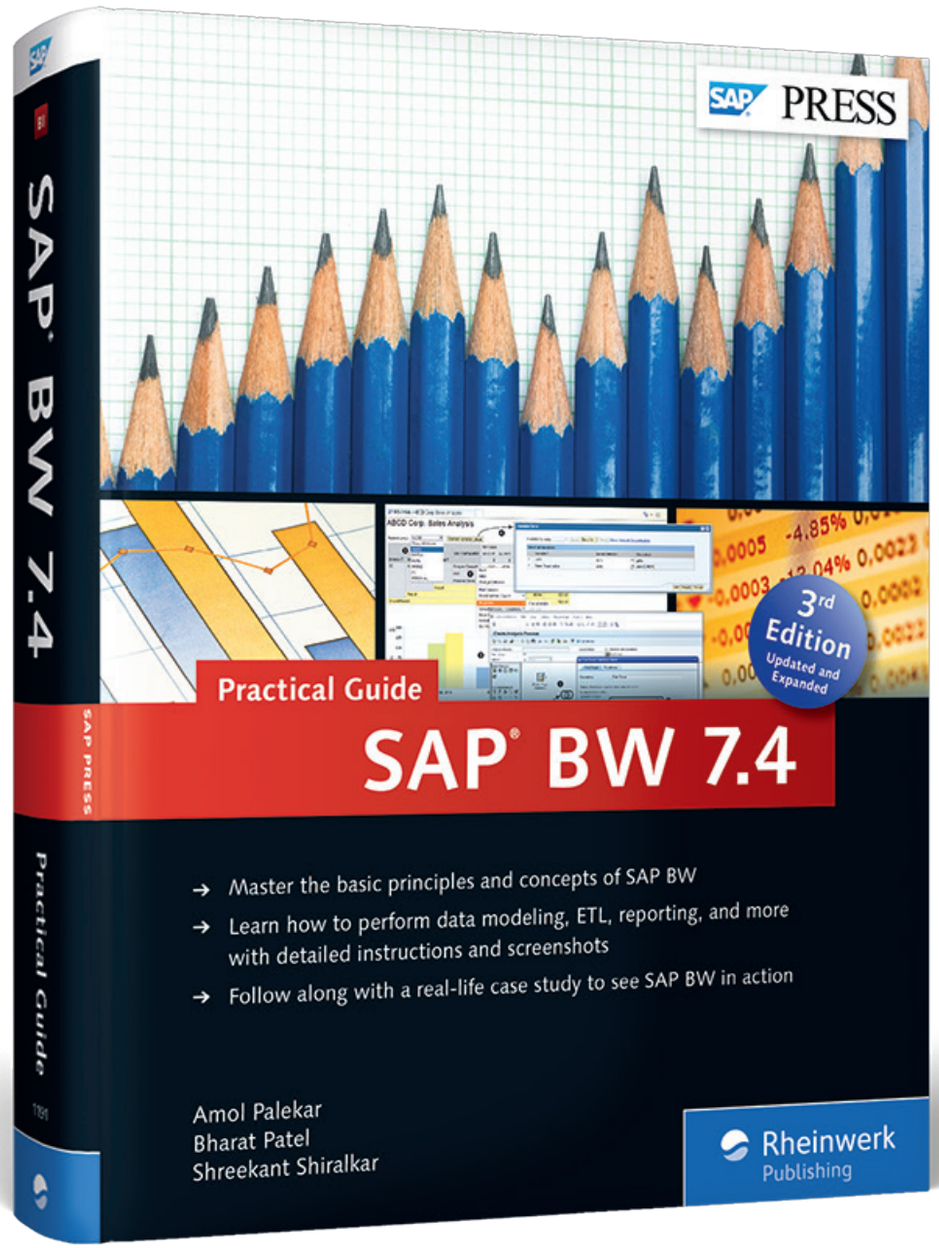


Reading Sample

In this sample chapter from the book, you'll be introduced to the different types of InfoProviders that exist in SAP BW. Having a firm understanding of how these objects (which include MultiProviders, HybridProviders, InfoSets, Open ODS views, and more) provide flexibility with SAP BW architecture is essential to using your data warehouse well.

Amol Palekar, Bharat Patel, Shreekant Shiralkar
SAP BW 7.4—Practical Guide
852 Pages, 2015, $69.95/€69.95
ISBN 978-1-4932-1191-3
www.sap-press.com/3733

Understanding InfoProviders and their design is essential for making use of the flexibility these objects provide in your overall BI solution architecture.

6 InfoProviders

Other than the basic InfoCubes and DataStore Objects (DSOs), several other InfoProviders are available in SAP BW:

- MultiProviders
- InfoSets
- VirtualProviders
- Semantically partitioned objects (SPOs)
- HybridProviders
- Transient providers
- Analytical indexes
- CompositeProviders
- Open Operational DataStore (ODS) views

Some of these are simply based on a logical definition and don't store any data physically. In some cases, these are based on a combination of two or more data targets such as InfoObjects (characteristics with master data), DSOs, and InfoCubes. These InfoProviders answer many business queries and save on efforts to extract and store the data. They provide flexibility in managing a data warehouse and offer scalability and efficiency. Queries can be created on the InfoProviders using standard query tools supplied by SAP BW. In this chapter, we provide an overview of these InfoProviders and explain possible scenarios where you might use them.

6.1 MultiProviders

Normally, the design of a data target such as an InfoCube or DSO is based on one business process; for example, an InfoCube for sales billing process data and another InfoCube for sales order process data. In this way, SAP BW may have multiple InfoCubes, each supporting an individual business process. Business information requirements may invite a situation where data from two different InfoCubes needs to be joined.

SAP BW supports queries based on a single InfoProvider. To support reporting requirements across multiple data providers, you don't need to load data from individual data providers to the new data provider. The system provides a better way to handle this situation, by way of a MultiProvider.

6.1.1 Introduction to MultiProviders

Because a MultiProvider exists only as a logical definition, it doesn't physically store data. The data lies in the underlying data providers, which define the MultiProvider. You can create a MultiProvider based on the following objects, as shown in Figure 6.1 (note that all possible objects aren't shown in this figure):

- DSOs
- InfoCubes
- SPOs
- HybridProviders
- InfoObjects
- InfoSets
- Aggregation levels
- Transient providers
- VirtualProviders

MultiProviders can be created based on any combination of these objects, for example, InfoCube to InfoCube (not limited to two InfoCubes), InfoCube to InfoObject, or DSO to InfoCube. Again, the number of objects included in the definition of a MultiProvider isn't limited to two.

Figure 6.2 shows an example of including two InfoCubes in a MultiProvider: one on planning data and the other on actual sales. While the actuals InfoCube stores

the data from actual sales, the plan InfoCube stores the data on sales planning. Another example of creating a MultiProvider based on InfoCubes and InfoObjects is having the sales InfoCube and the InfoObject 0MATERIAL. A query on such a MultiProvider might be used to identify slow-moving material.

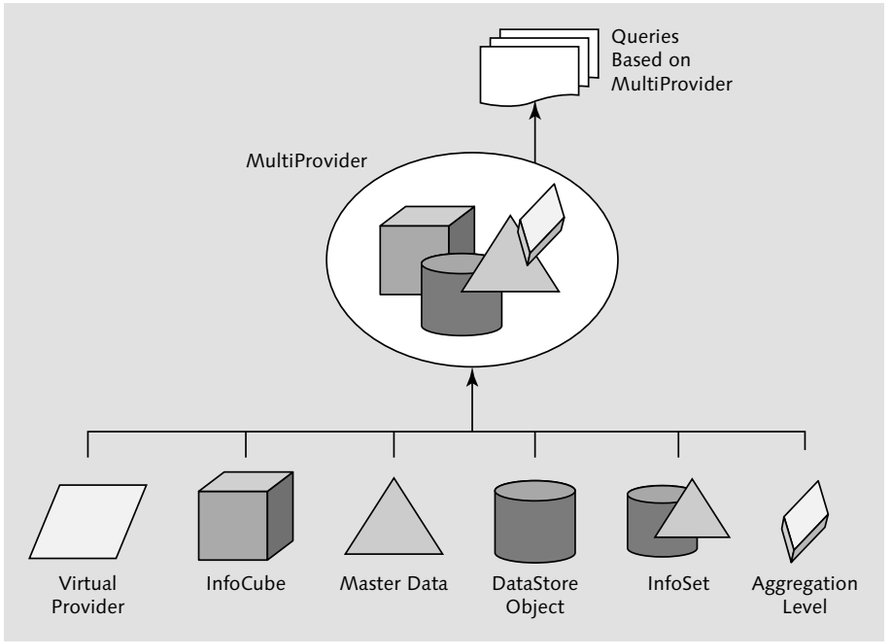


Figure 6.1 MultiProvider: A Logical Definition

You can also use a MultiProvider when the InfoCube you're creating becomes very large due to high data volume. In this case, you can split the InfoCube into identical smaller InfoCubes based on values of a logical characteristic such as fiscal year or company code. For reporting purposes, it's recommended that you create MultiProviders based on these InfoCubes and create queries on the MultiProvider. With newer version of SAP BW, instead of breaking a large InfoCube into multiple physical small InfoCubes, you can use an SPO, which is explained in Section 6.4.

A MultiProvider provides the following benefits:

- **Flexibility**
  Designing InfoCubes based on individual business processes is much simpler

than creating one complex InfoCube for multiple business processes, and it allows for the combination of various InfoCubes at a later stage. A MultiProvider allows you to keep your InfoCubes design simple and small.

- **Simple and easy**
  Small InfoCubes are easy to maintain.
- **Faster results**
  The system uses parallel processing when executing queries on a MultiProvider. As shown in Figure 6.2, when a query is executed on the planned sales versus actual sales MultiProvider, the system internally starts multiple subqueries in parallel. After the results of these queries are available, they are combined using the union operation and presented to the user.

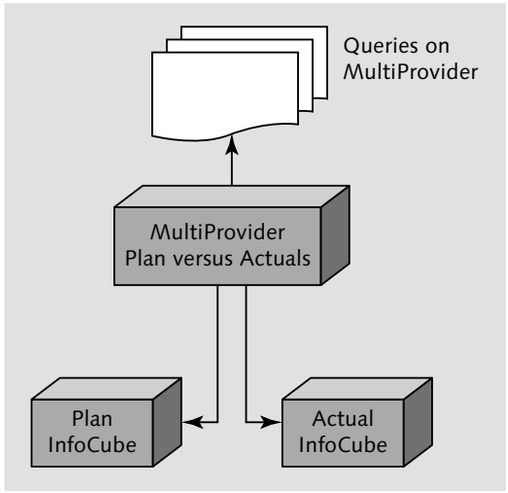


Figure 6.2 MultiProvider Example: Plan versus Actual

6.1.2 Designing MultiProviders

Having explained MultiProviders and their advantages, we'll now explain the step-by-step process of creating a MultiProvider in SAP BW.

Creating MultiProviders

Our example scenario requires that ABCD Corp.'s BI solution can report the status of actual sales compared to the planned sales forecast to ascertain whether

company sales are on target or whether action is needed to address variance. We'll use a sales InfoCube, which stores actual sales data, and we'll use a planning InfoCube, which stores the plan data. Let's now proceed with the creation of a MultiProvider based on plan versus actual sales.

Note

To explain the concept of a MultiProvider, we've referred to InfoCube BW_PLAN. For our example scenario, this InfoCube stores the sales planning data. This real-time InfoCube is discussed in detail when we explain the SAP BW Integrated Planning component in Chapter 12. If you want to practice creating a MultiProvider and the steps mentioned in the following section, first create InfoProvider BW_PLAN by referring to Chapter 12, Section 12.2.1. The definition of InfoCube BW_PLAN and the corresponding InfoObjects (BW_PROD, BW_VAL) is shown in Figure 12.7 in that section.

From the Database Warehousing Workbench screen (Transaction RSA1), select InfoProvider from the Modeling section, as shown in ❶ of Figure 6.3. A MultiProvider is created under an InfoArea, so select BW_InfoAREA ❷. Open the context menu of BW_Infoarea, and select Create MultiProvider ❸.

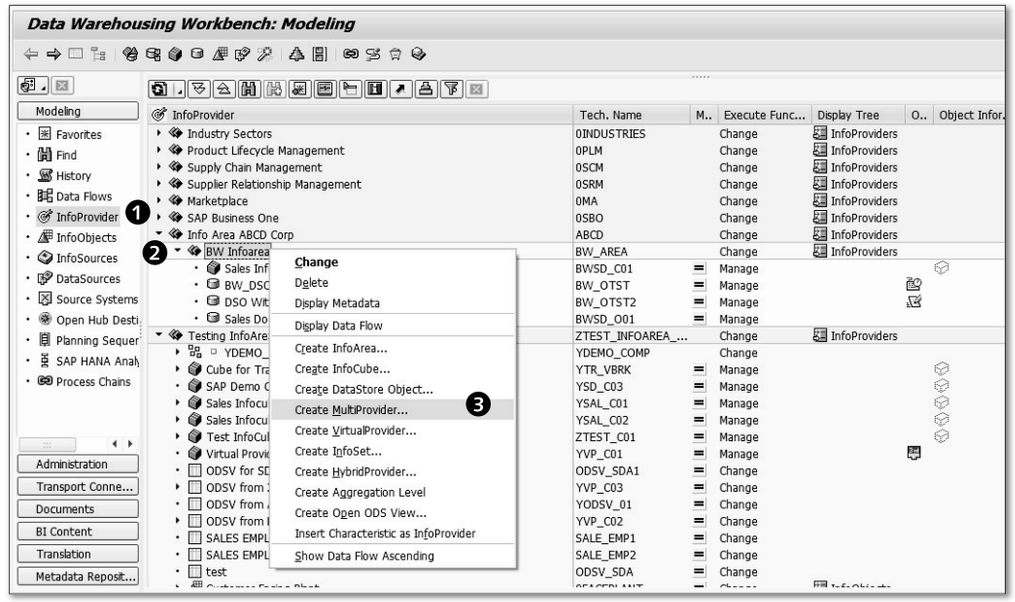


Figure 6.3 Creating a MultiProvider

The Create MultiProvider pop-up box appears, as shown in ❶ of Figure 6.4. You need to provide the unique technical name and description of the MultiProvider. Enter the technical name "BWSD_MUL1" ❷, and enter the description "Sales Actuals Vs Plan MultiProvider" ❸. Click on the Create icon ❹.

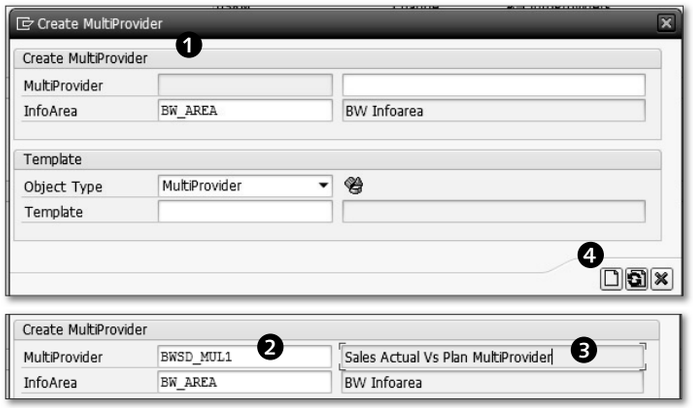


Figure 6.4 Creating a MultiProvider: Adding a Name and Description

The next screen is MultiProvider: Relevant InfoProviders , as shown in Figure 6.5. Because a MultiProvider is based on different data targets and/or InfoProviders (as shown earlier in Figure 6.1), this screen offers you selections based on the data targets and/or InfoProviders you want to include in the definition of the MultiProvider.

Different tabs are available based on various relevant InfoProviders (❶ of Figure 6.5). These tabs allow you to select various basic providers for the MultiProvider. For our example, select InfoCube BWSD_C01 (Sales InfoCube) and InfoCube BW_PLAN (Sales Planning Cube) ❷.

Because there may be a large number of data targets and/or InfoProviders available in the system, three different display options are available to list them ❸. By default, the system uses the Display All InfoProviders (Place Selected First) option. The search feature is also available to find specific InfoProviders ❹.

After the required InfoProviders are selected, click on the Continue icon ❺ to move ahead.

Figure 6.5 Creating a MultiProvider: Selecting InfoProviders

Defining MultiProviders

The system takes you to the Edit MultiProvider screen, as shown in Figure 6.6. The initial definition of MultiProvider BWSD_MUL1 is shown with included InfoCubes (BWSD_C01 and BW_PLAN) ❶. Four default dimensions are shown ❷: Data Package, Time, Unit, and Dimension 1. This screen also offers a section to include Navigation Attributes ❸ and Key Figures ❹.

Because both InfoCubes support different business processes, their definitions ought to be different. However, a few characteristics and key figures are common between the two. Table 6.1 provides a list of characteristics and key figures from both of the InfoCubes.

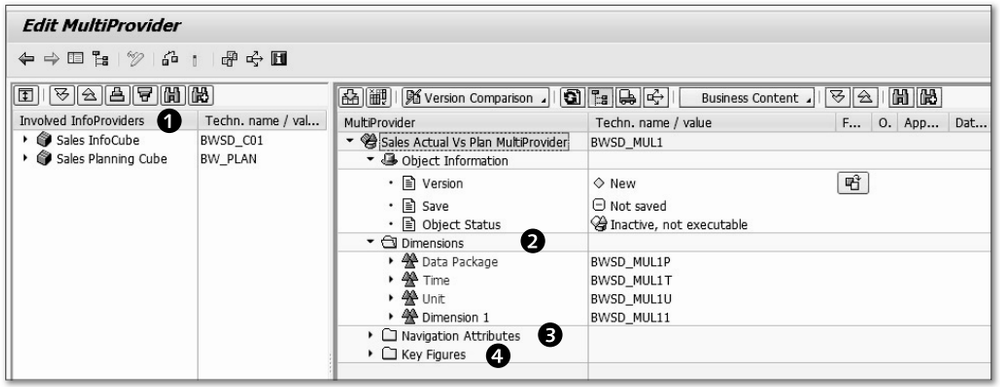


Figure 6.6 Edit MultiProvider Initial Screen

Type	Technical Name in BWSD_C01	Technical Name in BW_PLAN
Characteristics	0COMP_CODE	Not Available
	0DIVISION	0DIVISION
	0PLANT	Not Available
	0SALESORG	0SALESORG
	0DISTR_CHAN	Not Available
	0SALES_OFF	Not Available
	0SALES_GRP	Not Available
	0CO_AREA	Not Available
	BW_CUST	Not Available
	0MATERIAL	Not Available
	0MATL_GROUP	0MATL_GROUP
	0ACCNT_ASGN	Not Available
	0CUST_GROUP	Not Available
	0CUST_GRP1	Not Available
	0REGION	Not Available
	0SALES_DIST	Not Available
	0COUNTRY	Not Available
	0BILL_TYPE	Not Available
	0ITEM_CATEG	Not Available
	0BILL_CAT	Not Available

Table 6.1 Comparison of Definitions for Designing MultiProviders

Type	Technical Name in BWSD_C01	Technical Name in BW_PLAN
	0DOC_CATEG	Not Available
	Not Available	BW_PROD
Time Characteristics	0CALMONTH	Not Available
	0CALQUARTER	0CALQUARTER
	0CALYEAR	0CALYEAR
Key Figures	BW_QTY	Not Available
	0NET_WGT_DL	Not Available
	0GRS_WGT_DL	Not Available
	0COST	Not Available
	0NET_VALUE	Not Available
	0SUBTOTAL_1	Not Available
	0SUBTOTAL_2	Not Available
	0VALUE_LC	Not Available
	BW_PRICE	Not Available
	Not Available	BW_VAL

Table 6.1 Comparison of Definitions for Designing MultiProviders (Cont.)

We can see that only a few characteristics of InfoCube BWSD_C01 are available in the InfoCube BW_PLAN. One of the basic conditions in designing a MultiProvider is to have the identical technical name of the characteristics across InfoCubes (or other included data targets).

All of the characteristic InfoObjects included in InfoCube BW_PLAN have the same InfoObject in InfoCube BWSD_C01, except BW_PROD. BW_PROD is used to refer to a product or material, similar to the use of InfoObject 0MATERIAL of InfoCube BWSD_C01. The two InfoObjects, BW_PROD and 0MATERIAL, can't be matched while designing a MultiProvider. Although including such non-matching InfoObjects is possible in a MultiProvider definition, queries using them don't produce the proper results when such disjointed characteristics are used.

When defining the MultiProvider, we'll include the following characteristics and key figures:

- 0SALESORG
- 0DIVISION

- 0MATL_GROUP
- BW_PROD
- 0CALQUARTER
- 0CALYEAR
- 0NET_VALUE (actual value)
- BW_VAL (planning value)

The dimensions of InfoCube BWSD_C01 and InfoCube BW_PLAN are shown in ❶, ❷, and ❸ of Figure 6.7. In this figure, characteristics within the ORGANIZATION for SALES PLANNING CUBE dimension are shown in detail. Because a MultiProvider exists as a logical definition and doesn't physically store data, the design of various dimensions isn't as vital as it is during the design of a standard InfoCube. Dimension design while creating a standard InfoCube is a crucial step because standard InfoCubes store data physically.

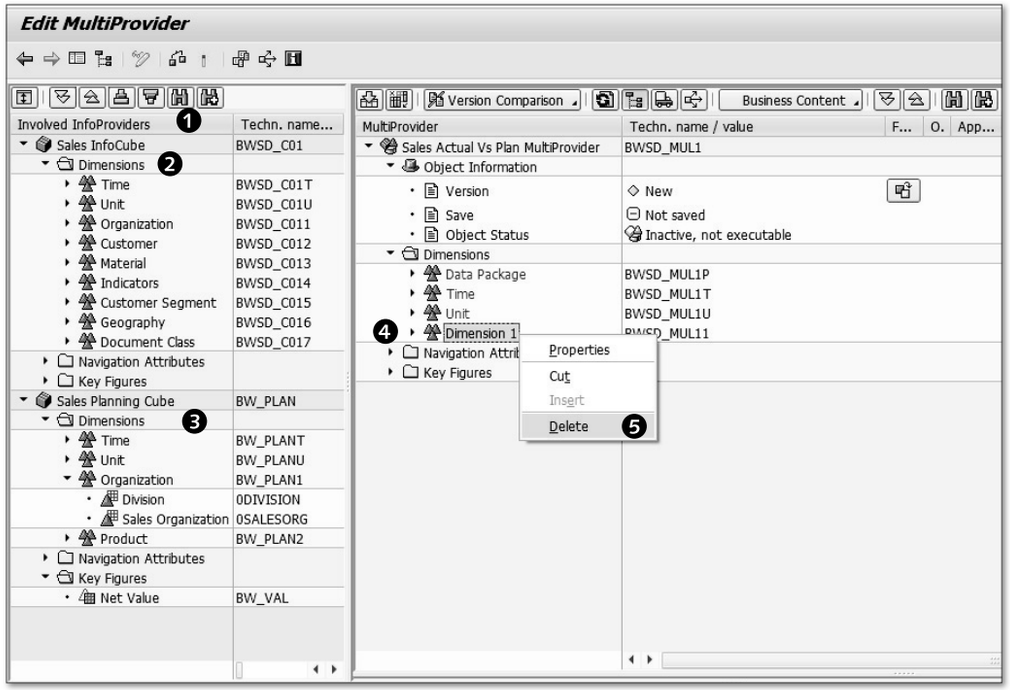


Figure 6.7 Edit MultiProvider: Deleting Dimension 1

A small error in dimension design can seriously and negatively affect the performance of queries. (Note that dimension design is crucial for SAP BW on non-SAP HANA databases.) This doesn't apply to creating a MultiProvider, but attaching characteristics in logical groupings in various dimensions helps BEx Query Designer—so it's still recommended that you design dimensions.

First, let's delete dimension 1, which is created by default by SAP BW. Select DIMENSION 1, as shown in ❹ of Figure 6.7, and open the context menu. Next, select DELETE ❺.

You can either create new dimensions or directly drag and drop dimensions from any of the InfoCubes, as shown in Figure 6.8. Select the ORGANIZATION dimension ❶, and then drag and drop it over DIMENSIONS ❷. The outcome of this activity is that the ORGANIZATION dimension is available in MultiProvider BWSD_MUL1, with both the characteristics (SALES ORGANIZATION and DIVISION) ❸ of the base InfoCube (BW_PLAN/Sales Planning Cube).

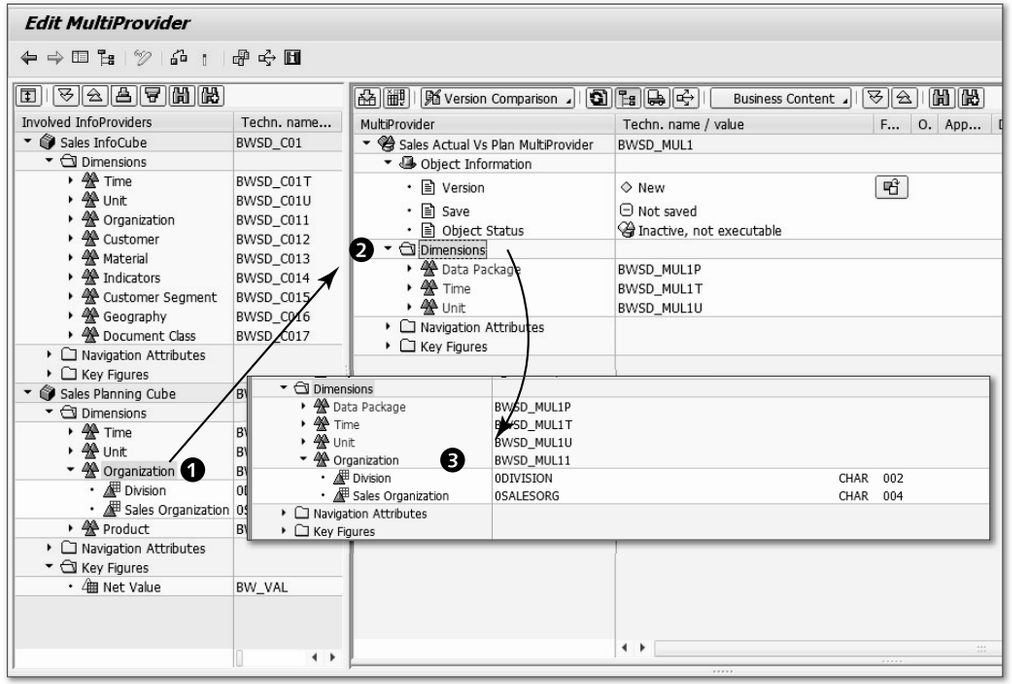


Figure 6.8 Edit MultiProvider: Dragging and Dropping Dimension

Now, drag and drop the PRODUCT dimension and the characteristics 0CALQUARTER and 0CALYEAR from the TIME dimension of InfoCube BW_PLAN to the TIME dimension of MultiProvider BWSD_MUL1. Also drag and drop key figures BW_VAL and 0NET_VALUE from the InfoCube to the MultiProvider.

The final design of MultiProvider BWSD_MUL1 is shown in Figure 6.9.

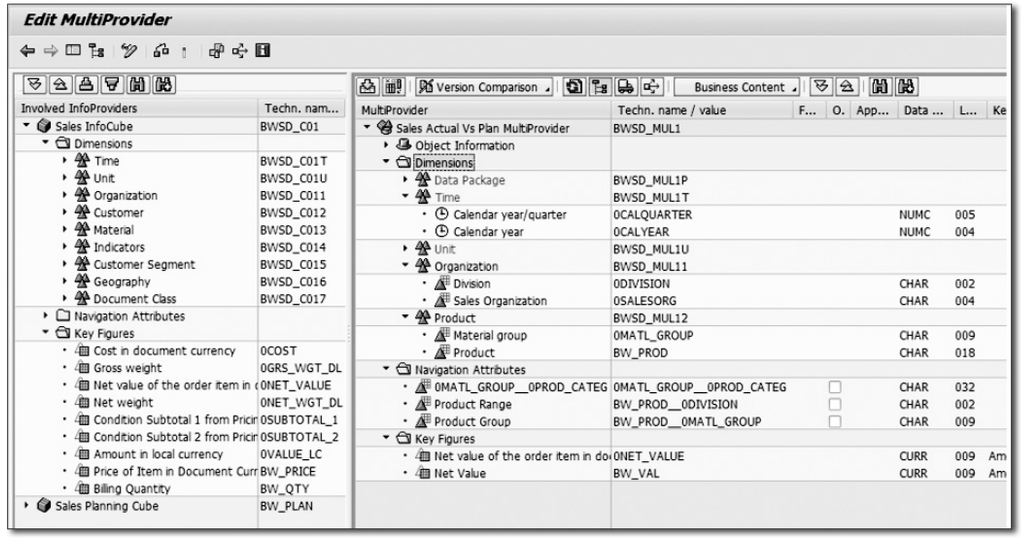


Figure 6.9 Edit MultiProvider: Dimensions and Key Figures

Identifying Characteristics

Now you need to identify each of the characteristics included in the MultiProvider and match them to the characteristics or navigation attributes of the base InfoCube. We'll explain this process next.

Click on the IDENTIFY CHARACTERISTICS icon, as shown in ❶ of Figure 6.10. The IDENTIFICATION OF PARTICIPATING CHARACTERISTICS/NAV. ATTR. box appears ❷. It shows each characteristic included in the MultiProvider and offers the matching characteristics or navigational attributes available from the included InfoCube.

This box shows the characteristic 0CALQUARTER ❸ from the MultiProvider and offers matching characteristic 0CALQUARTER from the SALES INFOCUBE (BWSD_C01) ❹. It also offers the characteristic 0CALQUARTER from the SALES PLANNING

CUBE (BW_PLAN) ❺. You need to confirm this matching by selecting the checkbox ❻.

You can select the next characteristics from the MultiProvider using the NEXT icon ❼. This process needs to be completed for all of the characteristics included in the MultiProvider.

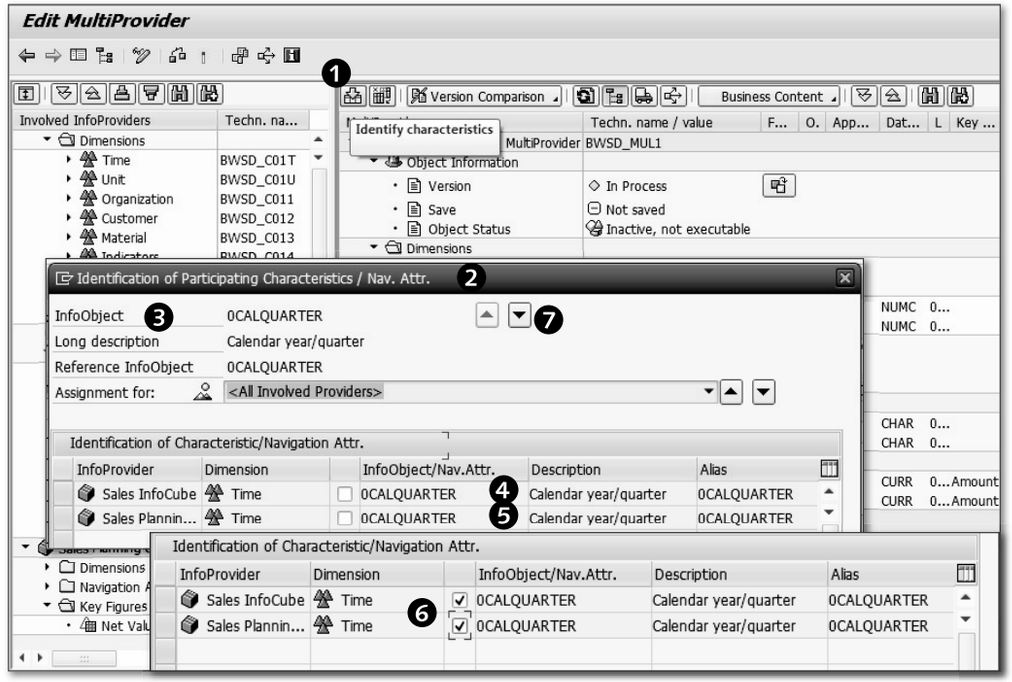


Figure 6.10 Edit MultiProvider: Identifying Characteristics

Now let's discuss the options for including characteristics in a MultiProvider, such as including characteristic 0DIVISION (see ❶ of Figure 6.11). 0DIVISION from the SALES INFOCUBE can be matched with 0DIVISION and BW_PROD__0DIVISION (0DIVISION is the navigation attribute of BW_PROD) from the SALES PLANNING CUBE ❷. You need to decide whether you want to match the 0DIVISION characteristics of InfoCube BWSD_C01 with the 0DIVISION characteristics of InfoCube BW_PLAN or with the 0DIVISION navigation attribute of BW_PROD. Select 0DIVISION characteristics from both of the InfoCubes ❸.

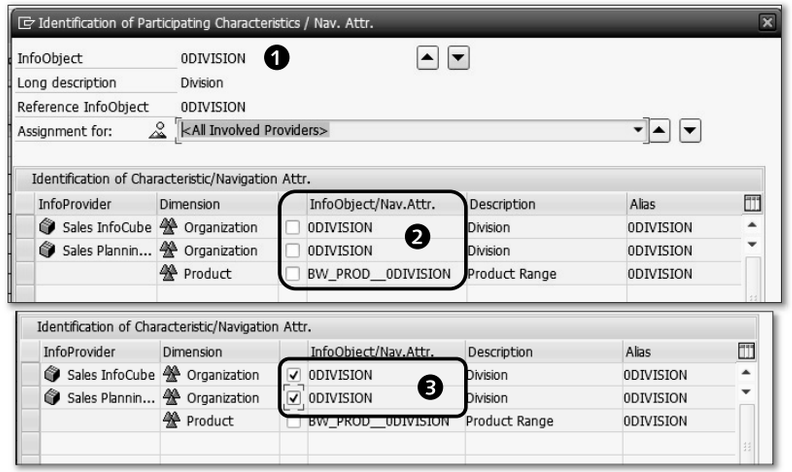


Figure 6.11 Edit MultiProvider: Identifying Characteristics from Multiple InfoCubes

You might also have a scenario where the characteristics can't be matched to characteristics or a navigation attribute in another InfoCube or data target included in the definition of a MultiProvider. For example, characteristic BW_PROD is included in the MultiProvider definition (❶ of Figure 6.12) and offered only from the Sales Planning Cube InfoCube ❷ because there's no matching characteristic available in the Sales InfoCube. In this case, you select the characteristics as we've shown in ❸ of Figure 6.12.

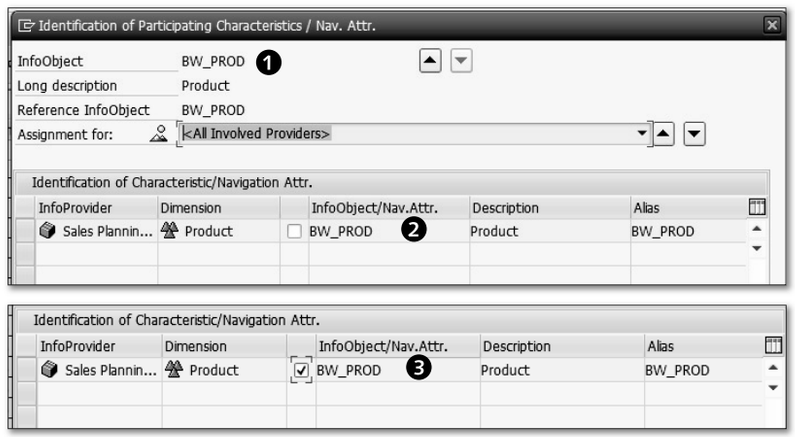


Figure 6.12 Edit MultiProvider: Identify Characteristics from Only One InfoCube

The final identification of characteristics is shown in Table 6.2.

MultiProvider Characteristics	Offer from InfoCube BWSD_C01	Offer from InfoCube BW_PLAN	Selected from InfoCube BWSD_C01	Selected from InfoCube BW_PLAN
0CALQUARTER	0CALQUARTER	0CALQUARTER	0CALQUARTER	0CALQUARTER
0CALYEAR	0CALYEAR	0CALYEAR	0CALYEAR	0CALYEAR
0SALESORG	0SALESORG	0SALESORG	0SALESORG	0SALESORG
0DIVISION	0DIVISION	0DIVISION BW_PROD__ 0DIVISION	0DIVISION	0DIVISION
BW_PROD	N/A	BW_PROD	N/A	BW_PROD
0MATL_GROUP	0MATL_GROUP	0MATL_GROUP BW_PROD__ 0MATL_GROUP	0MATL_GROUP	0MATL_GROUP

Table 6.2 Identification of Characteristics in a MultiProvider

Matching Key Figures

After all of the characteristics included in a MultiProvider are matched, you also need to match the key figures. Click on the Select Key Figures icon, as shown in ❶ of Figure 6.13. The Selection of Key Figures Involved box ❷ appears, and the system lists each key figure involved in the MultiProvider. The system offers the BW_VAL key figure ❸ from the Sales Planning Cube ❹. Because there's no matching key figure available in the Sales InfoCube, the system doesn't show any second key figures. Select BW_VAL ❺. You can go to the next key figure by clicking on the Next Object icon ❻.

The final identification of key figures is shown in Table 6.3.

MultiProvider Key Figure	Offer from InfoCube BWSD_C01	Offer from InfoCube BW_PLAN	Selected from InfoCube BWSD_C01	Selected from InfoCube BW_PLAN
BW_VAL	N/A	BW_VAL	N/A	BW_VAL
0NET_VALUE	0NET_VALUE	N/A	0NET_VALUE	N/A

Table 6.3 Identification of Key Figures

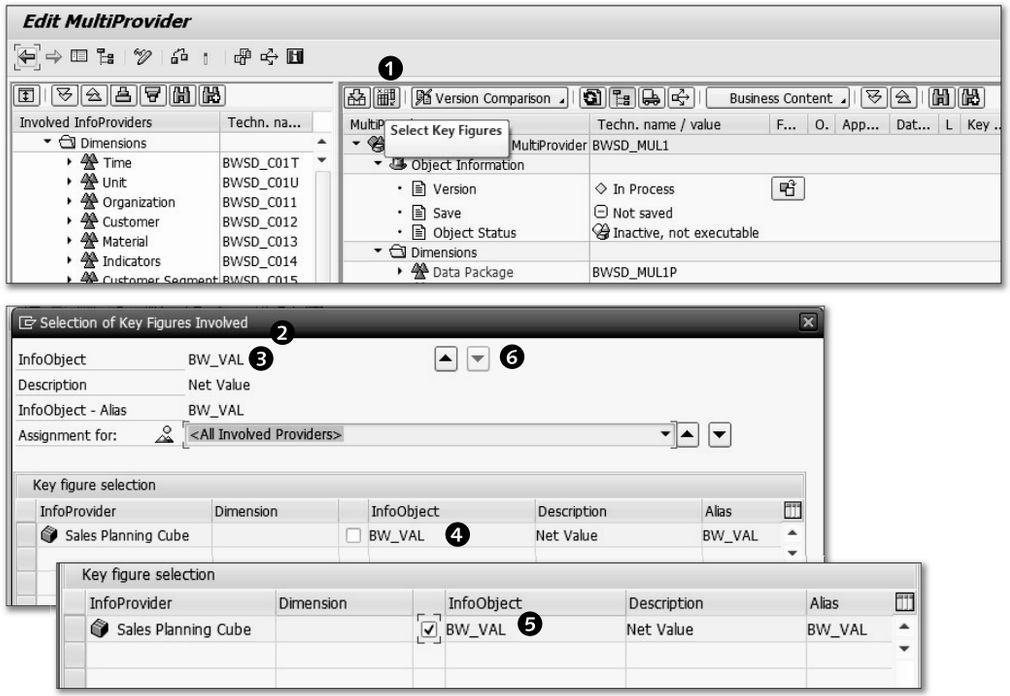


Figure 6.13 Edit MultiProvider: Selecting Key Involved

When identifying key figures, you can select key figures from both of the base InfoCubes (while including them in a MultiProvider). Selecting the same key figure (i.e., a key figure with the same technical name) from both of the InfoCubes would result in the summation of key figure values in the MultiProvider for the same values of characteristics.

Now click on the CHECK icon, as shown in ❶ of Figure 6.14, to check the definition of the MultiProvider. If everything is okay, the system reports a message ❷. At this moment, the MultiProvider is still not in ACTIVE status, so click on the ACTIVATE icon ❸. After it's successfully activated, the system changes the status of the MultiProvider to ACTIVE, EXECUTABLE ❹.

After successful activation of MultiProvider BWSD_MUL1, it's available under the BW INFOAREA, as shown in Figure 6.15.

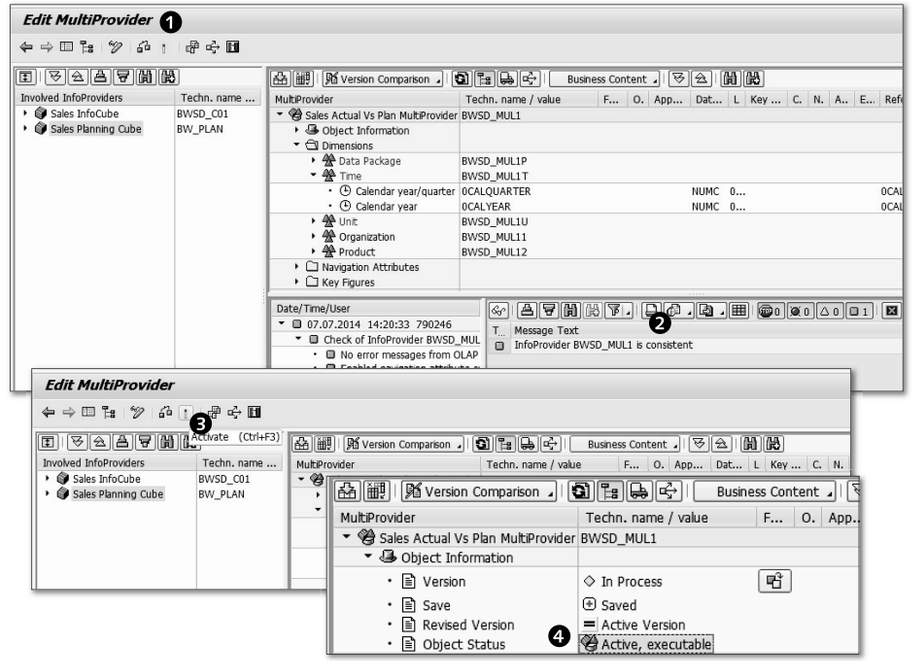


Figure 6.14 Checking and Activating the MultiProvider

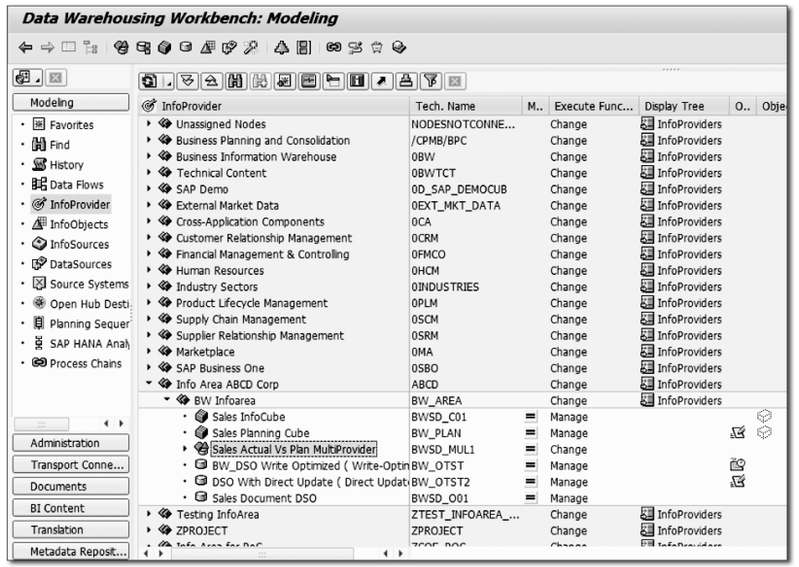


Figure 6.15 Activated MultiProvider Available under BW InfoArea

Now that we've covered the MultiProvider, let's move on to another InfoProvider: the InfoSet.

6.2 InfoSets

An InfoSet exists only as a logical definition, and thus doesn't physically store data. Queries can be created on an InfoSet using standard query tools supplied by SAP BW, and they get the data from underlying objects or sources. These sources can be InfoObjects (characteristics with master data), DSOs, or InfoCubes.

When it comes to InfoSets, you may be wondering how they differ from MultiProviders. Are they a subset of MultiProviders? Why does SAP BW have another object of the same nature? Can't we accomplish the same task by using a MultiProvider? The fundamental difference between the two is that MultiProviders use a union operation, while InfoSets use a join operation. Depending on your specific requirements, you may need either a MultiProvider or an InfoSet.

You can do most recent reporting using queries on an InfoSet. When you load master data into an InfoObject, the latest data loaded isn't immediately available for reporting because it's stored in the M (modified) version in the underlying tables associated with the InfoObjects. There is a process of activating master data, which turns an M version into an A (active) version. By default, only active version data is read to query, and only the InfoSet allows you to query on data that isn't active. The setting is shown in Section 6.2.5, where we explain the global properties of an InfoSet.

Usage Restriction

The following are restrictions in using an InfoSet:

- You can't define an InfoSet when the InfoCube is a right operand of a left outer join.
- SAP doesn't support InfoSets containing more than two InfoCubes.
- InfoCubes containing noncumulative key figures can't be part of an InfoSet.

In this section, we'll explain the types of joins, their implications and limitations, and the concept of transitive attribute reporting. We'll then describe the step-by-step procedure for creating InfoSets, including discussions of additional navigational capabilities and global properties.

6.2.1 Type of Joins

Using the join operation in InfoSets enables you to combine results from different underlying sources. In this section, we'll introduce you to the different types of joins found in InfoSets.

Inner and Left Outer Joins

By default, an InfoSet uses the inner join operation. An inner join checks the data in all underlying sources for the joining condition, and if it's available in all, it's passed to the result set.

InfoSets also offer outer joins, which can be used in some typical scenarios, such as for products that aren't being sold frequently (explained with an example later in this section). In this case, you can create an InfoSet based on InfoObject 0MATERIAL (which contains the material master data) and the DSO (in which material actual sales data is stored).

Although the basic idea of inner and left outer joins is simple, it's important to understand the difference between them. Let's look at an example.

Consider a scenario where an InfoObject for a customer (let's call it T1) has a sales office as an attribute. This sample data is shown in Table 6.4.

Customer Number	Sales Office
C1	SO1
C2	SO2
C3	SO3
C4	SO1
C6	SO2

Table 6.4 Customer InfoObject with Sales Office Attribute

The sales DSO (let's call it T2) contains the billing quantity information by customer and by month, as shown in Table 6.5.

Month	Customer Number	Billing Qty
10.2015	C1	100
10.2015	C2	120

Table 6.5 Data Sales DSO: Billing Quantity by Customer and by Month

Month	Customer Number	Billing Qty
10.2015	C3	150
10.2015	C5	140
10.2015	C6	110

Table 6.5 Data Sales DSO: Billing Quantity by Customer and by Month (Cont.)

When the InfoSet on T1 and T2 is created with an inner join on the customer number, the result looks similar to Table 6.6.

Customer Number	Sales Office	Month	Billing Qty
C1	SO1	10.2015	100
C2	SO2	10.2015	120
C3	SO3	10.2015	150
C6	SO2	10.2015	110

Table 6.6 Result of Making an InfoSet on T1 and T2 with an Inner Join

Note

The record for customers C4 and C5 isn't included in the result set because the data isn't available for these customers in both of the sources included in the InfoSet definition.

When the InfoSet on T1 and T2 is created with a left outer join on the customer number, while keeping the customer InfoObject on the left of the outer join, the result looks similar to Table 6.7.

Customer Number	Sales Office	Month	Billing Qty
C1	SO1	10.2015	100
C2	SO2	10.2015	120
C3	SO3	10.2015	150
C4	SO1	N/A	N/A
C6	SO2	10.2015	110

Table 6.7 Result of Making an InfoSet on T1 and T2 with a Left Outer Join

As you can see from these tables, there's a significant difference between the results produced depending on the type of join used, and it's essential to choose the right join for your reporting requirements.

As we've stated previously, the join selected between two InfoProviders included in an InfoSet definition is an inner join by default. However, you can change that to a left outer join if required. Select the InfoProvider T00002, and open the context menu, as shown in ❶ of Figure 6.16. Click on the Left Outer Join option ❷.

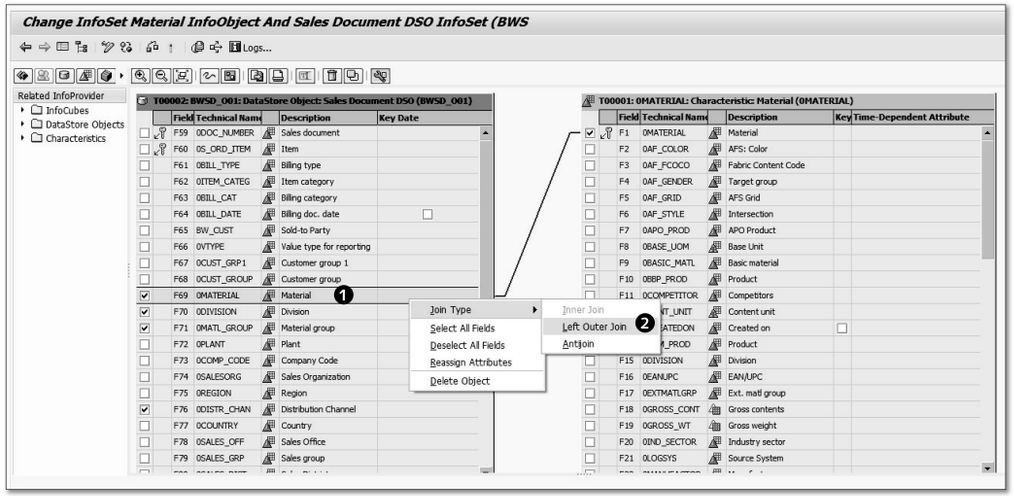


Figure 6.16 Selecting the Left Outer Join Option

The screen changes, as shown in Figure 6.17. Now the join shows the Left Outer Join.

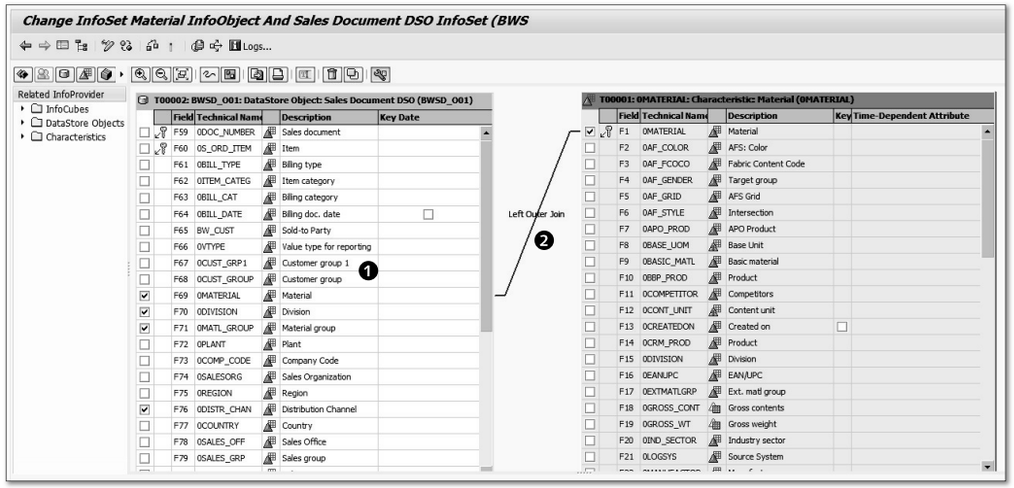


Figure 6.17 Left Outer Join in the InfoSet Definition

In this way, an inner join can be easily converted to a left outer join. Note, however, that where you select *from* is important. The selection process keeps the InfoProvider T00001 to the left side of the join. The query on this left outer join brings all of the rows from InfoProvider T00001, regardless of whether the matching value for the join condition (in our example, material) is available in InfoProvider T00002 or not.

Temporal Joins

InfoSets also offer a unique concept known as a *temporal join*, which is made available when an InfoSet is created with one of the InfoProviders that's included as an InfoObject with *time-dependent attributes*. In our example, InfoObject BW_CUST has 0SALESEMPLY as a time-dependent attribute. When we include InfoObject BW_CUST when creating the InfoSet, the temporal join can be used. We'll illustrate this with example data in Table 6.8 and Table 6.9.

Customer Number	Sales Office	Sales Employee	Valid From	Valid To
C1	SO1	EMP1	01-Jan-99	31-Dec-12
C1	SO1	EMP2	01-Jan-13	31-Mar-15
C1	SO1	EMP3	01-Apr-15	31-Dec-99
C2	SO2	EMP4	01-Apr-11	31-Mar-15
C2	SO2	EMP2	01-Apr-15	31-Dec-99
C3	SO1	EMP3	01-Jan-12	31-Dec-99

Table 6.8 Sample Master Data in InfoObject BW_CUST

Billing Document Number	Billing Date	Customer Number	Billing Qty
1234	25-Mar-15	C1	100
1235	25-Mar-15	C2	120
1236	25-Mar-15	C3	150
2115	03-Apr-15	C1	340
2116	03-Apr-15	C2	200
2117	03-Apr-15	C3	100

Table 6.9 Sample Data in DSO BWSD_O01

An InfoSet is created using InfoObjects BW_CUST and DSO BWSD_O01 with the customer number as the join between the two. The Billing Date field is used as a key date. (To understand how to set the key date, see the checkbox shown in ❹ of Figure 6.23, later in Section 6.2.3.)

The result of the query based on this InfoSet is shown in Table 6.10.

Billing Document Number	Billing Date	Customer Number	Sales Employee	Billing Qty
1234	25-Mar-15	C1	EMP2	100
1235	25-Mar-15	C2	EMP4	120
1236	25-Mar-15	C3	EMP3	150
2115	03-Apr-15	C1	EMP3	340
2116	03-Apr-15	C2	EMP2	200
2117	03-Apr-15	C3	EMP3	100

Table 6.10 Result of Query

As you can see, the value of the Sales Employee field is derived using the value of the billing date for each billing document number, as well as the value of the sales employee between the valid from and valid to dates for the same customer in BW_CUST.

Anti Joins

From SAP BW 7.3, a new type of join is introduced for InfoSets: the *anti join s*. An anti join is a type of join that is used to identify which data from an InfoProvider isn't contained in another InfoProvider. As an example, let's look at a material InfoObject that contains the data shown in Table 6.11 (for simplicity, only the material key is shown). Another DSO contains sales order data, as shown in Table 6.12 (for simplicity, only the sales order number information and the material information are shown).

Material
21000
31000

Table 6.11 Material Master Data for the Anti Join Example

Material
41000
51000
61000

Table 6.11 Material Master Data for the Anti Join Example (Cont.)

Sales Order Number	Material
100	21000
101	41000
102	21000
103	51000

Table 6.12 Sales Order Data for the Anti Join Example

When creating an InfoSet using these two tables, the DSO is defined as an anti join on the material field. When you execute the queries on an InfoSet that uses this anti join, you're provided with a list of the master data entries in the material field that don't appear in the DSO; in our example, these are 31000 and 61000 (Table 6.13).

Result Data
31000
61000

Table 6.13 Result of the Anti Join

6.2.2 Transitive Attribute Reporting

One of the capabilities of an InfoSet is allowing *transitive attribute reporting*, which is the process of reporting on a second-level attribute. As shown in Figure 6.18, InfoObject BW_CUST has several attributes, one of which is 0SALESEMPLY. In addition, InfoObject 0SALESEMPLY has its own attributes, one of which is 0CITY. Because of this relationship, 0CITY is known as a *transitive attribute* of BW_CUST.

While designing an InfoCube, you can switch on 0SALESEMPLY as the navigation attribute of BW_CUST and use 0SALESEMPLY in the query design wherever BW_

CUST is part of the InfoCube. But you can't use the attributes of 0SALESEMPLY (for example, 0CITY) in the query design when BW_CUST is part of the InfoCube.

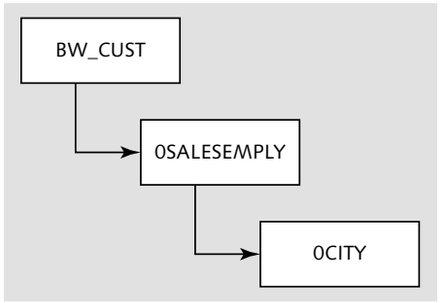


Figure 6.18 The Transitive Attribute of BW_CUST: 0CITY

In this scenario, you can use the InfoSet to get the information related to the transitive attribute.

6.2.3 Designing InfoSets

In the reference scenario, we have a requirement for analyzing which products of ABCD Corp. aren't getting sold in specific markets. This analysis requirement can be addressed using an InfoSet as explained next.

Creating InfoSets

To begin, open the Database Warehousing Workbench screen using Transaction RSA1. Under Modeling in the navigation section, select InfoProvider, as shown in ❶ of Figure 6.19. InfoSets are created under an InfoArea. From the tree section of the screen, select the InfoArea to which you want to attach your InfoSet ❷. In this example, BW Infoarea is selected.

Using the context menu of BW Infoarea, click on the Create InfoSet option ❸. The Create InfoSet box appears, as shown in Figure 6.20.

InfoSets require a unique technical name and description, as shown in ❶ of Figure 6.20. The technical name of the BW_AREA InfoArea in which the InfoSet would be grouped is displayed in the InfoArea field ❷. The Start With InfoProvider section ❸ offers three choices: DataStore Object, InfoObject, and InfoCube.

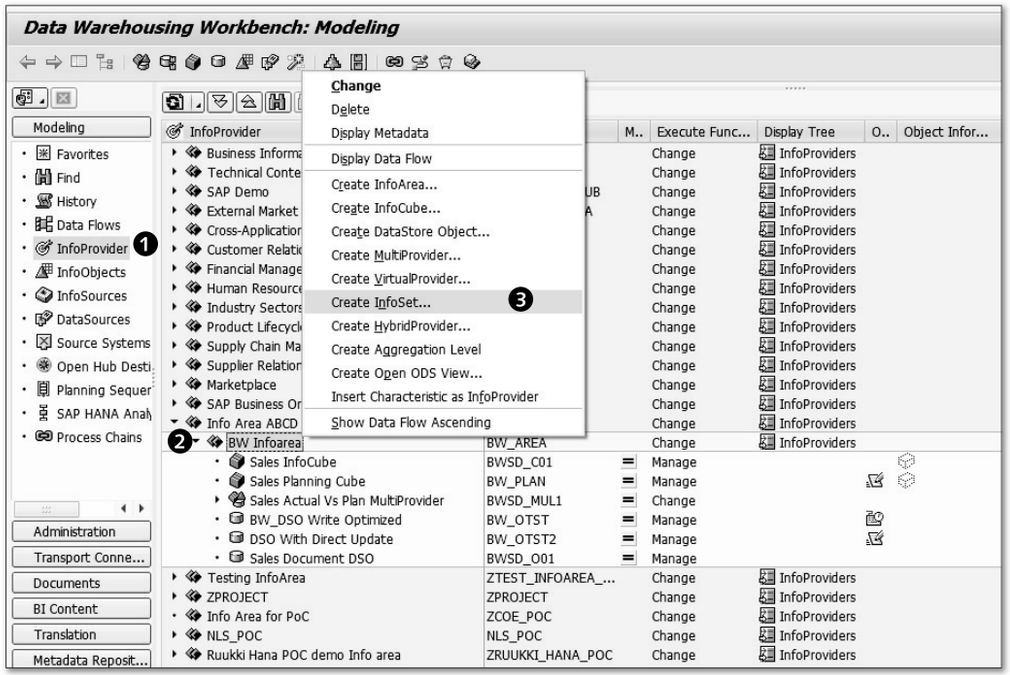


Figure 6.19 Creating the InfoSet

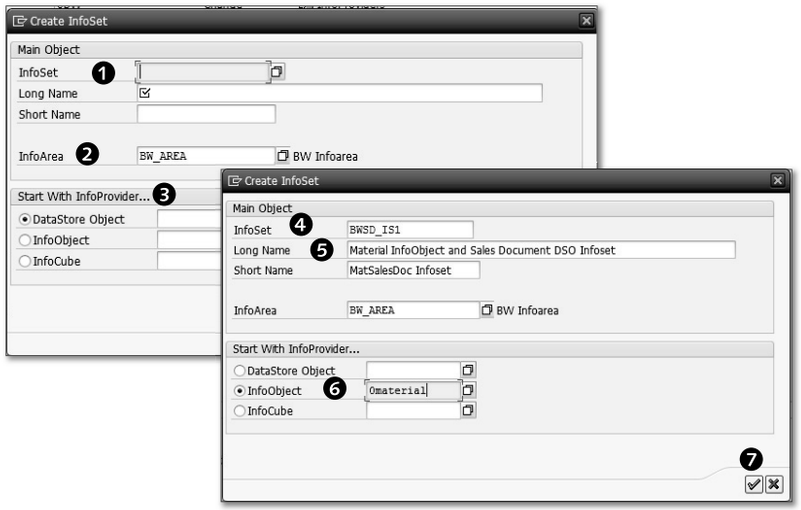


Figure 6.20 Initial Screen

In this example, we'll use the technical name "BWSD_IS1" ❹ and the description "Material InfoObject and Sales Document DSO Infoset" ❺. You can also choose to supply a short description, but only the long description is mandatory. We'll use "MatSalesDoc Infoset" in the SHORT NAME field.

The START WITH INFOPROVIDER area allows you to supply the technical name of the first source you want to include in the definition of InfoSet BWSD_IS1. In our example, we want to include an InfoSet based on InfoObject 0MATERIAL and DSO BWSD_O01. Select the INFOOBJECT radio button, and enter the name of the InfoObject "0material" ❻. Now click on the CONTINUE icon ❼. The resulting screen is shown in Figure 6.21.

InfoObject 0MATERIAL is included as part of InfoSet BWSD_IS1. SAP BW internally allocates the number T00001 to the first object that is included in this definition (see ❶ of Figure 6.21). For the subsequent objects you include, it gives the number T00002, and so on.

All attributes of InfoObject 0MATERIAL are shown in the vertical box format. The technical name of the attribute is shown in the TECHNICAL NAME column ❹.

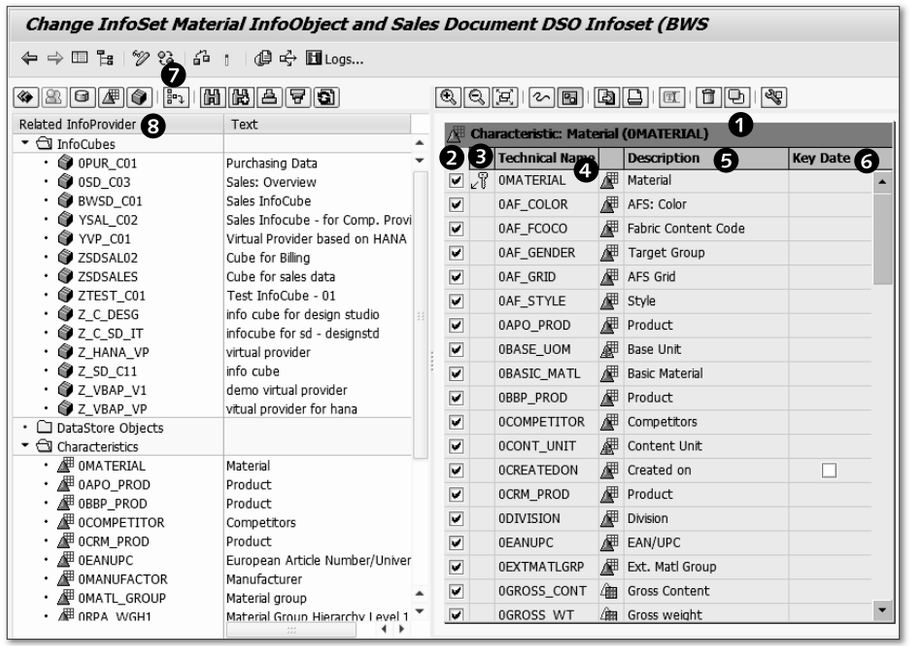


Figure 6.21 Understanding the Initial Screen

Not all columns have titles, so we've numbered them to explain their significance. Column ❷ of Figure 6.21 allows you to decide whether you want to include particular attributes of the 0MATERIAL InfoObject in the final definition of the InfoSet. By default, SAP BW includes all of the attributes. Column ❸ indicates whether the particular InfoObject is a key. For example, InfoObject 0MATERIAL is the key for InfoObject 0MATERIAL; for a DSO, all of the key fields are indicated as a key with a key icon. Columns ❹ and ❺ indicate the technical name and description of the attributes, respectively. Column ❻ indicates whether the field is included or excluded while defining the result for a temporal join. The checkbox is only shown when the field is a date field; for example, for field 0CREATEDON, the checkbox is shown, which is a date field.

The Related InfoProvider section ❽ gives the list of InfoObjects, DSOs, and InfoCubes that can be used to join 0MATERIAL; that is, all of the objects of which 0MATERIAL is a part and that are available for making an InfoSet join with InfoObject 0MATERIAL.

Defining InfoSets

Now you're ready to include your next source in the definition of InfoSet BWSD_IS1. Select the Insert InfoProvider icon, as shown in ❼ of Figure 6.21. The Insert InfoObject pop-up box appears, as shown in ❶ of Figure 6.22. Three choices (DataStore Object, InfoObject, and InfoCube) are available as InfoSet definitions and can include either of the objects. Our example InfoSet is based on InfoObject 0MATERIAL and DSO BWSD_O01.

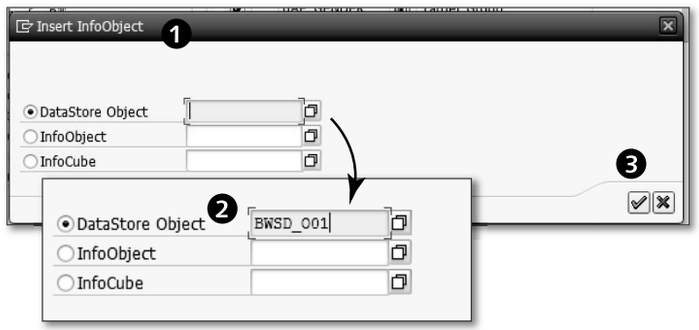


Figure 6.22 Inserting a Second Source

Next, select the DataStore Object radio button (if not selected already), and enter the technical name "BWSD_O01," as shown in ❷ of Figure 6.22. Now click on the Continue icon ❸.

The screen now changes, as shown in Figure 6.23. The DSO BWSD_O01 is displayed ❶. This time, the technical number attached to DSO BWSD_O01 is T00002, and column ❷ shows the Key icon for 0DOC_NUMBER and 0S_ORD_ITEM. These two InfoObjects are configured as key fields of DSO BWSD_O01. Column ❸ shows the technical name of the InfoObjects of DSO BWSD_O01. Column ❹ shows the key date.

Now both sources are available in the screen. We need to decide which fields to include in the InfoSet definition, and how to join the InfoSet. There are two ways you can select/deselect fields in the InfoSet definition. The first way is to select the individual checkbox to exclude or include it in the definition of the InfoSet. By default, SAP BW includes all of the fields in the definition of the InfoSet, so you need to uncheck the checkbox in the first column. This may be cumbersome when you want to include only a few required fields in the definition of an InfoSet.

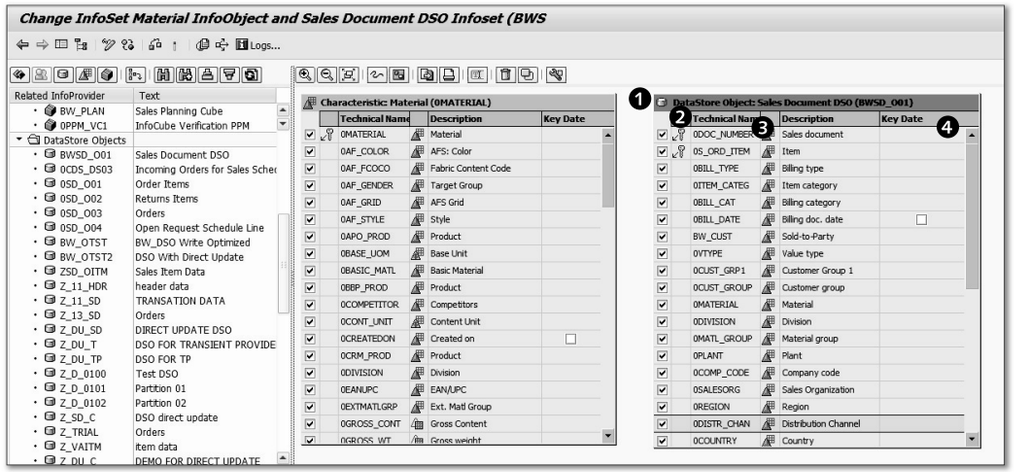


Figure 6.23 Second Source Inserted

The second way is to deselect all of the checkboxes using one single click and include only those that are required. First, open the context menu by right-clicking

anywhere on the source object for which you want to deselect all of the fields. For example, we want to deselect all of the fields for the 0MATERIAL source object. Open the context menu by right-clicking somewhere near the spot shown in ❶ of Figure 6.24, and then select DESELECT ALL FIELDS ❷. You can perform similar steps on the BWSD_O01 source object and deselect all of the fields of DSO BWSD_O01.

All of the fields from both included source objects are now deselected. Now select only the required fields by selecting the checkbox available in the first column. As shown in ❸, we've selected only fields 0MATERIAL, 0MATL_CAT, and 0MATL_TYPE from source object T00001 (all fields aren't displayed in this screen).

In a similar way, the fields 0MATERIAL, 0DIVISION, 0MATL_GROUP, 0DISTR_CHAN, 0SALES_OFF, BW_QTY, and 0UNIT are selected from source object BWSD_O01 ❹.

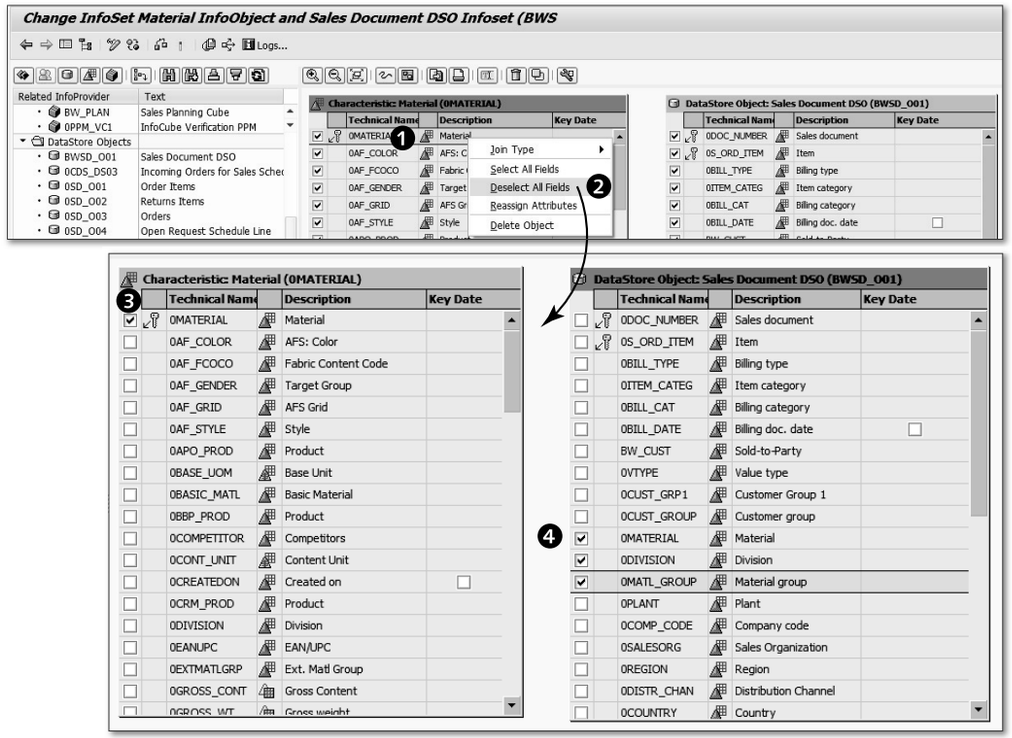


Figure 6.24 Selecting Required Fields

Joining Source Objects

The next step after selecting the fields is to join the two source objects. You can join a field from one source object to the same field or a field with a similar data type and length of another source field. This process is very simple. In our example, we want to join the 0MATERIAL field from the T00001 source object to the 0MATERIAL field of the T00002 source object. Select the 0MATERIAL field of T00001, keep the left mouse button pressed, and drag the mouse over to the 0MATERIAL target field of T00002. The dragged mouse icon pointer changes from the normal pointer icon to a pencil icon. Drop it on the target field.

SAP BW now creates a join between the 0MATERIAL field of the T00001 source object and the 0MATERIAL field of the T00002 source object. This is shown as a link from ❶ to ❷ in Figure 6.25. Internally, SAP BW makes this a join condition similar to *T00001.0MATERIAL = T00002.0MATERIAL*.

If required, you can create another join condition between the two fields of T00001 and T00002; however, our example scenario doesn't require another join condition. This completes the steps required to configure an InfoSet.

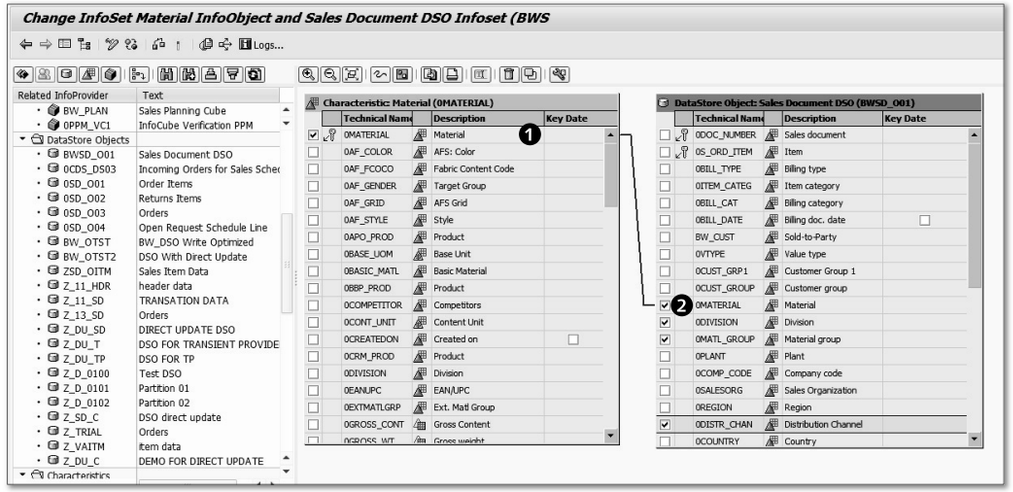


Figure 6.25 Joining Two Sources

Checking and Activating

Now check the definition using the CHECK icon, as shown in ❶ of Figure 6.26.

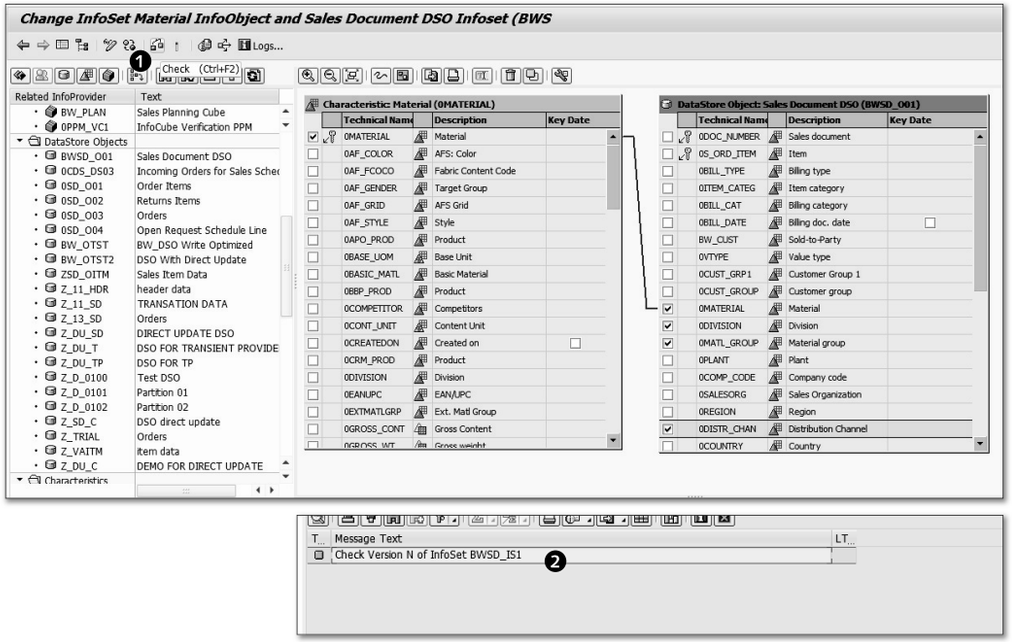


Figure 6.26 Checking the Definition of an InfoSet

Checked messages are available at the bottom of the screen, as shown in ❷ of Figure 6.26. Because the checking of InfoSet BWSD_IS1 hasn't resulted in any error messages, we can now activate the InfoSet definition in the SAP BW system.

To activate, click on the ACTIVATE icon , as shown in ❶ of Figure 6.27. Activation messages are shown in the lower section of the screen in a separate area ❷.

The final activation message is shown at the bottom of the screen ❸. Now the definition of the InfoSet is available for further use, for example, for creating a query based on this InfoSet or for creating a new MultiProvider that includes InfoSet BWSD_IS1.

As you can see in Figure 6.28, the newly created InfoSet BWSD_IS1 is available under BW INFOAREA BW_AREA.

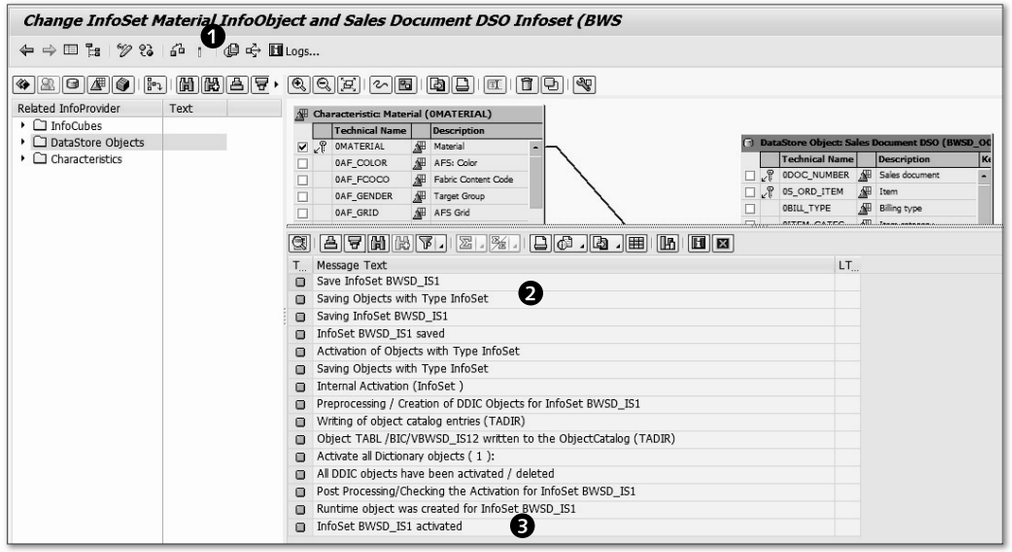


Figure 6.27 Activating the InfoSet

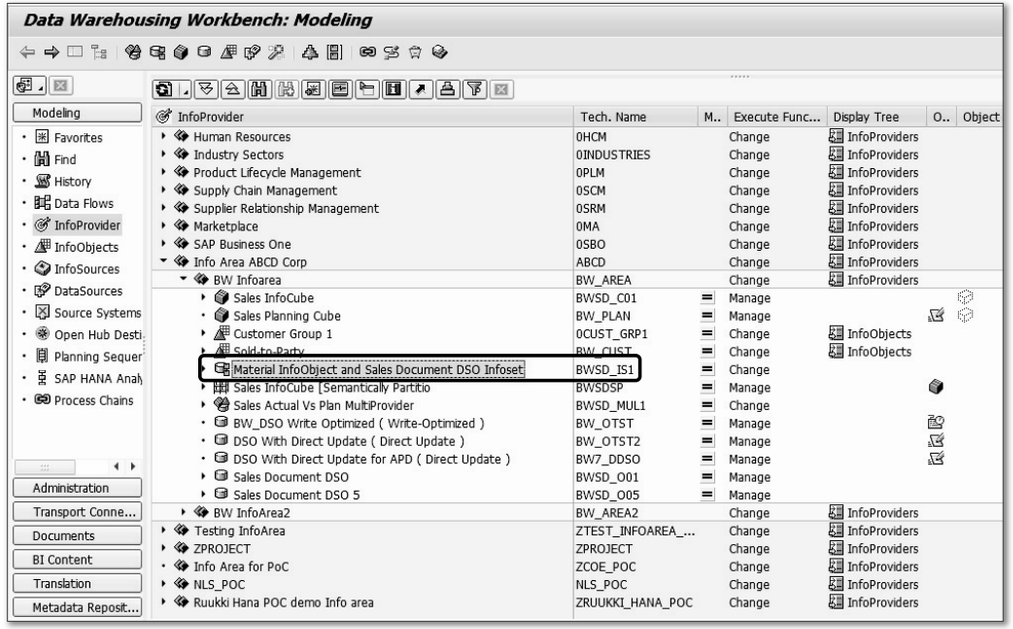


Figure 6.28 InfoSet Available in the Database Warehousing Workbench Screen

Changing the InfoSet Visual Setting

The InfoSet creation screen offers two types of display:

- Network display
- Tree display

So far we've seen the network display. You can change between the two settings by using the menu path SETTINGS • DISPLAY. The SETTING: INFOSET MAINTENANCE box appears, as shown in Figure 6.29. Select the HIERARCHY DISPLAY (TREE CONTROL) radio button ❶. Click on the CONTINUE icon ❷.

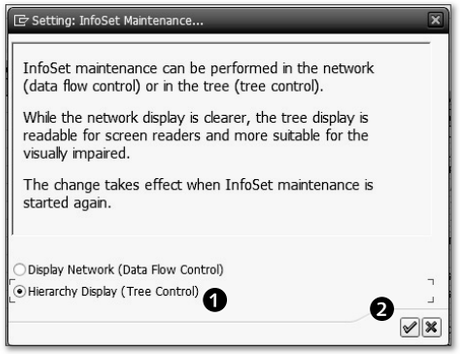


Figure 6.29 Changing the InfoSet Maintenance Screen Setting

The effect of this setting is visible the next time you open the InfoSet from the DATABASE WAREHOUSING WORKBENCH screen for editing. The screen should look like Figure 6.30.

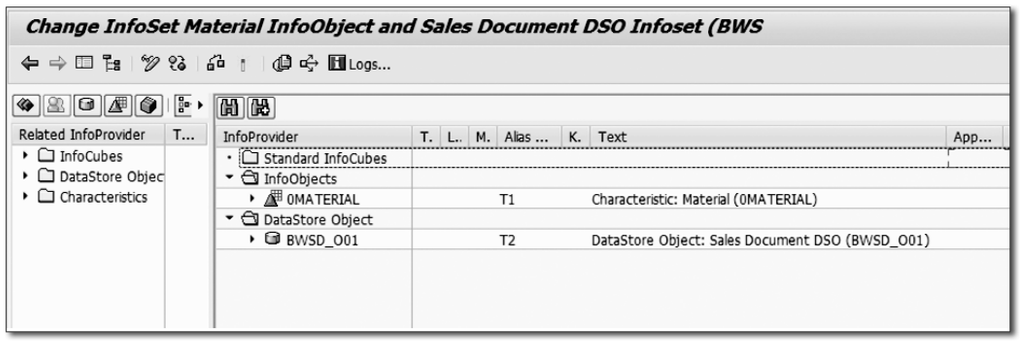


Figure 6.30 InfoSet Display in Tree Mode

The screen shown in Figure 6.30 is similar to what is explained in the LINK MAINTENANCE screen in the following section. The rest of the functionality remains the same.

6.2.4 Additional Navigation Capabilities

There are a few settings available for creating and editing an InfoSet. These settings are typical for the InfoSet only, so they are explained in detail in this section. Start the InfoSet definition in change mode. In the DATABASE WAREHOUSING WORKBENCH screen (Transaction RSA1), under MODELING in the navigation section, select INFOPROVIDER, as shown in ❶ of Figure 6.31. Select BW INFOAREA BW_AREA ❷. Using the context menu of InfoSet BWSD_IS1 ❸, select the CHANGE option ❹. The CHANGE INFOSET screen appears (Figure 6.32).

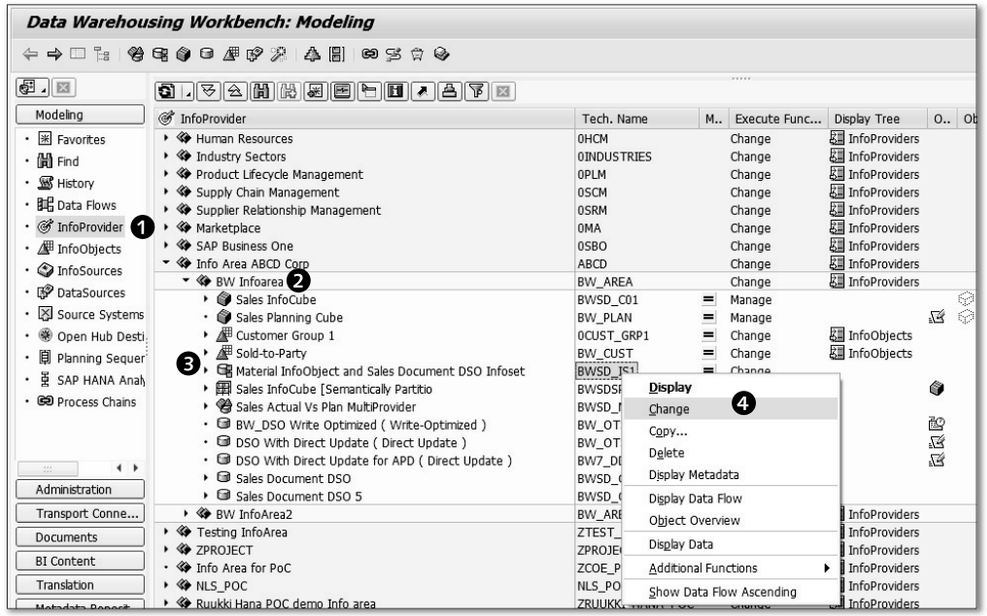


Figure 6.31 Editing the InfoSet

Auto Arrange

As you can see in ❶ of Figure 6.32, the join between two source objects isn't properly visible. Click on the AUTO ARRANGE icon ❷. The resulting screen is shown in the lower part of Figure 6.32, where you can view the join properly ❸

and ❹. In the process, you may also observe that before executing AUTO ARRANGE, source object T00001 on the left side of the screen has moved to the right, and source object T00002 has moved to the left of the screen. This movement doesn't affect the functionality of the InfoSet.

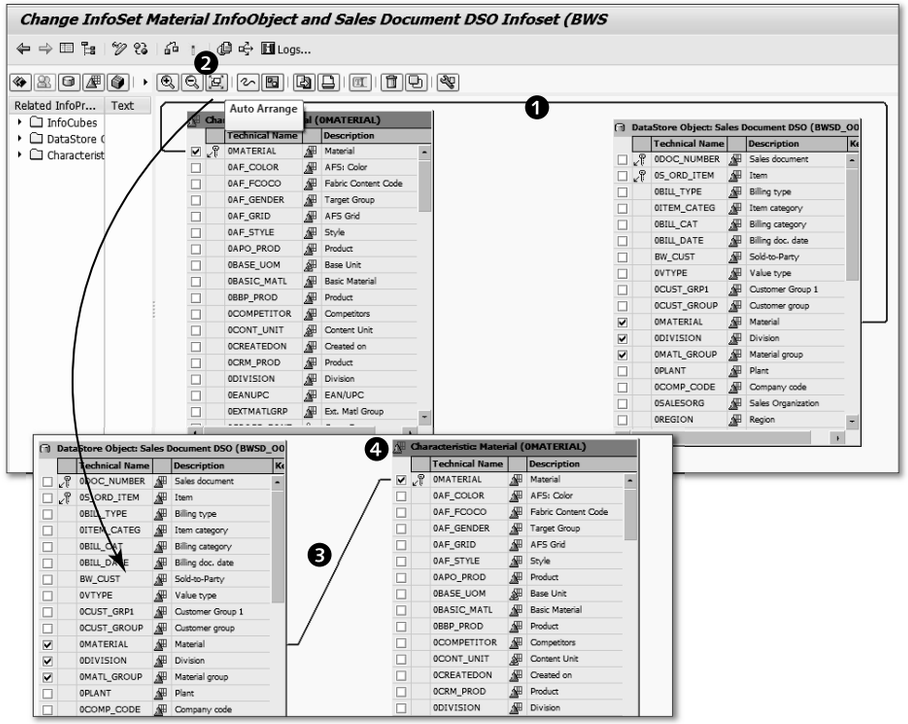


Figure 6.32 InfoSet Editing: Auto Arrange Facility

Navigator

When you're in the InfoSet change screen, sometimes all of the source objects won't fit on one screen. For example, as shown in Figure 6.33, the rightmost column (e.g., KEY DATE) isn't visible. There are three ways to adjust this view. The first method is to use the standard horizontal scroll bar available on any SAP BW screen, and scroll to the right side. The second method is to use the ZOOM OUT icon, which reduces the size of the font to accommodate more visible area on the same screen. There is also a ZOOM IN icon that increases the size of the font to accommodate less visible areas on the same screen with more clarity.

The third method is to use the NAVIGATOR function. Click on the SHOW/HIDE NAVIGATOR icon, as shown in ❶ of Figure 6.33. The result is that SAP BW adds a small box titled NAVIGATION ❷.

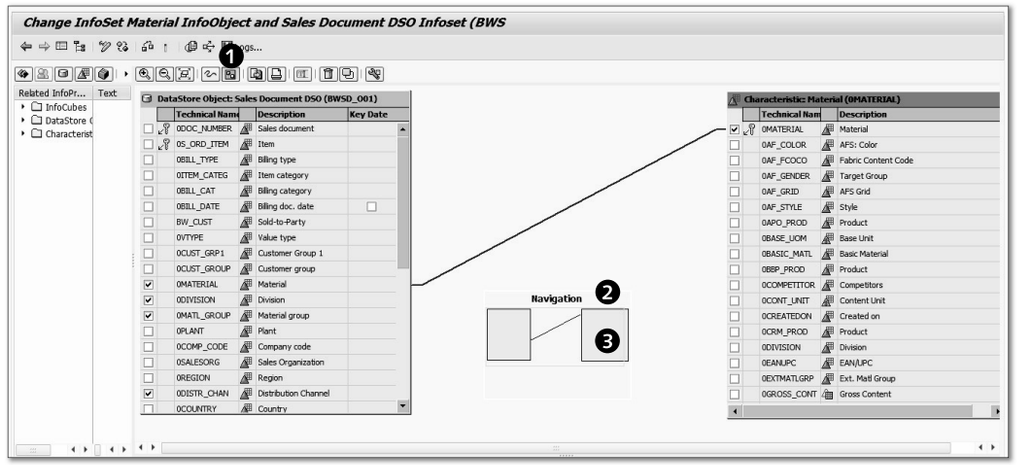


Figure 6.33 Using the Navigator Function

Here you can drag the navigation square box ❸ to the rightmost side of the NAVIGATION box. The resulting screen is shown in ❶ of Figure 6.34. The content on the extreme right-hand side of the InfoSet definition screen is now visible ❷.

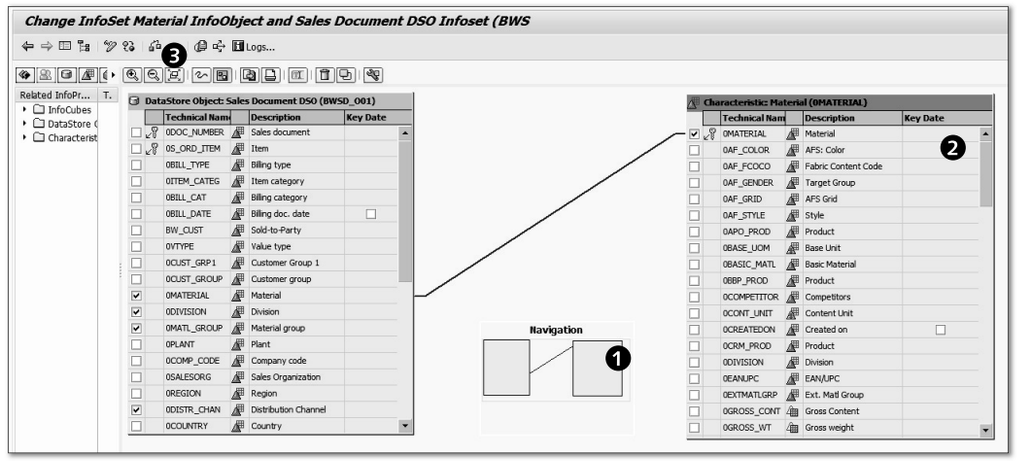


Figure 6.34 Result of Using the Navigator

Link Maintenance

When an InfoSet definition has two or more source objects and multiple joins, it may become difficult to move up and down or left and right to maintain the different joins involved in the definition of the InfoSet. An alternative way to maintain the join is by using link maintenance. Click on the Link Maintenance icon, as shown in ❸ of Figure 6.34. The Link Maintenance pop-up box appears (Figure 6.35).

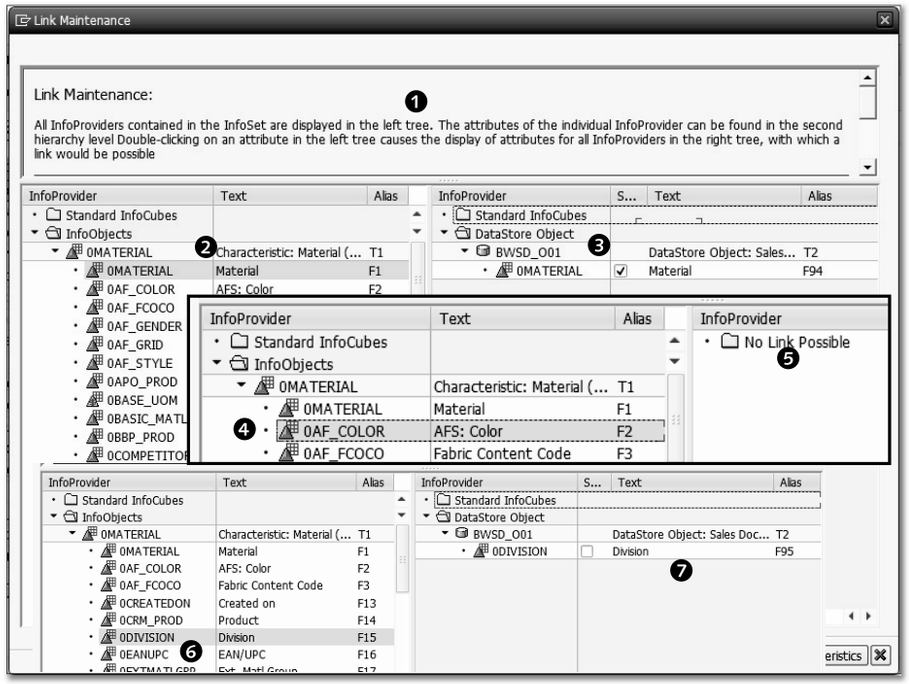


Figure 6.35 Link Maintenance

A brief description regarding the link maintenance functionality is shown in ❶ of Figure 6.35. A list of the included InfoProviders in the definition of the InfoSet is also given ❷. The right side panel is initially empty.

Expand the definition of InfoProvider 0MATERIAL, and double-click on the 0MATERIAL field ❷. The result of this action is shown on the right panel. The right panel now shows the 0MATERIAL field of DataStore object BWSD_O01 ❸.

This is the join we've created in our InfoSet, and the link maintenance functionality shows it in this simple way.

Double-click on the 0AF_COLOR field of InfoProvider 0MATERIAL ❹. This time, there's no matching field available in DataStore object BWSD_O01; on the right side, it's displaying No Link Possible ❺.

Double-click on the 0DIVISION field of InfoProvider 0MATERIAL ❻. On the right side panel, the 0DIVISION field of InfoProvider BWSD_O01 is displayed ❼. You can select the checkbox to create a join between the 0DIVISION field of InfoProvider 0MATERIAL and the 0DIVISION field of InfoProvider BWSD_O01.

Because our example InfoSet doesn't require this join, we'll leave it untouched.

6.2.5 Global Properties

A few properties are known as global properties for an InfoSet that can be accessed using the menu path GoTo • Global Properties. The InfoSet BWSD_IS1: Global Properties box appears, as shown in Figure 6.36. All four available settings are explained in this section.

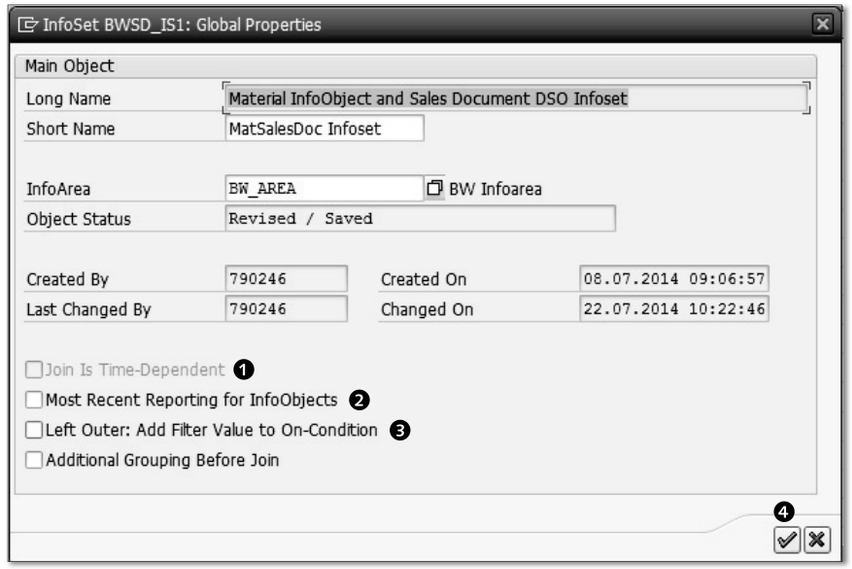


Figure 6.36 InfoSet Global Properties

Join Is Time-Dependent

This setting is display only (❶ of Figure 6.36). If any of the underlying InfoProviders included in the definition of the InfoSet is an InfoObject with a time-dependent attribute, this checkbox is checked, and the join created is known as a *time-dependent join*. In our example InfoSet, we've included InfoObject 0MATERIAL, but 0MATERIAL doesn't have any attributes that are time dependent, so this checkbox isn't available.

Most Recent Reporting for InfoObjects

An InfoSet can include an InfoObject as one of its underlying InfoProviders. In SAP BW, InfoObjects are used to store master data. The newly created master data (along with updated master data) is regularly loaded into this InfoObject. SAP BW doesn't make this newly loaded master data available to the queries right away. Newly loaded master data is first saved as an M version (modified version) in the underlying tables associated with the InfoObject. The data is only available for query after it's modified to become an A version (active version). To make newly loaded data active, you must process the attribute hierarchy change run, which is a normal part of the process when regular data loading takes place.

When you have a scenario where master data is loaded a number of times in SAP BW, there may be a delay in activating the newly loaded master data. However, using the settings available for InfoSets, you *can* report on this data, even if it isn't activated. To make this happen, check the Most Recent Reporting for InfoObjects setting, as shown in ❷ of Figure 6.36.

The first time the master data is loaded and activated, it's shown in Table 6.14. For the sake of simplicity, only a few columns are shown. Technically, SAP BW does have a few more columns to manage versions.

Customer Number	Version	City
C1	A	CITY-1
C2	A	CITY-2
C3	A	CITY-3

Table 6.14 Available Master Data

Now, the city of one customer has changed, and the newly loaded data (without the activation process) is shown in Table 6.15.

Customer Number	Version	City
C1	A	CITY-1
C2	A	CITY-2
C2	M	CITY-4
C3	A	CITY-3

Table 6.15 Data Loaded to Master: Activation Not Performed

If the InfoSet is queried with the configuration Most Recent Reporting for InfoObjects switched off, the result will be the data shown in Table 6.16.

Customer Number	City
C1	CITY-1
C2	CITY-2
C3	CITY-3

Table 6.16 Most Recent Reporting Not Configured

If the InfoSet is queried with the configuration Most Recent Reporting for InfoObjects switched on, the result will be the data shown in Table 6.17.

Customer Number	City
C1	CITY-1
C2	CITY-4
C3	CITY-3

Table 6.17 Most Recent Reporting Configured

Left Outer: Add Filter Value to On-Condition

This indicator (❸ of Figure 6.36) is useful if you've used a *left outer join* when defining the join in the InfoSet. It's used to control how a condition on a field of the left outer table is converted into a SQL statement while the query based on this InfoSet is executed. When the indicator is set, the condition is evaluated before the join. When the indicator isn't set, the condition is evaluated after the join.

The query result is different in both cases. You need to evaluate your reporting requirement and set the indicator accordingly.

Additional Grouping Before Join

By flagging the option for Additional Grouping Before Join, the system is informed about the grouping of the key figures. When this flag isn't set, key figure grouping doesn't take place until the table contents are joined.

6.3 VirtualProviders

Businesses often want or need to analyze information in real time, which requires establishing a real-time connection with the data from a source system. This can be accomplished without storing data in data targets such as InfoCubes or DSOs because SAP BW has objects called *VirtualProviders*.

Like the MultiProvider and the InfoSet, VirtualProviders exist only as a logical definition. However, it's conceptually different from its counterparts because it doesn't source the data from existing data targets within the data warehouse but accesses the data in real time from the sources (see Figure 6.37).

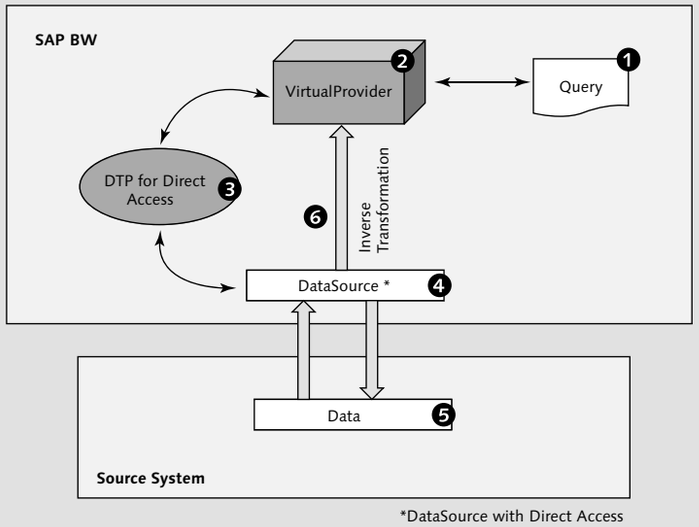


Figure 6.37 Illustration of VirtualProvider

We'll now explain the processes that enable reporting using a VirtualProvider. For instance, if we execute a query (see ❶ of Figure 6.37), the query is based on a

VirtualProvider ❷. The data for this query is then accessed in real time from the source system, using the direct access type data source ❹. The data is read from the source system ❺ and passed back to the query using the transformation ❻ and DTP for direct access ❸. Every navigation step in the query is a new process for reaching back to the source system for acquiring the relevant data set and relaying it to the query. The entire process of ETL happens in real time, and the data is converted to information and presented in the report. You can include the query variables to filter the data to be read from the source system. When a query is executed, the variable values are passed to the DataSource using inverse transformation.

The process of creating a VirtualProvider is very similar to creating a basic InfoCube (see Chapter 5, Section 5.3). A radio button is available for defining a VirtualProvider with four available options:

- Based on the Data Transfer Process for Direct Access
- Based on BAPI
- Based on Function Module
- Based on A HANA Model (applicable for SAP BW on HANA)

Note that conditions determine the applicability of a VirtualProvider. For instance, VirtualProviders should be used for scenarios that deal with a small data set, and the usage should be limited to a few users because the system has to execute the entire process from sourcing to presentation in real time.

VirtualProviders in SAP BW Powered by SAP HANA

In an SAP BW system powered by SAP HANA, VirtualProviders are also used to consume SAP HANA models in SAP BW.

6.4 Semantically Partitioned Objects (SPOs)

There may be scenarios where you're planning to store partitioned data based on certain characteristic values, such as data stored in identical InfoCubes, one for each sales organization, or data partitioned based on year. Prior to SAP BW 7.3, you could do this by creating these InfoCubes (one for each value of semantic partition) and then individually creating the transformations and DTPs for each of them, with one MultiProvider on top of it all. SAP provides a specific InfoProvider for this scenario: the *semantically partitioned object* (SPO). In this case, you

simply define a reference data model, which can be a standard InfoCube or DSO, for the data target. Then you define the criteria for the semantic partition and the transformations. With this information, the SAP BW system automatically generates the partitioned data targets and corresponding transformations and DTPs. Thus, using an SPO eliminates the need for the creation of multiple smaller data targets with the same definition. Only the reference data model needs to be maintained, and the changes are reflected in the partitions automatically.

6.4.1 Creating SPOs

Semantic partitioning is a property of the InfoProvider, so you can define a standard InfoCube or a DSO as an SPO at the time of creating the data target. This step-by-step procedure is explained next.

An SPO is created in the Database Warehousing Workbench screen, which is started using Transaction RSA1. Select InfoProvider under Modeling from the navigator section, as shown in ❶ of Figure 6.38.

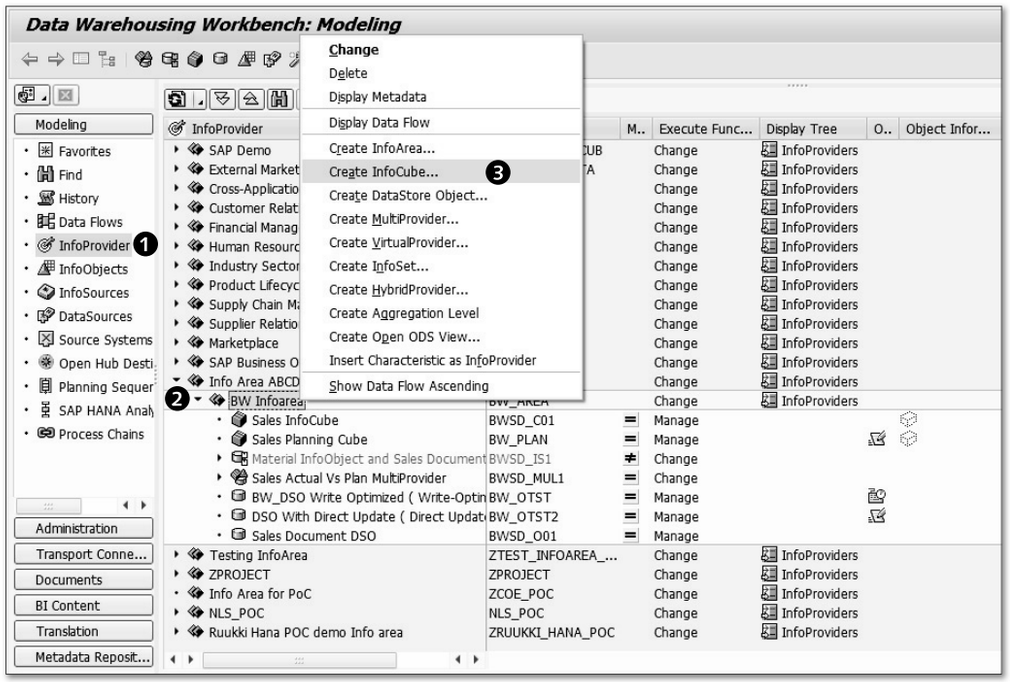


Figure 6.38 Creating an SPO: Part 1

Select the InfoArea under which you want to create your SPO ❷. From the context menu, select Create InfoCube ❸. The Create InfoCube screen appears, as shown in Figure 6.39.

Enter the technical name and description of the SPO (❶ of Figure 6.39). In our example, the SPO's technical name is "BWSDSP", and its description is "Sales InfoCube [Semantically Partitioned]". You can also use an existing InfoCube as a template to create a new SPO by entering the name of the existing InfoCube in the Template field. In our example, InfoCube "BWSD_C01" is used as a template ❷.

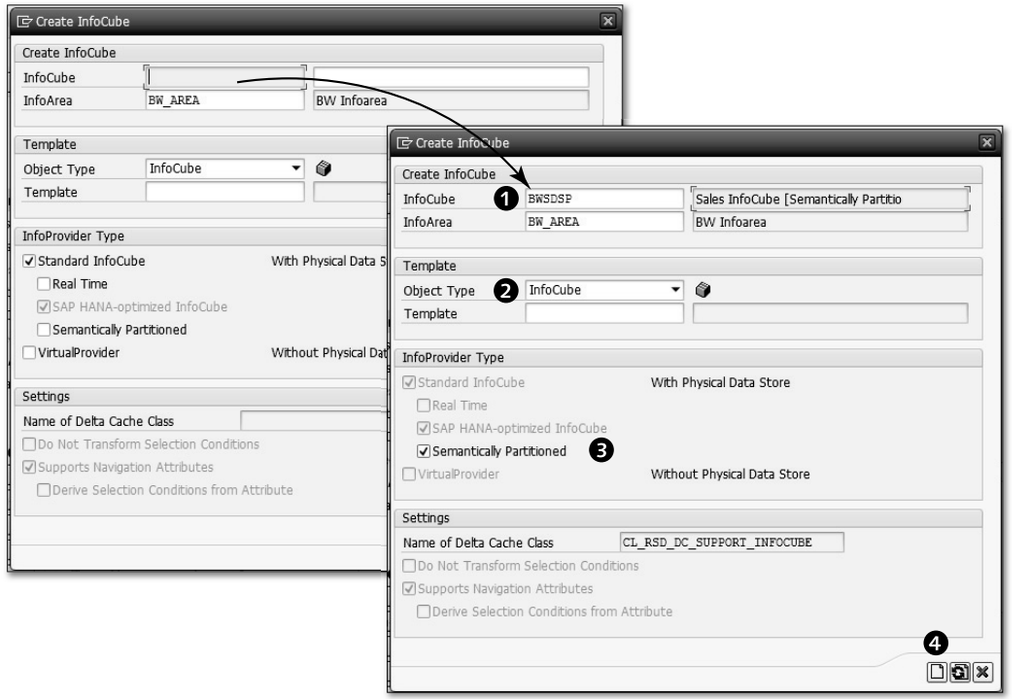


Figure 6.39 Creating an SPO: Part 2

Because this is a semantically partitioned InfoCube, select the Semantically Partitioned property, as shown in ❸ of Figure 6.39. Click on the Create icon ❹. This takes you to the screen where you define the reference data model for the SPO (Figure 6.40).

6.4.2 Defining SPOs

In the left panel shown in ❶ of Figure 6.40, there are six steps to define and configure an SPO:

- Maintain Object
- Maintain Partitions
- Start Activation
- Create Transformation
- Create Data Transfer Processes
- Create Process Chains

We consolidate our discussion of these steps into three major steps: managing partitions, creating transformations and DTPs for SPOs, and including SPOs in process chains.

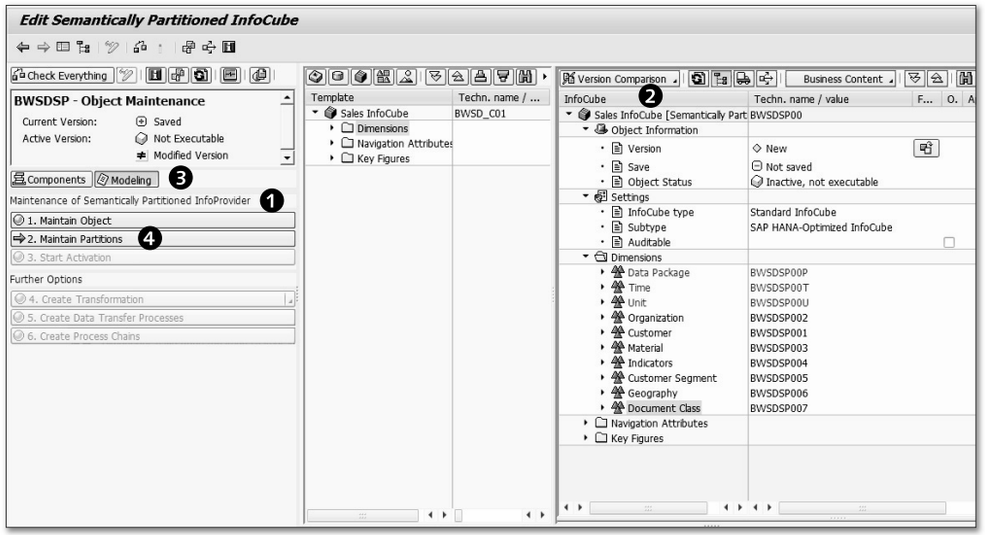


Figure 6.40 Editing an SPO

Managing Partitions

The right side of the screen, as shown in ❷ of Figure 6.40, displays the definition of InfoCube BWSD_C01, which is taken as the template. The system first creates the reference structure using InfoCube BWSD_C01 as the template. The system

names this reference structure BWSDSP00 and refers to it when a new partition for an SPO is designed. All of the dimensions, navigational attributes, and key figures of BWSD_C01 are available here. As shown in ❸ of Figure 6.40, the Modeling section is selected by default. The Component section displays the included components of the SPO. Click on Maintain Partitions ❹ to open the Select Partitioning Criteria screen, as shown in Figure 6.41.

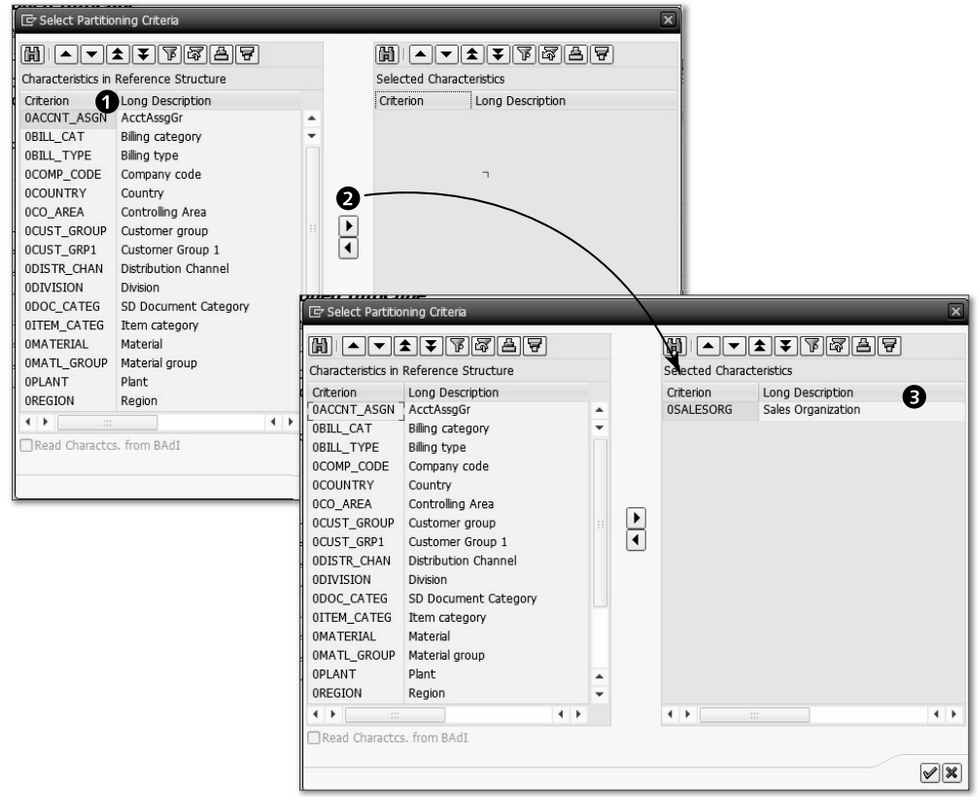


Figure 6.41 Partitioning Criteria

All of the characteristics of reference structure BWSDSP00 are displayed here (❶ of Figure 6.41). Select characteristics from the left-hand side that should be used to partition the data semantically; for our example, we use characteristic 0SALESORG (Sales Organization). Select 0SALESORG, and click on the Transfer icon ❷. Characteristic 0SALESORG is now available under Selected Characteristics ❸. Click on the OK icon ❹ to continue with the definition of the SPO. This brings

you to the MAINTENANCE OF CRITERIA FOR PARTITIONED OBJECT screen, as shown in Figure 6.42.

As shown in ❶ of Figure 6.42, the first partition is created by default and given the name PARTITION 01. Characteristic 0SALESORG is included as a partitioning characteristic ❷. By default, SINGLE VALUE is selected ❸, and a blank value is assigned to this partition ❹.

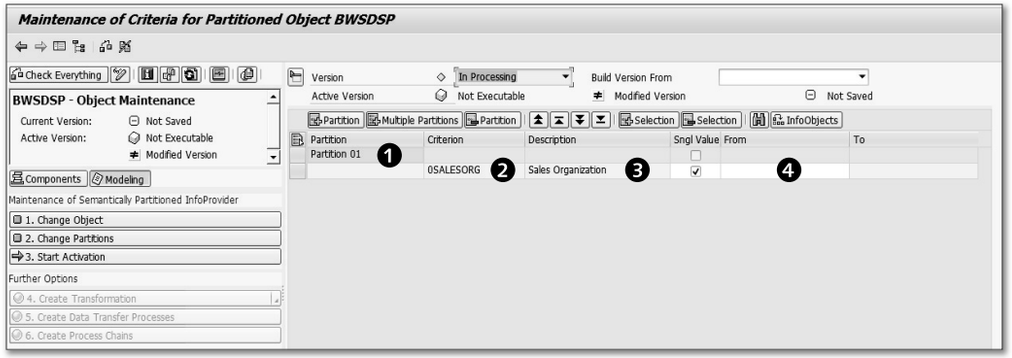


Figure 6.42 Creating Partitions

Our example needs three partitions based on sales organization 1000 (first partition), 2000 (second partition), and 3000 to 9999 (third partition). The procedure to configure this is shown in Figure 6.43.

Rename the first partition "SO 1000," as shown in ❶ of Figure 6.43. Also assign the value 1000 to sales organization in the FROM column ❷. To add a new partition, click on the PARTITION button ❸. This action adds one more blank partition, PARTITION 02 ❹. Rename this to "SO 2000" with the value 2000 assigned to it, as shown in ❺ and ❻. To add the third partition, repeat the process. Rename this new partition "Other SO" ❽, remove the SINGLE VALUE flag ❾, assign the value 3000 in the FROM column ❿, and the value 9999 in the TO column. The final screen should look like the one shown in Figure 6.44.

While configuring partitions, the system offers a few facilities. In a scenario where you need to add multiple partitions at once, click on the MULTIPLE PARTITIONS button (❶ of Figure 6.44). Using the DELETE PARTITION button ❷, you can remove a partition. The ADD SELECTION button ❸ allows you to add selections to

an existing partition; the DELETE SELECTION button ❹ allows you to remove selections from an existing partition.

Figure 6.43 Adding Partitions

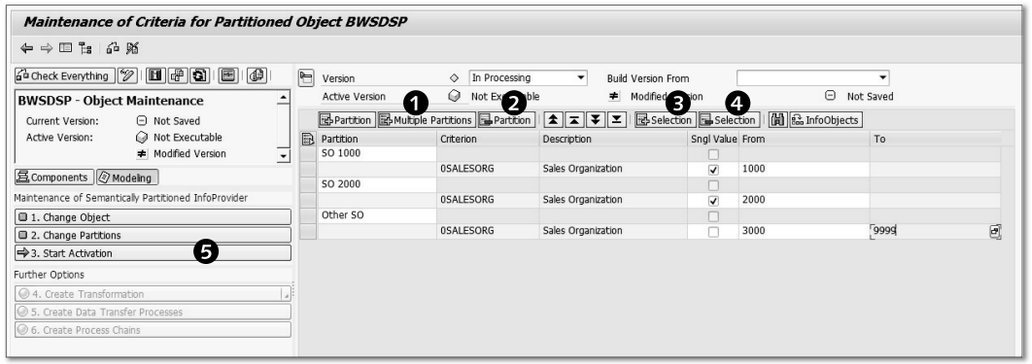


Figure 6.44 Partition Definition for an SPO

Our example scenario requires three partitions that are already configured and shown in Figure 6.44. Click on Start Activation ❺ to activate the SPO and associated objects. This brings up the Display logs screen, as shown in Figure 6.45.

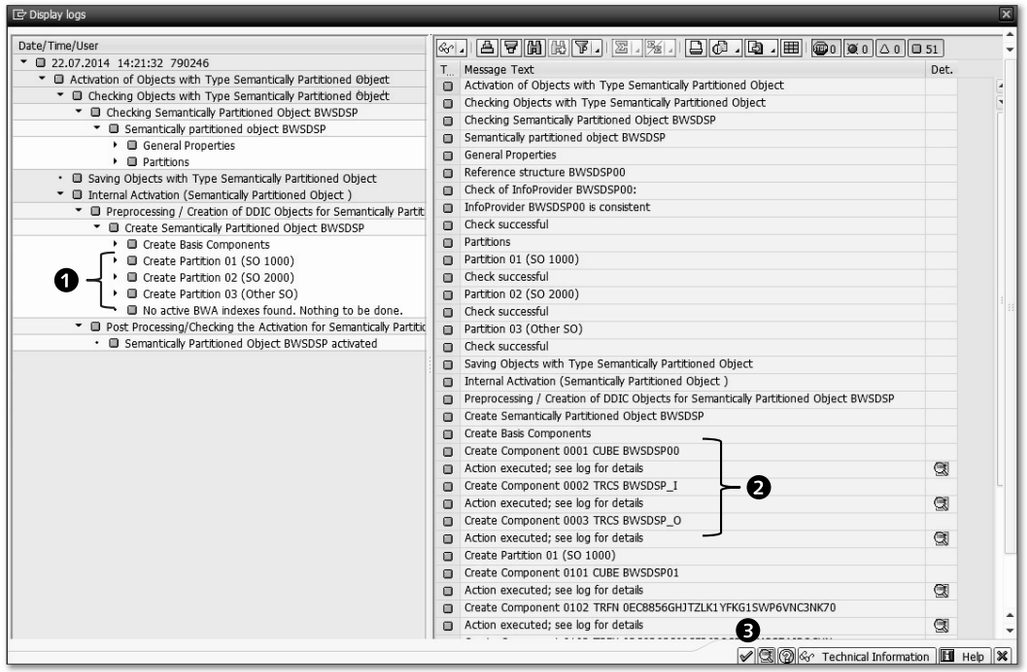


Figure 6.45 Activation Log

As shown in ❶ of Figure 6.45, three partitions are created. The system also assigns the names BWSDSP01, BWSDSP02, and BWSDSP03 to these partitions. All three partitions have the same structure as the reference structure BWSDSP00. Along with these three partitions, the system also generates a few other components ❷.

Component BWSDSP_I is used in creating transformations, which is explained next. Click on Continue ❸, and you're taken back to the Maintenance of Criteria for Partitioned Object BWSDSP screen, as shown in Figure 6.46.

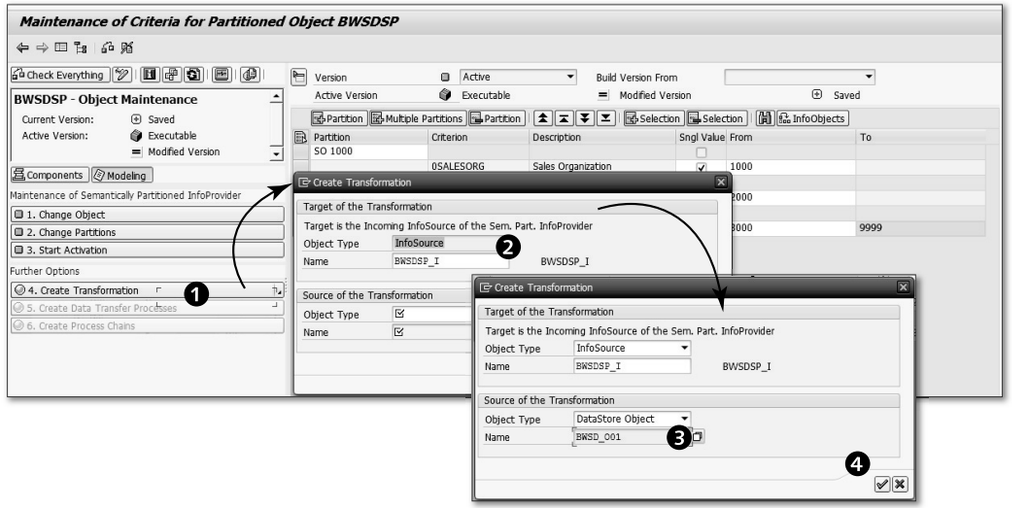


Figure 6.46 Creating a Transformation: Part 1

Creating Transformations and DTPs for SPOs

After maintaining partitions, the next activity is to create a transformation between the source and the defined SPO.

In our scenario, the source is the DSO BWSD_O01. Click on Create Transformation, as shown in ❶ of Figure 6.46 to open the Create Transformation screen. The system defaults the Target (InfoSource BWSDSP_I) ❷. This object is generated while activating the SPO (refer to ❷ of Figure 6.45), and internally connects to the SPO because there are multiple objects (depending on the partition created). Select DataStore Object as the source Object Type, and enter the name "BWSD_O01" ❸. Click on Continue ❹. This brings you to the Create Transformation screen, as shown in Figure 6.47.

As shown in Figure 6.47, the source of the transformation is DSO BWSD_O01 ❶, and the target is InfoSource BWSDSP_I ❷. After the required transformation is maintained, you need to activate it using the Activate icon ❸. You're back to the Edit Semantically Partitioned InfoCube screen, as shown in Figure 6.48.

Note

The transformation between a DSO and an InfoCube is explained in Chapter 7.

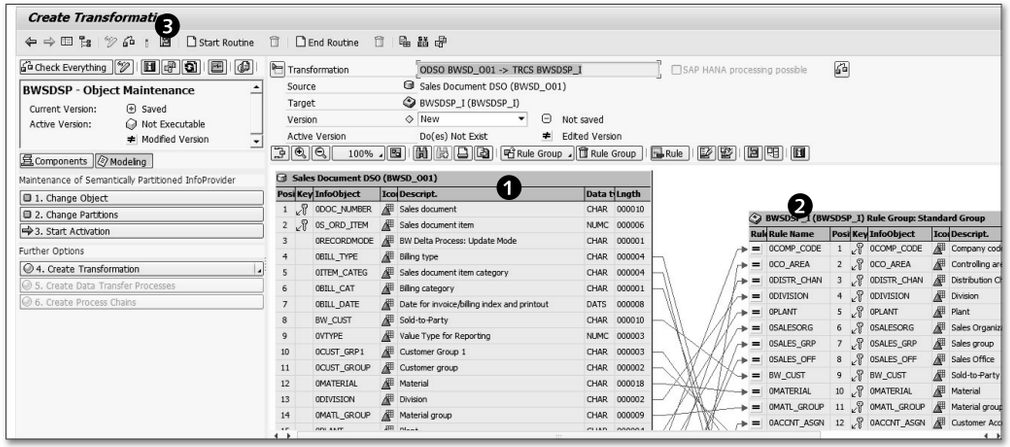


Figure 6.47 Creating a Transformation: Part 2

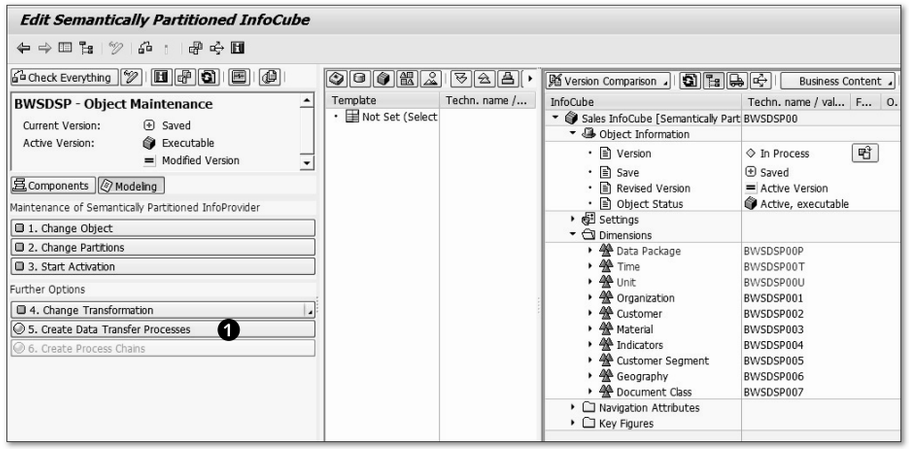


Figure 6.48 Editing a Semantically Partitioned InfoCube

After the transformation is created, you need to create the DTP between the source and the target. As shown in ❶ of Figure 6.48, click on the CREATE DATA TRANSFER PROCESSES option. This brings you to the GENERATE DATA TRANSFER PROCESSES screen, as shown in Figure 6.49.

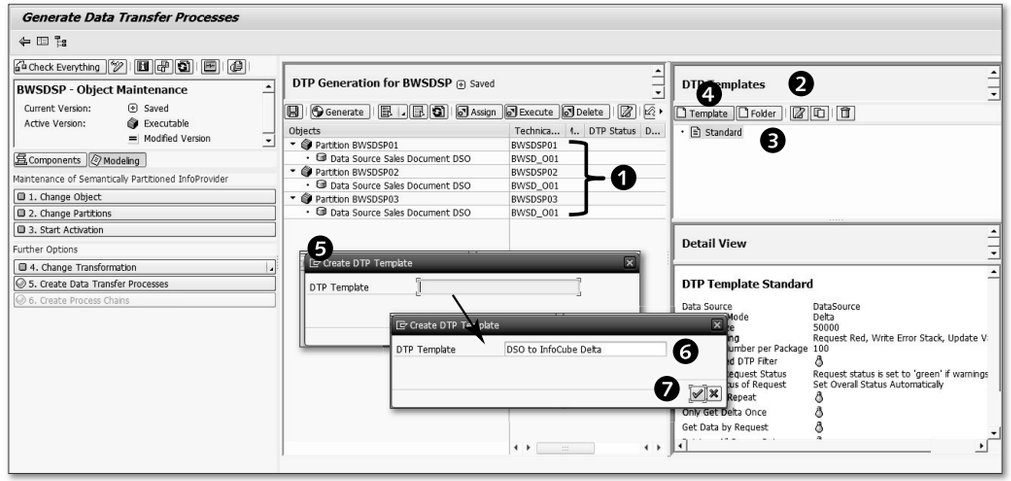


Figure 6.49 Generating Data Transfer Processes

As explained earlier, we've configured three partitions with the SPO BWSDSP. They are shown with their technical names: BWSDSP01, BWSDSP02, and BWSDSP03 (❶ of Figure 6.49). Under each partition, DSO BWSD_O01 is listed as the source because we've created a transformation between DSO BWSD_O01 and InfoSource BWSDSP_I. This InfoSource was generated when the partitions were configured, and it connects all partitioned objects to a common source.

Now we need to create the DTP between the DSO and the SPO. The system offers DTP templates, as shown in ❷ of Figure 6.49; we'll create our own DTP template using the STANDARD template. Select the STANDARD template ❸, and click on the TEMPLATE button ❹. This action brings up the CREATE DTP TEMPLATE pop-up box ❺, where you should enter the name of the template ❻ and click on CONTINUE ❼. The system opens the EDIT PARAMETERS screen, as shown in Figure 6.50.

Depending on the template you choose, the DataSource of the template varies. Because we've selected the standard template, DATASOURCE is selected, as shown in ❶ of Figure 6.50. Ensure that you also select GENERATE AUTOMATICALLY as the DTP FILTER ❷, and change the DATA SOURCE field to DATASTORE OBJECT by using the dropdown list ❸. As soon as you make this change, the number of options will change based on the DataSource selected. For example, the EXTRACTION FROM option wasn't available when the DATA SOURCE field was the DATASOURCE object ❹. For our scenario, there's no need to change any other option. Click on the

SAVE button ❺, and you're returned to the GENERATE DATA TRANSFER PROCESSES screen with the newly created DTP template, as shown in ❶ of Figure 6.51.

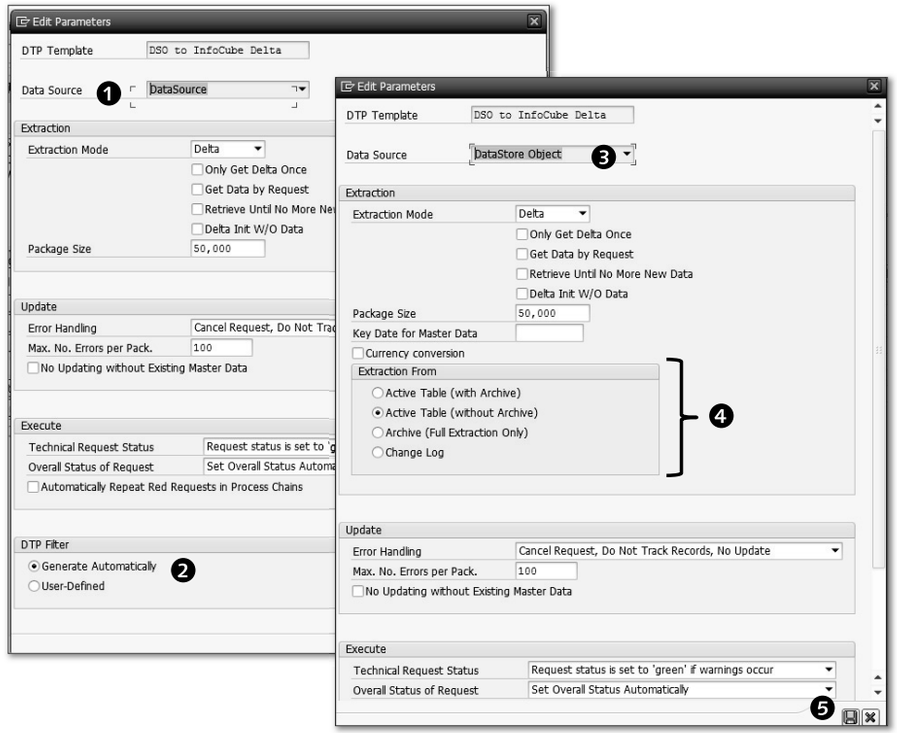


Figure 6.50 Edit Parameters: Copying a Data Transfer Process Template

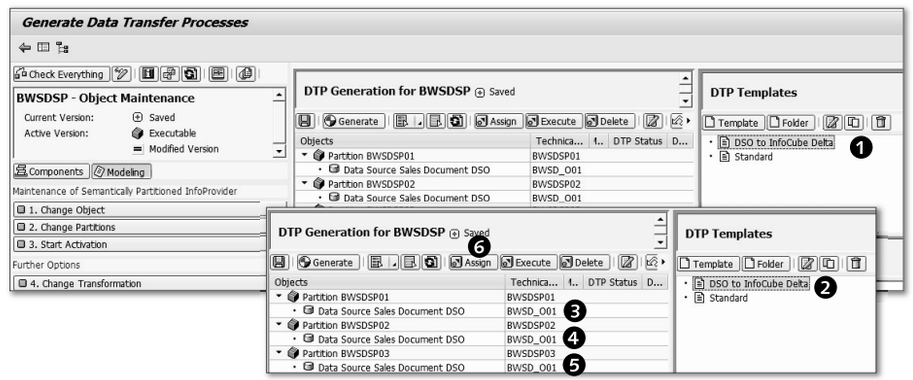


Figure 6.51 Assigning Data Transfer Processes

Now this DTP template needs to be assigned to the DataSource. Select the DTP template ❷, and also select DSO BWSD_D01 displayed under all three partition objects: BWSDSP01, BWSDSP02, and BWSDSP03 (❸, ❹, and ❺). Click on the ASSIGN button ❻. This creates three different DTPs, as shown in ❶, ❷, and ❸ of Figure 6.52.

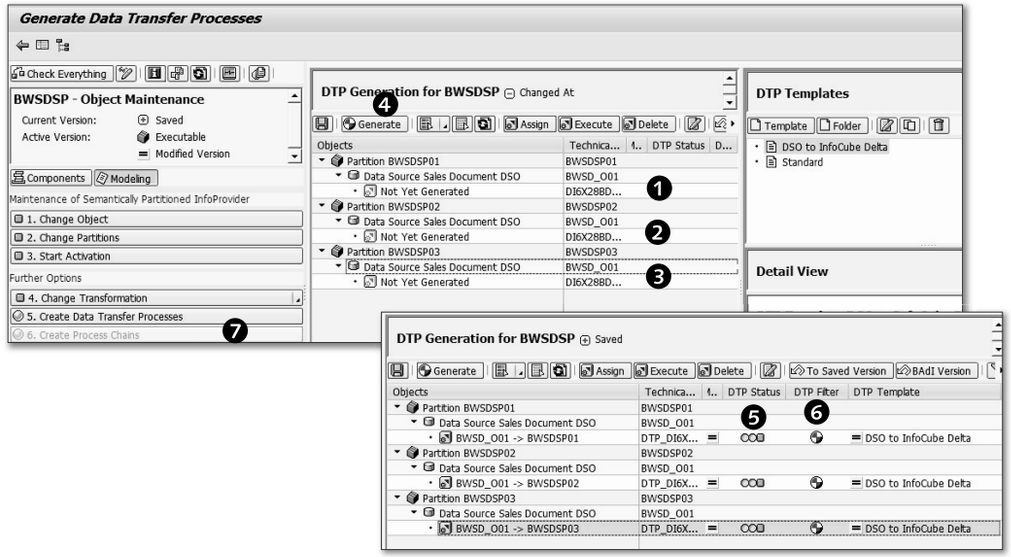


Figure 6.52 Generating Data Tranfer Processes

All of the DTPs generated are initially in inactive status; they need to be activated using the GENERATE button, as shown in ❹ of Figure 6.52. The system starts the generation of DTPs one by one. After all three DTPs are generated, they are shown as ACTIVE ❺. These DTPs have also taken into consideration the respective filter values for each of the partitions, which is displayed in the DTP FILTER column ❻.

Including SPOs in Process Chains

Click on CREATE PROCESS CHAINS, as shown in ❼ of Figure 6.52. This brings up the GENERATE PROCESS CHAINS screen, as shown in Figure 6.53.

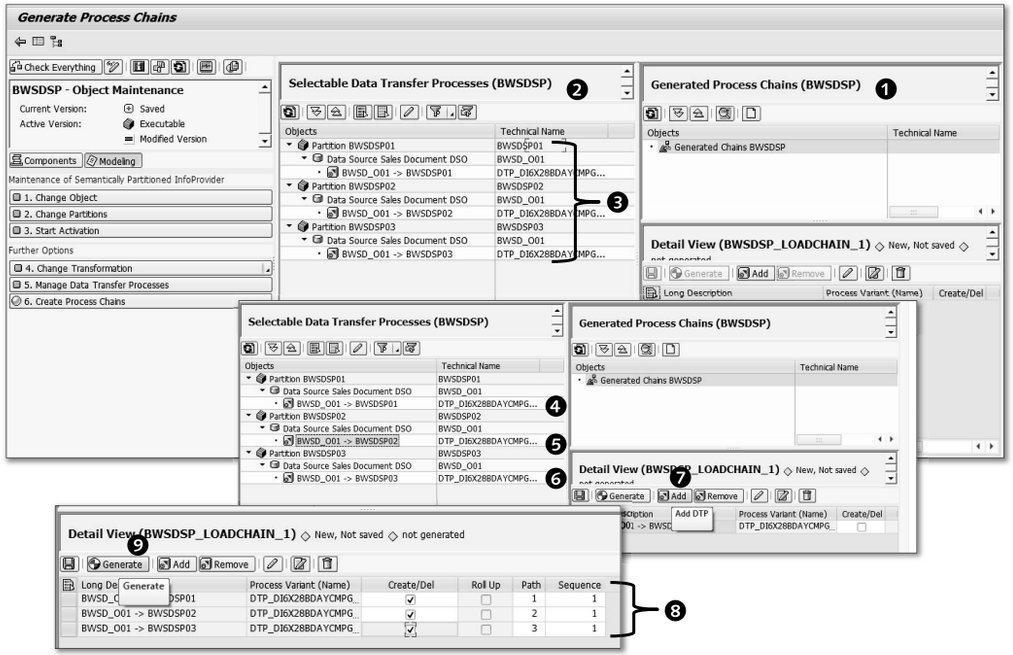


Figure 6.53 Generating Process Chains

On the right of the screen, the GENERATED PROCESS CHAINS (❶ of Figure 6.53) are displayed. In the middle of the screen, selectable DTPs are displayed ❷; these are the DTPs that were generated in the previous step. Select all three DTPs, as shown in ❹, ❺, and ❻. Click on the ADD button ❼, which adds all three selected DTPs to the DETAIL VIEW section ❽. Click on the GENERATE button ❾, which opens the MAINTAIN START PROCESS screen, as shown in Figure 6.54.

Click on the CHANGE SELECTIONS button, as shown in ❶ of Figure 6.54, which displays the START TIME pop-up box. Click on the IMMEDIATE button ❷, and then click on CHECK ❸ and SAVE ❹ to confirm the action. You're now back in the GENERATE PROCESS CHAINS screen, as shown in Figure 6.55.

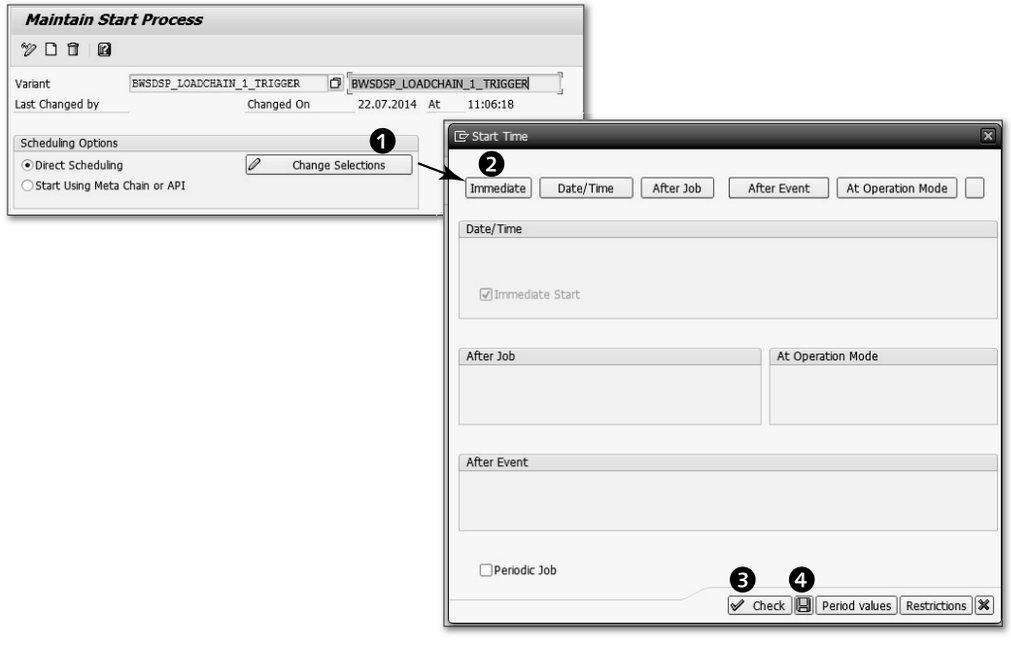


Figure 6.54 Maintaining the Start Process and Execution Time

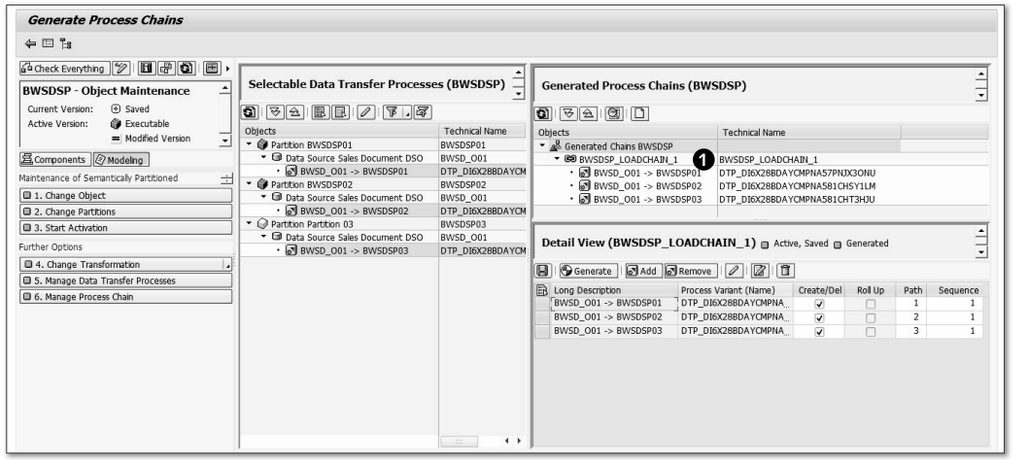


Figure 6.55 Generating Process Chains: Chain Created

You can now see that the system generated a process chain with the technical name BWSDSP_LOADCHAIN_1 and three DTPs under it (as shown in ❶ of Figure 6.55). You can view the process chain using Transaction RSPC, the details of which are shown in Figure 6.56. The execution of this generated process chain included deleting indexes for the InfoCube, executing the DTP (DSO to InfoCube), and creating indexes for the InfoCube.

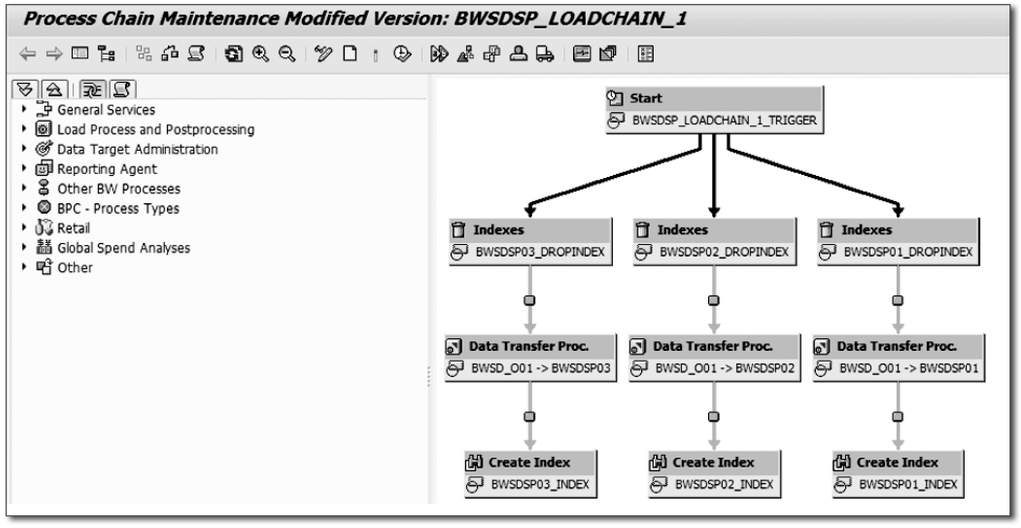


Figure 6.56 Process Chain Display

Note

Process chains are explained in more detail in Chapter 14.

This step completes the configuration of the SPO. You can view the SPO in the Database Warehousing Workbench screen by using Transaction RSA1, as shown in ❶ of Figure 6.57.

As shown in Figure 6.57, three InfoCubes (BWSDSP01, BWSDSP02, and BWSDSP03) are shown under SPI BWSDSP (❷, ❸, and ❹). InfoSource BWSDSP_I and the transformation between the InfoSource and the three partitions are also shown. Just like other InfoProviders, SPOs can be used by BEx tools for querying and reporting.

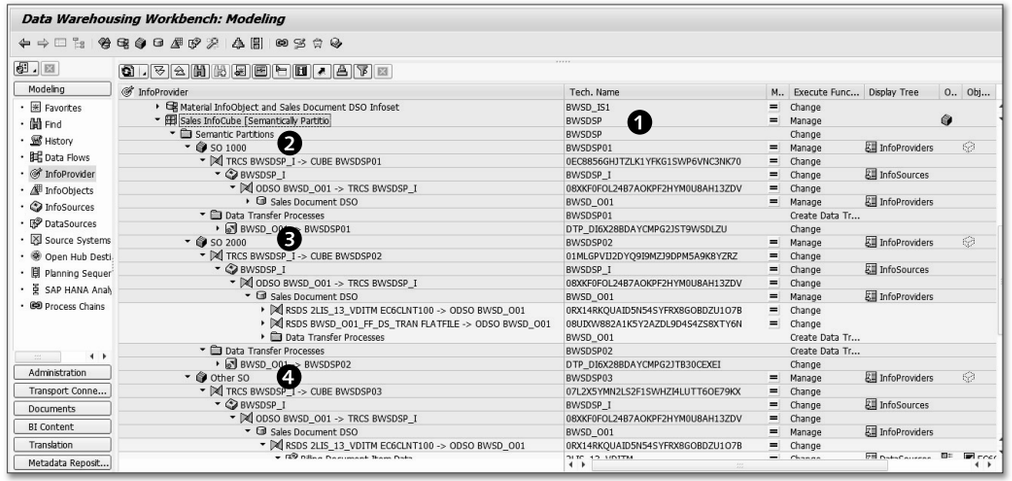


Figure 6.57 Semantically Partitioned InfoCube BWSDSP in the Database Warehousing Workbench Screen

6.5 HybridProvider

Sometimes, you'll need to combine real-time data with large volumes of historic data for reporting without compromising the performance of the reports. For larger data volumes, SAP Business Warehouse Accelerator (BWA) improves query performance. For real-time data, the solution could involve real-time data acquisition (explained in more detail in Section 6.5.1) or VirtualProviders (already explained in Section 6.3). The *HybridProvider* combines both of these components into one and presents it as a single InfoProvider for reporting. This combination allows data to be analyzed in real time along with the historical data and without much impact on performance. The HybridProvider was introduced with SAP BW 7.3.

A HybridProvider has two components: an InfoCube and an InfoProvider. The InfoCube stores the historical data and is loaded in BWA (although loading data into BWA isn't mandatory). The InfoProvider is used for real-time data. Based on the type of InfoProvider used to access real-time data, there are two types of HybridProviders:

- **HybridProvider based on DSOs**
  This HybridProvider is based on a combination of an InfoCube and a DSO. The

InfoCube stores the historical data, while new data is stored in the DSO using the real-time data acquisition (RDA) technique. This means that data is loaded in the DSO in real time through the real-time DTP, and then a DTP is used to load the data from this DSO to the InfoCube, as shown in Figure 6.58. RDA is only required if data is required in real time.

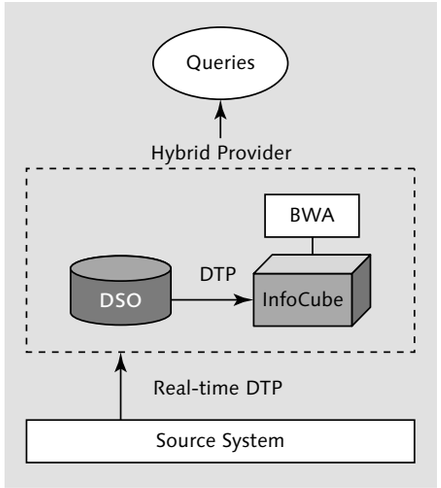


Figure 6.58 HybridProvider Based on DSO

When a query is executed on this HybridProvider, the system automatically determines what data is in the DSO and what part of the data can be accessed from the InfoCube.

- **HybridProvider based on direct access**
  This type of HybridProvider is based on the combination of an InfoCube and a VirtualProvider. Historical data is stored in the InfoCube and is available through BWA, and new data is read from the source system through the VirtualProvider at the time of query execution. Again BWA isn't mandatory. The structure of both the InfoCube and the VirtualProvider is the same. Data is loaded into the InfoCube using the standard DTP, and a DTP for direct access is used between the DataSource and the VirtualProvider to access the real-time data. On execution of the query based on this type of HybridProvider, the system automatically determines whether the data requested resides in the InfoCube or has to be read from the source system using a VirtualProvider. A conceptual overview of this type of HybridProvider is shown in Figure 6.59.

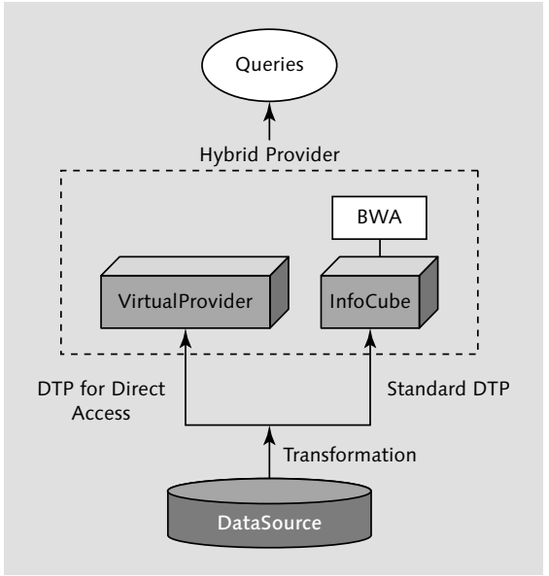


Figure 6.59 HybridProvider Based on Direct Access

Although the HybridProvider technically contains two separate InfoProviders within it, the system collectively treats it as one object. Also, after you define the HybridProvider, related objects, such as transformations and DTPs, are automatically generated. However, these generated DTPs and transformations can't be edited and should be used as generated by the system. The generated transformation between the DSO and the InfoCube (in the case of a HybridProvider based on a DSO) is always 1:1.

In this section, we explain the procedure of creating a HybridProvider. First, however, we start by explaining RDA, which is the concept leveraged in the functioning of a HybridProvider.

6.5.1 Real-Time Data Acquisition (RDA)

As shown in Figure 6.60, there are two ways to get data in real time: via the SAP source system and via external systems.

When data is accessed from the SAP source system, the SAP application (❶ of Figure 6.60) updates the data in the delta queue ❷ in real time. The daemon ❸ in SAP BW starts the InfoPackage for RDA ❹ at a defined interval, which can be

configured while defining the daemon and could be as low as one minute. The InfoPackage for RDA then saves the data in the Persistent Staging Area (PSA) in SAP BW ❺. The daemon also controls the execution of the DTP for RDA ❻, which reads the data from the PSA and loads it into the DSO ❼. This DSO can become part of the HybridProvider.

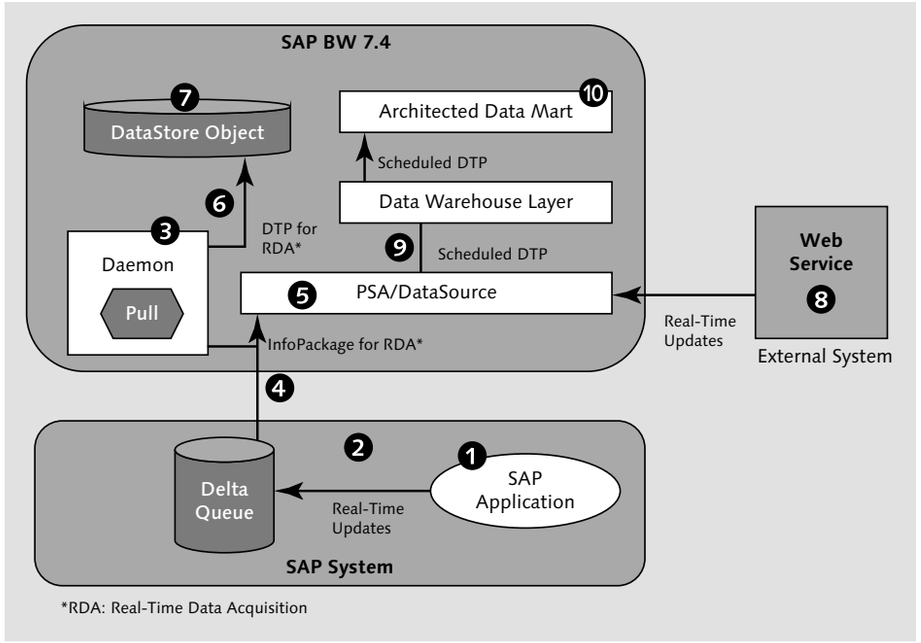


Figure 6.60 RDA Architecture

Daemons

Daemons are background activities that process InfoPackages and the DTPs assigned to them at regular intervals.

Persistent Staging Area (PSA)

A PSA is a transparent table whose structure is the same as that of the associated DataSource. We discuss PSAs in more detail in Chapter 7.

When data is accessed from an external system, the external system Web Services ❽ push real-time data to the PSA of the DataSource ❺ on the SAP BW side. This

PSA data can be loaded using the DTP for RDA, the execution of which is controlled by the daemon. The data available in this PSA can also be loaded into the data warehouse layer using the scheduled DTP ❾. The data from the data warehouse layer can be further updated to an architected data mart using the scheduled DTP ❿.

6.5.2 Creating a HybridProvider

A HybridProvider is created in the Database Warehousing Workbench screen, which is started using Transaction RSA1. Select InfoProvider under Modeling from the navigator section, as shown in ❶ and ❷ of Figure 6.61.

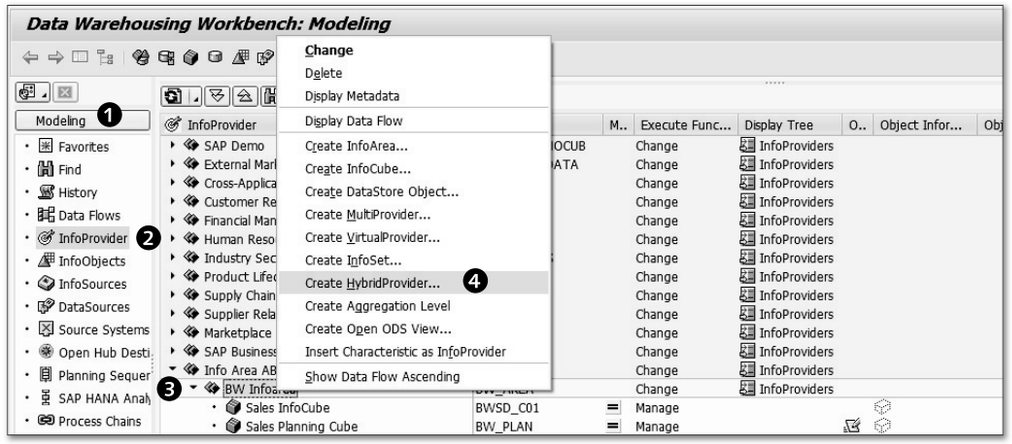


Figure 6.61 Creating a HybridProvider

Select the InfoArea BW_Infoarea under which you want to create your HybridProvider ❸. From the context menu, select Create HybridProvider ❹. The Create HybridProvider screen appears, as shown in ❶ of Figure 6.62.

Using a template (❷ of Figure 6.62) allows you to create a new HybridProvider based on an existing HybridProvider. The InfoCube is part of the definition of the HybridProvider, and the name of the InfoCube involved in the HybridProvider is shown under Involved InfoCube ❸. The type of the HybridProvider is selected from HybridProvider Type ❹.

Enter the name of the HybridProvider as "BW_HP1" with the description "Sales Document Hybrid Provider" ❺. We're creating our HybridProvider based on the

existing DSO BWSD_O04. You need to have created BWSD_O04 based on the definition per Table 6.18; the fields included in DSO BWSD_O04 are also shown in this table. Select BASED ON DATASTORE ❼.

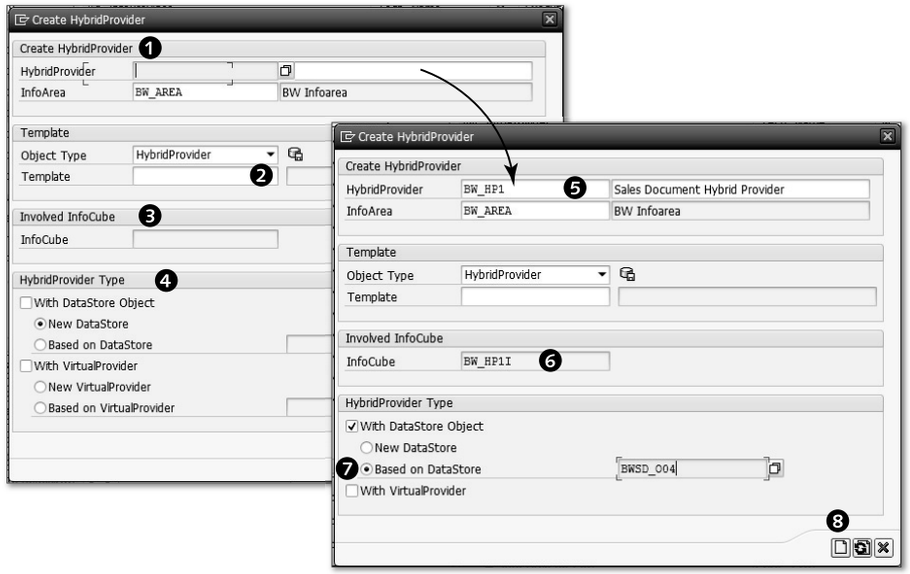


Figure 6.62 Editing a HybridProvider: Part 1

Field Type	InfoObject	Description
Data Fields	0BILL_DATE	Date for invoice/billing index and printout
	BW_CUST	Sold-to party
	0MATERIAL	Material
	0DIVISION	Division
	0SALESORG	Sales organization
	0DISTR_CHAN	Distribution channel
	0SALES_OFF	Sales office
	BW_QTY	Billing quantity
	0UNIT	Unit of measure
	0DOC_CURRCY	Document currency
	0NET_VALUE	Net value of the order item in document currency

Table 6.18 Design of DSO BWSD_O04

Field Type	InfoObject	Description
Key Fields	0DOC_NUMBER	Sales document
	0S_ORD_ITEM	Sales document item

Table 6.18 Design of DSO BWSD_O04 (Cont.)

As soon as you select this, the system selects the name of the involved InfoCube (BW_HP1I), as shown in ❻ of Figure 6.62. To continue with the definition, click on the CREATE icon ❽. This brings you to the screen shown in Figure 6.63.

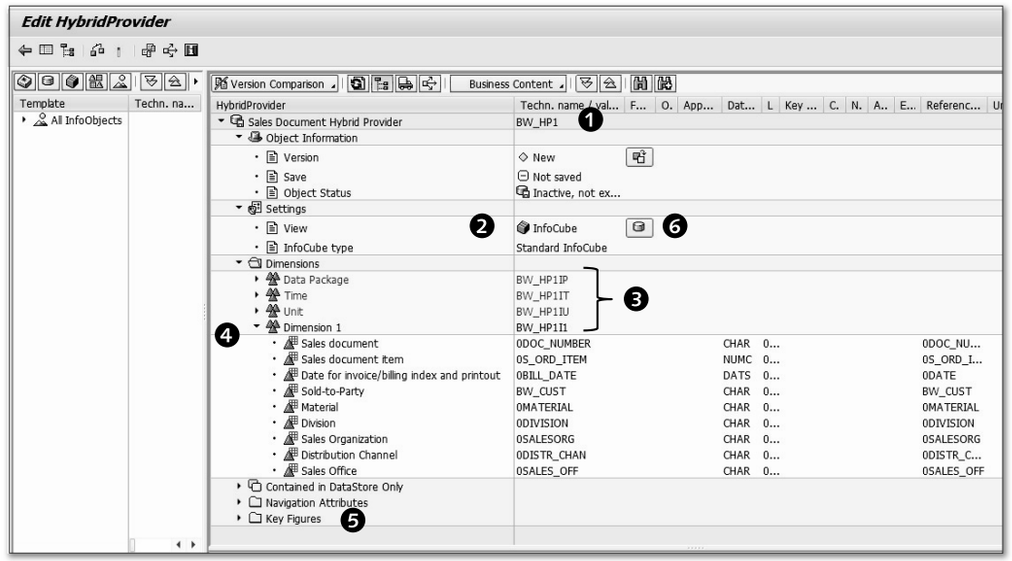


Figure 6.63 Editing a HybridProvider: Part 2

As shown in ❶ of Figure 6.63, the name of the HybridProvider is displayed. The system shows the INFOCUBE view ❷, which means that the system displays dimensions and key figures, as shown in ❸, ❹, and ❺. The system automatically creates the dimensions Data Package, Time, Unit, and Dimension 1, and all of the involved key figures are shown under the KEY FIGURE heading. You can jump to the DATASTORE OBJECT view by clicking on the DSO icon ❻. This switches the display to the DSO view of the HybridProvider, as shown in Figure 6.64.

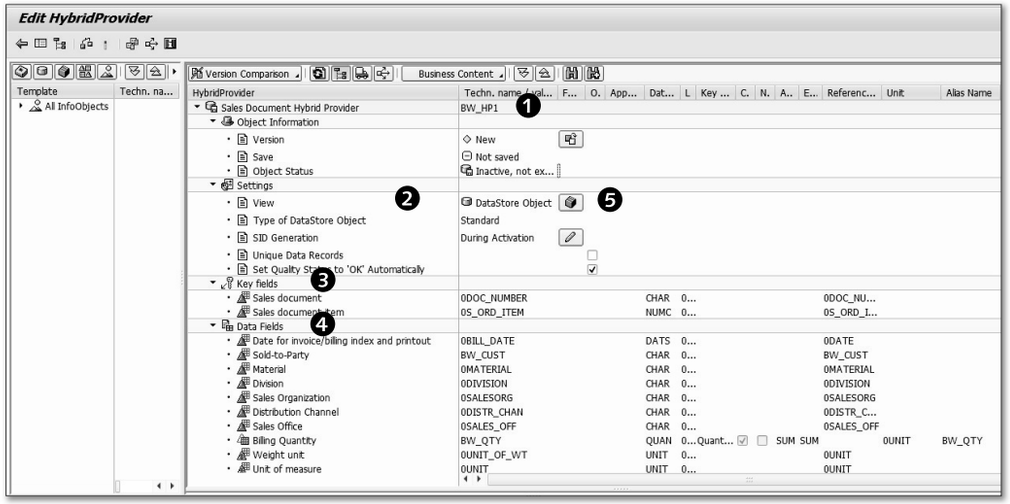


Figure 6.64 DataStore Object View of the Edit HybridProvider Screen

As shown in Figure 6.64, the definition of the DSO ❶ included in the HybridProvider is shown with SETTINGS ❷, KEY FIELDS ❸, and DATA FIELDS ❹. By clicking on the INFOCUBE icon ❺, you can switch back to the InfoCube view.

Click on the ACTIVATE icon to activate the HybridProvider. You can view the created HybridProvider using Transaction RSA1, as shown in Figure 6.65.

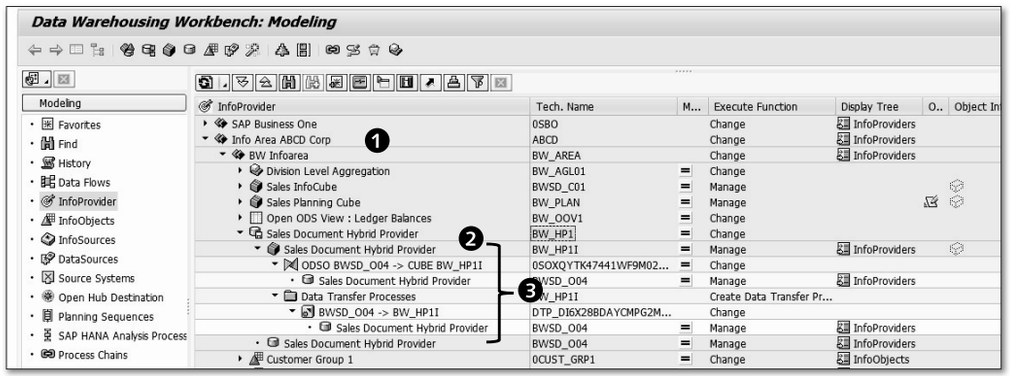


Figure 6.65 Created HybridProvider

The newly created HybridProvider BW_HP1 (❷ of Figure 6.65) is shown under the InfoArea BW_INFOAREA ❶. At the time of activating HybridProvider BW_HP1, the system also generated the associated transformation and DTP between DSO BWSD_O04 and InfoCube BW_HP1I ❸. The HybridProvider with direct access can be created following a similar procedure.

The HybridProvider provides the additional design flexibility of creating a data model with high-performance reporting on historical data combined with real-time data or data with reduced latency.

6.6 Transient Providers

Transient providers were introduced with the release of SAP BW 7.3. The objective behind transient providers is to provide the ability to perform ad hoc operational reporting directly on the SAP Online Transaction Processing (OLTP) system without the additional need to extract and stage the data in SAP BW; by using a transient provider, it's possible to report on real-time data (i.e., with zero latency). Using the transient provider, the standard BEx reporting tools can be used to create queries directly on the SAP OLTP tables and data. So, there's no need to have a separate SAP BW instance and to extract the data from the SAP OLTP system.

Note

In SAP BW powered by SAP HANA, the SAP HANA models published for consumption in SAP BW application are available as transient providers.

As a prerequisite to leverage the transient provider, the BI Client technical component must be installed on the same SAP OLTP system that has to be used for reporting; a transient provider by itself doesn't have any metadata, such as fields or InfoObjects. This InfoProvider is completely logical in nature and isn't visible in the DATABASE WAREHOUSING WORKBENCH screen. A conceptual overview of the transient provider is given in Figure 6.66.

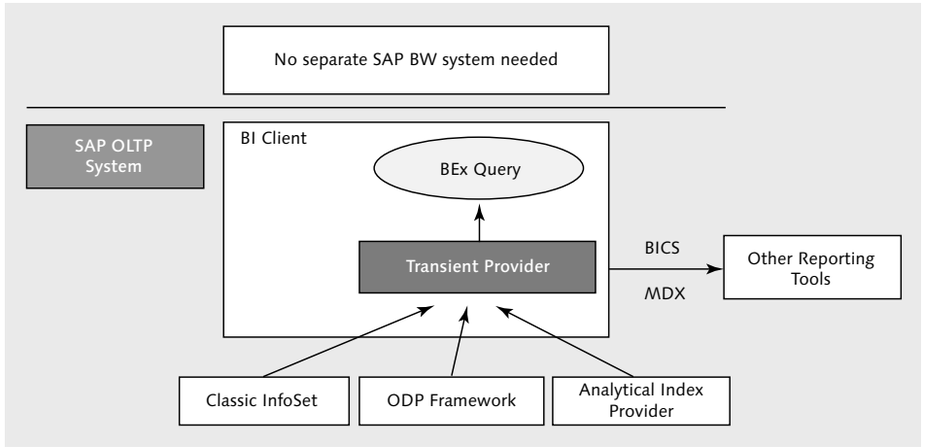


Figure 6.66 Transient Provider Concept

The source of data for a transient provider can be any of the following:

- A classic InfoSet in the OLTP system; these are client independent.
- DataSources from the SAP OLTP system; these are made available through the Operational Data Provisioning (ODP) framework.
- An analytical index created using the Analysis Process Designer (APD) (explained in Chapter 15).

Operational Data Provisioning (ODP)

ODP is a technical framework that's used to expose the standard SAP BW DataSources in the SAP OLTP system for third-party extraction tools (such as SAP Data Services) or to expose the SAP BW DataSources for operational reporting in the SAP OLTP system (as in the case of a transient provider).

As illustrated in Figure 6.66, the source from the preceding list is exposed as a transient provider to BEx Query Designer. From a BEx Query Designer perspective, this is just like any other regular SAP BW InfoProvider. All of the standard BEx Query Designer features are available when reporting on transient providers, too.

Apart from BEx reporting, transient providers can also be used to expose operational reporting to third-party reporting tools. For example, reporting tools from SAP BusinessObjects can connect with transient providers and thus get access to

operational data in SAP OLTP systems. These third-party applications can leverage the standard BI Consumer Services (BICS) connectivity or the MDX connection.

Transient providers are thus an important method of achieving real-time reporting directly on an SAP OLTP system in a way that also allows you to utilize the complete set of features offered by BEx tools.

Note

You don't have to create any persistent metadata or data models when creating a transient provider. The characteristics and key figures used in the transient provider are automatically generated from the source field definitions.

6.7 Analytical Index

An analytical index is another type of InfoProvider that was introduced in SAP BW 7.3. This is an in-memory InfoProvider, which means that its data is stored directly in memory. Thus, an analytical index can be created directly in BWA or in an SAP BW system that uses SAP HANA as a database. An analytical index can be a source for a transient provider or for a CompositeProvider, and it can be used to address ad hoc reporting needs.

The creation of an analytical index takes place using an analysis process. We'll discuss more about analytical indexes and the procedure to create one when we explain analysis processes and the APD in Chapter 15, Section 15.2.

6.8 CompositeProvider

CompositeProviders are also new since SAP BW 7.3. The SAP BW CompositeProvider is used to bring the data together from in-memory data targets/InfoProviders. This InfoProvider can leverage both union and join operations to combine the data set. The base InfoProviders must reside in memory to be included in a CompositeProvider. It means that if you're using SAP BW on a non-SAP HANA database, then BWA is a prerequisite for a CompositeProvider. In such cases, only those SAP BW InfoProviders that reside in BWA can be part of a CompositeProvider. The BWA prerequisite for CompositeProviders is irrelevant if the SAP BW system uses the SAP HANA database.

There are three types of CompositeProviders, and each serves different requirements for different user groups and application areas: a central CompositeProvider, ad hoc CompositeProvider, and local CompositeProvider. We'll explain the procedure to create a central CompositeProvider in the following section, and the local CompositeProvider is explained in Chapter 10.

6.8.1 Creating a CompositeProvider

A CompositeProvider is created in the DATA WAREHOUSING WORKBENCH screen, which is opened using Transaction RSA1. Select INFOPROVIDER under MODELING from the navigator section, as shown in ❶ of Figure 6.67. Select the InfoArea under which you want to create your CompositeProvider ❷. From the context menu, select CREATE COMPOSITEPROVIDER ❸. The CREATE COMPOSITEPROVIDER pop-up appears, as shown in Figure 6.68.

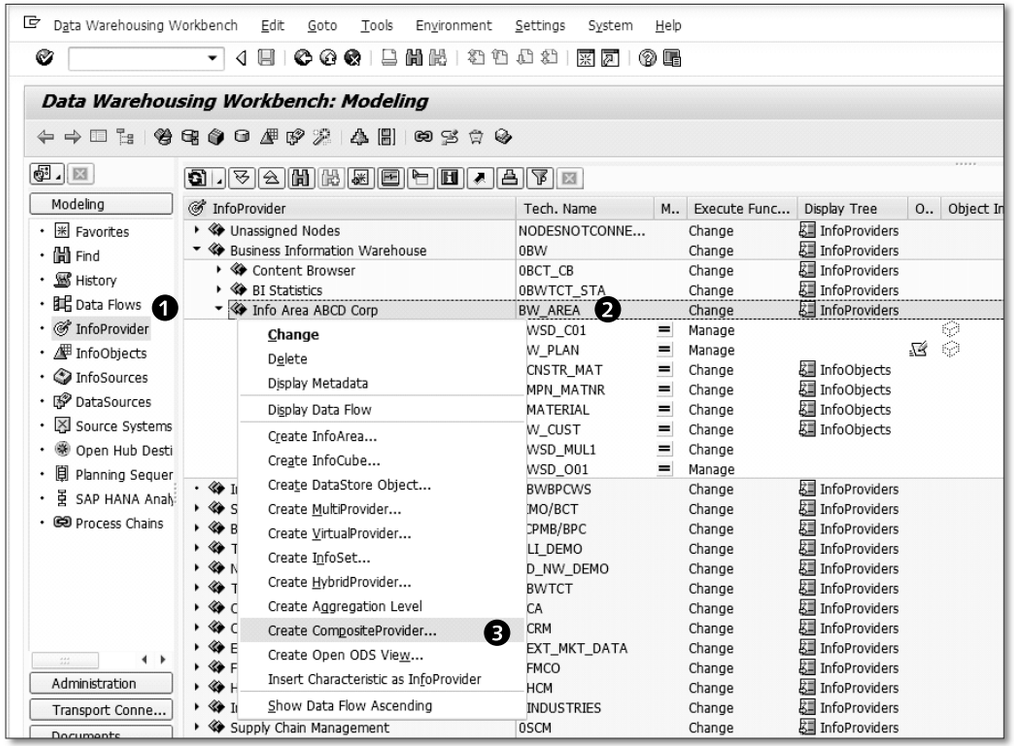


Figure 6.67 Creating a CompositeProvider

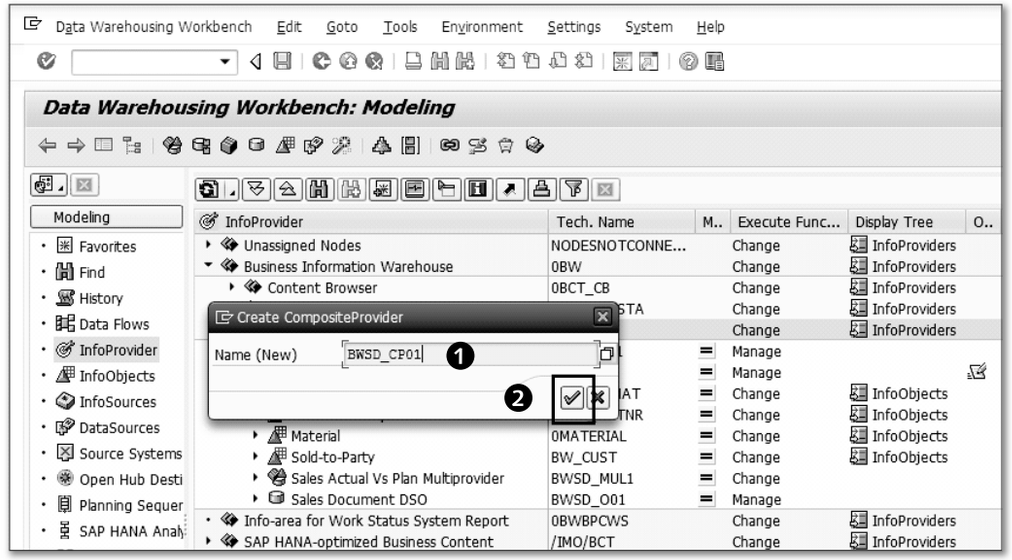


Figure 6.68 Entering the Technical Name for the CompositeProvider

Enter the technical name and description of the CompositeProvider (❶ of Figure 6.68). Click on the CREATE icon ❷. This takes you to the screen where you define the reference data model for the CompositeProvider (see Figure 6.69 in the next section).

To illustrate the creation of a CompositeProvider, let's assume a scenario where there is a business need to build a report that will show sales bonuses by sales office. This bonus calculation has to happen based on the sales commission percentage defined for each of the sales offices and the actual sales numbers. In SAP BW, the actual sales values are available in one of the InfoCubes (BWSD_C01), while sales commissions are stored in a separate master data InfoObject BW_SLCOM. To address this requirement, we would like to combine sales data and commission data in a CompositeProvider.

6.8.2 Defining CompositeProviders

The naming convention by SAP, that is, all CompositeProvider begin with "@3", is defaulted in the technical name for the InfoProvider in the right panel as shown in ❶ of Figure 6.69.

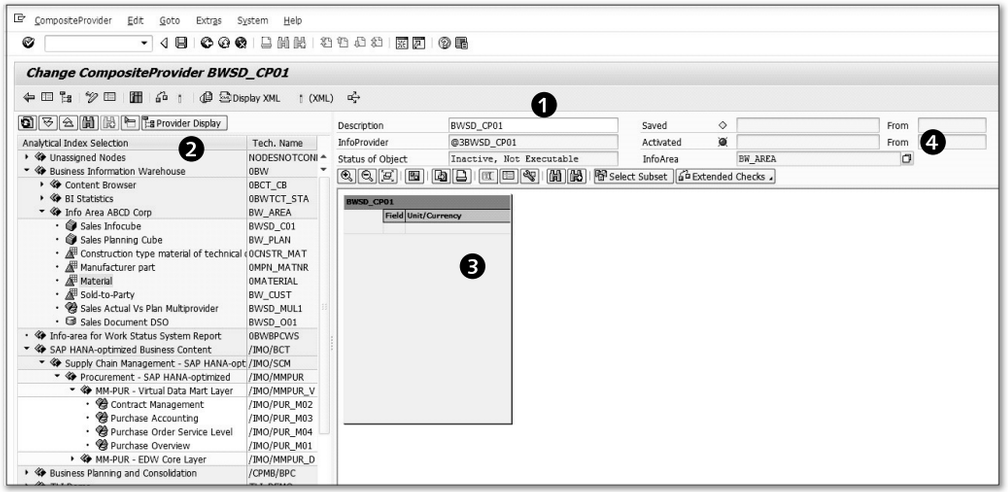


Figure 6.69 Reference Data Model for the CompositeProvider

The left panel provides all possible objects that could be used in creation of the CompositeProvider (❷ of Figure 6.69). The CompositeProvider is shown in the right panel ❸. We recommend identifying the CompositeProvider into a specific InfoArea by using the InfoArea dropdown menu in the right panel ❹.

Inserting Objects

From the left side panel, select InfoCube BWSD_C01 (❷ in Figure 6.70), and drag and drop it into the right panel. In the pop-up that appears ❸, select a Binding Type ❺. In this example, we chose Union for the purpose referred to earlier. Click on the Continue icon ❻, and the object is now available in the right panel (❶ of Figure 6.71).

Select the objects in BWSD_C01 that should be included in the CompositeProvider definition (❷ of Figure 6.71). From the context menu, select Add Selected Objects to add the selected objects to the CompositeProvider definition ❸.

All the characteristics of the inserted object are mapped 1:1 in the CompositeProvider (refer to ❸ of Figure 6.71).

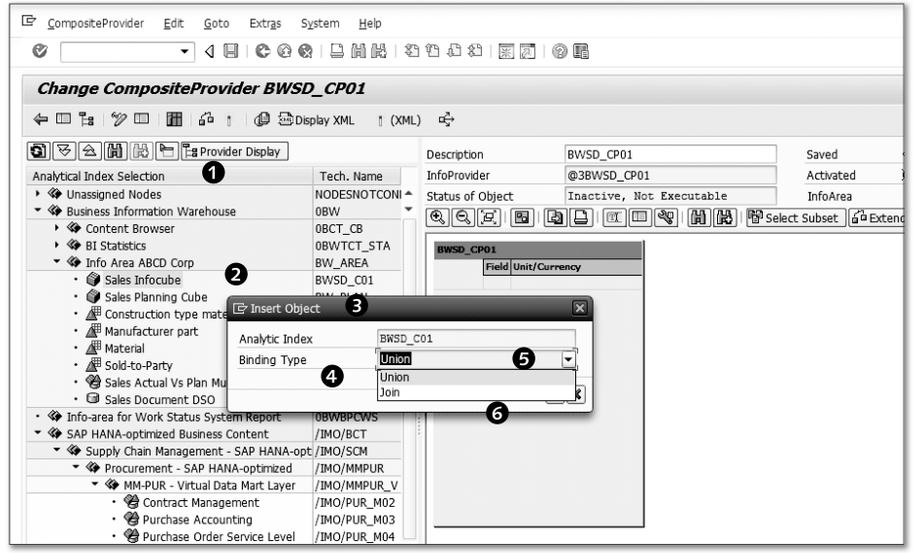


Figure 6.70 Inserting an Object

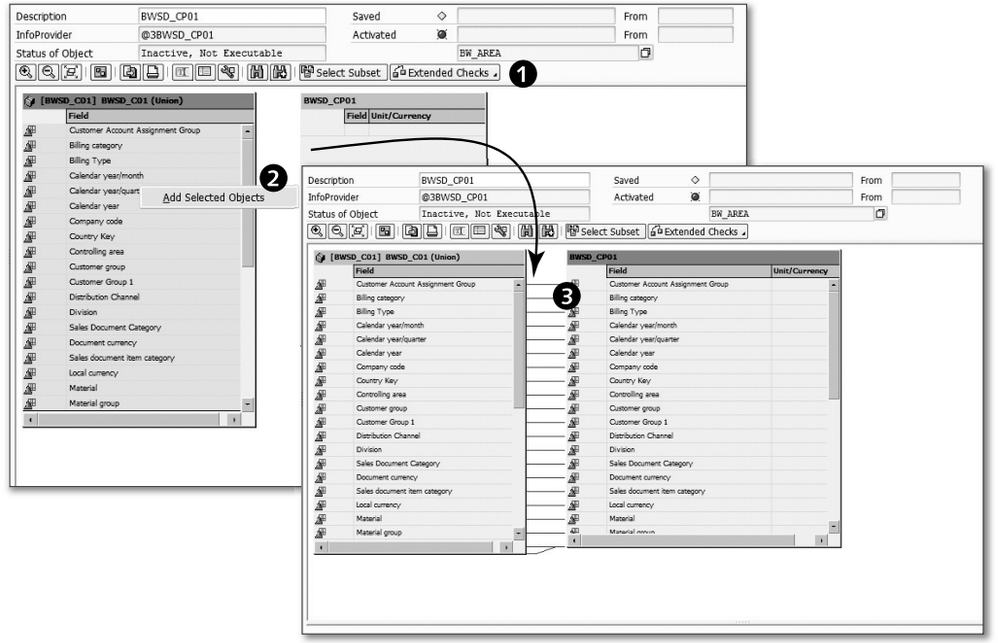


Figure 6.71 Defining a CompositeProvider: Part 1

Now identify other objects for the sales commission data from the panel on the left side by following the same process as for BWSD_C01. For this example, we selected BW_SLCOM (❶ of Figure 6.72) and chose JOIN in the BINDING TYPE field ❷.

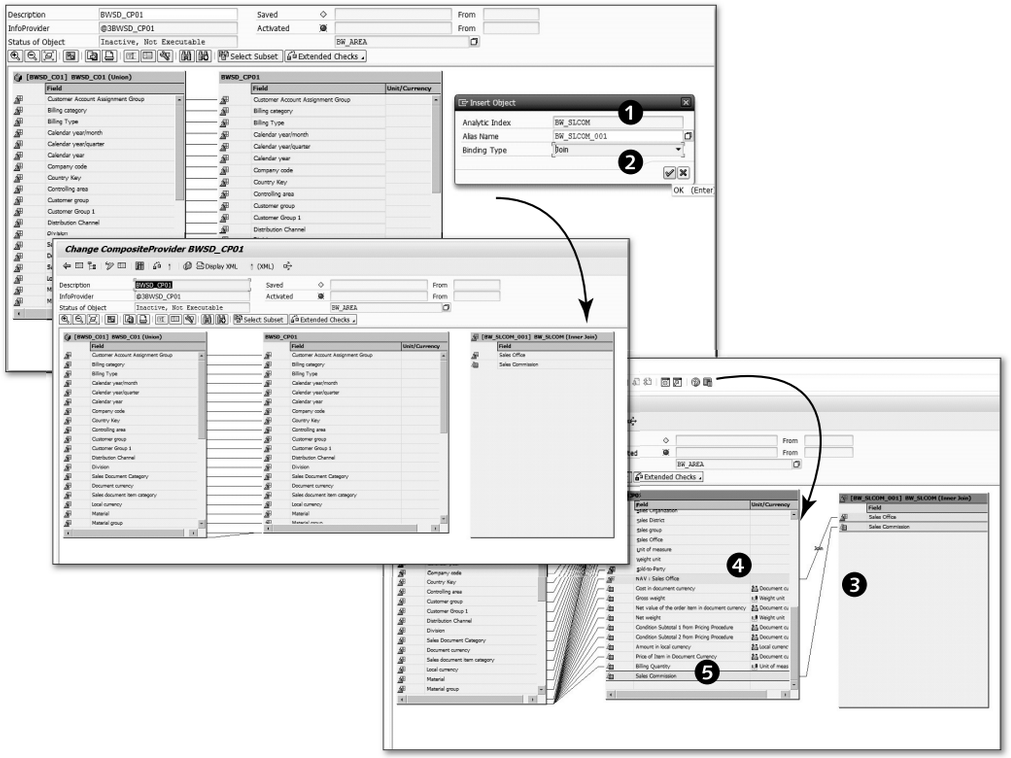


Figure 6.72 Defining a CompositeProvider: Part 2

The new object now is available in the right panel. The fields are mapped to the CompositeProvider via drag and drop (❸ of Figure 6.72). The join for SALES OFFICE ❹ is defined specifically while including the key figure for SALES COMMISSION ❺ in the CompositeProvider definition.

The EXTENDED CHECKS button (❶ of Figure 6.73) provides you options to analyze the defined joins and debug the definition of the CompositeProvider. Also, the context menu from the header of the inserted object gives you options to toggle between union and other join settings ❷.

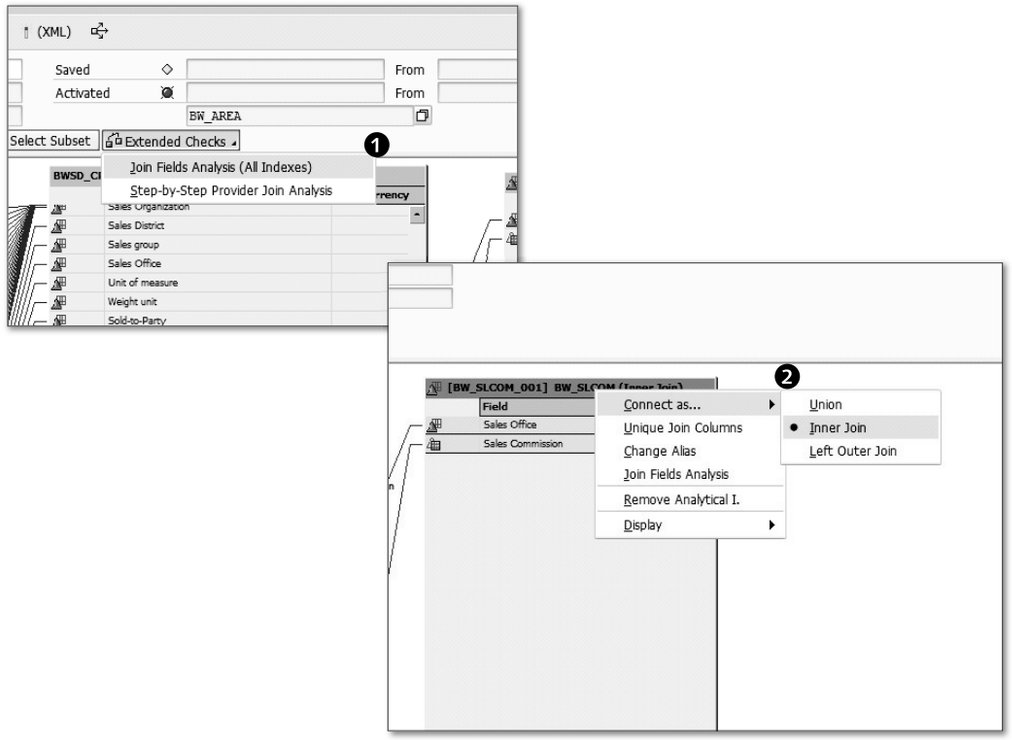


Figure 6.73 Defining a CompositeProvider: Part 3

Check the CompositeProvider using the CHECK icon (❶ of Figure 6.74), and take note of the system messages in the information panel that gets generated below ❷. When there are no errors reported, save the CompositeProvider by clicking on the SAVE icon in the menu bar ❸, and note the system messages in the information panel that gets generated below ❹. Finally, activate the CompositeProvider by using the ACTIVATE icon ❺, and note the system messages in the information panel that generated below ❻.

We recommend using the display data functionality to establish information output from the CompositeProvider. Click on the DISPLAY DATA icon (❶ of Figure 6.75) to see the options for filtering the data to be included or excluded in the output ❷. Choose the EXECUTE icon ❷, and check the output ❸.

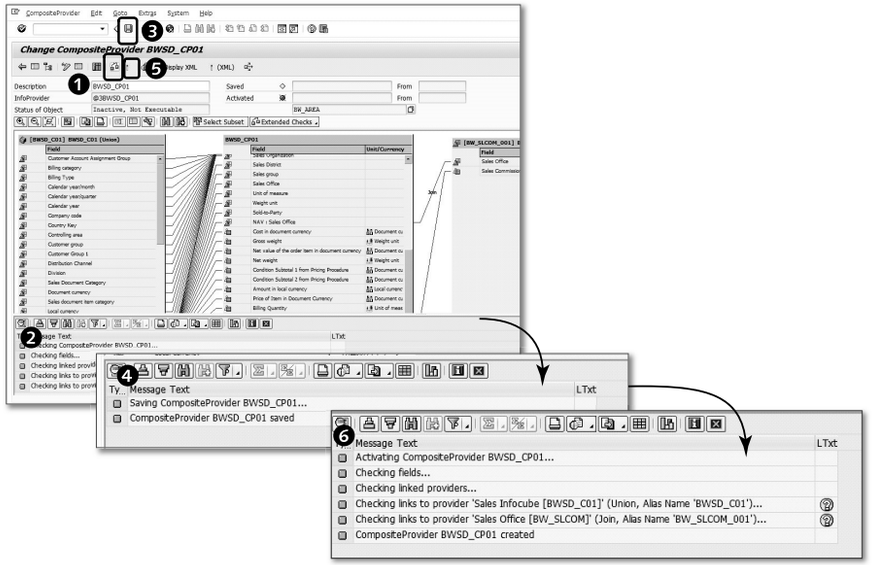


Figure 6.74 Saving and Activating a Composite Provider

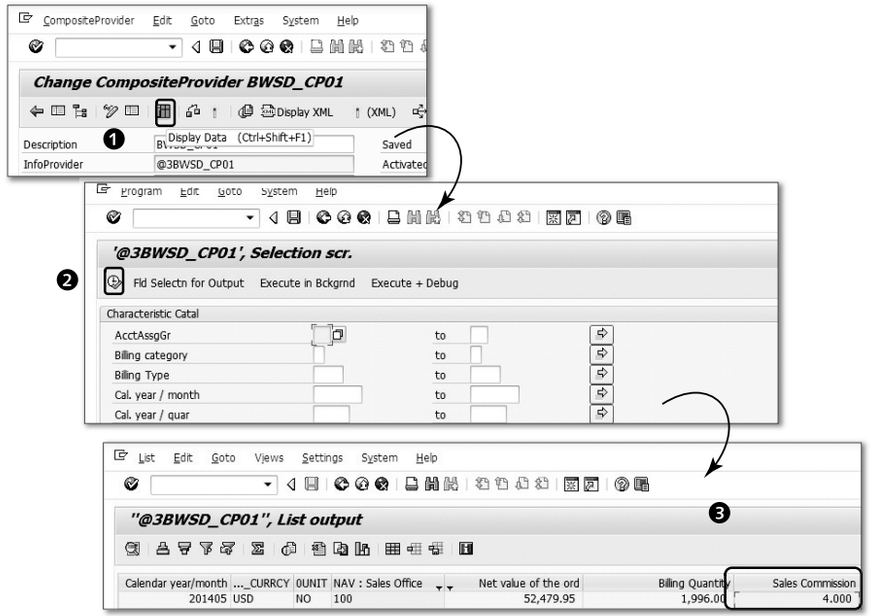


Figure 6.75 Displaying Data from a CompositeProvider

The newly created CompositeProvider is now available in the InfoArea BW_AREA within the Data Warehousing Workbench screen (❶ Figure 6.76).

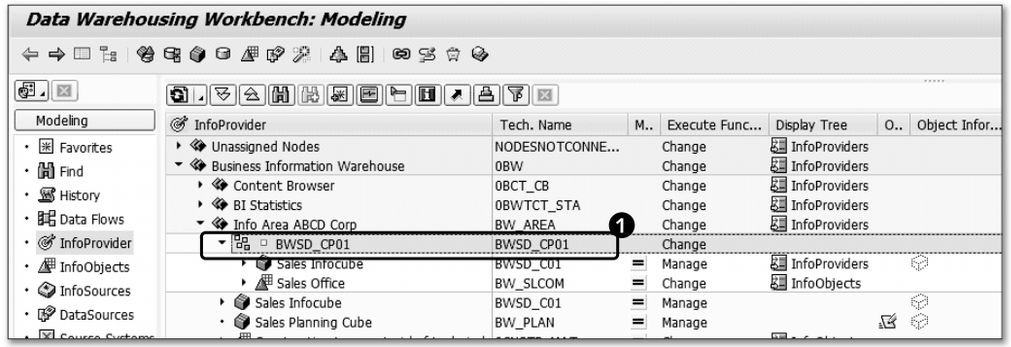


Figure 6.76 Newly Created CompositeProvider in the InfoArea

6.9 Open ODS View

The open ODS view (OOV), which was newly introduced in SAP BW 7.4, enables designers to build InfoProviders in a step-by-step manner. Until now, all the InfoProviders offered by SAP BW had to be designed using InfoObjects, which are the smallest building blocks of SAP BW and which are used in designing a number of other SAP BW objects. This makes it difficult to change the definition of an InfoObject after it's used in building other InfoProviders.

These InfoProviders give designers the flexibility to model InfoProviders based on source fields, and InfoObjects aren't mandatory. There is also a facility to attach an InfoObject to a field at the time of design to get benefits associated with the InfoObject (e.g., text of master data, navigational attributes, etc.). The main advantages of OOV are the following:

- Use of OOV as a VirtualProvider, with no data staging required
- Use of existing SAP BW objects to get additional information
- Flexibility of enhancing model
- Facility to load the data as the persistence layer

Prerequisite for the Open ODS View

Your SAP BW system must be on an SAP HANA database to use the open ODS view.

Let's now consider how to create OOV in an SAP BW system.

6.9.1 Creating an Open ODS View

As explained earlier, an OOV is a type of InfoProvider. An OOV is created in the Database Warehousing Workbench screen, which is opened using Transaction RSA1. Select InfoProvider ❶ under Modeling ❷ from the navigator section (see Figure 6.77).

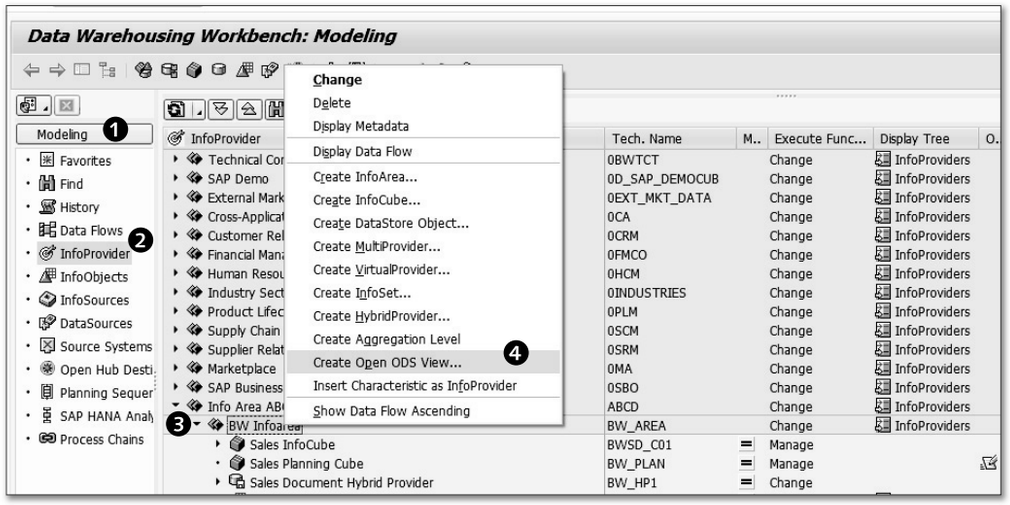


Figure 6.77 Creating an Open ODS View

An OOV is created under InfoArea. Select InfoArea BW_InfoArea ❸. From the context menu, select Create Open ODS View ❹ to open the Create Open ODS View screen as shown in ❶ of Figure 6.78.

You need to provide certain input while creating an OOV. Each OOV requires a unique name. Provide the name "BW_OOV1" and enter "Open ODS View – Ledger Balances" in the Long Description field (❷ of Figure 6.79).

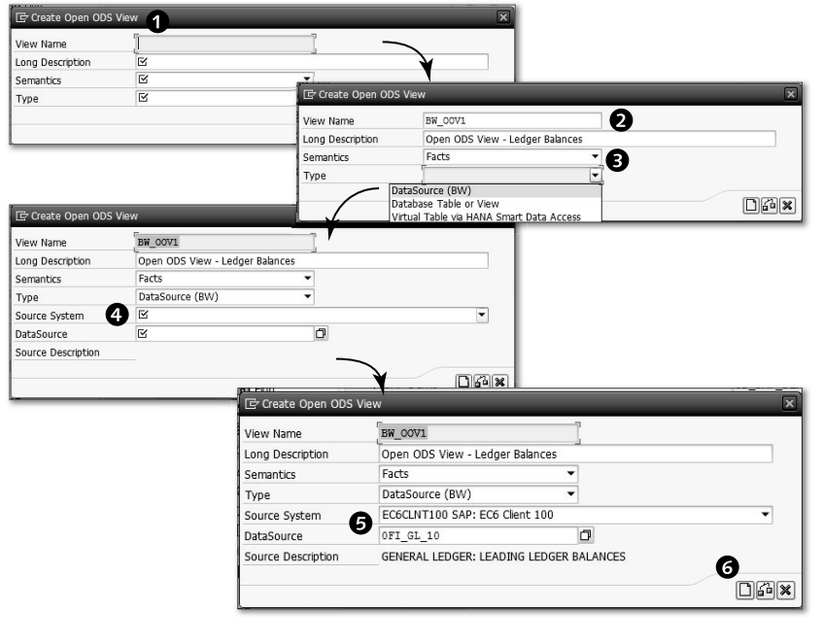


Figure 6.78 Open ODS View Input

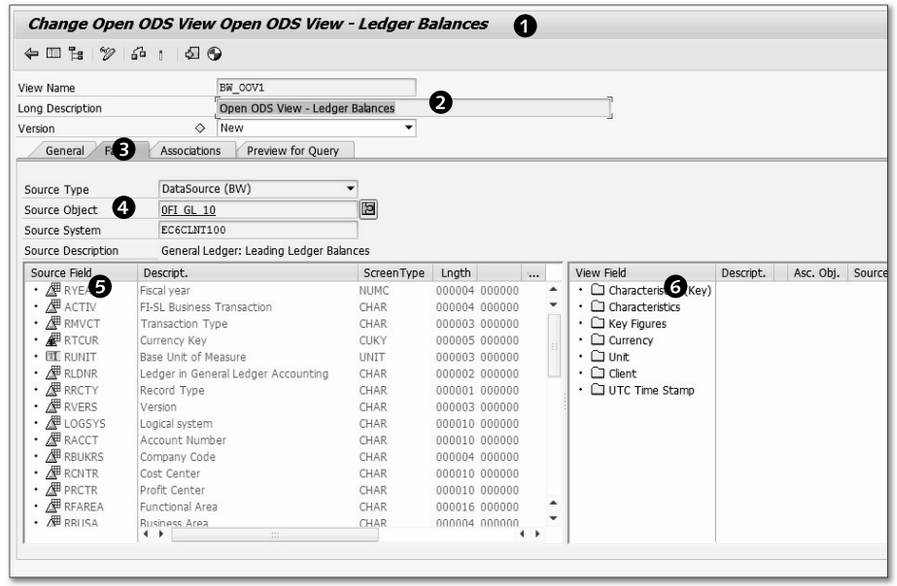


Figure 6.79 Changing the Open ODS View

An OOV is based on semantics, which are facts, master data, and texts, and we're creating an OOV based on facts, so click on the Facts tab ❸. The details of the source are shown ❹. The OOV Source Type field indicates what object stores the data: DataSource (BW), Database Table or View, or Virtual Table via HANA Smart Data Access. For our example OOV, select DataSource (BW) as the source type and 0FI_GL_10 as the source object. Once all of the details are added, click Create ❻, which opens the Change Open ODS View screen (Figure 6.78).

The Version field is initially shown as New. Below these fields, four tabs are available:

- General
- Facts
- Associations
- Preview of Query

The fields of this particular DataSource of source system EC6CLNT100 are shown under the Source Field column ❺. Along with the field name, the description, screen type, and length of the field are also displayed. Under the View Fields column ❻, the following seven folders are available:

- Characteristics (Key)
- Characteristics
- Key Figures
- Currency
- Unit
- Client
- UTC Time Stamp

You need to select source fields from the left side and move them to the right side under one out of seven sections. Let's see how it's done.

If you plan to move all the source fields from source to view, you can use the proposal feature. By clicking on the Create Proposal icon (❶ of Figure 6.80), you can move all the source fields to the View Field section. All the fields are moved from source to view based on their data types. You can remove the fields that aren't required from the View Field section. If you require only a few fields to be taken from the View Field section, you can take them one by one, as we'll describe next.

For example, let's move Account Number (RACCT field) to the Characteristics (Key) section. To do that, first select the RACCT field from the Source Field section ❷, right-click to open the context menu, and select Add to Characteristics (Key) ❸. This action brings RACCT files under the View Field section ❹. In the same way, get the RBUKRS field (Company Code) under the View Field section.

You can use an OOV in two different ways:

- As a virtual query provider
- To store output of a virtual query into a DSO as persistence data

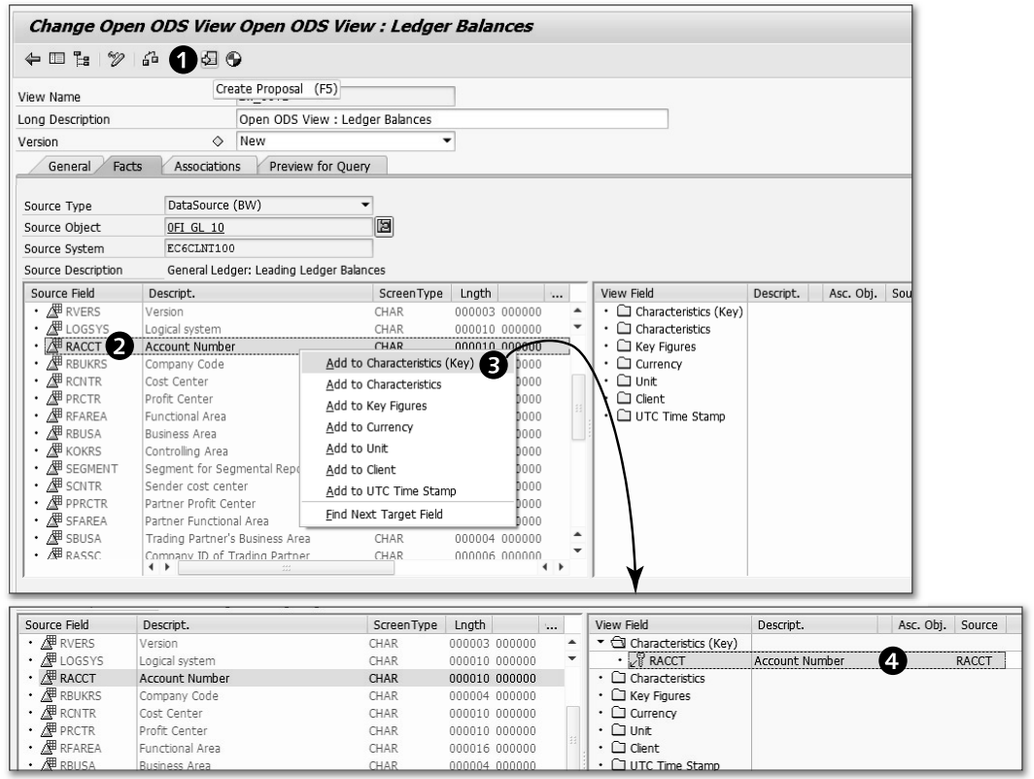


Figure 6.80 Adding to the Characteristics (Key) Section

If you plan to use an OOV to store data in a DSO as persistence, you need to move fields under the Characteristics (Key) section because these selected fields will define the key of the DSO. Our case requires storing data in a DSO, so we move

the required fields under the Characteristics (Key) section. If you're going to use an OOV as only a virtual query provider, you need not move any fields under the Characteristics (Key) section.

Before we move the fields under the Characteristics section, let's discuss how to open the Properties section for a field. There are number of ways to do it:

- From the menu bar, select Utilities • Position by single click (❶ of Figure 6.81). After selecting this option, whenever you click on a field under the View Field column, the Properties section opens for the selected field.
- From the menu bar, select Utilities • Position by double click. After selecting this option, whenever you double-click on a field under the View Field column, the Properties section opens for the selected field.
- Select a field under the View Field column for which you want to open the Properties section, and click on Show properties icon (❸ of Figure 6.81).

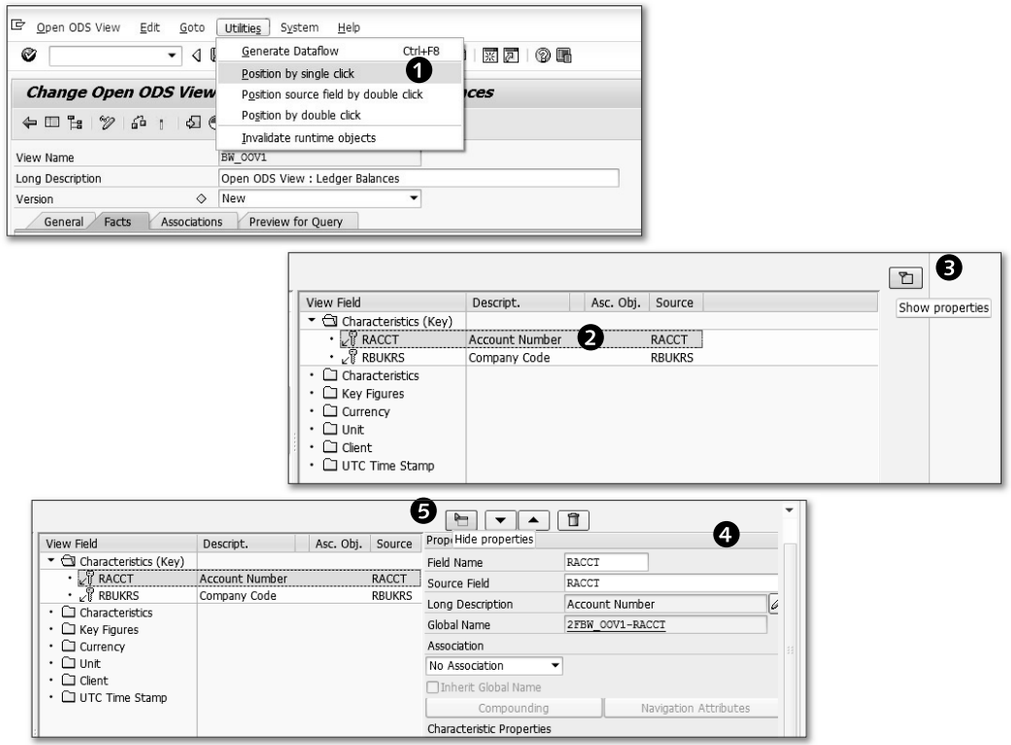


Figure 6.81 Opening the Properties for a Field

You can hide the Properties section by clicking the Hide properties icon ❺.

Now let's discuss how to add a field under the Characteristics section of View Field. Select the RBUSA (Business Area) field ❶, and from the context menu, select Add to Characteristics option ❷, as shown in Figure 6.82.

This action places the RBUSA field under the View Field section as shown in ❸ of Figure 6.82. For each field under View Field, you can set a number of properties. Open the Properties section ❹ using any of the methods explained earlier. You can set Association ❺, Characteristic Properties ❻, and Reporting Properties ❼ for each field. Each InfoObject created in SAP BW also has associated technical and semantic properties. You can take advantage of the associated properties of an InfoObject by associating fields of an OOV with the InfoObject. By associating the InfoObject with the fields, you can include InfoObject navigational fields and use the reporting properties in the design of the OOV.

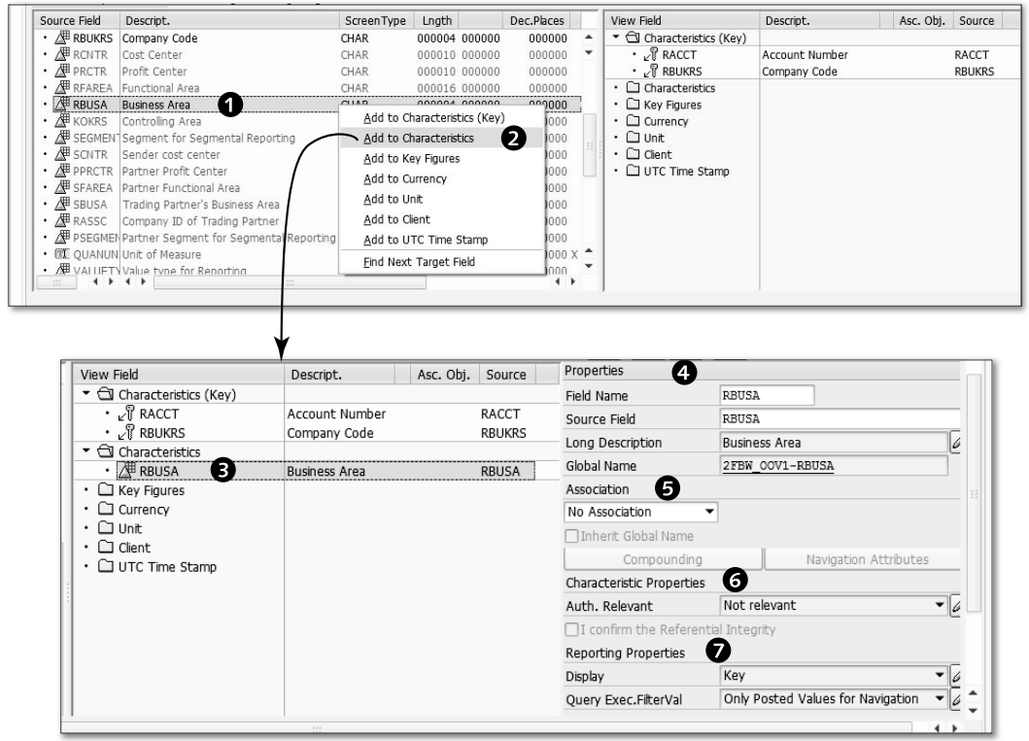


Figure 6.82 Adding to the Characteristics Section

You can also set the authorization relevance of the field, which helps in restricting certain data while viewing. Other properties are related to reporting, for example, displaying the data of fields as KEY, TEXT, or both.

As shown in Figure 6.83, you need to set the following fields when designing an OOV:

- CHARACTERISTICS (KEY)
  - RACCT
  - RBUKRS
- CHARACTERISTICS
  - RBUSA
  - RCNTR
  - KOKRS

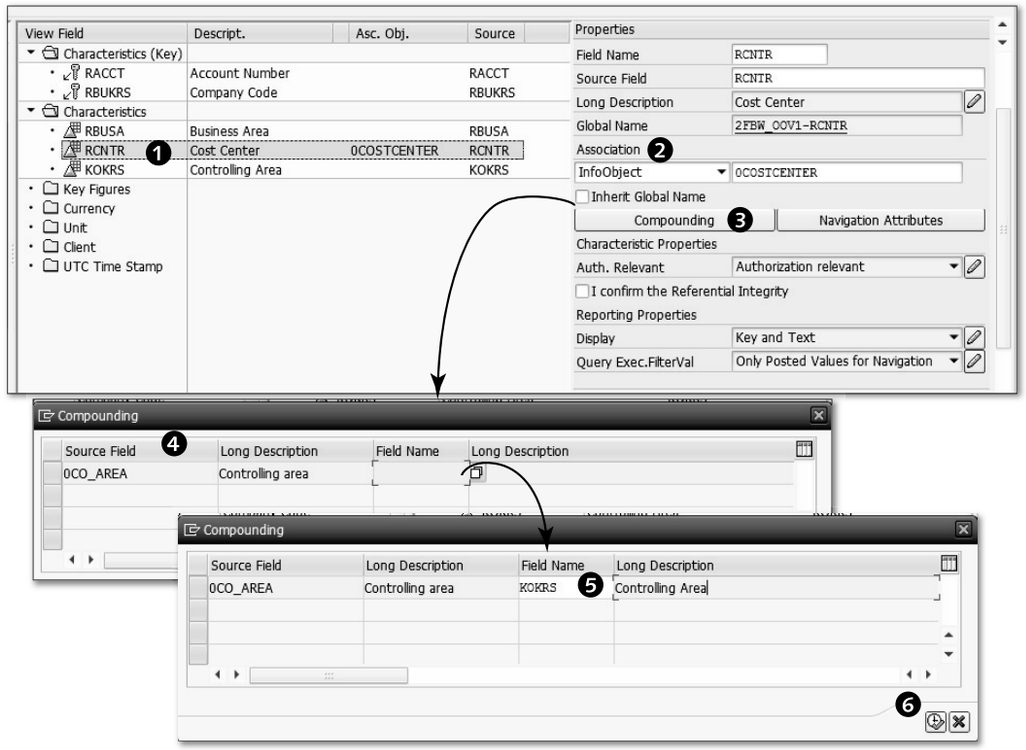


Figure 6.83 Setting Properties for a Field

Now let's see how to associate a field with an InfoObject and how to set a few properties. Select the RCNTR field from the VIEW FIELD section, and open the PROPERTIES section (❶ of Figure 6.83). In the ASSOCIATION dropdown, select INFOOBJECT ❷, enter "0COSTCENTER", and press the Enter key. This associates InfoObject 0COSTCENTER to the RCNTR field, which activates the COMPOUNDING and NAVIGATION ATTRIBUTES buttons ❸.

Next, click on COMPOUNDING to open the COMPOUNDING box ❹. InfoObject 0COSTCENTER is compounded by CONTROLLING AREA (this is the standard SAP relationship) as already shown in this popup box. InfoObject 0CO_AREA is already available, so you only need to provide the appropriate field name for this InfoObject. Enter "KOKRS" in the FIELD NAME box ❺. Click on the CONTINUE icon ❻.

You know from Chapter 3 that navigational attributes can be configured for InfoObject of type characteristics. Here, 0COSTCENTER has the type characteristics, and a few navigational attributes are defined for this InfoObject. While designing an OOV, SAP BW allows you to include navigational attributes so that it can be available while analyzing data on the OOV. To do this, click on the NAVIGATION ATTRIBUTES button as shown in ❶ of Figure 6.84.

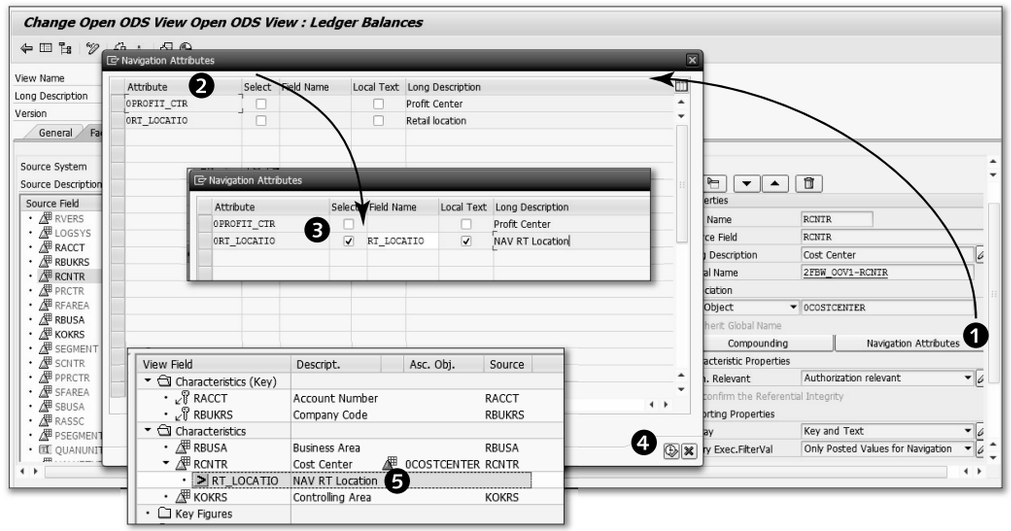


Figure 6.84 Adding Navigation Attributes

This action brings up the NAVIGATION ATTRIBUTES popup box ❷. Two navigation attributes are defined for InfoObject 0COSTCENTER as listed here. Select 0RT_LOCATIO by selecting the checkbox in the SELECT column next to 0RT_LOCATIO ❸. A field name is then proposed by the system. In this case, the field name is RT_LOCATIO. You can provide your own text for this field name as well. The same will appear in the query. To change the proposed text, select the LOCAL TEXT checkbox, and type your own text. We've entered "NAV RT Location" as the long description text. Click on CONTINUE ❹. The RT_LOCATIO field with the NAV RT LOCATION text is now available under the RCNTR field ❺.

6.9.2 Adding Key Figures

Now let's discuss how to add key figures to an OOV. Select the BALANCE field as shown in ❶ of Figure 6.85. Using the context menu, select ADD TO KEY FIGURES ❷. This action places the BALANCE field in the KEY FIGURES section under the VIEW FIELD column ❸. Each key figure is also associated with a number of properties ❹. A few of these properties are similar to characteristics, which are ASSOCIATION ❺ and REPORTING PROPERTIES, but some properties, such as AGGREGATION ❻, are only available to key figures.

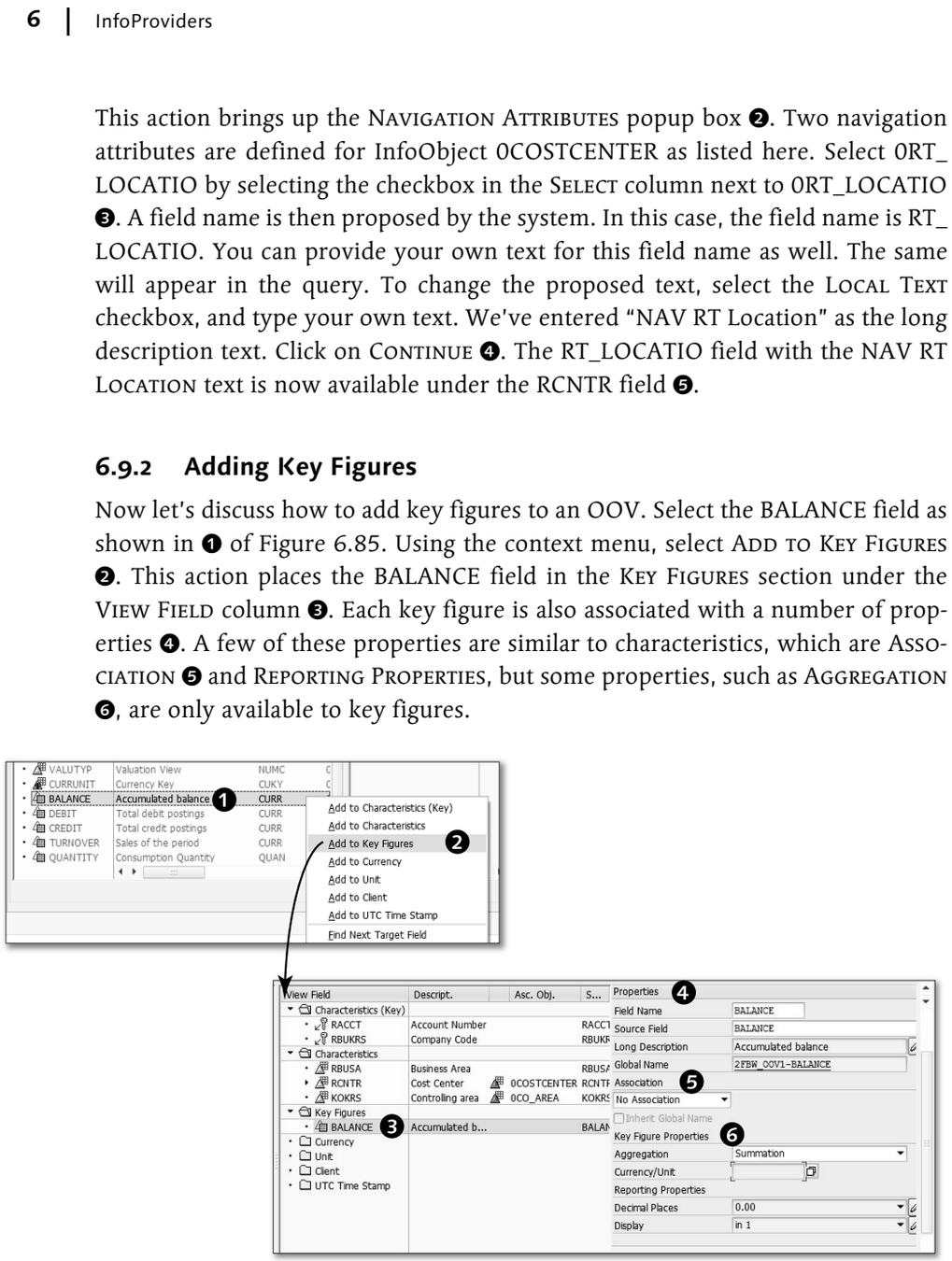


Figure 6.85 Adding Key Figure

Normally, a key figure is associated with a unit; for example, an expenditure during the year is 9 million US$. Here currency is the unit associated with the expenditure key figure, and the value is US$ for the currency unit. When designing an OOV, you can associate a unit for each key figure, as described next.

First, you need to add the field for unit under the VIEW FIELD section. For key figure BALANCE, the field associated for unit is RTCUR. Select the RTCUR field ❶, and from the context menu, select ADD TO UNIT ❷ as shown in Figure 6.86. This places the unit RTCUR field under the VIEW FIELD section under CURRENCY ❸. Now select the BALANCE field from the VIEW FIELD section ❹. Open the PROPERTIES section for the BALANCE field, and enter "RTCUR" in the CURRENCY/UNIT field ❺. This action associates the unit RTCUR field with key figure BALANCE.

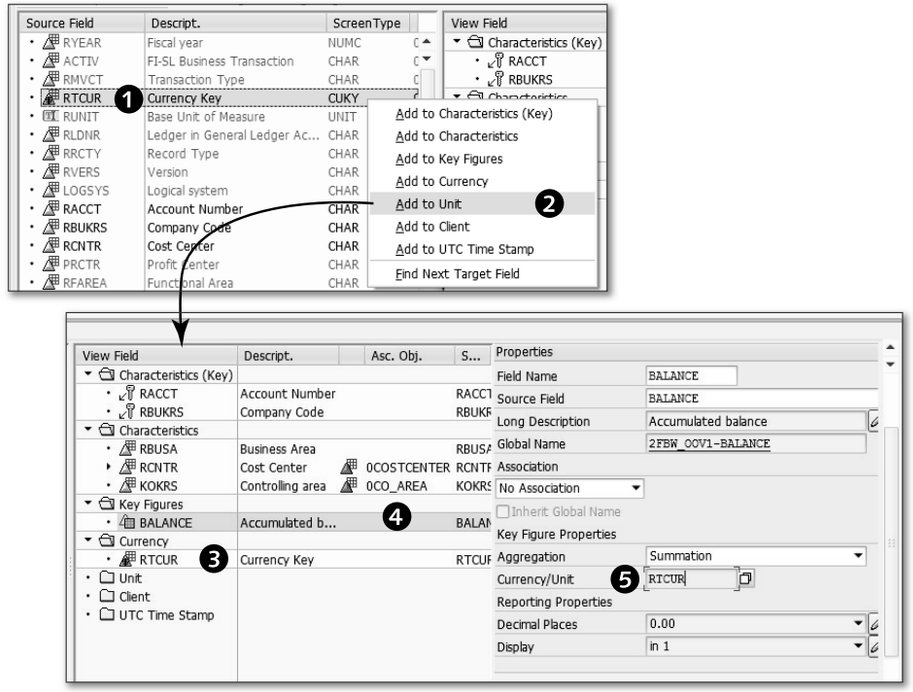


Figure 6.86 Adding to the Unit

6.9.3 Activating the Design

The example OOV design is now ready to be activated. Use the ACTIVATE icon as shown in ❶ of Figure 6.87 to make the OOV active ❷.

As indicated earlier, the OOV can also be used to store the data in a DSO. Let's see how the system helps you create a DSO and other related objects such as DTPs, transformations, and so on.

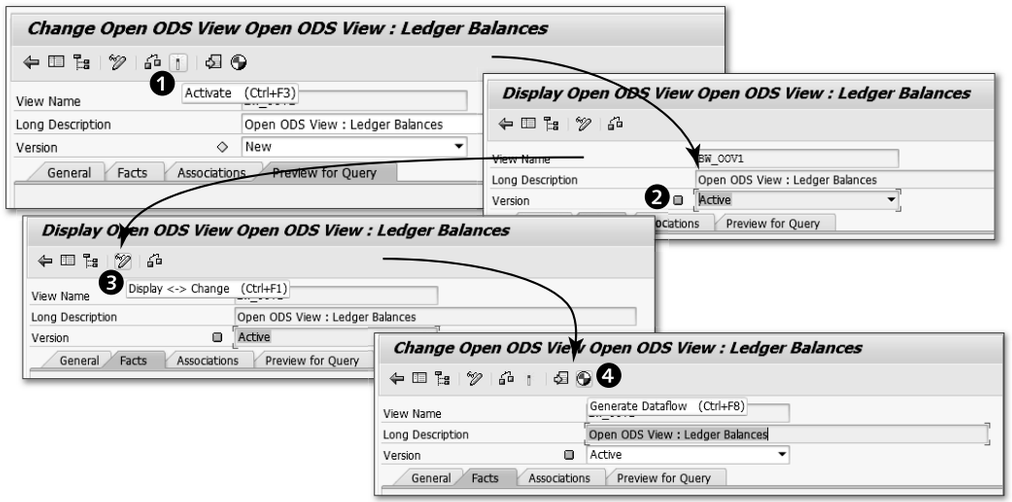


Figure 6.87 Activate and Generate Data Flow

You don't have to create the DSO from scratch; instead, you can create the DSO based on the OOV design. To do this, you need to go back to Change mode. Click on the DISPLAY < -- > CHANGE icon (refer to ❸ of Figure 6.87. Now you're in the CHANGE OPEN ODS VIEW screen. Click on the GENERATE DATAFLOW icon ❹. This action opens up a popup as shown in ❶ of Figure 6.88. The name of the DSO to be generated is proposed by the system, which, in our case, is 0FI_GL_1 ❷. Change this proposal to "BWOOV1" ❸. Also, in the DATA TYPES field, select BW DATA TYPES ❹, and keep the SOURCE OBJECT field set to AS IS ❺.

Click the CONTINUE icon ❻, which opens the CREATE DATA TRANSFER PROCESS screen to transfer data between DataSource 0FI_GL_10 and DSO BWOOV1 ❼. Keeping all the defaults as they are, click on the CONTINUE icon ❽. Another popup appears asking you to confirm the creation of a default transformation between DataSource 0FI_GL_10 and DSO BWOOV1.

Click on YES ❾ to create all the necessary objects, such as the DSO, DTP, and transformation. You can see your newly created OOV BW_OOV1 ❶ and DSO BWOOV1 ❷ under the INFOPROVIDER tree of your DATA WAREHOUSE WORKBENCH

screen as shown in Figure 6.89. This OOV can be used in your query designer tool to create queries.

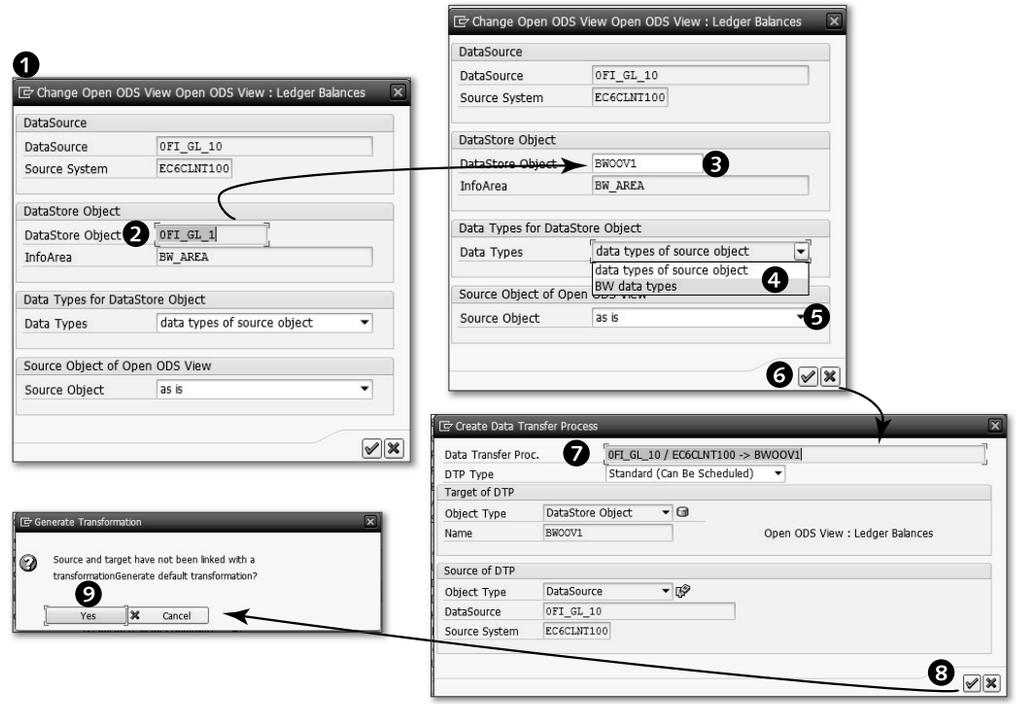


Figure 6.88 Automatic DSO/DTP/Transformation Generation for OOV

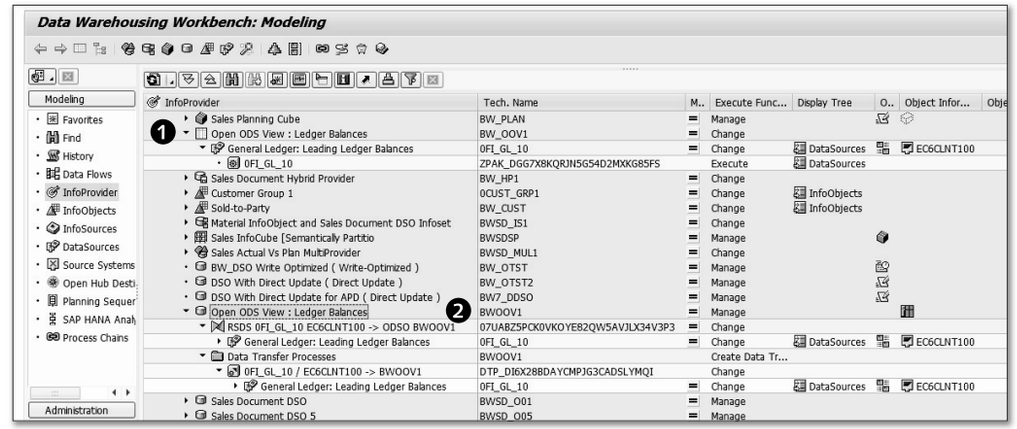


Figure 6.89 Newly Created OOV and DSO

6.9.4 Viewing Data with an Open ODS View

Now let's discuss how sample data can be viewed by default query using our newly developed OOV. In the Database Warehousing Workbench screen, select the OOV BW_OOV1, and then select Display from the context menu. The Display Open ODS View screen appears as shown in Figure 6.90.

On the menu bar, select Goto • Default Query ❶ to open the Query Monitor screen ❷. The query name is defaulted by the system ❸. In the Query Display field, there are a few display options, but for this example, choose HTML ❹. Now click on the Execute button ❺. This action uses the definition of OOV, goes to the source system, executes the DataSource, retrieves data, and displays the data as output ❻. Because we've taken only one key figure (Accumulated balance) in the definition of this OOV, it's displayed ❼. Now you move the Account Number and NAV RT Location from Free Characteristics to Rows ❽. Accumulated Balance ❾ is now shown for each Account Number and NAV RT Location.

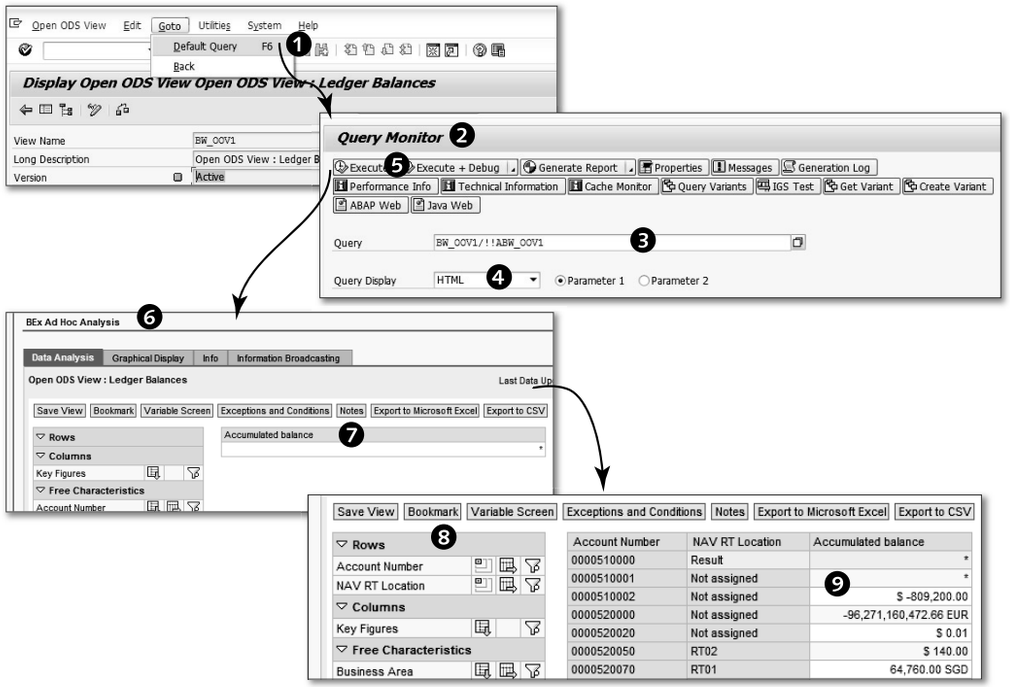


Figure 6.90 Querying on an OOV

6.10 Summary

In this chapter, we introduced you to several InfoProviders, including InfoProviders new to SAP BW 7.4. In certain cases, we discussed the procedures to create them and possible application scenarios for each of them; in other cases, we only briefly introduced them and then pointed you to the appropriate places in the book where we discuss them in more detail.

At this point, you should understand the concepts of modeling, data targets, and InfoProviders in SAP BW. With this foundation, we use the next chapter to explain the fundamental processes of extraction, transformation, and loading.

Contents

Index

A

B

C

D

E

F

G

H

I

J

K

L

M

N

O

P

Q

R

S

T

U

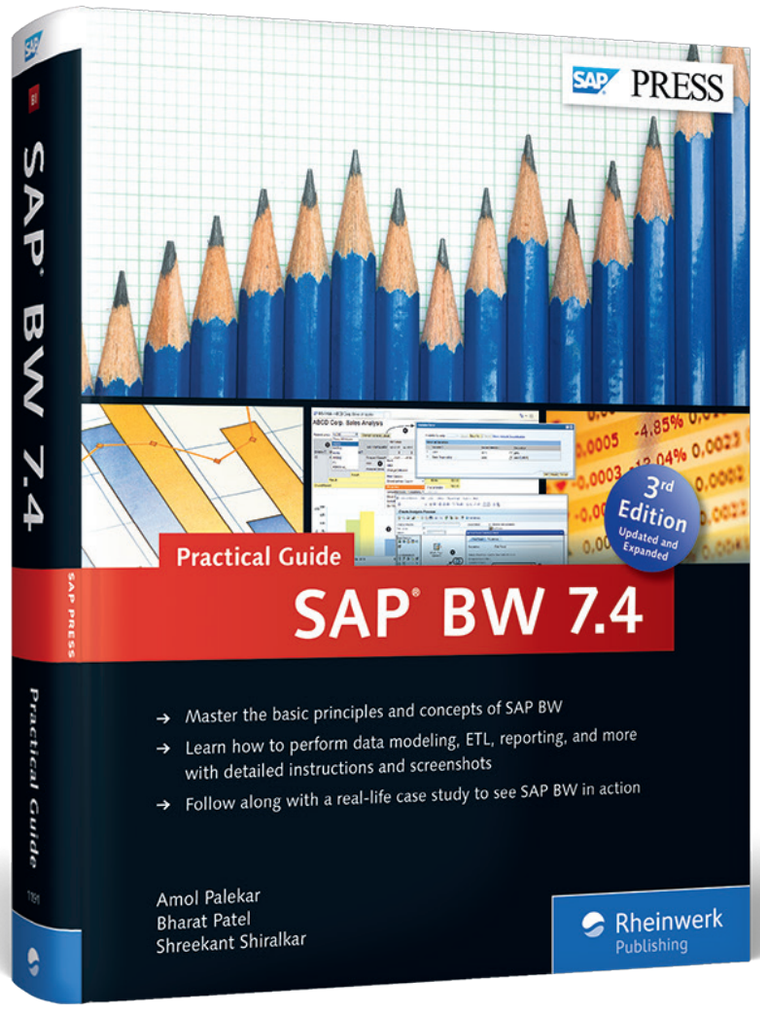
V

W

X

Z

SAP PRESS
First-hand knowledge.
Rheinwerk Publishing



Amol Palekar, Bharat Patel, Shreekant Shiralkar

SAP BW 7.4—Practical Guide

852 Pages, 2015, $69.95/€69.95
ISBN 978-1-4932-1191-3

www.sap-press.com/3733

Amol Palekar is Director of Analytics and Service Delivery at TekLink International Inc. He leads the Application Management Services practice in TekLink, and focuses on global delivery model and processes for BI application development and support engagements.

Bharat Patel has almost 20 years of IT experience, which covers the entire lifecycle management of data warehousing solutions, from evaluation and identification to upgrade and retirement. He is a regular faculty member at SAP Partner Academy and SAP Labs in India.

Sheekant Shiralkar is a senior management professional with experience in leading and managing business functions, as well as technology consulting. In a career spanning more than 27 years, he established, developed, and diversified business units for Fortune 500 companies.