after the join node. The hierarchical layout on the other hand expresses this quite naturally.

  In practice, this need not be of much concern. Whether the sequential or the hierarchical layout is more appropriate is a matter of balance between complexity of the view, accuracy of the graphical representation, and personal preference.

- The **Join Condition** lets you control the level of detail used to render joins. The options are:
   - Explicit: the join operation and the condition applied to the join each get their own node
   - Implicit: the condition and the join operation are represented by a single node.

  In the prior example on the `Atov_Pk_Item_ContiDep_Join` view, the default implicit option is used. The Explicit option may be useful in combination with the Join Condition option for the Lines selector. For an example, please refer to the Lines selection.

- The **Lines** selector allows you to control whether columns used in either join conditions or group by operations are explicitly drawn. The Lines selector has the following options:
   - Join Condition: when selected, lines will be drawn from each data source to each join that uses a column from that data source in the join condition. When enabling this option, it will be useful to also choose the “Explicit” option for the Join Condition setting (see above). This will allow you to distinguish between lines that denote join operands and the usage of columns in a join condition. The following screenshot of the data foundation of the `Atov_Pk_Item_ContiDep_Join` view illustrates both these settings:

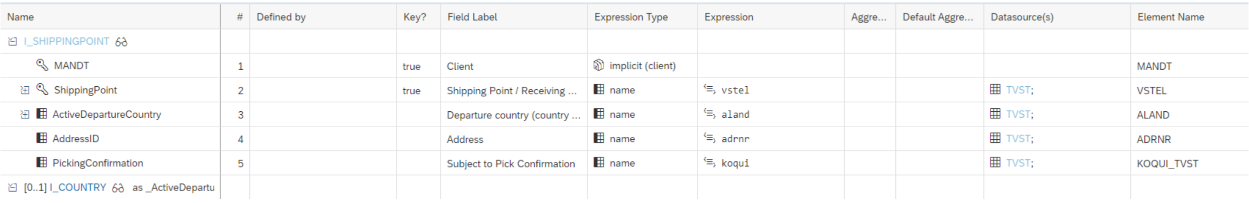


  - Group By List: when selected, lines will drawn from each data source of which a column appears in the group by list. Of course, this option is only useful in case the CDS view has a group by list.

- The **Extensions** option allows you to select any extensions you wish to include or exclude for the CDS object. Please note that this only works for CDS views that have extensions, if no extensions exist for the CDS view this field will remain blank.

3.2.3 CDS Columns Page

The columns page provides an overview of the columns in the CDS view, as well as the columns of the associated CDS views. This section will provide an overview of the functions of this page. Below you can find an example of the columns included in I_ShippingPoint for illustrative purposes.



Name	#	Defined by	Key?	Field Label	Expression Type	Expression	Aggre...	Default Aggre...	Datasource(s)	Element Name
I_SHIPPINGPOINT										
MANDT	1		true	Client	implicit (client)					MANDT
ShippingPoint	2		true	Shipping Point / Receiving ...	name	vstel			TVST;	VSTEL
ActiveDepartureCountry	3			Departure country (country ...	name	aland			TVST;	ALAND
AddressID	4			Address	name	adrnr			TVST;	ADRNR
PickingConfirmation	5			Subject to Pick Confirmation	name	koqui			TVST;	KOQUI_TVST
[0..1] I_COUNTRY as _ActiveDepartu										

3.2.3.1 Columns Toolbar

The column toolbar has three main functionalities:

1. The **Expand** button allows you to quickly expand all columns in the columns page to the desired of the three levels. The first layer defines the grouping that the columns belong to, this can be the CDS view or any of the associations in the CDS view. The second layer defines the names of the columns in that particular CDS view or association. The third layer can define any associations of a particular column.
2. The **Open Settings** button opens a dialog box that allows you to select which columns they want to see. Simply select or deselect any of the columns and select OK, this will automatically update the columns page.
3. The **Download** button allows you to download an Excel file of the columns page.

3.2.3.2 The Columns

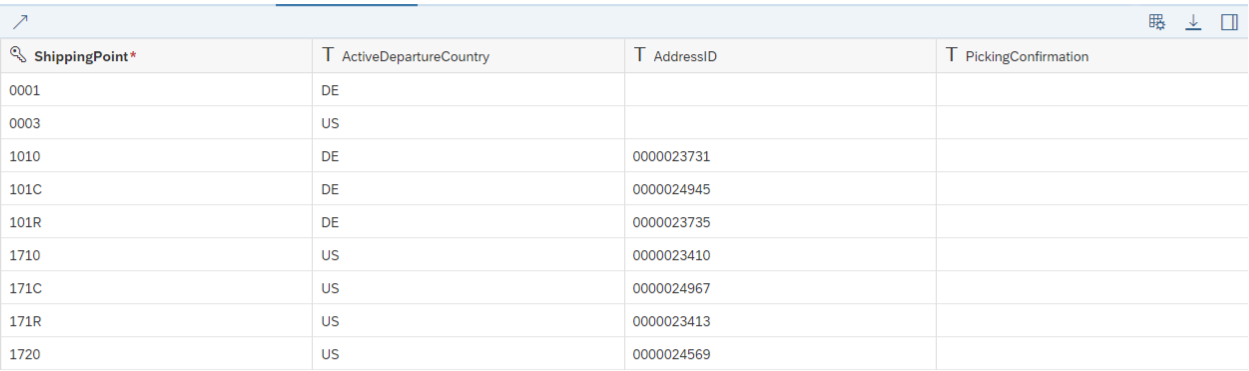
The following columns are included in the columns page, explained from left to right:

- The **Name** column consists of three layers, as explained in the Expand button, and defines the name of the column.
- The **#** is an incremental counter that starts from 1 for each layer. For example, in the column page of I_ShippingPoint the MANDT column is the 1st column.
- The **Defined By** column is empty, unless the column is defined by an extension. Then, the column will list the name of the extension, which you can click on to go to the extension definition.
- The **Key?** Column is empty, unless the column is a key field. Then, the column will have the value "true" filled.
- The **Field Label** column is a description of the column that can be used to identify what the column represents.
- The **Expression Type** defines how the expression is calculated and is determined by the implementing class.
- The **Expression** column is only filled for the CDS view columns are represents the source expression of the column. For example, the SAP defined base table column name, or the associated CDS view and the selected column.

- The **Aggregation** column defines if a column is aggregated, as well as the type of aggregation used.
- The **Default Aggregation** column defines the default aggregation for a column.
- The **Data source(s)** column is only filled for the CDS view columns and defined the source of the CDS view columns.
- The **Element Name** provides the SAP defined element name.
- The **Domain** refers to the defined domain that is associated with the element name.

3.2.4 CDS Data Preview

The CDS Data Preview page allows you to quickly preview a defined no. of rows from the CDS view for quick insights. Below you can find an example of the data preview page of the I_ShippingCondition CDS view.



ShippingPoint*	ActiveDepartureCountry	AddressID	PickingConfirmation
0001	DE		
0003	US		
1010	DE	0000023731	
101C	DE	0000024945	
101R	DE	0000023735	
1710	US	0000023410	
171C	US	0000024967	
171R	US	0000023413	
1720	US	0000024569	

The main section of the work area is dedicated to the data preview, which shows the selected CDS view columns and the selected no. of rows with their respective values. The column width can be adjusted if required.

On the bottom of the work area you can find the number of rows that were retrieved with the data preview, as well as the time it took to retrieve the data preview.

3.2.4.1 Data Preview Toolbar

The Data Preview page has a toolbar with four functionalities:

1. The **Toggle Data Preview of Associated Entities** button opens a dialog box that allows you to select additional associated entities for which a data preview is required. Doing so will add a data preview window at the bottom of the work area.
2. The **Open Settings** button opens a dialog box that allows you to select which columns they want to see. Simply select or deselect any of the columns and select OK, this will automatically update the data preview page.
3. The **Download** button allows you to download an Excel file of the data preview page.
4. The **Settings** button allows you to open the settings menu, in which you can define the no. of rows to be previewed as well as any filters they wish to apply on any of the columns from the preview.

3.2.5 CDS Dependencies Page

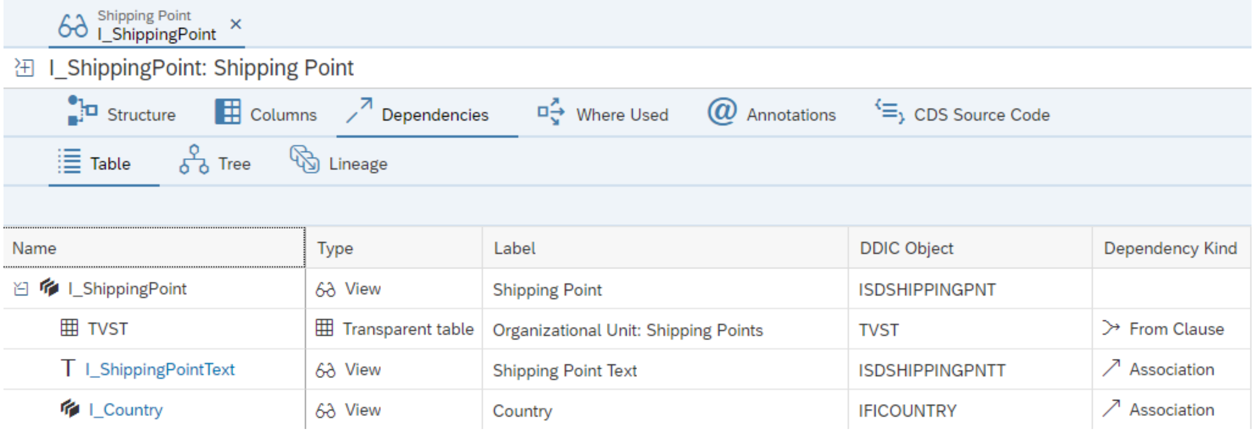
The dependencies page displays information about any other objects that the current CDS object relies upon. The dependency information is divided into 3 subpages:

- Object Dependency Table
- Dependency Tree Diagram
- Column Lineage Diagram

3.2.5.1 Object Dependency Table

The Object Dependency Table shows all items upon which the selected CDS view is dependent on in a tabular format. This allows you to quickly identify where data is coming from in a structured format. For example, in this case we can see that I_ShippingPoint directly selects data from the TVST table, together with two associations to intermediary CDS views.

For each item various elements are displayed in the table. You can click on a CDS view to directly open it as an additional tab for a more thorough analysis. The first hierarchy in the table is that of the selected view. Any additional hierarchies display the dependencies of the dependencies of the selected CDS view.



I_ShippingPoint: Shipping Point

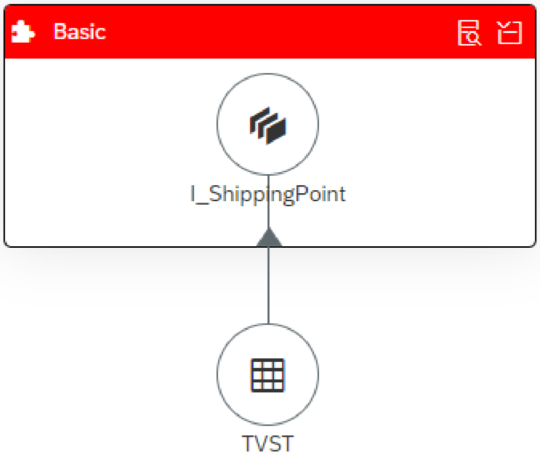
Name	Type	Label	DDIC Object	Dependency Kind
I_ShippingPoint	View	Shipping Point	ISDSHIPPINGPNT	
TVST	Transparent table	Organizational Unit: Shipping Points	TVST	From Clause
I_ShippingPointText	View	Shipping Point Text	ISDSHIPPINGPNTT	Association
I_Country	View	Country	IFICOUNTRY	Association

The top-right hand side offers a selection of:

- The **Expand** button which allows you to quickly expand or collapse the hierarchies in the table to the level of your preference.
- The **Properties** button which allows you to select the columns you wish to see displayed in the table. For example, if you're not interested in the ODP Data Source simply deselect it.
- The **Download** button which will generate the tabular overview in the .xlsx format which you can download and edit / display locally.

3.2.5.2 Dependency Tree Diagram

The dependency tree diagram is a different way of visualizing the dependencies of the CDS view. In this case, a dependency tree is generated based on the existing dependencies. For example, below you can find the default dependency tree of I_ShippingPoint:



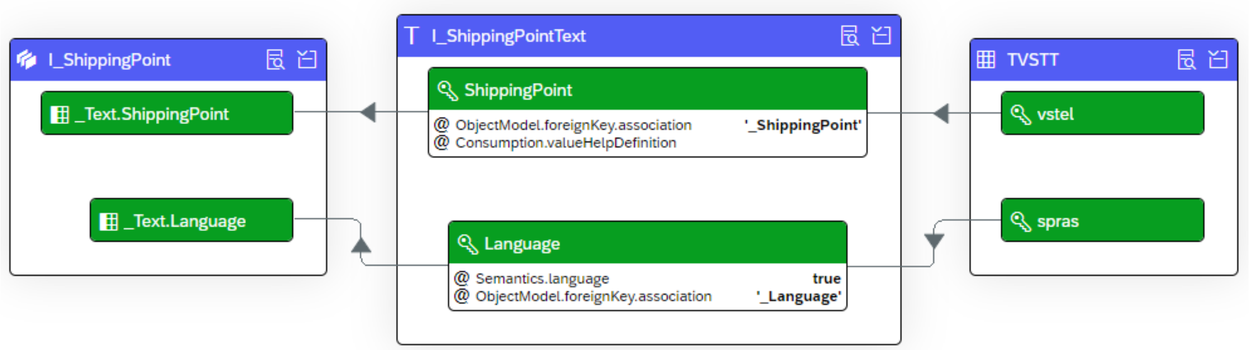
The dependency tree diagram toolbar is designed identically to the toolbar in the structure, for more information see 3.2.2.12. The user can click on any entity in the diagram to click through to the details page of the selected entity.

The settings tab on the right-hand side allows you to customize the elements of the tree:

- The **Entity Titles** setting changes what is displayed as the title of each entity in the tree diagram. You can choose between the entity name, the DDIC name and the label text. By default, the entity name is selected.
- The **Dependency Types** setting changes what types of dependencies are displayed in the diagram. You can choose to include and / or exclude associations, from clauses and extensions. By default, this excludes associations.
- The **Grouping** setting changes how elements are grouped in the tree diagram. This is particularly useful when you're looking to identify similarities / differences between entities such as package or data categories. You can choose from various options *(e.g.: Analytics Data Category, Package, VDM View Type)*. By default, this is set to the VDM View Type.

3.2.5.3 Column Lineage Diagram

The Column Lineage Diagram visualized the flow of data from the source of the data to the currently selected CDS view. This is a very useful feature for analyzing data flows, applied conditions and associated views. For example, below you will find a visualization of two key columns from the I_ShippingPoint CDS view:


I_ShippingPoint
_Text.ShippingPoint
_Text.Language
I_ShippingPointText
ShippingPoint
@ ObjectModel.foreignKey.association '_ShippingPoint'
@ Consumption.valueHelpDefinition
Language
@ Semantics.language true
@ ObjectModel.foreignKey.association '_Language'
TVSTT
vstel
spras

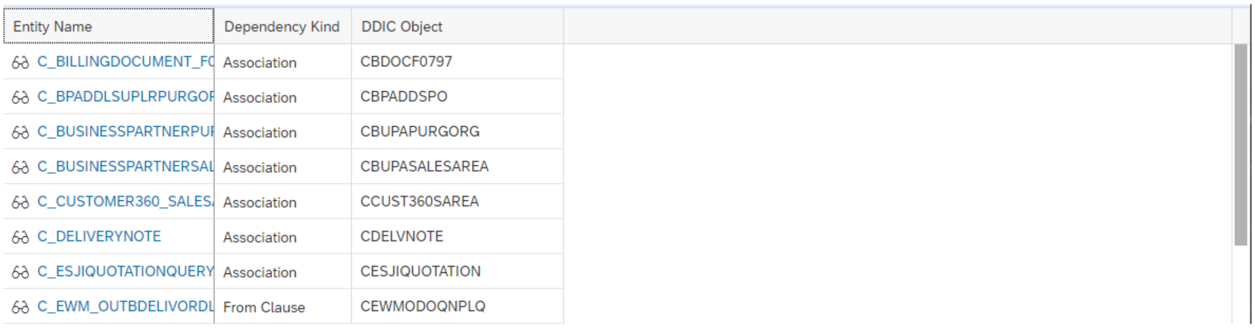
The dependency tree diagram toolbar is designed identically to the toolbar in the structure, for more information see 3.2.2.12.

The settings tab on the right-hand side allows you to select:

- The **Detail Level**, which defined the level of details displayed in the visualization. Selecting the full visualization shows the entire flow with all conditions, whereas the base column visualization only shows the flow to the base column and table.
- The **Columns**, which allow you to choose and pick which columns you would like to see visualized.

3.2.6 CDS Where-Used List

The Where-Used List provides an overview of all of the entities in which the selected CDS entity is used, sorted alphabetically on the DDIC object name of the entity that uses the CDS entity. This table also specifies the kind of dependency that the CDS entity has on the currently selected CDS view. Below you can find an example of a where used list for the I_ShippingCondition CDS view:

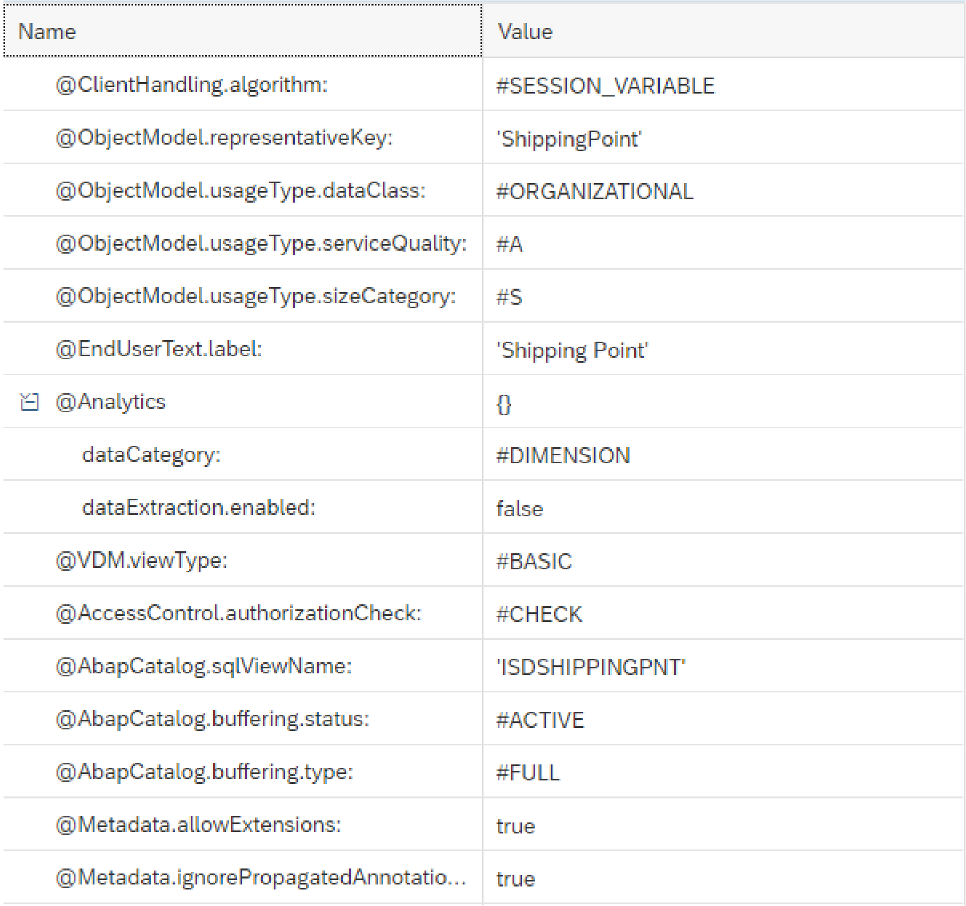


Entity Name	Dependency Kind	DDIC Object
C_BILLINGDOCUMENT_FC	Association	CBDOCF0797
C_BPADDLSUPLRPURGOF	Association	CBPADDSPO
C_BUSINESSPARTNERPUF	Association	CBUPAPURGORG
C_BUSINESSPARTNERSAL	Association	CBUPASALESAREA
C_CUSTOMER360_SALES/	Association	CCUST360SAREA
C_DELIVERYNOTE	Association	CDELVNOTE
C_ESJIQUOTATIONQUERY	Association	CESJIQUOTATION
C_EWM_OUTBDELIVORDL	From Clause	CEWMODOQNPLQ

3.2.7 CDS Annotations Page

The CDS annotations page provides a one stop overview of all of the annotations and their respective values of the selected CDS view. This allows you to easily identify and analyze the annotations that affect this CDS view.

For example, below you will find the CDS annotations of the CDS view I_ShippingPoint:



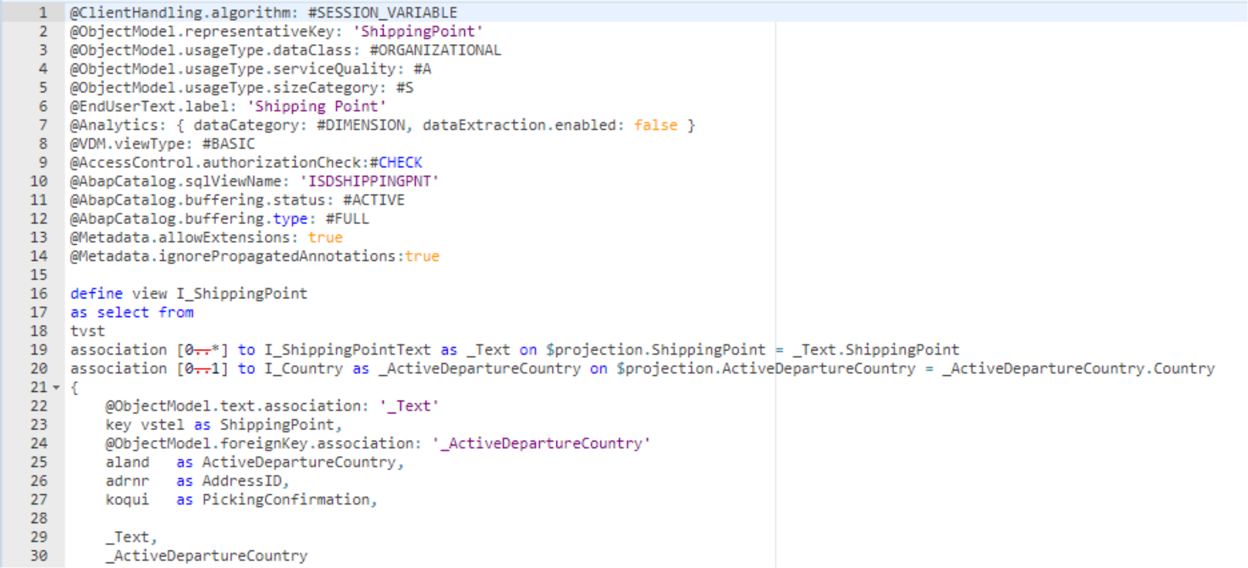
Name	Value
@ClientHandling.algorithm:	#SESSION_VARIABLE
@ObjectModel.representativeKey:	'ShippingPoint'
@ObjectModel.usageType.dataClass:	#ORGANIZATIONAL
@ObjectModel.usageType.serviceQuality:	#A
@ObjectModel.usageType.sizeCategory:	#S
@EndUserText.label:	'Shipping Point'
⊟ @Analytics	{}
dataCategory:	#DIMENSION
dataExtraction.enabled:	false
@VDM.viewType:	#BASIC
@AccessControl.authorizationCheck:	#CHECK
@AbapCatalog.sqlViewName:	'ISDSHIPPINGPNT'
@AbapCatalog.buffering.status:	#ACTIVE
@AbapCatalog.buffering.type:	#FULL
@Metadata.allowExtensions:	true
@Metadata.ignorePropagatedAnnotatio...	true

The top-right hand side offers a selection of:

- The **Expand** button ↓≡ which allows you to quickly expand or collapse the hierarchies in the table to the level of your preference.
- The **Properties** button ⚙ which allows you to select the columns you wish to see displayed in the table. For example, if you're not interested in the value of the annotations, you simply deselect it.
- The **Download** button ⤓ which will generate the tabular overview in the .xlsx format which you can download and edit / display locally.

3.2.8 CDS Source Code

The CDS Source Code does exactly what it depicts, it is a 1:1 representation of the CDS Code underlying the CDS view. You can use this to analyze and / or identify important aspects of the CDS views through code, whereas the other pasts of S4Explorer provide additional functionalities to do this. For example, below you can find the CDS source code of the I_ShippingPoint CDS view:



```
@ClientHandling.algorithm: #SESSION_VARIABLE
@ObjectModel.representativeKey: 'ShippingPoint'
@ObjectModel.usageType.dataClass: #ORGANIZATIONAL
@ObjectModel.usageType.serviceQuality: #A
@ObjectModel.usageType.sizeCategory: #S
@EndUserText.label: 'Shipping Point'
@Analytics: { dataCategory: #DIMENSION, dataExtraction.enabled: false }
@VDM.viewType: #BASIC
@AccessControl.authorizationCheck:#CHECK
@AbapCatalog.sqlViewName: 'ISDSHIPPINGPNT'
@AbapCatalog.buffering.status: #ACTIVE
@AbapCatalog.buffering.type: #FULL
@Metadata.allowExtensions: true
@Metadata.ignorePropagatedAnnotations:true

define view I_ShippingPoint
as select from
tvst
association [0..*] to I_ShippingPointText as _Text on $projection.ShippingPoint = _Text.ShippingPoint
association [0..1] to I_Country as _ActiveDepartureCountry on $projection.ActiveDepartureCountry = _ActiveDepartureCountry.Country
{
    @ObjectModel.text.association: '_Text'
    key vstel as ShippingPoint,
    @ObjectModel.foreignKey.association: '_ActiveDepartureCountry'
    aland   as ActiveDepartureCountry,
    adrnr   as AddressID,
    koqui   as PickingConfirmation,

    _Text,
    _ActiveDepartureCountry
```

The source code viewer also provides the ability to ease the reading of complicated sets of CDS source code by adding line numbering. In addition, the user is able to close and / or (re)open sets of code in the CDS source code by collapsing or expanding the clauses. This can be done by clicking on the collapse icon, located next to the row number.

The top-right hand side offers a selection of:

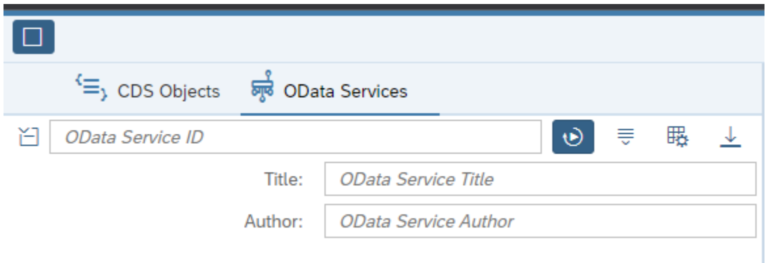
- The **Copy** button which will copy the CDS view source code to your clipboard.
- The **Download** button which will generate the CDS view source code in the. hdbdd format available for download.

4 Working with OData Services

The other main key functionality of S4Explorer is to enable you to analyze, understand and work with SAP OData Services more efficiently than ever before. The following chapter will elaborate on the OData Services Catalog, how to analyze entities in the catalog and all sub-features and options associated with it.

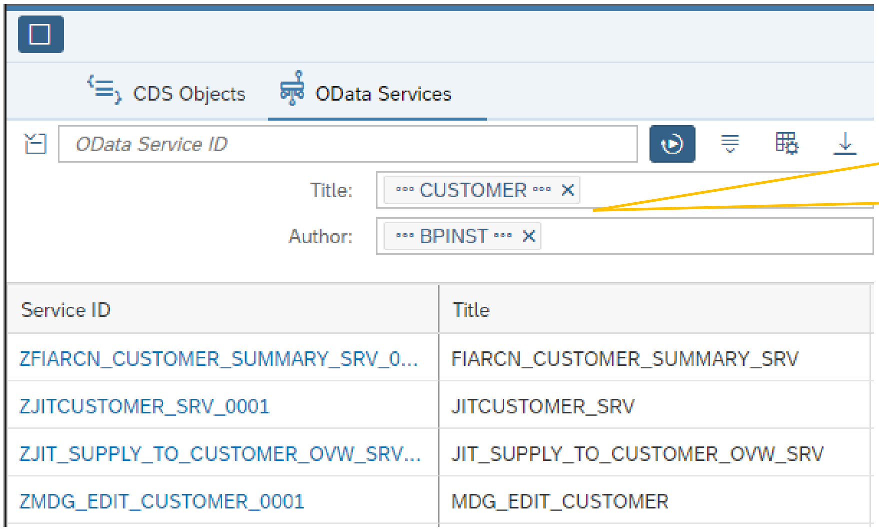
4.1 The OData Services Catalog

The ODATA Services Catalog is the collection of ODATA Services available in your SAP S/4HANA build. You can use the catalog to find and work with S/4HANA ODATA Services. Opening the catalog is as simple as selecting the ODATA Services in the top-left corner.



4.1.1 OData Catalog Fields Reference

The ODATA Catalog Fields Reference is a collection of attributes available in the ODATA catalog. You can use the title and author of the ODATA Service to search and identify the right ODATA service quickly and easily.



Title: CUSTOMER

Author: BPINST

Service ID	Title
ZFIARCN_CUSTOMER_SUMMARY_SRV_0...	FIARCN_CUSTOMER_SUMMARY_SRV
ZJITCUSTOMER_SRV_0001	JITCUSTOMER_SRV
ZJIT_SUPPLY_TO_CUSTOMER_OVW_SRV...	JIT_SUPPLY_TO_CUSTOMER_OVW_SRV
ZMDG_EDIT_CUSTOMER_0001	MDG_EDIT_CUSTOMER

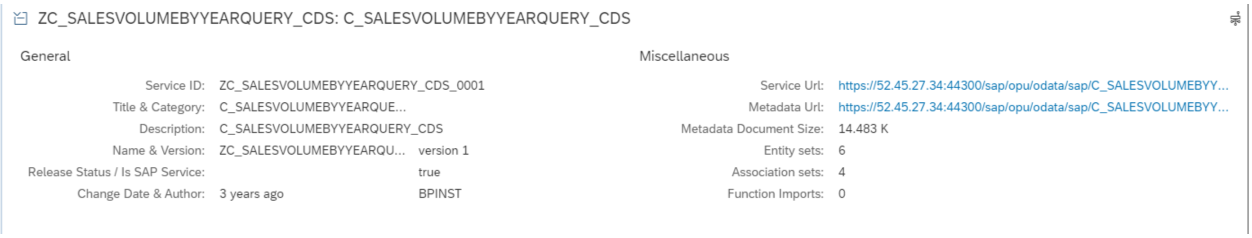
Searching for *CUSTOMER* and BPINST returns a list of all ODATA Services that match these criteria.

4.2 OData Detail Page

Selecting a ODATA Service Entity from the ODATA object catalog will opens its object detail page in the work area. The following sections will elaborate on the available functionalities S4Explorer offers to analyze and understand the ODATA object.

4.2.1 OData Detail Page Header

The page header provides an overview of a pre-defined set of key service characteristic. The page header can be minimized / maximized by clicking on the button left to the CDS view name. Below you can find an example of the detail page header for the ZC_SALESVOLUMEBYYEARQUERY_CDS Service:



ZC_SALESVOLUMEBYYEARQUERY_CDS: C_SALESVOLUMEBYYEARQUERY_CDS

General

Service ID:	ZC_SALESVOLUMEBYYEARQUERY_CDS_0001	
Title & Category:	C_SALESVOLUMEBYYEARQUE...	
Description:	C_SALESVOLUMEBYYEARQUERY_CDS	
Name & Version:	ZC_SALESVOLUMEBYYEARQU...	version 1
Release Status / Is SAP Service:		true
Change Date & Author:	3 years ago	BPINST

Miscellaneous

Service Url:	https://52.45.27.34:44300/sap/opu/odata/sap/C_SALESVOLUMEBYY...
Metadata Url:	https://52.45.27.34:44300/sap/opu/odata/sap/C_SALESVOLUMEBYY...
Metadata Document Size:	14.483 K
Entity sets:	6
Association sets:	4
Function Imports:	0

The following fields can be found in the page header:

- The **Service ID** refers to the unique ID of the ODATA Service entity, which is an attribute of the domain model. The Service ID is also shown in the search results table by default.
- The **Title**, or name, is the name given to the ODATA Service. The title is also displayed in the header of the detail page.
- The **Description** provides a brief overview of the function of the ODATA Service which can be used to identify if this is the correct service you would like to use / analyze.
- The **Release Status** indicates the release status of the ODATA Service entity.
- **IsSAP Service** indicates with a TRUE / FALSE marker whether the ODATA Service entity is an SAP defined entity.
- The **Changed when & by** provides an overview of the latest activated change date time, as well as the user responsible for these changes.
- The **Service Url** provides a clickable URL which will open an XML file containing the service aspect of the ODATA Service entity.
- The **Metadata Url** provides a clickable URL which will open an XML file containing the metadata of the ODATA Service entity.
  - Concurrently, the **Metadata Document Size** indicates the size in K of the listed metadata XML file.

- **Entity Sets** are named collections of entities *(E.G.: Customer, Employee, etc.)*. For example. The entity set Customers would be the collection of the entity's customers. The header displays the number of entity sets in the ODATA Service entity.

- **Association Sets** refer to the associations between entities described in relationships. Again, the detail page only lists the number of associations in the ODATA Service entity.

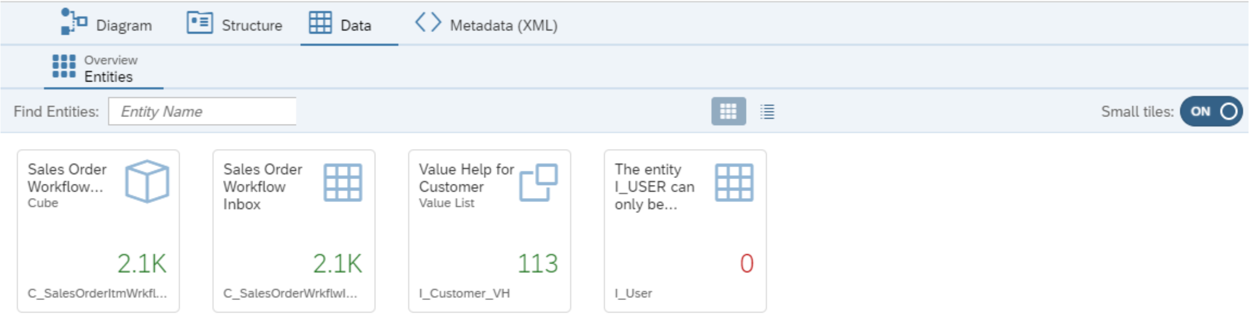
- The **Function Imports** refer to a action that can executed in the S/4HANA system, which the ODATA Service entity can call upon. The detail page lists the number of function imports in the ODATA Service Entity.

4.2.2 OData Structure Tree

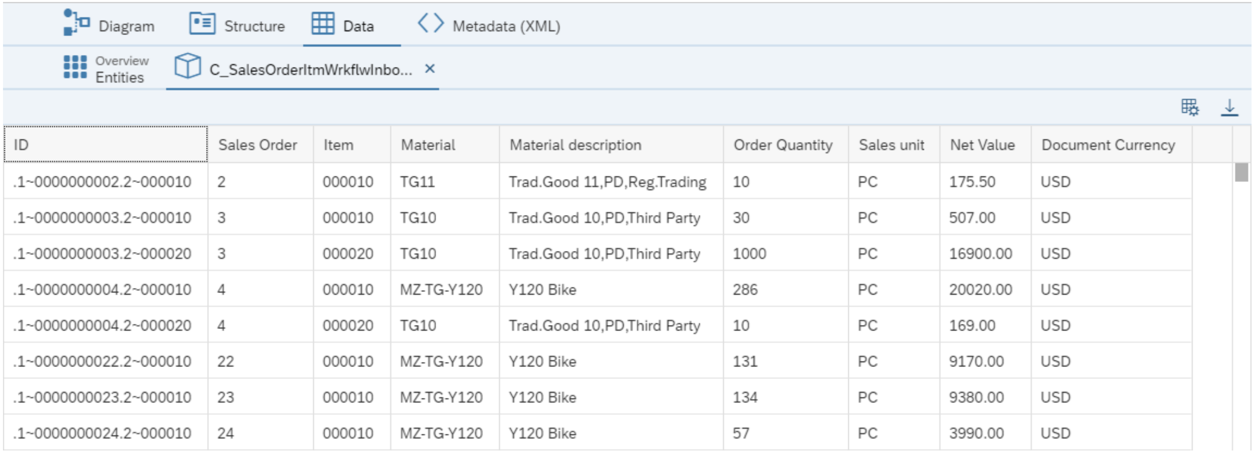
4.2.3 OData Structure Diagram

4.2.4 OData Data Preview Page

The ODAA Data Preview page allows you to get an insight in the entities of an ODATA Service Entity. You can select the entity you're interested in, which will take you to the data preview of the selected entity.



If the ODATA Entity that you selected has a ton of entities, you can select the Small tiles option to resize the tiles to fit in the workspace area. By default, entities are shown as tiles. However, if you wish, you can switch to a column table by switching these buttons .



ID	Sales Order	Item	Material	Material description	Order Quantity	Sales unit	Net Value	Document Currency
.1~0000000002.2~000010	2	000010	TG11	Trad.Good 11,PD,Reg.Trading	10	PC	175.50	USD
.1~0000000003.2~000010	3	000010	TG10	Trad.Good 10,PD,Third Party	30	PC	507.00	USD
.1~0000000003.2~000020	3	000020	TG10	Trad.Good 10,PD,Third Party	1000	PC	16900.00	USD
.1~0000000004.2~000010	4	000010	MZ-TG-Y120	Y120 Bike	286	PC	20020.00	USD
.1~0000000004.2~000020	4	000020	TG10	Trad.Good 10,PD,Third Party	10	PC	169.00	USD
.1~0000000022.2~000010	22	000010	MZ-TG-Y120	Y120 Bike	131	PC	9170.00	USD
.1~0000000023.2~000010	23	000010	MZ-TG-Y120	Y120 Bike	134	PC	9380.00	USD
.1~0000000024.2~000010	24	000010	MZ-TG-Y120	Y120 Bike	57	PC	3990.00	USD

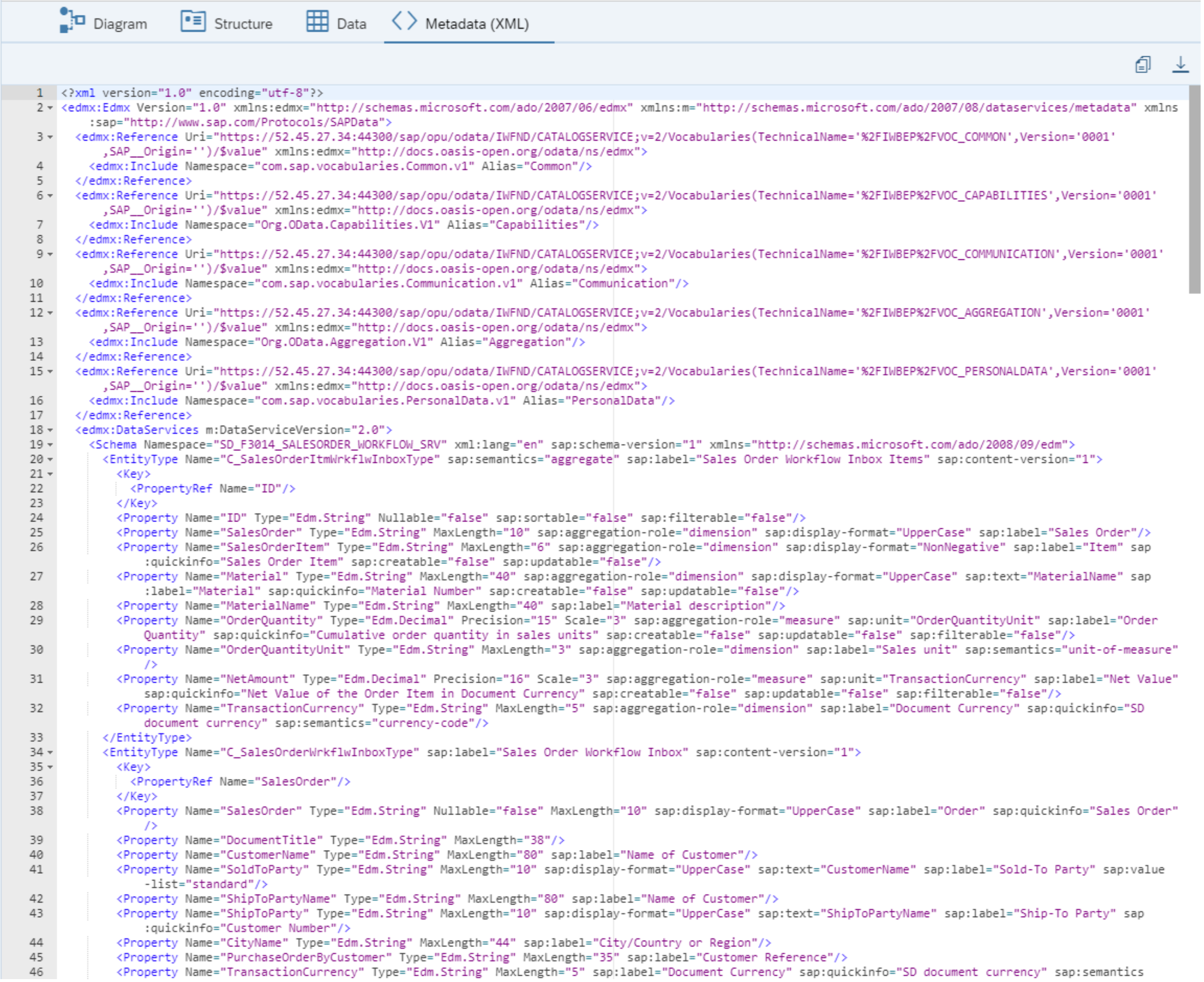
The column width can be adjusted if required. On the bottom of the work area, you can find the number of rows that were retrieved with the data preview.

The Data Preview page has a toolbar with four functionalities:

1. The **Open Settings** button opens a dialog box that allows you to select which columns they want to see. Simply select or deselect any of the columns and select OK, this will automatically update the data preview page.
2. The **Download** button allows you to download an Excel file of the data preview page.

4.2.5 Metadata (XML)

The metadata (XML) page provides the XML formatted metadata for you to browse, copy or download.



```xml
<?xml version="1.0" encoding="utf-8"?>
<edmx:Edmx Version="1.0" xmlns:edmx="http://schemas.microsoft.com/ado/2007/06/edmx" xmlns:m="http://schemas.microsoft.com/ado/2007/08/dataservices/metadata" xmlns
    :sap="http://www.sap.com/Protocols/SAPData">
  <edmx:Reference Uri="https://52.45.27.34:44300/sap/opu/odata/IWFND/CATALOGSERVICE;v=2/Vocabularies(TechnicalName='%2FIWBEP%2FVOC_COMMON',Version='0001'
      ,SAP__Origin='')/$value" xmlns:edmx="http://docs.oasis-open.org/odata/ns/edmx">
    <edmx:Include Namespace="com.sap.vocabularies.Common.v1" Alias="Common"/>
  </edmx:Reference>
  <edmx:Reference Uri="https://52.45.27.34:44300/sap/opu/odata/IWFND/CATALOGSERVICE;v=2/Vocabularies(TechnicalName='%2FIWBEP%2FVOC_CAPABILITIES',Version='0001'
      ,SAP__Origin='')/$value" xmlns:edmx="http://docs.oasis-open.org/odata/ns/edmx">
    <edmx:Include Namespace="Org.OData.Capabilities.V1" Alias="Capabilities"/>
  </edmx:Reference>
  <edmx:Reference Uri="https://52.45.27.34:44300/sap/opu/odata/IWFND/CATALOGSERVICE;v=2/Vocabularies(TechnicalName='%2FIWBEP%2FVOC_COMMUNICATION',Version='0001'
      ,SAP__Origin='')/$value" xmlns:edmx="http://docs.oasis-open.org/odata/ns/edmx">
    <edmx:Include Namespace="com.sap.vocabularies.Communication.v1" Alias="Communication"/>
  </edmx:Reference>
  <edmx:Reference Uri="https://52.45.27.34:44300/sap/opu/odata/IWFND/CATALOGSERVICE;v=2/Vocabularies(TechnicalName='%2FIWBEP%2FVOC_AGGREGATION',Version='0001'
      ,SAP__Origin='')/$value" xmlns:edmx="http://docs.oasis-open.org/odata/ns/edmx">
    <edmx:Include Namespace="Org.OData.Aggregation.V1" Alias="Aggregation"/>
  </edmx:Reference>
  <edmx:Reference Uri="https://52.45.27.34:44300/sap/opu/odata/IWFND/CATALOGSERVICE;v=2/Vocabularies(TechnicalName='%2FIWBEP%2FVOC_PERSONALDATA',Version='0001'
      ,SAP__Origin='')/$value" xmlns:edmx="http://docs.oasis-open.org/odata/ns/edmx">
    <edmx:Include Namespace="com.sap.vocabularies.PersonalData.v1" Alias="PersonalData"/>
  </edmx:Reference>
  <edmx:DataServices m:DataServiceVersion="2.0">
    <Schema Namespace="SD_F3014_SALESORDER_WORKFLOW_SRV" xml:lang="en" sap:schema-version="1" xmlns="http://schemas.microsoft.com/ado/2008/09/edm">
      <EntityType Name="C_SalesOrderItmWrkflwInboxType" sap:semantics="aggregate" sap:label="Sales Order Workflow Inbox Items" sap:content-version="1">
        <Key>
          <PropertyRef Name="ID"/>
        </Key>
        <Property Name="ID" Type="Edm.String" Nullable="false" sap:sortable="false" sap:filterable="false"/>
        <Property Name="SalesOrder" Type="Edm.String" MaxLength="10" sap:aggregation-role="dimension" sap:display-format="UpperCase" sap:label="Sales Order"/>
        <Property Name="SalesOrderItem" Type="Edm.String" MaxLength="6" sap:aggregation-role="dimension" sap:display-format="NonNegative" sap:label="Item" sap
            :quickinfo="Sales Order Item" sap:creatable="false" sap:updatable="false"/>
        <Property Name="Material" Type="Edm.String" MaxLength="40" sap:aggregation-role="dimension" sap:display-format="UpperCase" sap:text="MaterialName" sap
            :label="Material" sap:quickinfo="Material Number" sap:creatable="false" sap:updatable="false"/>
        <Property Name="MaterialName" Type="Edm.String" MaxLength="40" sap:label="Material description"/>
        <Property Name="OrderQuantity" Type="Edm.Decimal" Precision="15" Scale="3" sap:aggregation-role="measure" sap:unit="OrderQuantityUnit" sap:label="Order
            Quantity" sap:quickinfo="Cumulative order quantity in sales units" sap:creatable="false" sap:updatable="false" sap:filterable="false"/>
        <Property Name="OrderQuantityUnit" Type="Edm.String" MaxLength="3" sap:aggregation-role="dimension" sap:label="Sales unit" sap:semantics="unit-of-measure"
            />
        <Property Name="NetAmount" Type="Edm.Decimal" Precision="16" Scale="3" sap:aggregation-role="measure" sap:unit="TransactionCurrency" sap:label="Net Value"
            sap:quickinfo="Net Value of the Order Item in Document Currency" sap:creatable="false" sap:updatable="false" sap:filterable="false"/>
        <Property Name="TransactionCurrency" Type="Edm.String" MaxLength="5" sap:aggregation-role="dimension" sap:label="Document Currency" sap:quickinfo="SD
            document currency" sap:semantics="currency-code"/>
      </EntityType>
      <EntityType Name="C_SalesOrderWrkflwInboxType" sap:label="Sales Order Workflow Inbox" sap:content-version="1">
        <Key>
          <PropertyRef Name="SalesOrder"/>
        </Key>
        <Property Name="SalesOrder" Type="Edm.String" Nullable="false" MaxLength="10" sap:display-format="UpperCase" sap:label="Order" sap:quickinfo="Sales Order"
            />
        <Property Name="DocumentTitle" Type="Edm.String" MaxLength="38"/>
        <Property Name="CustomerName" Type="Edm.String" MaxLength="80" sap:label="Name of Customer"/>
        <Property Name="SoldToParty" Type="Edm.String" MaxLength="10" sap:display-format="UpperCase" sap:text="CustomerName" sap:label="Sold-To Party" sap:value
            -list="standard"/>
        <Property Name="ShipToPartyName" Type="Edm.String" MaxLength="80" sap:label="Name of Customer"/>
        <Property Name="ShipToParty" Type="Edm.String" MaxLength="10" sap:display-format="UpperCase" sap:text="ShipToPartyName" sap:label="Ship-To Party" sap
            :quickinfo="Customer Number"/>
        <Property Name="CityName" Type="Edm.String" MaxLength="44" sap:label="City/Country or Region"/>
        <Property Name="PurchaseOrderByCustomer" Type="Edm.String" MaxLength="35" sap:label="Customer Reference"/>
        <Property Name="TransactionCurrency" Type="Edm.String" MaxLength="5" sap:label="Document Currency" sap:quickinfo="SD document currency" sap:semantics
```

The metadata viewer also provides the ability to ease the reading of complicated sets of ODATA metadata code by adding line numbering. In addition, the user is able to close and / or (re)open sets of code in the ODATA metadata code by collapsing or expanding the clauses. This can be done by clicking on the collapse icon, located next to the row number.

The top-right hand side offers a selection of:

- The **Copy** button which will copy the ODATA view metadata code to your clipboard.
- The **Download** button which will generate the ODATA view metadata code in the. XML format available for download.

5 How To

The following chapter will provide some powerful use cases from an end-to-end perspective to demonstrate the power of S4Explorer, or to provide inspiration for end-users to achieve more with S4Explorer.

5.1 Investigating CDS Views for Consumption

Let's imagine you're an implementation consultant, and a client wants to build a dashboard which visualizes the open quotations. The client wants to use CDS views to extract the data from their S/4HANA system. A key KPI for the sales director is the net amount of open quotations. You gather the following from discussions:
Required Outcome: An overview of the quotations which have an open status.

Calculations: Total Amount of EUR of open quotations.

Key Reporting Dimensions:

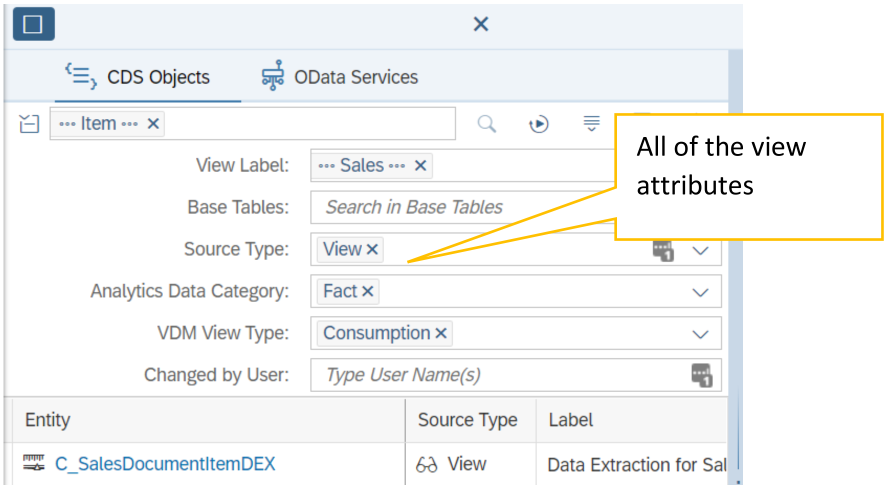
1. Sales Organization.
2. Distribution Channel.
3. Sold-To-Party.
4. Sales Quotation Date.

Your goal is to determine whether the S/4HANA content is sufficient for the development team to create the required outcome. If this is not the case, you want to provide the development team with as much information as possible to continue investigation / development on content to meet the requirements.

You open up S/4Explorer and open the CDS Objects Catalog. Using the smart search functionality, you identify you want to look at consumption level views, containing fact data, which are related to the sales item line level.

As you can see, S/4Explorer indicates that there is a consumption view with fact data which looks promising.

By clicking on the entity in the search result list you open up the CDS view for further inspection.



••• Item ••• ×

View Label: ••• Sales ••• ×

Base Tables: *Search in Base Tables*

Source Type: View ×

Analytics Data Category: Fact ×

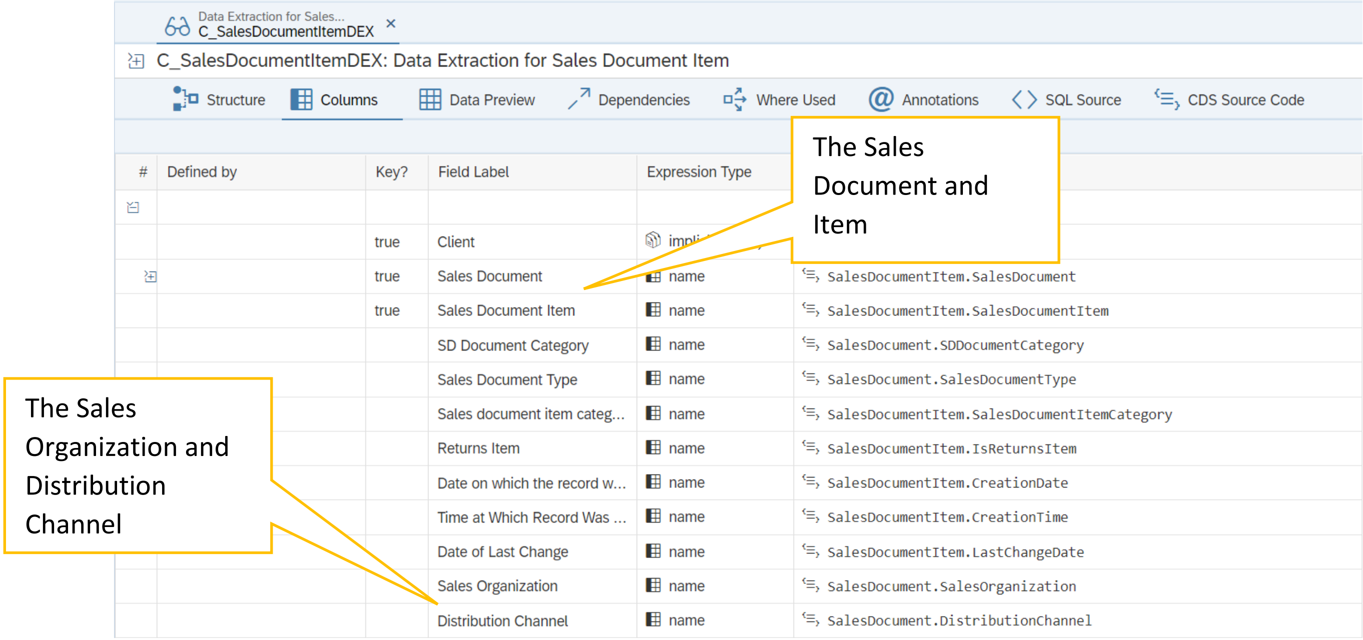
VDM View Type: Consumption ×

Changed by User: *Type User Name(s)*

Entity	Source Type	Label
C_SalesDocumentItemDEX	View	Data Extraction for Sal

All of the view attributes

In the CDS view you head to the columns table to inspect the included columns, where you find that all the required columns are included in the view. *(NetAmount is listed lower in the table)*



Data Extraction for Sales...
C_SalesDocumentItemDEX ×

C_SalesDocumentItemDEX: Data Extraction for Sales Document Item

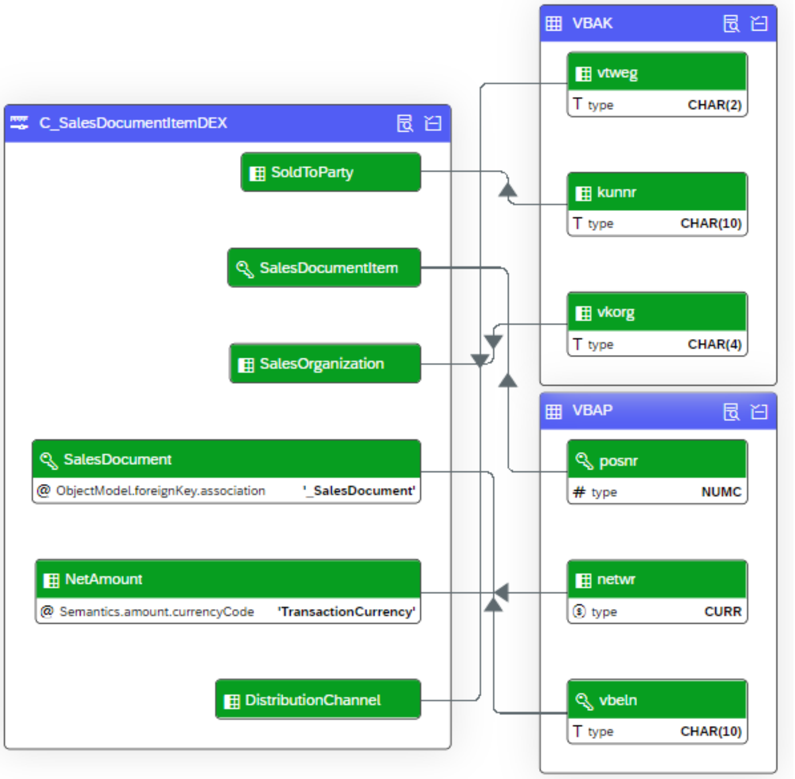
#	Defined by	Key?	Field Label	Expression Type	
		true	Client	impli	
		true	Sales Document	name	SalesDocumentItem.SalesDocument
		true	Sales Document Item	name	SalesDocumentItem.SalesDocumentItem
			SD Document Category	name	SalesDocument.SDDocumentCategory
			Sales Document Type	name	SalesDocument.SalesDocumentType
			Sales document item categ...	name	SalesDocumentItem.SalesDocumentItemCategory
			Returns Item	name	SalesDocumentItem.IsReturnsItem
			Date on which the record w...	name	SalesDocumentItem.CreationDate
			Time at Which Record Was ...	name	SalesDocumentItem.CreationTime
			Date of Last Change	name	SalesDocumentItem.LastChangeDate
			Sales Organization	name	SalesDocument.SalesOrganization
			Distribution Channel	name	SalesDocument.DistributionChannel

The Sales Document and Item

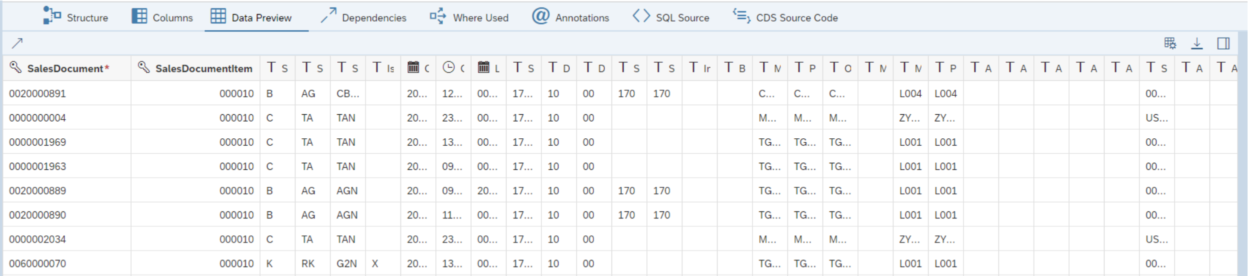
The Sales Organization and Distribution Channel

Upon further investigation of the structure, you open up the dependencies tab and analyze the flow of the required columns to this consumption level view.

This confirms that the required fields from the Sales Document Item and Header table are included in the field.



Finally, you open the data preview tab to confirm with some example quotations that the values are as expected. You have the option to inspect the records using S4Explorer. Alternatively, you can export the records to excel to do a more sophisticated analysis if needed. Indeed, the CDS view displays the NetAmount per line item of each quotation, which you can easily group to document level if needed.



SalesDocument*	SalesDocumentItem	T S	T S	T S	T Is	📅 C	🕒 (	📅 L	T S	T D	T D	T S	T S	T Ir	T B	T M	T P	T O	T M	T M	T P	T A	T A	T A	T A	T A	T S	T A	T A
0020000891	000010	B	AG	CB...		20...	12...	00...	17...	10	00	170	170			C...	C...	C...		L004	L004						00...		
0000000004	000010	C	TA	TAN		20...	23...	00...	17...	10	00					M...	M...	M...		ZY...	ZY...						US...		
0000001969	000010	C	TA	TAN		20...	13...	00...	17...	10	00					TG...	TG...	TG...		L001	L001						00...		
0000001963	000010	C	TA	TAN		20...	09...	00...	17...	10	00					TG...	TG...	TG...		L001	L001						00...		
0020000889	000010	B	AG	AGN		20...	09...	20...	17...	10	00	170	170			TG...	TG...	TG...		L001	L001						00...		
0020000890	000010	B	AG	AGN		20...	11...	00...	17...	10	00	170	170			TG...	TG...	TG...		L001	L001						00...		
0000002034	000010	C	TA	TAN		20...	23...	00...	17...	10	00					M...	M...	M...		ZY...	ZY...						US...		
0060000070	000010	K	RK	G2N	X	20...	13...	00...	17...	10	00					TG...	TG...	TG...		L001	L001						00...		

Using the detail header page, you take the ODP DataSource and provide it to the ETL team. They can now easily extract the data from the S/4HANA system, from which it can be modeled and exposed the reporting tool of your choice.

5.2 Analyzing Query Results

Let’s imagine you’re a data analyst who has been assigned the task of identifying why the Sales Profit Margin % KPI is much lower than expected. From the discussions with your manager you identify the following:

The problem: The Sales Profit Margin % is significantly lower than expected.

The Query: 2CCSDSALESVLMQ.

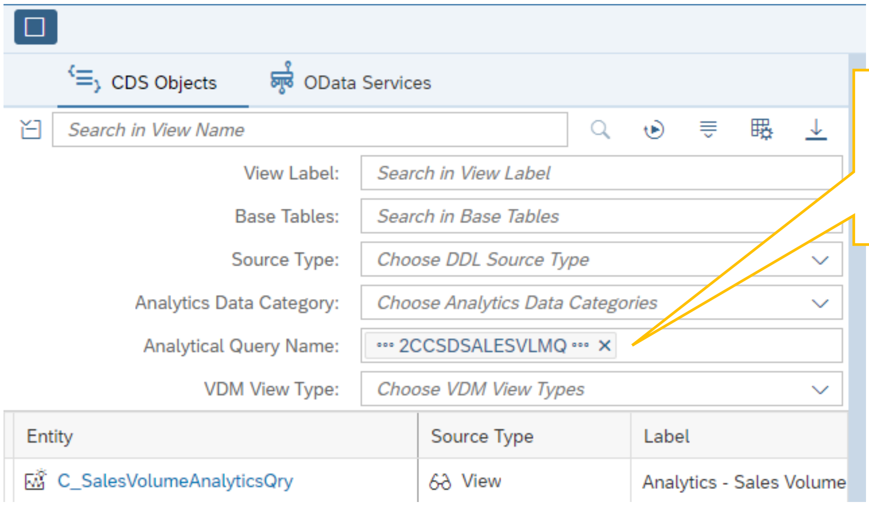
Calculation: Net Profit / Net Sales.

Key Conditions: The Net Amounts should include:

1. Invoices.
2. Debit Memos.
3. Pro Forma Invoices.

Your goal is to investigate the root cause of the KPI difference by tracking the KPI calculation and it’s components through various stages. Your outcome is to raise a detailed ticket to the support team to resolve any issues found.

You search for the analytical query that was provided by entering it in the analytical query name search box. If it’s not there by default, simply open the selection box, and select the Analytical Query Name to add it in.



Search in View Name

View Label: Search in View Label

Base Tables: Search in Base Tables

Source Type: Choose DDL Source Type

Analytics Data Category: Choose Analytics Data Categories

Analytical Query Name: 2CCSDSALESVLMQ

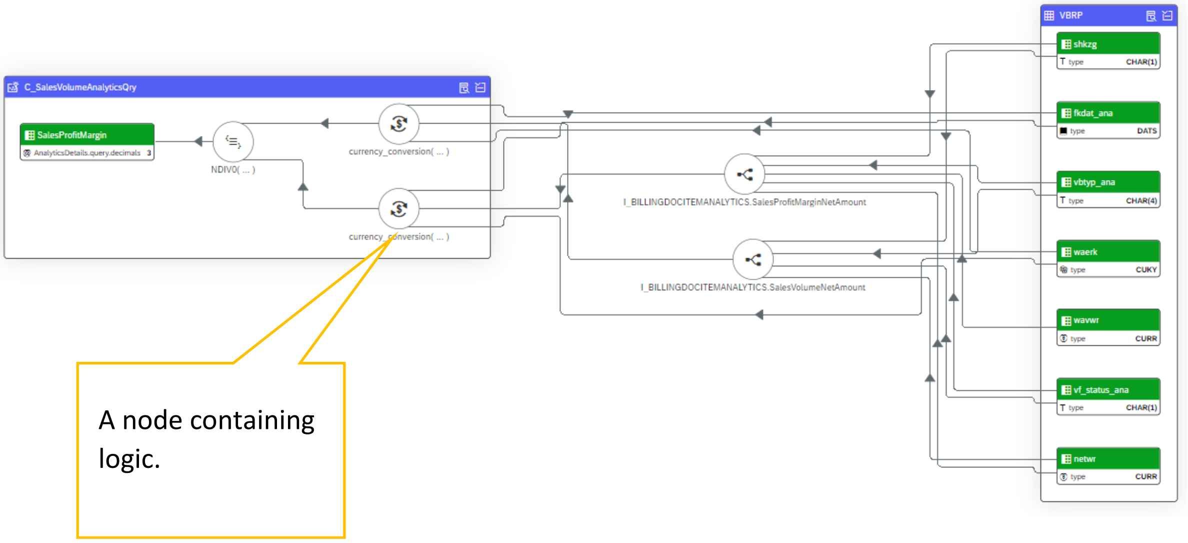
VDM View Type: Choose VDM View Types

Entity	Source Type	Label
C_SalesVolumeAnalyticsQry	View	Analytics - Sales Volume

Input the query name

By selecting the C_SalesVolumeAnalyticsQry you open the detailed view on this query. There are many ways to analyse the logic of the KPI, such as the source code or by looking at the structure of the query and identifying all of the sources. However, most efficient would be to open a lineage view on the KPI to isolate the calculations and have a direct overview.

In the dependencies tab, select lineage view and under the settings pane select the column SalesProfitMargin. Additionally, given we're interested in analysing the logic of the KPI we can select the detail level Transformations under the settings pane to show all transformations.

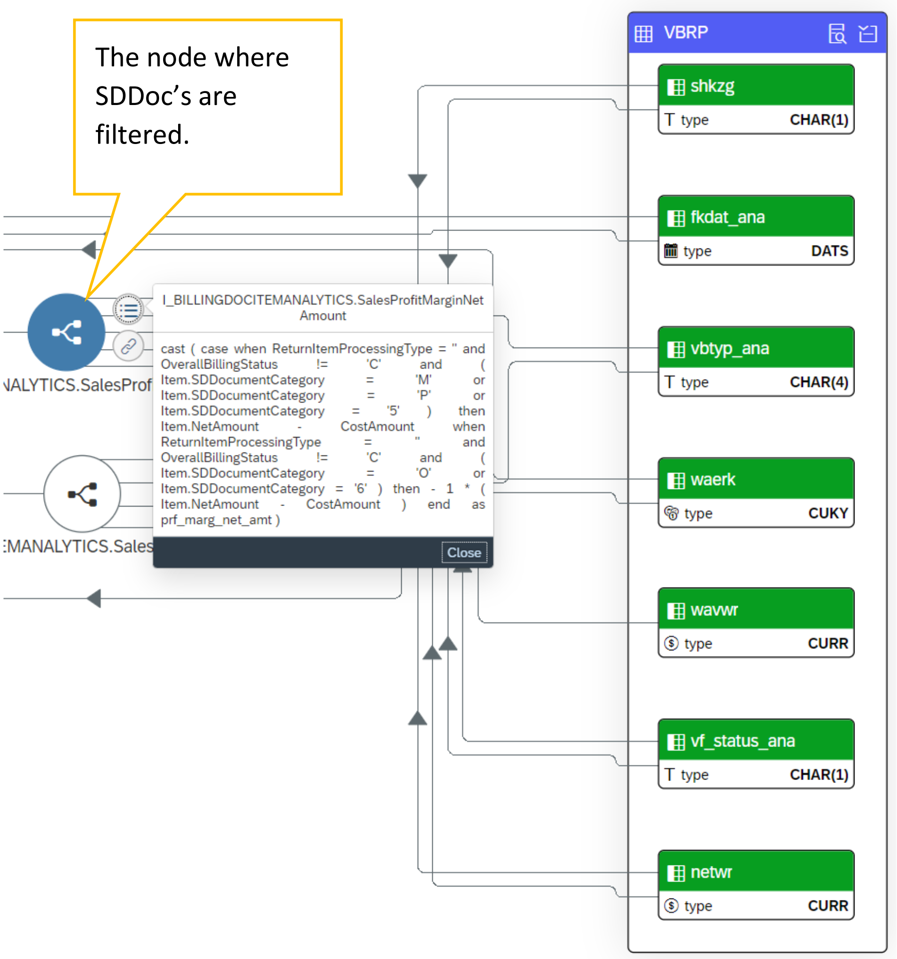

C_SalesVolumeAnalyticsQry
SalesProfitMargin
AnalyticsDetails.query.decimals 3
NDIV0(...)
currency_conversion(...)
currency_conversion(...)
I_BILLINGDOCITEMANALYTICS.SalesProfitMarginNetAmount
I_BILLINGDOCITEMANALYTICS.SalesVolumeNetAmount
VBRP
shkzg
type CHAR(1)
fkdat_ana
type DATS
vbtyp_ana
type CHAR(4)
waerk
type CUKY
wavwr
type CURR
vf_status_ana
type CHAR(1)
netwr
type CURR

A node containing logic.

Each of the nodes can be clicked on to open the specific logic that is applied. After going through the logic of the nodes we encounter the following:

This node indicates a WHERE clause on the types of sales documents that will be used in the calculation. As an SAP analyst you're probably familiar with these types. From the syntax we can derive that Pro Forma Invoices, with doc type "U" are not included in the KPI calculation.

With this knowledge you can easily send a detailed ticket to the support team indicating the location of the logic, the logic you want adjusted and what you need to have adjusted.



The node where SDDoc's are filtered.

I_BILLINGDOCITEMANALYTICS.SalesProfitMarginNetAmount

```
cast ( case when ReturnItemProcessingType = '' and
OverallBillingStatus != 'C' and (
Item.SDDocumentCategory = 'M' or
Item.SDDocumentCategory = 'P' or
Item.SDDocumentCategory = '5' ) then
Item.NetAmount - CostAmount when
ReturnItemProcessingType = '' and
OverallBillingStatus != 'C' and (
Item.SDDocumentCategory = 'O' or
Item.SDDocumentCategory = '6' ) then - 1 * (
Item.NetAmount - CostAmount ) end as
prf_marg_net_amt )
```

5.3 Identifying Dependencies

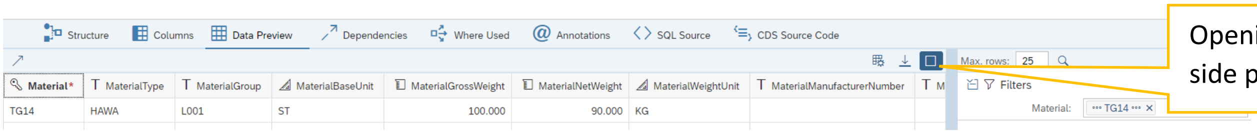
Imagine you're a master data management steward, who has been assigned the task of discontinuing material TG14. You gather the following from your assignment.

Outcome: To discontinue material TG14.

Associated Tasks: You will have to consider:

1. Which plants are currently associated with the material?
2. Your manager wants a list of existing views that currently consume the material dimension to assess the impact of the discontinuation.

From previous experience you know that you can find material master data in the I_Material CDS view. You open it up on S4Explorer and go to the data preview section to perform your analysis. By expanding the side bar you can apply a filter on the data preview, since you're only interested in the TG14 material.



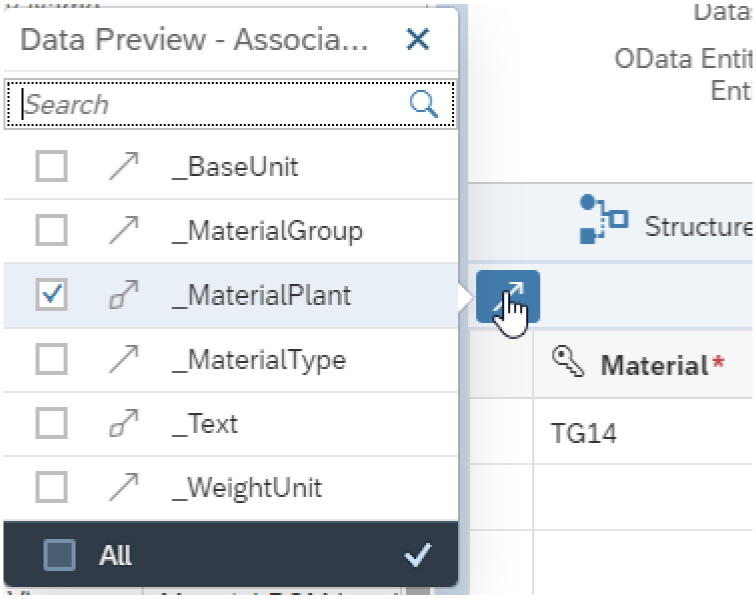
Material*	MaterialType	MaterialGroup	MaterialBaseUnit	MaterialGrossWeight	MaterialNetWeight	MaterialWeightUnit	MaterialManufacturerNumber	M
TG14	HAWA	L001	ST	100.000	90.000	KG		

Max. rows: 25

Material: TG14

Opening the side panel

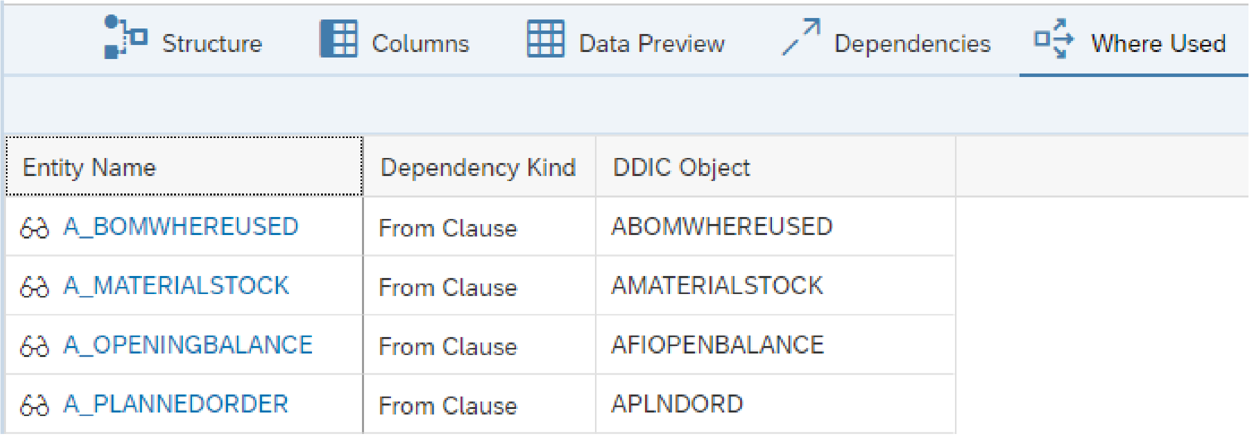
The arrow on the top left corner can be clicked to analyse the associations of a record. For example, in this case we're interested in the plants associated with this particular material. We open the context menu, select the association to MaterialPlant and confirm our choice. Underneath the data preview a new window will open which will show all associated plants for this particular material.



☐ _BaseUnit
☐ _MaterialGroup
☑ _MaterialPlant
☐ _MaterialType
☐ _Text
☐ _WeightUnit

Material*
TG14

Additionally, we want a list of views consuming the material dimension to assess the impact on the existing infrastructure. You can easily find this by opening up the Where Used page which will generate a table overview of all the entities that use the current view, the association that they have to the current view as well as the particular DDIC Object they are dependent on.



Entity Name	Dependency Kind	DDIC Object
A_BOMWHEREUSED	From Clause	ABOMWHEREUSED
A_MATERIALSTOCK	From Clause	AMATERIALSTOCK
A_OPENINGBALANCE	From Clause	AFIOPENBALANCE
A_PLANNEDORDER	From Clause	APLNDORD

Of course, you can download this overview to Excel to edit and share with your manager and perform any additional analyses if needed by using the download button located in the top right corner of the workspace. .

6 Frequently Asked Questions