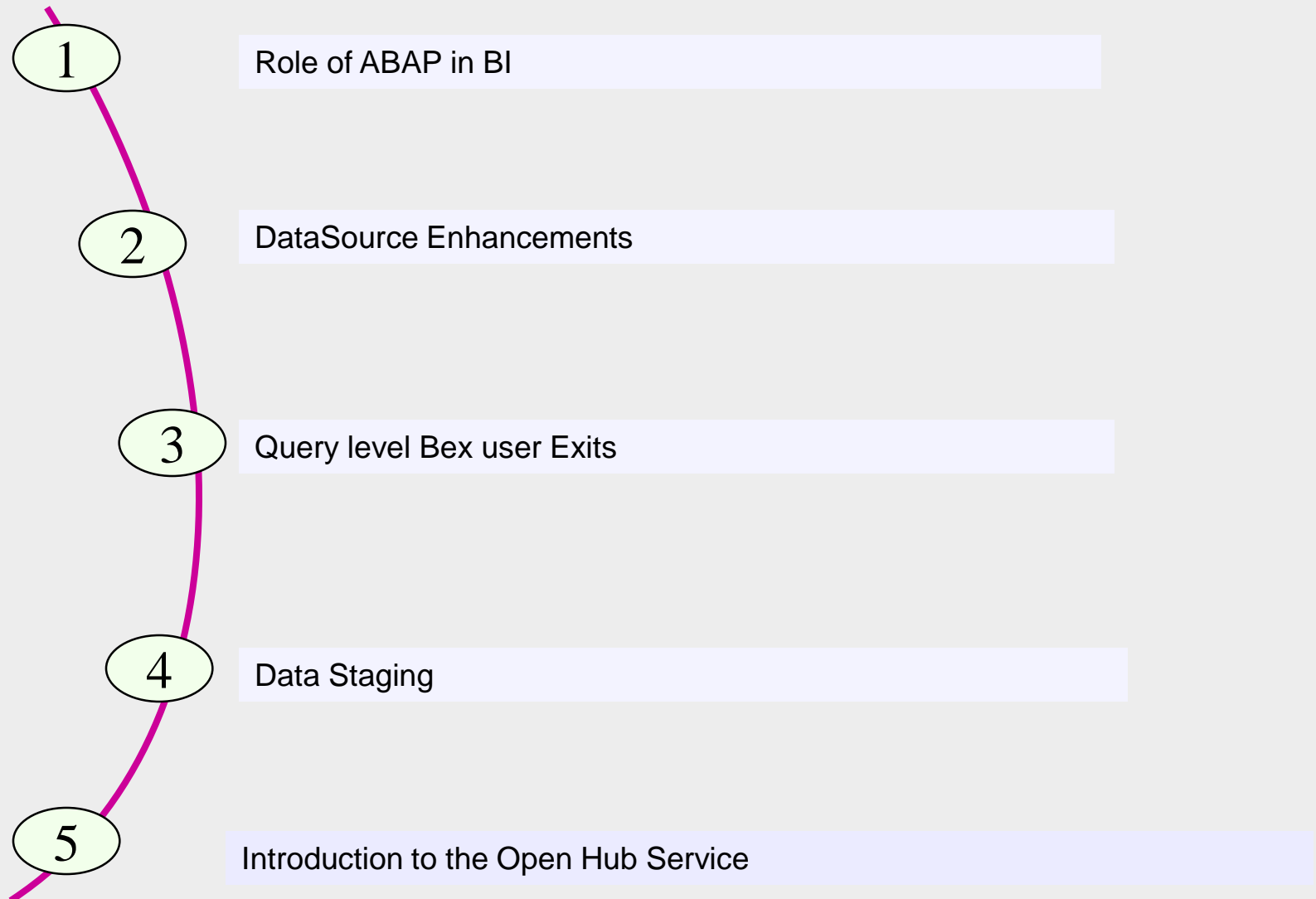


# SAP BW

## Lesson 11: ABAP, Open-hub Overview

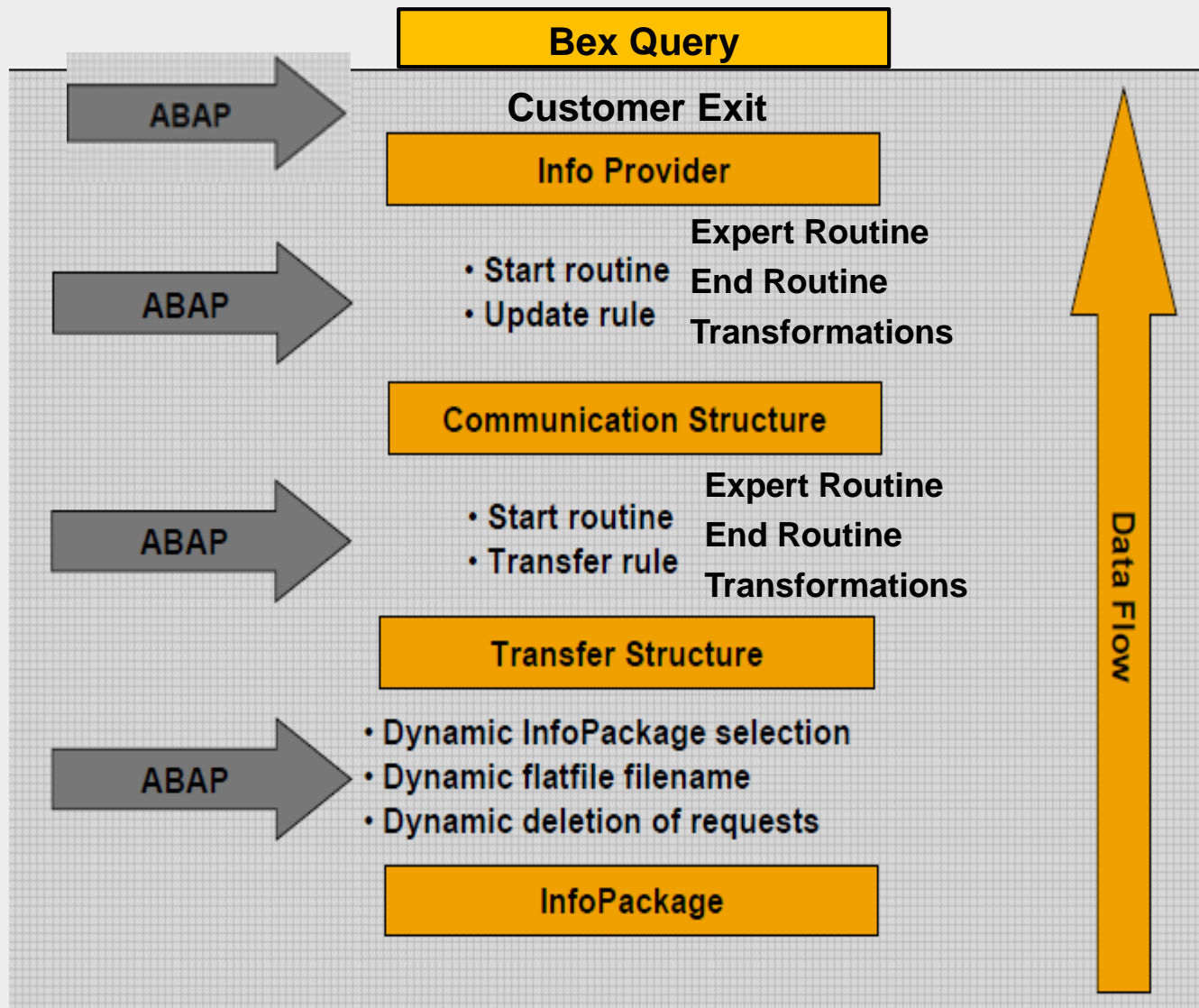


# BI related ABAP Overview & open-hub



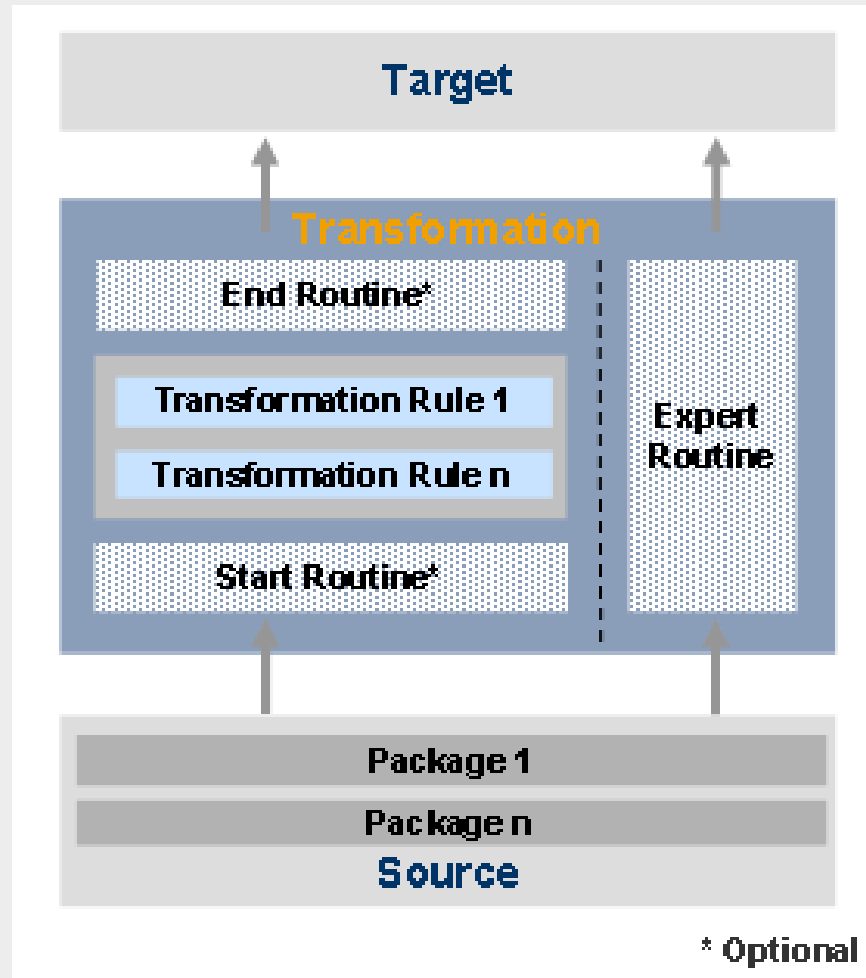


# 1. Role of ABAP in BI – 3.x





# 1.1 BI - Transformation





## 2. DataSource Enhancements

- Creating generic Data Sources using function modules for Data Extraction
- Enhancing Data Sources for transaction data, master data attributes, and texts
- Using the Service API function enhancement to enrich data
- Overview of the options available for enhancing data in the source system





# Using the Tools for Generic Data Extraction

- When should we use the generic data extraction tools to create a DataSource?
- Business Content does not include a DataSource for your application
- You want to implement a delta method on your generic DataSource that cannot be implemented by using the generic delta functionality (timestamp, date ...)
- The application does not allow you to create additional application specific generic extractors (CO-PA, FI-SL, LIS).
- You use your own programs in the SAP system to populate your own tables
- You have to extract data from several DB tables when:
  - Using a view is not possible because of insufficient JOIN
  - Only some fields are relevant to these tables
  - Data has to be enriched with information not available in the BW system




# Tools for Generic Data Extraction: RSO2

**Maintain Generic DataSources**




 

**DataSource**

☒ Transaction data      ZSAPI\_SFLIGHT\_SIMPLE 

☐ Master Data Attributes

☐ Texts

 Create       Change       Display


**DataSources for**


- Transaction data
- Master data attributes
- Texts

# Creating DataSources for Generic Data Extraction



**Create DataSource for Transactn data: ZSAPI\_SFLIGHT\_SIMPLE**

 Generic Delta

DataSource	ZSAPI_SFLIGHT_SIMPLE	Extraction from View
Applic. Component	SAP-R/3 	Extraction from Query
Obj. status	New	Extraction by FM

Extraction from DB View

View/Table  
ExtractStruct.

Extraction from SAP Query

InfoSet

Extraction by Function Module

Function Module  
Extract.Struct.

Texts

Short Description  
Medium description  
Long description

Function Module	RSAX_BIW_GET_DATA_SIMPLE
Extract.Struct.	SFLIGHT

Short Description	SFLIGHT
Medium description	SFLIGHT
Long description	SFLIGHT





# Enhancing Business Content DataSources

## ➤ **Reasons:**

- You want to add extra attributes to the characteristics (master data) shipped with the system
- You want to populate additional fields that you appended to the extract structure with data at the time of extraction
- You want to change/enhance texts or hierarchies

## ➤ **Prerequisites:**

- The additional data is available at the time the extraction takes place
- All information that is needed to determine the additional data clearly is available

# Creating and Maintaining Customer Appends



In BW-IMG  
(TA SBIW)

-DMATERIAL\_ATTR

Material Number

1

Enhance Extraction Structure

Create Struct.

Structure name ZABIW\_MARA\_S



2

3

Dictionary: Change Append Structure

Append structure ZABIW\_MARA\_S

Short description Append for BIW\_MARA\_S

Component

ZZDISMM

...

Component type

DISMM

...

# Developing Service API Function Enhancements



In BW-IMG  
(TA SBIW)

-QMATERIAL\_ATTR

Material Number

1  
Function Enhancement

Enhancement	Impl	Exp	
Function exit	✓ ✓	⚠  ●	RSAP0001 Customer function calls EXIT_SAPLRSAP_001 EXIT_SAPLRSAP_002 EXIT_SAPLRSAP_003 EXIT_SAPLRSAP_004

## Project Management of SAP Enhancements



Project ZBWABAP

Create

### Subobjects

- ☐ Attributes
- ☐ Enhancement assignment
- ☒ Components
- ☐ Documentation

Display

Change



### 3. Query level Bex user Exits

Currency/Unit | Advanced |  
General Replacement Path Details Default Values

**Description**  
OTB Free Stock Region  
☐ Use Standard Text

**Technical Name**  
ZORG\_EMKT

**Global Settings**  
Type of Variable  
Characteristic Value  
Processing By  
Customer exit  
Reference Characteristic  
Sales Organization

Bex Variable

Exit name	Short text
<input type="checkbox"/> RSAP0001	Customer function calls in the service API
<input checked="" type="checkbox"/> RSR00001	BW: Enhancements for global variables in reporting
<input type="checkbox"/> RSR00002	BW: Virtual characteristics and key figures in Reporting
<input type="checkbox"/> RSR00003	BW: Moving characteristic values

Function Builder: Display EXIT\_SAPLRRS0\_001

Function module EXIT\_SAPLRRS0\_001 Active

Attributes Import Export Changing Tables Exceptions Source code

```
1 FUNCTION EXIT_SAPLRRS0_001.  
2 *"  
3 *" "Lokale Schnittstelle:"  
4 *"  
5 *" IMPORTING  
6 *" VALUE(I_VNAM) LIKE RSEGLOBV-VNAM  
7 *" VALUE(I_VARTYP) LIKE RSEGLOBV-VARTYP  
8 *" VALUE(I_IOBJNM) LIKE RSEGLOBV-IOBJNM  
9 *" VALUE(I_S_COB_PRO) TYPE RSD_S_COB_PRO  
10 *" VALUE(I_S_RKBID) TYPE RSR_S_RKBID  
11 *" VALUE(I_PERIV) TYPE RRO01_S_RKB1F-PERIV  
12 *" VALUE(I_T_VAR_RANGE) TYPE RRS0_T_VAR_RANGE  
13 *" VALUE(I_STEP) TYPE I DEFAULT 0  
14 *" EXPORTING  
15 *" VALUE(E_T_RANGE) TYPE RSR_T_RANGESID  
16 *" VALUE(E_MEEHT) LIKE RSEGLOBV-MEEHT  
17 *" VALUE(E_MEFAC) LIKE RSEGLOBV-MEFAC  
18 *" VALUE(E_WAERS) LIKE RSEGLOBV-WAERS  
19 *" VALUE(E_WHFAC) LIKE RSEGLOBV-WHFAC  
20 *" CHANGING  
21 *" VALUE(C_S_CUSTOMER) TYPE RRO04_S_CUSTOMER OPTIONAL  
22 *"  
23  
24 INCLUDE ZXRSPU01 .  
25  
26  
27 ENDFUNCTION.  
28
```

Customer Exit  
in CMOD



# BEx-Variables

➤ One way to fill a variable value is to use ABAP coding.

➤ We can differentiate between two alternatives:

1. Fill the value independently with a program
2. Fill the value dependent on an other variable value

➤ Default information:

- For filling variables the standard SAP Enhancement RSR00001 is used
- See the documentation related to this enhancement for details and example coding (transaction SMOD)
- In the function module of this enhancement you will find include ZXRSRU01 where you implement your program code
- The exit is called four times, called "steps" (I\_STEP)



# BEx-Variables (cont.)

➤Default information:

- Step 1 (I\_STEP = 1) is called before the processing of the variable pop-up and gets called for every variable of the processing type, "customer exit." You can use this step to fill your variable with a default or proposal value.
- Step 2 (I\_STEP = 2) is called after processing of the variable pop-up. This step is called only for those variables that are not marked as "ready for input" and are set to "mandatory variable entry".
- Step 3 (I\_STEP = 3) is called after all variable processing and gets called only once and not per variable. Here you can validate the user entries.
- Step 0 (I\_STEP = 0) is called for variables that are used in authorizations objects.



## 4. Data Staging

**Overview**

**InfoPackages**

**DTPs**

**Transformations**

**Update rules**

# Sequential Order to Work with SAP BI Structures



**The BW ABAP routines are processed sequentially for all records and all fields:**

**InfoPackage** – Create dynamic filenames (for flat file import)

**InfoPackage** – Dynamic selection – selection criteria 1 ... n

**Transfer rule** – Start routine

**Transfer rule** – Dynamic routines to combine field by field from the transfer structure to the communication structure

**Update rule** – Start routine

**Update rule** – Dynamic routines to combine field by field from the communication structure to the infoprovider key figures or to the ODS-object data fields

**Update rule** – Dynamic routines to combine field by field from the communication structure to the infoprovider characteristics or the ODS-object key fields

**InfoPackage** – Deletion or taking out of requests after the loading process:

**Transformations (BI 7)** – Start Routine, End Routine , Expert Routine



The screenshot shows the SAP Data Selection dialog box. The 'Data Selection' tab is highlighted with a red circle. The 'Enter Selections (Optional):' table has the following data in the first row:

InfoObject	Technical ...	Name	From Value	To Value	Ty...	D...	Type (Varia...	FData...	Field ...	Conv...
@COSTELMNT	COSTELMNT				6		ABAP routine	CHAR	10	

The word 'DYNAMIC' is written in the center of the table and circled in red. To the right of the table, there is a vertical toolbar with icons for selection, navigation, and help.

Below the table, there are navigation buttons (left arrow, right arrow) and a text input field.

On the right side of the image, there is a grey box containing the following text:

Date field (due date is actual day):

- Get yesterday (00:00 to 24:00h)
- Get last week (Monday to Sunday)
- Get last month (first to last day of last month)
- Get last quarter (first to last day of last quarter)
- Get last year (1st of Jan. to 31st of Dec. of previous year)

No date field:

- Free boundary (also a period boundary is possible)

All fields:

- ABAP Routine
- OLAP Variable



# Infopackages : Dynamic Determination of Filenames

➤ Should you upload from client workstation or the application server?

➤ It is preferable to use the server path ☐ advantages:

- Data load in batch mode is possible
- Path is physically equal to all administrators
- Scenario in which to use it:
- Transaction data is delivered monthly by flat file
- The naming convention includes the name of the month within the filename
- Directory paths are maintained for every year
- Coding:
- Concatenate:
- Directory path
- Actual year
- Actual month

Data Selection External data Processing Data Targets Update Schedule

Load transaction data from the source system

Load External Data from ☒ Client Workstation ☐ Application Server

File Is Data File Create Routine Delete

Name of file C:\temp\IT\_COSTCENTER\_TRANS\_2003011.csv

File Type ☐ ASCII File (CR Separator) ☒ CSV file Hex

Data Separator ; ☐ Hex

Escape Sign " ☐ Hex

Separator for Thousands .

Character Used for Decimal Point ,

Currency Conversion for External Sys ☒

Number of Header Rows to be Ignored 1 Preview

# DTPs



Extraction    Update    Execute

Data Source    DataStore Object    ZATRT031  
Store inventory DSO (Daily)

Extraction Mode    Full    Filter    Semantic Groups

Package Size    Filter

Key Date for Master    Change Selection

☐ Currency conversion

Extraction From...

- ☐ Active Table (W)
- ☒ Active Table (W)
- ☐ Archive (Full Ext)
- ☐ Change Log

Base Unit	to	
Calendar Day	to	
Document currency	to	
Local currency	to	
Material	to	
Plant	to	
Data Record	to	0
Update Mode	to	
Sales unit	to	

DTP Filter Selection Screen

```

form compute_CALDAY
  tables l_t_range structure rssidrange
  using i_r_request type ref to IF_RSBK_REQUEST_ADMINTAB_VIEW
  i_fieldnm type RSFIELDNM
  changing p_subrc like sy-subrc.
*
*   Insert source code to current selection field
*$$$ begin of routine - insert your code only below this line
DATA : gv_calday_p TYPE cobjectv,
       gv_calday TYPE sy-datum.

CLEAR : gv_calday_p,
       gv_calday.

SELECT SINGLE value
  FROM zbw_upd_param
  INTO gv_calday_p
  WHERE odsobject = 'SNAPSHOT'
        AND fldname = 'CALDAY'.

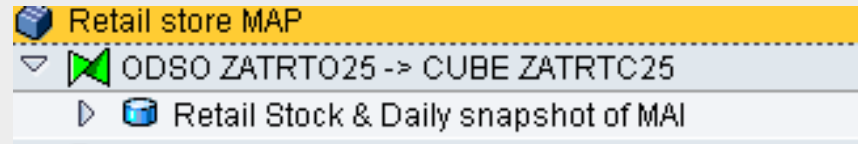
IF sy-subrc = 0.
  IF NOT gv_calday_p IS INITIAL.
    gv_calday = gv_calday_p.
    gv_calday = gv_calday - 1.
  ELSE.
    gv_calday = sy-datum - 1.
  ENDIF.
ENDIF.

l_t_range-fieldname = i_fieldnm.
l_t_range-sign = 'I'.
l_t_range-option = 'EQ'.
l_t_range-low = gv_calday.
APPEND l_t_range.

p_subrc = 0.
  
```

DTP Filter : ABAP Code

# Transformations :



Posi	Ke	InfoObject	Ico	Descript.
1	🔑	OMATERIAL	🏠	Material.
2	🔑	OPLANT	🏠	Plant
3		ORECTOTSTCK	🏠	Receipt Quantity Total Stock
4		OISSTOTSTCK	🏠	Issue Quantity Total Stock
5		OBASE_UOM	🏠	Base Unit of Measure
6		ORECVS_VAL	🏠	Value Received into Valuated Stock
7		OISSVS_VAL	🏠	Value issued from valuated stock
8		ZSTOCKQTY	🏠	Stock Quantity
9		OPRICE_AVG	🏠	Moving Average Price / Periodic Unit Price
10		OCALDAY	🕒	Calendar Day
11		OLOC_CURRCY	🏠	Local currency
12		OCURRENCY	🏠	Currency Key
13		ORECORDMODE	🏠	BW Delta Process: Update Mode
14		ZSTKVALUE	🏠	Stock Value @ MAP

Rule Group: Standard Group							
Ru	Rule Name	Posi	Ke	InfoObject	Ico	Descript.	Int
=	OMATERIAL	1	🔑	OMATERIAL	🏠	Material.	<input type="checkbox"/>
🏠	ZAFS_MAT	2	🔑	ZAFS_MAT	🏠	AFS Material	<input type="checkbox"/>
=	OPLANT	3	🔑	OPLANT	🏠	Plant	<input type="checkbox"/>
🏠	DAF_STCAT	4	🔑	DAF_STCAT	🏠	AFS Stock Category	<input type="checkbox"/>
=	ORECTOTSTCK	5		ORECTOTSTCK	🏠	Receipt Quantity Total Stock	
=	OISSTOTSTCK	6		OISSTOTSTCK	🏠	Issue Quantity Total Stock	
=	ORECVS_VAL	7		ORECVS_VAL	🏠	Value Received into Valuated Stock	
=	OISSVS_VAL	8		OISSVS_VAL	🏠	Value issued from valuated stock	
=	ZSTOCKQTY	9		ZSTOCKQTY	🏠	Stock Quantity	
=	OPRICE_AVG	10		OPRICE_AVG	🏠	Moving Average Price / Periodic Unit Price	
=	ZSTKVALUE	11		ZSTKVALUE	🏠	Stock Value @ MAP	
▶	ZLICSPAN	15	🔑	ZLICSPAN	🏠	Licensee & Spanish indicator	



# Transformations : Start / End Routines

## Transformation Display

```
83
84 TYPES: BEGIN OF gty_material,
85     material TYPE /bi0/oimaterial,
86     af_stcat TYPE /bi0/oiaf_stcat,
87     zafs_mat TYPE /bic/oizafs_mat,
88 END OF gty_material.
89
90 DATA: git_material TYPE STANDARD TABLE OF gty_material.
91
92 ***--End of Changes OMATERIAL V 1.0 (GKRISHNA)
93
94 *$$$ end of global - insert your declaration only before this line *--
95
96 METHODS
97     start_routine
98     IMPORTING
99         request                type rsrequest
100        datapackid             type rsdatapid
101     EXPORTING
102         monitor                type rstr_ty_t_monitors
103     CHANGING
104         SOURCE_PACKAGE         type _ty_t_sc_1
105     RAISING
106         cx_rsrou_t_abort.
107 METHODS
108     inverse_start_routine
109     IMPORTING
110         i_th_fields_outbound    TYPE rstran_t_field_inv
111         i_r_selsel_outbound     TYPE REF TO cl_rsmnds_set
112         i_is_main_selection     TYPE rs_bool
113         i_r_selsel_outbound_complete TYPE REF TO cl_rsmnds_set
114         i_r_universe_inbound    TYPE REF TO cl_rsmnds_universe
115     CHANGING
116         c_th_fields_inbound     TYPE rstran_t_field_inv
117         c_r_selsel_inbound      TYPE REF TO cl_rsmnds_set
118         c_exact                 TYPE rs_bool.
119 ENDCCLASS.                "routine DEFINITION
```

```
260 * <-> result package
261 *-----
262 METHOD end_routine.
263 *=== Segments ===
264
265 FIELD-SYMBOLS:
266     <RESULT_FIELDS> TYPE _ty_s_TG_1.
267
268 DATA:
269     MONITOR_REC TYPE rstmonitor.
270
271 *$$$ begin of routine - insert your code only below this line
272 ... "insert your code here
273 *-- fill table "MONITOR" with values of structure "MONITOR_REC"
274 *-- to make monitor entries
275 ... "to cancel the update process
276 * raise exception type CX_RSROUT_ABORT.
277
278 FIELD-SYMBOLS: <fs> type st_zsd_d10,
279               <RP> TYPE _ty_s_TG_1.
280 If RESULT_PACKAGE is not INITIAL.
281     select doc_number req_date
282     from /bic/aZSD_D1000
283     into table t_zsd_d10
284     FOR ALL ENTRIES IN RESULT_PACKAGE
285     where doc number = RESULT_PACKAGE-doc number.
```



# Start Routine : Classic usage

➤ Particularly the start routine of the transformation gives.

- "classic" scenarios of usage:
- Selective deletion of records delivered with the data package
- Coding example:
  - DELETE SOURCE\_PACKAGE WHERE VKORG = "1000."

➤ Advantage:

- Smaller data package -- loading performance improves
- Reasons why (examples):
- The InfoPackage from the data source does not offer selection criteria
- The processing time with a selection from the InfoPackage sent to the source system takes considerably more time than without selection
- The deletion criteria is easy to code within the start routine
- Fill an internal table from a DDIC select
- Complex data cleansing or data consolidation from various
- source systems eventually with a look up as ETL process



# End Routine : Classic usage

- Particularly the End routine of the transformation gives “classic” scenarios of usage:
- Complex logic for fields which need look up to another DSO or master data to get data for those fields which are not present at Source Targets
- “Result\_package” is internal table which contain the data



# Field Routine: Classic usage

➤ "Source\_fields" contains the row level record data at transformation and in order to update value of field we have update "RESULT". We can use field routine for small if. Else logic or simple operation like get Net value from Price and Order qty.

Description

Target InfoObj. OREQ\_DATE Schedule line delivery date

Rule Type Routine

Transfer Routine ☐

Source Fields of Rule:

InfoObject	Icon	Long Description	Type	Lng...	Conv.R...	IOAssignmnt	Long Description
ODOC_NUMBER		Sales document	CHAR	10	ALPHA		

```
METHOD compute_OREQ_DATE.
```

Target Fields of Rule:

InfoObject	Icon	Long Description	Type	Lng...	Conv.R...
OREQ_DATE		Schedule line deliv...	DATS	8	

```
* IMPORTING
*   request      type rsrequest
*   datapackid   type rsdatapid
*   SOURCE_FIELDS-DOC_NUMBER TYPE /BIO/OIDOC_NUMBER
* EXPORTING
*   RESULT type _ty_s_TG_1-REQ_DATE
```

```
DATA:
  MONITOR_REC TYPE rsmonitor.
```





# Update Rules : InfoCube Characteristic ODS-Object Key Field

## ➤Interface:

- Like in the start routine global declaration within the header.
- The form routine "compute\_key\_field" delivers the following parameters:
- The table MONITOR for monitoring
- The communication structure COMM\_STRUCTURE with all fields from the data source
- The current record number of the loop over the data package
- The number of total records
- RESULT delivers to the marked characteristic the result value
- RETURNCODE says whether the current record will be processed or not
- ABORT clarifies whether the whole data package will be processed or not

# Update Rules : InfoCube Characteristic ODS-Object Key Field



**Change Source**

Characteristic: Controlling area      Key Fig.: Quantity

Source

- ☐ Source Chars
- ☐ Constant
- ☐ Master Data Attrib. of
- ☐ Formula
- ☒ Routine
- ☐ Initial Value

Controlling area

**Update - Controlling area: Create routine**

Name: controlling area

Editor



# Update Rules : Time References

Kennzahlberechnung   Merkmale   Zeitbezug

Merkmal: Datum des Ausgleichs   Methode: Zeitverteilung   Quellfelder: Datum des Ausgleichs

Quelle ändern

Merkmal: Datum des Ausgleichs   Kennzahl: Betrag in Hauswährung

Quellfeld:

- ☐ Quellmerkmal
- ☐ Konstante
- ☐ Stammdatenattribut von
- ☒ Formel
- ☐ Routine
- ☒ Zeitverteilung
- ☐ Initialwert

Belegdatum: ☒ normaler Kalender   ☐ Fabrikkalender

Within update rules there are various possibilities to reference time dependencies:

## Characteristic update rule with time reference

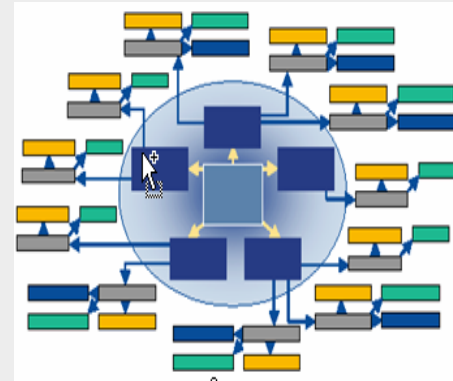
- It is possible to update data target characteristics of type "DATE" via an automatic time distribution
- The field is fed from a data field out of the communication structure
- For example, the data target characteristic calendar day is fed from the source characteristic calendar month
- The effect is that every summarized key figure has to be distributed to the calendar days of the month
- In addition to that, it can be defined whether the company calendar has to be used

## Time characteristics

- Within the frame time reference various data target time characteristics can be filled automatically from only one time characteristic of the data source

Zeitmerkmal	Methode	Quellfelder	Automatische ...
Kalendertag	←	Datum, an de...	
Kalenderjahr / Monat	←	Datum, an de...	✓
Kalendermonat	←	Datum, an de...	✓
Kalenderjahr / Quartal	←	Datum, an de...	✓
Kalenderjahr / Woche	←	Datum, an de...	✓
Kalenderjahr	←	Datum, an de...	✓

## 5. Open Hub Destination



# Prepare Me



- Purpose
- Use

## THE MEANS

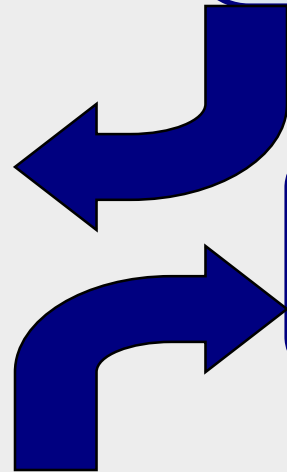
Open Hub Services

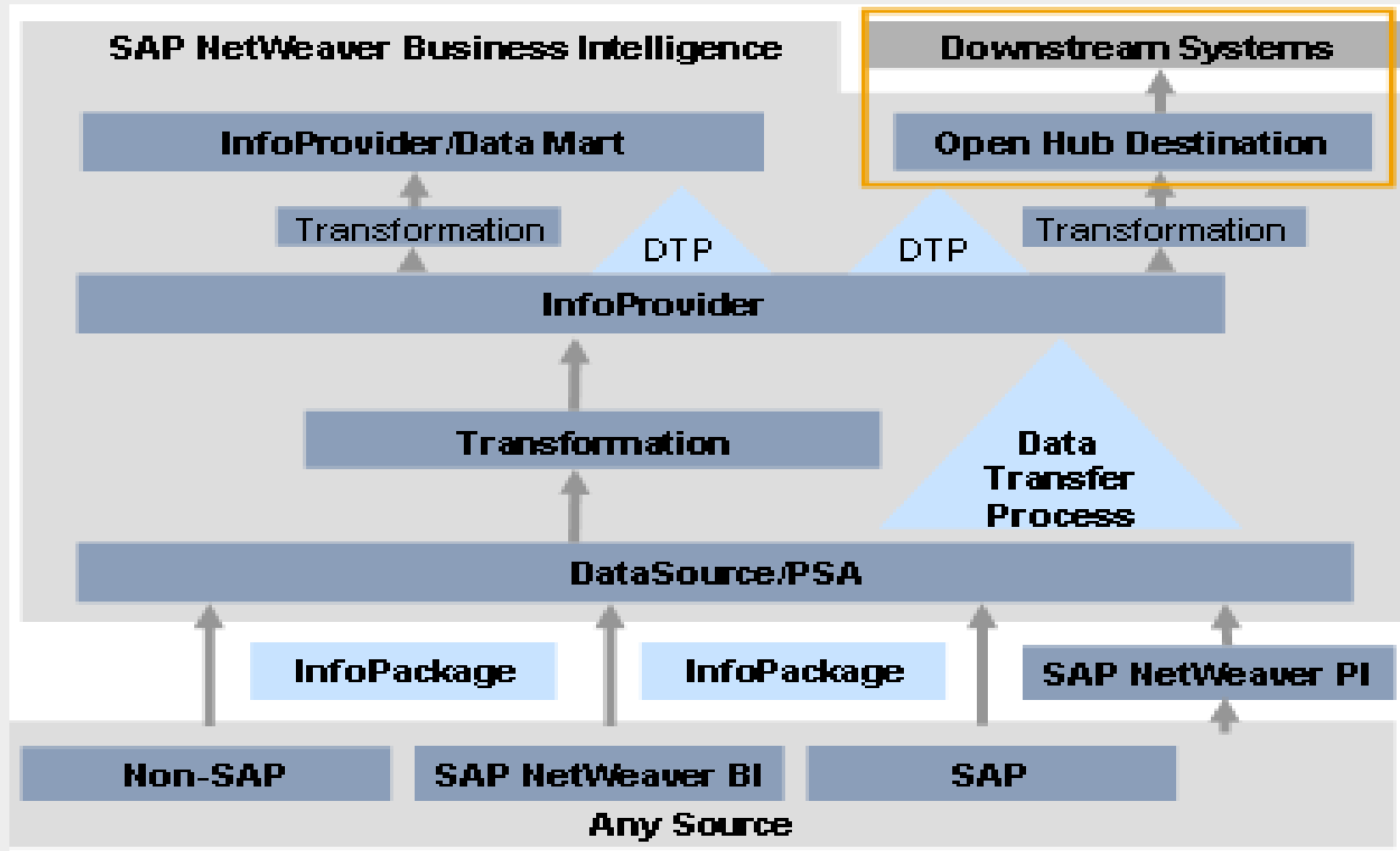
## THE GOAL

To get Operational Data out of SAP BI

Move data out of SAP BI  
system through  
different channels

HOW?  
Shows this  
course







# Purpose

- Open hub destination is the object that allows you to distribute data from a BI system to non-SAP data marts, analytical applications, and other applications.
- It ensures controlled distribution across multiple systems.
- Open hub destination defines the target to which the data is transferred.

# Use



- Database tables (in the database for the BI system) and flat files can act as open hub destinations. You can extract the data from a database to non-SAP systems using APIs and a third-party tool.
- In earlier releases, the open hub destination was called as InfoSpoke and it was not as tightly integrated into the data flow as Open hub destination is.



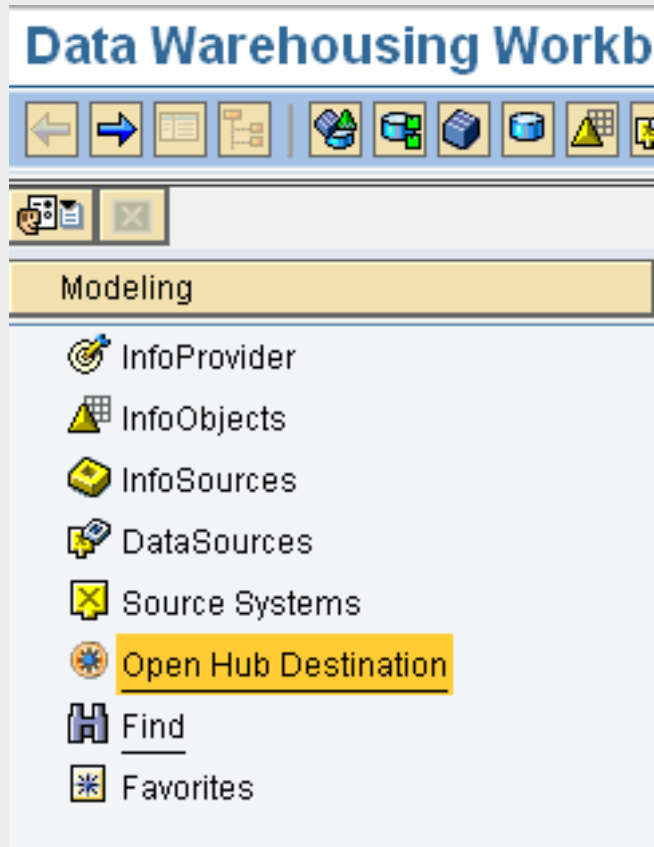


# OHS – Types

- Using OHS we can write data from SAP BI systems into below mentioned data targets:
  - Database Tables.
  - Flat File.
  - Third-Party Tool.
  
- The open hub destination contains all the information about a data target:
  - Type of destination.
  - Name of the flat file or database table and its properties.
  - Field list and its properties.





# OHS – Access




Using RSA1 transaction, Data Warehousing Workbench, we can access OHS under the Modeling tab

# OHS - Creation



 Create Open Hub Destination 

Open Hub Destination

Z\_TRAININ 

Description


Training OHS

InfoArea

BW\_TRAINING



Template

ObjectType

☒ 

Name

☒

[illegible]

# OHS – using Flat File



Open Hub Destination  Training OHS

Destination | Field Def.

Destination Type

☒ Application Server

Server name

Type of File Name

Appl. Server File Name

Directory

Separator

---

Open Hub Destination  Training OHS

Destination | Field Def.

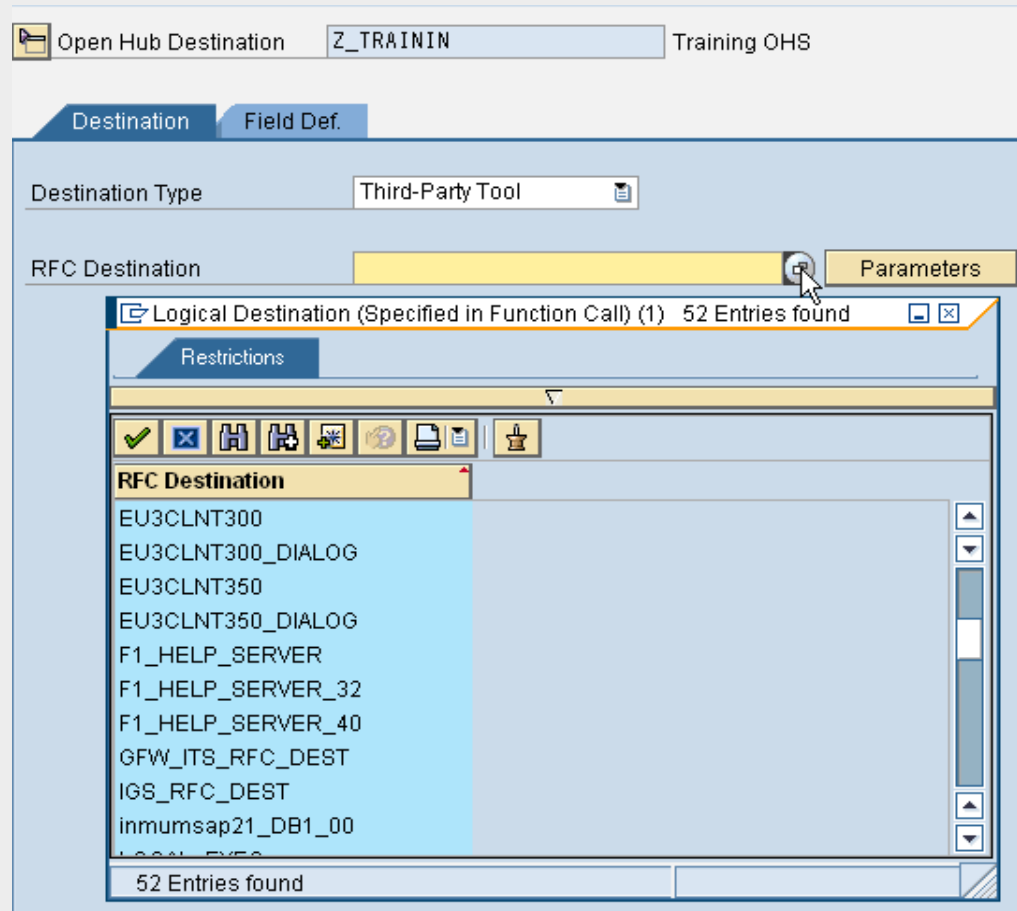
Template Flds

	Pos.	Se...	Field	Description	InfoObject Templa...	Type	Lng...	Data element
	1	<input type="checkbox"/>	MATERIAL	Material	0MATERIAL	CHAR	18	/BI0/OIMATERIAL
	2	<input type="checkbox"/>	MATL_CAT	Material Category	0MATL_CAT	CHAR	2	/BI0/OIMATL_CAT
	3	<input type="checkbox"/>	MATL_GROUP	Material group	0MATL_GROUP	CHAR	9	/BI0/OIMATL_GROUP
	4	<input type="checkbox"/>	MATL_TYPE	Material type	0MATL_TYPE	CHAR	4	/BI0/OIMATL_TYPE



# OHS – using Third-Party Tool

- Using Third-party tool, we can transfer data from SAP BI system, to different systems by linking them to BI system.
- You create an RFC destination for your third-party tool and enter it in the definition of the OHS.





# Transactions Used

➤RSA1 : Data Warehousing Workbench.



Browse through the below links for Self Study

1. [http://help.sap.com/saphelp\\_nw70ehp2/helpdata/en/e3/e60138fed083de10000009b38f8cf/frameset.htm](http://help.sap.com/saphelp_nw70ehp2/helpdata/en/e3/e60138fed083de10000009b38f8cf/frameset.htm)
2. <http://service.sap.com/bi>