



TIBCO® Data Virtualization

SAP BW Adapter Guide

Version 8.7.0 | October 2023

Contents

| | |
|---|-----------|
| Contents | 2 |
| Using SAP BW with TDV | 3 |
| About SAP BW Data Sources | 3 |
| SAP BW Basic Tab | 4 |
| SAP BW Advanced Tab | 6 |
| Reintrospecting SAP BW | 8 |
| Introspecting Large Datasets with SAP BW | 9 |
| Creating a View from a Cube | 10 |
| Generating an OLAP View Execution Plan | 14 |
| Accessing Data Lineage for an OLAP View | 14 |
| SQL Support for SAP BW | 14 |
| Security for SAP BW with TDV | 38 |
| SAP BW Global Properties | 41 |
| TIBCO Product Documentation and Support Services | 43 |
| How to Access TIBCO Documentation | 43 |
| Release Version Support | 44 |
| How to Contact TIBCO Support | 44 |
| How to Join TIBCO Community | 45 |
| Legal and Third-Party Notices | 46 |

Using SAP BW with TDV

This topic describes important usage information for SAP BW data sources.

- [About SAP BW Data Sources](#)
- [Reintrospecting SAP BW](#)
- [Introspecting Large Datasets with SAP BW](#)
- [Creating a View from a Cube](#)
- [Generating an OLAP View Execution Plan](#)
- [Accessing Data Lineage for an OLAP View](#)
- [SQL Support for SAP BW](#)
- [Security for SAP BW with TDV](#)
- [SAP BW Global Properties](#)

About SAP BW Data Sources

Requirements

- You need an SAP BW account with API-level access enabled for your organization to work with this data source type.
- For installation requirements, see the TDV Installation and Upgrade Guide.
- See the TDV User Guide Chapter “Configuring Advanced Adapters” section “Installing the SAP Java Connector Library” for more details on where to download SAP JCo connection library and how to install it.

Large data sources with complex schemas and big tables with lots of columns have a correspondingly large amount of metadata that can pose special challenges for TDV. For TDV, it is not the amount of data stored, but rather the complexity and amount of metadata that defines the data source that is the most important factor for performance, particularly during introspection. For additional information on how to use TDV when introspecting large data source, see the *TDV User Guide*.

Considerations for introspection:

- Selecting a parent node selects all of the children resources in that node.
- Resources are organized by InfoArea as in BEx Query Analyzer. InfoProviders are further organized into three subfolders: Hierarchies, InfoProviders, and Queries.
- Introspecting resources is time-intensive, so it is best to select only the resources necessary for your project. If you find you need additional resources later, you can always add them through the Add/Remove Resources in Studio.
- Avoid using the Find field during introspection of SAP BW data sources. TDV cannot load all of the SAPBW metadata before introspection. Attempts to use the introspection Find feature can result in SAP BW database service interruptions.

Application Views for SAP

The organization of the views and folders mirrors that of SAP's Business Object Repository (BOR). Field names are aliased to provide human-readable names. In some cases several objects in BOR are synthesized in a single query to produce a more detailed, unified view of the data.

A common pattern employed in application views is that of List and Details. Views with names ending in "List" return a minimal set of columns that serve to uniquely identify an object. The identifiers are passed as arguments into a corresponding view, its name ending in "Details," to produce a more extensive set of columns.

Filter Data from Application Views

There are two ways to filter data from SAP using application views: using a filter provided by SAP itself, or by filtering within TDV. Both methods are used by application views depending on the view and the capabilities of SAP.

SAP BW Basic Tab

| Field | Description |
|-------------------------|---|
| Application Server | Name of the machine hosting SAP BW or the host machine's IP address. For load-balanced configurations of SAP BW, leave this property empty. |
| SAP Router String | Routing entry. If an SAP Router is used to connect to the Application Server, enter its routing entry here. For example: <code>/H/saprouter/H/194.117.106.130/S/3297/H/</code> |
| System Number | Two-digit gateway service number of the SAP BW system. |
| Client | Three-digit client number of the SAP BW system. |
| User and Password | Valid user name and password to SAP BW. |
| Save Password check box | <p>This option works in combination with the Pass-through Login option. By default, this option is disabled and un-editable. It becomes editable when you select the Pass-through Login option.</p> <p>If you accept the default, the password is saved and the Pass-through Login option remains disabled. Then, you can perform the following operations without having to supply the password again:</p> <ul style="list-style-type: none"> • Introspect the current data source. • Reintrospect the data source. • Add/Remove data source resources. • Perform query/update/insert operations on a table in the data source. • Invoke a stored procedure. <p>Refresh a cached view based on the data source resources.</p> |
| Pass-through Login | Works in combination with the Save Password option. By default, this mode is Disabled and the password is saved. This mode is referred to as non-pass-through mode. Refer to the details given for the Save Password option. If you select the Enabled option, the Save Password option becomes editable. This mode is referred to as pass-through mode. |

The operations you can and cannot perform in **pass-through mode** depend on if Save Password is checked as follows:

| Save password? | Operations you can perform | Operations you cannot perform |
|----------------|---|-------------------------------|
| Yes | Introspection. You do not have to resupply the password. | N/A |
| No | <ul style="list-style-type: none"> Query/update/insert/delete operations. You need to resupply the original login credentials for the current session. Reintrospection, Add/Remove data source resources. You will be prompted to resupply the password that was used when the data source was originally introspected. | Schedule reintrospection. |

SAP BW Advanced Tab

Some properties refer directly to the configuration of the SAP BW Server and must be provided by an SAP BW Administrator. Other properties are specific to TDV and how it interacts with SAP BW.

| Option | Description |
|----------------------------------|---|
| Maximum connections in pool | SAP JCo parameter specifying the maximum number of simultaneous connections in an SAP BW connection pool. |
| Maximum idle connections in pool | SAP JCo parameter specifying the maximum number of simultaneous idle connections, by destination, to keep open in an SAP connection pool. Set to 0 to disable connection pooling, closing connections after each request. |
| Maximum rows | Maximum number of rows to return from an SAP BW query. Default: |

| Option | Description |
|--|--|
| | <p>250,000. This setting prevents the return of datasets that are too large. To return all rows from the query, set this property to zero. This maximum row count is cumulative for all nodes, so queries or data sources using the “leaf nodes only” mode might receive significantly fewer rows than the maximum rows setting.</p> <p>This setting only works on rows being returned from DataStore Objects.</p> |
| Show leaf nodes only check box | Determine how TDV handles hierarchical data from SAP BW. If unchecked, all nodes of the data hierarchy are returned. If checked, only members of dimensions which do not contain children are returned, and summary data is suppressed. Default: checked. |
| Include children of selected members check box | Display the children of the introspected components in the resource tree. |
| Allow large (>1M cells) datasets check box | Lets you introspect large datasets (those with more than 1M cells). Default: unchecked. |
| Display Descriptive Names | <p>Display only descriptive names in the resource tree and in the Introspection dialog box. (Technical names still appear in the popup when the cursor hovers over the resource name, and on the resource’s Info tab.)</p> <p>Use this radio button consistently for any given SAP BW data source; otherwise, results can be unpredictable.</p> |
| Display Technical and Descriptive Names | Display both technical names and descriptive names in the resource tree and in the Introspection dialog box. |
| Display Technical and Descriptive Names, Use Technical Names | Display technical names in the resource tree, and both the technical names and the descriptive names in the Introspection dialog box. |

| Option | Description |
|---------------------------|-------------|
| In TDV Resource Namespace | |

Reintrospecting SAP BW

Because each SAP BW data source contains resources that are unique to it, it is best to follow a few special steps when you reintrospect to make sure that you can see all expected metadata within Studio.

If you removed any SAP BW resources from the Studio view of the SAP BW data source metadata, reintrospection restores those resources to the Studio resource tree.

Note: If you reintrospect an SAP BW data source, keep the descriptive/technical name choice originally used for that data source. If you change it, results are unpredictable.

To reintrospect SAP BW

From Studio, open the SAP BW data source.

1. Click ADD/Remove Resources.
2. Select one or more schemas that you want to introspect or reintrospect.
Avoid using the Find field during introspection of SAP BW data sources. TDV cannot load all of the SAPBW metadata before introspection. Attempts to use the introspection Find feature can result in SAP BW database service interruptions.
3. Select the following check boxes:
 - Re-Introspect previously introspected resources
 - Allow partial introspection, omitting resources with errors
 - Detect New Resources During Re-Introspection
4. Click Next.
5. Review the summary.
6. Click Finish to begin reintrospection.
7. When the status message indicates that reintrospection was successful or completed, click OK to close the dialog.

Introspecting Attributes with Duplicate Descriptive Names

To include attributes with duplicate descriptive names while introspecting the datasource: Open this properties file using your favorite text editor:

```
<TDV_install_dir>\apps\dml\app_ds_sapbw\conf\product.properties
```

8. Add/edit the following property:

```
sapbw.introspector.includeKeysInColumnNames=true
```

Setting this value prepends the technical name to the descriptive names.

9. Restart TDV Server.
10. In your data source properties, select the advanced introspection option, "Display Technical and Descriptive Names, Use Technical Names in TDV Resource Namespace".
11. Reintrospect the data source, or introspect a new one.

Introspecting Large Datasets with SAP BW

Large data sources with complex schemas, big tables, lots of columns, or many dimensions have a correspondingly large amount of metadata that can pose special challenges for TDV. For TDV, it is not the amount of data stored, but rather the complexity and amount of metadata that defines the data source that is the most important factor for performance, particularly during introspection.

To work with large datasets and prevent runaway MDX queries when using SAP BW within TDV

When defining your SAP BW data source, on the Advanced tab, set the Allow large (>1M cells) datasets check box. For other data source definition information, see [About SAP BW Data Sources](#).

This specifies that large datasets that are greater than 1M cells can be introspected. By default, datasets that are larger than one million cells cannot be introspected.

1. Define an OLAP View, following the instructions in [Creating a View from a Cube](#).
2. For your OLAP View, generate the Execution Plan. See [Generating an OLAP View Execution Plan](#).
3. Review the Estimated Rows Returned value that is displayed for the Execution Plan.

4. Review the other information displayed in the execution plan. If TDV estimates that your query could return more than one million cells of data from the SAP BW database, the FETCH node of the Cardinality Statistics panel displays a message indicating this and suggesting that you define some constraints on the query to limit the data set to return.
5. Determine if limiting the number of dimensions used for the OLAP View could improve retrieval of the dataset. See [Creating an OLAP View](#).
6. Determine if adding filter conditions would help improve performance. See [Creating Filter Conditions on an OLAP View](#).
7. Save your work and run another Execution Plan for comparison.
8. Set up and use a test system to determine if the performance of the OLAP View is acceptable.

Creating a View from a Cube

Studio provides a way to create views from multidimensional cubes. The Studio OLAP View editor lets you graphically design and display cube slices. Studio introspects and displays the available dimensions, measures, hierarchies, and members of a selected cube to allow graphical selection of the parameters that create basic SQL queries. Basic queries are converted into MDX to retrieve cube data into a table cursor for viewing and use.

After you have the OLAP cube data in a simplified table view, the data can be processed further by procedure, by triggers, or by inclusion of the OLAP View into other views to process or cache data that might be difficult to handle or slow to retrieve using direct queries to the SAP BW source.

The Info tab of an OLAP View reveals the full and formal resource name, type, owner, and lock information.

This topic includes:

- [Creating an OLAP View](#)
- [Creating Filter Conditions on an OLAP View](#)
- [Using the OLAP View Editor SQL Tab](#)

Creating an OLAP View

The Studio OLAP View editor lets you graphically design and display cube slices. This editor effectively flattens a cube into a two-dimensional object that can be used by relational database tools.

To create an OLAP view

Make sure you have the SAP BW Data Adapter installed following the instructions in the TDV Installation and Upgrade Guide.

1. Create and introspect the SAP BW data source.
2. In the Studio resource tree, select a location, right-click, and choose New OLAP View.
3. Open an SAP BW data source and select the cube on which to base the view.
4. Name your new cube view and click OK to proceed.

Note: Uninitialized or unconfigured SAP BW cube resources return detailed error messages that might report various BAPI misconfiguration states. Studio introspection of misconfigured SAP BW cubes passes these data source configuration errors to the Studio user for review. The error messages should be forwarded to the SAP BW resource owner. Further configuration of the SAP BW cube might be necessary if that resource is to be used.

The OLAP View editor is opened and displayed in Studio. The TDV-defined resource name of the cube that you selected appears next to the Associated OLAP Cube label.

The Select tab displays available dimensions and measures of the associated OLAP view.

5. Select the Available Dimensions that you want in your flattened view and use the arrows to move them into the Selected Dimensions field.

Use restraint in the selection of dimensions. Every dimension in the SELECT is cross-joined, making it possible to return millions of cells and cause excessive memory usage in shared libraries. For example, SELECT * is not recommended on SAP BW resources.

6. Select the Available Measures that you want in your flattened view and use the arrows to move them into the Selected Measures field.

Both dimensions and rows appear as table columns, with dimensions displayed, followed by measures.

7. Execute the query created by your selections by clicking the Execute button. The first 50 rows of results appear.

8. (Optional) Change the order in which the selected dimensions and measures are requested from the data source using the up- and down-arrows.

The results of query execution shown in the Result tab persist even after other changes are made to selections.

9. Execute the query again to see the most recent changes.
10. Save the view to make changes permanent.

Creating Filter Conditions on an OLAP View

Filter conditions for an OLAP view created from an SAP BW cube can be set using the Where tab that is part of the OLAP View editor. You can use filters to restrict the result set to cross-linked dimensional rows that match the selected dimension hierarchy members.

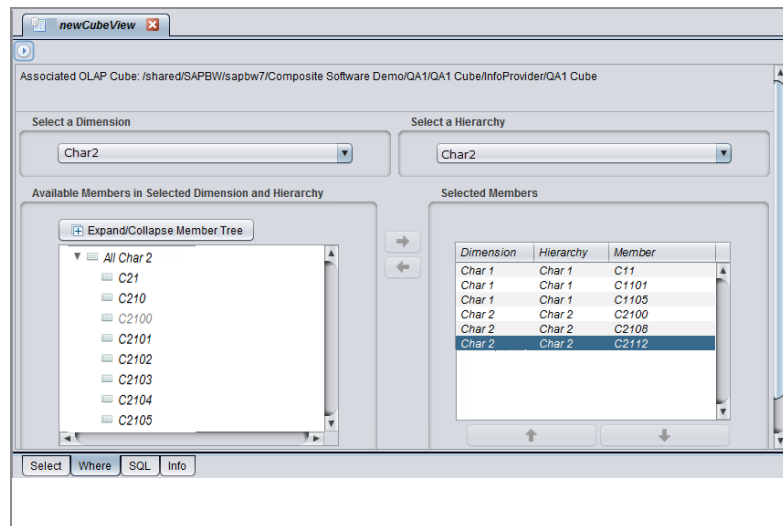
Member selection creates a WHERE filter condition, and all returned rows must have members equal to the selections. All additional selected members add a filter by means of an AND in the WHERE clause. All rows returned by multiple selected member filters must satisfy all filter conditions.

To set filter conditions on an OLAP view

Open a view that you have created from an SAP BW cube.

11. Select the Where tab.

The Where tab displays the available dimensions in the associated OLAP view.



12. Select a dimension and/or a hierarchy from the list.

In SAP BW cube views, selecting a dimension displays its hierarchy levels.

13. Use the Expand/Collapse Member Tree button to query the cube for available members.

For very large cubes, the round-trip to the data source can take more than a minute.

14. Select a member by clicking it and moving it to the Selected Members field.

Member selection creates a WHERE filter condition, and all returned rows must have members equal to the selections. All additional selected members add a filter by means of an AND in the WHERE clause. All rows returned by multiple selected member filters must satisfy all filter conditions.

15. Save the OLAP View.

If the resource indicates it is impacted, some unconfigured state or error has occurred, in which case you should check your View.

Selecting dimensions, measures, hierarchies, and members immediately generates the SQL associated with such selections.

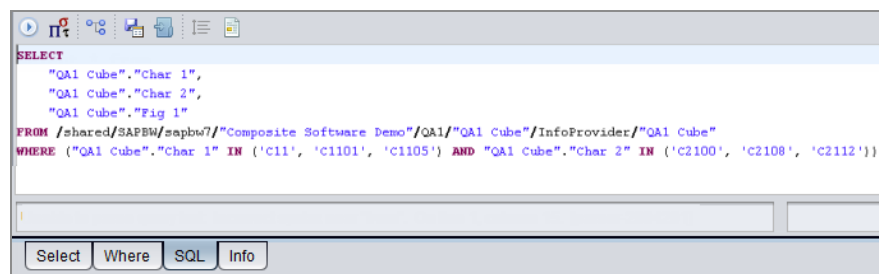
Using the OLAP View Editor SQL Tab

The SQL tab of the OLAP View editor displays the SQL that will be run against the SAP BW data source to retrieve data. You can use the tab to review how selections on the other editor tabs have been interpreted and you can use the tab to edit the SQL query directly.

To use the SQL tab

Open a view that you have created from an SAP BW cube.

16. Select the SQL tab.



17. Inspect the SQL generated by your selections at any time by opening the SQL tab.

18. Edit the SQL displayed in the SQL tab.

Note: Directly editing the SQL disables the Select and Where tabs of the OLAP View editor.

Generating an OLAP View Execution Plan

Execution plans are important tools when designing resources to be consumed by others, especially if performance is a high priority.

To generate and OLAP View Execution Plan

Open a view that you have created from an SAP BW cube.

1. Select the SQL tab.
2. Click Show Execution Plan on the editor toolbar.
3. Review the Execution Plan Node and Field Contents. Descriptions for how to interpret much of this information can be found in the TDV User Guide.
4. Review the Estimated Rows Returned value.

Accessing Data Lineage for an OLAP View

Data lineage can be use to determine where metadata originates from. This can help you track how changes to certain data sources can have effects on data that clients are consuming.

To generate and OLAP View Execution Plan

Open a view that you have created from an SAP BW cube.

1. Select the SQL tab.
2. Click the Show Lineage Panel toolbar button on the editor toolbar.

The Lineage panel opens in the lower section of the view's editor and displays all the resources involved with the view in a graphical format. Descriptions for how to interpret much of this information can be found in the *TDV User Guide*.

SQL Support for SAP BW

This section describes SQL support for SAP BW and its resource types—functions, tables, and ABAP queries.

- [Understanding SAP BW with TDV](#)
- [Introspection Resources](#)
- [Capabilities](#)
- [Query Mapping](#)
- [Multidimensional Queries](#)
- [Joins and Query Performance](#)

Understanding SAP BW with TDV

TDV has made the interface between TDV and SAP BW as efficient and accurate as possible. However, due to some product limitations and proprietary interfaces, TDV and SAP BW results might differ, and performance issues might occur when accessing SAP BW data using TDV. The following sections provide additional information.

- [Differences in Queries and Results](#)
- [Performance Considerations](#)

Differences in Queries and Results

Queries in SAP BW using BEx are created differently in TDV. In TDV, SQL queries generate MDX. The process is different in SAP BW using BEx. This disparity can cause differences in results. Workarounds are described in this section. See [Introspection Resources](#) for information about how TDV interprets SAP BW resources in TDV.

Queries with Input Values

Queries can require input values. To handle queries with input values in TDV, set the values using a WHERE clause and provide values for columns that begin with an underscore (_). For an example, see [Variables](#).

Queries with Filters

BEx queries that contain filters might not return the same results in TDV as they do in BEx. To apply the filters in TDV, recreate them using a WHERE clause. This limitation is imposed by SAP BW; TDV does not have access to the filters.

Additional Columns in Query Results

BEx queries do not return the same columns in TDV as they do in BEx. For example, you might see 1,000 columns in TDV listing every property of every dimension in the query, even if you did not include those properties in BEx. This is because the SAP BW API does not give TDV access to all of the items selected for output in the query. TDV only gets a list of InfoObjects. From there, TDV expands each one to include all of its columns.

Performance Considerations

Performance of TDV's SAP BW Adapter is slower and more resource-intensive on the SAP BW server than accessing SAP BW through native SAP tools, because SAP does not give TDV access to proprietary APIs.

When performance issues occur when using TDV with SAP BW, you can reproduce them outside of TDV by copying the MDX statements generated by TDV (by examining the TDV cs_server.log) and entering them into the MDX test environment (transaction MDXTEST in SAP GUI). This allows SAP BW administrators to optimize settings and improve performance of the MDX statements.

Introspection Resources

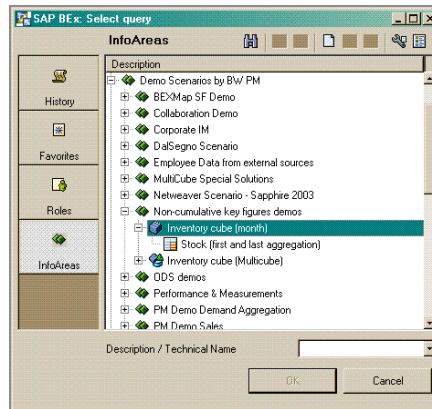
This section describes the details of introspecting SAP BW resources.

- [Resource Hierarchy in SAP BW](#)
- [Resource Hierarchy in TDV](#)
- [Metadata Mapping, Multi-Dimensional](#)
- [Metadata Mapping, ODS](#)

Resource Hierarchy in SAP BW

The organization of SAP BW resources in TDV mirrors SAP BW as much as possible. InfoAreas in SAP BW become folders and subfolders in TDV. Supported InfoObjects are organized into subfolders.

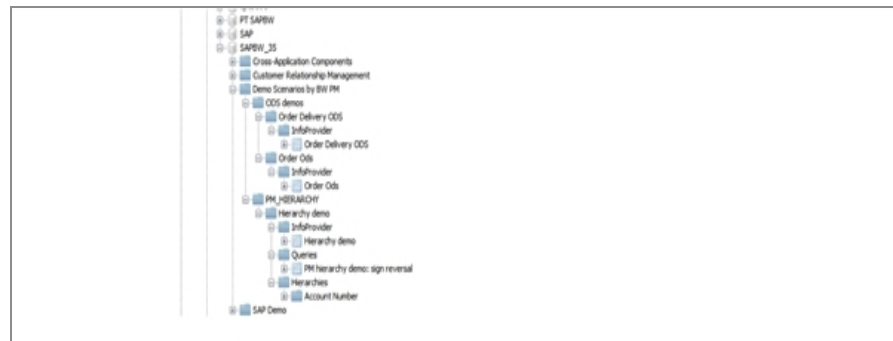
In the following screenshot of BeX Analyzer, nodes have been expanded in InfoAreas to reveal the InfoProvider Inventory cube (month) which contains the Query Stock (first and last aggregation).



The next section shows how the equivalent resources look in TDV.

Resource Hierarchy in TDV

As with BEx Analyzer, TDV displays only the objects that the user is authorized to see.



InfoAreas in BEx Analyzer are displayed as folders in Studio, and these folders are named after the InfoArea's technical name or description. Moving the mouse over an item reveals a tool-tip with its technical name and type.

InfoProviders contain up to three subfolders to group resources by type, as described below.

- **InfoProvider:** This folder contains a single resource, the InfoProvider itself. This resource provides direct access to the InfoProvider without requiring construction of a Query.
- **Queries:** This folder contains all of the Query objects built from the InfoProvider. Query objects, also known as Query Cubes, are views into an InfoProvider created in BEx Analyzer.

- Hierarchies: This folder contains subfolders for each Basic Characteristic referenced within the InfoProvider. Each Characteristic contains at least one Hierarchy resource. ODS Objects do not contain the Hierarchies folder.

Metadata Mapping, Multi-Dimensional

The data structures present in multi-dimensional InfoProvider, Query, or Hierarchy objects are transformed into a flat namespace of columns in TDV. InfoProviders and Queries contain three types of data relevant to TDV: Characteristics, Attributes, and Key Figures. Queries may contain Variables as well. Hierarchies contain only Characteristics and Attributes. The next screen shot illustrates the column types in a typical Query.

| Name | Type / Reference | Native Type | |
|---|------------------|-----------------------|----------------|
| Account Number | ab VARCHAR(60) | [0ACCOUNT] | Characteristic |
| Account Number [Key] | ab VARCHAR(64) | [0ACCOUNT] | |
| Account Number [Level] | 1 TINYINT | [0ACCOUNT] | |
| Account Number [Hierarchy] | ab VARCHAR(64) | [0ACCOUNT] | |
| Account Number Key | ab VARCHAR(255) | [20ACCOUNT] | Attribute |
| Account Number Name | ab VARCHAR(20) | [10AGGOUNT] | |
| Account Number Medium Name | ab VARCHAR(40) | [50ACCOUNT] | |
| Account Number Balance Sheet Acct (Key) | ab VARCHAR(255) | [20BAL_FLAG] | |
| Account Number Ind.: Cost Element (Key) | ab VARCHAR(255) | [20CSTEL_FLAG] | |
| Account Number G/L account (Key) | ab VARCHAR(255) | [20GLACC_FLAG] | |
| Account Number Source System (Key) | ab VARCHAR(255) | [20LOGSYS] | |
| Account Number Source System (Medium Name) | ab VARCHAR(40) | [50LOGSYS] | |
| Account Number Planning Item (Key) | ab VARCHAR(255) | [20SEM_POSIT] | |
| Account Number Planning Item (Name) | ab VARCHAR(20) | [10SEM_POSIT] | |
| Account Number Planning Item (Medium Name) | ab VARCHAR(40) | [50SEM_POSIT] | |
| Account Number Planning Item (Long Name) | ab VARCHAR(60) | [40SEM_POSIT] | |
| Chart of accounts | ab VARCHAR(60) | [0CHRT_ACCTS] | |
| Chart of accounts [Key] | ab VARCHAR(64) | [0CHRT_ACCTS] | |
| Chart of accounts [Level] | 1 TINYINT | [0CHRT_ACCTS] | |
| Chart of accounts [Hierarchy] | ab VARCHAR(64) | [0CHRT_ACCTS] | |
| Chart of accounts Key | ab VARCHAR(255) | [20CHRT_ACCTS] | |
| Chart of accounts Long Name | ab VARCHAR(60) | [40CHRT_ACCTS] | |
| Chart of accounts Source System (Key) | ab VARCHAR(255) | [20LOGSYS] | |
| Chart of accounts Source System (Medium Name) | ab VARCHAR(40) | [50LOGSYS] | |
| Calendar Year | ab VARCHAR(60) | [0CALYEAR] | |
| Calendar Year [Key] | ab VARCHAR(64) | [0CALYEAR] | |
| Calendar Year [Level] | 1 TINYINT | [0CALYEAR] | |
| Calendar Year [Hierarchy] | ab VARCHAR(64) | [0CALYEAR] | |
| Calendar Year Key | ab VARCHAR(255) | [20CALYEAR] | |
| Currency | ab VARCHAR(60) | [0CURRENCY] | |
| Currency [Key] | ab VARCHAR(64) | [0CURRENCY] | |
| Currency [Level] | 1 TINYINT | [0CURRENCY] | |
| Currency [Hierarchy] | ab VARCHAR(64) | [0CURRENCY] | |
| Currency Key | ab VARCHAR(255) | [20CURRENCY] | |
| Currency Name | ab VARCHAR(20) | [10CURRENCY] | |
| Currency Medium Name | ab VARCHAR(40) | [50CURRENCY] | |
| Cumulative Balance | ab VARCHAR(128) | [Measures].[0BALANCE] | Key Figure |
| Cumulative Balance * | 10 DOUBLE | [Measures].[0BALANCE] | |
| _rownum | 1 BIGINT | Row number | Internal |
| _options | ab VARCHAR(128) | Query options | |
| _mdx | ab VARCHAR(4096) | MDX statement | |

Characteristics

Each Characteristic is represented by four columns containing Characteristic metadata, differentiated by naming conventions:

- **Characteristic**—This column's name is just the Characteristic's short description. It contains the member caption. The native type is the technical name of the Characteristic.
- **Characteristic [Key]**—The key column contains the member name, without square brackets. The native type is the technical name of the Characteristic.
- **Characteristic [Level]**—The level column contains the level of the current member, where zero is the root of the hierarchy. The native type is the technical name of the Characteristic.
- **Characteristic [Hierarchy]**—The hierarchy column contains the hierarchy in which the current member belongs (64 character VARCHAR). The native type is the technical name of the Characteristic.

Attributes

Following the Characteristic columns are Characteristic Attributes. Each Attribute is named using the Characteristic short description, a space, and the Attribute description. Its native type is the technical name of the Attribute. Its data type is VARCHAR, with the length determined by SAP BW metadata, defaulting to 255 characters if metadata is not available.

Key Figures

Each Key Figure is represented as two columns. One column contains the formatted value; the other contains the unformatted, numeric value. Both columns are named using the Key Figure name, but the numeric value field has an asterisk (*) suffix to differentiate it. The native type is the technical name of the Key Figure. The data type of the formatted Key Figure is a 128 character VARCHAR. The data type of the unformatted value is DOUBLE.

Variables

Variables only appear in Query objects. They can be identified by the single underscore before their names. The data type is VARCHAR of length determined by SAP BW metadata. The native type is the technical name of the variable.

Internal Columns

Internal columns are created for use within TDV SQL and do not correspond to objects in SAP BW. Their names are prefaced with two underscores, and they appear at the end of the

list of columns. Each Internal column has a unique function, as discussed in [Query Mapping](#).

Metadata Mapping, ODS

The InfoObjects in an ODS InfoProvider are transformed into a set of columns in TDV.

Each InfoObject becomes a column, named using the InfoObject's short text. The native type is a concatenation of the InfoObject type and its SAP data type, for example "CHA C" is a Characteristic with SAP data type C. InfoObject types are listed below:

CHA—Characteristic

DPA—Data packet characteristic

KYF—Key figure

TIM—Time characteristic

UNI—Unit of measurement

The data type of the column is determined by its SAP data type. The supported SAP data types and their corresponding TDV data types are:

C—VARCHAR

D—DATE

F—DOUBLE

g—VARCHAR

I—BIGINT

N—VARCHAR

P—DECIMAL, NUMERIC

T—TIME

X—VARCHAR

Capabilities

Capabilities characterize the features and limitations of data sources. For example, an Oracle data source can execute subqueries, while SAP BW cannot. Capabilities are consulted when a query is processed so that data sources receive only the query processing work they support; otherwise, TDV performs the work itself.

This section includes the following topics:

- [MDX Capabilities](#)
- [ODS Capabilities](#)

MDX Capabilities

The following table lists commonly used capabilities and whether they are supported in queries against SAP BW. Pushed indicates whether query processing can be passed to SAP BW. For efficient queries, minimize use of non-push capabilities.

| Capability | Supported | Pushed | Notes |
|---------------------|----------------------------|--------|---|
| Filter | Yes, with special usage | Yes | Filters are mapped to MDX as specified in Query Mapping . |
| Filter-IN | Yes, with special behavior | Yes | See Query Mapping |
| Functions-CAST | Yes | Yes | |
| ORDER BY | Yes | Yes | Filters are mapped to MDX as specified in Query Mapping . |
| CASE | Yes | No | |
| DISTINCT | Yes | No | |
| Filter-BETWEEN | Yes, with special usage. | No | Filters are mapped to MDX as specified in Query Mapping |
| Functions-aggregate | Yes | No | |
| Functions-others | Yes | No | |
| GROUP BY | Yes | No | |
| Join | Yes | No | See Joins and Query Performance . |

| Capability | Supported | Pushed | Notes |
|--------------|-----------|--------|-------|
| Subquery | Yes | No | |
| UNION | Yes | No | |
| DELETE | No | No | |
| Filter-LIKE | No | No | |
| INSERT | No | No | |
| Transactions | No | No | |
| UPDATE | No | No | |

ODS Capabilities

The following table lists capabilities that apply to ODS Objects, whether they are supported against ODS Objects, and whether query processing can be pushed to SAP BW. For efficient queries, minimize use of non-push capabilities. Capabilities for ODS objects are:

| Capability | Supported | Pushed | Notes |
|------------|-----------|--------|---|
| CASE | Yes | No | |
| DELETE | No | No | |
| DISTINCT | Yes | No | |
| Filter | Yes | Yes | All conditional operators are supported. The keywords OR, AND, and grouping of terms with parentheses have no impact on the query other than to indicate the list of filters. Filters containing NULL literals are ignored. |

| Capability | Supported | Pushed | Notes |
|---------------------|-----------|--------|--|
| Filter—BETWEEN | Yes | Yes | |
| Filter—IN | Yes | Yes | See notes on “Filter—BETWEEN” above. |
| Filter—LIKE | Yes | No | |
| Functions—aggregate | Yes | No | |
| Functions—CAST | Yes | Yes | |
| Functions—others | Yes | No | |
| GROUP BY | Yes | No | |
| INSERT | No | No | |
| Join | Yes | No | See Joins and Query Performance . |
| ORDER BY | Yes | Yes | Order is always ascending due to limitations of the SAP BAPI. ASC and DESC keywords are ignored. |
| Subquery | Yes | No | |
| Transactions | No | No | |
| UNION | Yes | No | |
| UPDATE | No | No | |

Query Mapping

This section describes how queries are mapped from SQL to MDX. This mapping is critical to content and performance.

- [Characteristics](#)
- [Attributes](#)

- [Key Figures](#)
- [Ordering](#)
- [Slicers](#)
- [Variables](#)
- [Custom MDX](#)
- [Query Options](#)
- [Row Number](#)
- [Time-Dependent Queries](#)
- [Enabling Tracing](#)
- [Working with Large Batch Sizes](#)

Characteristics

Characteristics selected in SQL are projected onto the Query Axis ROWS (Axis 1). Multiple Characteristics are cross-joined, resulting in a Cartesian product of all selected Characteristics. This creates a rows-and-columns data set for TDV. Selecting many Characteristics could result in long-running queries and huge result sets.

A Characteristic projected without a filter adds its MEMBERS set to the cross-join. If the active Hierarchy contains more than two levels, the MDX DRILLDOWNLEVEL function is applied to produce a set containing all nodes of the Hierarchy. Additionally, the MDX HIERARCHIZE function is applied to order the members in their hierarchical order.

To restrict the Characteristic to specific members, apply a filter (WHERE clause) to the Characteristic or the Characteristic [Key] column. For example, the following query produces a cross-join of Company code, Division, and Sales organization, and restricts Sales organization to the one with key 2200:

```
SELECT
```

```
"Company code", "Division", "Sales organization", "Costs in Document  
currency (SAP Demo)", "Billed Quantity (SAP Demo)"
```

```
FROM
```

```
"SAP DemoCube"
```

```
WHERE
```

```
"Sales organization [Key]" = '2200'
```

If a member key cannot be found in the Hierarchy, a warning is logged but the query continues. The log is in the file `cs_server.log`, located in the subdirectory logs of `<TDV_install_dir>`.

If you do not know the member key, you can use the member text, also known as the “caption,” but note that member texts may not be unique. To avoid ambiguous queries, it is best to filter on member keys.

The following query accomplishes the same effect as the query above, but refers to Sales organization 2200 by its text designation (Paris):

```
SELECT
```

```
"Company code", "Division", "Sales organization", "Costs in Document  
currency (SAP Demo)", "Billed Quantity (SAP Demo)"
```

```
FROM
```

```
"SAP DemoCube"
```

```
WHERE
```

```
"Sales organization" = 'Paris'
```

Characteristics have the following functionality:

- All conditional operators and the BETWEEN keyword are supported. Operators other than equals (=) and not equals (<>) have special meaning for ranges, discussed next.
- To create a range from the natural order of the hierarchy, use the greater than (>) or less than (<) symbol.

- Ranges can be specified only on the Characteristic [Key] column.
- The keywords OR, AND, IN, and grouping of terms with parentheses have no impact on the query other than to indicate the list of members.
- Members can be excluded by using the not equals (<>) operator or the NOT keyword. These map to the EXCEPT set expression.
- If no members can be found for a member text, an exception is thrown.
- Filters are not allowed on the column Characteristic [Level] and cause an exception.
- Hierarchy filters are supported. To specify a non-default Hierarchy, filter on the Characteristic [Hierarchy] column. All operators are treated as equals (=). The filter value is a Hierarchy key (enclosed in square brackets) or name. Values that cannot be resolved to a Hierarchy, and multiple Hierarchy filters on the same Characteristic, cause an exception.

Attributes

Attributes, also known as Dimension Properties, typically contain data related to the Characteristic. For example, the Characteristic “Company Code” contains an attribute named “Country (Name).”

Filter behavior for Attributes is similar to that of ordinary SQL queries:

- All conditional operators are supported, plus the BETWEEN keyword.
- The keywords OR, AND, and NOT, and grouping of terms using parentheses, are supported.
- The IN keyword is supported.
- IS NULL is mapped to test for empty strings.

Unlike in MDX, Attributes can be selected without their parent Characteristic. TDV selects the parent Characteristic automatically. This adds a Characteristic to the cross-join, which increases the load on SAP BW, the network, and TDV.

Key Figures

Key Figures, also known as Measures, contain numeric data that is aggregated based on the sets present in the query. They are projected on the Query Axis COLUMNS (Axis 0).

Both the formatted and unformatted (numeric) values are available as columns and behave identically in queries.

Filter behavior for Key Figures is similar to that of ordinary SQL queries. The following filters are supported:

- NULL is treated as empty string. IS NULL is mapped to test for empty string.
- All conditional operators are supported, as well as the BETWEEN keyword.
- The keywords OR, AND, and NOT are supported. Grouping of terms with parentheses is also supported.
- The IN keyword is supported.

Ordering

Ordering of data is accomplished with the ORDER BY clause, adapted to work with SAP BW in the following ways:

- Ordering on Characteristic, Attribute, and Key Figure is supported.
- Ordering by more than one column is not allowed and causes an exception.
- The sort keywords ASC and DESC are supported, within the hierarchy. To sort regardless of hierarchy, specify the orderBreaksHierarchy query option. For more information, see the section [Query Options](#).
- Ordering on Internal, Characteristic [Level] or Characteristic [Hierarchy] columns cannot be pushed to SAP BW. To order on these columns, create a Parameterized Query and a View that selects from it.

Slicers

When a filter is placed on a column that is not projected, TDV treats the filter as a Slicer, placing it on the Slicer Axis. A Slicer qualifies tuples for each cell of the query, but does not affect the Query Axis. For example:

```
SELECT
```

```
Company Code, Division, Sales
```

```
FROM
```

```
DemoCube
```

WHERE

Calendar Month/Year = 'JAN 2001'

The above query returns the Sales Key Figure for the cross-join of Company Code and Division Characteristics, with cells restricted to tuples including the January 2001 member of the Calendar Month/Year Characteristic. This may result in many empty rows where Company Code/Division contains no Sales data, but these are removed by default. (To display empty rows, specify the query option `includeEmptyRows`. For more information, see the [Query Options](#).)

Slicers have the following behavior:

- The only column types permitted as Slicers are Characteristic and Characteristic [Key]. Others are ignored.
- All conditional operators except not equals (\neq) are treated as equals ($=$).
- Multiple members are combined into a set. The keywords OR and AND are treated identically as means of delineating members belonging to the set.
- Members can be excluded by using the not equals (\neq) operator or the NOT keyword. These map to the EXCEPT set expression.
- To specify a range from the natural order of the hierarchy, use greater than ($>$) or less than ($<$).
- Ranges without an upper and lower bound cause an exception.
- Only the Characteristic [Key] column type can be used to specify ranges. Ranges on other columns cause an exception.
- If the Slicer Axis consists of more than one member, it may be necessary to specify the order of members in the MDX. This is not possible from SQL, as the TDV query engine does not preserve the order of terms in a WHERE clause. To control the query to this extent, use custom MDX. See [Custom MDX](#) for further information.

Note: In BeX Query Designer, Slicers are called Filters.

Variables

Variables are features of SAP BW Queries. To provide a Variable value, add a filter to the SQL statement. Depending on the variable type, the value can be a member key (enclosed in square brackets), member text, or literal.

In the following example, the Variable Sales organization is assigned to members whose text matches the values USA Philadelphia or USA Denver.

```
SELECT
    "Business Partner", "Calendar Year/Month",
    "Returns Quantity", "Returns Value"
FROM
    "Returns per Business Partner"
WHERE
    "_Sales organization" IN ('USA Philadelphia', 'USA Denver')
```

The annotation of a Variable column includes guidance for how to use it within a query.

Rules for using Variables in queries are as follows:

- Variables with selection type Complex are allowed to contain multiple values.
- Variables with selection type Value are only allowed to contain a single value.
- All conditional operators and the BETWEEN keyword are supported.
- The keywords OR, AND, IN, and grouping of terms with parentheses ,are supported, but have no impact on the query other than to indicate the list of variable assignments.
- Multiple values for the same variable are mapped to INCLUDING or, with the keyword NOT, EXCLUDING.
- NULL is treated as empty string. IS NULL is mapped to test for empty string.

Custom MDX

The Internal column “__mdx” in InfoProvider and Query resources allows custom MDX to be sent directly to SAP BW, bypassing SQL-to-MDX query mapping in TDV. This helps when

you need to access features of MDX without using SQL; but then MDX queries cannot be used with other queries. TDV has no visibility to tune them.

The following is an example of using custom MDX in a query. The MDX function TOPCOUNT is used to provide ranking and ordering which cannot be accomplished in SQL:

```
SELECT
```

```
"Sales Personnel", "Calendar Year/Month",  
"Net value of the invoice item in the docCurrency (SAP Demo)"
```

```
FROM
```

```
"SAP DemoCube"
```

```
WHERE
```

```
"__mdx" =
```

```
'SELECT NON EMPTY { [Measures].[@D_NETVLINV] } ON AXIS(0),
```

```
NON EMPTY {TOPCOUNT({[@D_SALE_EMP].[LEVEL01].MEMBERS *
```

```
[@CALMONTH].[LEVEL01].MEMBERS }, 3,
```

```
[Measures].[@D_NETVLINV]}) ON AXIS(1)
```

```
FROM [$@D_DECU]
```

```
WHERE [@CALYEAR].[1998]'
```

Developing MDX can be challenging. It is best to work within a specialized OLAP tool to develop the initial query, test it, and then paste it into Studio.

Additionally, a number of rules must be followed for the MDX to execute properly within TDV:

- Axis 0 must contain Key Figures only.
- Axis 1 must contain Characteristics only.
- No other axes are permitted.
- The SELECT clause in TDV must contain corresponding columns for the Key Figures, Characteristics, and Attributes projected in the MDX statement. If any columns are missing or if any extraneous columns are introduced, the query fails, returns no data, or returns incomplete data.

Query Options

An internal column named “__options” (two underscores) is provided for every InfoProvider and Query resource. This column accepts a comma-delimited list of query options, which can modify the behavior of the SAP BW query in ways that cannot be expressed in SQL.

Valid options are the following:

- **leafNodes**: This option overrides the selection of the Data Source advanced property named Show leaf nodes only for the duration of the query. With this option, only members of dimensions that do not contain children are returned.
- **allNodes**: This option overrides the selection of the advanced property named Show leaf nodes only for the duration of the query. All nodes are returned exactly as received from SAP BW, regardless of their position in the hierarchy.
- **orderBreaksHierarchy**: When ordering rows with ORDER BY, the default behavior is to order members first by their position in the hierarchy, and then by the Characteristic, Attribute, or Key Figure specified in the ORDER BY clause. This option indicates that the hierarchy should be broken when ordering data.
- **keyDate**: Used for time-dependent queries. For more information see [Time-Dependent Queries](#).
- **includeEmptyRows**: This option specifies that the NON EMPTY clause is to be omitted from the ROWS axis for this query. Depending on the SAP BW data and query, this may result in empty values in Characteristic or Attribute columns. By default, empty rows are removed. Here is an example of the includeEmptyRows option:


```

SELECT

"Company code",

"Division",

"Sales organization",

"Costs in Document currency (SAP Demo)",

"Billed Quantity (SAP Demo)",

"__rownum"

FROM

/shared/QA_SAPBW/Sources/SAPBW_35/"SAP Demo"/"SAP Demo Sales and
Distribution"/"SAP DemoCube"/InfoProvider/"SAP DemoCube"

WHERE "Sales organization" = 'Paris'

AND "__options" = 'includeEmptyRows'

```

- includeEmptyColumns: Specifies that the NON EMPTY clause is omitted from the COLUMNS axis for this query. This may result in empty values in Key Figure columns. By default, empty columns are removed.

Row Number

The natural order of data from SAP BW may be important to preserve. Without an explicit ORDER BY clause, results from SAP BW are ordered hierarchically. Once the data is reordered due to a join or other SQL operation, it cannot be restored through ORDER BY or any other operator in TDV because it relies on the hierarchical order available only to SAP BW.

For this reason, an Internal column named “__rownum” (two underscores) exists for every SAP BW resource in TDV. Values in this column range from 0 to n, indicating the natural order of data as returned from SAP BW. This makes it possible to preserve the natural order of the data. This column can be selected, but not in a filter expression.

Time-Dependent Queries

Hierarchy nodes and other data in SAP BW can change. By default, queries run using the current date. To set a different date, create a filter on the “__options” Internal column, using the keyDate option and a date in YYYY-MM-DD format.

In the following example, the query is executed with a key date of January 1, 2006:

```
SELECT

"Sold-to party", "Net value of the invoice item in the docCurrency (SAP
Demo)"

FROM

SAP_DemoCube

WHERE

"__options" = 'keyDate 2006-01-01'
```

Enabling Tracing

Tracing logs every interaction with SAP BW to files so that data values can be examined and validated against output in TDV. Combined with the debug output in cs_server.log, this can be helpful for troubleshooting queries.

To enable tracing

Edit this properties file:

```
<TDV install directory>\apps\dlm\app_ds_sapbw\conf\product.properties
```

1. Change the property sapbw.trace to true.

2. Restart the TDV Server.
3. Check the trace files that are stored in the following directory:

```
<TDV_install_dir>\apps\d1m\app_ds_sapbw
```

Note: Tracing affects performance and creates large files on disk, so be sure to change the `sapbw.trace` property to false and restart TDV Server when tracing is no longer needed.

Working with Large Batch Sizes

If your developers work with queries that return large amounts of data, or depend on the amount of available memory, you might be able to improve performance by adjusting the batch size.

Typically, the batch size should not be changed. If the setting is too high, the server requires too much memory to process the results from SAP BW and can crash. If the setting is too low, query performance can suffer because the server might unnecessarily split up single queries into multiple queries. However, depending on other factors such as network latency and bandwidth, it is possible that increasing the batch size could result in improved performance on queries that return amounts of data larger than the batch size.

- If the TDV server has a large amount of memory available to the JVM, the batch size might be increased.
- If the TDV server has a shortage of memory, the batch size might be decreased.

To enable working with large files

Open this properties file using your favorite text editor:

```
<TDV_install_dir>\apps\d1m\app_ds_sapbw\conf\product.properties
```

4. Change the property `sapbw.batchSize` setting.

By default, `sapbw.batchSize` is set to 25000 cells (TDV requests 25000 cells for each call to SAP BW). If there are more cells in the response, TDV makes more requests for 25000 cells until all of the data is transferred from SAP BW to TDV. The minimum setting for `sapbw.batchSize` is 1; the maximum is 999999.

5. Restart TDV Server.

Multidimensional Queries

Queries against SAP BW are multidimensional, which introduces a new behavior to TDV that may be unexpected to users accustomed to querying relational databases.

- [The Show Contents Feature](#)
- [Create OLAP Views Using SAP BW Data Sources](#)
- [Work with Views](#)
- [Filter on Levels](#)
- [Clearing the SAP BW Metadata Cache](#)

The Show Contents Feature

The feature Show Contents available in Studio is not recommended on any SAP BW resource. Show Contents performs a `SELECT *` query, which results in a cross-join of every possible Characteristic in the SAP BW resource.

Create OLAP Views Using SAP BW Data Sources

When creating a new view that uses an SAP BW data source in combination with other data sources, or when using multiple data sources of any type, use the New OLAP View command in Studio to create a new view.

If you are creating a view that calls only one SAP BW data source, you can create a New OLAP View in Studio. For more information about OLAP Views, see the *TDV User Guide*.

Work with Views

TDV Server makes a number of assumptions when optimizing queries, which can cause confusion. This is especially true when building Views that call other Views, or when using Views from outside of TDV using JDBC or ODBC. Because a cross-join is used to query SAP BW, when a Characteristic column is removed from a query due to optimization, the meaning of the underlying MDX query changes dramatically.

For example, if a View includes the Characteristics “Company Code” and “Division” and “Key Figure Sales,” an ODBC query selecting only “Division” and “Sales” produces a different data set than a query selecting all columns. To ensure the same base data are returned regardless of the query, wrap any Views using SAP BW in a Parameterized Query,

to query users. This prevents TDV from applying relational optimization rules to the multi-dimensional query.

Filter on Levels

You can filter on level columns (Dimension [Level]). For example, you can add a WHERE clause to the SQL statement that filters the [Level] field:

```
SELECT Dimension1, Dimension2, Measure
```

```
FROM Cube
```

```
WHERE "Dimension1 [Level]" = 2
```

This would return Measure for the cross product of Dimension1 (members beginning from level 2 and below) and Dimension2 (all members).

Clearing the SAP BW Metadata Cache

TDV caches SAP BW metadata to improve query performance. These caches are automatically cleared after every TDV restart, but you might want to do this at other times.

For example, your SAP BW data source might become out of sync with the SAP BW database, even after re-introspecting the data. If the metadata is still out of sync after re-introspection, you can restart the TDV instance, or you can clear the metadata cache as described in this section.

To clear the metadata cache

Open the SAP BW data source.

6. Select the Re-Introspection tab.
7. Click Clear Metadata Cache.

Joins and Query Performance

Joins cannot be pushed to SAP BW. Executing joins in TDV can degrade performance, because a table scan would be required, and it would fetch every row of the joined tables.

The technology that TDV uses to connect with SAP BW is not optimized for large data sets, so table scans should be avoided.

A semijoin is the best way to reduce the number of SAP BW rows retrieved and processed. To force a semijoin to occur in a query, add the option immediately before the table to be joined. Example:

```
SELECT *
```

```
FROM X INNER {OPTION SEMIJOIN} JOIN Y ON X.Key = Y.Key
```

Values of X.Key are collected and passed in a query to Y as the filter:

```
SELECT * FROM X
```

```
SELECT * FROM Y WHERE Key IN ({values_of_X.Key_from_previous_query})
```

If X has many rows, queries against Y can be lengthy. If the queries against Y become too large, TDV automatically partitions them and reassembles the results into a unified set.

Put the table that returns the larger number of rows on the right side of the join whenever possible. When running a new query for the first time, activate the Execution Plan tab in the Studio and click Execute and Show Statistics. Examine each node's row count and query after processing has begun to make sure that filters are pushed down to SAP BW. This is a good way to see the mechanics of a semijoin in action. If the interaction between TDV Server and SAP BW is still unclear and performance is poor, enable debug logging for the adapter as described in the *TDV Installation and Upgrade Guide*.

Security for SAP BW with TDV

This topic describes TDV support for SAP BW security features. It assumes knowledge of SAP BW's security infrastructure.

- [Required Authorizations](#)
- [Troubleshooting Security-Related Errors](#)

Required Authorizations

The following authorizations are required to log into SAP BW from TDV and introspect Cubes and Queries::

```
Class: AAAB (Cross-application Authorization Objects)
Object: S_RFC (Authorization Check for RFC Access)
Field: Activity . Value : 16 ( Execute)
```

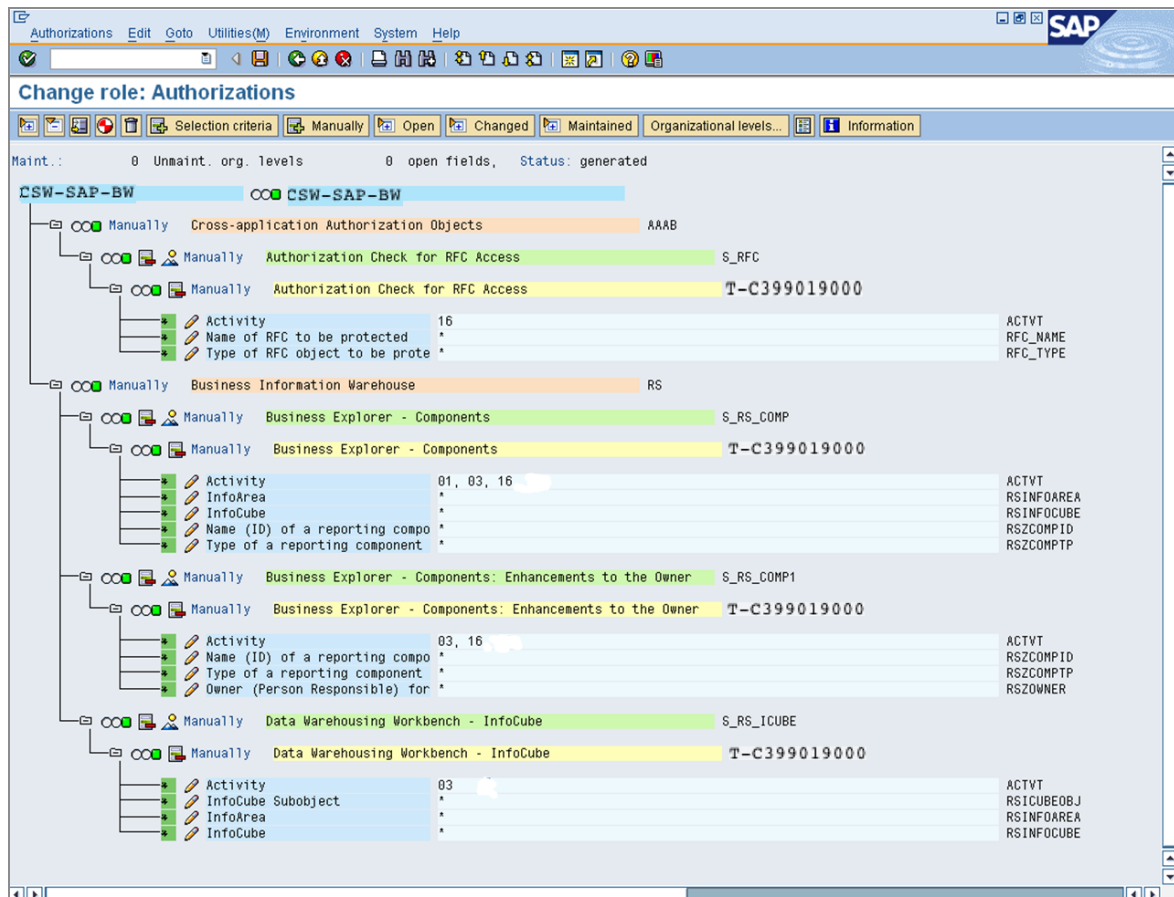
In addition, the following authorization objects can be used to limit the InfoAreas visible to the TDV user, as well as limiting the InfoProviders and Queries accessible:

```
Class: RS (Business Information Warehouse),
Object: S_RS_COMP (Business Explorer - Components)
Field: Activity . Value : 01 ( Create), 03 ( Display) and 16 ( Execute)
```

Note: The SAP ODBO interface does not modify InfoObjects but still Activity 01 is required. Otherwise SAP BW introspection does not fetch any metadata.

```
Class: RS (Business Information Warehouse),
Object: S_RS_COMP1 (Business Explorer - Components: Enhancements to the Owner)
Field: Activity . Value : 03 ( Display) and 16 ( Execute)
```

```
Class: RS (Business Information Warehouse),
Object: S_RS_ICUBE (Administrator Workbench - InfoCube)
Field: Activity . Value : 03 ( Display)
```



Troubleshooting Security-Related Errors

This section describes common security-related errors and their remedies.

RFC Authorization

ERROR: User TESTUSER1 has no RFC authorization for function group SYST

TDV is logging into SAP with an SAP BW user that lacks the authorization object S_RFC. Set authorization S_RFC_ALL to grant access to all RFCs, or restrict them to only the OLAP BAPIs required by TDV:

BAPI_MDPROVIDER_GET_DIMENSIONS
 BAPI_MDPROVIDER_GET_MEASURES
 BAPI_MDPROVIDER_GET_PROPERTIES
 BAPI_MDPROVIDER_GET_CATALOGS
 BAPI_MDPROVIDER_GET_CUBES

BAPI_MDPROVIDER_GET_HIERARCHYS
BAPI_MDPROVIDER_GET_LEVELS
BAPI_MDPROVIDER_GET_MEMBERS
BAPI_MDPROVIDER_GET_VARIABLES
BAPI_MDDATASET_GET_CELL_DATA
BAPI_MDDATASET_DELETE_OBJECT
BAPI_MDDATASET_CREATE_OBJECT
BAPI_MDDATASET_SELECT_DATA
BAPI_MDPROVIDER_SET_KEY_DATE
BAPI_MDDATASET_GET_AXIS_DATA
BAPI_MDDATASET_GET_AXIS_INFO

SAP BW Global Properties

This topic describes the SAP BW global properties file that you can edit to control how TDV works with SAP BW. Global properties apply to all instances of the SAP BW data source. You can edit the properties to control how TDV works with SAP BW.

SAP BW Global Trace Level Property

Global properties are stored in the file:

<TDV _install_directory>\apps\d1m\app_ds_sapbw\conf\product.properties

Edit global properties using a text editor such as Notepad on Windows. TDV must be restarted for property changes to take effect.

There currently is only one global property for SAP BW.

sap.jco.traceLevel

Default Value: 2

Description: Turns on the SAP JCo trace. Allowed levels are zero through ten. The commonly used ones are:

0—Nothing

- 1—Errors
- 2—Errors and warnings
- 3—Info messages, errors and warnings
- 4—Execution path, info messages, errors and warnings
- 5—Verbose execution path, info messages, errors and warnings
- 6—Verbose execution path, limited data dumps, info messages, errors and warnings
- 7—Full execution path, data dumps with metadata, verbose info messages, errors and warnings
- 8—Full execution path, full data dumps with metadata, verbose info messages, errors and warnings

TIBCO Product Documentation and Support Services

For information about this product, you can read the documentation, contact TIBCO Support, and join the TIBCO Community.

How to Access TIBCO Documentation

Documentation for TIBCO products is available on the [TIBCO Product Documentation](#) website, mainly in HTML and PDF formats.

The [TIBCO Product Documentation](#) website is updated frequently and is more current than any other documentation included with the product.

Product-Specific Documentation

The following documentation for this product is available on the [TIBCO® Data Virtualization](#) page.

- **Users**
 - TDV Getting Started Guide
 - TDV User Guide
 - TDV Web UI User Guide
 - TDV Client Interfaces Guide
 - TDV Tutorial Guide
 - TDV Northbay Example
- **Administration**
 - TDV Installation and Upgrade Guide
 - TDV Administration Guide
 - TDV Active Cluster Guide
 - TDV Security Features Guide
- **Data Sources**

TDV Adapter Guides

TDV Data Source Toolkit Guide (Formerly Extensibility Guide)

- **References**

TDV Reference Guide

TDV Application Programming Interface Guide

- **Other**

TDV Business Directory Guide

TDV Discovery Guide

- *TIBCO TDV and Business Directory Release Notes* Read the release notes for a list of new and changed features. This document also contains lists of known issues and closed issues for this release.

Release Version Support

TDV 8.5 is designated as a Long Term Support (LTS) version. Some release versions of TIBCO® Data Virtualization products are selected to be long-term support (LTS) versions. Defect corrections will typically be delivered in a new release version and as hotfixes or service packs to one or more LTS versions. See also

https://docs.tibco.com/pub/tdv/general/LTS/tdv_LTS_releases.htm.

How to Contact TIBCO Support

Get an overview of [TIBCO Support](#). You can contact TIBCO Support in the following ways:

- For accessing the Support Knowledge Base and getting personalized content about products you are interested in, visit the [TIBCO Support](#) website.
- For creating a Support case, you must have a valid maintenance or support contract with TIBCO. You also need a user name and password to log in to [TIBCO Support](#) website. If you do not have a user name, you can request one by clicking **Register** on the website.

How to Join TIBCO Community

TIBCO Community is the official channel for TIBCO customers, partners, and employee subject matter experts to share and access their collective experience. TIBCO Community offers access to Q&A forums, product wikis, and best practices. It also offers access to extensions, adapters, solution accelerators, and tools that extend and enable customers to gain full value from TIBCO products. In addition, users can submit and vote on feature requests from within the [TIBCO Ideas Portal](#). For a free registration, visit [TIBCO Community](#).

Legal and Third-Party Notices

SOME TIBCO SOFTWARE EMBEDS OR BUNDLES OTHER TIBCO SOFTWARE. USE OF SUCH EMBEDDED OR BUNDLED TIBCO SOFTWARE IS SOLELY TO ENABLE THE FUNCTIONALITY (OR PROVIDE LIMITED ADD-ON FUNCTIONALITY) OF THE LICENSED TIBCO SOFTWARE. THE EMBEDDED OR BUNDLED SOFTWARE IS NOT LICENSED TO BE USED OR ACCESSED BY ANY OTHER TIBCO SOFTWARE OR FOR ANY OTHER PURPOSE.

USE OF TIBCO SOFTWARE AND THIS DOCUMENT IS SUBJECT TO THE TERMS AND CONDITIONS OF A LICENSE AGREEMENT FOUND IN EITHER A SEPARATELY EXECUTED SOFTWARE LICENSE AGREEMENT, OR, IF THERE IS NO SUCH SEPARATE AGREEMENT, THE CLICKWRAP END USER LICENSE AGREEMENT WHICH IS DISPLAYED DURING DOWNLOAD OR INSTALLATION OF THE SOFTWARE (AND WHICH IS DUPLICATED IN THE LICENSE FILE) OR IF THERE IS NO SUCH SOFTWARE LICENSE AGREEMENT OR CLICKWRAP END USER LICENSE AGREEMENT, THE LICENSE(S) LOCATED IN THE “LICENSE” FILE(S) OF THE SOFTWARE. USE OF THIS DOCUMENT IS SUBJECT TO THOSE TERMS AND CONDITIONS, AND YOUR USE HEREOF SHALL CONSTITUTE ACCEPTANCE OF AND AN AGREEMENT TO BE BOUND BY THE SAME.

This document is subject to U.S. and international copyright laws and treaties. No part of this document may be reproduced in any form without the written authorization of TIBCO Software Inc.

TIBCO, TIBCO logo, Two-Second Advantage, TIBCO Spotfire, TIBCO ActiveSpaces, TIBCO Spotfire Developer, TIBCO EMS, TIBCO Spotfire Automation Services, TIBCO Enterprise Runtime for R, TIBCO Spotfire Server, TIBCO Spotfire Web Player, TIBCO Spotfire Statistics Services, S-PLUS, and TIBCO Spotfire S+ are either registered trademarks or trademarks of TIBCO Software Inc. in the United States and/or other countries.

Java and all Java based trademarks and logos are trademarks or registered trademarks of Oracle Corporation and/or its affiliates.

All other product and company names and marks mentioned in this document are the property of their respective owners and are mentioned for identification purposes only.

This software may be available on multiple operating systems. However, not all operating system platforms for a specific software version are released at the same time. See the readme file for the availability of this software version on a specific operating system platform.

THIS DOCUMENT IS PROVIDED “AS IS” WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, OR NON-INFRINGEMENT.

THIS DOCUMENT COULD INCLUDE TECHNICAL INACCURACIES OR TYPOGRAPHICAL ERRORS. CHANGES ARE PERIODICALLY ADDED TO THE INFORMATION HEREIN; THESE CHANGES WILL BE INCORPORATED IN NEW EDITIONS OF THIS DOCUMENT. TIBCO SOFTWARE INC. MAY MAKE IMPROVEMENTS AND/OR CHANGES IN THE PRODUCT(S) AND/OR THE PROGRAM(S) DESCRIBED IN THIS DOCUMENT AT ANY TIME.

THE CONTENTS OF THIS DOCUMENT MAY BE MODIFIED AND/OR QUALIFIED, DIRECTLY OR INDIRECTLY, BY OTHER DOCUMENTATION WHICH ACCOMPANIES THIS SOFTWARE, INCLUDING BUT NOT LIMITED TO ANY RELEASE NOTES AND "READ ME" FILES.

This and other products of TIBCO Software Inc. may be covered by registered patents. Please refer to TIBCO's Virtual Patent Marking document (<https://www.tibco.com/patents>) for details.

Copyright © 2002-2023 Cloud Software Group, Inc All Rights Reserved.