Understanding DSO (DataStore Object) Part 2: Write-Optimized DSO

Applies to:

SAP NetWeaver BW.

Summary

This is the second of a three part series of documents containing each and every detail about DSOs and their implementation for beginners in SAP BI. For advanced users also, this document has many small but usually ignored helpful facts. Other parts: Part 1, Part 3.

This document is based on Write-Optimized DSO and their implementation. Every minute detail has been included.

Author: Shyam Uthaman

Company: Accenture Services Pvt. Ltd.

Created on: 5 May 2011

Author Bio

Shyam Uthaman is working as SAP-BI Consultant.

He is working simultaneously on multiple projects for different clients in Accenture.

Table of Contents

DataStore Object 3
Definition 3
Use 3
Definition 4
Use 4
Structure 4
Use in BEx Queries 5
Designing a Write-Optimized DSO 5
Settings in Write-Optimized DSO 9
Type of DataStore Object 9
Do Not Check Uniqueness of Data 9
Including Key Fields and Data Fields in the DSO 10
Final Steps 10
Related Content 11
Disclaimer and Liability Notice 12

DataStore Object

Definition

A DataStore object serves as a storage location for consolidated and cleansed transaction data or master data on a document (atomic) level.

This data can be evaluated using a BEx query.

A DataStore object contains key fields (such as document number, document item) and data fields that, in addition to key figures, can also contain character fields (such as order status, customer). The data from a DataStore object can be updated with a delta update into InfoCubes (standard) and/or other DataStore objects or master data tables (attributes or texts) in the same system or across different systems.

Unlike multidimensional data storage using InfoCubes, the data in DataStore objects is stored in transparent, flat database tables. The system does not create fact tables or dimension tables.

Use

Overview of DataStore Object Types

Type	**Structure**	**Data Supply**	**SID Generation**
Standard DataStore Object	Consists of three tables: activation queue, table of active data, change log	From data transfer process	Yes
Write-Optimized DataStore Objects	Consists of the table of active data only	From data transfer process	No
DataStore Objects for Direct Update	Consists of the table of active data only	From APIs	No

Write-Optimized DataStore Objects

Definition

A DataStore object that consists of just one table of active data. Data is loaded using the data transfer process.

Use

Data that is loaded into write-optimized DataStore objects is available immediately for further processing.

They can be used in the following scenarios:

- You use a write-optimized DataStore object as a temporary storage area for large sets of data if you are executing complex transformations for this data before it is written to the DataStore object. The data can then be updated to further (smaller) InfoProviders. You only have to create the complex transformations once for all data.
- You use write-optimized DataStore objects as the EDW layer for saving data. Business rules are only applied when the data is updated to additional InfoProviders.

The system does not generate SIDs for write-optimized DataStore objects and you do not need to activate them. This means that you can save and further process data quickly. Reporting is possible on the basis of these DataStore objects. However, we recommend that you use them as a consolidation layer, and update the data to additional InfoProviders, standard DataStore objects, or InfoCubes.

Structure

Since the write-optimized DataStore object only consists of the table of active data, you do not have to activate the data, as is necessary with the standard DataStore object. This means that you can process data more quickly.

The loaded data is not aggregated; the history of the data is retained. If two data records with the same logical key are extracted from the source, both records are saved in the DataStore object. The record mode responsible for aggregation remains, however, so that the aggregation of data can take place later in standard DataStore objects.

The system generates a unique technical key for the write-optimized DataStore object. The standard key fields are not necessary with this type of DataStore object. If there are standard key fields anyway, they are called semantic keys so that they can be distinguished from the technical keys. The technical key consists of the Request GUID field (0REQUEST), the Data Package field (0DATAPAKID) and the Data Record Number field (0RECORD). Only new data records are loaded to this key.

You can specify that you do not want to run a check to ensure that the data is unique. If you do not check the uniqueness of the data, the DataStore object table may contain several records with the same key. If you do not set this indicator, and you do check the uniqueness of the data, the system generates a unique index in the semantic key of the InfoObject. This index has the technical name "KEY". Since write-optimized DataStore objects do not have a change log, the system does not create delta (in the sense of a before image and an after image). When you update data into the connected InfoProviders, the system only updates the requests that have not yet been posted.

Use in BEx Queries

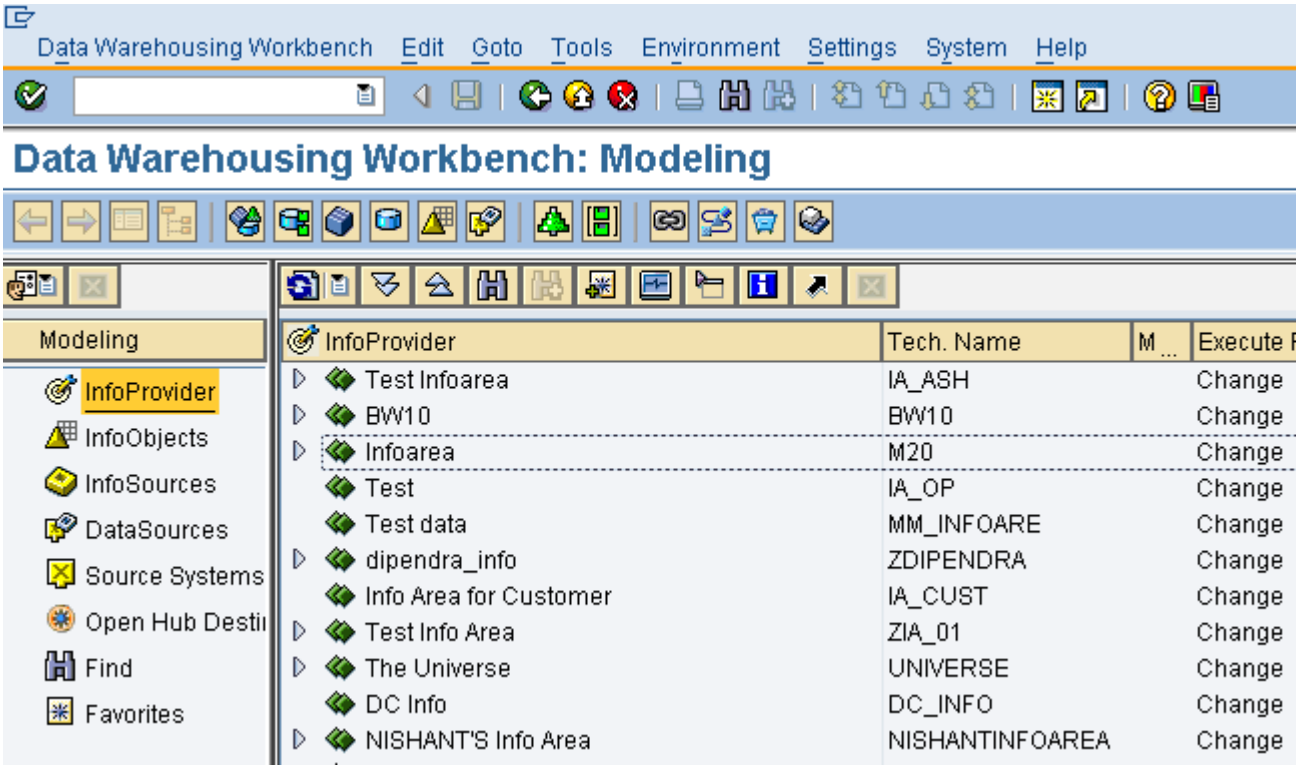
For performance reasons, SID values are not created for the characteristics that are loaded. The data is still available for BEx queries. However, in comparison to standard DataStore objects, you can expect slightly worse performance because the SID values have to be created during reporting.

If you want to use write-optimized DataStore objects in BEx queries, we recommend that they have a semantic key and that you run a check to ensure that the data is unique. In this case, the write-optimized DataStore object behaves like a standard DataStore object. If the DataStore object does not have these properties, you may experience unexpected results when the data is aggregated in the query.

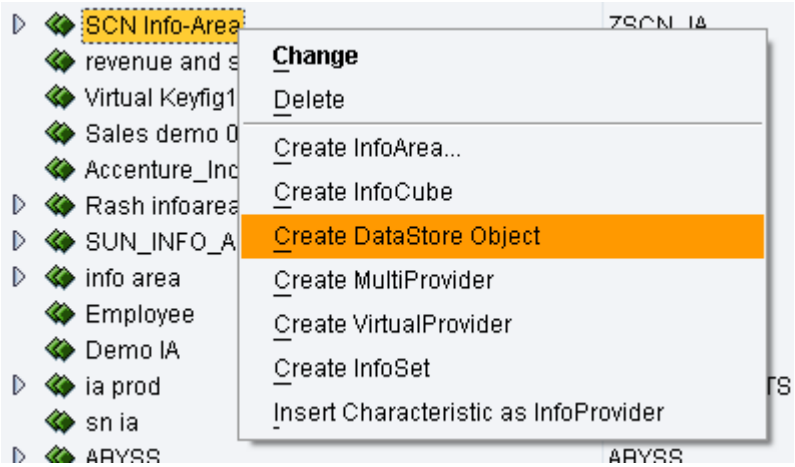
Designing a Write-Optimized DSO

To create a Write-Optimized DSO, first go to **RSA1** transaction screen.

You will reach the DWW screen where you have to select **InfoProvider** under the **Modeling** pane present on the left side of the screen as shown below.



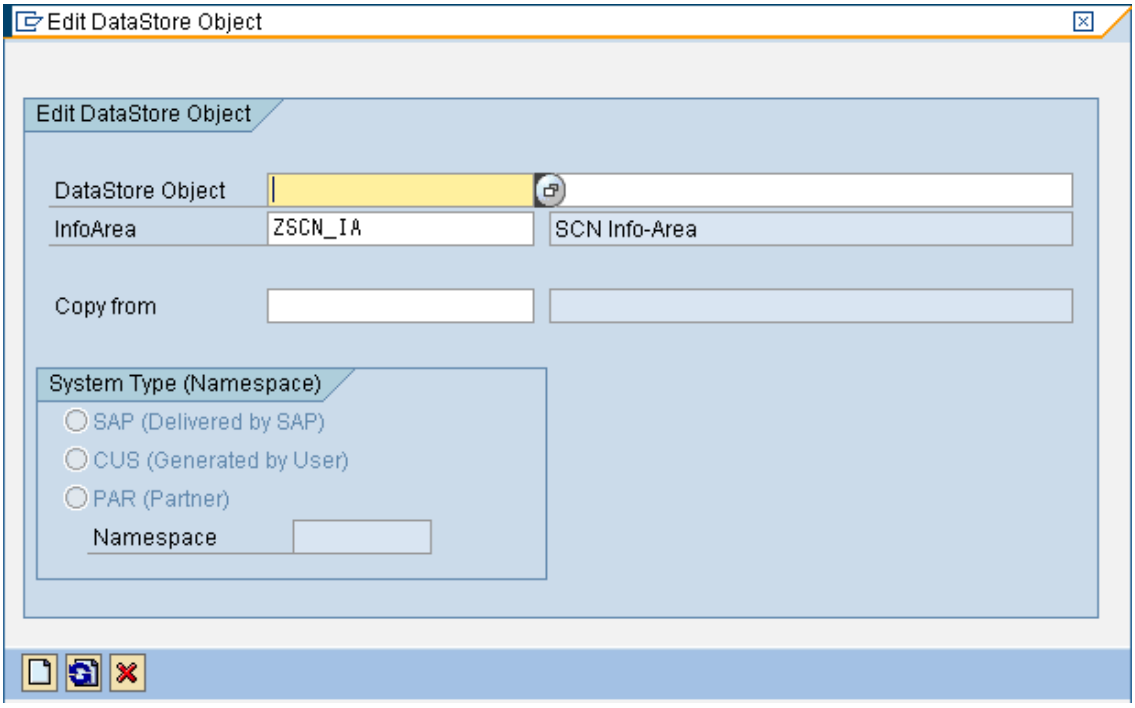
Now right click on your InfoArea and select **Create DataStore** Object from the context menu as shown below.



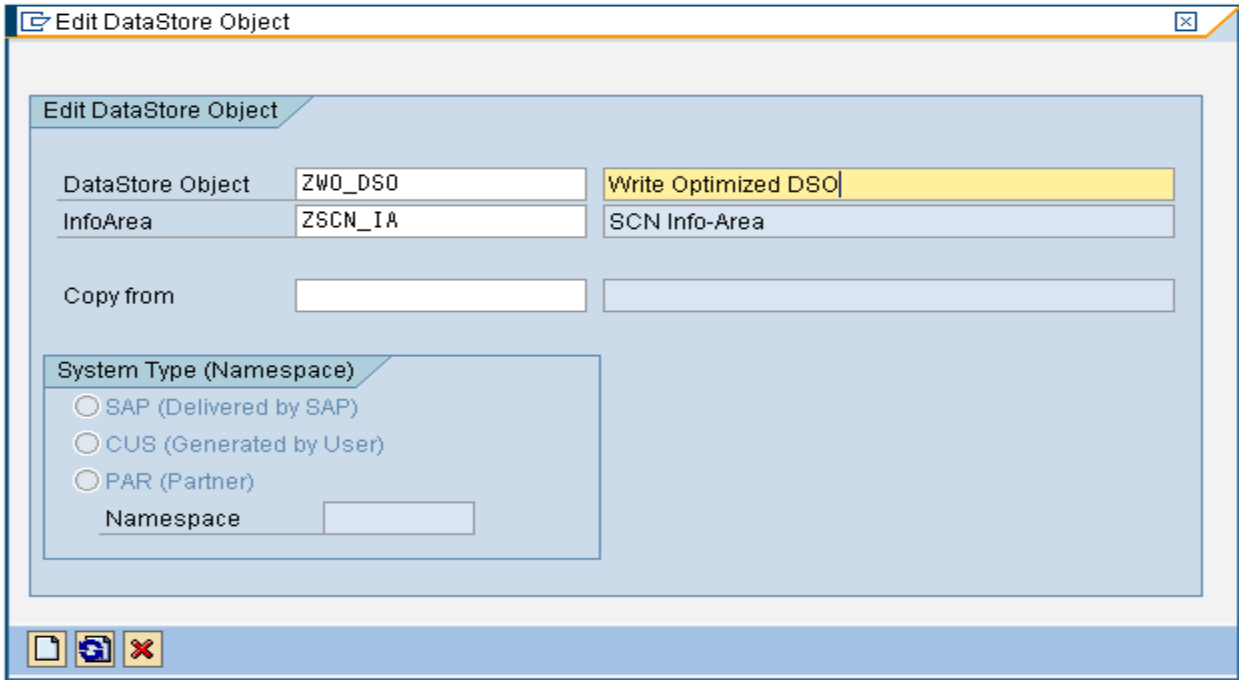
You will get the following window wherein you are required to fill in the technical name and description of the DSO to be created.

SAP has also provided an option of copying the entire DSO structure from another DSO using the **Copy from** Text-Field. This structure can be modified later.

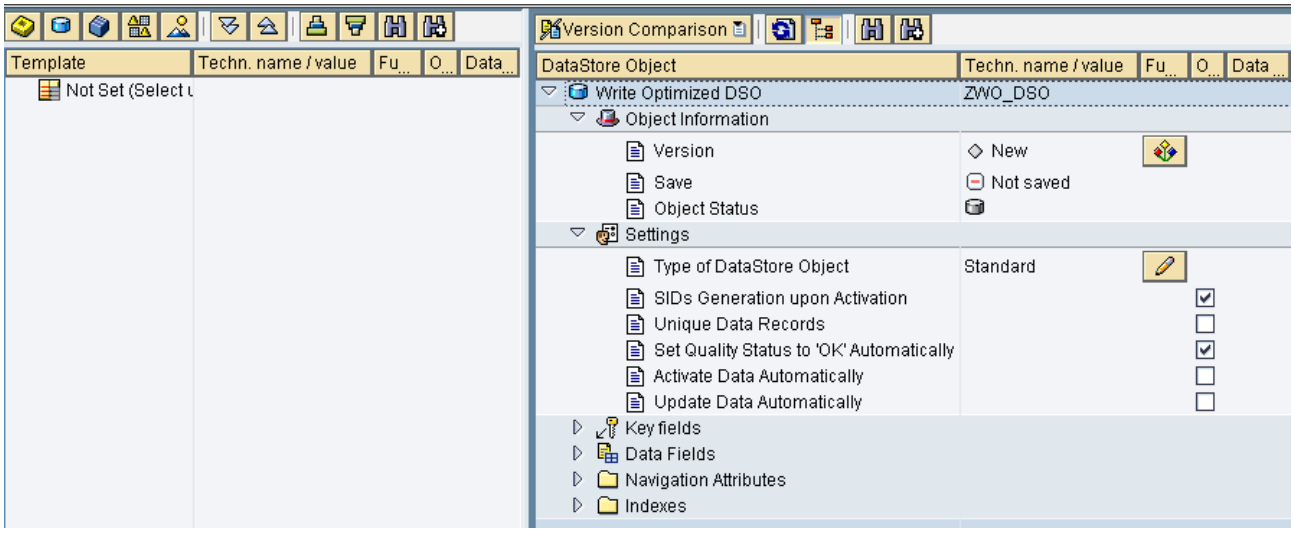
We will use the Structure of the Standard DSO which we created in the first Part of the document for matters of simplicity and consistency.



The filled in details are shown below. Press the Create button to continue.

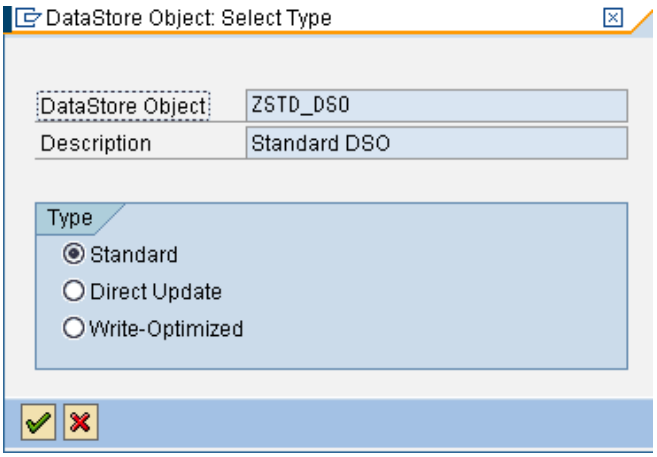


You will reach the following screen for editing the DSO.

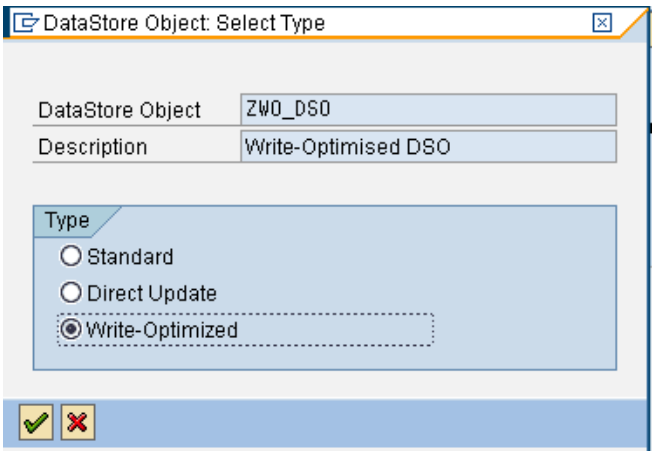


By Default, the DSO type is created as a standard type. This can be changed by clicking on the Change icon.

You will get the following pop-up.



Now select the radio-button for Write-Optimized DSO, and go back to the previous screen by pressing  .



Now, the DSO has become a Write Optimized one as seen below.

DataStore Object	Techn. name / value	Fu...	O...	Data ...	L	Key Fi...	C...	N...	Ag...	Ex...
Write Optimised DSo	ZWO_DSO									
Object Information										
Version	New									
Save	Not saved									
Object Status										
Settings										
Type of DataStore	Write-Optimized									
Do Not Check Un			☐							
Technical Key (Gener										
Semantic Key										
PRODUCT	ZPROD_SW			CHAR	60					
CUSTOMER	ZCUS_SW			CHAR	10					
COST ELEMENT	ZCEL_SW			CHAR	10					
cost character	ZCCH_SW			CHAR	20					
Sales document	0DOC_NUMBER			CHAR	10					
Calendar Year/We	0CALWEEK			NUMC	06					
Data Fields										
COST	ZCOST_SW			CURR	09	Amount	☑	☐	SUM	SUM
REVENUE	ZREV_SW			CURR	09	Amount	☑	☐	SUM	SUM
QUANTITY	ZQUAN_SW			QUAN	09	Quantity	☑	☐	SUM	SUM
Navigation Attributes										
Indexes										

Settings in Write-Optimized DSO

As you can see, the following settings are available in a Write-optimized DSO.

All of them will be explained one-by-one below in detail.

Settings		
Type of DataStore Object	Write-Optimized	
Do Not Check Uniqueness of Data		☐

Type of DataStore Object

This setting has been explained above while modifying the type of DSO from Standard to Write-Optimized.

Do Not Check Uniqueness of Data

By default, this option is not checked. In this case, a unique index called key is created with the fields included in the semantic Key section. While loading the data, Do Not Check Uniqueness of Data is checked with respect to fields in the semantic key. If this indicator is checked, the key index isn't generated, and the DSO can have several records with the same semantic key value. This significantly improves the loading performance.

Including Key Fields and Data Fields in the DSO

The DSO contains 2 kinds of fields namely, the Semantic key and the data field.

The combination of Semantic keys is responsible for :

Identifying error in incoming records or Duplicate records

Protecting Data Quality such that all subsequent Records with same key are written into error stack along with incorrect Data Records.

Processing the error records or duplicate records, Semantic Group is defined in DTP.

Note: if we are sure there are no incoming duplicate or error records, Semantic Groups need not be defined.

All other objects can be included as data fields.

Similar to a Standard DSO, the same 2 ways are used to provide input InfoObjects into the DSO:

- Using Templates
- Using Direct Input

We have explained these methods in detail in the previous document.

Final Steps

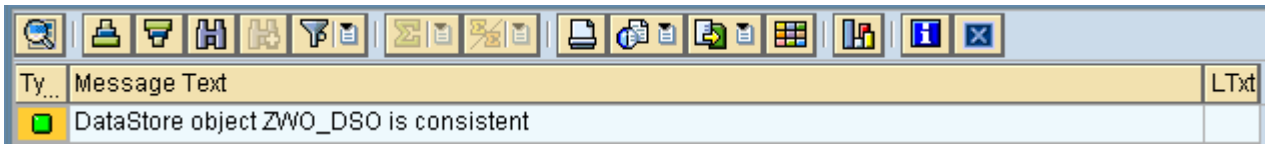
Now our DSO structure design is complete.

Now we follow through the usual routine of Save, Check and Activate.

Save using the button.

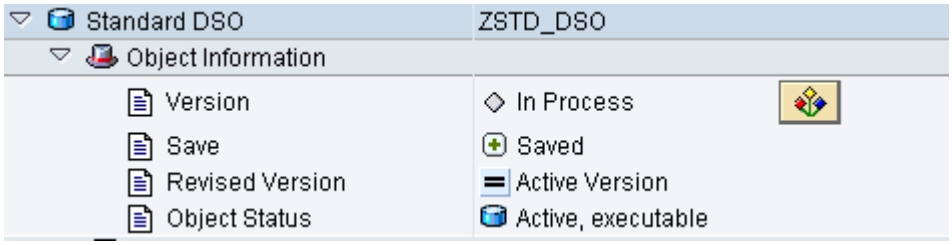
Press to check for errors.

The following message confirms that there are no errors in design.



Press the button to activate the DSO.

The Object Information menu now shows the DSO as active.



The Write-Optimized DSO design is now complete.

Related Content

http://help.sap.com/saphelp_nw04s/helpdata/en/f9/45503c242b4a67e10000000a114084/content.htm

http://help.sap.com/saphelp_nw70/helpdata/en/b6/de1c42128a5733e10000000a155106/content.htm

http://wiki.sdn.sap.com/wiki/display/BI/Write+Optimized+DSO

Understanding DSO (DataStore Object) Part 1: Standard DSO

Understanding DSO (DataStore Object) Part 3: Direct Update DSO

Disclaimer and Liability Notice