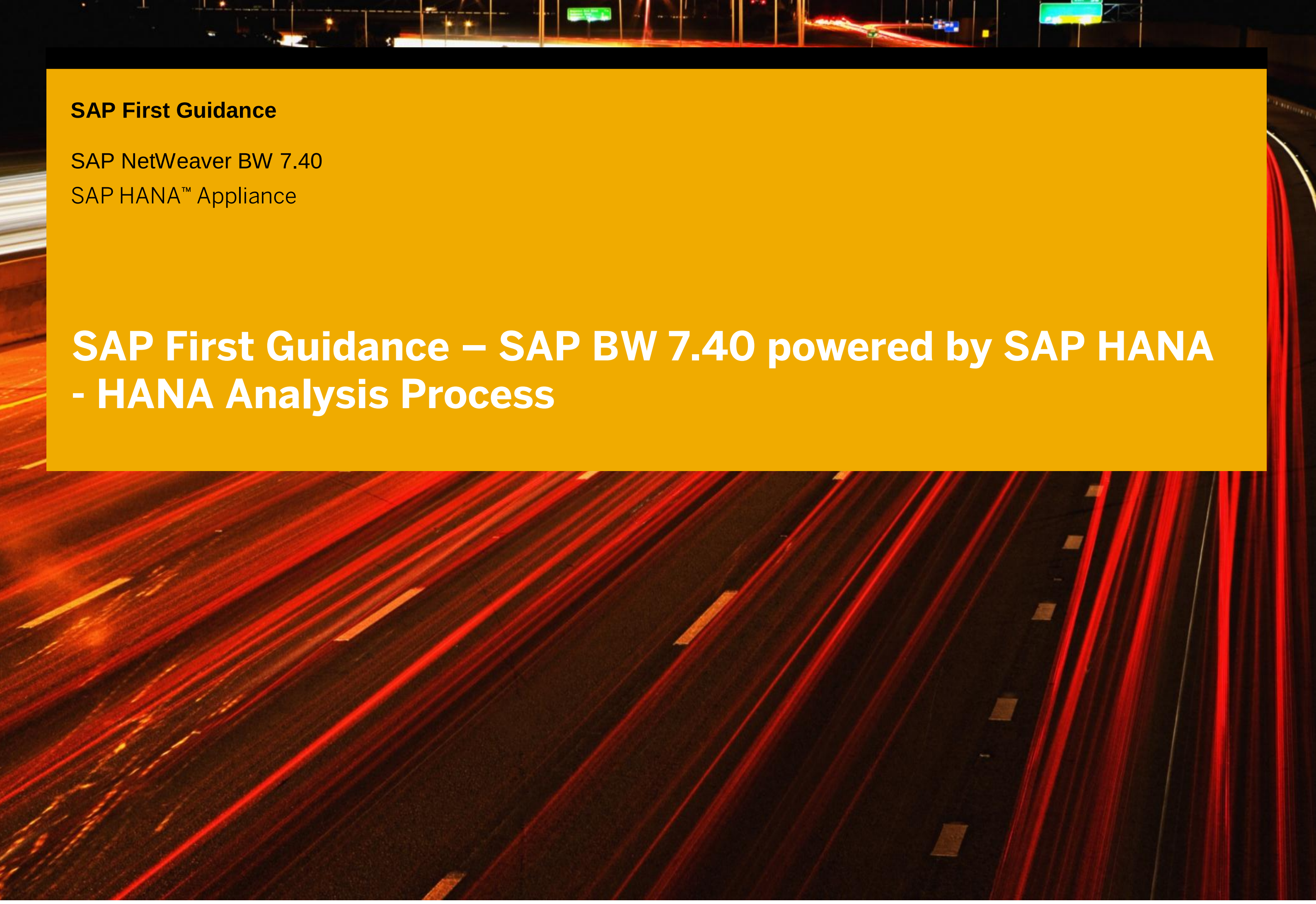


SAP First Guidance

SAP NetWeaver BW 7.40
SAP HANA™ Appliance

SAP First Guidance – SAP BW 7.40 powered by SAP HANA - HANA Analysis Process

Applicable Releases:

SAP BW 7.40 SP5 powered by SAP HANA

This First Guidance Document is the beginning 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 on HANA.

Version 2.0

June 2014

Author: Gordon Witzel

SAP "How-to" Guides are intended to simplify the product implementtation. 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 expressively 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
1.00	First draft of this guide
2.00	Added section regarding referential integrity and unit / currency handling

Typographic Conventions

Type Style	Description
Example Text	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

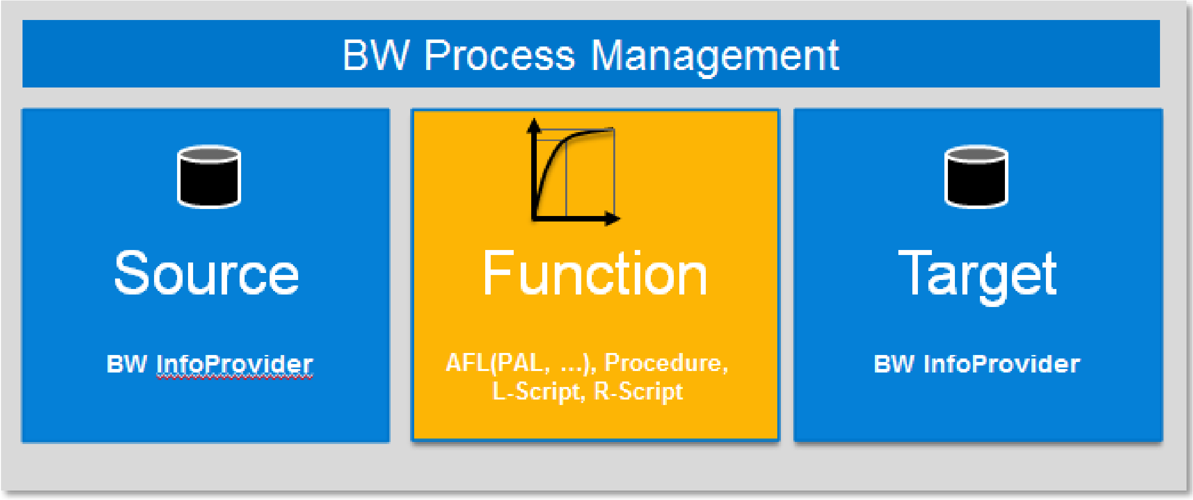
The Best-Run Businesses Run SAP™

Table of Contents

1. Introduction

Starting with SAP BW 7.4 SP5 on HANA a new feature to analyze data from certain perspectives, for example to calculate ABC classes or to calculate scoring classes, is introduced. This new feature is called SAP HANA Analysis Process and enables the data warehouse modeler to analyze the data using different predefined or own written functions or scripts. HANA provides the integration numerous specialized libraries like PAL, AFL, R to understand the correlation of the data in the existing EDW.

Via BW's new HANA Analysis Process (HAP) capability it is now possible to use all of these on top of BW models and consume the result as a BW model.



Picture 1: Integration of HAP into **BW system process management**

The new HANA Analysis Process (HAP) offers enhanced analysis capabilities in an unprecedented performance. There are many new capabilities to execute the HANA Analysis Process (HAP).

It is possible to execute HANA functions from different libraries (e.g. PAL, AFL, R) directly on BW InfoProvider data like clustering, association algorithms, regression analysis, anomaly detection, weighted score, exponential smoothing, etc.

Execution of complex and data intensive processes on HANA without losing the integrity and integration with the BW environment is now possible and the result of the HANA Analysis Process for further processing could be materialized in BW automated. By using the Standard BW framework scheduling and background processing is supported as well and one more strength of the deep integration into BW on HANA.

2. Scenario

In schema SAPHBW (BW managed schema) a Composite Provider YEPM_CP_SO (EPM Sales Orders (CV based)) was created. This Composite Provider serves as Source Provider and holds the transactional data for the HANA Analysis Process based on the InfoCubes 0EPM_C03 and 0EPM_C04.

InfoCube 0EPM_C03 contains the Open Sales Orders based on the Enterprise Procurement Model. The InfoCube 0EPM_C04 contains the Completed Sales Orders based on the Enterprise Procurement Model.

An ODP based Datasource YSHEPM_DS_SO is used to extract the relevant EPM data from the source.

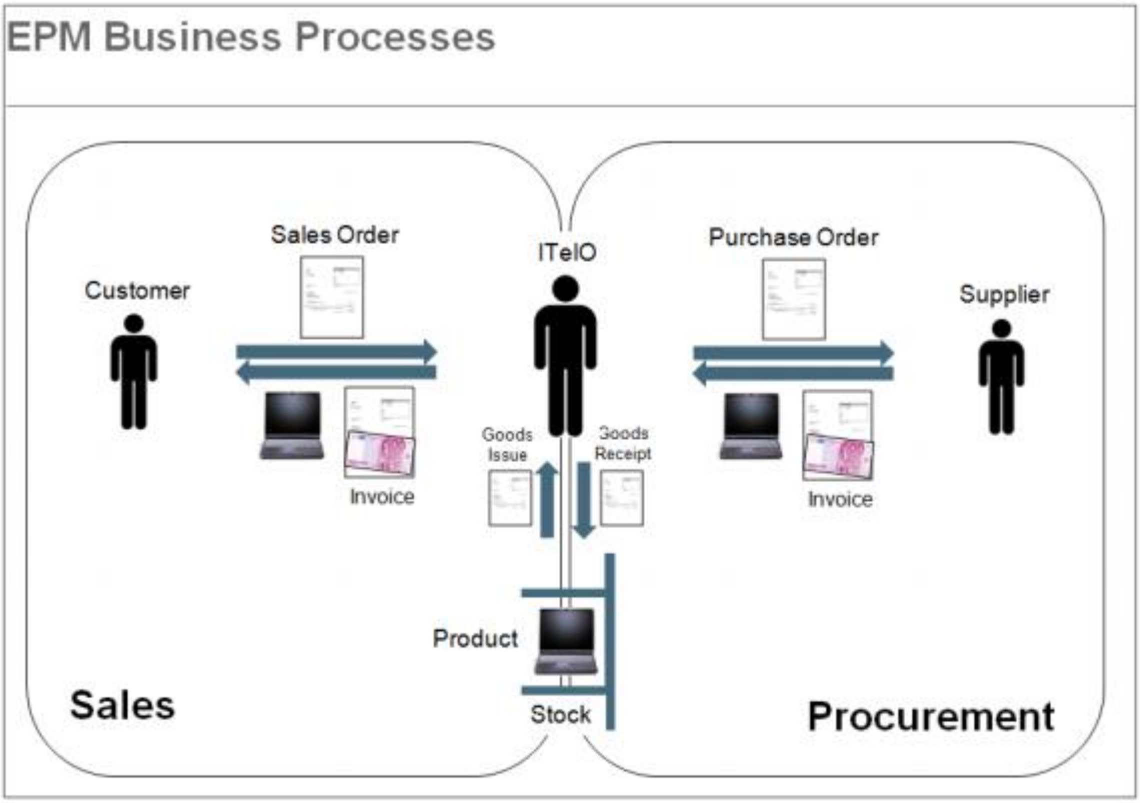
The dataflow for this InfoCubes is quite simple – the first step is load the data into a data propagation DSO and in the second step the data are loading into the relevant InfoCubes.

The NetWeaver Enterprise Procurement Model (EPM) is an application intended to be used for demonstration and testing purposes.

For more information regarding the Enterprise Procurement Model (EPM) please check this document:

Enterprise Procurement Model (EPM)

The following picture shows an overview of the basic business processes in EPM.



Picture 2: EPM Business Processes

This Composite provider acts as a Source for different HANA Analysis Processes and the document describes different ways of execution HANA Analysis Processes. It starts with a simple ABC analysis and describes the source, function and the possible data targets. Furthermore it describes the different ways to analyze the data within HANA Analysis Processes – using prefabricated procedures or scripts or procedures. By using the procedures it is possible to use a generated HANA Procedure or to use an ABAP Managed Database Procedure. This document explains the details, the use cases, the advantages and disadvantages of each feature in detail.

The Composite Provider is the Source and the HAP is based on this data. In the next chapters different HAP will described and the advantages and disadvantages will be elaborated.

The underlying basis InfoCube 0EPM_C03 contains the Open Sales Orders and the InfoCube 0EPM_C04 contains the Completed Sales orders and will be loaded regularly.

3. Prerequisite

You can use the SAP HANA Analysis Process if you are using a SAP HANA database in combination with SAP BW 7.4 SP5 or higher.

In general SAP BW 7.4 SP5 enables the usage of SAP HANA elements such as virtual

tables, HANA tables, Calculation and/or Analytical views in the definition of BW metadata objects such as OpenODSView and Composite Provider. This new SAP HANA elements can used as Sources and/or targets for the SAP HANA analysis processes if this is needed.

To run a SAP HANA analysis process a specialized procedure on the database will be generated and triggered accordingly.

To optimize the performance of the executed procedure and to stabilize the Hana database the usage of the script server is a prerequisite for executing any HANA Analysis Process. The script server is one of the SAP HANA database servers, which has its own memory management and is therefore separated from the main server.

Therefore please start the script server as described in note:

1650957 - SAP HANA Database: Starting the Script Server

If the script server is not started the following error message will be raised:
34091 "No ScriptServer available. See SAPNote 1650957 for further information."

Install the Application Function Library (AFL), which includes the PAL (SAP HANA Predictive Analysis Library). For more information, see the following sections in the SAP HANA Update and Configuration Guide.

- Installing AFL: Section 4.1 Adding Application Function Libraries on an SAP HANA System on page 34
- Updating AFL: Section 4.2 Performing Automated Updates for Application Function Libraries (AFLs) on page 34.

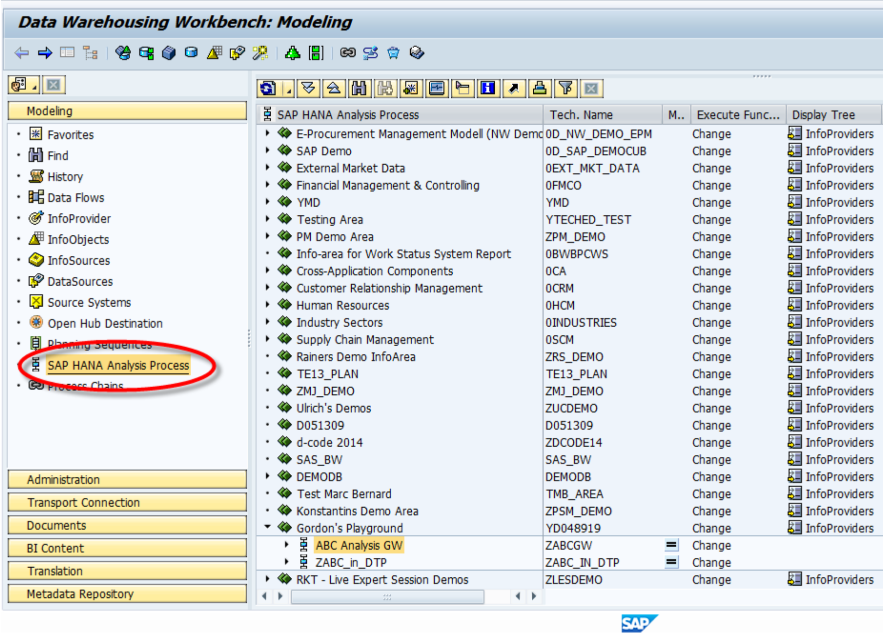
The revision of the AFL must match the revision of SAP HANA.

Each release of the AFL has a version in the form of <revision_number>.<patch_level>. For example, AFL 40.01 refers to revision 40 and patch level 01, and it should be installed with SAP HANA revision 40 only.

4. First Example: Creating a SAP HANA Analysis Process

As a first example a simple ABC Analysis will be described to get an overview of the possible Sources, function/scripts and data targets.

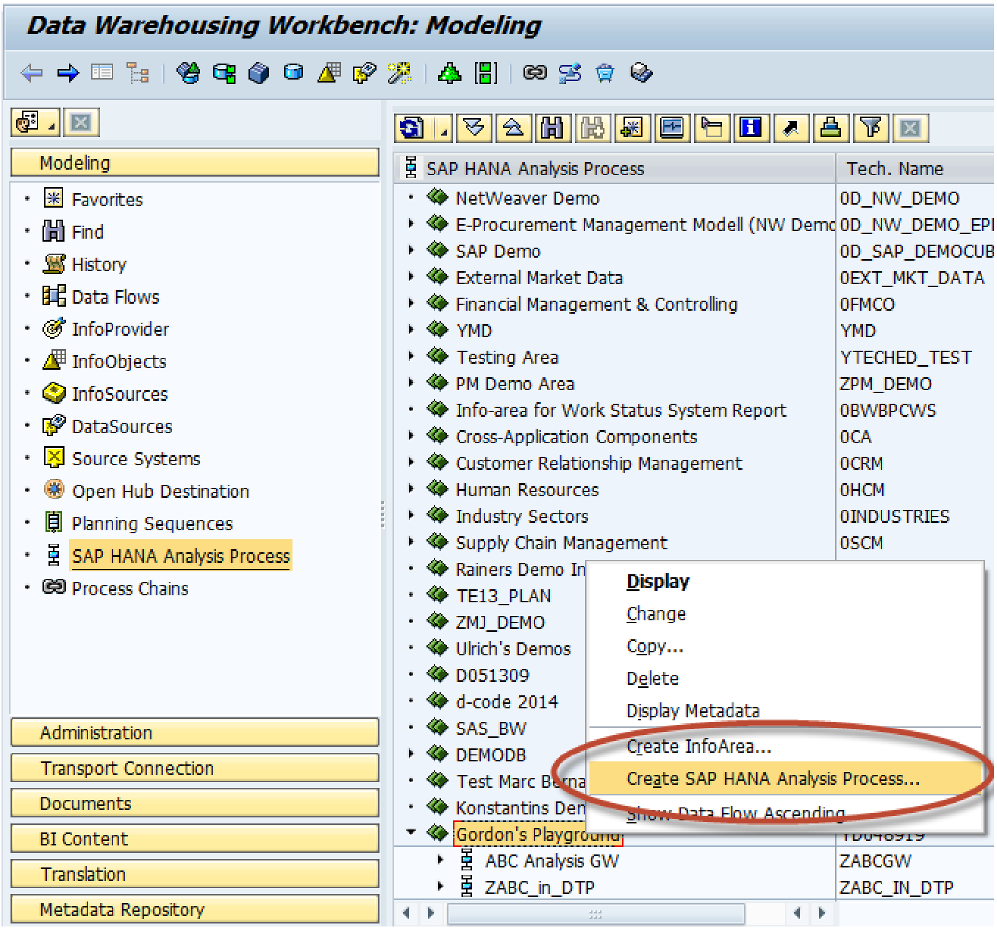
Logon to the BW system and choose SAP HANA Analysis Process in the Data Warehousing Workbench under Modeling (TA RSA1).



SAP HANA Analysis Process	Tech. Name	M..	Execute Func...	Display Tree
E-Procurement Management Modell (NW Demo	0D_NW_DEMO_EPM		Change	InfoProviders
SAP Demo	0D_SAP_DEMOCUB		Change	InfoProviders
External Market Data	0EXT_MKT_DATA		Change	InfoProviders
Financial Management & Controlling	0FMCO		Change	InfoProviders
YMD	YMD		Change	InfoProviders
Testing Area	YTECHED_TEST		Change	InfoProviders
PM Demo Area	ZPM_DEMO		Change	InfoProviders
Info-area for Work Status System Report	0BWBPCWS		Change	InfoProviders
Cross-Application Components	0CA		Change	InfoProviders
Customer Relationship Management	0CRM		Change	InfoProviders
Human Resources	0HCM		Change	InfoProviders
Industry Sectors	0INDUSTRIES		Change	InfoProviders
Supply Chain Management	0SCM		Change	InfoProviders
Rainers Demo InfoArea	ZRS_DEMO		Change	InfoProviders
TE13_PLAN	TE13_PLAN		Change	InfoProviders
ZMJ_DEMO	ZMJ_DEMO		Change	InfoProviders
Ulrich's Demos	ZUCDEMO		Change	InfoProviders
D051309	D051309		Change	InfoProviders
d-code 2014	ZDCODE14		Change	InfoProviders
SAS_BW	SAS_BW		Change	InfoProviders
DEMODB	DEMODB		Change	InfoProviders
Test Marc Bernard	TMB_AREA		Change	InfoProviders
Konstantins Demo Area	ZPSM_DEMO		Change	InfoProviders
Gordon's Playground	YD048919		Change	InfoProviders
ABC Analysis GW	ZABCGW	=	Change	
ZABC_in_DTP	ZABC_IN_DTP	=	Change	
RKT - Live Expert Session Demos	ZLESDEMO		Change	InfoProviders

Picture 3: open SAP HANA Analysis Process

In the next step choose Create SAP HANA Analysis Process in the preferred InfoArea. Than enter a name and description for your analysis process.



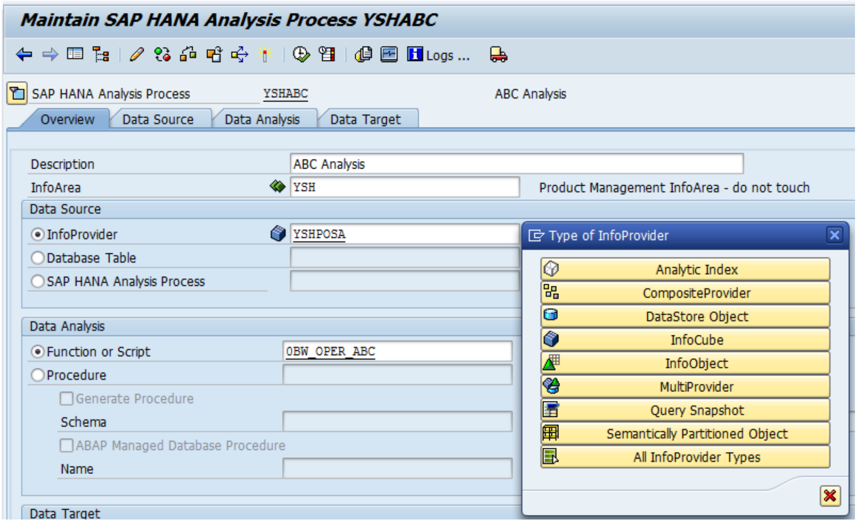
SAP HANA Analysis Process	Tech. Name
NetWeaver Demo	0D_NW_DEMO
E-Procurement Management Modell (NW Demo	0D_NW_DEMO_EPI
SAP Demo	0D_SAP_DEMOCUB
External Market Data	0EXT_MKT_DATA
Financial Management & Controlling	0FMCO
YMD	YMD
Testing Area	YTECHED_TEST
PM Demo Area	ZPM_DEMO
Info-area for Work Status System Report	0BWBPCWS
Cross-Application Components	0CA
Customer Relationship Management	0CRM
Human Resources	0HCM
Industry Sectors	0INDUSTRIES
Supply Chain Management	0SCM
Rainers Demo In	
TE13_PLAN	
ZMJ_DEMO	
Ulrich's Demos	
D051309	
d-code 2014	
SAS_BW	
DEMODB	
Test Marc Berna	
Konstantins Den	
Gordon's Playground	TD048919
ABC Analysis GW	ZABCGW
ZABC_in_DTP	ZABC_IN_DTP

Picture 4: Definition of HAP

In the next step, the Source of the HAP needs to be defined.

It is possible to use the following Sources:

- InfoProvider like:
  - Analytic Index
  - Composite Provider
  - DataStore Object
  - InfoCube
  - InfoObject
  - MultiProvider
  - Query Snapshots
  - Semantically Partitioned Object
- Database tables
- SAP HANA Analysis Process



Maintain SAP HANA Analysis Process YSHABC

SAP HANA Analysis Process YSHABC ABC Analysis

Overview | Data Source | Data Analysis | Data Target

Description: ABC Analysis

InfoArea: YSH Product Management InfoArea - do not touch

Data Source

● InfoProvider: YSHPOSA

○ Database Table

○ SAP HANA Analysis Process

Data Analysis

● Function or Script: 0BW_OPER_ABC

○ Procedure

□ Generate Procedure

Schema

□ ABAP Managed Database Procedure

Name

Data Target

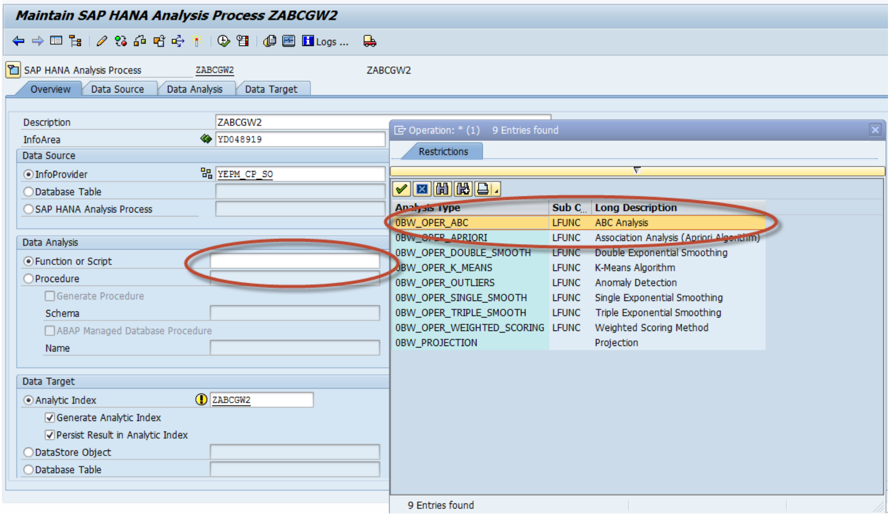
Type of InfoProvider

- Analytic Index
- CompositeProvider
- DataStore Object
- InfoCube
- InfoObject
- MultiProvider
- Query Snapshot
- Semantically Partitioned Object
- All InfoProvider Types

Picture 5: HAP possible sources

As you can see in Picture 4, there are different sources possible. For the ABC Analysis we use an existing Composite Provider YEPM_CP_SO.

After that, it is important to define the function for the execution of the Analysis. In this first example we use the ABC Analysis as a preconfigured function.



Maintain SAP HANA Analysis Process ZABCGW2

SAP HANA Analysis Process ZABCGW2 ZABCGW2

Overview | Data Source | Data Analysis | Data Target

Description: ZABCGW2

InfoArea: YD048919

Data Source

● InfoProvider: YEPM_CP_SO

○ Database Table

○ SAP HANA Analysis Process

Data Analysis

● Function or Script

○ Procedure

□ Generate Procedure

Schema

□ ABAP Managed Database Procedure

Name

Data Target

● Analytic Index: ZABCGW2

☑ Generate Analytic Index

☑ Persist Result in Analytic Index

○ DataStore Object

○ Database Table

Operation: * (1) 9 Entries found

Restrictions

Analysis Type	Sub C...	Long Description
0BW_OPER_ABC	LFUNC	ABC Analysis
0BW_OPER_APRIORI	LFUNC	Association Analysis (Apriori Algorithm)
0BW_OPER_DOUBLE_SMOOTH	LFUNC	Double Exponential Smoothing
0BW_OPER_K_MEANS	LFUNC	K-Means Algorithm
0BW_OPER_OUTLIERS	LFUNC	Anomaly Detection
0BW_OPER_SINGLE_SMOOTH	LFUNC	Single Exponential Smoothing
0BW_OPER_TRIPLE_SMOOTH	LFUNC	Triple Exponential Smoothing
0BW_OPER_WEIGHTED_SCORING	LFUNC	Weighted Scoring Method
0BW_PROJECTION		Projection

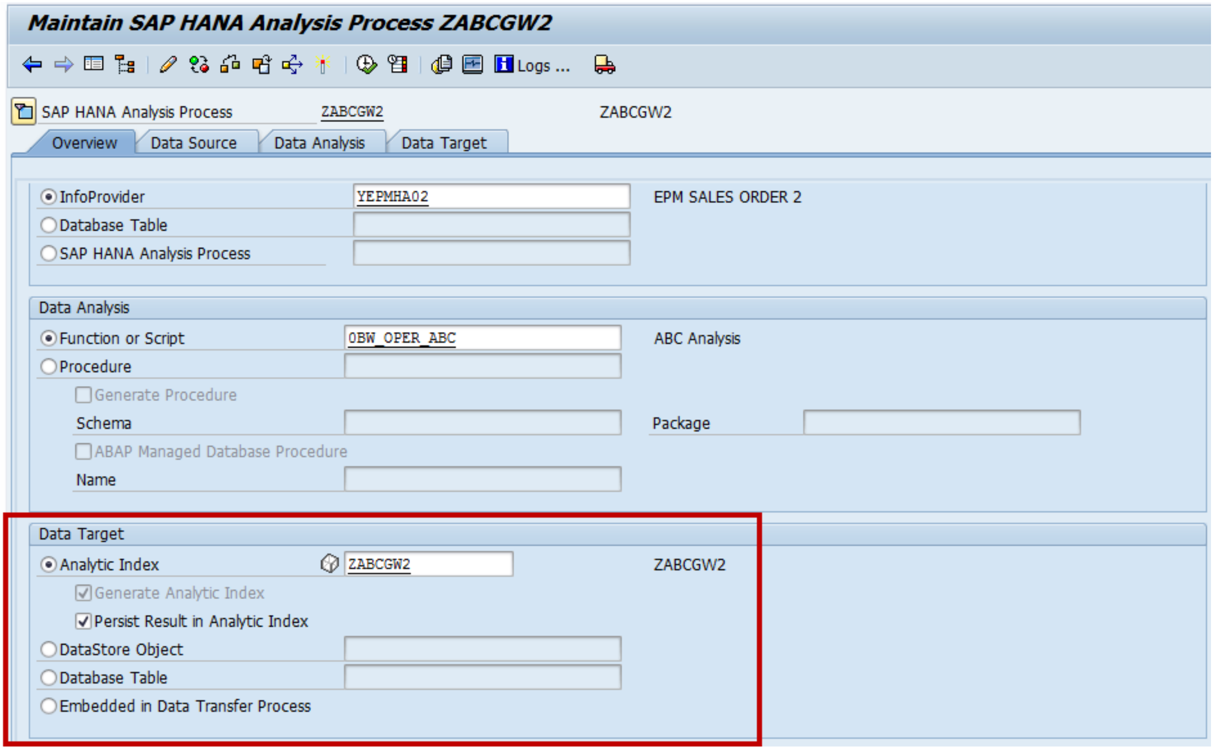
9 Entries found

Picture 6: HAP functions

In the next step the target needs to be defined. It is possible to use the following data targets:

- Analytic Index
- DataStore Object
- Database table
- Embedded in Data Transfer Process

We will use an Analytic Index to persist the results. If the Index does not exist already, it is possible to generate the index by activating the HAP. The flag to generate analytic index is set automatically if the Analytic Index does not exist already. Furthermore the flag to persist the Results in the Analytic Index is set automatically as well. If the persistence of the results is not needed this flags needs to be eliminated. For maintaining the Analytic Index please refers to chapter Error! Reference source not found. Error! Reference source not found..



Maintain SAP HANA Analysis Process ZABCGW2

SAP HANA Analysis Process ZABCGW2 ZABCGW2

● InfoProvider YEPMHA02 EPM SALES ORDER 2
○ Database Table
○ SAP HANA Analysis Process

Data Analysis

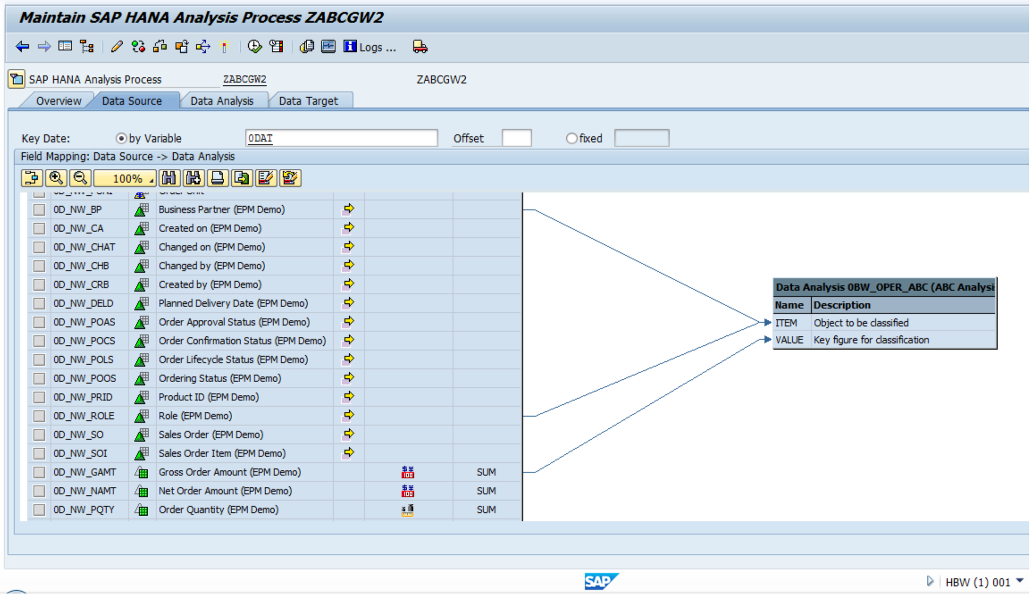
● Function or Script 0BW_OPER_ABC ABC Analysis
○ Procedure
□ Generate Procedure
Schema
Package
□ ABAP Managed Database Procedure
Name

Data Target

● Analytic Index ZABCGW2 ZABCGW2
☑ Generate Analytic Index
☑ Persist Result in Analytic Index
○ DataStore Object
○ Database Table
○ Embedded in Data Transfer Process

Picture 7: HAP data targets

Now the Source for the data needs to be defined. In the second tab it is important to map the Source data fields into the Data Analysis Fields. In this example we map 0D_NW_BP (Business Partner (EPM Demo)) to the analysis item and 0D_NW_GAMT (Gross Order Amount (EPM Demo)) to Key Figure for Classification (Item).



Maintain SAP HANA Analysis Process ZABCGW2

SAP HANA Analysis Process ZABCGW2 ZABCGW2

Key Date: ● by Variable 0DAT Offset ○ fixed

Field Mapping: Data Source -> Data Analysis

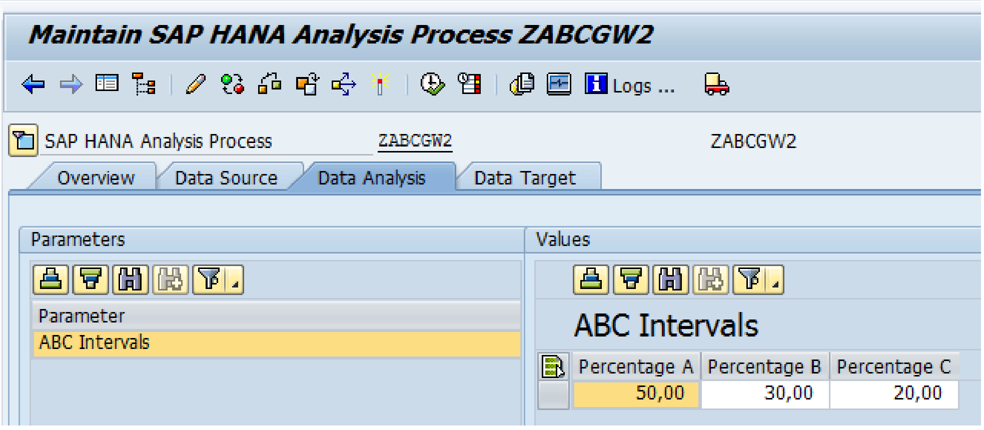
☐	0D_NW_BP	Business Partner (EPM Demo)	
☐	0D_NW_CA	Created on (EPM Demo)	
☐	0D_NW_CHAT	Changed on (EPM Demo)	
☐	0D_NW_CHB	Changed by (EPM Demo)	
☐	0D_NW_CRB	Created by (EPM Demo)	
☐	0D_NW_DELD	Planned Delivery Date (EPM Demo)	
☐	0D_NW_POAS	Order Approval Status (EPM Demo)	
☐	0D_NW_POCS	Order Confirmation Status (EPM Demo)	
☐	0D_NW_POLS	Order Lifecycle Status (EPM Demo)	
☐	0D_NW_POOS	Ordering Status (EPM Demo)	
☐	0D_NW_PRID	Product ID (EPM Demo)	
☐	0D_NW_ROLE	Role (EPM Demo)	
☐	0D_NW_SO	Sales Order (EPM Demo)	
☐	0D_NW_SOI	Sales Order Item (EPM Demo)	
☐	0D_NW_GAMT	Gross Order Amount (EPM Demo)	SUM
☐	0D_NW_NAMT	Net Order Amount (EPM Demo)	SUM
☐	0D_NW_PQTY	Order Quantity (EPM Demo)	SUM

Data Analysis 0BW_OPER_ABC (ABC Analysi

Name	Description
ITEM	Object to be classified
VALUE	Key figure for classification

Picture 8: Mapping of Source fields

Now the Data Analysis Tab has to be maintained. For the ABC analysis in this example we choose 50 Percentage for A, 30 Percentage for B and 20 Percentage for C. This means the best 50 percent Business Partners based on the Gross Order Amount are in category A, the next 30 percent are in Category B and the last 20 percent are in category C. This means the Business Partners with the highest Gross Order Amount (50 percent) categorized as A, the next 30 percent are categorized as B and the last 20 percent are categorized as C.



Maintain SAP HANA Analysis Process ZABCGW2

SAP HANA Analysis Process ZABCGW2 ZABCGW2

Parameters

Parameter
ABC Intervals

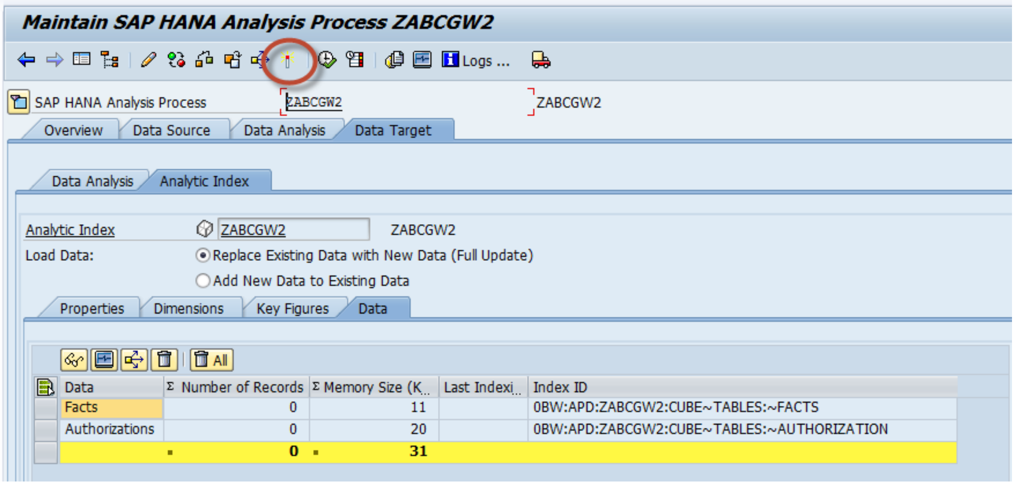
Values

ABC Intervals

Percentage A	Percentage B	Percentage C
50,00	30,00	20,00

Picture 9: Maintain the ABC Intervals

In the Data Target tab exists the possibilities to replace the existing data with the new data (standard) or to add new data to existing data. Furthermore it is possible to check the existing data and check the Properties, Dimensions and Key Figures. To activate the HAP press the Activation button. Now the HAP is ready to run and can scheduled via process chain or directly executed.



Maintain SAP HANA Analysis Process ZABCGW2

SAP HANA Analysis Process ZABCGW2 ZABCGW2

Analytic Index ZABCGW2 ZABCGW2

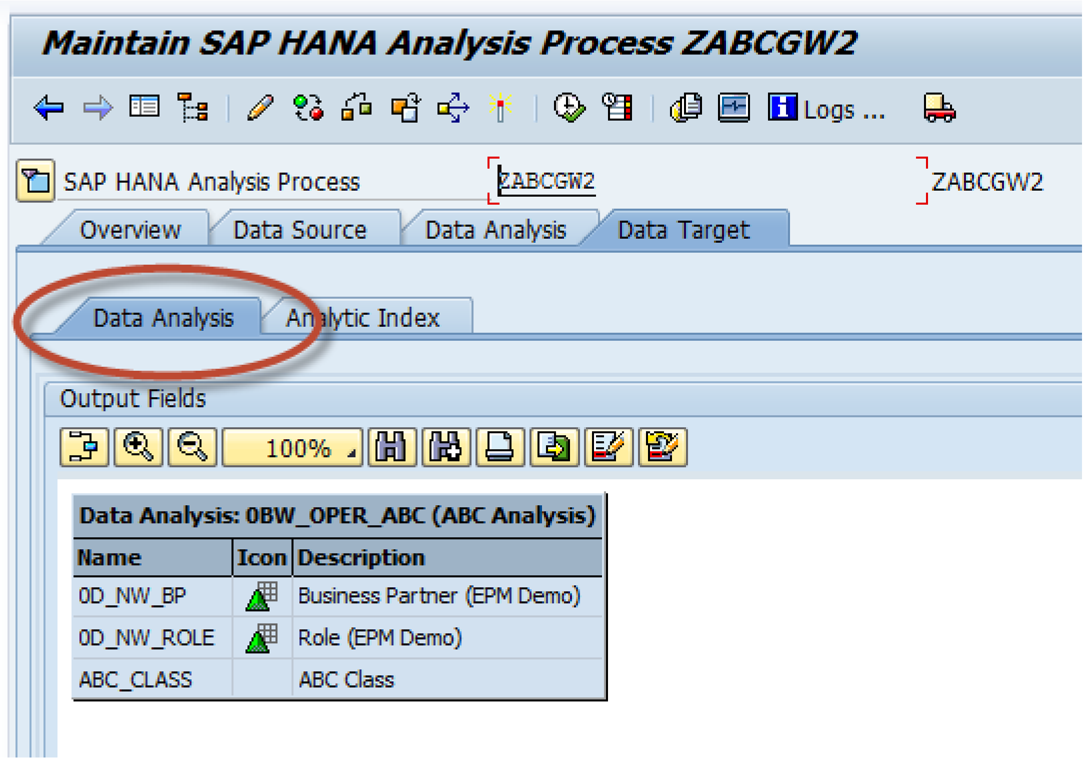
Load Data: ● Replace Existing Data with New Data (Full Update)

○ Add New Data to Existing Data

Data	Σ Number of Records	Σ Memory Size (K..	Last Indexi..	Index ID
Facts	0	11		0BW:APD:ZABCGW2:CUBE~TABLES:~FACTS
Authorizations	0	20		0BW:APD:ZABCGW2:CUBE~TABLES:~AUTHORIZATION
	▪ 0	▪ 31		

Picture 10: Data Target of the ABC Analysis

In the Data Analysis Tab the Output fields are shown. It is possible to use this Output fields to load the results from the HAP into other data targets (regular BW processes if the data is persist in a DSO or the HAP is an embedded DTP process).



Maintain SAP HANA Analysis Process ZABCGW2

SAP HANA Analysis Process ZABCGW2 ZABCGW2

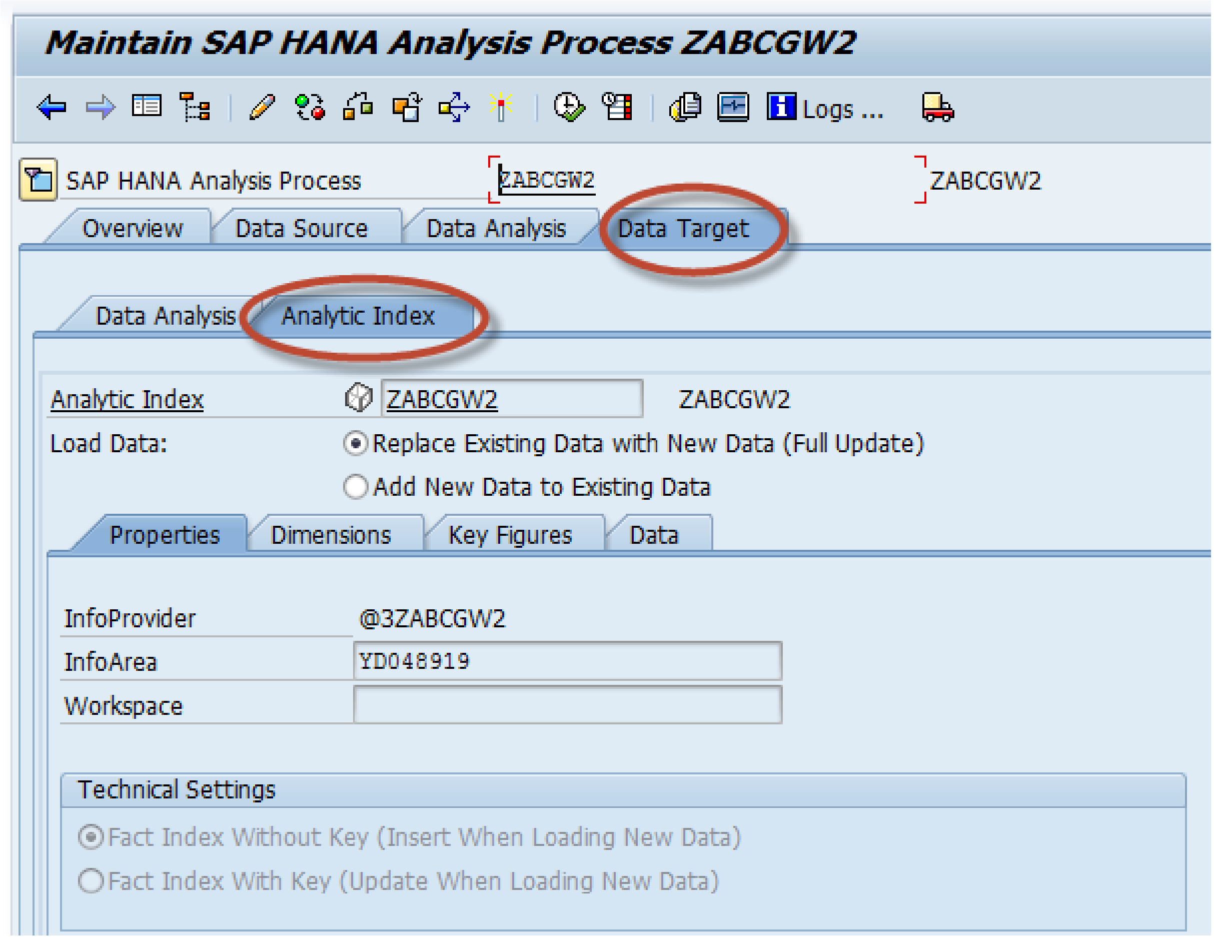
Data Analysis: 0BW_OPER_ABC (ABC Analysis)

Name	Icon	Description
0D_NW_BP	▲	Business Partner (EPM Demo)
0D_NW_ROLE	▲	Role (EPM Demo)
ABC_CLASS		ABC Class

Picture 11: Data Target – Output Fields of the ABC Analysis

Another tab is the Analytic Index Tab when the HAP is using an Analytic Index as a Target. Within this tab it is possible to see the Properties, Dimensions, key Figures and Data of the chosen HAP.

In the Analytic Tab it is possible to decide if the data should load as a Full with deleting the existing data or if the new data should be add as new data to existing ones.



Maintain SAP HANA Analysis Process ZABCGW2

SAP HANA Analysis Process ZABCGW2 ZABCGW2

Analytic Index ZABCGW2 ZABCGW2

Load Data: ● Replace Existing Data with New Data (Full Update)

○ Add New Data to Existing Data

InfoProvider @3ZABCGW2

InfoArea YD048919

Workspace

Technical Settings

● Fact Index Without Key (Insert When Loading New Data)

○ Fact Index With Key (Update When Loading New Data)

Picture 12: Analytic Index Tab

If the definition is done the HAP can be executed, scheduled and used for further analysis (e.g. Queries).
Additionally a video is published to explain the steps for the HAP based on an ABC analysis:

http://www.sdn.sap.com/irj/scn/index?rid=/library/uuid/30ff0242-2c97-3110-298a-d5e42168384f&overridelayout=true

5. Sources for HANA Analysis Processes

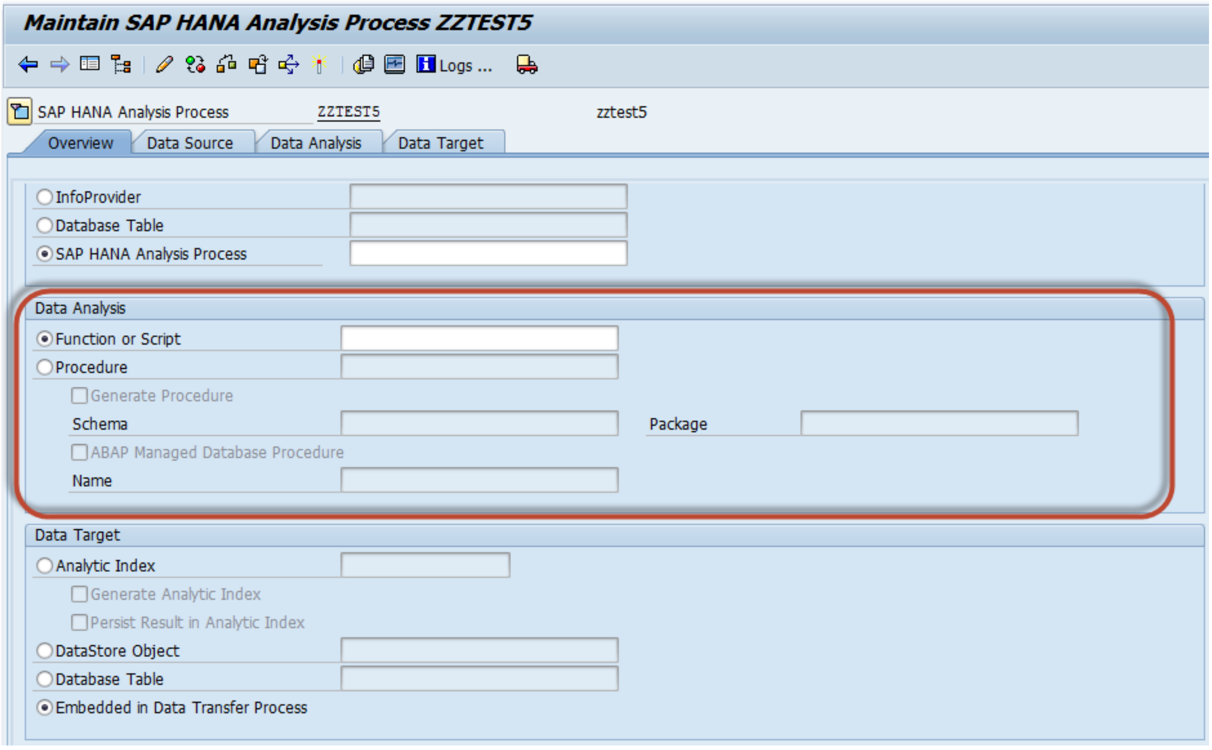
This chapter will document the different sources for the HAP shortly. For a detailed documentation of the possible sources please refer to:
Appendix - Definition of the HAP Sources
In general it is possible to use the following Sources for Hana Analysis Processes:

- InfoProvider like:
  - Analytic Index
  - Composite Provider
  - Data Store Object
  - InfoCube
  - Info Object
  - Multi Provider
  - Query Snapshot
  - Semantically Partitioned Object
- Database tables
- SAP HANA Analysis Process

It strongly depends on the use case and the existing data model case which Source should be used. In general all InfoProviders with an existing column view on the SAP HANA database are supported as data sources for the HAP.

6. Data Analysis for HANA Analysis Processes

In the data analysis tab you have to choose the preferred kind of the data analysis.
It is possible to use prefabricated procedures or scripts or procedures. By using the procedures it is possible to use a generated HANA Procedure or to use an ABAP Managed Database Procedure.



Maintain SAP HANA Analysis Process ZZTEST5

SAP HANA Analysis Process ZZTEST5 zztest5

○ InfoProvider
○ Database Table
● SAP HANA Analysis Process

Data Analysis

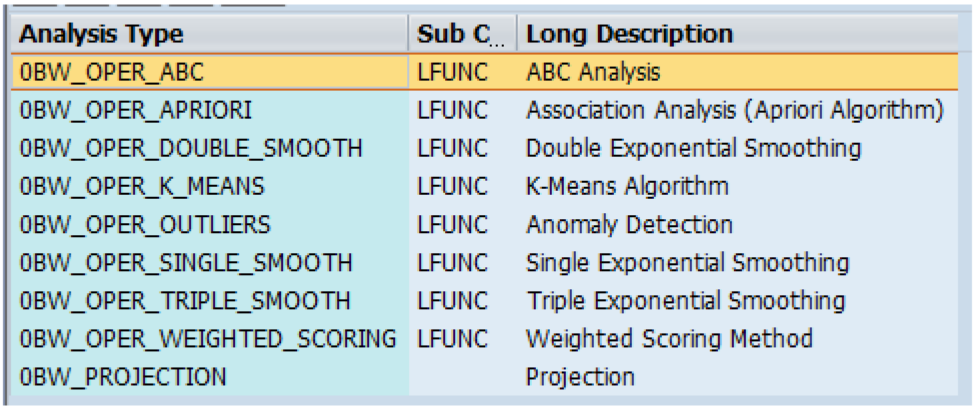
● Function or Script
○ Procedure
□ Generate Procedure
Schema
Package
□ ABAP Managed Database Procedure
Name

Data Target

○ Analytic Index
□ Generate Analytic Index
□ Persist Result in Analytic Index
○ DataStore Object
○ Database Table
● Embedded in Data Transfer Process

Picture 13: Data Analysis

By using a function it is possible to select a prefabricated procedure. SAP offers a couple of these procedures at the moment and will be extended by the time:



Analysis Type	Sub C...	Long Description
0BW_OPER_ABC	LFUNC	ABC Analysis
0BW_OPER_APRIORI	LFUNC	Association Analysis (Apriori Algorithm)
0BW_OPER_DOUBLE_SMOOTH	LFUNC	Double Exponential Smoothing
0BW_OPER_K_MEANS	LFUNC	K-Means Algorithm
0BW_OPER_OUTLIERS	LFUNC	Anomaly Detection
0BW_OPER_SINGLE_SMOOTH	LFUNC	Single Exponential Smoothing
0BW_OPER_TRIPLE_SMOOTH	LFUNC	Triple Exponential Smoothing
0BW_OPER_WEIGHTED_SCORING	LFUNC	Weighted Scoring Method
0BW_PROJECTION		Projection

Picture 14: HAP Functions

These functions can be used as preconfigured functions and all this functions are documented in the SAP HANA Predictive Analysis Library (PAL).

All the functions are documented here:

http://help.sap.com/hana/SAP_HANA_Predictive_Analysis_Library_PAL_en.pdf

For the preconfigured functions all the coding and the framework is done and these functions can be used quite fast.

The benefits of these preconfigured functions are:

- Fast implementation time of the analysis
- Coding is already done
- Standard BW transportation connection

If the function is not sufficient enough there is a possibility to develop an ABAP managed database procedure or a standalone database procedure.

6.1 Generated HANA Procedure

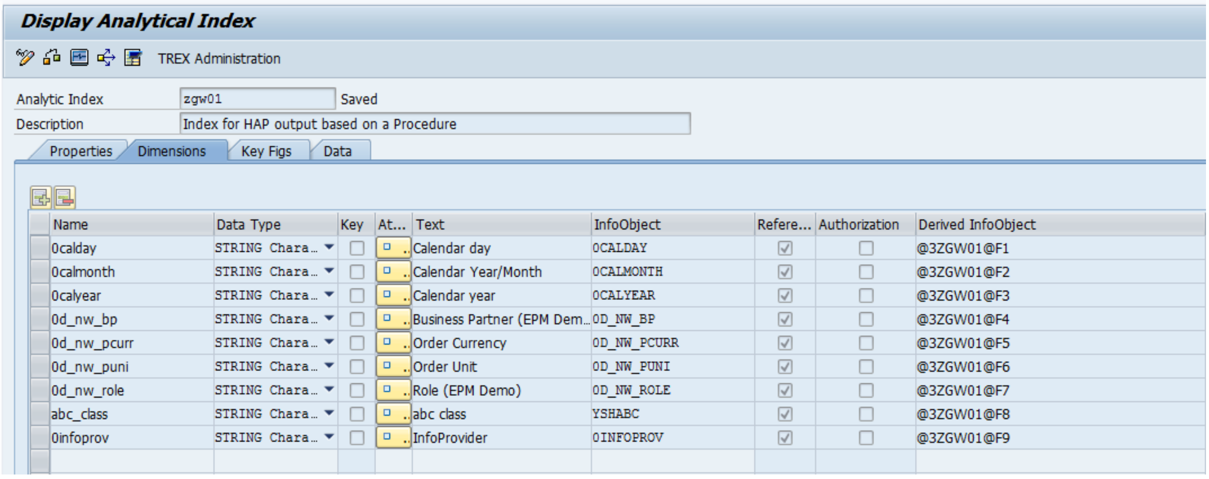
A further possibility for data analysis is to use own written HANA procedures. In our example we will create a simple coding and we will write the output into an Analytic Index.

In the first step the creation of an Analytic Index is needed if other fields than the Input fields from the Source are required.

At the current version of HAP it is only possible to use the same structure from the Input fields in the Output fields when an Analytic Index is used as a target and this Analytic Index does not exist and needs to be generated. In this case the generated Analytic Index contains the same fields than the Input Structure (in this case from the Composite Provider YEPMHA02) defined in the Data Source tab of the HAP.

It is not possible to change the automatic generated Analytic Index and use more or less fields than the Input structure. To cover this requirement we create a new Analytic Index and use this in the HAP as a data target.

To create an Analytic Index we used the TA RSDD_LTIP and created a sample Analytic Index with a couple of dimensions and one Key Figure.



Display Analytical Index

Analytic Index: zgw01 Saved

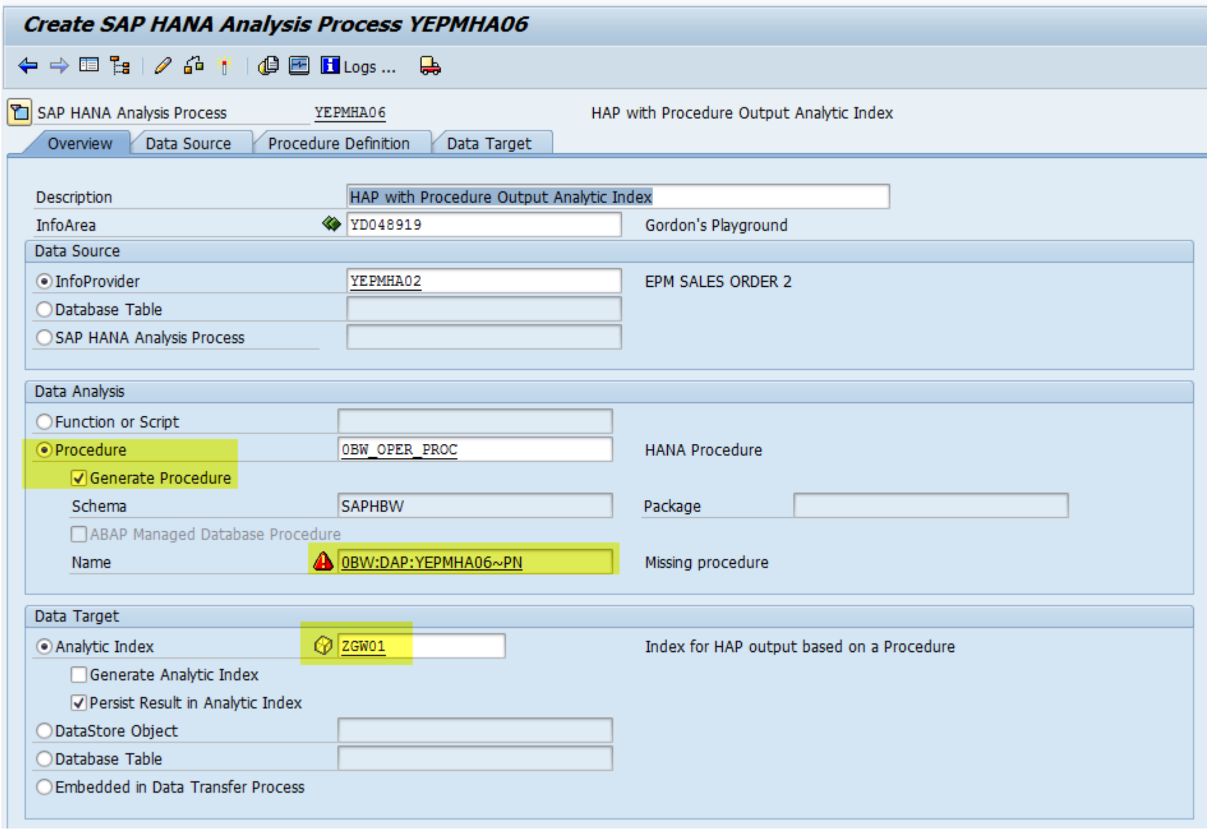
Description: Index for HAP output based on a Procedure

Name	Data Type	Key	At...	Text	InfoObject	Refere...	Authorization	Derived InfoObject
0calday	STRING Chara...	□		Calendar day	0CALDAY	☑	□	@3ZGW01@F1
0calmonth	STRING Chara...	□		Calendar Year/Month	0CALMONTH	☑	□	@3ZGW01@F2
0calyear	STRING Chara...	□		Calendar year	0CALYEAR	☑	□	@3ZGW01@F3
0d_nw_bp	STRING Chara...	□		Business Partner (EPM Dem...	0D_NW_BP	☑	□	@3ZGW01@F4
0d_nw_pcurr	STRING Chara...	□		Order Currency	0D_NW_PCURR	☑	□	@3ZGW01@F5
0d_nw_puni	STRING Chara...	□		Order Unit	0D_NW_PUNI	☑	□	@3ZGW01@F6
0d_nw_role	STRING Chara...	□		Role (EPM Demo)	0D_NW_ROLE	☑	□	@3ZGW01@F7
abc_class	STRING Chara...	□		abc class	YSHABC	☑	□	@3ZGW01@F8
0infoprov	STRING Chara...	□		InfoProvider	0INFOPROV	☑	□	@3ZGW01@F9

Picture 15: Analytic Index for HANA Procedure

In the next step we created the HAP with the generated HANA procedure.

Therefore we create a new HAP with the following settings and we use the new created Analytic Index as a data target.



Create SAP HANA Analysis Process YEPMHA06

SAP HANA Analysis Process: YEPMHA06 HAP with Procedure Output Analytic Index

Description: HAP with Procedure Output Analytic Index

InfoArea: YD048919 Gordon's Playground

Data Source

- ● InfoProvider: YEPMHA02 EPM SALES ORDER 2
- ○ Database Table
- ○ SAP HANA Analysis Process

Data Analysis

- ○ Function or Script
- ● Procedure: 0BW_OPER_PROC HANA Procedure
  - ☑ Generate Procedure
  - Schema: SAPHBW Package:
  - □ ABAP Managed Database Procedure
  - Name: 0BW:DAP:YEPMHA06~PN Missing procedure

Data Target

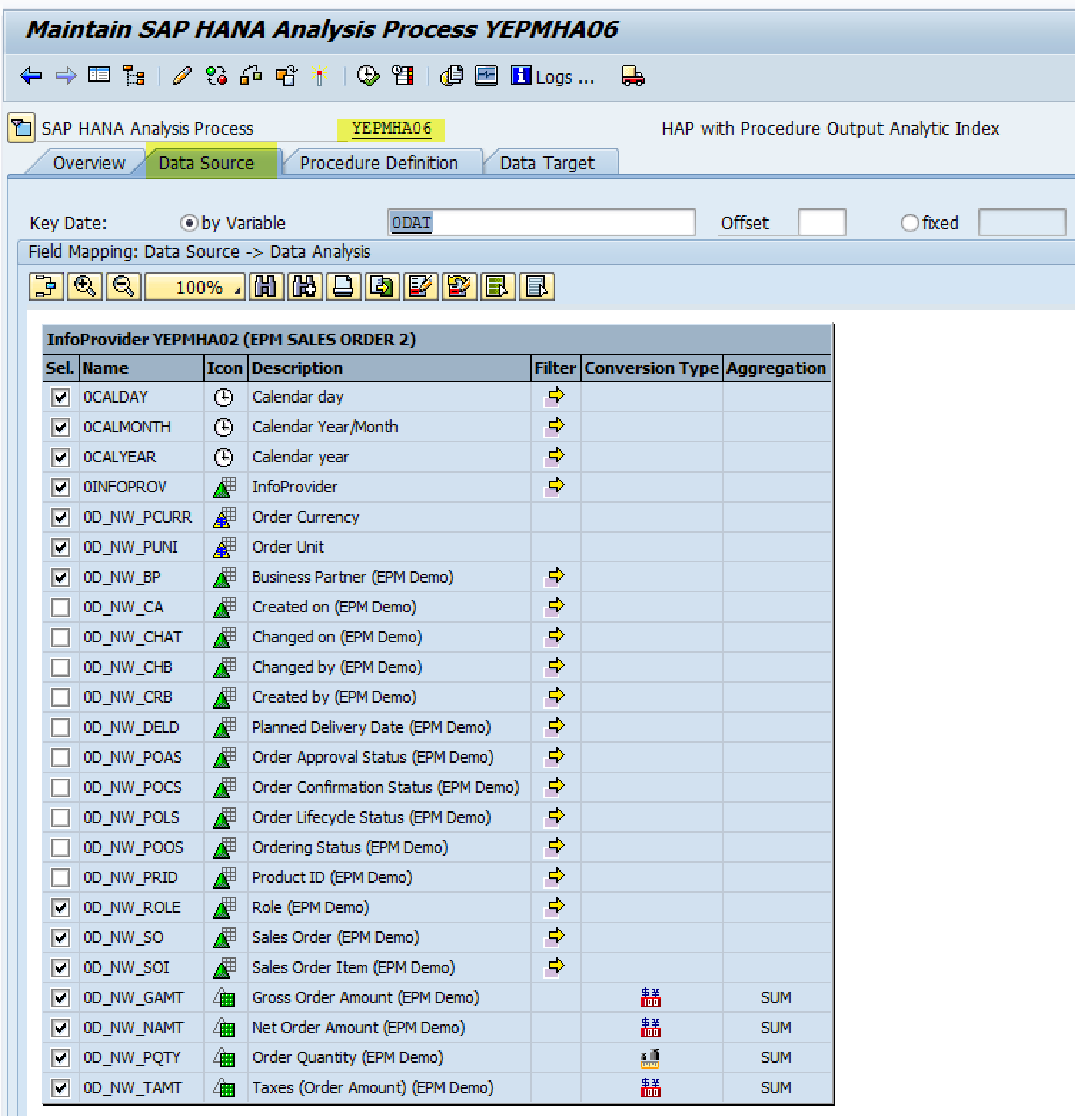
- ● Analytic Index: ZGW01 Index for HAP output based on a Procedure
  - □ Generate Analytic Index
  - ☑ Persist Result in Analytic Index
- ○ DataStore Object
- ○ Database Table
- ○ Embedded in Data Transfer Process

Picture 16: HANA Procedure generated

As you can see a new procedure will be generated and now the procedure needs to be defined.

First we have to choose the needed Input fields. These fields can be chosen from the Source – in our example from the underlying InfoProvider.

Please note: In the filter it is possible to restrict the values of the fields like in a simple where clause in SQL. If you set a filter this filter takes place before the procedure runs and restricts the input.



SAP HANA Analysis Process YEPMHA06 HAP with Procedure Output Analytic Index

Key Date: ● by Variable 0DAT Offset ○ fixed

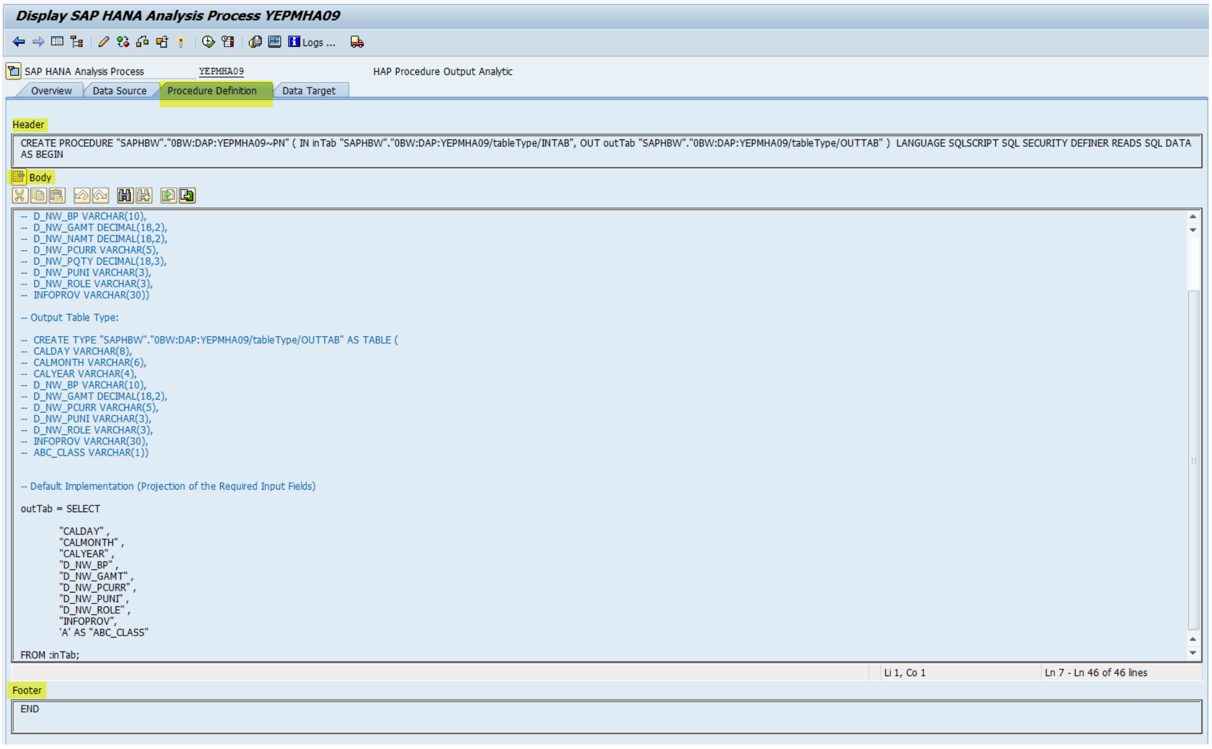
Field Mapping: Data Source -> Data Analysis

InfoProvider YEPMHA02 (EPM SALES ORDER 2)

Sel.	Name	Icon	Description	Filter	Conversion Type	Aggregation
☑	0CALDAY	◷	Calendar day	⇨		
☑	0CALMONTH	◷	Calendar Year/Month	⇨		
☑	0CALYEAR	◷	Calendar year	⇨		
☑	0INFOPROV	▦	InfoProvider	⇨		
☑	0D_NW_PCURR	▦	Order Currency			
☑	0D_NW_PUNI	▦	Order Unit			
☑	0D_NW_BP	▦	Business Partner (EPM Demo)	⇨		
□	0D_NW_CA	▦	Created on (EPM Demo)	⇨		
□	0D_NW_CHAT	▦	Changed on (EPM Demo)	⇨		
□	0D_NW_CHB	▦	Changed by (EPM Demo)	⇨		
□	0D_NW_CRB	▦	Created by (EPM Demo)	⇨		
□	0D_NW_DELD	▦	Planned Delivery Date (EPM Demo)	⇨		
□	0D_NW_POAS	▦	Order Approval Status (EPM Demo)	⇨		
□	0D_NW_POCS	▦	Order Confirmation Status (EPM Demo)	⇨		
□	0D_NW_POLS	▦	Order Lifecycle Status (EPM Demo)	⇨		
□	0D_NW_POOS	▦	Ordering Status (EPM Demo)	⇨		
□	0D_NW_PRID	▦	Product ID (EPM Demo)	⇨		
☑	0D_NW_ROLE	▦	Role (EPM Demo)	⇨		
☑	0D_NW_SO	▦	Sales Order (EPM Demo)	⇨		
☑	0D_NW_SOI	▦	Sales Order Item (EPM Demo)	⇨		
☑	0D_NW_GAMT	▤	Gross Order Amount (EPM Demo)		▥	SUM
☑	0D_NW_NAMT	▤	Net Order Amount (EPM Demo)		▥	SUM
☑	0D_NW_PQTY	▤	Order Quantity (EPM Demo)		▥	SUM
☑	0D_NW_TAMT	▤	Taxes (Order Amount) (EPM Demo)		▥	SUM

Picture 17: Input fields for generated Hana Procedure

The next step is the definition of the procedure is needed. In the generated procedure the header with the create statement and the table definition is done automatically. The body needs to be coded use case specific and you can use the output structure as the data target.



```
CREATE PROCEDURE "SAPHBW"."0BW:DAP:YEPMHA09~PN" ( IN inTab "SAPHBW"."0BW:DAP:YEPMHA09/tableType/INTAB", OUT outTab "SAPHBW"."0BW:DAP:YEPMHA09/tableType/OUTTAB" )  LANGUAGE SQLSCRIPT SQL SECURITY DEFINER READS SQL DATA
AS BEGIN
```

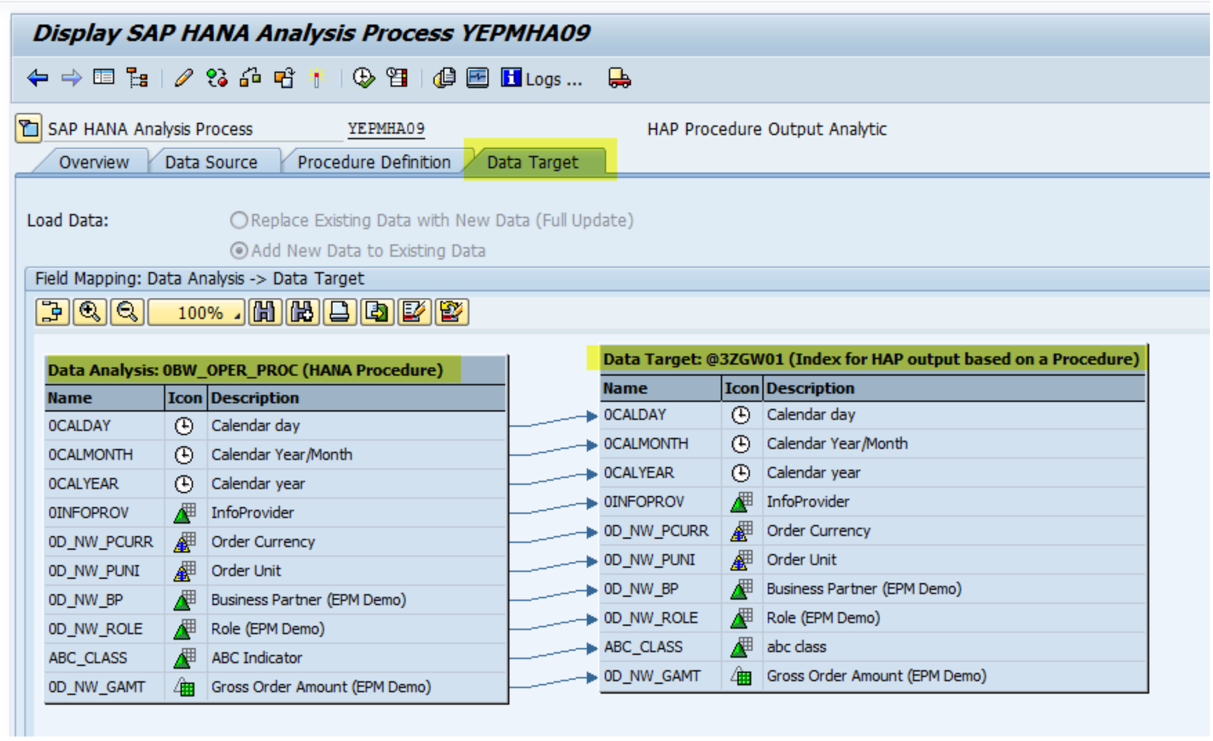
```
--  D_NW_BP VARCHAR(10),
--  D_NW_GAMT DECIMAL(18,2),
--  D_NW_NAMT DECIMAL(18,2),
--  D_NW_PCURR VARCHAR(5),
--  D_NW_PQTY DECIMAL(18,3),
--  D_NW_PUNI VARCHAR(3),
--  D_NW_ROLE VARCHAR(3),
--  INFOPROV VARCHAR(30))

-- Output Table Type:

--  CREATE TYPE "SAPHBW"."0BW:DAP:YEPMHA09/tableType/OUTTAB" AS TABLE (
--  CALDAY VARCHAR(8),
--  CALMONTH VARCHAR(6),
--  CALYEAR VARCHAR(4),
--  D_NW_BP VARCHAR(10),
--  D_NW_GAMT DECIMAL(18,2),
--  D_NW_PCURR VARCHAR(5),
--  D_NW_PUNI VARCHAR(3),
--  D_NW_ROLE VARCHAR(3),
--  INFOPROV VARCHAR(30),
--  ABC_CLASS VARCHAR(1))


-- Default Implementation (Projection of the Required Input Fields)

outTab = SELECT

        "CALDAY" ,
        "CALMONTH" ,
        "CALYEAR" ,
        "D_NW_BP" ,
        "D_NW_GAMT" ,
        "D_NW_PCURR" ,
        "D_NW_PUNI" ,
        "D_NW_ROLE" ,
        "INFOPROV",
        'A' AS "ABC_CLASS"

FROM :inTab;
```

```
END
```

Picture 18: Generated Hana Procedure

In the body the coding needs to be implemented. Here you have all possibilities to manipulate the data and send the data into the Output structure which is defined in the data target.

In the Data Target tab you can see the mapping between the Output structure defined in the Procedure definition and the fields of the data target. This view is only for demonstration of the internal mapping – there is no possibility to change the mapping or to add/delete fields from the target at the moment.



Display SAP HANA Analysis Process YEPMHA09

SAP HANA Analysis Process YEPMHA09 HAP Procedure Output Analytic

Load Data: ○ Replace Existing Data with New Data (Full Update)
● Add New Data to Existing Data

Field Mapping: Data Analysis -> Data Target

Data Analysis: 0BW_OPER_PROC (HANA Procedure)

Name	Icon	Description
0CALDAY		Calendar day
0CALMONTH		Calendar Year/Month
0CALYEAR		Calendar year
0INFOPROV		InfoProvider
0D_NW_PCURR		Order Currency
0D_NW_PUNI		Order Unit
0D_NW_BP		Business Partner (EPM Demo)
0D_NW_ROLE		Role (EPM Demo)
ABC_CLASS		ABC Indicator
0D_NW_GAMT		Gross Order Amount (EPM Demo)

Data Target: @3ZGW01 (Index for HAP output based on a Procedure)

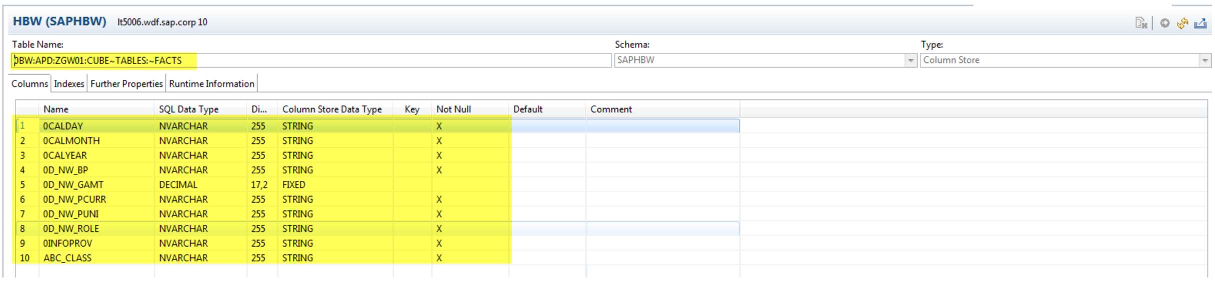
Name	Icon	Description
0CALDAY		Calendar day
0CALMONTH		Calendar Year/Month
0CALYEAR		Calendar year
0INFOPROV		InfoProvider
0D_NW_PCURR		Order Currency
0D_NW_PUNI		Order Unit
0D_NW_BP		Business Partner (EPM Demo)
0D_NW_ROLE		Role (EPM Demo)
ABC_CLASS		abc class
0D_NW_GAMT		Gross Order Amount (EPM Demo)

Picture 19: Data Target of a Generated Hana Procedure

Please note:

The sequence of the attributes in the select statement needs to be the in the same order of the fields in the underlying table from the output structure - in this case of the Analytic Index ZGW01.

Therefore it is recommended to search for the table in the Hana studio to ensure that the right sequence is used in the SQL statement of the procedure.



HBW (SAPHBW) lt5006.wdf.sap.corp 10

Table Name: 0BW:APD:ZGW01:CUBE~TABLES:~FACTS

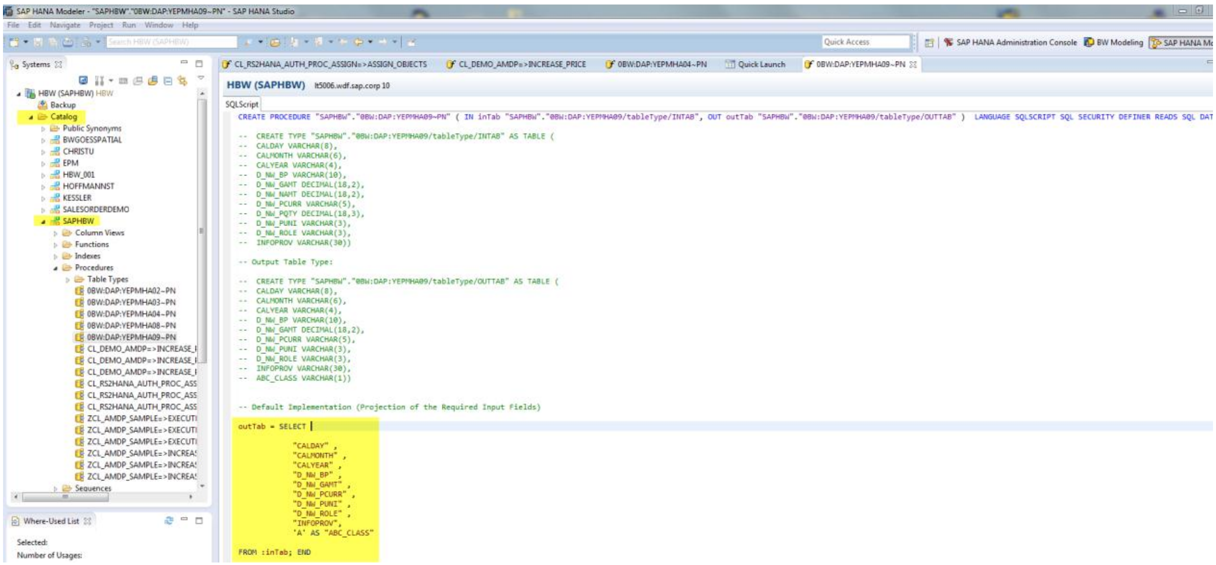
Schema: SAPHBW

Type: Column Store

	Name	SQL Data Type	Di...	Column Store Data Type	Key	Not Null	Default	Comment
1	0CALDAY	NVARCHAR	255	STRING		X		
2	0CALMONTH	NVARCHAR	255	STRING		X		
3	0CALYEAR	NVARCHAR	255	STRING		X		
4	0D_NW_BP	NVARCHAR	255	STRING		X		
5	0D_NW_GAMT	DECIMAL	17,2	FIXED				
6	0D_NW_PCURR	NVARCHAR	255	STRING		X		
7	0D_NW_PUNI	NVARCHAR	255	STRING		X		
8	0D_NW_ROLE	NVARCHAR	255	STRING		X		
9	0INFOPROV	NVARCHAR	255	STRING		X		
10	ABC_CLASS	NVARCHAR	255	STRING		X		

Picture 20: Table of the Analytic Index

The generated procedure for the HAP is also visible in the HANA studio in the schema defined during the definition of the HAP.

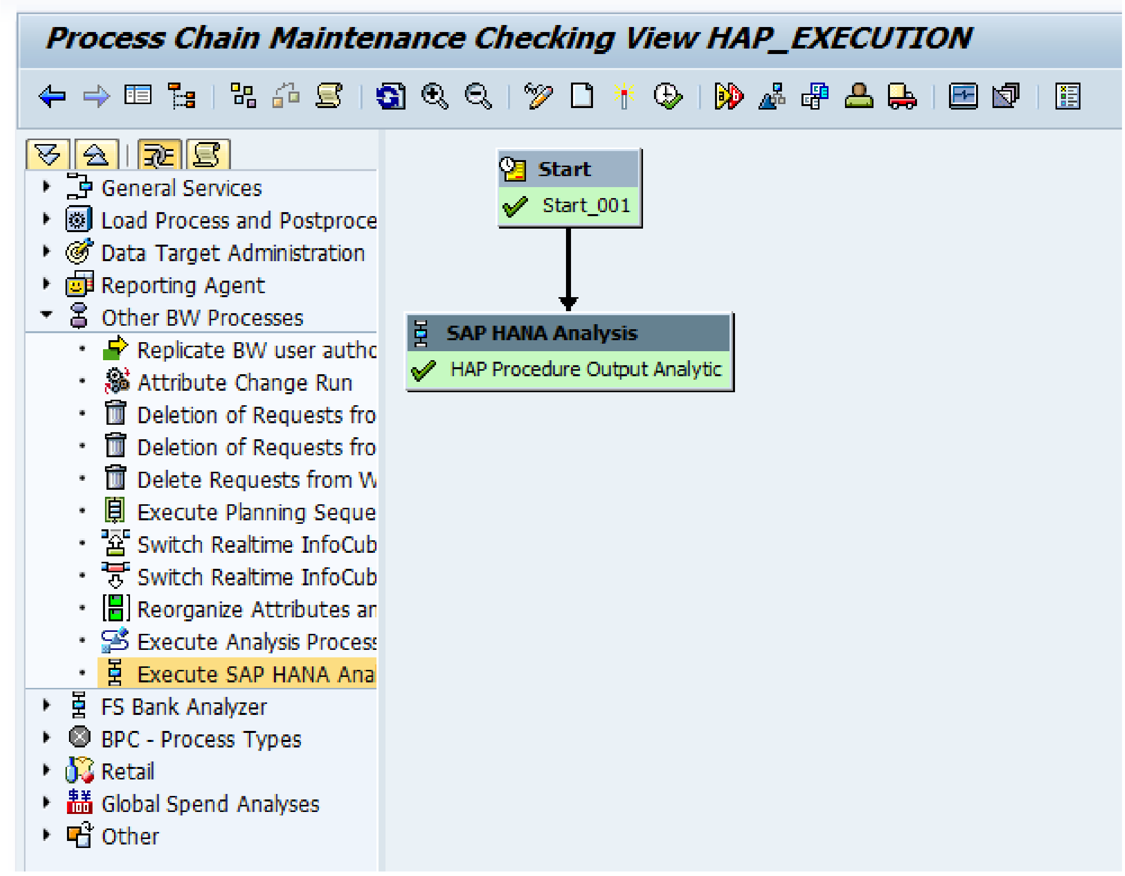


```
CREATE PROCEDURE "SAPHBW"."0BW:DAP:YEPMHA09~PN" ( IN inTab "SAPHBW"."0BW:DAP:YEPMHA09/tableType/INTAB", OUT outTab "SAPHBW"."0BW:DAP:YEPMHA09/tableType/OUTTAB" )  LANGUAGE SQLSCRIPT SQL SECURITY DEFINER READS SQL DAT

--  CREATE TYPE "SAPHBW"."0BW:DAP:YEPMHA09/tableType/INTAB" AS TABLE (
--  CALDAY VARCHAR(8),
--  CALMONTH VARCHAR(6),
--  CALYEAR VARCHAR(4),
--  D_NW_BP VARCHAR(10),
--  D_NW_GAMT DECIMAL(18,2),
--  D_NW_NAMT DECIMAL(18,2),
--  D_NW_PCURR VARCHAR(5),
--  D_NW_PQTY DECIMAL(18,3),
--  D_NW_PUNI VARCHAR(3),
--  D_NW_ROLE VARCHAR(3),
--  INFOPROV VARCHAR(30))

-- Output Table Type:

--  CREATE TYPE "SAPHBW"."0BW:DAP:YEPMHA09/tableType/OUTTAB" AS TABLE (
--  CALDAY VARCHAR(8),
--  CALMONTH VARCHAR(6),
--  CALYEAR VARCHAR(4),
--  D_NW_BP VARCHAR(10),
--  D_NW_GAMT DECIMAL(18,2),
--  D_NW_PCURR VARCHAR(5),
--  D_NW_PUNI VARCHAR(3),
--  D_NW_ROLE VARCHAR(3),
--  INFOPROV VARCHAR(30),
--  ABC_CLASS VARCHAR(1))


-- Default Implementation (Projection of the Required Input Fields)

outTab = SELECT

            "CALDAY" ,
            "CALMONTH" ,
            "CALYEAR" ,
            "D_NW_BP" ,
            "D_NW_GAMT" ,
            "D_NW_PCURR" ,
            "D_NW_PUNI" ,
            "D_NW_ROLE" ,
            "INFOPROV",
            'A' AS "ABC_CLASS"

FROM :inTab; END
```

SAP HANA Modeler - "SAPHBW"."0BW:DAP:YEPMHA09~PN" - SAP HANA Studio
Quick Access
SAP HANA Administration Console
BW Modeling
CL_RS2HANA_AUTH_PROC_ASSIGN=>ASSIGN_OBJECTS
CL_DEMO_AMDP=>INCREASE_PRICE
0BW:DAP:YEPMHA04~PN
Quick Launch
0BW:DAP:YEPMHA09~PN
HBW (SAPHBW) lt5006.wdf.sap.corp 10
SQLScript
Systems
Catalog
SAPHBW
Procedures
Where-Used List

Picture 21: Generated Hana Procedure in the HANA Studio

In the next step the HAP with the embedded procedure is defined and can be executed directly in BW or executed via a process chain.



Picture 22: HAP in Process Chain

The advantage of the Generated Hana Procedure is that the input and output variables automatically assigned and the developer needs to be concerned only about the content/body of the procedure and therefore the development of this procedure is faster compare to the non-generated procedure or the ABAP Managed Database Procedure.

A disadvantage is at the moment that the structure of the input and output structure is very limited and it is not possible in the moment to add or remove some fields in the output structure.

Another disadvantage is that the generated procedure cannot be transported in the BW standard framework – there is no option yet to transport the Hana Content via the BW standard transportation mechanism. In this case you have to create a separate Hana transport containing the code of these Hana procedures.

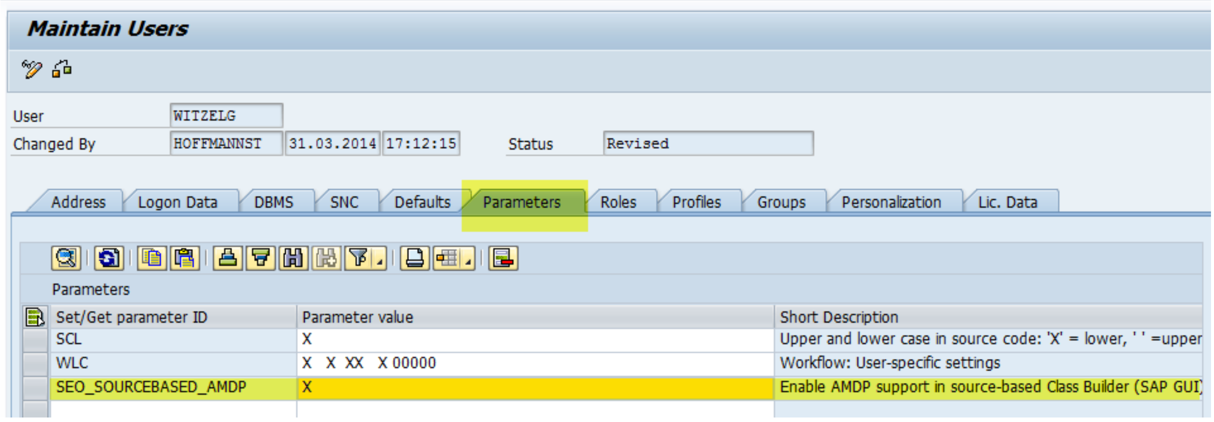
6.2 ABAP Managed Database Procedure

An ABAP Managed Database Procedure (AMDP) is a classed based framework which offers a top down approach for managing and calling Hana procedures in ABAP and is implemented as a method of a global class marked with specific interfaces. The corresponding database procedure is created at the first call of the AMDP method.

The main benefit of this approach is that only the AMDP class has to be transported with the standard ABAP transport mechanism and no Hana delivery unit and no Hana transport container is needed.

ABAP Managed Database Procedures can be created in the ABAP Eclipse environment or in the standard transactions se80 or se24 but only when a special user parameter is set.

To set this special user parameter it is recommended to use TA SU01. Therefore enter the name of the required user, press ‘change’ and go to the tab ‘Parameters’ and add the parameter SEO_SOURCEBASED_AMDP.



Maintain Users

User WITZELG

Changed By HOFFMANNST 31.03.2014 17:12:15 Status Revised

Parameters

Set/Get parameter ID	Parameter value	Short Description
SCL	X	Upper and lower case in source code: 'X' = lower, ' ' =upper
WLC	X X XX X 00000	Workflow: User-specific settings
SEO_SOURCEBASED_AMDP	X	Enable AMDP support in source-based Class Builder (SAP GUI)

Picture 23: Prerequisite for ABAP Managed Database Procedures

It is now possible to open se80 or se24 to create new classes.

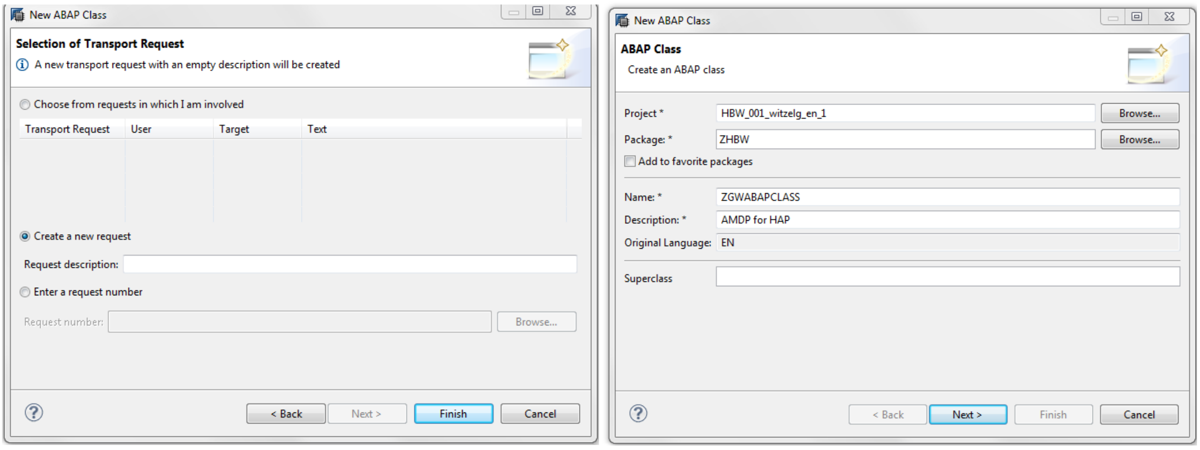
In the next step the AMDP will be created in a step by step guidance.

6.2.1 Create an AMDP Class

1. Start the ABAP Development Tools (aka ABAP in Eclipse) - or your SAP HANA Studio - and open the ABAP perspective by selecting menu entry Window > Open perspective > Others..., and choosing the ABAP entry in the appearing dialog box.

2. Go to your ABAP project in the Project Explorer and create a new class in the package of your choice by selecting the context menu entry New... > ABAP Class

3. Maintain the required information and click on Next.
   Select a transport request if required and confirm the creation dialog.



Selection of Transport Request

A new transport request with an empty description will be created

○ Choose from requests in which I am involved

Transport Request	User	Target	Text

● Create a new request

Request description:

○ Enter a request number

Request number: Browse...

ABAP Class

Create an ABAP class

Project *: HBW_001_witzelg_en_1 Browse...

Package: *: ZHBW Browse...

□ Add to favorite packages

Name: *: ZGWABAPCLASS

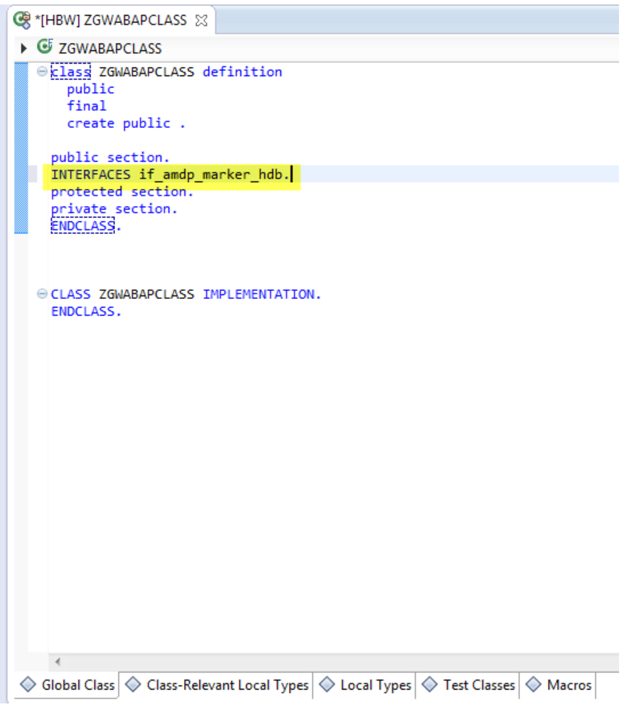
Description: *: AMDP for HAP

Original Language: EN

Superclass:

Picture 24: Create a new Abap Class

4. Go to the class definition and insert following line directly after PUBLIC SECTION:
   INTERFACESif_amdp_marker_hdb.



```
class ZGWABAPCLASS definition
  public
  final
  create public .

public section.
INTERFACES if_amdp_marker_hdb.
protected section.
private section.
ENDCLASS.


CLASS ZGWABAPCLASS IMPLEMENTATION.
ENDCLASS.
```

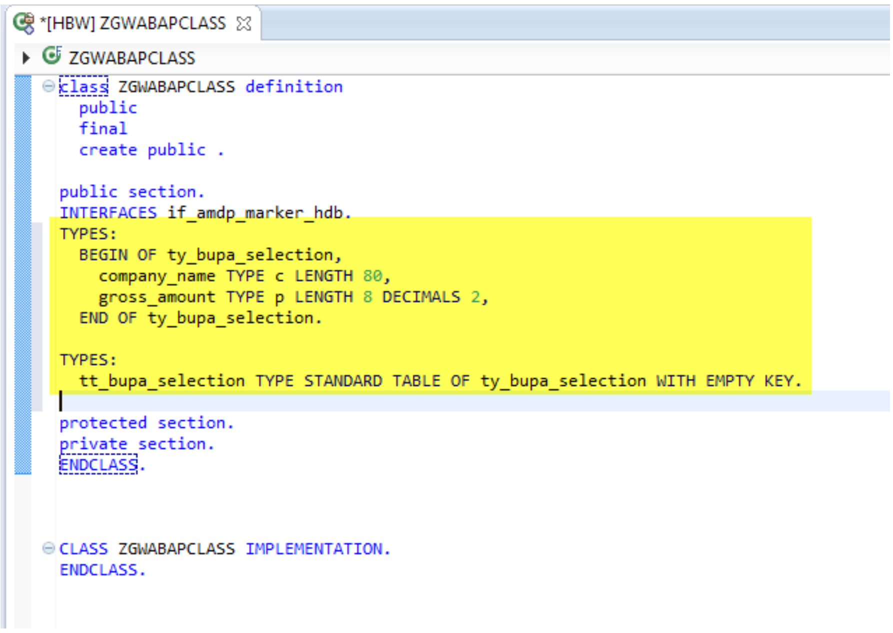
Picture 25: Insert the Tag Interface for Hana

5. Save.

6.2.2 Declare an AMDP Method

Now the method get_top_and_flop which will be implemented as AMDP method later on needs to be declared. This method has an importing parameter for specifying the number of customers to be retrieved for each category (top and flop), and two exporting parameter getting for the two requested result sets.

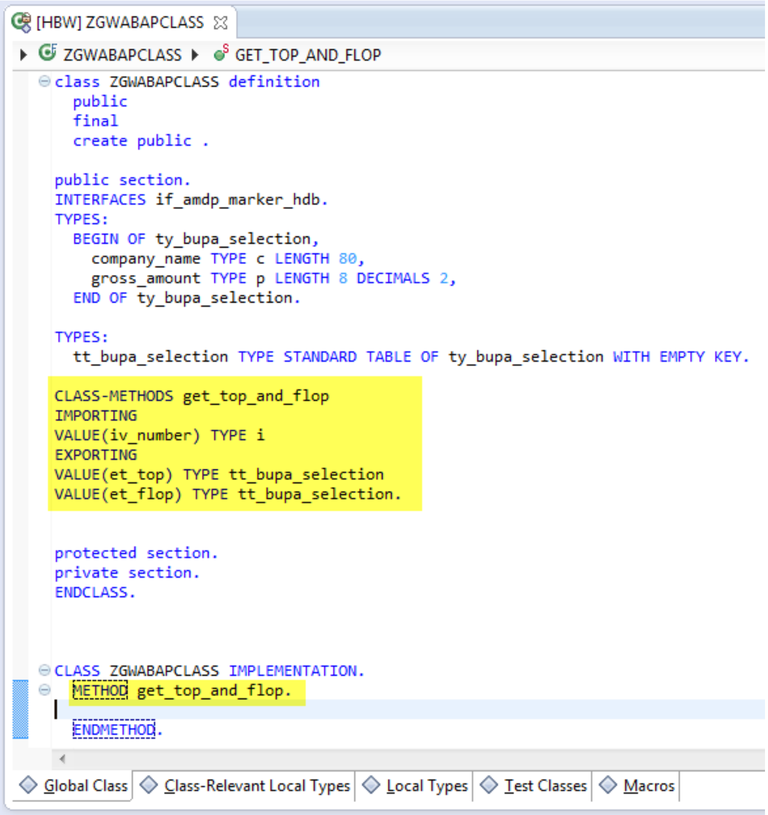
1. Define the two types ty_bupa_selection and tt_bupa_selection in the PUBLIC SECTION of the class definition - With ty_bupa_selection defining the table line of our return set, and tt_bupa_selection defining the type of the returned tables.



```
class ZGWABAPCLASS definition
  public
  final
  create public .

public section.
INTERFACES if_amdp_marker_hdb.
TYPES:
  BEGIN OF ty_bupa_selection,
    company_name TYPE c LENGTH 80,
    gross_amount TYPE p LENGTH 8 DECIMALS 2,
  END OF ty_bupa_selection.

TYPES:
  tt_bupa_selection TYPE STANDARD TABLE OF ty_bupa_selection WITH EMPTY KEY.

protected section.
private section.
ENDCLASS.


CLASS ZGWABAPCLASS IMPLEMENTATION.
ENDCLASS.
```

Picture 26: Declare an AMDP Method

2. Now define the public static method get_top_and_flop.



```
class ZGWABAPCLASS definition
  public
  final
  create public .

public section.
INTERFACES if_amdp_marker_hdb.
TYPES:
  BEGIN OF ty_bupa_selection,
    company_name TYPE c LENGTH 80,
    gross_amount TYPE p LENGTH 8 DECIMALS 2,
  END OF ty_bupa_selection.

TYPES:
  tt_bupa_selection TYPE STANDARD TABLE OF ty_bupa_selection WITH EMPTY KEY.

CLASS-METHODS get_top_and_flop
IMPORTING
VALUE(iv_number) TYPE i
EXPORTING
VALUE(et_top) TYPE tt_bupa_selection
VALUE(et_flop) TYPE tt_bupa_selection.


protected section.
private section.
ENDCLASS.



CLASS ZGWABAPCLASS IMPLEMENTATION.
  METHOD get_top_and_flop.

  ENDMETHOD.
```

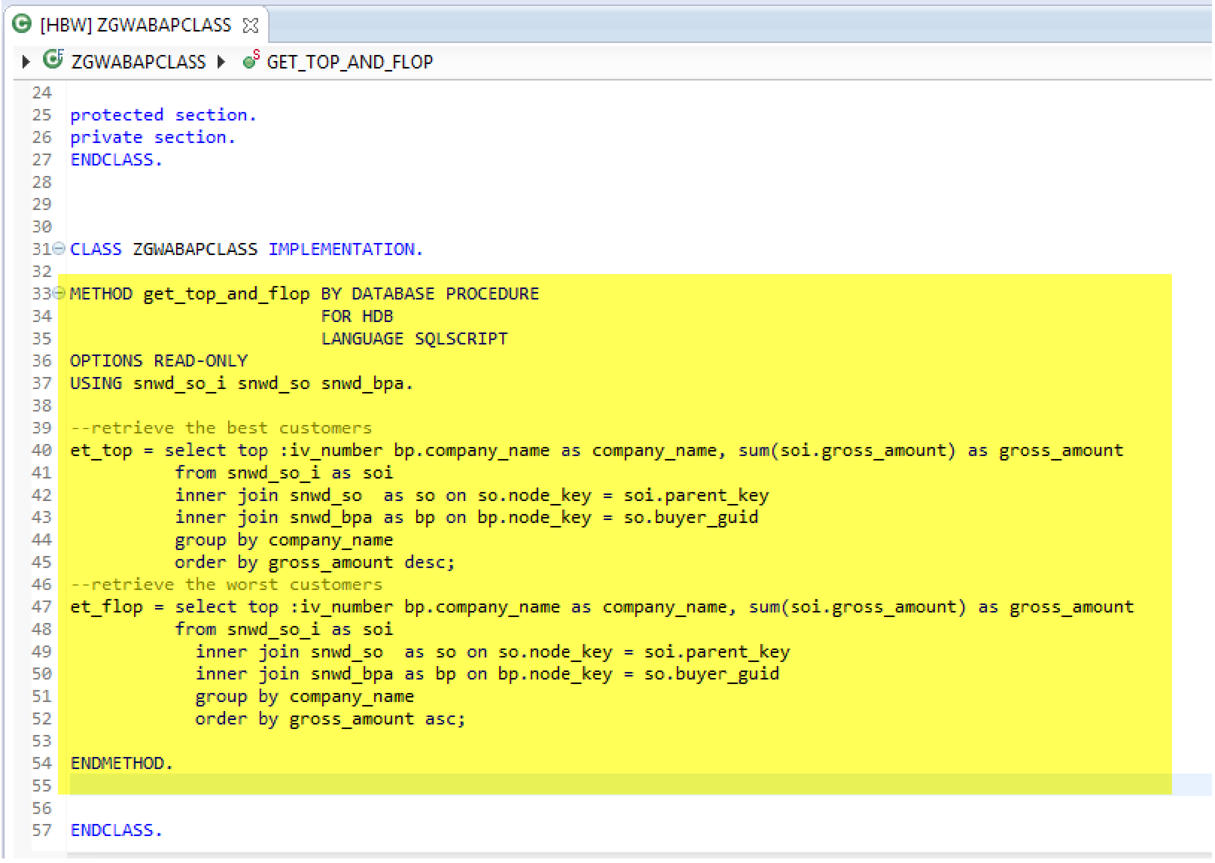
Picture 27: Insert the AMDP Method

3. An error will be displayed in the editor due to the missing method implementation. Just use the Quick Fix (Ctrl+1) function “Add implementation for get_top_and_flop” to quickly solved this issue.
4. Save your AMDP class.

6.2.3 Implement the AMDP Method

Now implement a relatively simple SQLScript-based AMDP method, which retrieves the best and worst customers depending on the gross amount of open invoices.

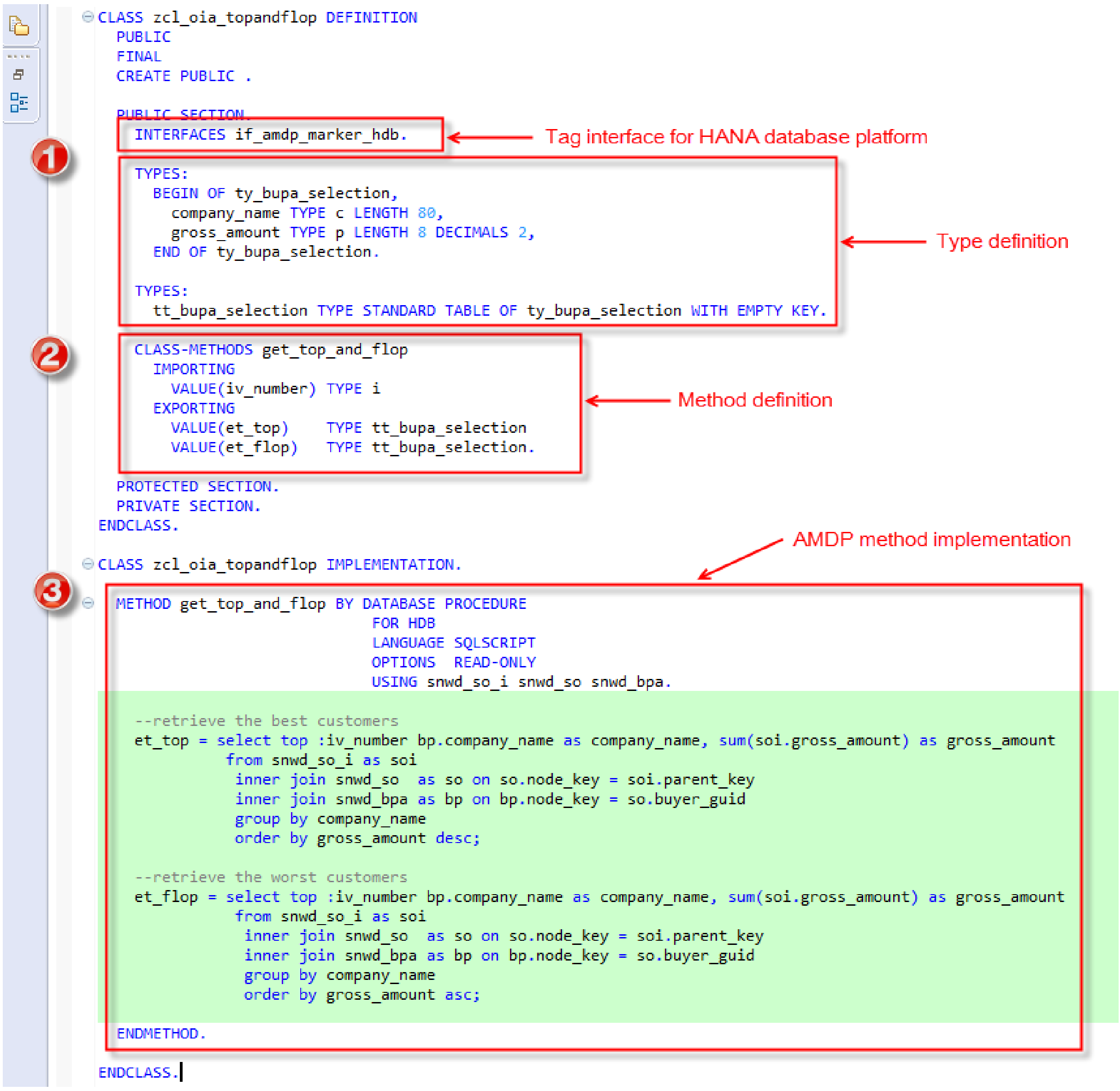
1. Mark the method implementation as an AMDP method and implement the database procedure by developing an SQLScript .



```

protected section.
private section.
ENDCLASS.



CLASS ZGWABAPCLASS IMPLEMENTATION.

METHOD get_top_and_flop BY DATABASE PROCEDURE
                        FOR HDB
                        LANGUAGE SQLSCRIPT
OPTIONS READ-ONLY
USING snwd_so_i snwd_so snwd_bpa.

--retrieve the best customers
et_top = select top :iv_number bp.company_name as company_name, sum(soi.gross_amount) as gross_amount
          from snwd_so_i as soi
          inner join snwd_so  as so on so.node_key = soi.parent_key
          inner join snwd_bpa as bp on bp.node_key = so.buyer_guid
          group by company_name
          order by gross_amount desc;
--retrieve the worst customers
et_flop = select top :iv_number bp.company_name as company_name, sum(soi.gross_amount) as gross_amount
          from snwd_so_i as soi
            inner join snwd_so  as so on so.node_key = soi.parent_key
            inner join snwd_bpa as bp on bp.node_key = so.buyer_guid
            group by company_name
            order by gross_amount asc;

ENDMETHOD.


ENDCLASS.
```

Picture 28: Implement the AMDP Method

2. Save and Activate the Procedure.
3. The AMDP method is developed and ready to use in the HAP.



```
CLASS zcl_oia_topandflop DEFINITION
  PUBLIC
  FINAL
  CREATE PUBLIC .

  PUBLIC SECTION.
    INTERFACES if_amdp_marker_hdb.

    TYPES:
      BEGIN OF ty_bupa_selection,
        company_name TYPE c LENGTH 80,
        gross_amount TYPE p LENGTH 8 DECIMALS 2,
      END OF ty_bupa_selection.

    TYPES:
      tt_bupa_selection TYPE STANDARD TABLE OF ty_bupa_selection WITH EMPTY KEY.

    CLASS-METHODS get_top_and_flop
      IMPORTING
        VALUE(iv_number) TYPE i
      EXPORTING
        VALUE(et_top)    TYPE tt_bupa_selection
        VALUE(et_flop)   TYPE tt_bupa_selection.

  PROTECTED SECTION.
  PRIVATE SECTION.
ENDCLASS.

CLASS zcl_oia_topandflop IMPLEMENTATION.

  METHOD get_top_and_flop BY DATABASE PROCEDURE
                             FOR HDB
                             LANGUAGE SQLSCRIPT
                             OPTIONS  READ-ONLY
                             USING snwd_so_i snwd_so snwd_bpa.

    --retrieve the best customers
    et_top = select top :iv_number bp.company_name as company_name, sum(soi.gross_amount) as gross_amount
              from snwd_so_i as soi
               inner join snwd_so  as so on so.node_key = soi.parent_key
               inner join snwd_bpa as bp on bp.node_key = so.buyer_guid
               group by company_name
               order by gross_amount desc;

    --retrieve the worst customers
    et_flop = select top :iv_number bp.company_name as company_name, sum(soi.gross_amount) as gross_amount
               from snwd_so_i as soi
                inner join snwd_so  as so on so.node_key = soi.parent_key
                inner join snwd_bpa as bp on bp.node_key = so.buyer_guid
                group by company_name
                order by gross_amount asc;

  ENDMETHOD.

ENDCLASS.
```

Tag interface for HANA database platform

Type definition

Method definition

AMDP method implementation

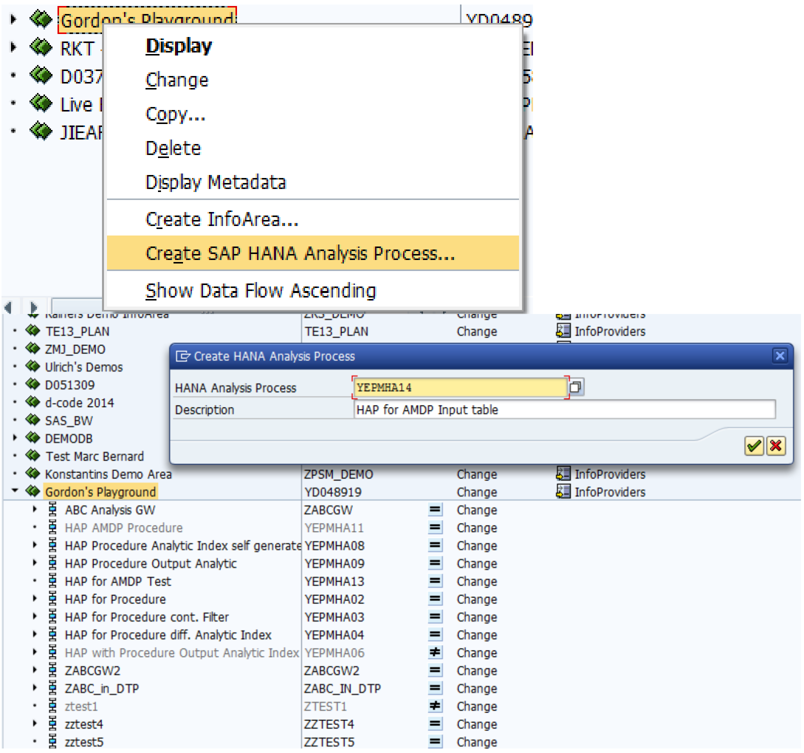
Picture 29: Overview of the AMDP Method

6.2.4 Creating a HAP and consuming the AMDP Method

Due to the fact, that HAP are not working with scalar Input or Output parameters in the moment a slightly different AMDP is used in this example. (2017624 - Procedures with scalar input and/or output parameters are not supported)

In this example the input parameter is defined as a table and the Select statement in the SQL script is hardcoded just to demonstrate the workflow and the possibilities of this HAP.

1. Create a new HAP.



HANA Analysis Process: YEPMHA14

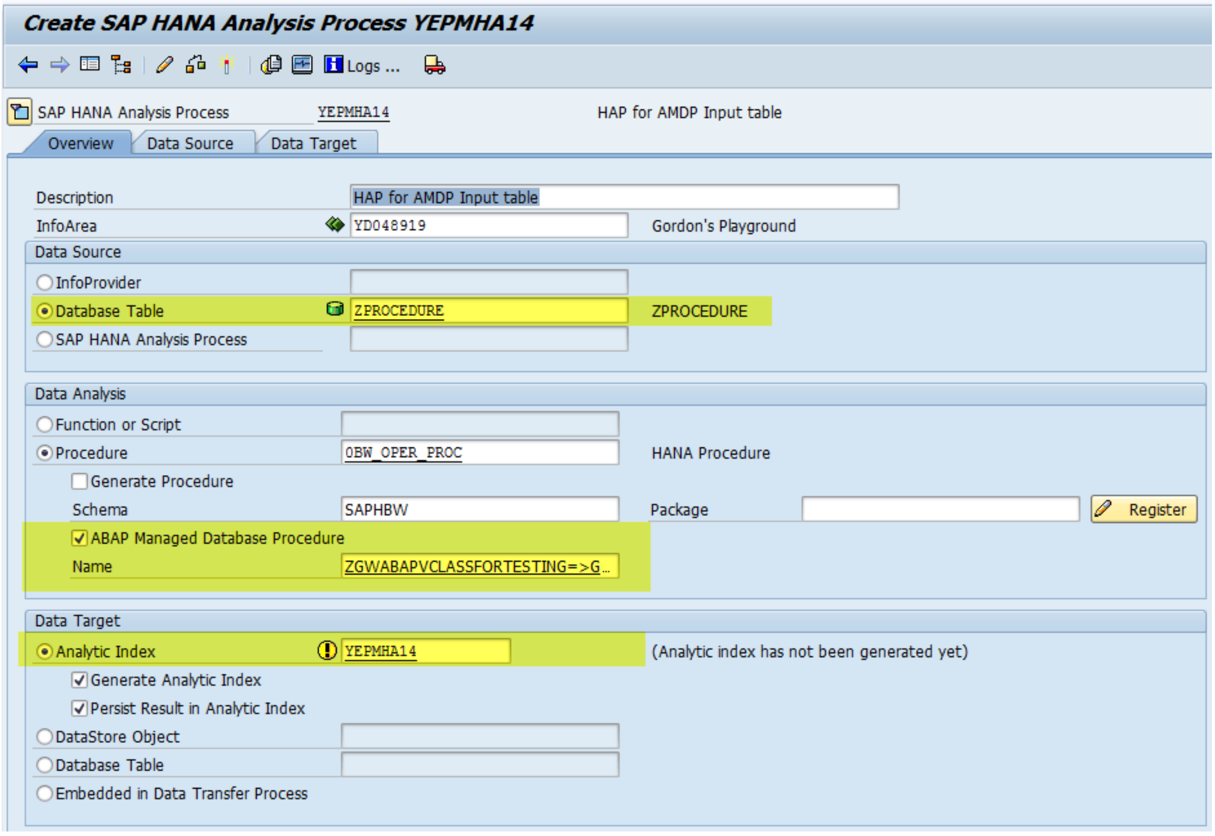
Description: HAP for AMDP Input table

TE13_PLAN	TE13_PLAN		Change	InfoProviders
ZMJ_DEMO				
Ulrich's Demos				
D051309				
d-code 2014				
SAS_BW				
DEMODB				
Test Marc Bernard				
Konstantins Demo Area	ZPSM_DEMO		Change	InfoProviders
Gordon's Playground	YD048919		Change	InfoProviders
ABC Analysis GW	ZABCGW	=	Change	
HAP AMDP Procedure	YEPMHA11	=	Change	
HAP Procedure Analytic Index self generate	YEPMHA08	=	Change	
HAP Procedure Output Analytic	YEPMHA09	=	Change	
HAP for AMDP Test	YEPMHA13	=	Change	
HAP for Procedure	YEPMHA02	=	Change	
HAP for Procedure cont. Filter	YEPMHA03	=	Change	
HAP for Procedure diff. Analytic Index	YEPMHA04	=	Change	
HAP with Procedure Output Analytic Index	YEPMHA06	≠	Change	
ZABCGW2	ZABCGW2	=	Change	
ZABC_in_DTP	ZABC_IN_DTP	=	Change	
ztest1	ZTEST1	≠	Change	
zztest4	ZZTEST4	=	Change	
zztest5	ZZTEST5	=	Change	

Picture 30: Create a HAP based on a AMDP

2. Design the HAP – use table ZPROCEDURE as a Source. In this simple example only one value is posted in this table – this value acts as an input parameter for the AMDP and defines the number of the best and worst customers depending on the gross amount of open invoices.

The Output is defined as an Analytic Index. The Analytic Index will be generated once the HAP was activated successfully.



Create SAP HANA Analysis Process YEPMHA14

SAP HANA Analysis Process YEPMHA14 HAP for AMDP Input table

Description: HAP for AMDP Input table
InfoArea: YD048919 Gordon's Playground

Data Source
- ○ InfoProvider
- ● Database Table: ZPROCEDURE ZPROCEDURE
- ○ SAP HANA Analysis Process

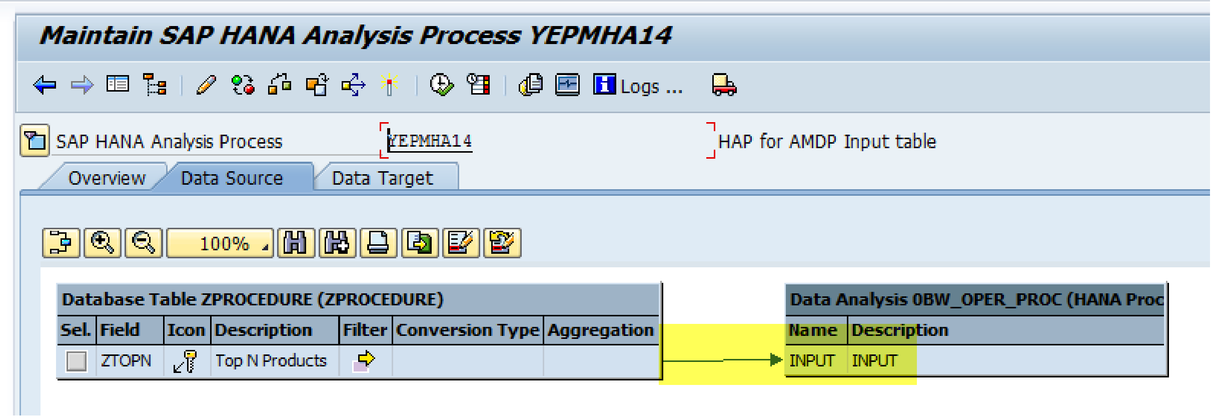
Data Analysis
- ○ Function or Script
- ● Procedure: 0BW_OPER_PROC HANA Procedure
- □ Generate Procedure
- Schema: SAPHBW Package:
- ☑ ABAP Managed Database Procedure
- Name: ZGWABAPVCLASSFORTESTING=>G...

Data Target
- ● Analytic Index: YEPMHA14 (Analytic index has not been generated yet)
- ☑ Generate Analytic Index
- ☑ Persist Result in Analytic Index
- ○ DataStore Object
- ○ Database Table
- ○ Embedded in Data Transfer Process

Picture 31: Define the HAP based on a AMDP

3. Save the existing one.
4. Map the Input parameter of the Input Source to the Input fields of the Hana procedure.

In this example we map the number of the number of the best and worst customers depending on the gross amount of open invoices from table ZPROCEDURE (field ZTOPN) to the input parameters from the HANA procedure which is a AMDP in this example.



Maintain SAP HANA Analysis Process YEPMHA14

SAP HANA Analysis Process YEPMHA14 HAP for AMDP Input table

Database Table ZPROCEDURE (ZPROCEDURE)

Sel.	Field	Icon	Description	Filter	Conversion Type	Aggregation
□	ZTOPN		Top N Products			

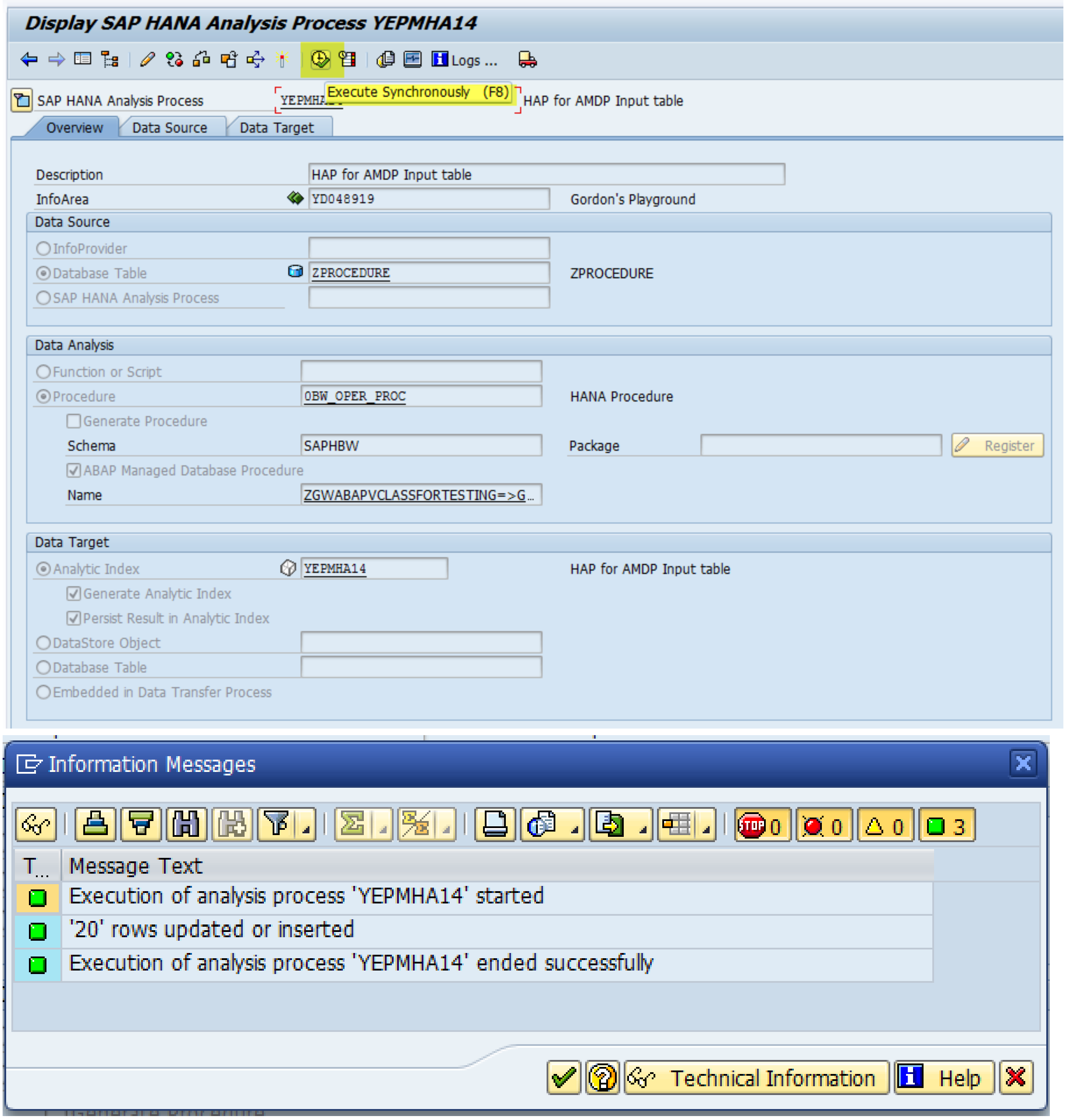
Data Analysis 0BW_OPER_PROC (HANA Proc

Name	Description
INPUT	INPUT

Picture 32: Mapping the HAP based on a AMDP

5. Save and activate the HAP.

6. Execute the HAP by pressing execute or F8.



Display SAP HANA Analysis Process YEPMHA14

SAP HANA Analysis Process YEPMHA14 HAP for AMDP Input table

Description: HAP for AMDP Input table

InfoArea: YD048919 Gordon's Playground

Data Source

○ InfoProvider

● Database Table: ZPROCEDURE ZPROCEDURE

○ SAP HANA Analysis Process

Data Analysis

○ Function or Script

● Procedure: 0BW_OPER_PROC HANA Procedure

□ Generate Procedure

Schema: SAPHBW Package: Register

☑ ABAP Managed Database Procedure

Name: ZGWABAPVCLASSFORTESTING=>G...

Data Target

● Analytic Index: YEPMHA14 HAP for AMDP Input table

☑ Generate Analytic Index

☑ Persist Result in Analytic Index

○ DataStore Object

○ Database Table

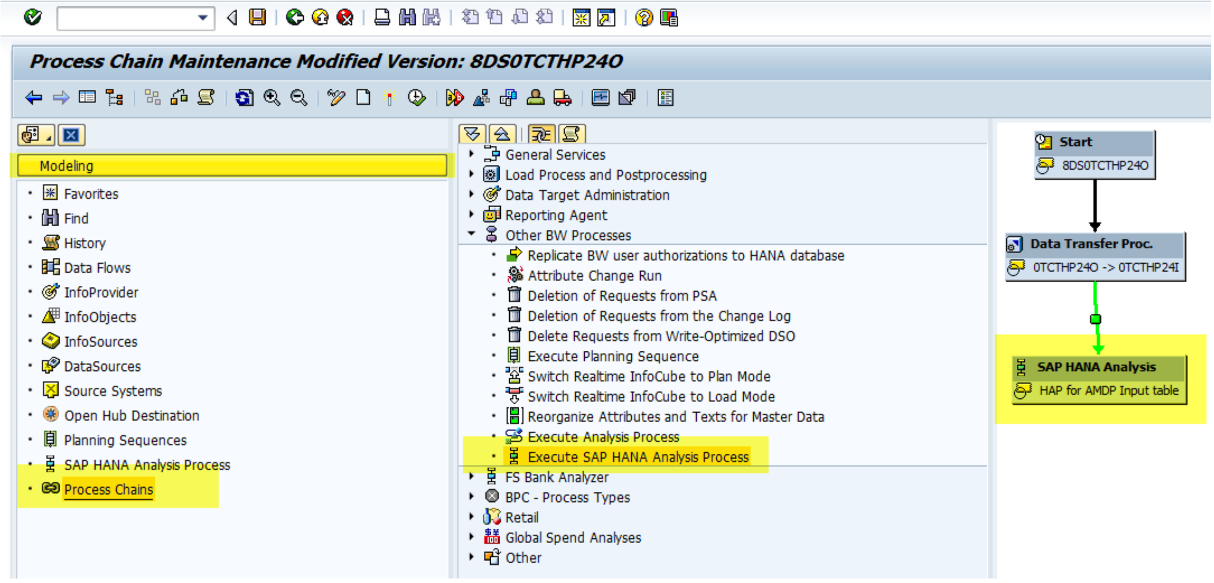
○ Embedded in Data Transfer Process

Information Messages

T...	Message Text
🟩	Execution of analysis process 'YEPMHA14' started
🟩	'20' rows updated or inserted
🟩	Execution of analysis process 'YEPMHA14' ended successfully

Picture 33: Execution of the HAP based on a AMDP

Now it is possible to integrate this HAP in any process chain to schedule the execution.


Process Chain Maintenance Modified Version: 8DSOTCTHP240
Modeling
Favorites
Find
History
Data Flows
InfoProvider
InfoObjects
InfoSources
DataSources
Source Systems
Open Hub Destination
Planning Sequences
SAP HANA Analysis Process
Process Chains
General Services
Load Process and Postprocessing
Data Target Administration
Reporting Agent
Other BW Processes
Replicate BW user authorizations to HANA database
Attribute Change Run
Deletion of Requests from PSA
Deletion of Requests from the Change Log
Delete Requests from Write-Optimized DSO
Execute Planning Sequence
Switch Realtime InfoCube to Plan Mode
Switch Realtime InfoCube to Load Mode
Reorganize Attributes and Texts for Master Data
Execute Analysis Process
Execute SAP HANA Analysis Process
FS Bank Analyzer
BPC - Process Types
Retail
Global Spend Analyses
Other
Start
8DSOTCTHP240
Data Transfer Proc.
0TCTHP240 -> 0TCTHP24I
SAP HANA Analysis
HAP for AMDP Input table

Picture 34: Execution of the HAP in a process chain

For a full introduction into SQL-Script it is recommend having a look at the HANA development guide

(http://help.sap.com/saphelp_hana/sap_hana_developer_guide_en.pdf)

and the HANA SQLScript reference (http://help.sap.com/hana/SAP_HANA_SQL_Script_Reference_en.pdf).

The big advantage of the AMDP based HAP is the fact that those procedures can be transported via the standard ABAP transport mechanisms in BW and no Hana artefacts needs to be created and transported manually.

Of course there are some known limitations in AMDP at the moment:

Limitations in the Procedure Definition

- Naming.
  AMDP creates DB procedures with a special naming convention which is derived from the ABAP definition. It is not possible to create DB procedures with a procedure name or parameter names. Also the usage of camel case is not possible.
- Usable data types.
  The HANA type system is not completely covered by the ABAP stack. Since the procedure definition is derived from ABAP types, it is not possible to create DB procedures that contain HANA data types which are not supported by the ABAP stack, i.e. DATE, TIME, SECONDDATE, TIMESTAMP, TINYINT, BIGINT, SMALLDECIMAL, DECIMAL, REAL, VARCHAR, ALPHANUM, SHORTTEXT, TEXT.
- Result views.
  Database procedures can be defined with a result view. This concept is not supported with AMDP and will later be supported with the support of table functions.

Limitations in the Procedure Code

- No DDL Statements allowed
  DDL statements for creation of DB tables or views in the default schema are not supported. It is recommended to use the ordinary DDIC functionality instead.
- No transaction control statements allowed.
  The SQL statements COMMIT and ROLLBACK are (currently) not supported directly inside SQLScript database procedures.

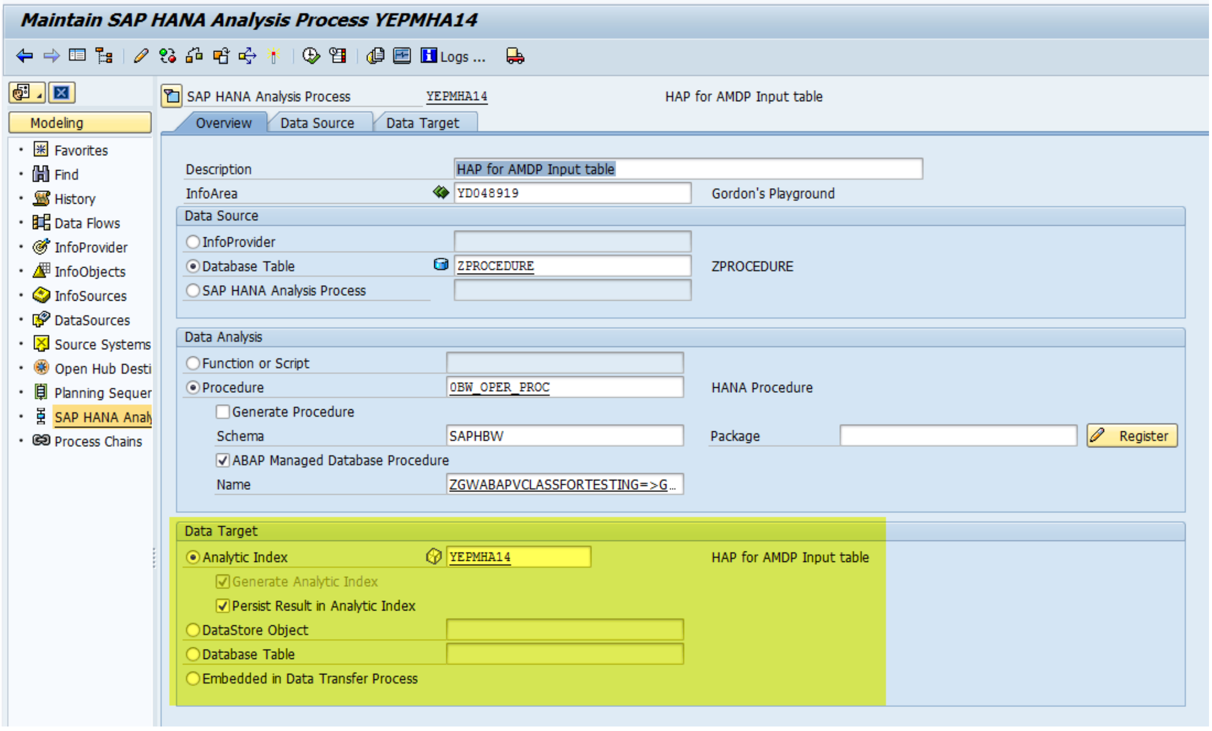
It is still possible to issue a commit via the EXEC statement. It has to be made clear, that the transaction management should be done from ABAP to ensure transactional consistency (e.g. triggering of follow up processing like update tasks, etc.).

- Only access to ABAP managed DB artifacts in default schema.
  Within AMDPs only DB objects of the own/default schema can be accessed, that are ABAP managed, i.e. AMDPs or DDIC tables/views/CDS views.
- Some calc engine functions (e.g. CE_COLUMN_TABLE) are not usable.
  The calc engine function CE_COLUMN_TABLE only works on column store tables and not on views. This is a major problem, since AMDPs or LM tools like SUM/DMO replace the table access with a view-based access.
  All calc engine functions which cannot operate on views are not supported. This is the case for CE_COLUMN_TABLE, which can simply be substituted with a SQL statement.
- No DB changes of buffered tables.
  Updates to buffered tables are not supported, since the table buffer would become out of sync.
- Only ASCII printable characters allowed.
  The usage of non ASCII printable characters in the source code has indicated several problems with the syntax-check, error positioning and procedure generation. It is strongly recommended to avoid the problematic characters. The first occurrence of such a character is indicated with a warning.

7. Output for HANA Analysis Processes

It is possible to choose a data target for each HAP – the possible options are:

- Analytic Index
- Data Store Object
- Database Table
- Embedded in DTP



Maintain SAP HANA Analysis Process YEPMHA14

SAP HANA Analysis Process YEPMHA14 HAP for AMDP Input table

Description: HAP for AMDP Input table

InfoArea: YD048919 Gordon's Playground

Data Source

○ InfoProvider

● Database Table: ZPROCEDURE ZPROCEDURE

○ SAP HANA Analysis Process

Data Analysis

○ Function or Script

● Procedure: 0BW_OPER_PROC HANA Procedure

□ Generate Procedure

Schema: SAPHBW Package:

☑ ABAP Managed Database Procedure

Name: ZGWABAPVCLASSFORTESTING=>G...

Data Target

● Analytic Index: YEPMHA14 HAP for AMDP Input table

☑ Generate Analytic Index

☑ Persist Result in Analytic Index

○ DataStore Object

○ Database Table

○ Embedded in Data Transfer Process

Picture 35: data target options for HAP

If you choose the Analytic Index it is possible to create a new one or to use an existing one. If you choose a new one, the flag for Generate Analytic Index will set automatically and the Analytic Index will be created during the activation of the HAP.

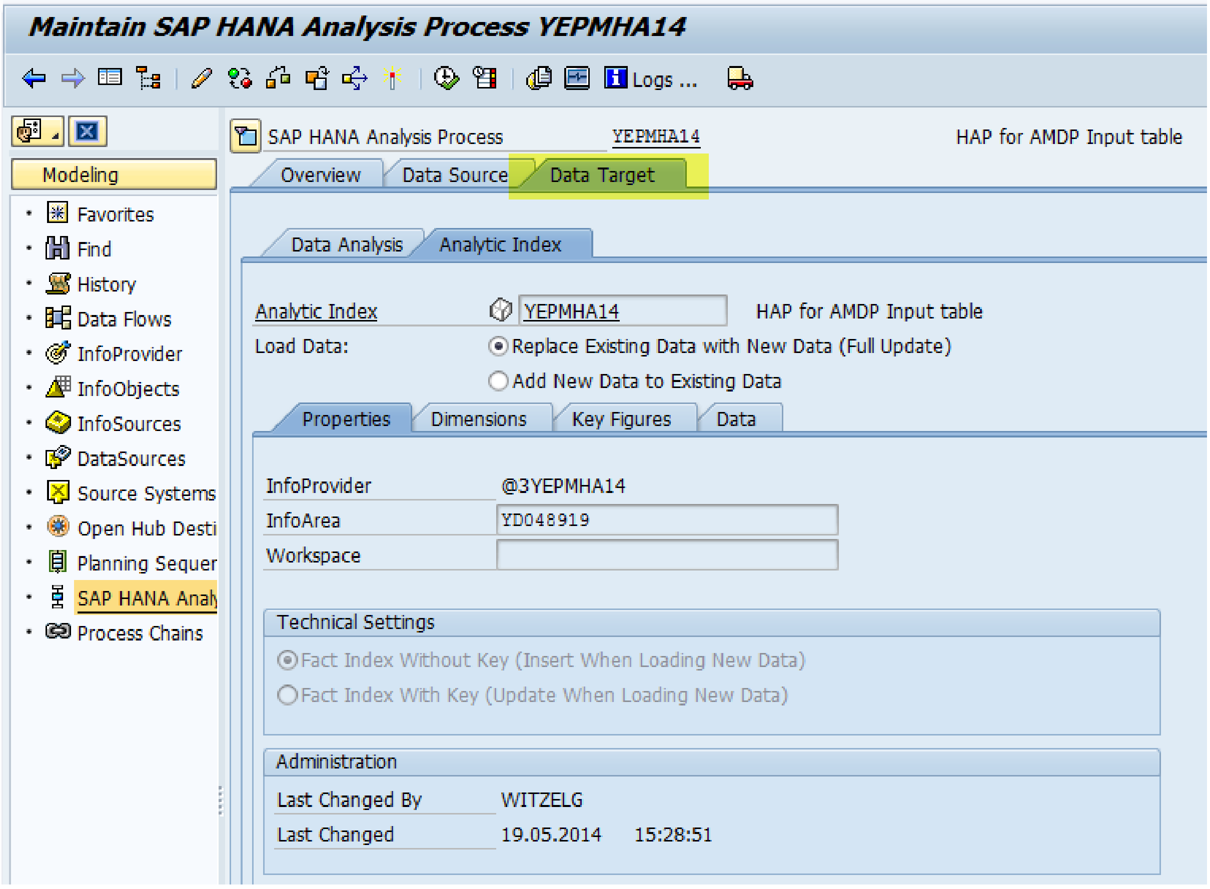
Furthermore it is possible to persist the results in the Analytic Index.

Another option for a data target is a Data Store Object – in this case only direct update DSOs are supported and can be used as a data target for HAP.

A database table is also possible as a target – here you can use each table which is known in the ABAP dictionary.

If you choose the option to use the HAP in a DTP then the option Embedded in Data Transfer Process should be used. It is now possible to use the HAP in a DTP. If you use this option, the HAP it is not possible to execute the HAP – you have to execute this HAP embedded in a DTP.

On the Data Target tab page it is possible to maintain the loading behavior settings and edit the data target.



Maintain SAP HANA Analysis Process YEPMHA14

SAP HANA Analysis Process YEPMHA14 HAP for AMDP Input table

Analytic Index: YEPMHA14 HAP for AMDP Input table

Load Data: ● Replace Existing Data with New Data (Full Update)
○ Add New Data to Existing Data

InfoProvider	@3YEPMHA14
InfoArea	YD048919
Workspace	

Technical Settings

● Fact Index Without Key (Insert When Loading New Data)
○ Fact Index With Key (Update When Loading New Data)

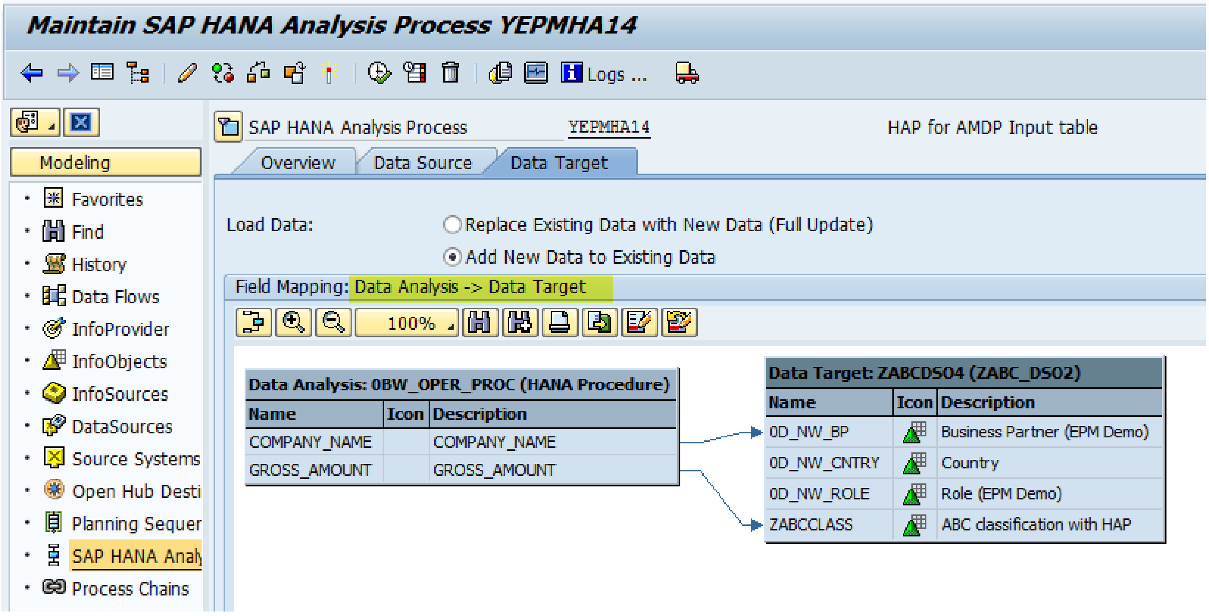
Administration

Last Changed By	WITZELG
Last Changed	19.05.2014 15:28:51

Picture 36: data target tab for HAP

If the Data Target is an Analytic Index it is possible to maintain the Analytic Index as well – there are the same possibilities like in TA RSDD_LTIP.

If you choose a DSO or a Database table as a target it is necessary to map the fields from the data analysis to the data target.



Maintain SAP HANA Analysis Process YEPMHA14

SAP HANA Analysis Process YEPMHA14 HAP for AMDP Input table

Load Data: ○ Replace Existing Data with New Data (Full Update)
● Add New Data to Existing Data

Field Mapping: Data Analysis -> Data Target

Data Analysis: 0BW_OPER_PROC (HANA Procedure)

Name	Icon	Description
COMPANY_NAME		COMPANY_NAME
GROSS_AMOUNT		GROSS_AMOUNT

Data Target: ZABCDSO4 (ZABC_DSO2)

Name	Icon	Description
0D_NW_BP		Business Partner (EPM Demo)
0D_NW_CNTRY		Country
0D_NW_ROLE		Role (EPM Demo)
ZABCCLASS		ABC classification with HAP

Picture 37: data target mapping

For all data targets it is required to select the required loading behavior – there are 2 options:

- Overwrite Existing Data with New Data
- Add New Data to Existing Data

8. Summary

The HAP offers a great opportunity to use Predictive Analytics or Data Mining functionalities based on BW data with the power of SAP HANA.

It is possible to analyze the data using different functions or scripts and HANA provides numerous specialized libraries like PAL, AFL, R to understand the correlation of the data in the existing EDW and accelerate the usage of the consolidated EWD data for data mining.

By using a SAP HANA database it is recommended to use the SAP HANA Analysis Process instead of Analysis Process Designer (APD). The process is then performed exclusively in SAP HANA, thus improving performance dramatically.

The main difference in functionality between the SAP HANA analysis process and APD is that only InfoProviders that a column view exists for on the SAP HANA database are supported as data sources.

In the future more and more algorithm will be developed for the usage of the HAP and the importance of the APD will decrease.

9. Appendix

9.1 Analytic Index as a HAP Source

An analytic index is a data container whose data is stored in the SAP HANA database.

The analytic index stores its data in a star schema and is an InfoProvider that a BEx query can be executed on. There is no need for modelling or data provisioning of this InfoProvider. Since it is created and filled according the definition process.

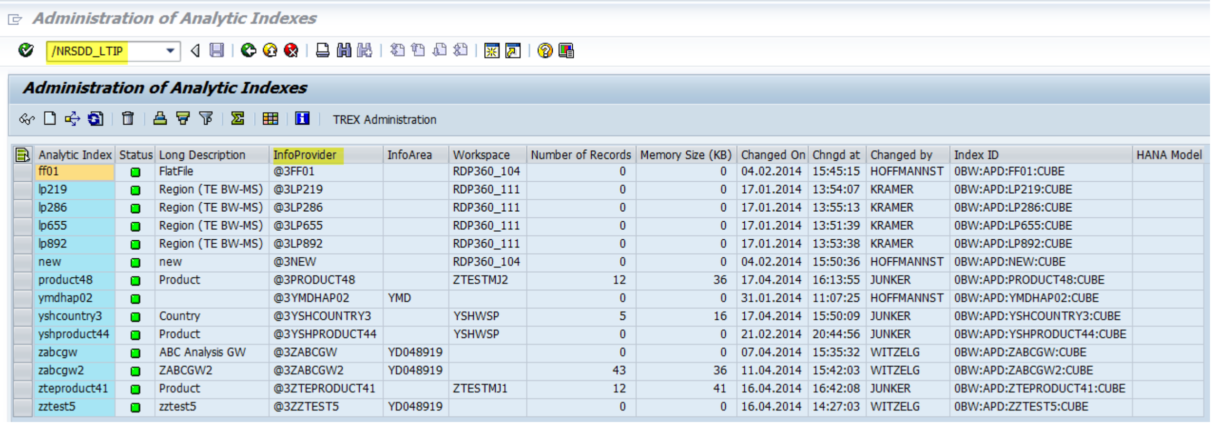
Analytic indexes can be created and filled with (transformed) data quickly. They are intended for ad-hoc scenarios. They can also be created as InfoProviders without reference to InfoObjects. They are therefore not integrated into the metadata repository and cannot be transported.

Analytic indexes that are generated by SAP HANA models being published to the BW system cannot be used as data targets in Analysis Process Designer. The data is not stored in the analytic index in this case. It simply represents a view of the data contained in SAP HANA and for analysis based on this Provider Type a Hana Analysis Process is strongly recommended.

Furthermore a backup and restore process is not implemented for analytic indexes. To store (persist) the data on a long-term basis a different InfoProvider type should be selected.

When creating an Analytic Index, a new InfoProvider in the '@3'-namespace is derived and can directly be used for reporting scenarios. This '@3'-namespace prevents naming conflicts and shows the difference between ad-hoc and standard scenarios.

To maintain an analytic index the transaction RSDD_LTIP in the BW backend needs to be called.



Administration of Analytic Indexes

Analytic Index	Status	Long Description	InfoProvider	InfoArea	Workspace	Number of Records	Memory Size (KB)	Changed On	Chngd at	Changed by	Index ID	HANA Model
ff01	🟩	FlatFile	@3FF01		RDP360_104	0	0	04.02.2014	15:45:15	HOFFMANNST	0BW:APD:FF01:CUBE	
lp219	🟩	Region (TE BW-MS)	@3LP219		RDP360_111	0	0	17.01.2014	13:54:07	KRAMER	0BW:APD:LP219:CUBE	
lp286	🟩	Region (TE BW-MS)	@3LP286		RDP360_111	0	0	17.01.2014	13:55:13	KRAMER	0BW:APD:LP286:CUBE	
lp655	🟩	Region (TE BW-MS)	@3LP655		RDP360_111	0	0	17.01.2014	13:51:39	KRAMER	0BW:APD:LP655:CUBE	
lp892	🟩	Region (TE BW-MS)	@3LP892		RDP360_111	0	0	17.01.2014	13:53:38	KRAMER	0BW:APD:LP892:CUBE	
new	🟩	new	@3NEW		RDP360_104	0	0	04.02.2014	15:50:36	HOFFMANNST	0BW:APD:NEW:CUBE	
product48	🟩	Product	@3PRODUCT48		ZTESTMJ2	12	36	17.04.2014	16:13:55	JUNKER	0BW:APD:PRODUCT48:CUBE	
ymdhap02	🟩		@3YMDHAP02	YMD		0	0	31.01.2014	11:07:25	HOFFMANNST	0BW:APD:YMDHAP02:CUBE	
yshcountry3	🟩	Country	@3YSHCOUNTRY3		YSHWSP	5	16	17.04.2014	15:50:09	JUNKER	0BW:APD:YSHCOUNTRY3:CUBE	
yshproduct44	🟩	Product	@3YSHPRODUCT44		YSHWSP	0	0	21.02.2014	20:44:56	JUNKER	0BW:APD:YSHPRODUCT44:CUBE	
zabcgw	🟩	ABC Analysis GW	@3ZABCGW	YD048919		0	0	07.04.2014	15:35:32	WITZELG	0BW:APD:ZABCGW:CUBE	
zabcgw2	🟩	ZABCGW2	@3ZABCGW2	YD048919		43	36	11.04.2014	15:42:03	WITZELG	0BW:APD:ZABCGW2:CUBE	
zteproduct41	🟩	Product	@3ZTEPRODUCT41		ZTESTMJ1	12	41	16.04.2014	16:42:08	JUNKER	0BW:APD:ZTEPRODUCT41:CUBE	
zztest5	🟩	zztest5	@3ZZTEST5	YD048919		0	0	16.04.2014	14:27:03	WITZELG	0BW:APD:ZZTEST5:CUBE	

Picture 38: Analytic Index Maintenance

More documentation can be found here:

http://help.sap.com/saphelp_nwes72/helpdata/EN/8d/9bdcb2b5e04772b727baaa1405e10f/content.htm

9.2 Composite Provider as a HAP Source

A CompositeProvider is an InfoProvider that combines data from several analytic indexes or from other InfoProviders (by union or inner join or left outer join) and makes this data available for reporting and analysis. The used union and join operations in the CompositeProvider are pushed down to HANA leveraging the power of the calculation engine inside HANA.

A CompositeProvider can only be transported if it consists solely of InfoProviders and it is not possible to use non-cumulative key figures.

For real-time ABAP Managed Database Procedure s, the open request is not read in the query, even the corresponding data integrity is setting in the Query Designer.

The CompositeProvider consolidates the number of InfoProvider types and harmonizes the modeling of mixed BW on HANA scenarios and therefore this InfoProvider is the recommended Provider Type for the future.

Please note:

In order to use the Eclipse based BW modeling for developing objects like the Composite Provider you need to install the BW modeling tools (see http://scn.sap.com/community/netweaver-bw-hana/blog/2014/01/29/sap-bw-modelling-tools for details).

For more information please read the following document:

http://www.sdn.sap.com/irj/scn/index?rid=/library/uuid/a0c5d872-fb76-3110-f288-bf714be4a406&overridelayout=true

9.3 DataStore Object as a HAP Source

A DataStore object serves as a storage location for consolidated and cleansed transaction data or master data at document (atomic) level. This data can be evaluated using a BEx query.

A DataStore object contains key fields and data fields that can also contain character fields as key figures. The data in 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.

The cumulative update of key figures is supported for DataStore objects, just as it is with InfoCubes, but with DataStore objects it is also possible to overwrite data fields. This is particularly important with document-related structures.

It is possible to create the following DataStore objects:

- Standard
- Write-optimized
- DataStore objects for direct update
- Semantically partitioned

9.4 InfoCube as a HAP Source

An InfoCube is a type of InfoProvider and describes a closed data set (from an analysis point of view) a self-contained dataset, for example, a business-orientated area.
An InfoCube is comprised of a set of relational tables arranged according to the enhanced star schema: a large fact table in the middle surrounded by several dimension tables.
The InfoCube receives the data by means of a data transfer process. It is then available as an InfoProvider for analysis and reporting purposes.

The SAP HANA-optimized InfoCube is a standard InfoCube that is optimized for use with SAP HANA. When you create SAP HANA-optimized InfoCubes, you can assign characteristics and key figures to dimensions. The system does not create any dimension tables apart from the package dimension however. The SIDs (master data IDs) are written directly to the fact table. This improves system performance when loading data. Since dimensions are omitted, no DIM IDs (dimensions keys) have to be created.

For more information please read the following documents:

Aspects of the HANA-optimized InfoCube:

http://www.saphana.com/servlet/JiveServlet/previewBody/1363-102-2-1810/SDN_HANA_opt_InfoCube%20FINAL.PDF
For Inventory InfoCubes:
http://www.sdn.sap.com/irj/scn/go/portal/prtroot/docs/library/uuid/50df1d69-4d86-2f10-a2ac-8b95b98dbe85?QuickLink=index&overridelayout=true&58965606021293

9.5 InfoObjects as a HAP Source

Business evaluation objects are known in BW as InfoObjects. They are divide into characteristics (for example, customers), key figures (for example, revenue), units (for example, currency, amount unit), time characteristics (for example, fiscal year) and technical characteristics (for example, request number).

InfoObjects are the smallest information units in BW. They structure the information needed to create data targets.

InfoObjects with attributes or texts can be either a pure data target or an InfoProvider (if it is being reported).

9.6 Multi Provider as a HAP Source

A MultiProvider is a type of InfoProvider that combines data from a number of InfoProviders and makes it available for reporting purposes. The MultiProvider does not itself contain any data. Its data comes entirely from the InfoProviders on which it is based. These InfoProviders are connected to one another by a union operation.

InfoProviders and MultiProviders are the objects or views that are relevant for reporting.

Please note:

The new CompositeProvider is an InfoProvider type introduced with BW7.40.

It offers the possibility to combine InfoProviders via UNION, INNER JOIN and LEFT OUTER JOIN.

A CompositeProvider always has a corresponding HANA ColumnView that contains the modeled JOIN and UNION operations. Thus JOIN and UNION are executed by the HANA CalcEngine, which offers a much better performance compared to the application server.

9.7 Query Snapshot as a HAP Source

On the Query Properties screen in Query Monitor, you can stipulate whether the selected query should be used as an InfoProvider.

The advantage in doing this is supported to calculate the query's functions in the Analytic Server before reading the data. This improves performance, especially in the case of simple queries that contain a very large amount of data.

The following are three scenarios for using a query as an InfoProvider:

•It is possible to use a query as the data source for an analysis process and use SAP BW Analysis Process Designer to create analytical indexes on it.

• It is possible to use a query as the data source of a data transfer process with extraction mode Full. This allows to extract query data and to distribute it using an Open Hub Destination to downstream applications, to a SAP HANA database for example. The usage of OLAP functions in the staging process with the query as the data source of a data transfer process will also provided.

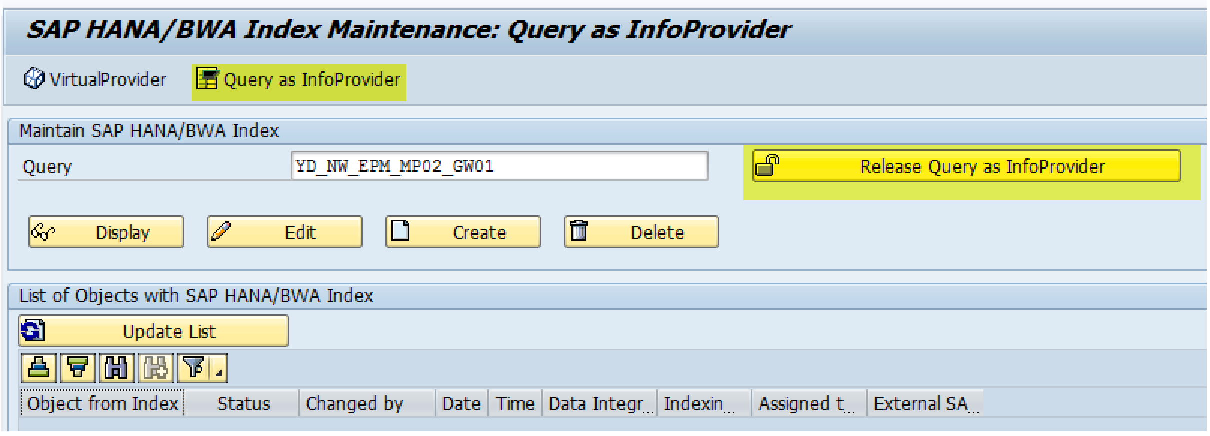
• It is possible to use a query - especially if it contains mass data -over the Analytic Server's relational interface as a data source for the SAP BusinessObjects Data Federator and perform analyses with this, for example in SAP BusinessObjects Web Intelligence.

More information's are documented here:

http://help.sap.com/saphelp_nw73/helpdata/en/4b/c347cd494650e9e10000000a15822b/content.htm

There are different ways to create and fill a Query Snapshot and use the Query Snapshot as an InfoProvider.

One way is to use transaction RSDDB and create a Query Snapshot by using the Release Query as InfoProvider button.



SAP HANA/BWA Index Maintenance: Query as InfoProvider

Maintain SAP HANA/BWA Index

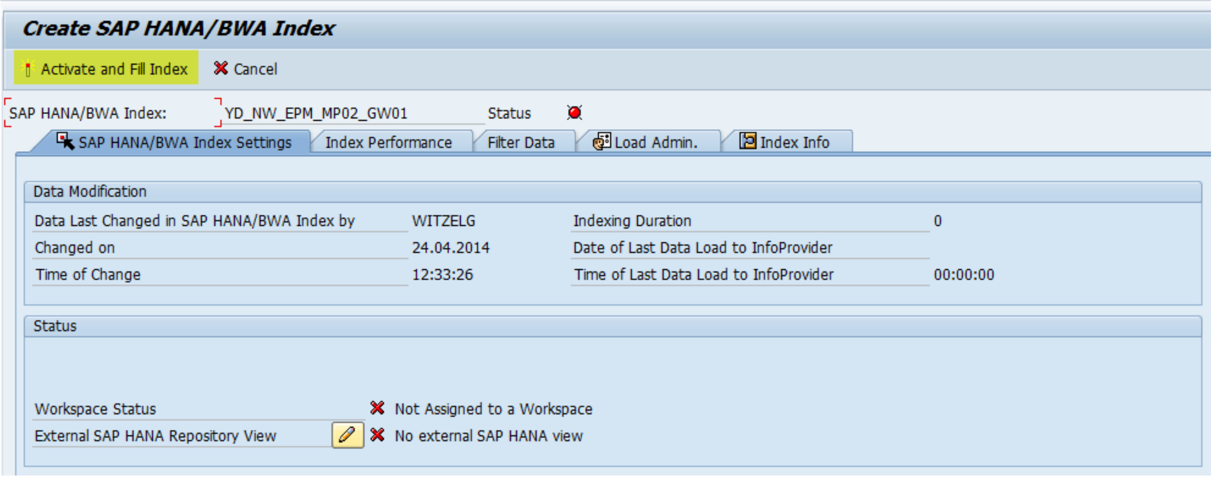
Query: YD_NW_EPM_MP02_GW01

List of Objects with SAP HANA/BWA Index

Object from Index	Status	Changed by	Date	Time	Data Integr...	Indexin...	Assigned t...	External SA...

Picture 39: Query as InfoProvider RSDDB

After this step create (press the create button in RSDDB) the Query Snapshot and fill the index.



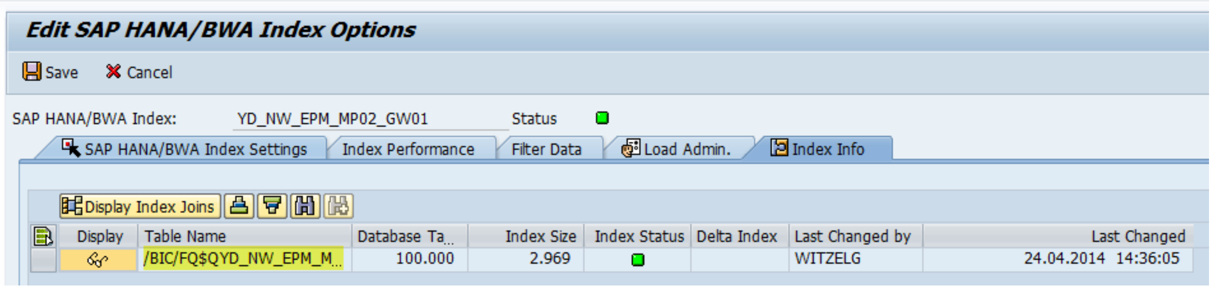
SAP HANA/BWA Index: YD_NW_EPM_MP02_GW01 Status ●

Data Modification

Data Last Changed in SAP HANA/BWA Index by	WITZELG	Indexing Duration	0
Changed on	24.04.2014	Date of Last Data Load to InfoProvider	
Time of Change	12:33:26	Time of Last Data Load to InfoProvider	00:00:00

Status

Workspace Status	✖ Not Assigned to a Workspace
External SAP HANA Repository View	✖ No external SAP HANA view



SAP HANA/BWA Index: YD_NW_EPM_MP02_GW01 Status 🟩

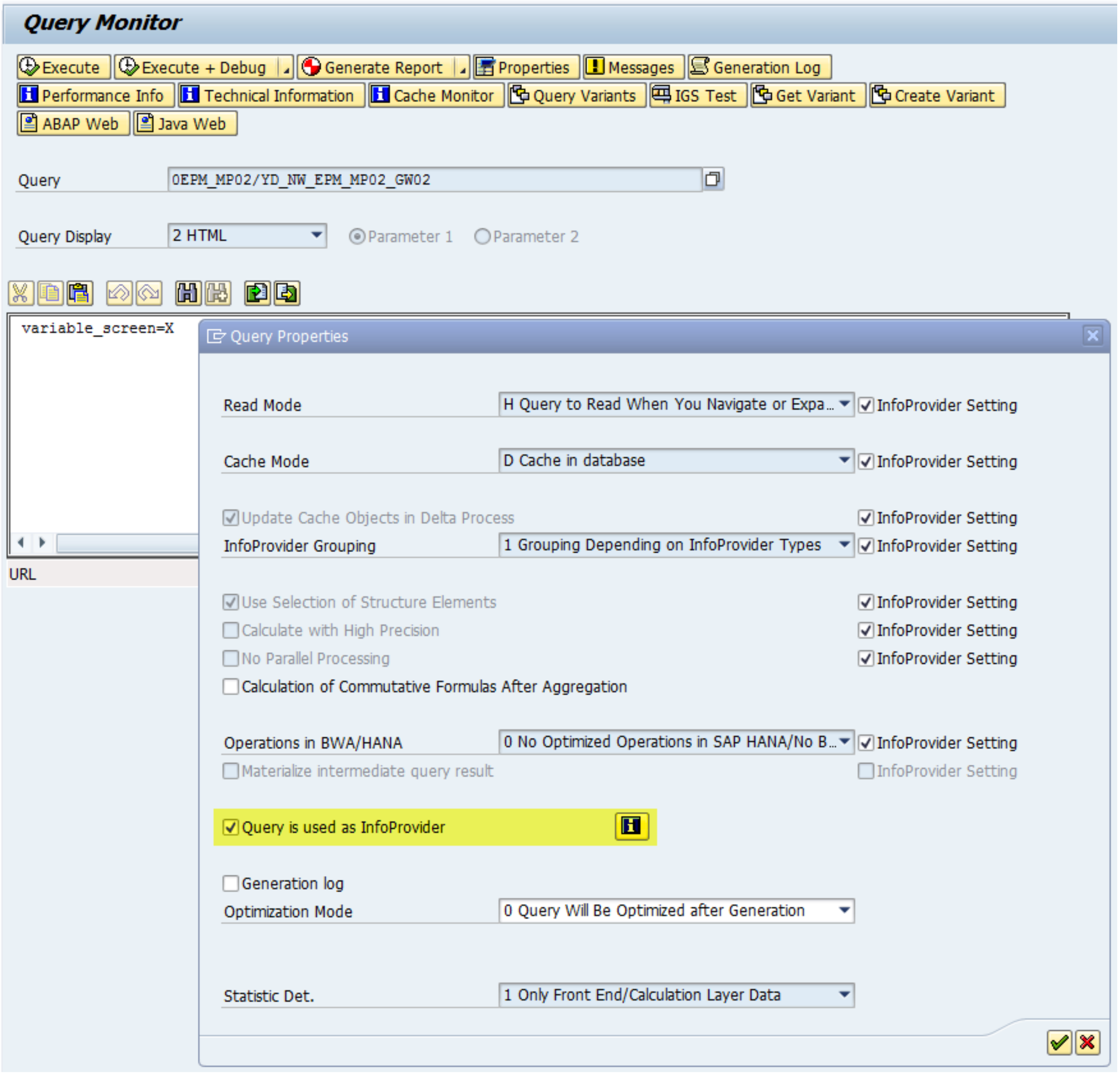
Display	Table Name	Database Ta...	Index Size	Index Status	Delta Index	Last Changed by	Last Changed
	/BIC/FQ$QYD_NW_EPM_M...	100.000	2.969	🟩		WITZELG	24.04.2014 14:36:05

Picture 40: Fill Query Snapshot

Another way to create a Query Snapshot is the use of Transaction RSRT – in the query properties tick the button Query is used as InfoProvider and afterwards goes to transaction RSDDB and activate and fill the Index as described above.

Technically an InfoProvider in the '@Q'-namespace is derived and the query result is written to a flat table.

To update the existing query snapshots automatically this application process needs to be triggered in a process chain for automatic execution with process chain type Create/Fill SAP HANA/BWA Index Initially.



Query: 0EPM_MP02/YD_NW_EPM_MP02_GW02

Query Display: 2 HTML ● Parameter 1 ○ Parameter 2

variable_screen=X

Read Mode: H Query to Read When You Navigate or Expa... ☑ InfoProvider Setting

Cache Mode: D Cache in database ☑ InfoProvider Setting

☑ Update Cache Objects in Delta Process ☑ InfoProvider Setting

InfoProvider Grouping: 1 Grouping Depending on InfoProvider Types ☑ InfoProvider Setting

☑ Use Selection of Structure Elements ☑ InfoProvider Setting

□ Calculate with High Precision ☑ InfoProvider Setting

□ No Parallel Processing ☑ InfoProvider Setting

□ Calculation of Commutative Formulas After Aggregation

Operations in BWA/HANA: 0 No Optimized Operations in SAP HANA/No B... ☑ InfoProvider Setting

□ Materialize intermediate query result □ InfoProvider Setting

☑ Query is used as InfoProvider

□ Generation log

Optimization Mode: 0 Query Will Be Optimized after Generation

Statistic Det.: 1 Only Front End/Calculation Layer Data

Picture 41: Query Snapshot in RSRT

9.8 SPO as a HAP Source

A semantically partitioned object is an InfoProvider that consists of several InfoCubes or DataStore objects with the same structure.

Semantic partitioning is a property of the InfoProvider. You specify this property when creating the InfoProvider. Semantic partitioning divides the InfoProvider into several small, equally sized units (partitions).

A semantically partitioned object offers the following advantages compared to standard InfoCubes or standard DataStore objects:

- Better performance with mass data:

  The larger the data volume, the longer the runtimes required for standard DataStore objects and standard InfoCubes. Semantic partitioning means that the data sets are distributed over several data containers. This means that runtimes are kept short even if the data volume is large.

- Close data connection:

  Error handling is better. If a request for a region ends with an error, for example, the entire InfoProvider is unavailable for analysis and reporting. With a semantically partitioned object, the separation of the regions into different partitions means that only the region that caused the error is unavailable for data analysis.
- Working with different time zones:

  EDW scenarios usually involve several time zones. With a semantically partitioned object, the time zones can be separated by the partitions. Data loading and administrative tasks can therefore be scheduled independently of the time zone.

For more information's please take a look:

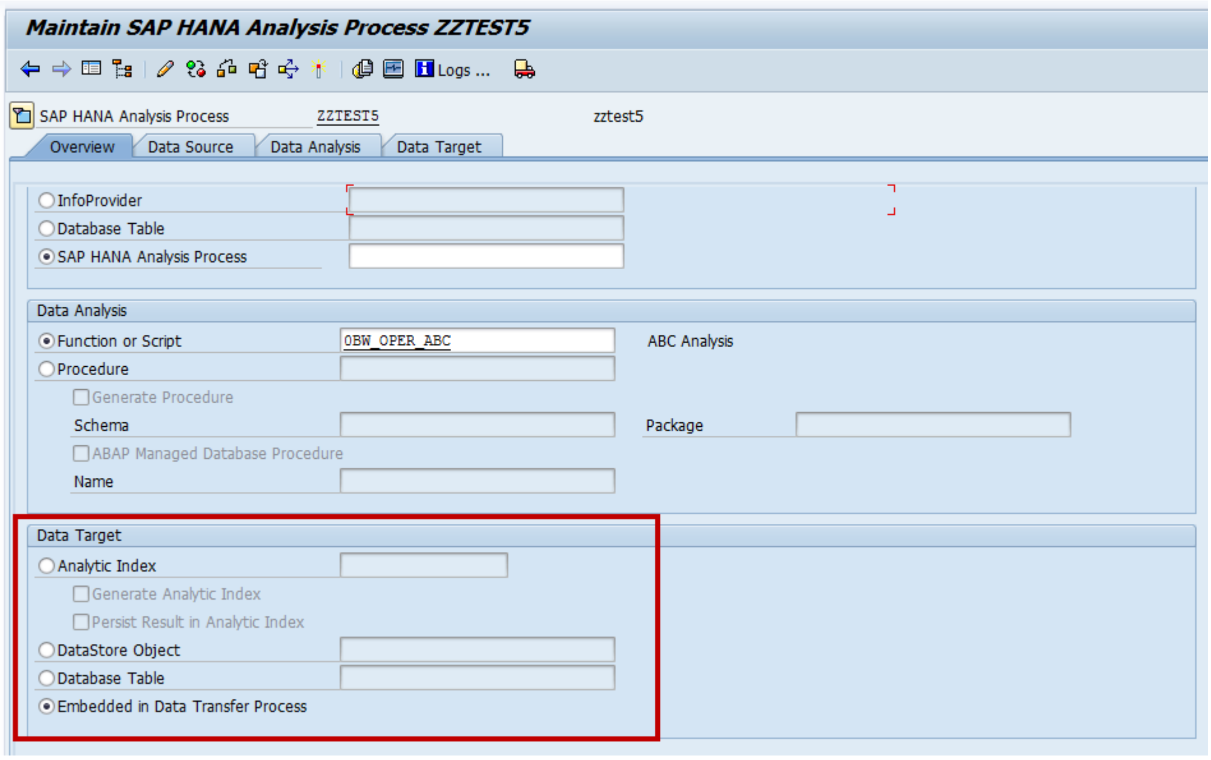
http://help.sap.com/saphelp_nw73ehp1/helpdata/en/d1/468956248e4d9ca351896d54ab3a78/content.htm?frameset=/en/fe/fc5f493e454de085e24077aaf02cdf/frameset.htm

9.9 Database tables as a HAP Source

It is also supported to use any table which is known in BW. It doesn't matter if this is a fact table, an attribute or a text table. Important is, that this table is known in the ABAP dictionary in BW.

9.10 SAP HANA Analysis Process as a HAP Source

Another possibility is to use a SAP HANA Analysis Process as a source. This needs to be a HAP where the data target is embedded in a Data Transfer Process. It is only possible to use this kind of HAP as a data target – it is not possible to use HAP which is not DTP embedded as a source.



Maintain SAP HANA Analysis Process ZZTEST5

SAP HANA Analysis Process ZZTEST5 zztest5

○ InfoProvider
○ Database Table
● SAP HANA Analysis Process

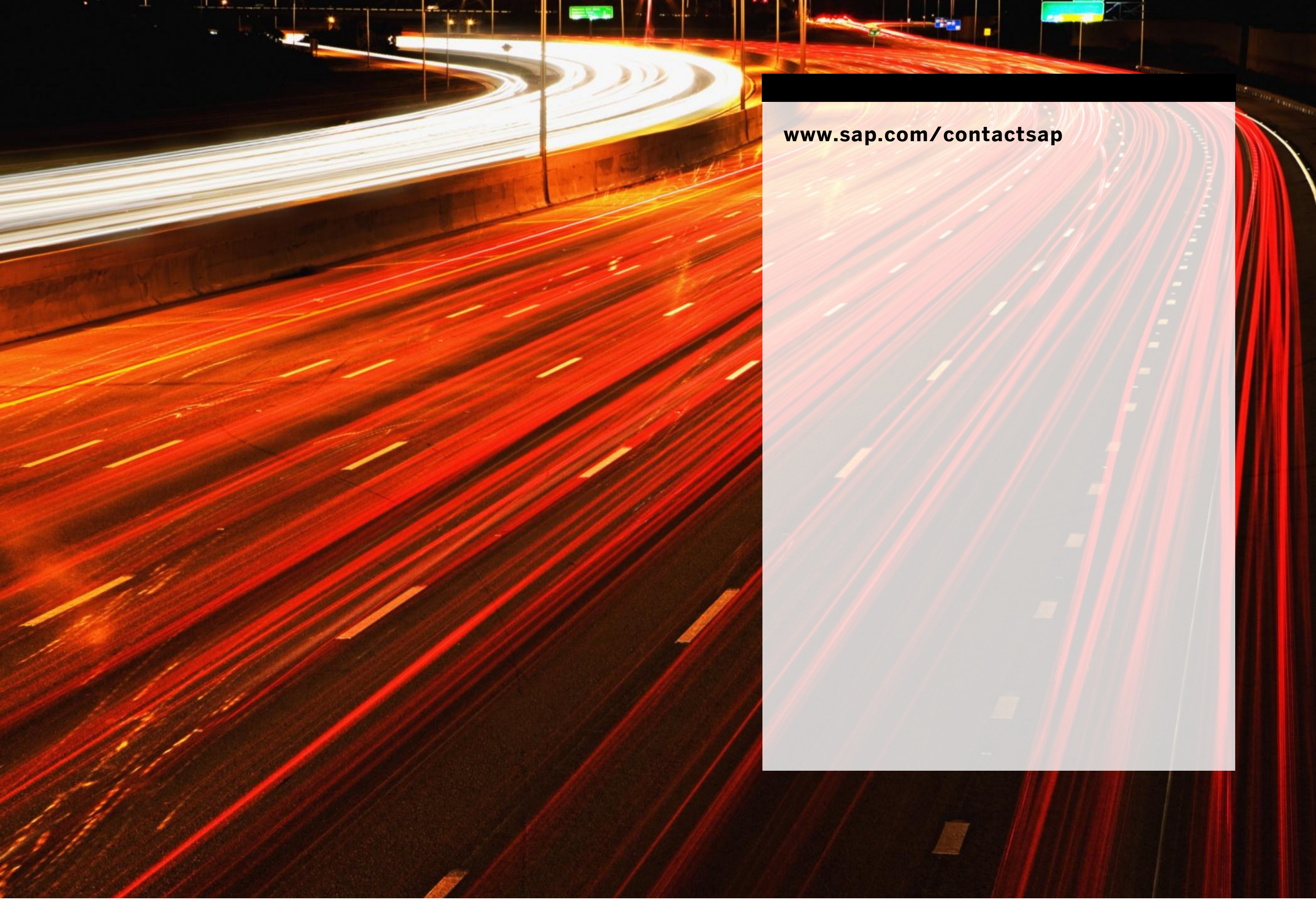
Data Analysis

● Function or Script 0BW_OPER_ABC ABC Analysis
○ Procedure
□ Generate Procedure
Schema
Package
□ ABAP Managed Database Procedure
Name

Data Target

○ Analytic Index
□ Generate Analytic Index
□ Persist Result in Analytic Index
○ DataStore Object
○ Database Table
● Embedded in Data Transfer Process

Picture 42: Data target embedded in DTP



www.sap.com/contactsap

The Best-Run Businesses Run SAP™