

SAP First Guidance
SAP BW and SAP BW/4HANA

SAP HANA™

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

Applicable Releases:

SAP BW 7.4 or 7.5, powered by SAP HANA
SAP BW/4HANA 1.0

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 and SAP BW/4HANA.

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 and SAP BW/4HANA.

Version 1.4

April 2017



The Best-Run Businesses Run SAP™

© Copyright 2017 SAP SE. All rights reserved.

No part of this publication may be reproduced or transmitted in any form or for any purpose without the express permission of SAP SE. The information contained herein may be changed without prior notice.

Some software products marketed by SAP SE and its distributors contain proprietary software components of other software vendors.

Microsoft, Windows, Excel, Outlook, and PowerPoint are registered trademarks of Microsoft Corporation.

IBM, DB2, DB2 Universal Database, System i, System i5, System p, System p5, System x, System z, System z10, System z9, z10, z9, iSeries, pSeries, xSeries, zSeries, eServer, z/VM, z/OS, i5/OS, S/390, OS/390, OS/400, AS/400, S/390 Parallel Enterprise Server, PowerVM, Power Architecture, POWER6+, POWER6, POWER5+, POWER5, POWER, OpenPower, PowerPC, BatchPipes, BladeCenter, System Storage, GPFS, HACMP, RETAIN, DB2 Connect, RACF, Redbooks, OS/2, Parallel Sysplex, MVS/ESA, AIX, Intelligent Miner, WebSphere, Netfinity, Tivoli and Informix are trademarks or registered trademarks of IBM Corporation.

Linux is the registered trademark of Linus Torvalds in the U.S. and other countries.

Adobe, the Adobe logo, Acrobat, PostScript, and Reader are either trademarks or registered trademarks of Adobe Systems Incorporated in the United States and/or other countries.

Oracle is a registered trademark of Oracle Corporation.

UNIX, X/Open, OSF/1, and Motif are registered trademarks of the Open Group.

Citrix, ICA, Program Neighborhood, MetaFrame, WinFrame, VideoFrame, and MultiWin are trademarks or registered trademarks of Citrix Systems, Inc.

HTML, XML, XHTML and W3C are trademarks or registered trademarks of W3C®, World Wide Web Consortium, Massachusetts Institute of Technology.

Java is a registered trademark of Sun Microsystems, Inc.

JavaScript is a registered trademark of Sun Microsystems, Inc., used under license for technology invented and implemented by Netscape.

SAP, R/3, SAP NetWeaver, Duet, PartnerEdge, ByDesign, SAP BusinessObjects Explorer, StreamWork, and other SAP products and services mentioned herein as well as their respective logos are trademarks or registered trademarks of SAP SE in Germany and other countries.

Business Objects and the Business Objects logo, BusinessObjects, Crystal Reports, Crystal Decisions, Web Intelligence, Xcelsius, and other Business Objects products and services mentioned herein as well as their respective logos are trademarks or registered trademarks of Business Objects Software Ltd. Business Objects is an SAP company.

Sybase and Adaptive Server, iAnywhere, Sybase 365, SQL Anywhere, and other Sybase products and services mentioned herein as well as their respective logos are trademarks or registered trademarks of Sybase, Inc.

Sybase is an SAP company.

All other product and service names mentioned are the trademarks of their respective companies. Data contained in this document serves informational purposes only. National product specifications may vary.

The information in this document is proprietary to SAP. No part of this document may be reproduced, copied, or transmitted in any form or for any purpose without the express prior written permission of SAP SE.

This document is a preliminary version and not subject to your license agreement or any other agreement with SAP. This document contains only intended strategies, developments, and functionalities of the SAP® product and is not intended to be binding upon SAP to any particular course of business, product strategy, and/or development. Please note that this document is subject to change and may be changed by SAP at any time without notice.

SAP assumes no responsibility for errors or omissions in this document. SAP does not warrant the accuracy or completeness of the information, text, graphics, links, or other items contained within this material. This document is provided without a warranty of any kind, either express or implied, including but not limited to the implied warranties of merchantability, fitness for a particular purpose, or non-infringement.

SAP shall have no liability for damages of any kind including without limitation direct, special, indirect, or consequential damages that may result from the use of these materials. This limitation shall not apply in cases of intent or gross negligence.

The statutory liability for personal injury and defective products is not affected. SAP has no control over the information that you may access through the use of hot links contained in these materials and does not endorse your use of third-party Web pages nor provide any warranty whatsoever relating to third-party Web pages.

SAP "How-to" Guides are intended to simplify the product implementation. While specific product features and procedures typically are explained in a practical business context, it is not implied that those features and procedures are the only approach in solving a specific business problem using SAP NetWeaver. Should you wish to receive additional information, clarification or support, please refer to SAP Consulting.

Any software coding and/or code lines / strings ("Code") included in this documentation are only examples and are not intended to be used in a productive system environment. The Code is only intended better explain and visualize the syntax and phrasing rules of certain coding. SAP does not warrant the correctness and completeness of the Code given herein, and SAP shall not be liable for errors or damages caused by the usage of the Code, except if such damages were caused by SAP intentionally or grossly negligent.

Disclaimer

Some components of this product are based on Java™. Any code change in these components may cause unpredictable and severe malfunctions and is therefore expressly prohibited, as is any decompilation of these components.

Any Java™ Source Code delivered with this product is only to be used by SAP's Support Services and may not be modified or altered in any way.



The Best-Run Businesses Run SAP™

Document History

Document Version	Description
1.0	First official release of this guide
1.1	Support for InfoObject and InfoCube as target object types (BW 7.5)
1.2	Support for start, end, and field routines and updated screen shots (BW 7.5 SP 4).
1.3	Changes to AMDP signature (BW 7.5 SP 4) Added reference to blog
1.4	Added note about using AMDP outside of transformations (chapter 2.3.1) Clarification on technical AMDP names and transporting (chapter 6.5)



The Best-Run Businesses Run SAP™

Typographic Conventions

Type Style	Description
<i>Example Text</i>	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
Example text	Emphasized words or phrases in body text, graphic titles, and table titles
Example text	File and directory names and their paths, messages, names of variables and parameters, source text, and names of installation, upgrade and database tools.
Example text	User entry texts. These are words or characters that you enter in the system exactly as they appear in the documentation.
<Example text>	Variable user entry. Angle brackets indicate that you replace these words and characters with appropriate entries to make entries in the system.
EXAMPLE TEXT	Keys on the keyboard, for example, F2 or ENTER.

Icons

Icon	Description
	Caution
	Note or Important
	Example
	Recommendation or Tip



The Best-Run Businesses Run SAP™

Table of Contents

1.	Business Scenario.....	1
2.	Background Information	1
2.1	Considerations	2
2.1.1	Criticality	2
2.1.2	Feasibility	2
2.1.3	Complexity	2
2.1.4	Skills	2
2.2	Restrictions.....	2
2.2.1	Data Targets	3
2.2.2	Data Validation	3
2.2.3	Error Handling	4
2.2.4	Other Restrictions	4
2.3	Recommendations.....	4
2.3.1	BW Best Practices	4
2.3.2	SQLScript Best Practices.....	4
3.	Prerequisites.....	6
3.1	SAP HANA.....	6
3.2	SAP HANA Studio.....	7
3.3	SAP HANA SQLScript.....	7
3.4	SAP BW Modeling Tools.....	7
3.5	SAP BW Data Transfer Process	8
3.6	ABAP Development Tools.....	8
3.7	ABAP Managed Database Procedures	8
3.8	System Security.....	9
3.9	User Security	9
4.	Scenario.....	10
5.	Step-by-Step Procedure.....	11
5.1	Create Key Figure	11
5.2	Create Data Target	12
5.3	Create Transformation	13
5.4	Maintain Transformation Logic.....	15
5.5	Activate BW Transformation	16
5.6	Testing	17



The Best-Run Businesses Run SAP™

6.	Appendix.....	19
6.1	Further Information.....	19
6.2	Troubleshooting.....	19
6.3	Debugging	19
6.4	Modularization	19
6.5	Transporting.....	20
6.6	Leveraging SAP HANA Options.....	20
6.7	BW Delta Process: Record Mode.....	20
6.8	SAP HANA Execution of Transformation.....	22
6.9	SAP BW/4HANA.....	22



The Best-Run Businesses Run SAP™

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 define 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 routines that are coded in ABAP, you can define a BW transformation as a so called **AMDP 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 look at the "[SAP First Guidance SAP BW 7.40 on HANA – HANA Analysis Processes](#)" document.

For more details, please refer to the detailed blog about "SAP HANA-based BW Transformations" at <http://scn.sap.com/community/bw-hana/blog/2016/05/24/hana-based-bw-transformation> (highly recommended).

2.1 Considerations

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 ABAP-managed Database Procedures. We rather recommend carefully evaluating several aspects of the transformations to decide if a change to AMDP 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 AMDP 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 AMDP 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 (non-cumulative data targets still require some ABAP processing).

2.2.1 Data Targets

The following objects are supported as targets:

- DataStore Objects (classic): standard and write-optimized types
- DataStore Objects (advanced): all types but not with “InfoCube” settings¹
- Semantically-Partitioned Objects based on DataStore Objects
- Open hub destinations with database tables (without database connection) or third-party tools

As of SAP BW 7.5 also the following targets are supported:

- DataStore Objects (advanced): all types including with “InfoCube” settings
- InfoObjects
- InfoCubes

For the last three targets, extraction and transformation are performed using SAP HANA and SID handling is executed using BW ABAP. Therefore, using AMDP Script does not improve performance for simple transformations using for example 1:1 assignments.

2.2.2 Data Validation

Data which is transformed using an AMDP 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 pit falls:

- 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 YYYYMM 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

¹ Properties “Activate Data” and “All Characteristics are Key, Reporting on Union of Inbound and Active Table” are set

2.2.3 Error Handling

As of SAP BW 7.5 SP 4, error handling can be enabled using the corresponding flag in the Data Transfer Process (DTP). Setting the flag will change the signature of the ABAP managed database procedure and allow passing error messages back to SAP BW (see blog for details).

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
- Customer-defined functions in the formula editor are not supported
- To read data from DataStore objects, the entire key must be provided
- Near-line storage (NLS) 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](#) (see Appendix “SAP HANA Execution of Transformations”).

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 AMDP Script, which of course is processed in SAP HANA as well.

Note

SAP retains the right to change the naming and signature of the AMDP class at any time. Therefore, we strongly recommend **not** to use the AMDP class **outside** of the transformation that it belongs to.

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 (recommended: SPS 12 or higher)
- SAP BW Release 7.4 Support Package 8 or higher (recommended: 7.5 SP 4 or higher)

Development Tools:

- SAP HANA Studio (recommended: version 2.3 or higher)
- SAP BW Modeling Tools for Eclipse (recommended: version 1.14 or higher)
- ABAP Development Tools for Eclipse (recommended: version 2.60 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. 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 must 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 can 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 \(EPM\)](#)

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).

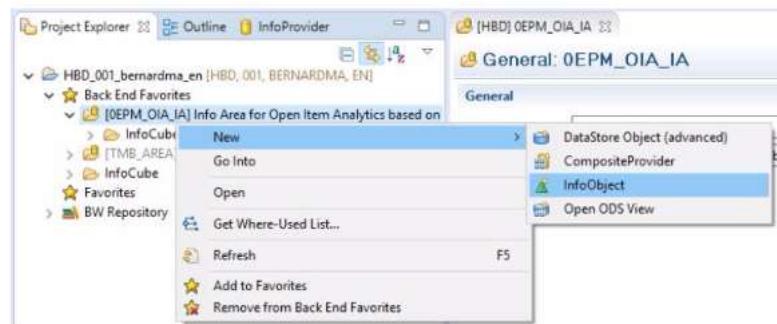
InfoProvider Data Model	Techn.name
Completed Sales Orders (NW Demo EPM)	0EPM_C04
Sales Order	0EPM_C041
Sales Order (EPM Demo)	0D_NW_SO
Sales Order Item	0EPM_C042
Sales Order Item (EPM Demo)	0D_NW_SOI
Business Partner	0EPM_C043
Business Partner (EPM Demo)	0D_NW_BP
Business Partner Role	0EPM_C044
Role (EPM Demo)	0D_NW_ROLE
Product	0EPM_C045
Product ID (EPM Demo)	0D_NW_PRID
Status	0EPM_C046
Ordering Status (EPM Demo)	0D_NW_POOS
Order Lifecycle Status (EPM Demo)	0D_NW_POLS
Order Confirmation Status (EPM Demo)	0D_NW_POCs
Order Approval Status (EPM Demo)	0D_NW_POAS
Planned Delivery Date	0EPM_C047
Planned Delivery Date (EPM Demo)	0D_NW_DELD
Created By	0EPM_C048
Created by (EPM Demo)	0D_NW_CRB
Changed by	0EPM_C049
Changed by (EPM Demo)	0D_NW_CHB
Changed on	0EPM_C04A
Changed on (EPM Demo)	0D_NW_CHAT
Created On	0EPM_C04B
Created on (EPM Demo)	0D_NW_CA
Time	0EPM_C04T
Calendar year	0CALYEAR
Calendar Year/Month	0CALMONTH
Calendar day	0CALDAY
Data Package	0EPM_C04P
Change Run ID	0CHNGID
Record type	0RECORDTP
Request ID	0REQUID
Unit	0EPM_C04U
Order Currency	0D_NW_PCURR
Order Unit	0D_NW_PUNI
Key Figures	1KYFNM
Taxes (Order Amount) (EPM Demo)	0D_NW_TAMT
Order Quantity (EPM Demo)	0D_NW_PQTY
Net Order Amount (EPM Demo)	0D_NW_NAMT
Gross Order Amount (EPM Demo)	0D_NW_GAMT

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” (*0BWTCT*). The example uses an Expert Routine but you could also implement this as a field routine as well.

5.1 Create Key Figure

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

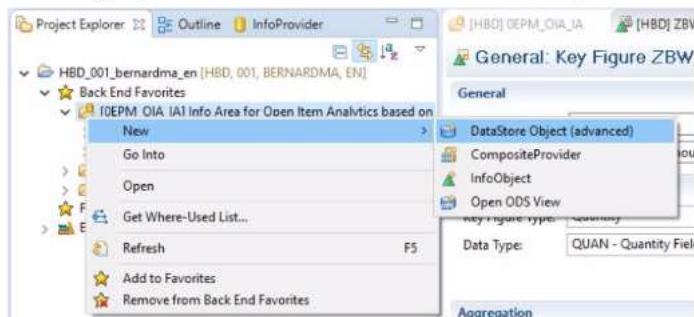


Create key figure *ZBW_AVERG* with the following settings: Type = Amount, Data Type = CURR, Unit InfoObject = *0D_NW_PCURR*:

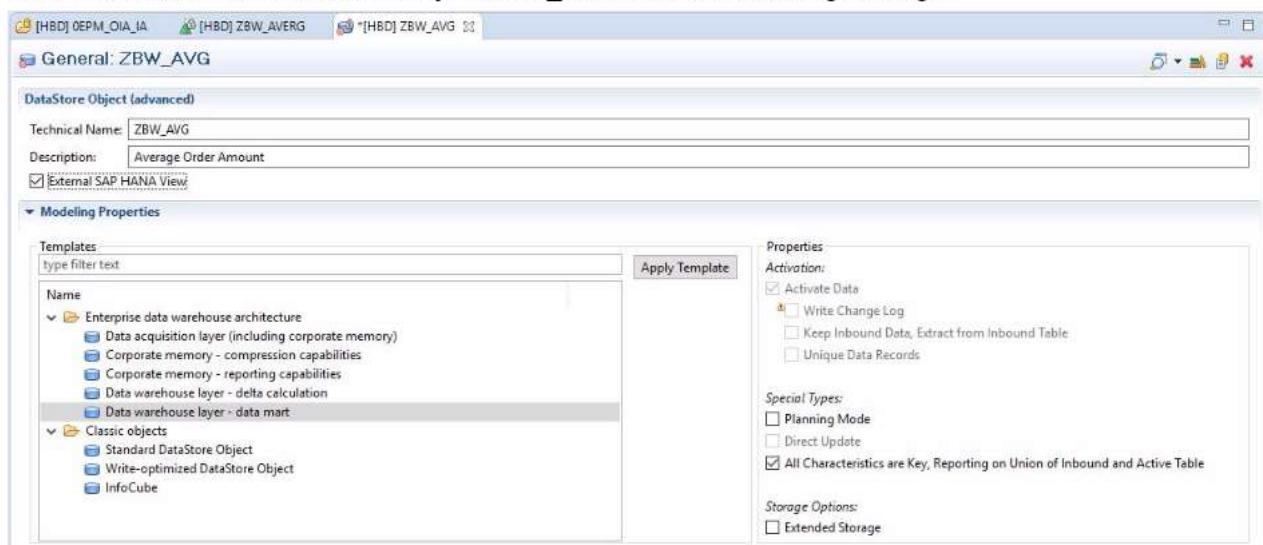
Save and activate the key figure.

5.2 Create Data Target

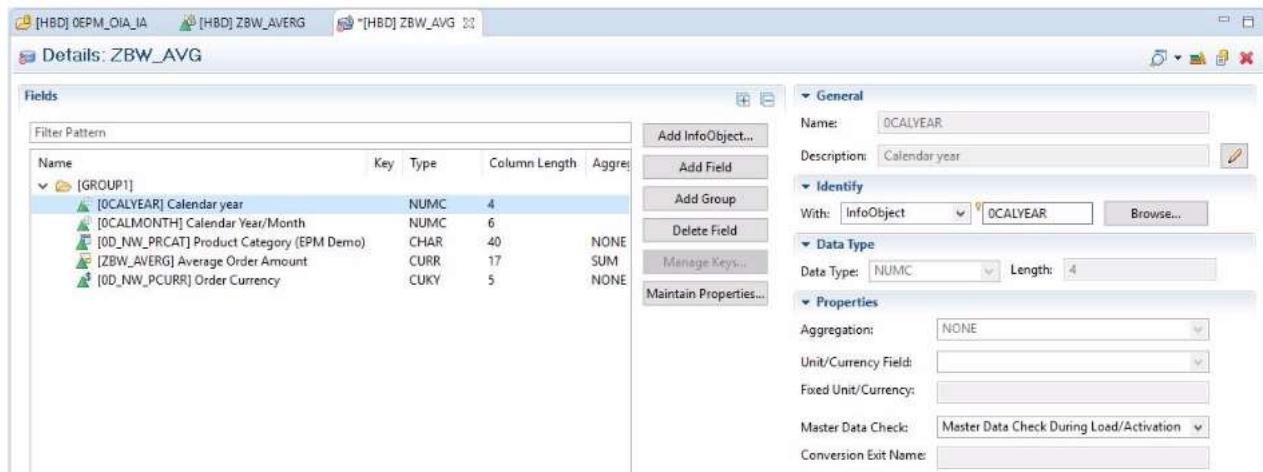
As a target, we use an Advanced DataStore Object.



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



And add the following InfoObjects:



Save and activate the DSO.

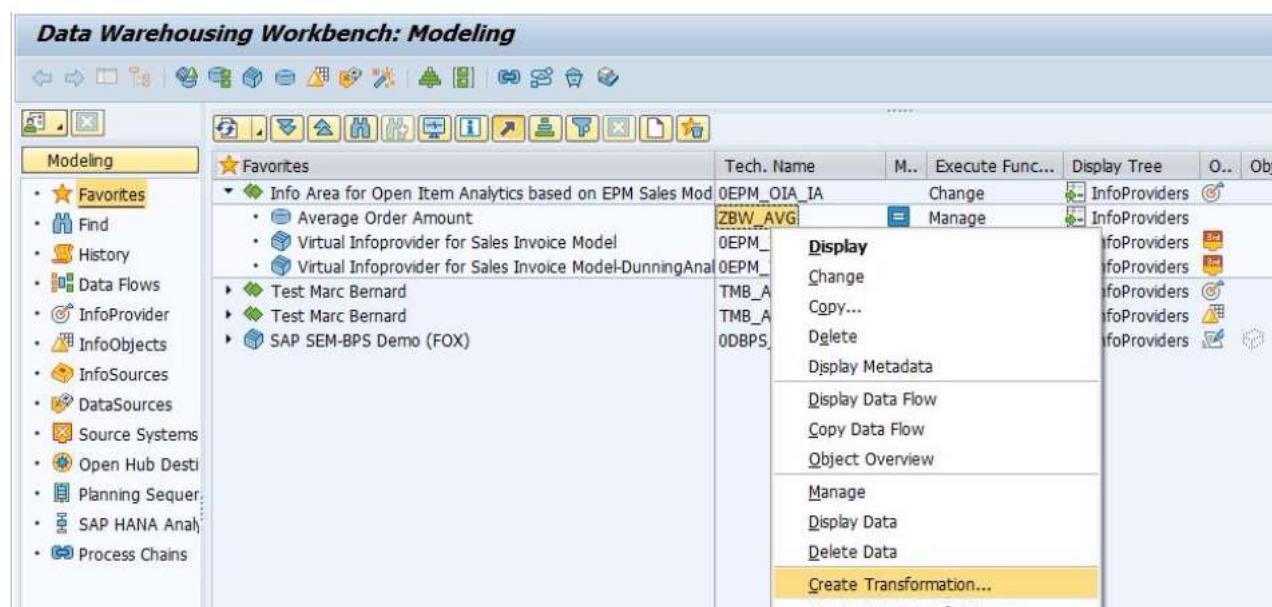
5.3 Create Transformation

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

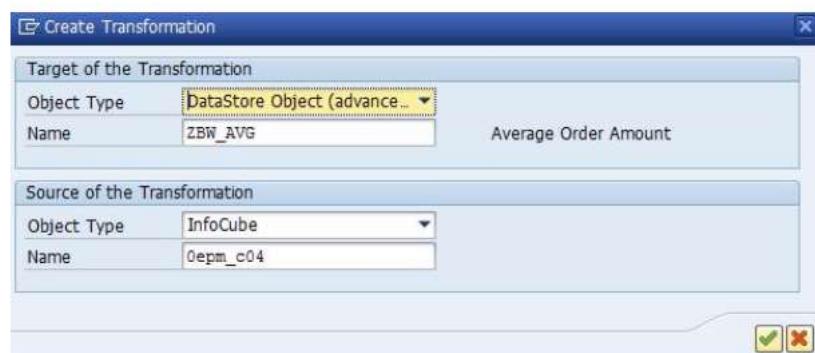
Note

To create a transformation, you must open a SAP GUI connection (SAP Logon icon in toolbar or “Open SAP GUI...” in the Navigation Menu).

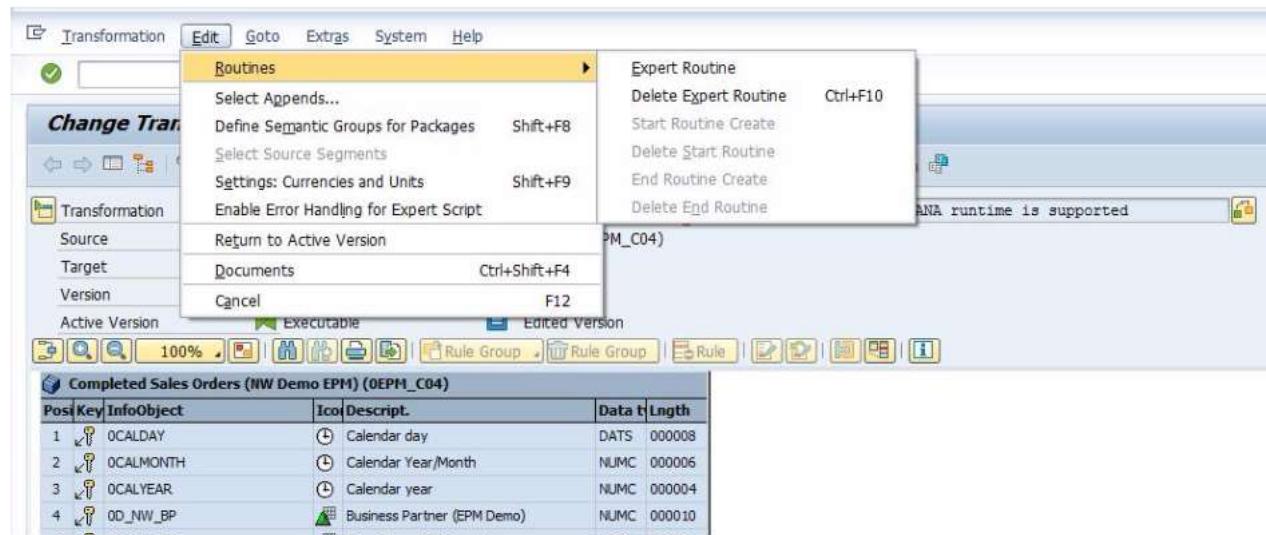
Start the BW Data Warehousing Workbench (transaction `RSA1`) and find your DataStore Object `ZBW_AVG` (in the InfoProvider tree or under favorites). Then select “Create Transformation...” from the context menu:



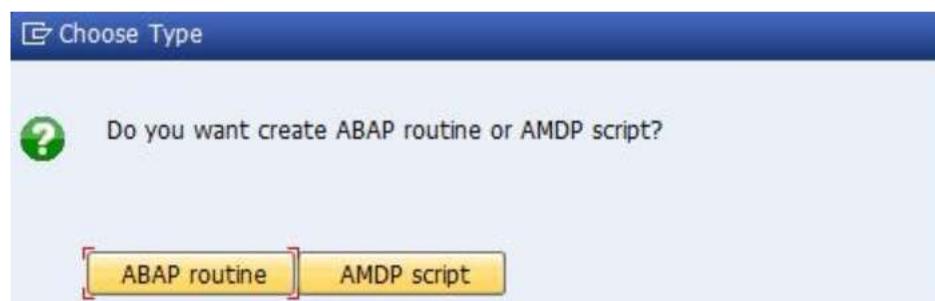
Select InfoCube `0EPM_C04` as a source and DataStore Object `ZBW_AVG` as a target:



We do not want to use the suggested default field mappings but use an AMDP instead. Select **Edit \neq Routines \neq ... Routine Create** from the menu:



Confirm the subsequent dialog box to create the procedure as either a classic ABAP routine or as an AMDP script. This dialog box shows only once since you cannot mix ABAP and AMDP routines.



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.

```

[BHD] ZBW_AVERG [HBD] ZBW_AVG [HBD] OEPM_C04 [HBD] HBD_001_BERNARDMA_EN
▶ [BIC/513Y60GVREXW2HLRTTV0O22U6 >
@ [BIC/513Y60GVREXW2HLRTTV0O22U6 definition
public
create public .

public section.

interfaces IF_AMDP_MARKER_HDB .

types:
begin of TN_S_IN1,
  CALDAY type /BIO/OICALDAY, " InfoObject: OCALDAY
  CALMONTH type /BIO/OICALMONTH, " InfoObject: OCALMONTH
  CALYEAR type /BIO/OICALYEAR, " InfoObject: OCALYEAR
  D_NW_BP type /BIO/OID_NW_BP, " InfoObject: OD_NW_BP
  D_NW_CA type /BIO/OID_NW_CA, " InfoObject: OD_NW_CA
  D_NW_CHAT type /BIO/OID_NW_CHAT, " InfoObject: OD_NW_CHAT
  D_NW_CHB type /BIO/OID_NW_CHB, " InfoObject: OD_NW_CHB
  D_NW_CRB type /BIO/OID_NW_CRB, " InfoObject: OD_NW_CRB
  D_NW_DELD type /BIO/OID_NW_DELD, " InfoObject: OD_NW_DELD
  D_NW_GAMT type /BIO/OID_NW_GAMT, " InfoObject: OD_NW_GAMT
end of TN_S_IN1.
  
```

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 *OEPM_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 must mention the tables in the *US/ING* 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 must do is fill the output table per your business requirements. In our simple example, a single SQL statement is sufficient:

```

outTab =
  select calyear, calmonth, d_nw_prid_d_nw_prcat AS d_nw_prcat, d_nw_pcurre,
  '' 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_pcurre;
  
```

Save and activate the AMDP class and method.

Important

The order and data type of the fields must match the type definition of the output table exactly. If data type conversions are required, then you must code these in the select clause of the statement.

Error handling was delivered with SAP BW 7.5 SP 4. As a result, the method declaration has changed, including the SAP HANA Expert Script. To keep existing SAP HANA Expert Scripts valid the method declaration will not change during the upgrade to SP 4, for more information see
<http://scn.sap.com/community/bw-hana/blog/2016/08/29/hana-based-bw-transformation--new-features-delivered-by-750-sp04>

5.5 Activate BW Transformation

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

The screenshot shows the SAP BW Transformation 'Change Transformation' dialog. The transformation is named 'CUBE OEPM_C04 -> ADSO ZBW_AVG'. The source is 'Completed Sales Orders (NW Demo EPM) (OEPM_C04)' and the target is 'Average Order Amount (ZBW_AVG)'. The version is 'Active' and the active version is 'Executable'. The SAP HANA Execution Possible checkbox is checked. Below the transformation details, there are two tables:

- Completed Sales Orders (NW Demo EPM) (OEPM_C04)**

Pos	Key	InfoObject	Icon	Descript.	Data t	Length
1		0CALDAY		Calendar day	DATS	000008
2		0CALMONTH		Calendar Year/Month	NUMC	000006
3		0CALYEAR		Calendar year	NUMC	000004
4		0D_NW_BP		Business Partner (EPM Demo)	NUMC	000010
5		0D_NW_CA		Created on (EPM Demo)	DATS	000008
6		0D_NW_CHAT		Changed on (EPM Demo)	DATS	000008
7		0D_NW_CHB		Changed by (EPM Demo)	NUMC	000010
8		0D_NW_CRB		Created by (EPM Demo)	NUMC	000010
9		0D_NW_DELD		Planned Delivery Date (EPM Demo)	DATS	000008
10		0D_NW_GAMT		Gross Order Amount (EPM Demo)	CURR	000017
11		0D_NW_NAMT		Net Order Amount (EPM Demo)	CURR	000017
12		0D_NW_PCURR		Order Currency	CUKY	000005
13		0D_NW_POAS		Order Approval Status (EPM Demo)	CHAR	000001
14		0D_NW_POCSS		Order Confirmation Status (EPM Demo)	CHAR	000001
15		0D_NW_POLS		Order Lifecycle Status (EPM Demo)	CHAR	000001
16		0D_NW_POOS		Ordering Status (EPM Demo)	CHAR	000001
17		0D_NW_PQTY		Order Quantity (EPM Demo)	QUAN	000017
18		0N_NW_DDTN		Delivery DTN (EPM Demo)	DATE	000010
- Average Order Amount (ZBW_AVG)**

Pos	Key	InfoObject	Icon	Descript.
1		0CALYEAR		Calendar year
2		0CALMONTH		Calendar Year/Month
3		0D_NW_PRCAT		Product Category (EPM Demo)
4		0D_NW_PCURR		Order Currency
5		0RECORDMODE		BW Delta Process: Update Mode
6		ZBW_AVERG		Average Order Amount

A connection arrow labeled 'SAP HANA Script Processing' points from the source table to the target table.

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.

The screenshot shows the SAP Data Warehousing Workbench: Modeling interface. On the left, there's a navigation bar with icons for Favorites, Find, History, Data Flows, InfoProvider, InfoObjects, InfoSources, and DataSources. The 'Modeling' tab is selected. In the center, there's a tree view under 'Favorites' with nodes like 'Info Area for Open Item Analytics based on EPM Sales Mod', 'Average Order Amount', 'Data Transfer Processes', 'Virtual Infoprovider for Sales Invoice Model', 'Virtual Infoprovider for Sales Invoice Model-DunningAnal...', 'Test Marc Bernard', 'Test Marc Bernard', and 'SAP SEM-BPS Demo (FOX)'. A context menu is open over the 'Data Transfer Processes' node, with the option 'Create Data Transfer Process...' highlighted. Below this, a dialog box titled 'Create Data Transfer Process' is open. It shows the 'Data Transfer Proc.' field set to '0EPM_C04 -> ZBW_AVG', 'DTP Type' set to 'Standard (Can Be Scheduled)', 'Target of DTP' section with 'Object Type' as 'DataStore Object (advanced)' and 'Name' as 'ZBW_AVG' (description: Average Order Amount), and 'Source of DTP' section with 'Object Type' as 'InfoCube' and 'Name' as '0EPM_C04' (description: Completed Sales Orders (NW Demo ...)). At the bottom of the dialog are two buttons: a green checkmark and a red X.

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.

Display Data Transfer Process

Process Chain Maintenance Change to Real-Time DTP RDA Monitor

Data Transfer Proc. OEPM_C04 -> ZBW_AVG SAP HANA Execution

ID DTP_5I3Y60GVREXW2HLUYTONOYW8L

DTP Type Standard (Can Be Scheduled)

Version Active Saved

Extraction Update Execute

Source Object InfoCube
OEPM_C04
Completed Sales Orders (NW Demo EPM)

Extraction Mode Full

Request Selection Only retrieve last request

Parallel Extraction

Package Size 50,000 records

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.

For more details, please refer to the detailed blog about “SAP HANA-based BW Transformations” at <http://scn.sap.com/community/bw-hana/blog/2016/05/24/hana-based-bw-transformation> (highly recommended).

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.

If 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

To debug SAP HANA SQLScript, 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 an AMDP 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. In SAP BW, the **technical name** of the AMDP class */BIC/<GUID>* will be **different** in the target systems. For SAP BW/4HANA 1.0, see chapter 6.9.

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 *ORECORDMODE* 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 must 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.

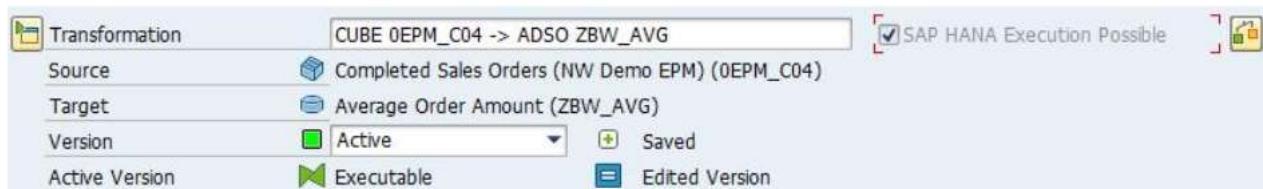
This kind of record is used in the change log of a Data Store object in order to save the value from the update. This is for a possible rollback and roll-forward for key figures with minimum or maximum aggregation. This record also has the update value for characteristics (in this case, it is the same 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.

6.8 SAP HANA Execution of Transformation

Before you create a transformation based on AMDP Script we recommend to check if standard transformation rules can be used. When maintaining a transformation, choose “Check” to the right of the transformation’s name field, you can check for yourself whether the transformation can be performed in SAP HANA. The system then attempts to create the transformation in SAP HANA. If this is successful, the transformation is flagged with Can be performed in SAP HANA. If the check is not successful, you can find out why by viewing the log.



Most formulas are supported. However, using any of the following formula operands will prevent execution in SAP HANA:

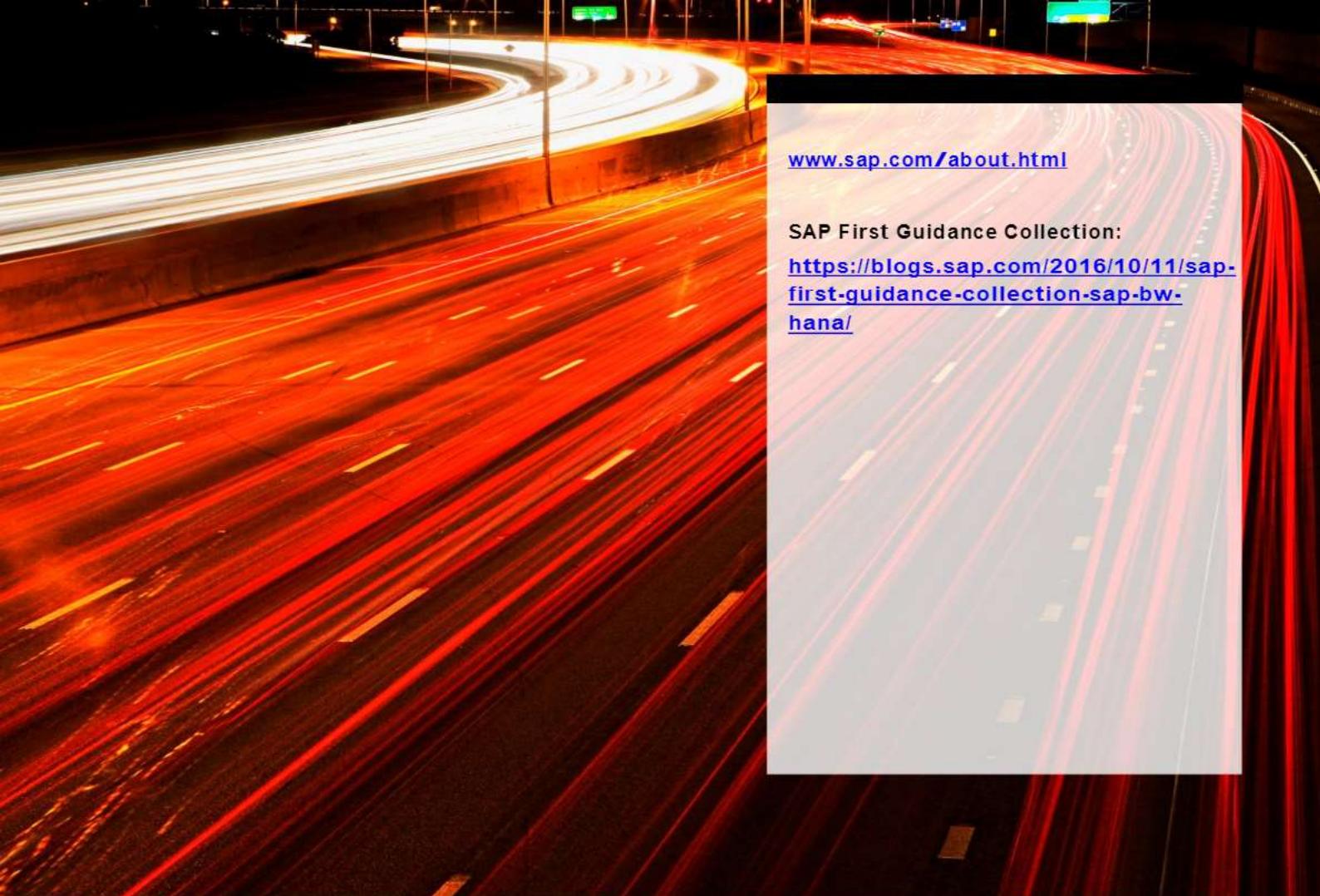
DATECONV, WEEK_TO_1ST_DAY, FISCPER_CALMONTH, CALMONTH_FISCPER, CONDENSE, SKIP_RECORD, ABORT_PACKAGE, SKIP_RECORD_AS_ERROR

6.9 SAP BW/4HANA

When maintaining transformations with AMDP routines, the system creates class */BIC/<ID>_M*, where the user enters the code. Depending on the type of transformation, there are several methods called *GLOBAL_START*, *GLOBAL_END*, and *GLOBAL_EXPERT*, as well as methods named *S<SEGID>_G<GROUPID>_R<RULEID>* for individual field routines.

At time of activation, the system generates class */BIC/<ID>_A* containing the most recent code, which is used during the data staging process. Editing this class is not allowed.

Note: SAP retains the right to change the naming of these classes and methods. It is therefore recommended NOT to call the embedded code from within other custom developments.



www.sap.com/about.html

SAP First Guidance Collection:

<https://blogs.sap.com/2016/10/11/sap-first-guidance-collection-sap-bw-hana/>



The Best-Run Businesses Run SAP™