

Updates to Chapter 5 of the Printed Book

This document contains updates to Chapter 5 of the printed book to add coverage of Advanced DSOs. These additions have been incorporated in the current version of the e-book.

5.4 Advanced DSOs

The ADSO is positioned to become the central modeling object in SAP BW. Note that this does not involve SAP HANA-optimized DSOs, whose usage have become obsolete. For this reason, it is no longer possible to convert DSOs to SAP HANA-optimized DSOs. Currently, the new ADSO is poised to replace all InfoProviders that provide data in separate tables—for instance, InfoCubes, DSOs, and write-optimized DSOs. The InfoProviders that have been used so far will be retained and usable for compatibility reasons. The ADSO can consist of simple fields as well as InfoObjects.

Compared to a classic DSO, the ADSO provides faster activation and load times and new modeling options, among other things. Request management has also been optimized for this object, which is why ADSOs are particularly suited for large amounts of data and frequent load processes. For the sake of clarity, we'll use the terms *ADSO* and *classic DSO* in the following sections.

5.4.1 Prerequisites and Usage Options for Advanced DSOs

You can use ADSOs as long as you already use SAP BW 7.4 SPS 08 and SAP HANA 1.0 in revision 83 or higher. However, we advise you not to utilize ADSO before SPS 09.

It is also important that you import all necessary SAP notes. Because they involve a high number of corrections, SAP recommends that you not import them manually using Transaction SNOTE, but rather in a partially automated approach, including a special note analysis program. To

Import necessary
SAP Notes

implement the corrections, first download program Z_BW_NOTE_ANALYZER attached to SAP Note 2094475. Then execute the program via Transaction SE38. The program requires an XML file that includes information about which corrections must be imported to the SAP BW on SAP HANA system. The relevant XML file and a short instruction are available in SAP Note 2070577. Once your system is updated, you also require the SAP BW modeling tools in version 1.5 or higher, because you cannot create ADSOs in the classic SAP GUI, but only via the SAP BW modeling tool (see Section 5.2.1).

Structure The ADSO can consist of up to three tables: *inbound table*, *change log*, and *active data table*. Depending on the intended purpose, however, you don't always need to create all three tables. The data is first loaded to the inbound table. Depending on the purpose, it is then directly read from this table or from the active data table after an activation process. The change log provides the change history, which is required for updating delta records in other InfoProviders.

Areas of usage Currently, the ADSO is used solely for reporting purposes. In the future, however, it is supposed to be available for planning scenarios, as well. It can be deployed in various layers of the LSA++ architecture. To optimally complete the various tasks, you can specify *modeling properties* upon creation. You can adapt these properties later, if required. [Figure 5.35](#) shows an overview of possible modeling properties.



Figure 5.35 Options for Creating an ADSO

Modeling properties If you select the **ACTIVATE/COMPRESS DATA** option, the data is transferred to the active data table after it has been loaded to the inbound table. Here, you can choose from three options:

▶ WRITE CHANGE LOG

If you select this option, the change history is saved in the change log and can then be used for delta extraction.

▶ KEEP INBOUND DATA, EXTRACT FROM INBOUND TABLE

If you select this option, no data is retained in the change log. The extraction thus always reverts to the data of the inbound table, irrespective of whether you load data in the delta or full procedure.

▶ UNIQUE DATA RECORDS

With this option, you specify that only unique data records are loaded to the ADSO. In this case, the active data table contains unique data records only. The data is not aggregated.

With the ALL CHARACTERISTICS ARE KEY, REPORTING ON UNION OF INBOUND AND ACTIVE TABLE option, you specify that all characteristics are part of the key. In this case, a BEx query accesses both the inbound table and the active data table (that is, a union operation is carried out via both tables). You should make sure that only additive deltas are loaded. When you choose this option, the data is aggregated as if you deployed an InfoCube.

With these modeling options, you can deploy the ADSO optimally in the various layers of the LSA++. SAP has created templates (model templates) so that you don't have to make the required settings yourself. These templates provide the optimal settings for each layer of the LSA++ (see [Figure 5.36](#)).

Usage pursuant to LSA++

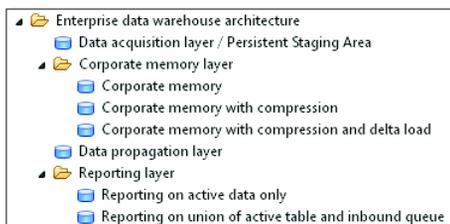


Figure 5.36 Model Templates with Predefined Properties

[Table 5.1](#) provides an overview of the templates offered and their respective purposes and predefined properties. Note that the naming of the templates is based on the classic LSA.

Model templates

Template	Modeling Properties	Description
DATA ACQUISITION LAYER	none	Only fields are used for creating the ADSO. It consists of only the inbound table and can be compared with the Persistent Staging Area (PSA). In this context, the ADSO serves as an inbound storage for data from the source systems in SAP BW.
CORPORATE MEMORY	none	Fields and InfoObjects can be used for creating the ADSO. It consists of the inbound table only and can be deployed as a replacement for a write-optimized, classic DSO. The inbound table is accessed to execute a BEx query and for extraction.
CORPORATE MEMORY WITH COMPRESSION	ACTIVATE/COMPRESS DATA	The ADSO consists of the inbound table and the table for active data. The active table is accessed to execute a BEx query. For extraction, however, the inbound table and the table for active data are used.
CORPORATE MEMORY WITH COMPRESSION AND DELTA LOAD	ACTIVATE/COMPRESS DATA • KEEP INBOUND DATA, EXTRACT FROM INBOUND TABLE	The active table is accessed to execute a BEx query. For extracting data, however, only the inbound table is used.
DATA PROPAGATION LAYER	ACTIVATE/COMPRESS DATA • WRITE CHANGE LOG	The data propagation layer forms the consistent foundation to update data. The object modeled this way corresponds to a classic DSO. When a BEx query is executed, it accesses the active table. The change log is accessed in case of extraction. Optionally, you can choose the UNIQUE DATA RECORDS property if you load unique data records only.

Table 5.1 ADSO Templates and Their Predefined Properties

Template	Modeling Properties	Description
REPORTING ON ACTIVE DATA ONLY	ACTIVATE/COMPRESS DATA	The ADSO contains the data that is accessed for analysis in BEx queries. For reporting, the data is read from the active data table only. In case of extraction, however, the active data table and the inbound table are accessed.
REPORTING ON UNION OF ACTIVE TABLE AND INBOUND QUEUE	<ul style="list-style-type: none"> ▶ ACTIVATE/COMPRESS DATA ▶ ALL CHARACTERISTICS ARE KEY, REPORTING ON UNION OF INBOUND AND ACTIVE TABLE 	The object modeled this way corresponds to a standard InfoCube. When executing a BEx query and for extraction, the active data table and the inbound table are accessed.

Table 5.1 ADSO Templates and Their Predefined Properties (Cont.)

5.4.2 Creating Advanced DSOs

You can create an ADSO using SAP BW modeling tools only. To do so, start Eclipse or SAP HANA Studio with the corresponding enhancements (see Section 5.2.1). First, open the BW MODELING perspective under WINDOWS • OPEN PERSPECTIVE • OTHER. Alternatively, you can directly click on the BW MODELING button. In the PROJECT EXPLORER, create a new SAP BW project by selecting NEW • PROJECT • BW PROJECT in the context menu. In the next step, select an SAP BW system and enter your login credentials. This opens the view of InfoAreas in the PROJECT EXPLORER (see [Figure 5.37](#)).

Create an SAP BW project in Eclipse



Figure 5.37 Project Explorer After You've Created an SAP BW Project

Create ADSO To now create an ADSO, right-click on an InfoArea. Various InfoProviders are offered for creation in the context menu. Select **DATASTORE OBJECT (ADVANCED)** (see [Figure 5.38](#)).

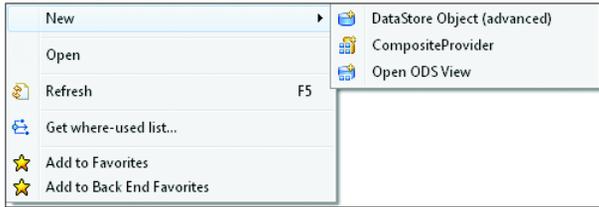


Figure 5.38 Calling Wizards for Creating an ADSO

In the creation wizard, enter a name and a description of the new ADSO (as illustrated in [Figure 5.39](#)). By means of the **BROWSE** button next to the **OBJECT TEMPLATE** field, you can select existing SAP BW InfoProviders as a template for defining the modeling options. This doesn't need to be an ADSO. You can also use any existing classic DSO, an InfoObject, or an InfoCube as a template.

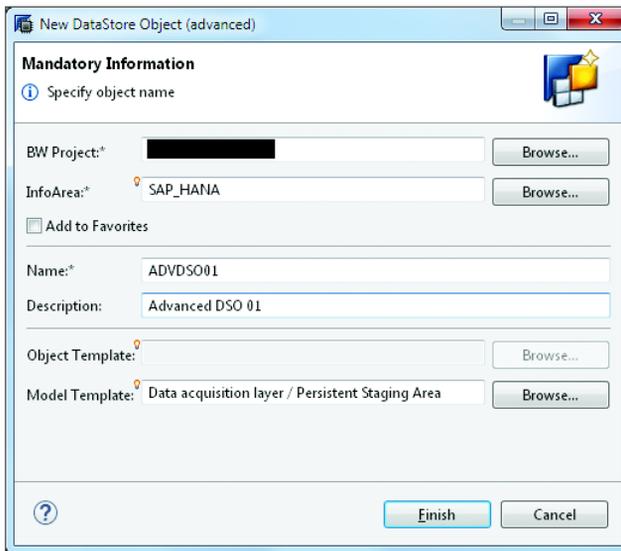


Figure 5.39 Creating an ADSO Based on an InfoProvider or Template

If you select an InfoProvider as a template, the MODEL TEMPLATE field below is grayed out and no longer available for usage. If you want to create an ADSO without using an SAP BW object as a template, you can use the BROWSE button next to the MODEL TEMPLATE field to select one of the templates. If, for example, you want to use your ADSO as an inbound storage in SAP BW, you can select the DATA ACQUISITION LAYER/PERSISTENT STAGING AREA template under MODEL TEMPLATE (see [Figure 5.39](#)). Then click on FINISH.

Like a classic DSO, you can model an ADSO via InfoObjects. For this purpose, you can search for InfoObjects via a search mask and add them to the ADSO (see [Figure 5.40](#)).

Modeling via Info-Objects

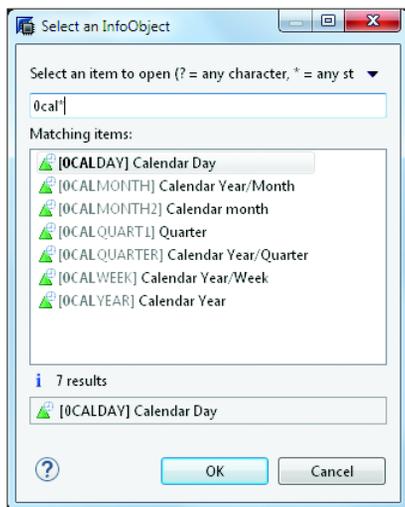


Figure 5.40 Adding InfoObjects to an Advanced DSO

As a new option, you can use fields with simple data types instead of InfoObjects. To do so, go to the DETAILS tab and click the ADD FIELD button (see [Figure 5.41](#)). To define the field properties, you can enter various pieces of information on the right-hand side of the screen. First, you should enter a description for the field under DESCRIPTION. Under IDENTIFY, you can specify in the WITH dropdown menu whether you want to use an InfoObject for the definition. If not, keep the setting FIELD.

Modeling via fields

Now, enter the DATA TYPE and its length. Then you can select an aggregation type from the dropdown next to AGGREGATION, provided that you have defined the appropriate data types for this purpose.

Define key fields Fields can be sorted as required using groups. To define a key, click on the KEY COMPONENTS button. In the dialog window that opens, select the appropriate key fields, click on the ADD button, and then choose OK. To delete fields again, click on DELETE. Finally, activate your ADSO using the  icon.

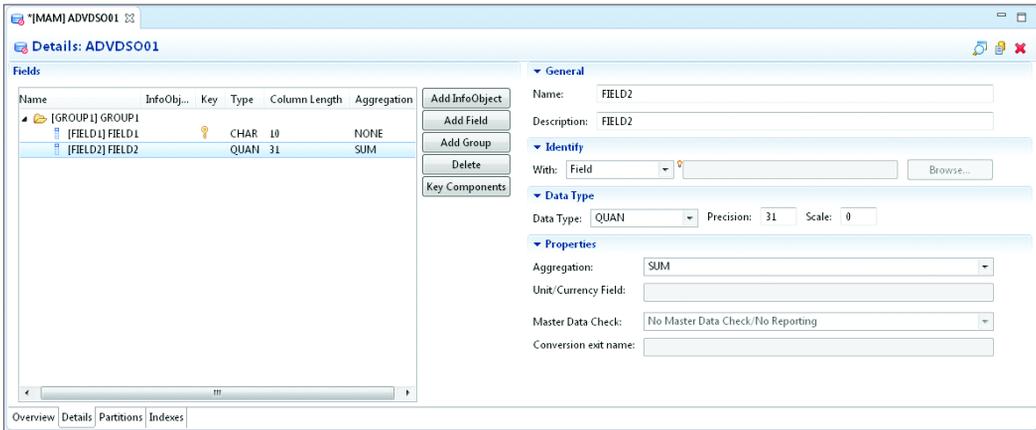


Figure 5.41 Using Fields for Modeling

Partitioning By means of partitioning, you can split data of an ADSO into various small, physically separated units. This separation enables you to improve the performance during data analysis. Partitioning may also be required if you are loading a large amount of data, because a table can only contain a maximum of 2 billion data records.

Indexes You can also create secondary indexes for an ADSO. The index is usually applied to the active data table. If this table doesn't exist, the index is applied to the inbound table.

5.5 Additional Chapter 5 Updates

5.5.1 Update to Section 5.1.2, Code Push-Down in Transformations

When you activate a transformation, the system checks whether it can be executed in SAP HANA. Whether this will be possible basically depends on the operations and objects used. The following operations and objects are not supported for executing the transformation in SAP HANA:

- ▶ Queries, as InfoProviders are not supported as a source.
- ▶ ABAP routines are not supported (rule type routine, routines for characteristics, start, end, and expert).
- ▶ Rule groups are not supported.
- ▶ The full key must be provided for reading from DSOs.
- ▶ Nearline connection is not supported.

DSOs (standard and write-optimized), ADSOs, semantically partitioned objects, and Open Hub destinations are supported as target objects.

5.5.2 Update to Section 5.1.3

As described in Section 5.1, you no longer need to build reports on InfoCubes for performance reasons. Instead, you can use existing classic DSOs as the source of BEx queries in the data model. When you create new data models, you can also use the new ADSOs instead of InfoCubes or classic DSOs.

