First Guidance... Using SAP HANA SQLScript in SAP BW Transformations

Applicable Releases:
SAP BW 7.4 or 7.5, powered by SAP HANA

This First Guidance document is part of a series of documents that should help to better understand the various concepts of SAP BW powered by SAP HANA.
The documents are still “work in progress”, so these guides are not intended to be exhaustive so far, and might never be. The purpose of these documents is to deliver additional information besides SAP Help and blogs to get a better understanding of the concepts of SAP BW powered by SAP HANA.

Version 1.0
January 2016
### Document History

<table>
<thead>
<tr>
<th>Document Version</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.0</td>
<td>First official release of this guide</td>
</tr>
</tbody>
</table>
### Typographic Conventions

<table>
<thead>
<tr>
<th>Type Style</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Example Text</strong></td>
<td>Words or characters quoted from the screen. These include field names, screen titles, pushbuttons labels, menu names, menu paths, and menu options. Cross-references to other documentation</td>
</tr>
<tr>
<td><strong>Example text</strong></td>
<td>Emphasized words or phrases in body text, graphic titles, and table titles</td>
</tr>
<tr>
<td><strong>Example text</strong></td>
<td>File and directory names and their paths, messages, names of variables and parameters, source text, and names of installation, upgrade and database tools.</td>
</tr>
<tr>
<td><strong>Example text</strong></td>
<td>User entry texts. These are words or characters that you enter in the system exactly as they appear in the documentation.</td>
</tr>
<tr>
<td><code>&lt;Example text&gt;</code></td>
<td>Variable user entry. Angle brackets indicate that you replace these words and characters with appropriate entries to make entries in the system.</td>
</tr>
<tr>
<td><strong>EXAMPLE TEXT</strong></td>
<td>Keys on the keyboard, for example, F2 or ENTER.</td>
</tr>
</tbody>
</table>

### Icons

<table>
<thead>
<tr>
<th>Icon</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>🚨</td>
<td>Caution</td>
</tr>
<tr>
<td>🎨</td>
<td>Note or Important</td>
</tr>
<tr>
<td>📌</td>
<td>Example</td>
</tr>
<tr>
<td>🔄</td>
<td>Recommendation or Tip</td>
</tr>
</tbody>
</table>
1. **Business Scenario**

You are looking for a way to define custom transformation rules which cannot be implemented using standard BW functionality and you want to leverage the power of the SAP HANA database platform. Typically, custom transformation rules are implemented using ABAP coding routines – either for individual fields or as start, end, or expert routines. However, these routines are always processed on the BW application server. With SAP BW powered by SAP HANA you now have the option to defined custom transformation rules, which are processed completely on the SAP HANA database effectively eliminating the need to move data over the network between database and the application server. Not only does this option provide maximum performance, but also allows to leverage many of the powerful capabilities of the SAP HANA platform.

2. **Background Information**

As of SAP BW 7.4 Support Package 5 when running on the SAP HANA platform, all transformations will be processed in the SAP HANA database as far as possible. This enables fast processing of transformations in many cases already, when using standard rule types including formulas and look-ups in DataStore Objects or master data. However, any transformations containing custom routines (i.e. ABAP coding) is still executed on the SAP application server.

With the availability of 7.4 Support Package 8, you have the option to implement customer-specific transformations directly in the SAP HANA database. Similarly to expert routines that are coded in ABAP, you can define a BW transformation as a so called SAP HANA Expert Script. This script is implemented using an ABAP Managed Database Procedure (AMDP) in the SAP HANA SQLScript language.

**Tip**

In a similar manner, data can be processed in SAP HANA using so called HANA Analysis Processes (HAP). If you want to know more about HAPs, please take a look at the “SAP First Guidance SAP BW 7.40 on HANA – HANA Analysis Processes” document.
2.1 Considerations

Generally speaking, executing transformation in SAP HANA is a performance feature. It, therefore, does not make sense to change all custom ABAP routines in your system to SAP HANA Expert Script. We rather recommend carefully evaluating several aspects of the transformations to decide if a change to SAP HANA Expert Script should be pursued.

2.1.1 Criticality

Good candidates are transformations that are business or time critical. Transformations that do not meet service level agreements (SLAs) or are close to exceeding these SLAs should be looked at first. In addition to the total runtime of the transformation, take also the frequency of data loads into consideration. For example, it might be worth saving a few minutes if a load is processed every hour, but not for a nightly batch job.

2.1.2 Feasibility

Determine the root cause of long runtimes of your transformation. Good candidates are transformations, where most time is spent during ABAP processing or during transfer of the data from database to the application server. If most time is consumed during database processing already, there will be little to no benefit of converting to SAP HANA Expert Script. If the database selects are inefficient in the first place, the tuning the ABAP SELECT statements might be an option.

2.1.3 Complexity

Evaluate the complexity of the custom transformation logic. Some ABAP routines are incredibly complex and would be time consuming to implement as a SAP HANA Expert Script. Also ABAP and the ABAP-based transformations offer advanced and easy-to-use debugging methods to verify the results of the own code during the implementation/ development phase (also see Appendix).

2.1.4 Skills

Ask yourself, if the business logic in SAP HANA SQLScript can be maintained in a sustainable way. Sufficient knowledge and experience in SQLScript will be required not only for the initial implementation but also for future changes. For example, you do not want to be stranded with SQLScript you cannot maintain after the consulting partner has left.

2.2 Restrictions

Transformations in SAP HANA are not possible in all cases. You can check existing transformations for the possibility to be pushed down in the transformation maintenance screen. If the transformation cannot be pushed down, you get the list of unsupported features within the transformation.

Please take the restrictions listed in this chapter into careful consideration when modeling data flows in SAP BW (the list is based on SAP BW 7.4 Support Package 12 and 7.5 Support Package 1).

Note

The logic of the transformation is reflected in a so-called SAP HANA Calculation Scenario on top of the source DataProvider, which is generated at activation time of the transformation. At runtime of the transformation an “INSERT AS SELECT” statement is executed on this Calculation Scenario, and the INSERT writes the data directly into the tables of the target DataProvider.
2.2.1 Data Targets
The following objects are supported as targets:
- DataStore Objects (classic: standard and write-optimized)
- DataStore Objects (advanced)
- Semantically-Partitioned Objects based on DataStore Objects
- Open hub destinations with database tables (without database connection) or third-party tools

2.2.2 Data Validation
Data which is transformed using a SAP HANA Expert Script is written directly into the data target. The system will not perform any data validation. The output of the transformation must be provided in tabular format using a predefined order of columns and values must adhere to the given data types.

⚠️ CAUTION
The output data must match not just the data type used in SAP HANA but also the BW-specific definition for each field.

Here is a list of possible pitfalls:
- NULL values are not allowed in any case
- Initial values for InfoObjects depend on the data type
  For example, initial NUMC fields must be filled with zeroes only, although the database field allows for all characters
- Allowed characters for InfoObjects of type CHAR are defined in BW customizing (transaction RSKC)
- Lowercase letters are only allowed if the corresponding setting is turned on for the InfoObject
- Fields must be filled using the BW system internal value representation (especially if conversion exits are in place)
  For example, fiscal periods are stored in YYYYMMM format but externally shown as MMM/YYYY to the user
- Values are not checked against BW master data tables
- If a constant is assigned to an InfoObject, the exact (internal) value must be provided for the corresponding field

2.2.3 Error Handling
In the current version, one significant restriction is the lack of error handling capabilities. It is not possible to raise errors or pass error messages back from SAP HANA to the BW data load framework. This means, for example, that the transformation has to complete successfully for all records and you can't mark erroneous records and send them to a PSA for later processing.

2.2.4 Other Restrictions
The following objects are not supported for executing transformations in SAP HANA:
- Queries as InfoProviders are not supported as the source
- ABAP routines are not supported
- Rule groups are not supported
- To read data from DataStore objects, the entire key must be provided
- Near-line connections are not supported
2.3 Recommendations

2.3.1 BW Best Practices
It is strongly recommended to use standard transformation capabilities as much as possible. Many transformations that customers have coded manually in the past could be accomplished by standard transformation rules today. Especially if you are not familiar with the most recent BW release, we encourage you to review the standard transformation rule types and the transformation and formula libraries.

It is possible in SAP BW to chain several transformations between source and target together using InfoSources. When running on the SAP HANA database, we recommend in such case using a maximum of two transformations:

The first transformation should include all fields that can be processed using standard transformation rules (including formulas). This will ensure that this transformation can be processed completely in SAP HANA.

The second transformation should then include all custom logic based on a SAP HANA Expert Script, which of course is processed in SAP HANA as well.

2.3.2 SQLScript Best Practices
To obtain maximum performance, it is highly recommended to use set-based processing in SAP HANA. This means using SQL statements that work on the complete data request rather than loops (or database cursors) which process records individually.

For more details, see “Best Practices for Using SQLScript” and “SAP HANA Troubleshooting and Performance Analysis Guide”.

3. Prerequisites

This section lists the prerequisites that must be in place before you can use SAP HANA SQLScript in BW transformations.

Minimum Releases and Support Packages:
- SAP HANA Release 1.0 Support Package Stack 8 or higher
- SAP BW Release 7.4 Support Package 8 or higher

Development Tools:
- SAP HANA Studio (recommended: version 2.1 or higher)
- SAP BW Modeling Tools for Eclipse (recommended: version 1.12 or higher)
- ABAP Development Tools for Eclipse (recommended: version 2.54 or higher)

System Configuration:
- SAP BW Data Transfer Process settings
- ABAP Managed Database Procedure implementation
- System privileges to allow usage of ABAP Managed Database Procedures
- User ID which is registered as developer and has proper authorizations

3.1 SAP HANA

Push down of transformation logic to the database is supported only for SAP BW systems running on the SAP HANA platform. The minimum version for leveraging this capability is SAP HANA 1.0 Support Package Stack 8.

Tip
You can leverage many powerful SAP HANA capabilities like predictive functions or smart data access. Please see the SAP HANA Server Installation and Update Guide for details on how to install these features if necessary.

Resources:
Online Documentation for SAP HANA
SAP HANA Academy on YouTube
Installing or Updating SAP HANA Components
3.2 SAP HANA Studio
SAP HANA Studio enables technical users to manage the SAP HANA database, to create and manage user authorizations, and to create new or modify existing models of data in the SAP HANA database. It is a client tool, which can be used to access local or remote SAP HANA databases.

Download:
SAP Development Tools for Eclipse / SAP HANA Studio

Resources:
SAP HANA Studio Installation and Update Guide

3.3 SAP HANA SQLScript
The purpose of SAP HANA SQLScript is to embed data-intensive application logic into the database. In order to process the full transformation logic in SAP HANA, you will need to know how to program in SAP HANA SQLScript. As a prerequisite the SAP HANA Script Server must be activated and running.

Resources:
Online Documentation for SAP HANA SQL and SQLScript References
SAP HANA Developer Center / Community
SAP Note 1650957 - SAP HANA Database: Starting the Script Server

3.4 SAP BW Modeling Tools
The recommended way to maintain objects in SAP BW powered by SAP HANA is to use the so called SAP BW Modeling Tools for Eclipse (BW-MT). Using the Modeling Tool provides integration with the ABAP Development Tools (ADT) and you can for example jump directly from a BW transformation to the maintenance of the associated ABAP Managed Database Procedure.

Download:
SAP Development Tools for Eclipse / Modeling Tools for SAP BW powered by SAP HANA

Resources:
SAP BW Modeling Tools Install guide
SAP First Guidance - Implementing BW-MT
SAP Note 1954169 - BWMT - Installation Details
SAP Note 2094475 - Report to implement BW-MT notes
3.5 SAP BW Data Transfer Process

If you have a data flow in which one transformation contains an ABAP routine and another transformation contains an ABAP Managed Database Procedures, you cannot define a data transfer process for this data flow. In this case, the transformations have to be changed in such a way that either the ABAP routine or the ABAP Managed Database Procedures can be used, but not both.

Resources:
Processing the Data Transfer Process in SAP HANA

3.6 ABAP Development Tools

ABAP Development Tools for SAP NetWeaver (ADT) are required to code the BW transformation logic that you want to push down to the SAP HANA database. The code will be encapsulated in a so called ABAP Managed Database Procedure (AMDP).

Note
AMDP classes can only be edited with the ABAP Development Tools (ADT).

Download:
SAP Development Tools for Eclipse / ABAP Development Tools for SAP NetWeaver

Resources:
Online Documentation for ABAP Development Tools
ABAP in Eclipse Community

3.7 ABAP Managed Database Procedures

ABAP Managed Database Procedures provides a framework for managing and calling stored procedures. ABAP Managed Database Procedure (or AMDP procedure) is a procedure written in a database-specific language (for this guide SAP HANA SQLScript) implemented in an AMDP method of an AMDP class. BW will generate one AMDP class per transformation that you want to process in the SAP HANA database.

Resources:
Online Documentation for ABAP Managed Database Procedures
ABAP Development Community
Self-paced openSAP course for ABAP Development for SAP HANA
3.8 System Security

Before your BW system is able to manage SQL Script procedures on the SAP HANA database, the SAP <SID> system user requires the following authorizations as a user of the database system:

- Privilege Execute on the object GET_PROCEDURE_OBJECTS of the schema SYS
- Privilege Execute on the object TRUNCATE_PROCEDURE_OBJECTS of the schema SYS

Missing authorizations can be detected using transaction SICK.

Resources:
SAP Note 1899222 - ABAP Managed Database Procedures: Technical prerequisites

3.9 User Security

Your user must be registered in the BW system as a developer. The registration needs to be performed by your system administrator on the SAP Support Portal. In addition, the user must have authorizations to create and maintain ABAP objects (authorization object S_DEVELOP).

Resources:
Developer Registration
4. Scenario

We will use the NetWeaver Enterprise Procurement Model (EPM), which is an application intended to be used for demonstration and testing purposes, as a basis for the examples given in this guide.

For more information regarding the Enterprise Procurement Model please check the following document:

Enterprise Procurement Model (EPM)

In particular, we will use the “Completed Sales Orders (NW Demo EPM)” InfoCube `0EPM_C04` as a data source. You can generate data for this InfoCube using a tool (see linked documentation).
5. Step-by-Step Procedure

In this section, we show how to implement transformations using ABAP Managed Database Procedures (AMDP). As a simple example, we calculate the average order amount per product category and calendar month. We will use “InfoArea for Open Item Analytics based on EPM Sales Model” (0EPM_OIA_IA), which you can find in the BW Repository under the “Technical Content” (0BWTC).

5.1 Create Key Figure

We want to use a new key figure for storing the average order amount.

Create key figure ZBW_AVERG with the following settings: Type = Amount, Data Type = CURR, Unit InfoObject = 0D_NW_PCURR:

Save and activate the key figure.
5.2 Create Data Target

As a target we use an Advanced DataStore Object.

Create the Advanced DataStore Object `ZBW_AVG` with the following settings:

And add the following InfoObjects:

Save and activate the DSO.
5.3 Create Transformation

As the next step, we create a transformation between the InfoCube 0EPM_C04 and the new DataStore Object.

**Note**

In order to create a transformation, you have to open a SAP GUI connection (SAP Logon icon in toolbar or “Open SAP GUI...” in the Navigation Menu).

Start the BW Data Warehousing Workbench (transaction RSA1) and find your DataStore Object ZBW_AVG (in the InfoProvider tree or under favorites). The select “Create Transformation...” from the context menu:

Select InfoCube 0EPM_C04 as a source and DataStore Object ZBW_AVG as a target:
We do not want to use the suggested default field mappings but use an AMDP instead. Select *Edit* → *Routines* → *SAP HANA Expert Script Create* from the menu:

Confirm the subsequent dialog box in order to create the procedure.

**Note**

If this is the first time you use the ABAP Development Tools in Eclipse, the system will prompt you to create a new ABAP Project. Just use the same connection as to your BW System.

The system generates an AMDP class and method as a shell for your custom transformation.

The method is using a standardized signature which simply contains an input and an output table corresponding to the definition of your data source (in our case the *0EPM_C04* InfoCube) and the data target (in our case the *ZBW_AVG* DataStore Object).
The order of fields depends on the type of InfoProvider and is predefined by the system. In addition to the fields of the data source, the input table contains a RECORD field for the row number. When using DataStore Objects as a target, the output table contains an additional RECORDMODE field (CHAR 1) which defines what happens to each record during activation of the data load request (see Appendix).

You will find the generated AMDP class under your system, favorites, and the $TMP package. You can also see that the method needs to be implemented in SAP HANA and will be “read-only” meaning no data shall be changed inside the procedure.

**Note**

If you want to access other ABAP dictionary tables or views or other AMDP procedures, you will have to mention the tables in the USING clause of the method (see AMDP documentation). It is not allowed to modify any other part of the method signature.

You can see that several types have been created by default to match data source and target (these are the same as for Expert Routines). For easy reference the names of the InfoObjects are included as comments. You might notice an error message. This is shown only initially since no code has been maintained yet to fill the output table.

### 5.4 Maintain Transformation Logic

You are ready to enter the SAP HANA SQLScript code defining your custom transformation logic. Scroll to the bottom of the class implementation to find method PROCEDURE. All you have to do is fill the output table according to your business requirements. In our simple example, a single SQL statement is sufficient:

```sql
outTab =
    select calyear, calmonth, d_nw_prid__d_nw_prcat AS d_nw_prcat, d_nw_pcurr, ' ' AS recordmode, AVG (d_nw_gamt) AS '/BIC/ZBW_AVERG', ' ' as record
    FROM :inTab
GROUP BY calyear, calmonth, d_nw_prid__d_nw_prcat, d_nw_pcurr;
```

Save and activate the AMDP class and method.

**Important**

The order and data type of the fields has to match the type definition of the output table exactly. If data type conversions are required, then you have to code these in the select clause of the statement.
5.5 Activate BW Transformation

After you have maintained the code for the AMDP procedure, go back to the BW transformation and activate it.

5.6 Testing

Now the BW transformation is complete and ready to be executed. Go back to the maintenance of the BW transformation and create a Data Transfer Process to test it.
Save and activate the DTP. The “SAP HANA Execution” checkbox should be turned on indicating that the complete transformation will be executed in SAP HANA and no data will be transferred to the BW application server during processing.
6. Appendix

6.1 Further Information
You can find more details about transformations in SAP HANA in the SAP Documentation. Also see SAP Note 2057542 - Recommendation: Usage of HANA-based Transformations.

6.2 Troubleshooting
In case of errors for example during check or activation of the transformation, please review SAP Notes 2067912 and 2117312, which list important corrections related to SAP HANA transformations and analysis processes for 7.4 Support Package 8 or higher.
In the system runs into out-of-memory situations, please review SAP Note 2230080, “DTP: Out of memory situation during ‘SAP HANA Execution' and the 'Request by Request' Extraction”.

6.3 Debugging
In order to debug the SAP HANA Expert Scripts, you can enhance the AMDP class with debug procedures. This expert option is enabled by setting parameter DTP_HANA_DEBUG_SCRIPT in table RSADMIN to “X” (requires SAP Note 2159872 or 7.4 Support Package 12 or higher)
As of SAP HANA 1.0 Support Package Stack 10, you can set break-points directly in the AMDP class. These break-points will then behave identical to external break-points in regular ABAP code.

6.4 Modularization
If complex custom logic needs to be developed, then it is recommended to modularize the script code (just like in other programming languages). While it is technically possible to call other SAP HANA database procedures from within the AMDP used by BW, we recommend not using this technique. Instead, one should code the logic using the AMDP framework as well i.e. develop other “modules” using your own AMDP class and methods and call them from the AMDP used by BW. This has the benefit that all code will be handled by the SAP Transport Management System.

Important
It is not possible to add methods to the generated AMDP class since these would be deleted when the class is generated again. However, you can use calls to your own AMDP class implementation (and its methods).

You can find an example of how to call one AMDP from another AMDP in the online documentation.

6.5 Transporting
When adding a BW transformation, which contains a SAP HANA Expert Script, to a transport request, the BW system will include the code and signature of the corresponding PROCEDURE method of the ABAP Manage Database Procedure.
In the target system, BW will then generate the AMDP class based on the transported code and signature (the technical name of the AMDP class /BIC/<GUID> remains the same in all systems).
6.6 Leveraging SAP HANA Options

SAP HANA Options provide additional features to the core edition of the SAP HANA platform. For more details, please see the online documentation.

**Note**
To use the SAP HANA options in a production system, you must purchase the appropriate software license from SAP.

Here are some ideas on how you could use these features in a BW context.

- **SAP HANA Advanced Data Processing (Text Analysis):**
  Use text analysis to discover and classify entities in your unstructured data.

- **SAP HANA Spatial:**
  Analyze geospatial data. For example, determine the distance between two locations, and persist the results in BW.

6.7 BW Delta Process: Record Mode

The `0RECORDMODE` field describes how a record is updated in the delta process when loading DataStore Objects. The various delta processes support different combinations of the seven possible characteristic values.

**SPACE:** The record delivers an after image.

The status is transferred after something is changed or added. You can update the record into an InfoCube only if the corresponding before image exists in the request.

**'X': The record delivers a before image**

The status is transferred before data is changed or deleted.

All record attributes that can be aggregated have to be transferred with a reverse +/- sign. The reversal of the sign is carried out either by the extractor (default) or the Service API. In this case, the indicator field is inverted in case of cancellation must be set in the DataSource for the relevant extraction structure field.

These records are ignored if the update is a non-additive update of a DataStore object.

The before image is complementary to the after image.

**'A': The record delivers an additive image.**

For attributes that can be aggregated, only the change is transferred. For attributes that cannot be aggregated, the status after a record has been changed or created is transferred. This record can replace an after image and a before image if there are no non-aggregation attributes or if these cannot be changed. You can update the record into an InfoCube without restriction, but this requires an additive update into a DataStore object.
'D': The record has to be deleted.
Only the key is transferred. This record (and its DataSource) can only be updated into a DataStore object.

'R': The record delivers a reverse image.
The content of this record is the same as the content of a before image. The only difference is with a DataStore object update: Existing records with the same key are deleted.

'N': The record delivers a new image.
The content of this record is the same as for an after image without a before image. When a record is created, a new image is transferred instead of an after image.
The new image is complementary to the reverse image.

'Y': The record is an update image.
This kind of record is used in the change log of a Data Store object in order to save the value from the update. This is for a possible rollback and roll-forward for key figures with minimum or maximum aggregation. This record also has the update value for characteristics (in this case, it is the same as the after image).
Null values are stored for key figures with totals aggregation. An update image is only required when the value from the update is smaller or larger than the before image for at least one key figure with minimum or maximum aggregation.

See also transaction SE11 → Data Element RODMUPDMOD → Go to → Documentation → Display.