SAP BW Training

Lesson 04: Data Acquisition Part 1 – ETL Components

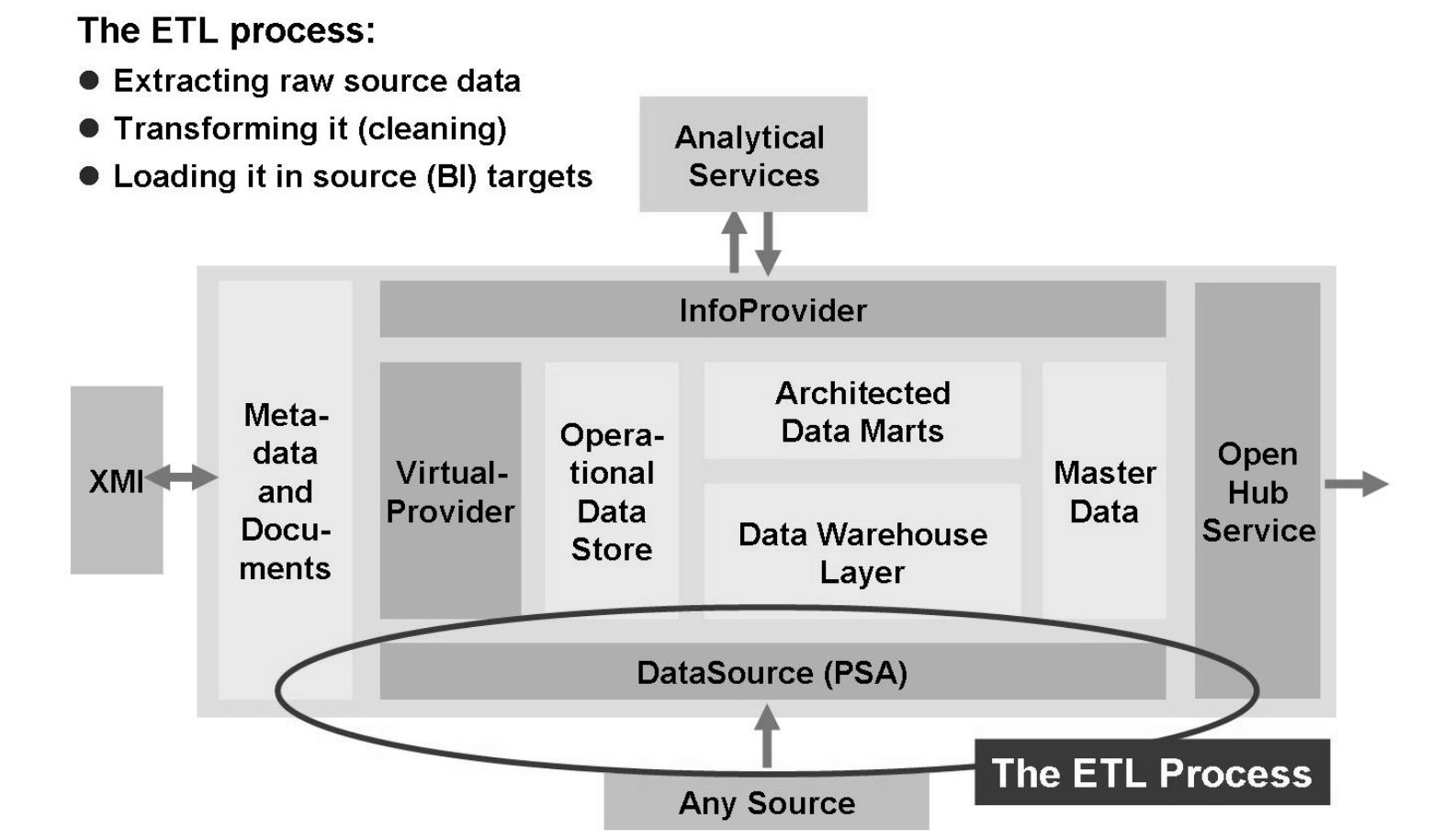
Capgemini

Components of ETL

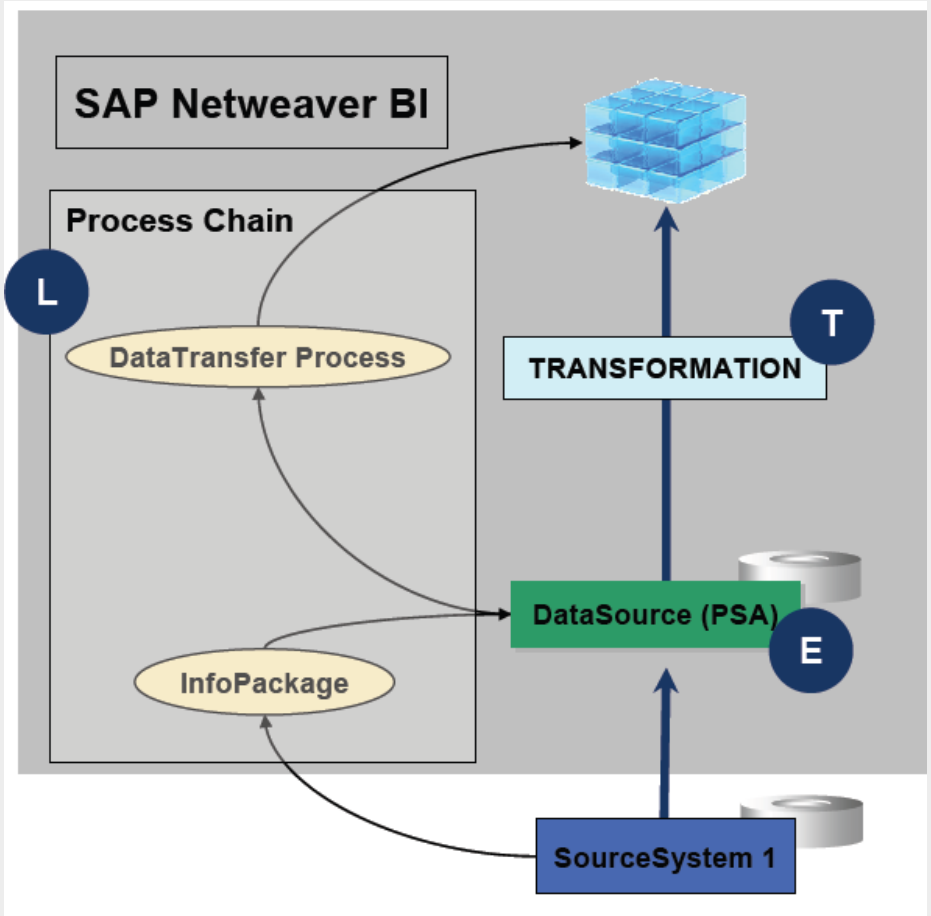
➢Components of ETL are:

- Source system
- DataSource
- InfoPackage
- InfoSource
- Transformation
- DTP

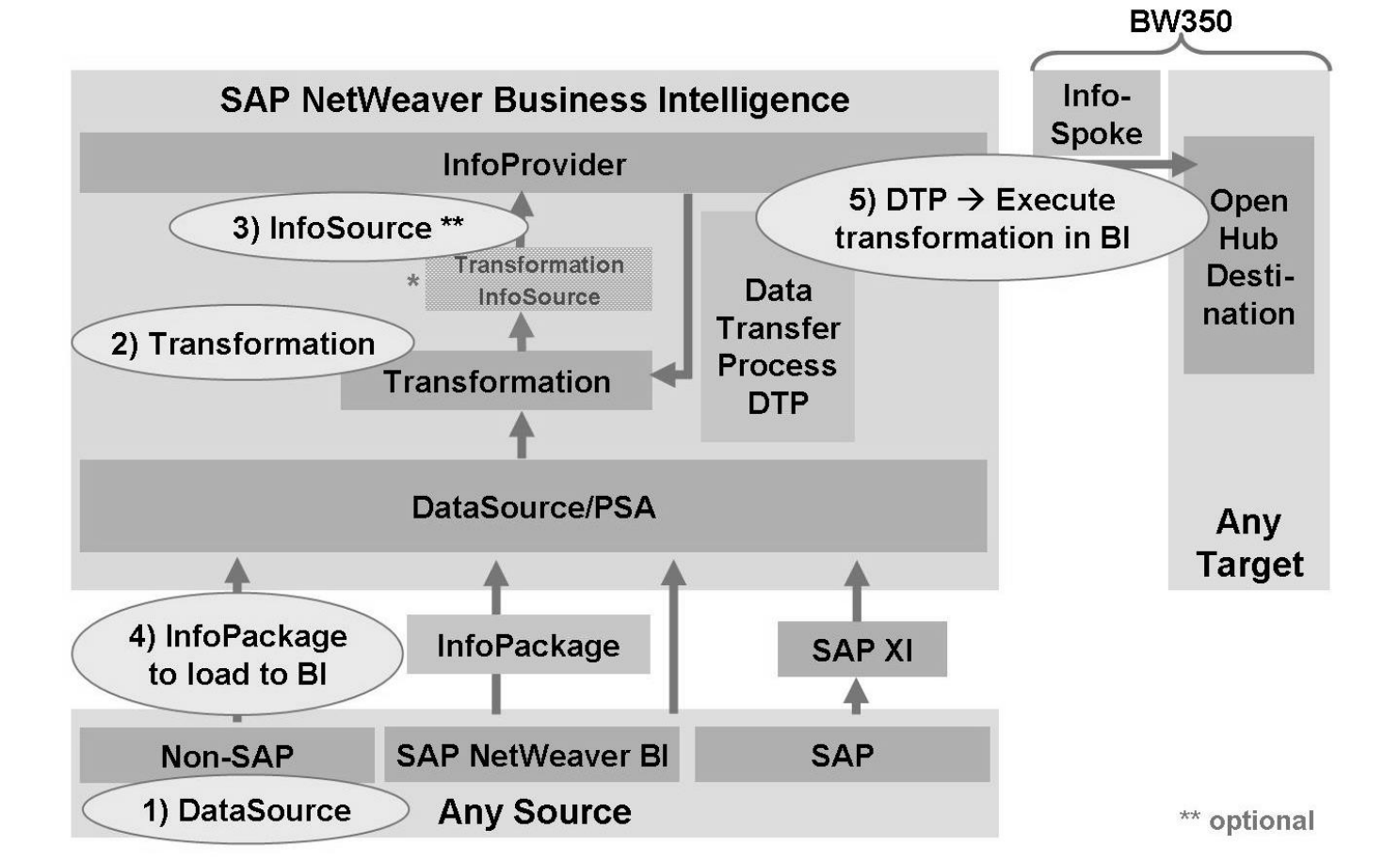
ETL Process


The ETL process:
Extracting raw source data
Transforming it (cleaning)
Loading it in source (BI) targets
Analytical Services
InfoProvider
XML
Meta-data and Docu-ments
Virtual-Provider
Opera-tional Data Store
Architected Data Marts
Data Warehouse Layer
Master Data
Open Hub Service
DataSource (PSA)
Any Source
The ETL Process

Simple Data Flow in BW



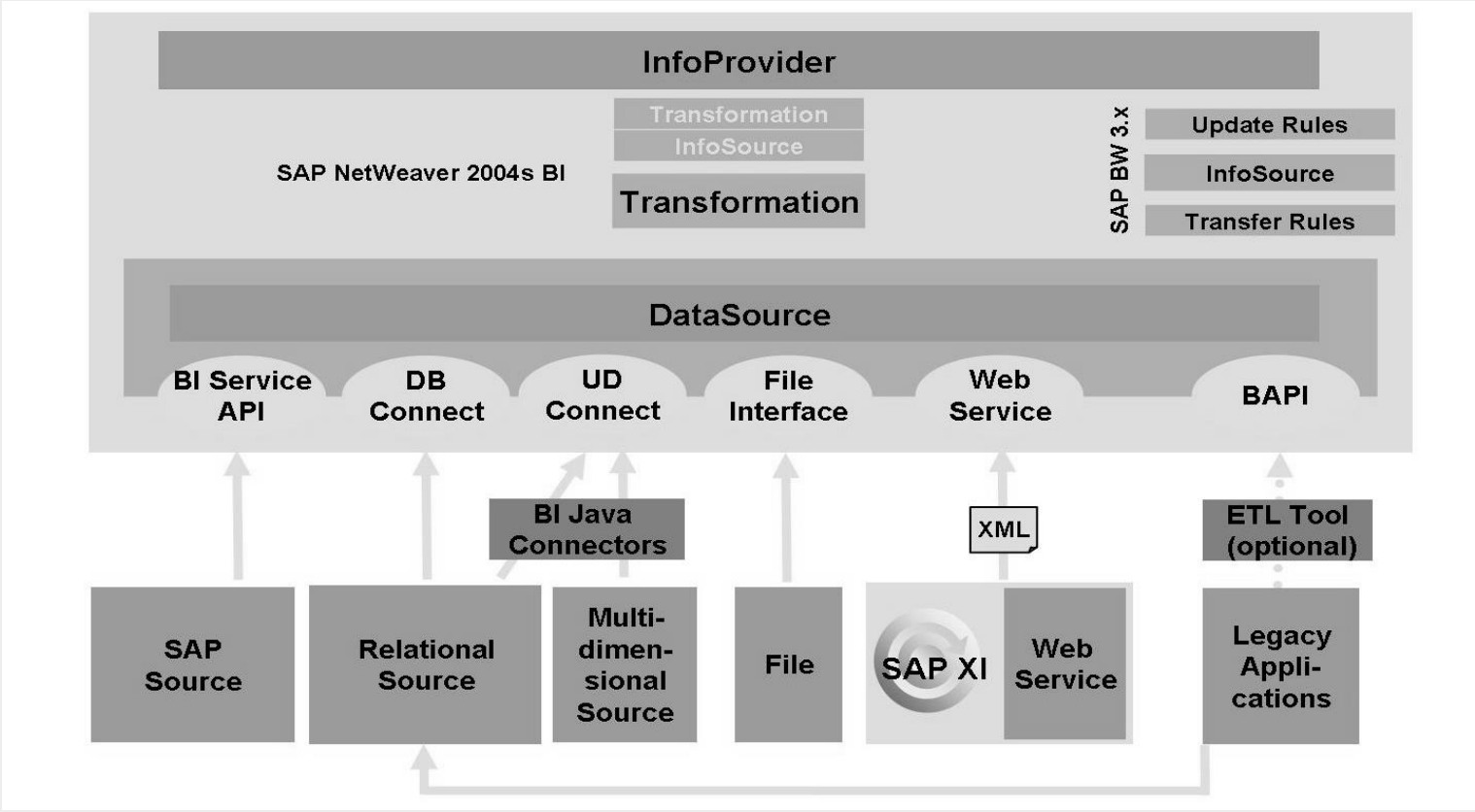
BI Data Flow Details



Source System

Source System

➢All systems that provide the SAP Business Information Warehouse with data are described as source systems.



Source System Types

1. SAP systems
2. BI systems

3. Flat files for which metadata is maintained manually and transferred to BI using a file interface

4. Database management systems into which data is loaded from a database supported by SAP using *DB Connect*, without using an external extraction program

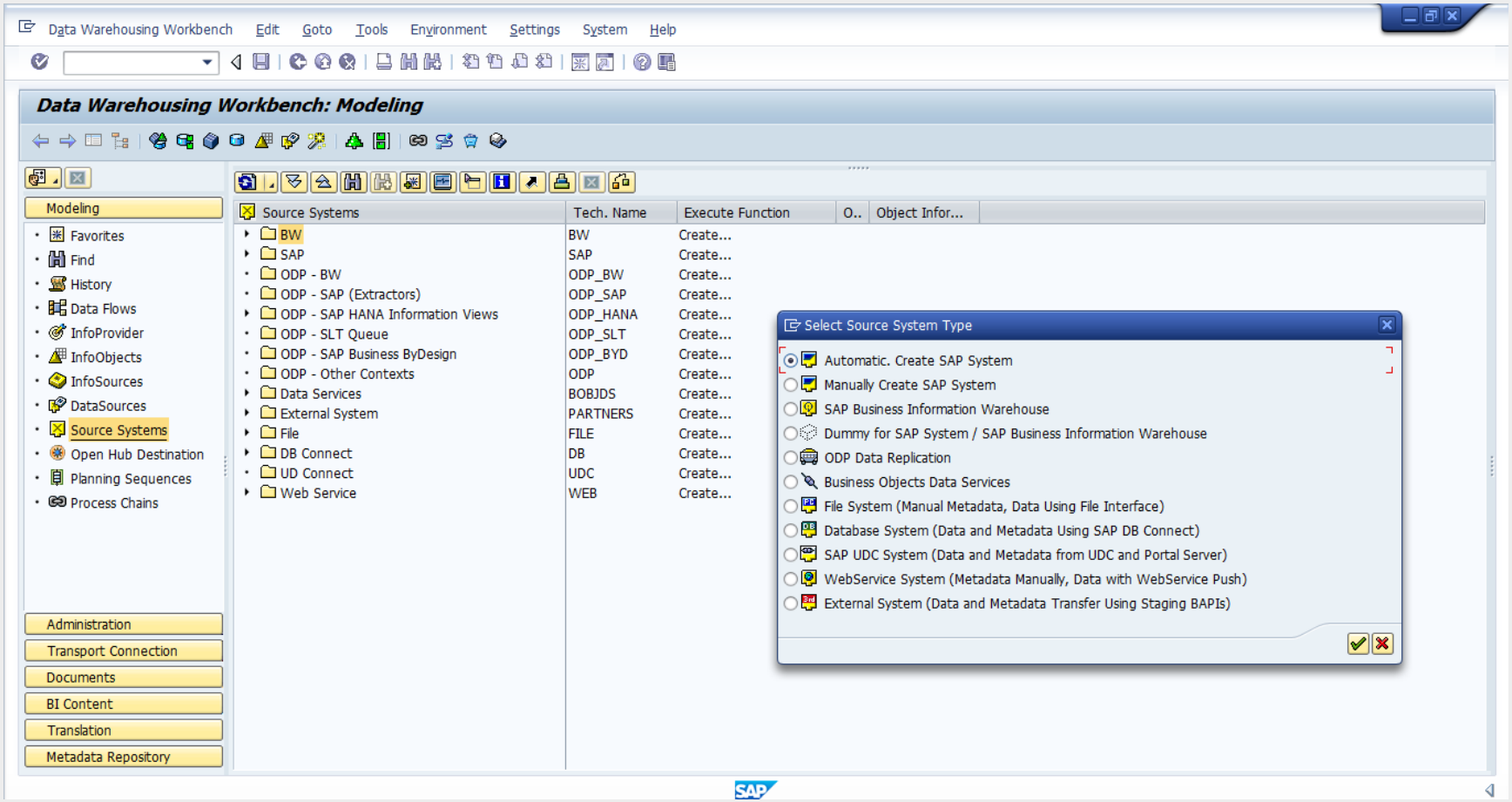
5. Relational or multidimensional sources that are connected to BI using *UD-Connect*

6. Web Services that transfer data to BI by means of a push

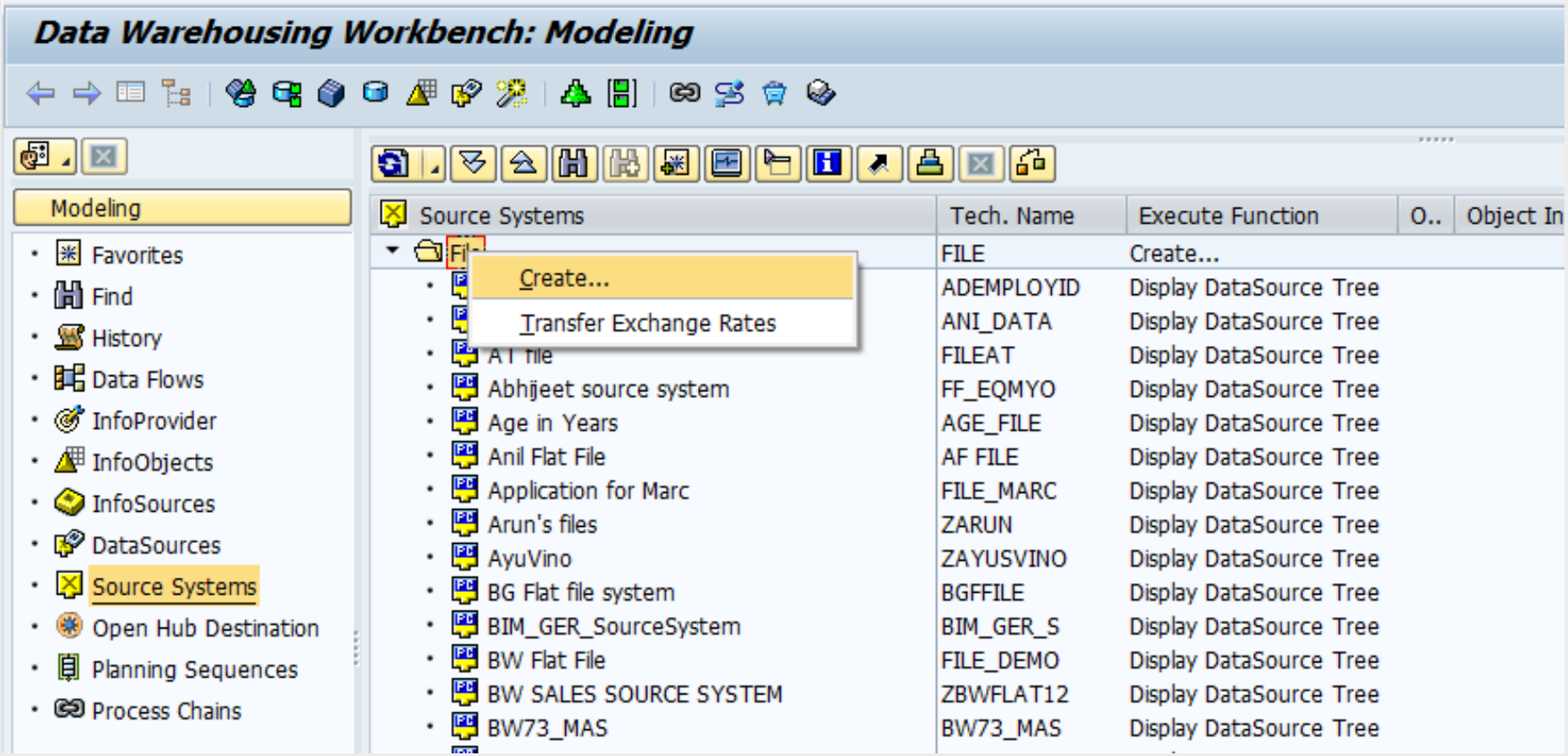
7. Non-SAP systems for which data and metadata is transferred using staging BAPIs

8. Data services is a direct connection between SAP NetWeaver BW 7.3 and SAP Business Objects Data Integrator . It enables you to establish connections between SAP NetWeaver BW and non-SAP systems, and trigger the generation of metadata and data flows.

Source Systems Types Shown in RSA1

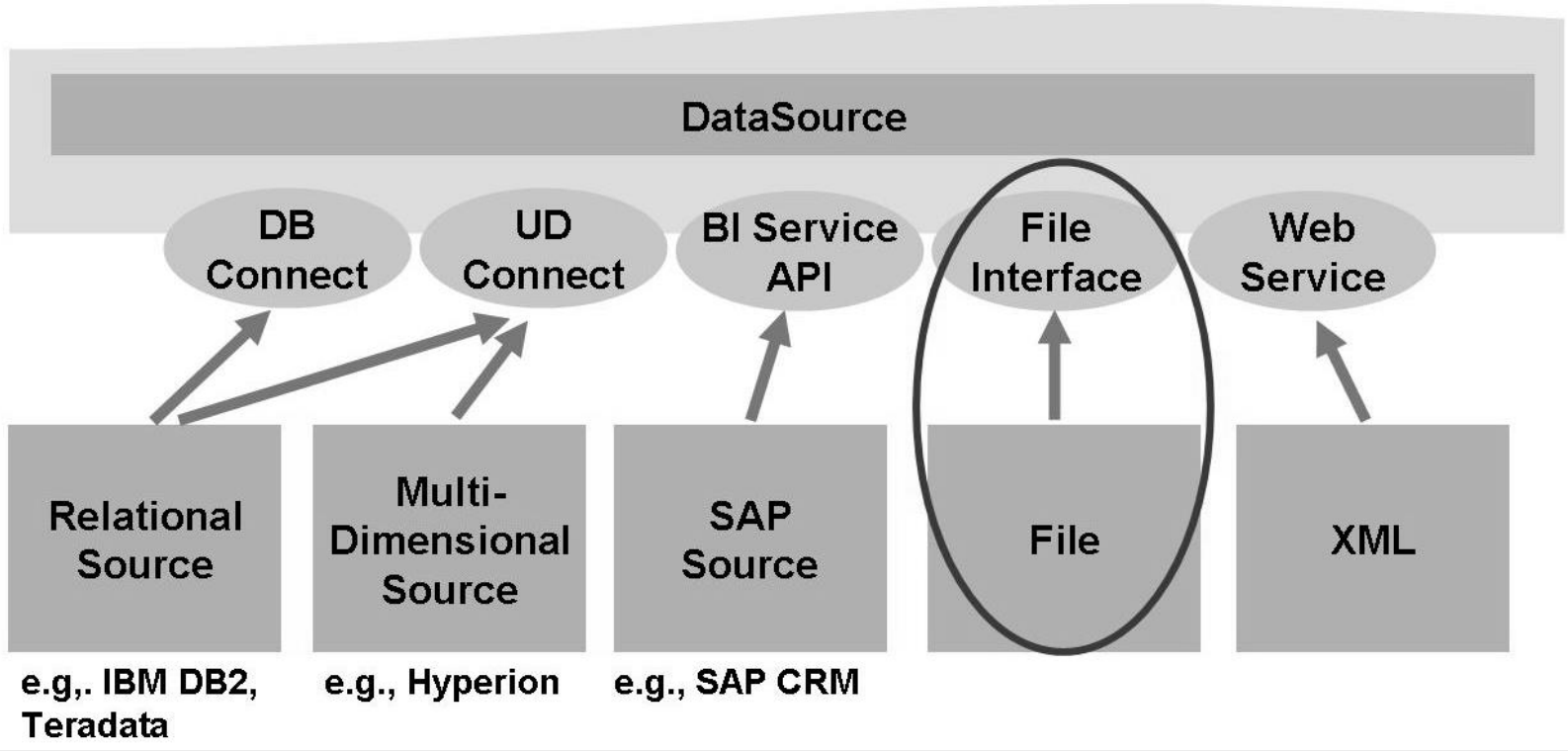


Creating a Flat File Source System



DataSource

DataSources Based on Flat Files



DataSource (RSDS)

- DataSources are used for extracting data from a source system and for transferring data into the BI.
- DataSources make the source system data available on request to the BI in the form of the (if necessary, filtered and enhanced) extraction structure.
- The DataSources subdivide the data that is provided by a source system into self-contained business areas.

DataSource (RSDS)

➢In SAP source systems, the DataSources are assigned to applications, for example HR (Human Resources).

The advantage here is that you are able to specifically access data evaluation for this application.

➢The information required for the extraction process, that is, the BI-relevant properties of the DataSource, is

replicated in the BI by comparing metadata.

DataSource Types

1. DataSources for Transaction Data
2. DataSources for Master Data
    i. for Attributes ***_ATTR***
    ii. for Texts ***_TEXT***
    iii. for Hierarchies ***_HIER***

Example: Refer to RSA5 in SAP R/3 for DataSources

DataSource MetaData (RSA2)

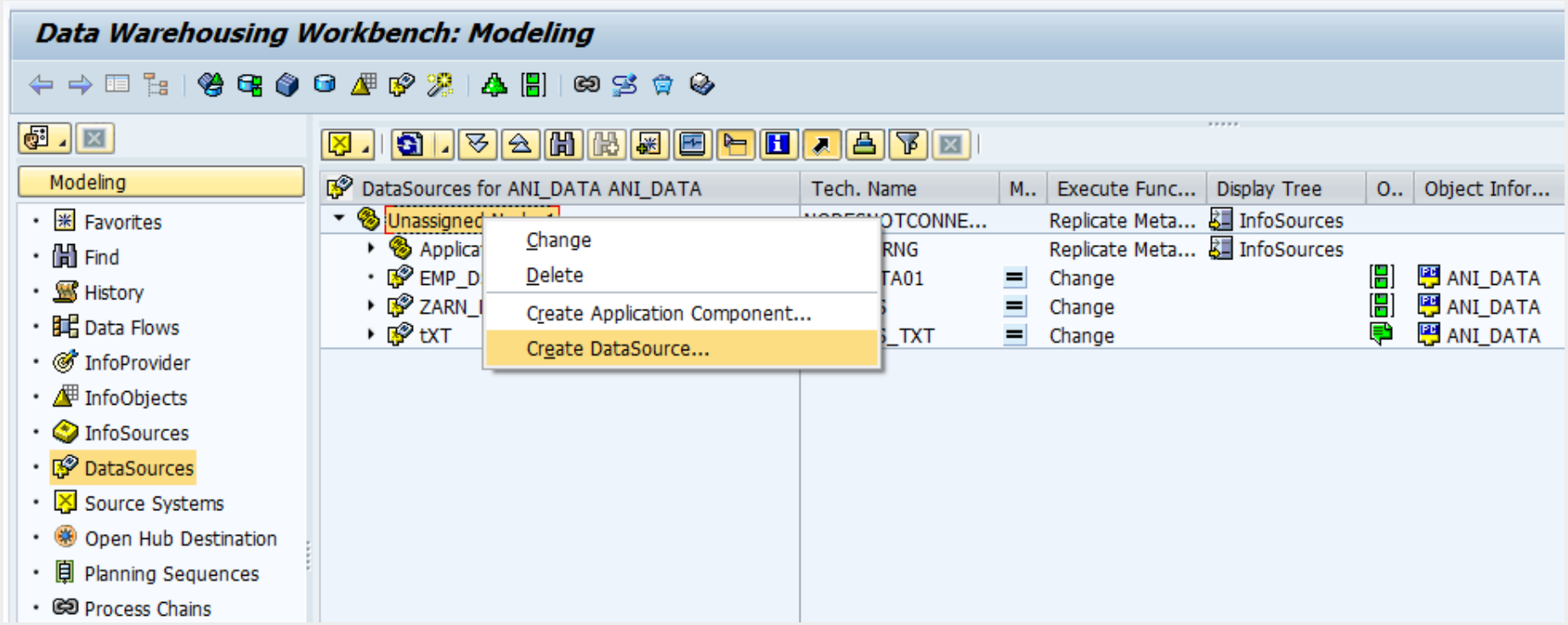
➢DS-Metadata: Describes the data to be extracted and how this data is to be extracted.

➢Metadata is Source-system Specific:

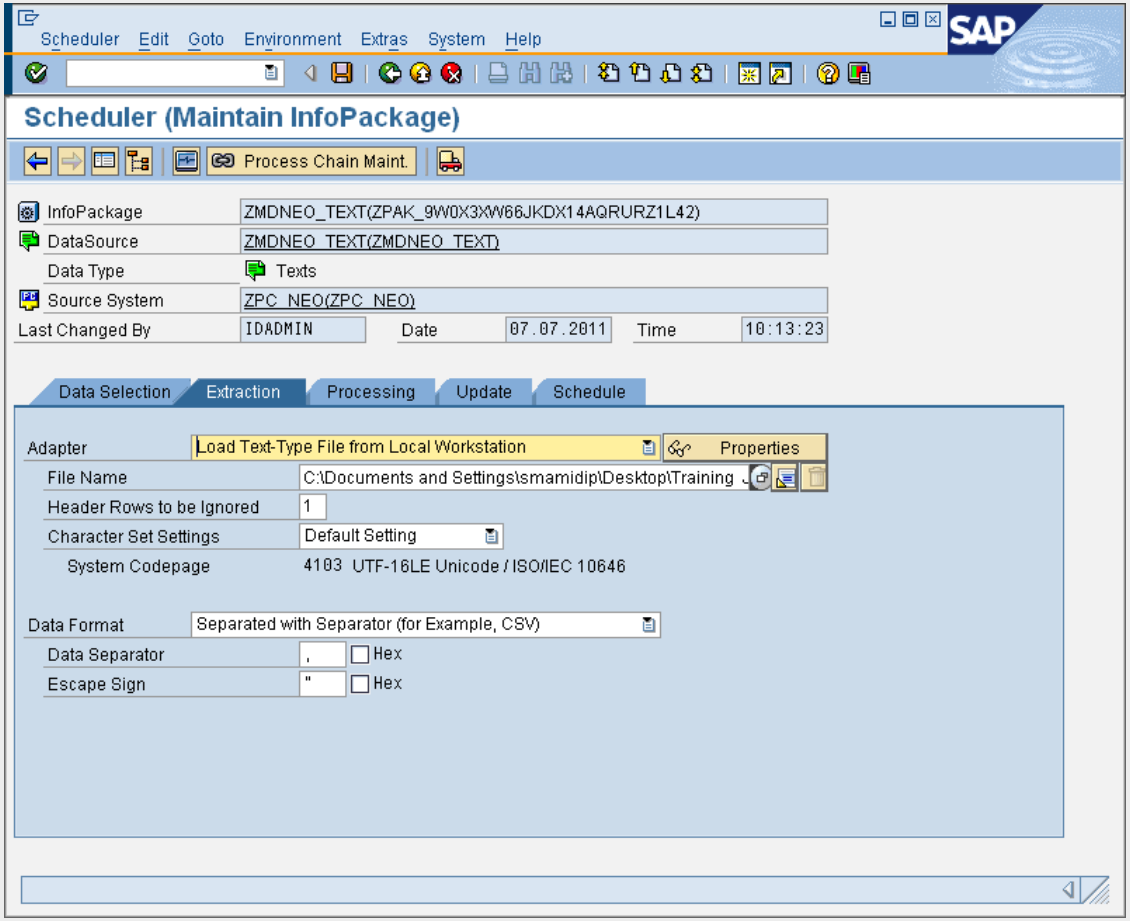
- A number of logically related fields that are offered in a flat structure
- Extraction structure, for transferring data to BI.
- Application Component
- Extraction method
- Extractor
- Delta process
- Transfer method

Example: Refer to RSA2 for DataSources

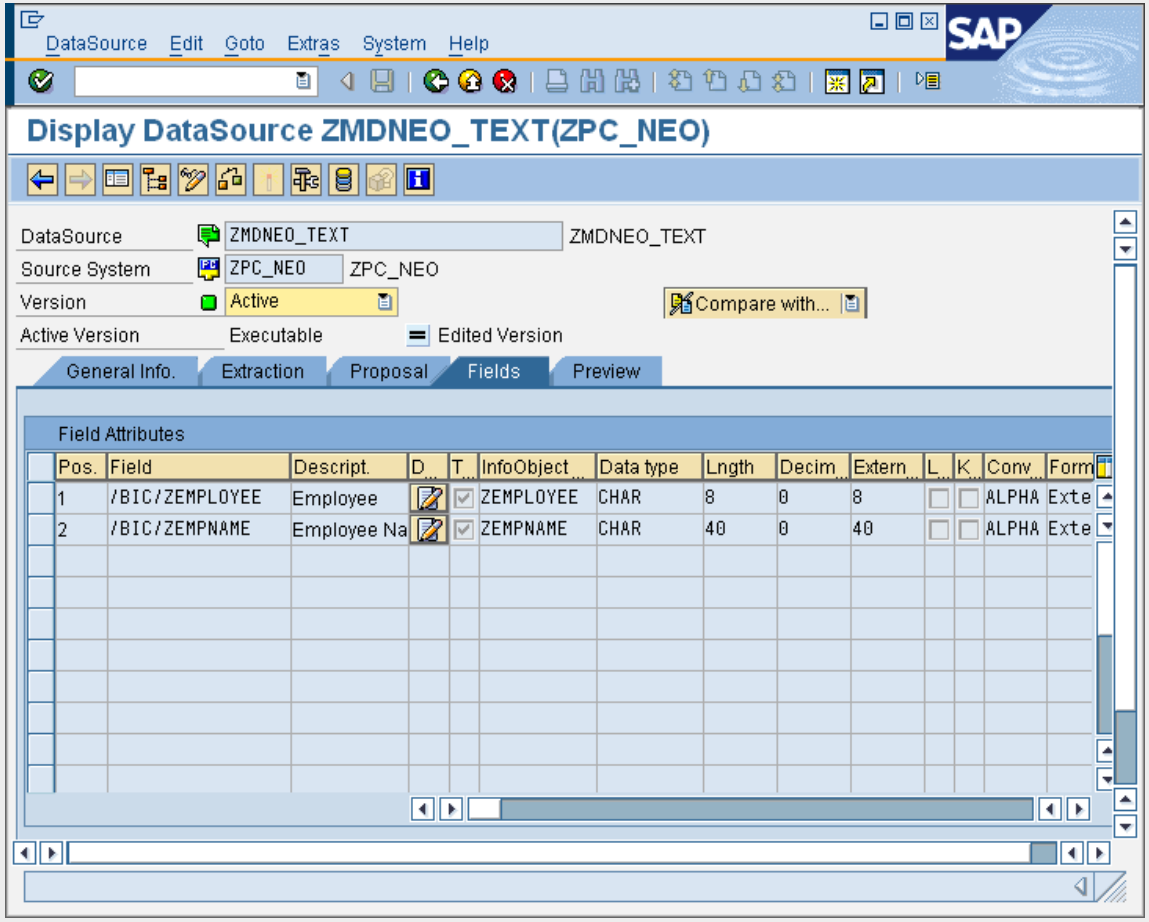
Create a DataSource



File System DataSource: Extraction Tab



DataSource: Fields



PSA

PSA (Persistent Staging Area)

- PSA is the inbound storage area for data from the source systems in BI system.
- The requested data is saved, unchanged from the source system
- Requested data is stored in the DataSource structure format in transparent, relational database tables in the BI system.
- The data format remains unchanged, meaning that no summarization or transformations take place, as is the case with InfoCubes

InfoPackage

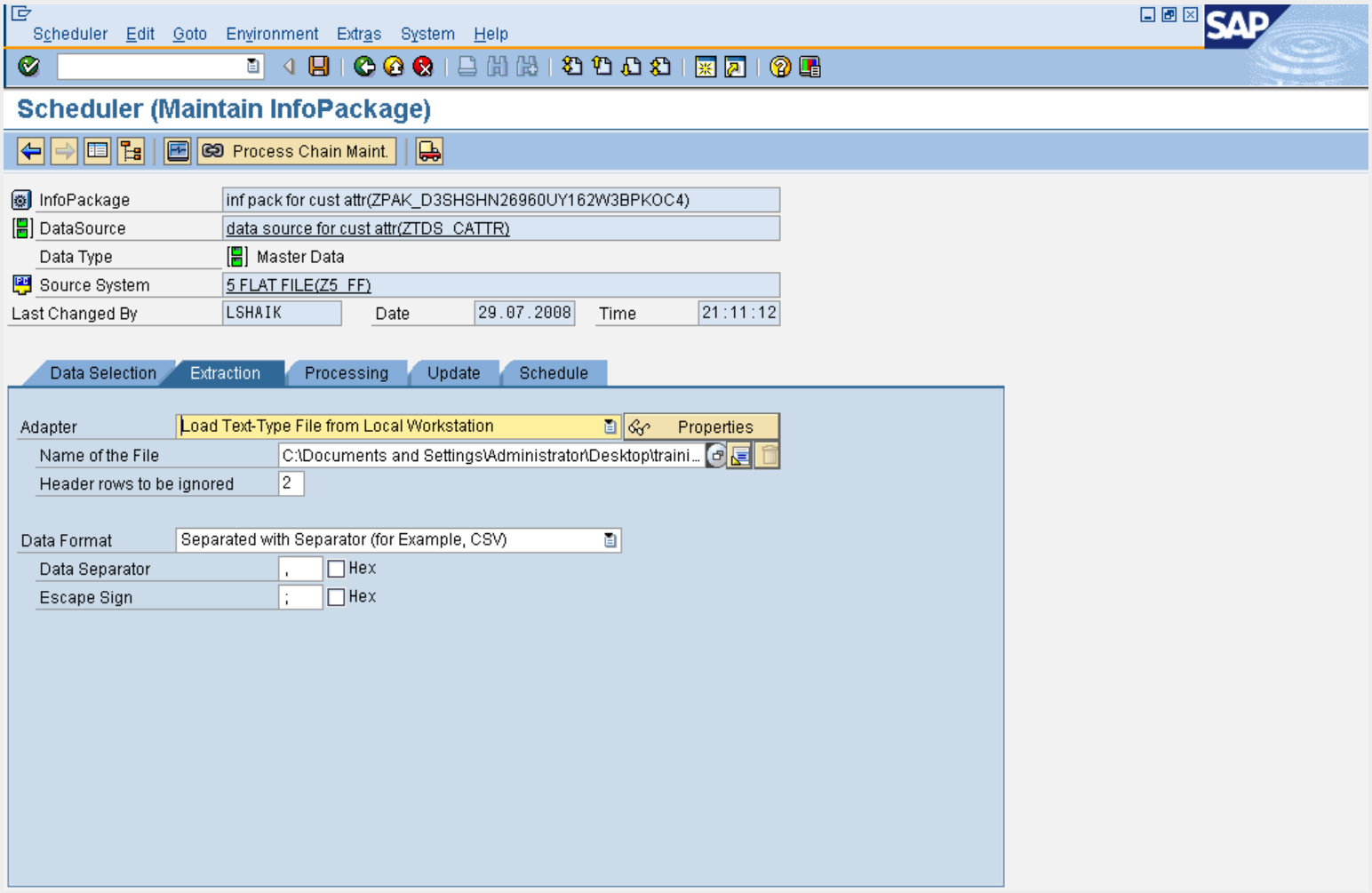
Infopackage

➢Infopackage is used to load data from source system to PSA in BI system.

➢Following are the different tabs of an Infopackage

- Data Selection : Restricts the data to be loaded
- Extraction : For selection of flat files
- Processing : PSA staging options
- Update : Full / Delta / Initial Load
- Schedule : Foreground or Background processing

InfoPackage



Transformation

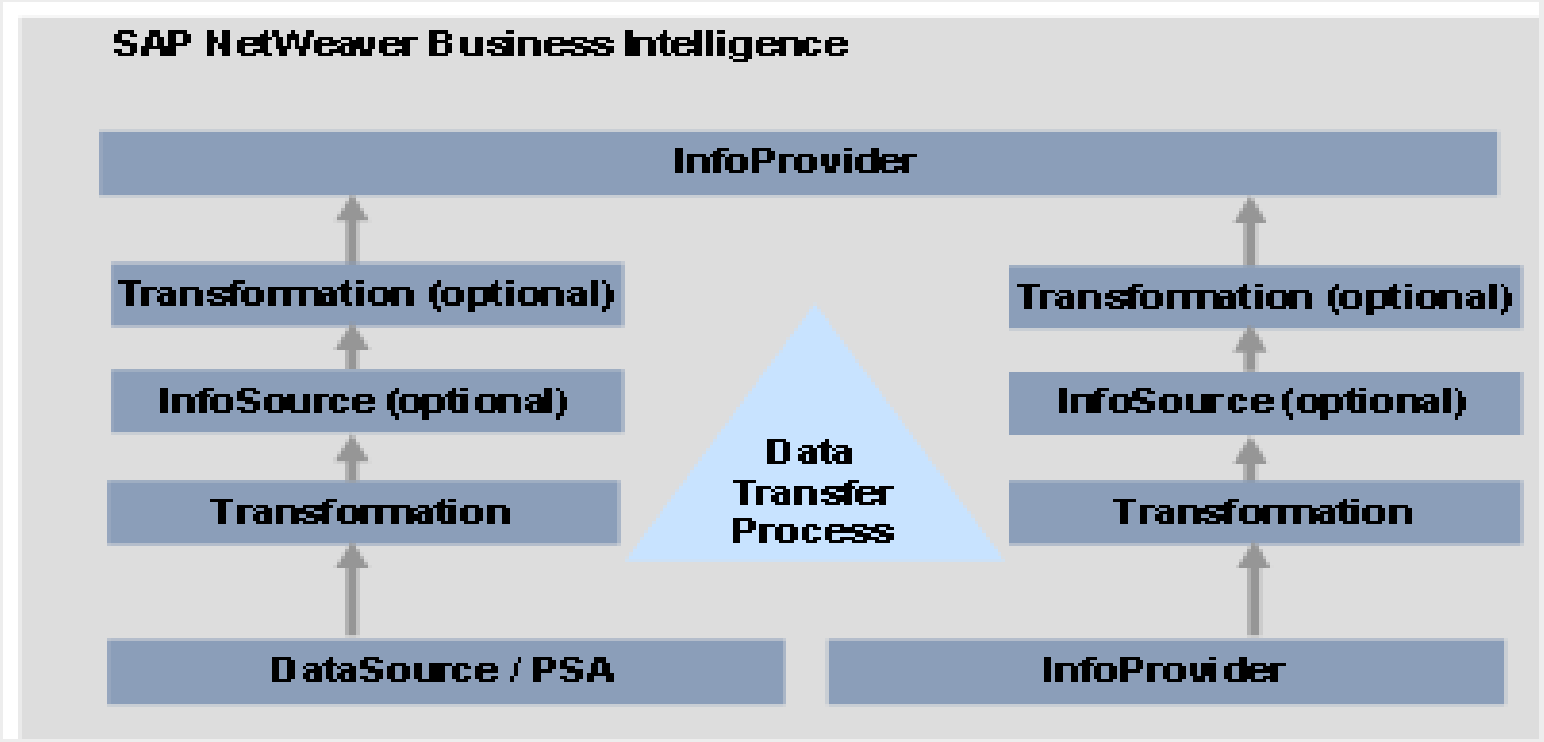
Transformation - Definition

➢The process that facilitates to consolidate, cleanse, and integrate data.

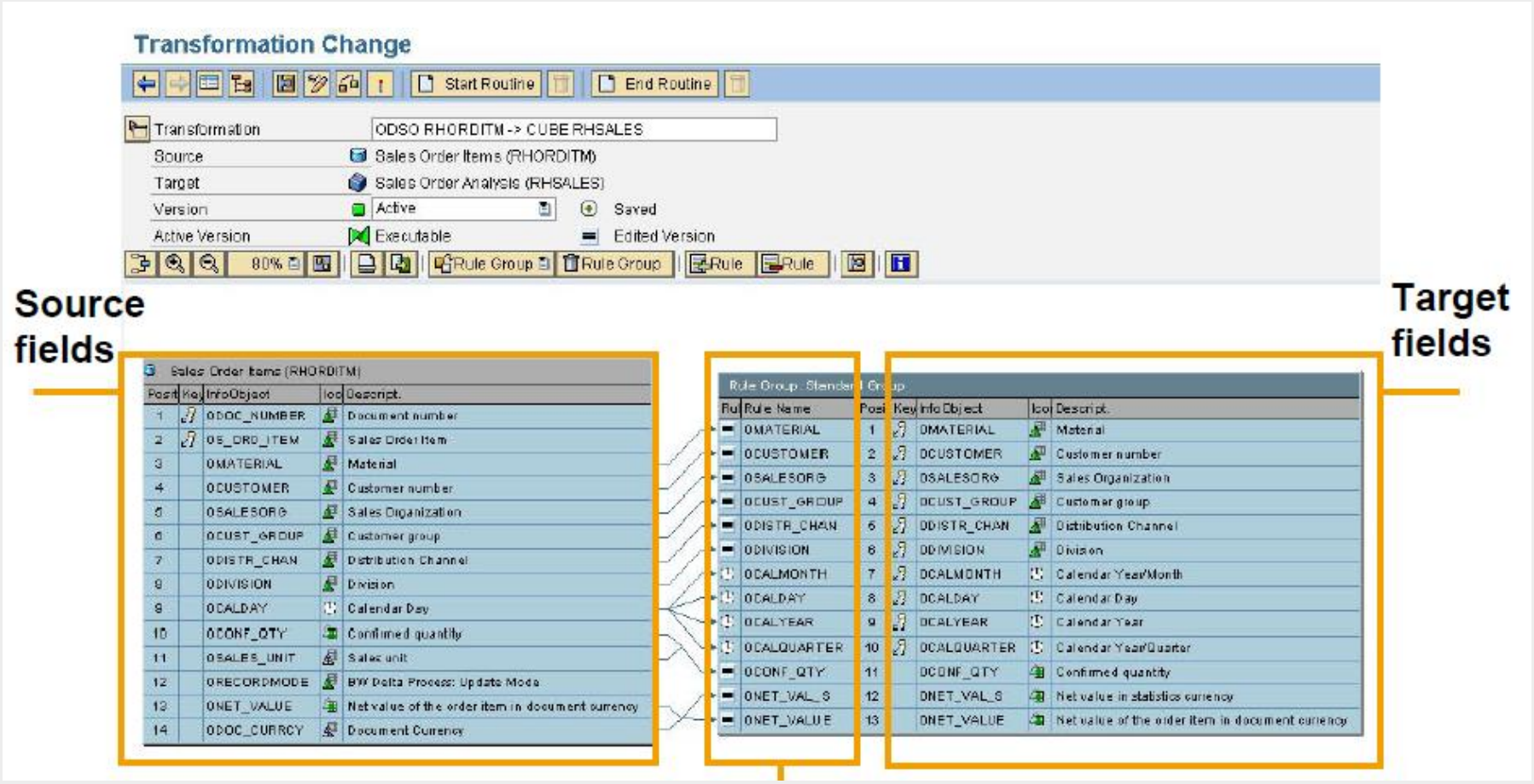
- Semantically synchronize data from heterogeneous sources.
- A transformation converts the fields of the source into the format of the target.

Transformation in the Flow

- Improved performance, flexibility and usability
- Unit Conversion Capabilities for unit conversion during data load (and reporting)



Transformation GUI



Transformation Rules

•Transformation rules: Transformation rules map any number of source fields to at least one target field. You can use different rules types for this.

•Rule type: A rule type is a specific operation that is applied to the relevant fields using a transformation rule.

•Transformation type: The transformation type determines how data is written into the fields of the target.

•Rule group: A rule group is a group of transformation rules. It contains one transformation rule for each key field of the target. A transformation can contain multiple rule groups

Transformation - Rule Types

1. Direct Assignment
2. Constants
3. Formula
4. Reading Master Data
5. Reading DSO
6. Routine
7. Time Update
8. Initial
9. Unit of measure conversion & Currency Translation

Transformation(update) Type

Def: To control how a key figure/data field is updated to the InfoProvider.

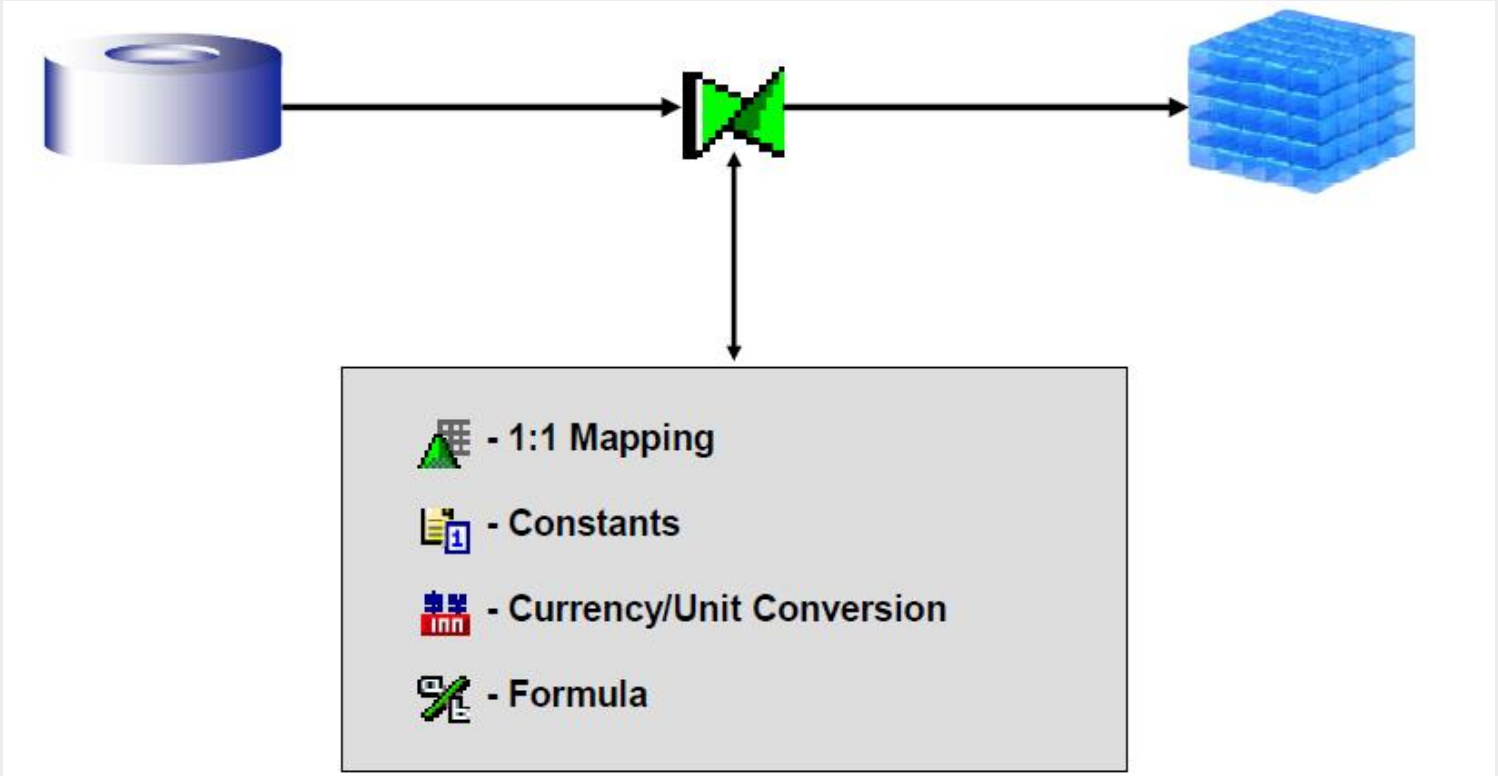
BI

InfoProvider / Update Type	InfoCube	Data Store Object		Char.-InfoObject
	Key Figure	Data Field		Attribute / Text
		Char.	Key Figure	
Addition	X	-	X	-
Maximum	X	-	X	-
Minimum	X	-	X	-
Overwrite	-	X	X	X
No Update	X	X	X	X

Scenarios for Transformation

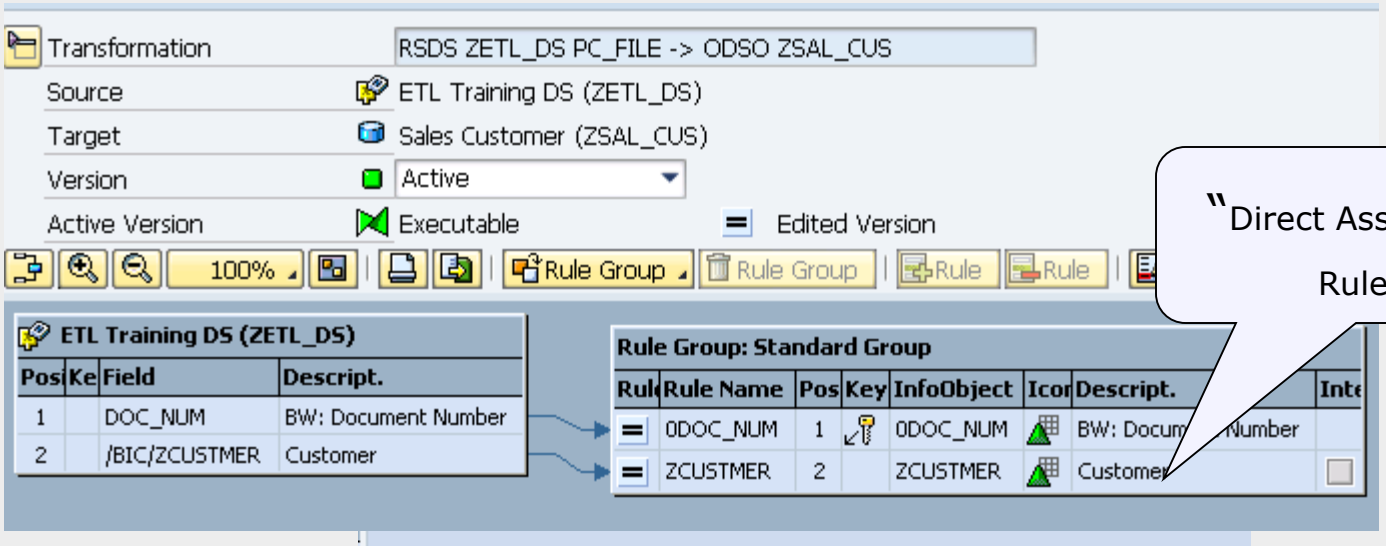
- On the Fly Transformation
- Data Enrichment – Cross Reference (LOOKUPs)
- Conditional Update

On the Fly Transformation



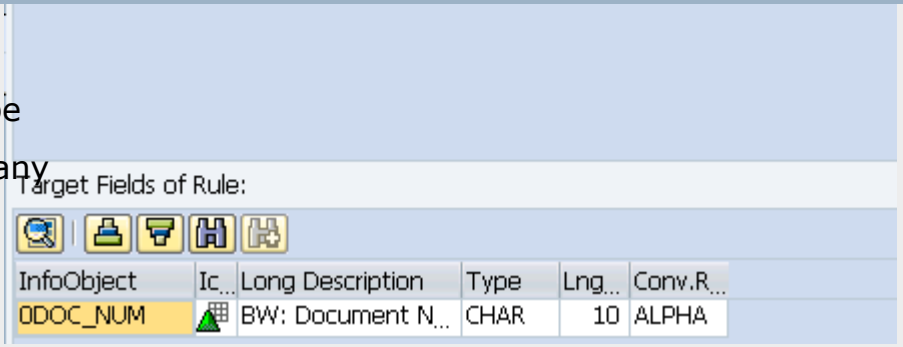
Transformation on record level by using standard transformation features of BI.
No master data or InfoProvider read.

1:1 mapping (Direct Assignment)

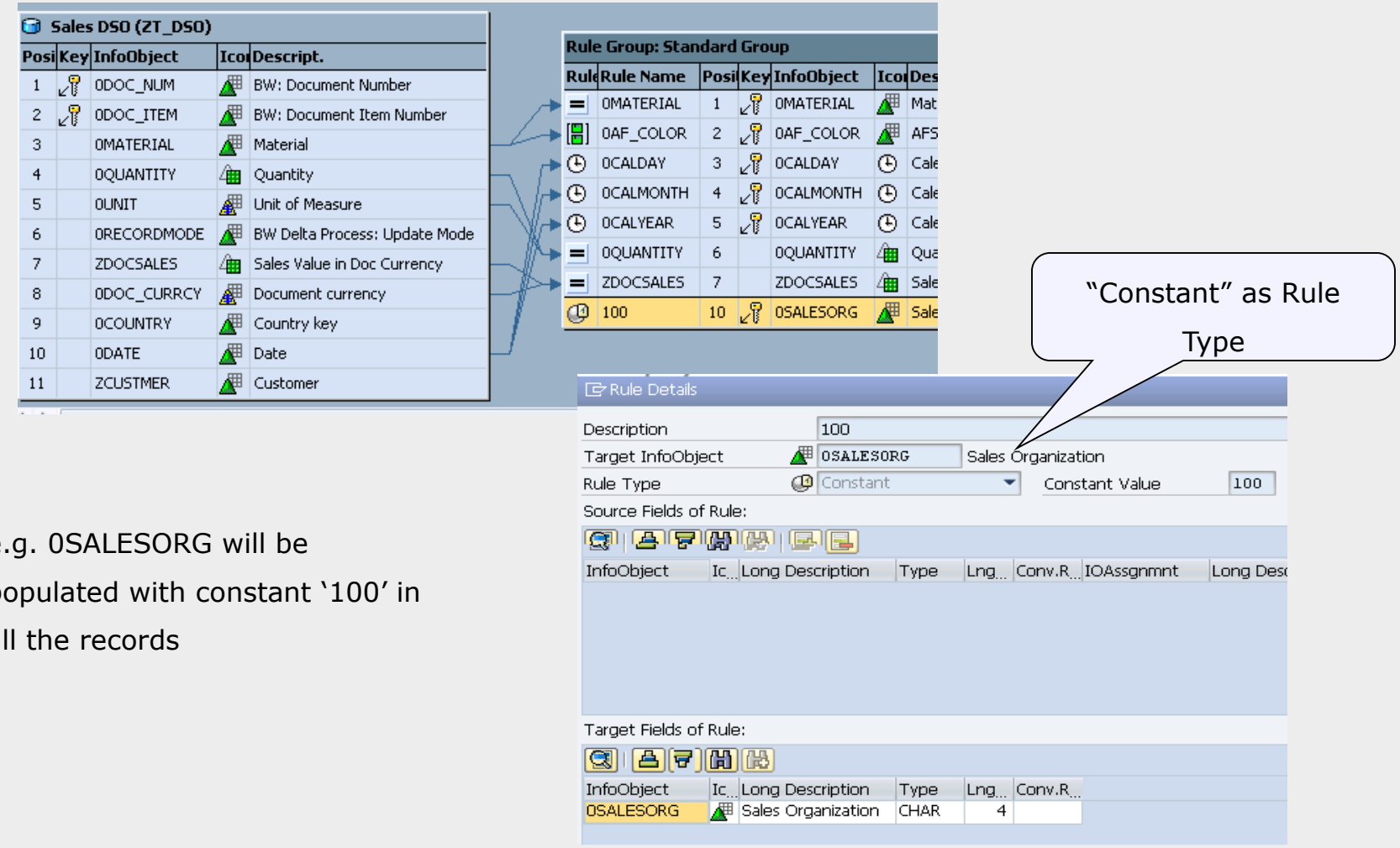


"Direct Assignment" as Rule Type

Field Values will be updated without any transformation



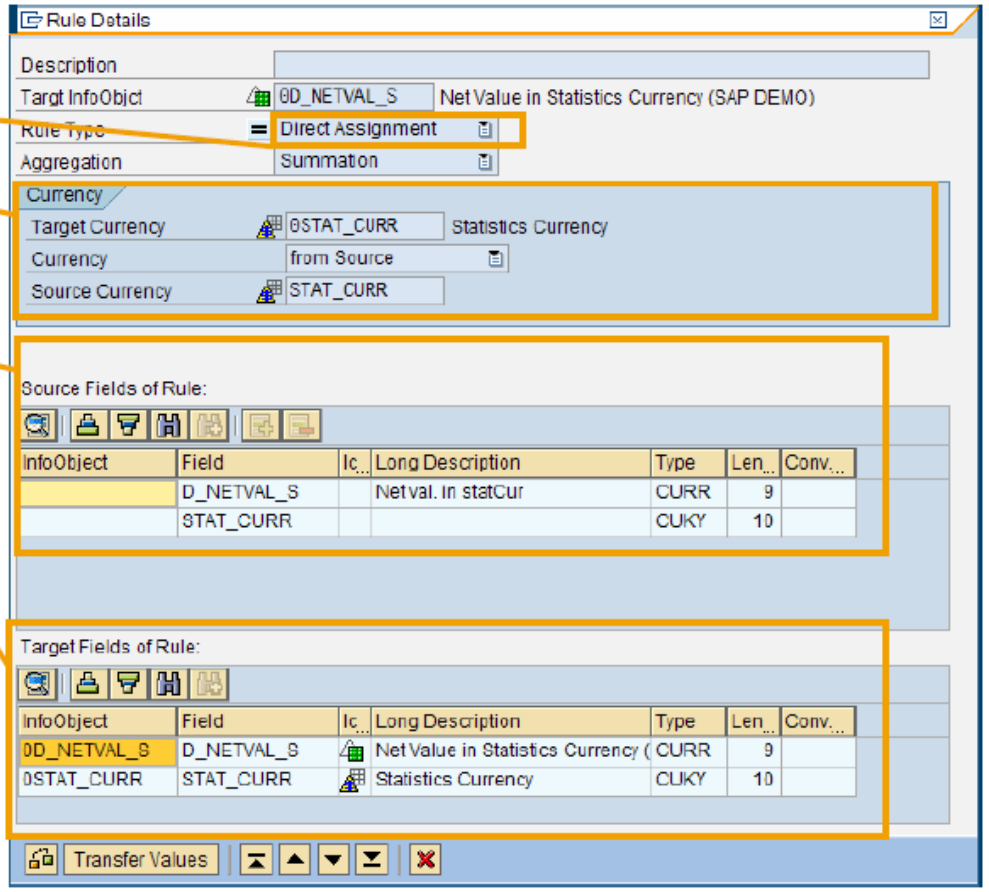
Constants



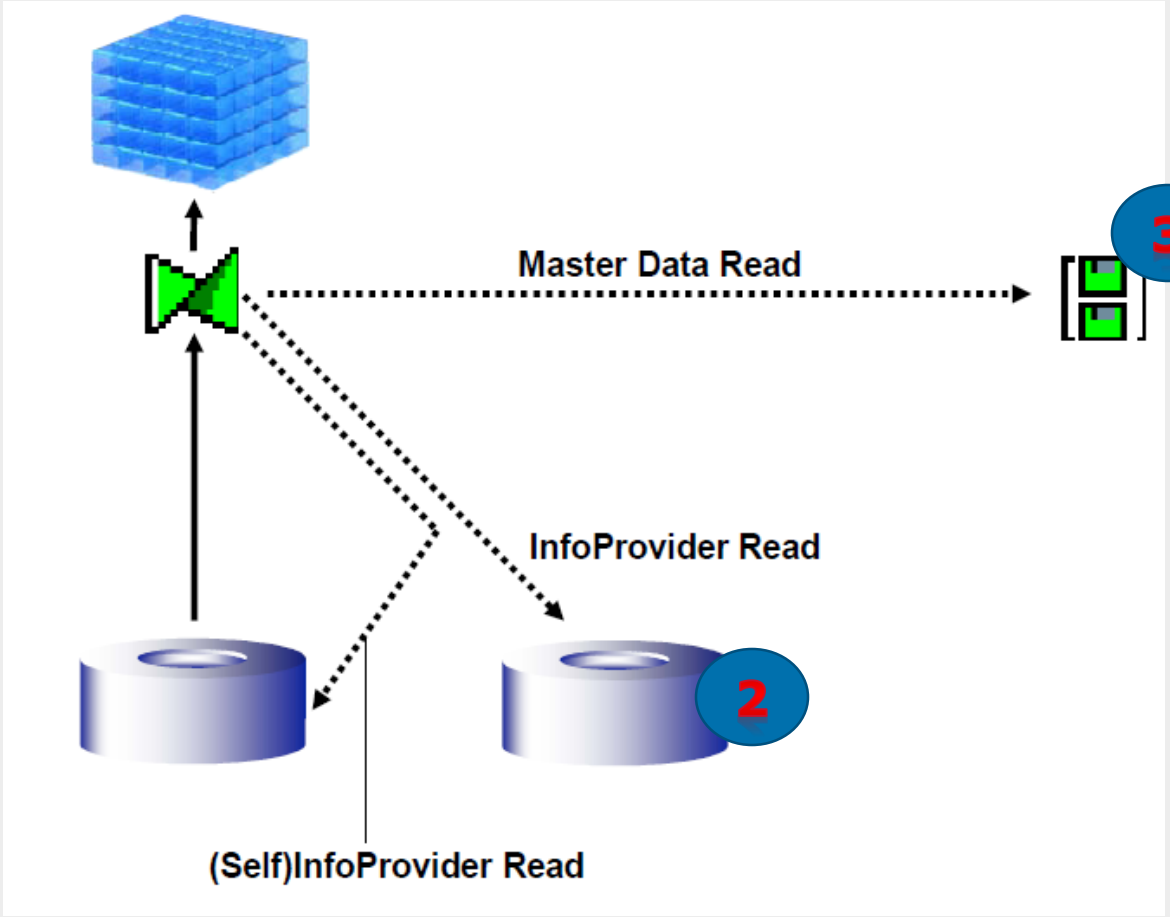
e.g. 0SALESORG will be populated with constant '100' in all the records

Currency/Unit Conversion

- **Information on**
  - **Rule type**
  - **Currency/ Unit Conversion**
  - **Source fields**
  - **Target fields**

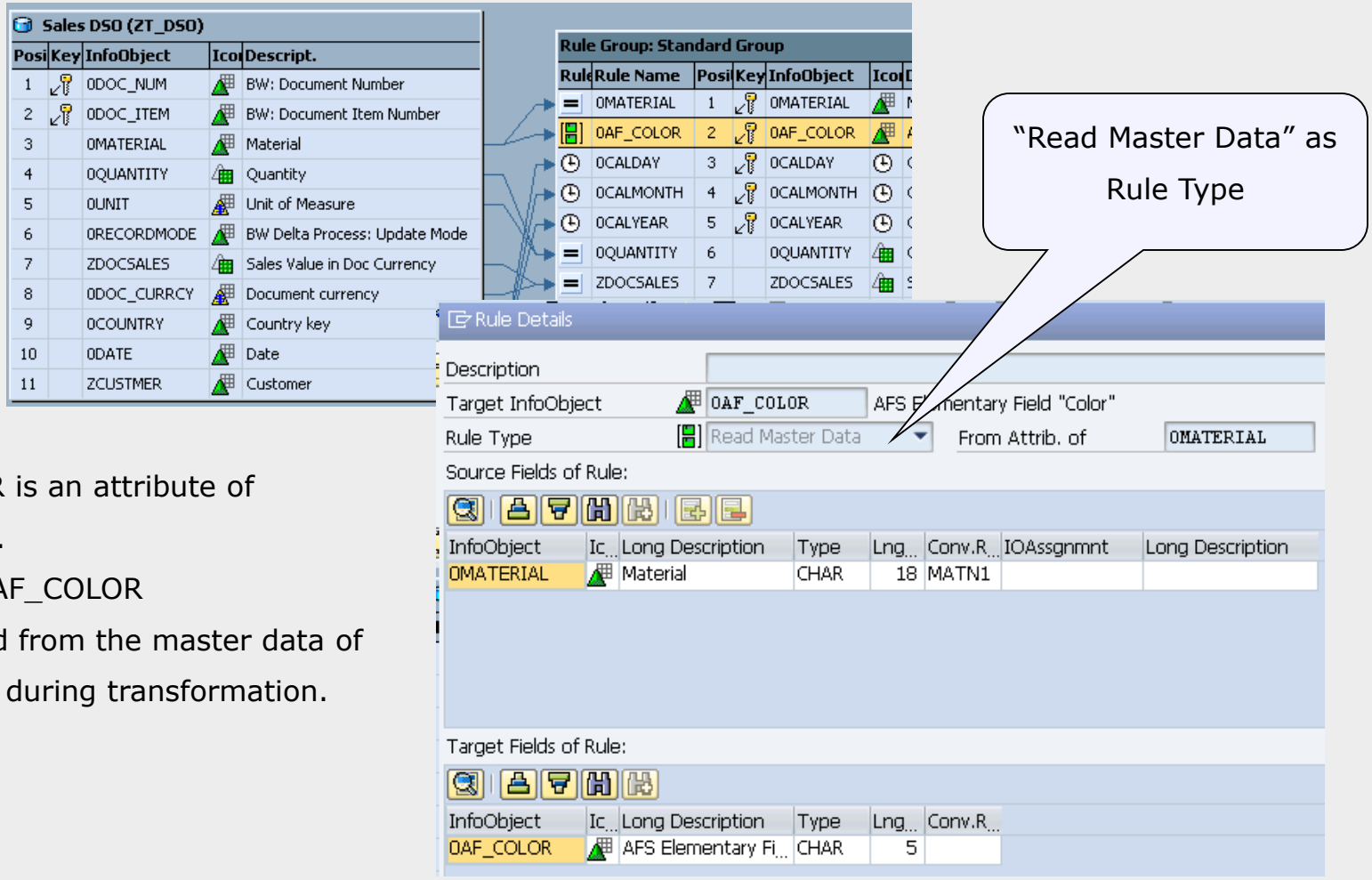


Data Enrichment - Cross Reference (LOOKUPs)



Read Master Data

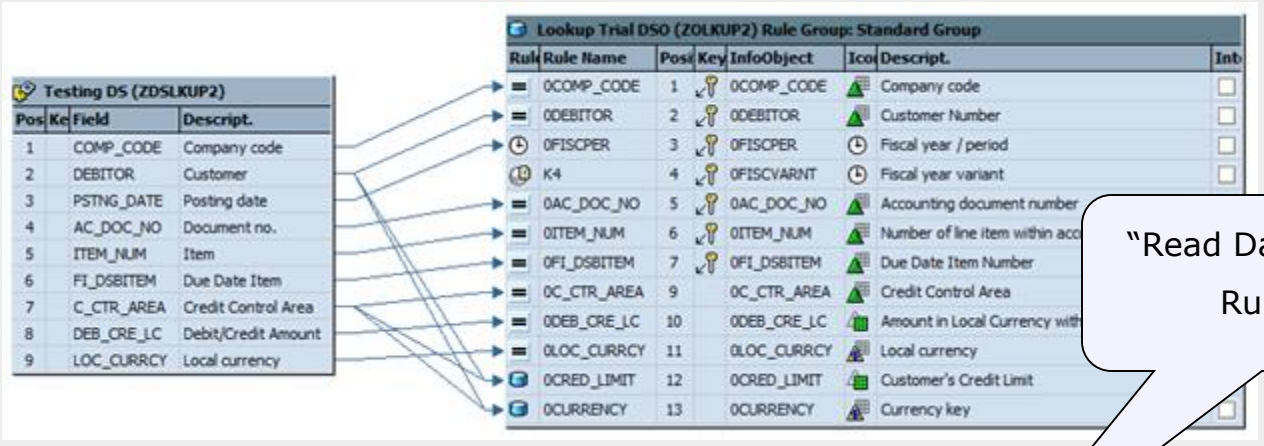
Ade.g.



0AF_COLOR is an attribute of 0MATERIAL.

Value for 0AF_COLOR will be read from the master data of 0MATERIAL during transformation.

Read Data Store

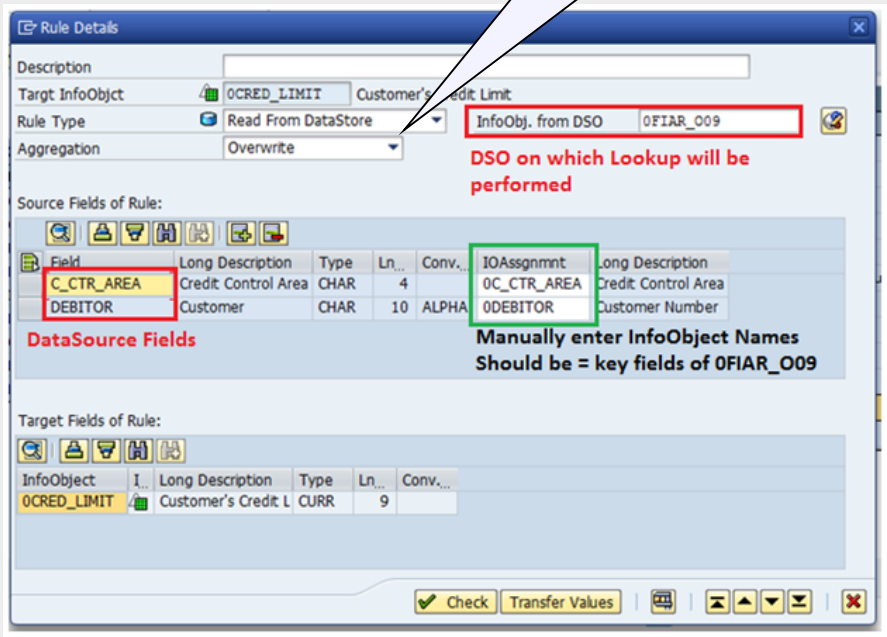


"Read Data Store" as Rule Type

Eg:

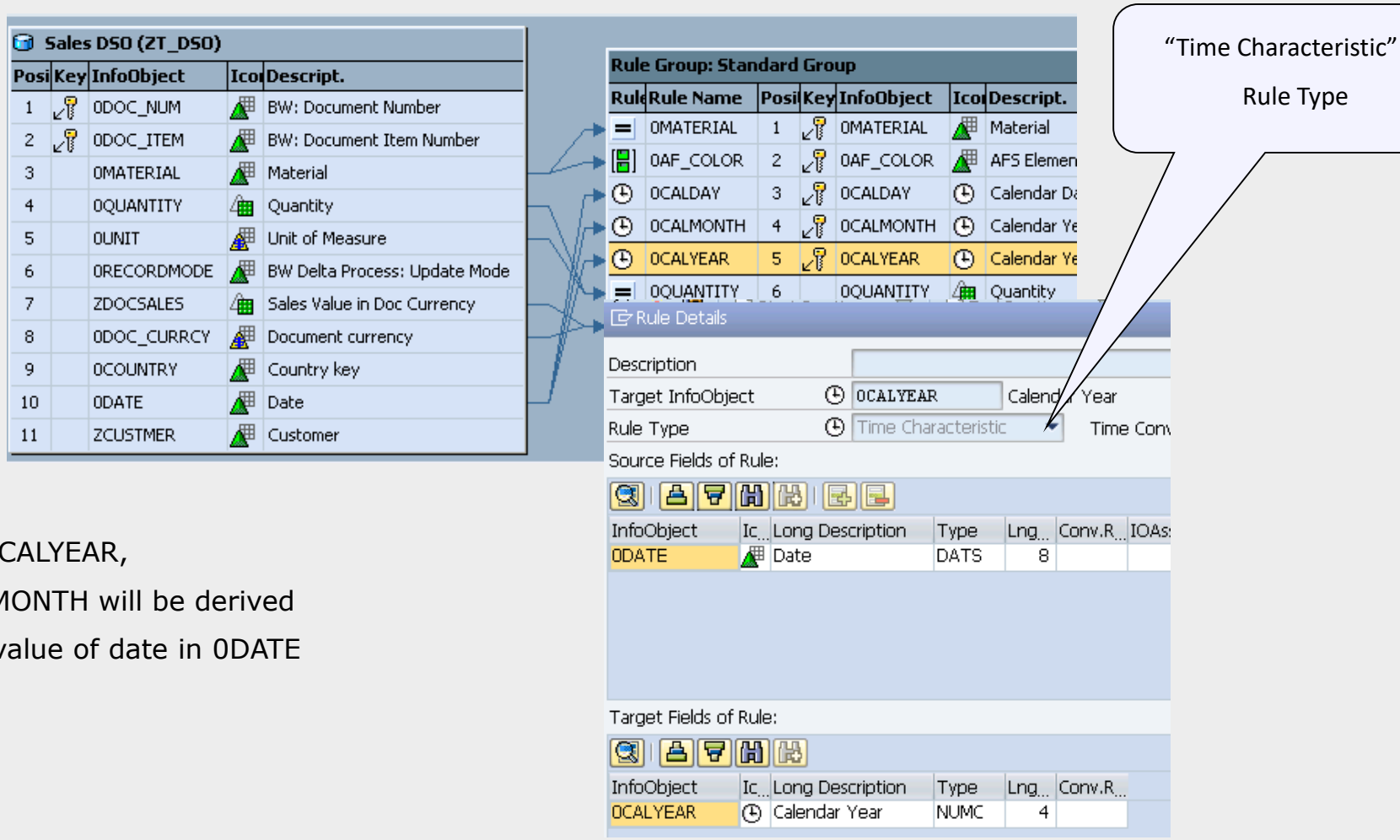
Value of 0CRED_LIMT will be read from DSO 0FIAR_O09 during the transformation based on C_CTR_AREA and DEBITOR

This feature is most useful if you have simple lookups, for instance get Field X from DSO Y based on the lookup field Z and write it out in field X of the target.



Time Characteristic

Ad



e.g. 0CALYEAR,

0CALMONTH will be derived

from value of date in 0DATE

Transformation Routines

➢The different types of routines that are available within Transformations are listed below:

- The Start Routine
- Routine for updating Key Figures
- Routine for updating Characteristics
- End Routine
- Expert Routine
- Rule Group

Transformation Routines

➢Routines in a Transformation are executed in the following sequence:

- The Start Routine
- Transformation Rules and routines for updating Characteristics / Key Figures
- End Routine

Start Routine

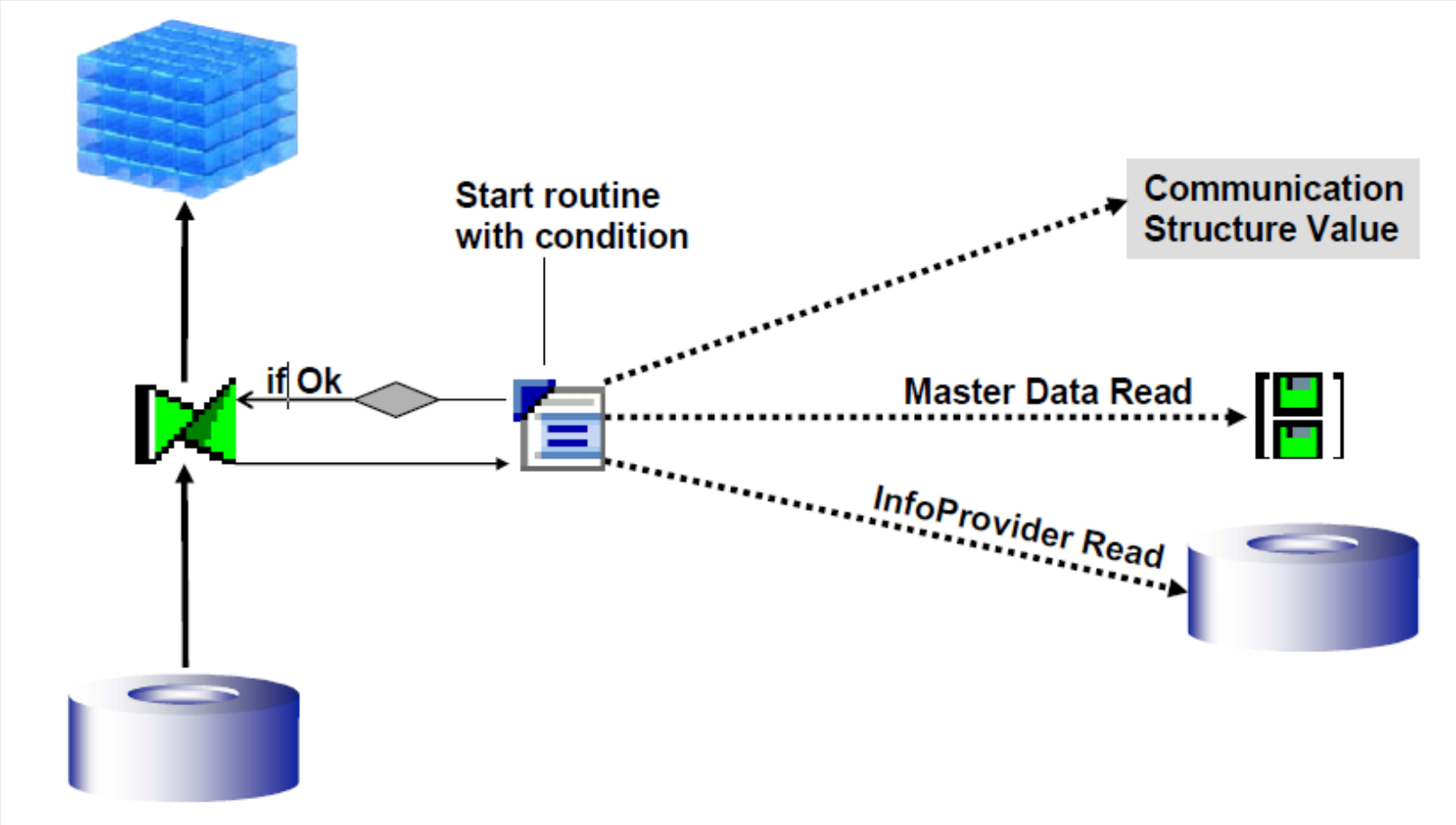
➢Use

- Preparation of data before transformation
- Package – Based

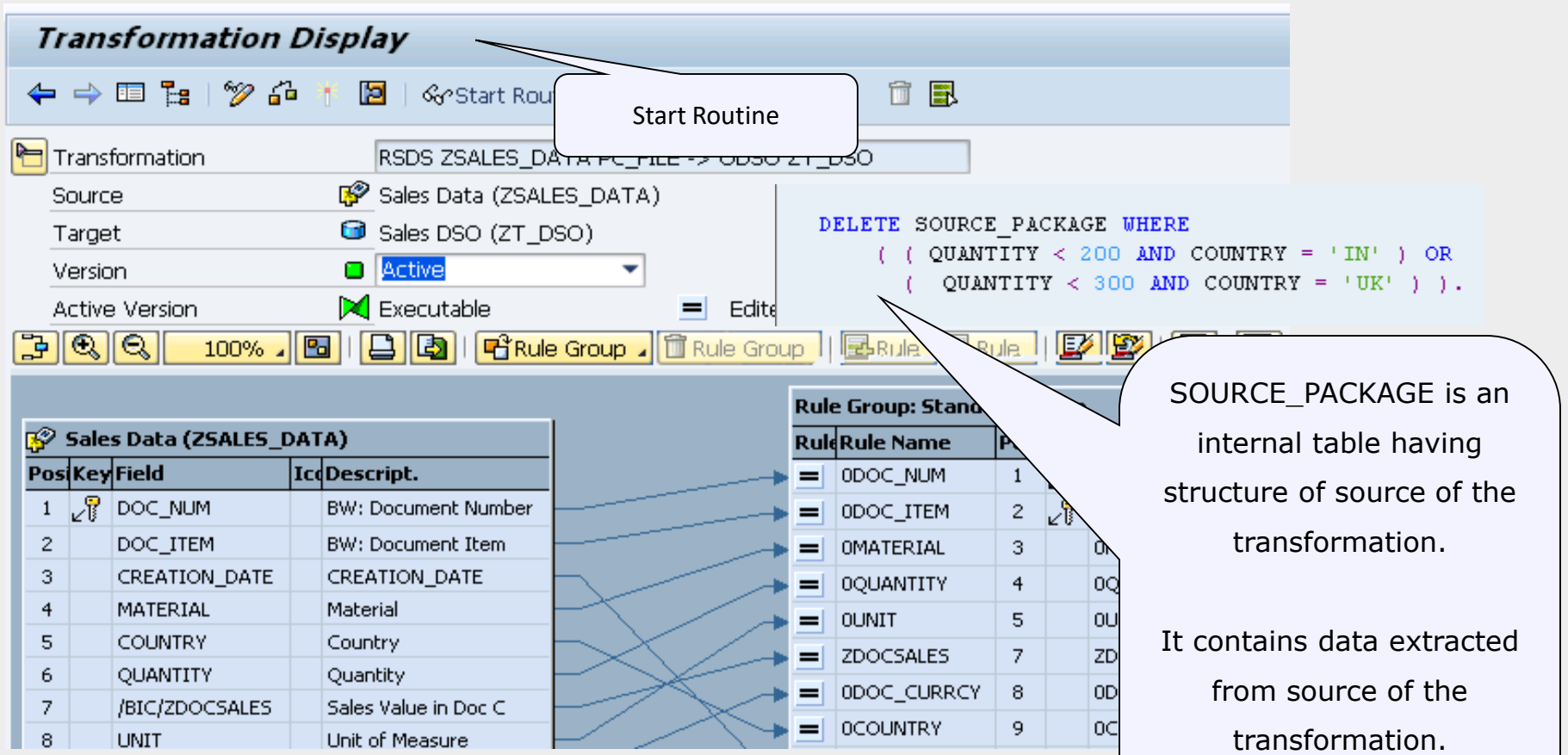
➢Example:

- deletion of records that are not required for updating.(helps to avoid processing / transformation of records that are not required to be updated in the target)
- Performance: Buffering Internal tables that can be used for transformation rules (rather than reading the data base tables one by one).

Conditional Update using Start Routine



Transformation (Start Routine)



SOURCE_PACKAGE is an internal table having structure of source of the transformation.

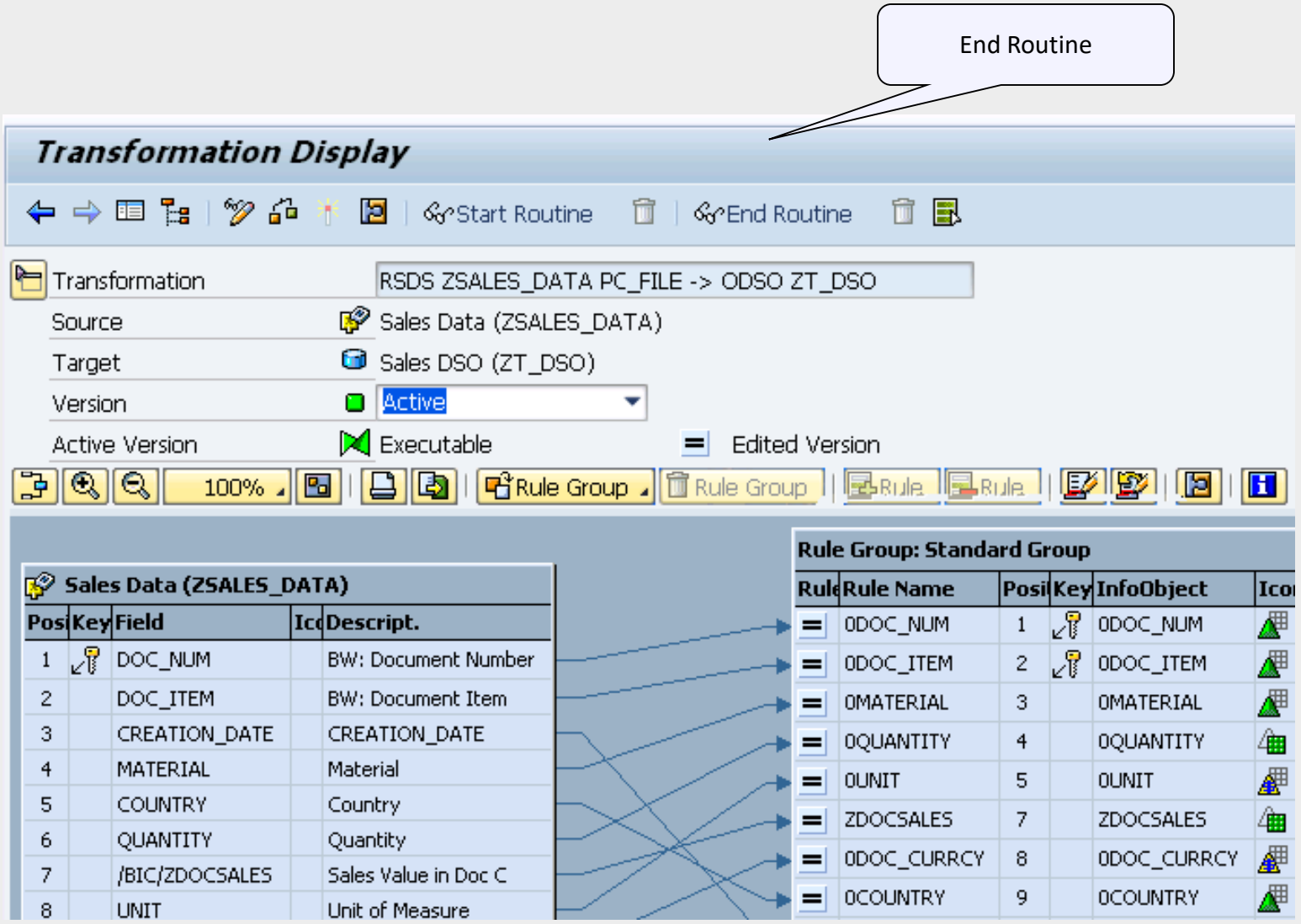
It contains data extracted from source of the transformation.

End Routine

- Use
  - Post-preparation of data after transformation
  - Package based

- Example:
  - Deletion of records after transformation that are not required for updating
  - Validation checks of records after updating.
  - E.g. Customer no. is in target but not in source. Records belonging to customer 'CUST10' should not be updated to target.
  - Customer no. will be derived during end routine processing and later the records will be deleted based on that.

Transformation (End Routine)



End Routine

```
IF RESULT_PACKAGE[] IS NOT INITIAL  .

  SELECT DOC_NUM
     /BIC/ZCUSTMER
  FROM /BIC/AZSAL_CUS00
  INTO TABLE IT_CUST
  FOR ALL ENTRIES IN RESULT_PACKAGE
  WHERE DOC_NUM = RESULT_PACKAGE-DOC_NUM .

ENDIF.

SORT IT_CUST ASCENDING BY DOC_NUM.

LOOP AT RESULT_PACKAGE INTO  WA_RESULT_PACKAGE .

  lv_tabix = sy-tabix.
  READ TABLE IT_CUST INTO wa_cust WITH KEY
         DOC_NUM = WA_RESULT_PACKAGE-DOC_NUM BINARY SEARCH .
  WA_RESULT_PACKAGE-/BIC/ZCUSTMER = wa_cust-ZCUSTMER .
  MODIFY RESULT_PACKAGE FROM WA_RESULT_PACKAGE INDEX LV_TABIX .

ENDLOOP .

DELETE RESULT_PACKAGE WHERE /BIC/ZCUSTMER = 'CUST10' .
```

RESULT_PACKAGE is an internal table having structure of target of the transformation.

It has to contain the records that are to be updated to target.

1.First populate the field that is not in the source but added in the target (Customer in this example) through cross reference from other infoprovider DSO ZSAL_CUS in this example.

2.Delete the records that are not required to be updated based on selection criteria of that field.

Expert Routine

- Use

- Transformations that cannot be expressed declaratively for the functional or performance reason

- Example:-

- Performance: reading several database tables can be implemented faster when knowing the application logic.

- Pivoting: transpose a wide data record into several smaller records -> can be easily implemented using the expert routine.

DTP

Data Transfer Process

➢Purpose: To transfer data within BI from a persistent object to another object in accordance with certain transformations and filters.

- In this respect, it replaces the data mart interface and the Info Package. As of SAP Net Weaver 2004s, the Info Package only loads data to the entry layer of BI (PSA).

➢Features:

- Processing more transparent.
- Optimized parallel processing improves the performance of the transfer process
- Separate delta processes for different targets and
- different filter options between the persistent objects on various levels.
  - For example, you can use filters between a Data Store object and an Info Cube.

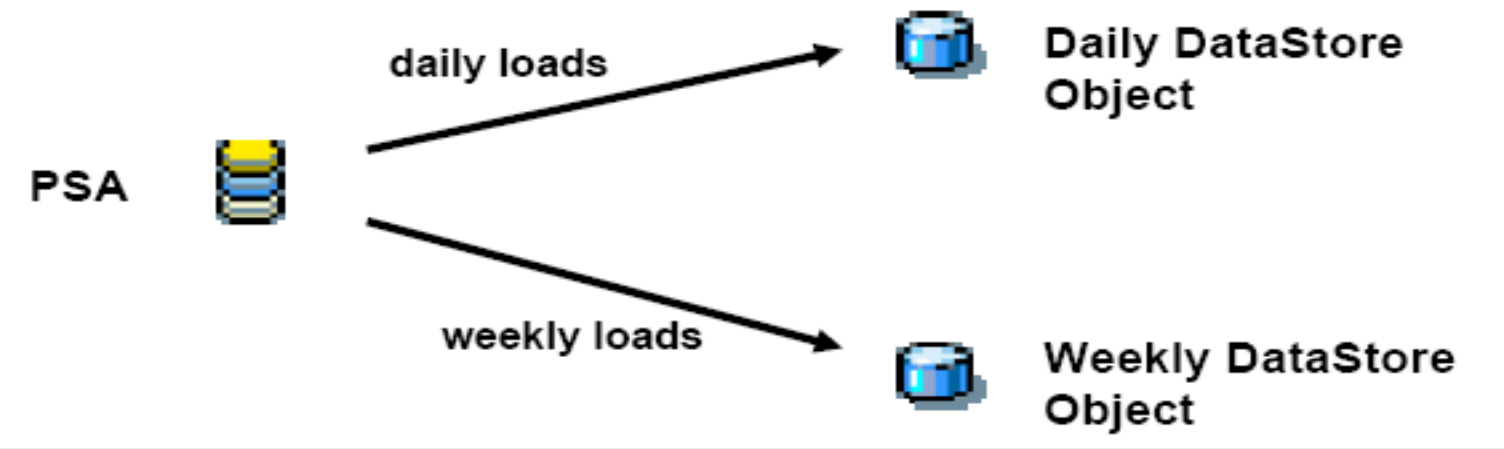
➢Data transfer processes are used for

- Standard data transfer,
- Real-time data acquisition and
- Accessing data directly.(Used for Virtual Providers.)

Data Transfer Process

➢The most important advantage of DTP is delta logic can be separately handled for separate data targets

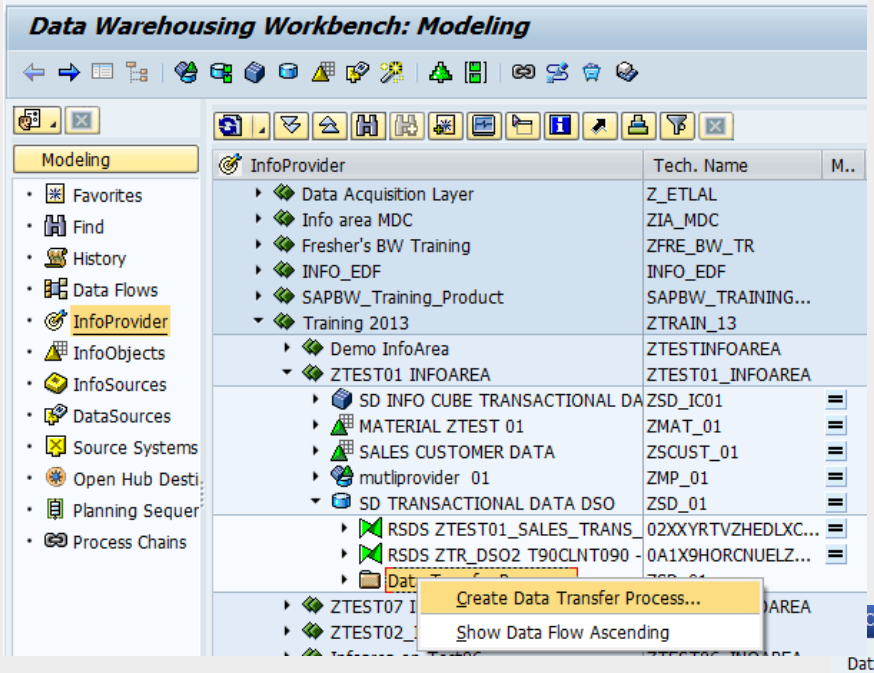
- Example for separation for delta logic
- Delta logic is a part of DTP
  - One Source PSA
  - Two targets : One DSO keeping daily data and other one keeping weekly data

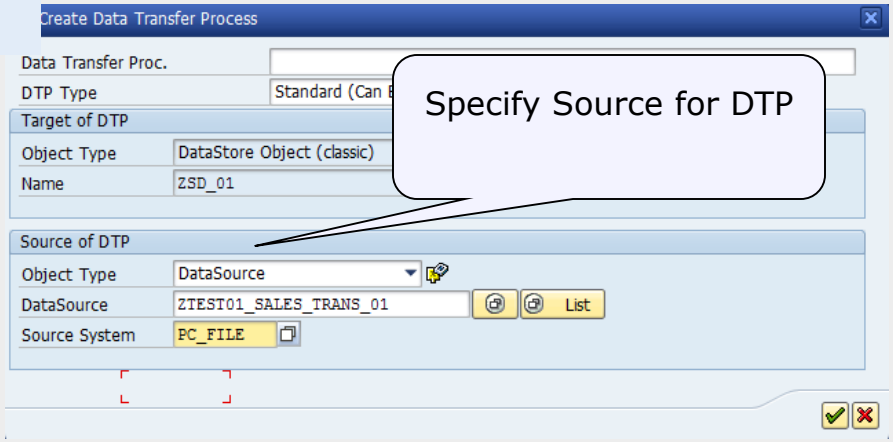


Benefits of Data Transfer Process

- Loading data from one layer to others except Info sources.
- Separation of delta mechanism for different data targets.
- Enhanced filtering in dataflow.
- Improved transparency of staging processes across data warehouse layers.
- Improved performance : optimized parallelization
- Enhanced error handling in the form of error stack
- Enables real-time data acquisition.

Create Data Transfer Process





RSA1

Modelling →

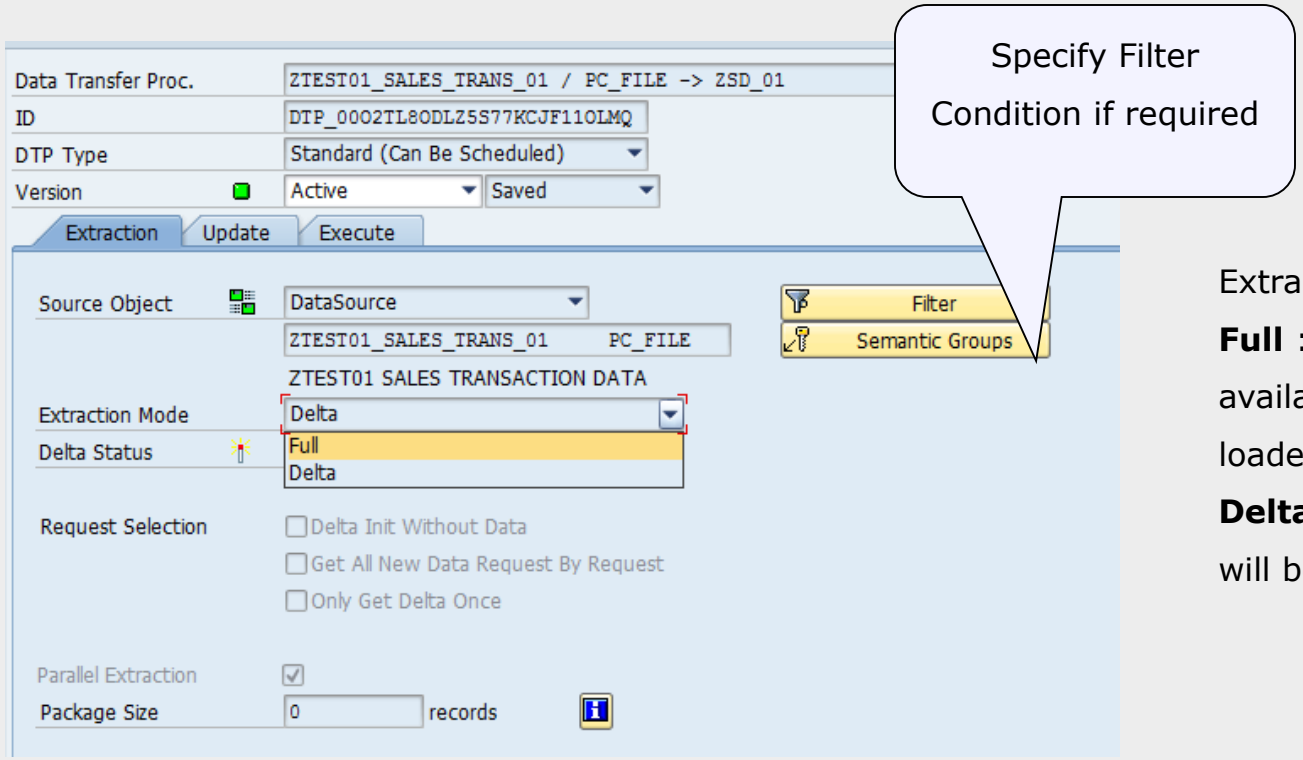
InfoProvider→

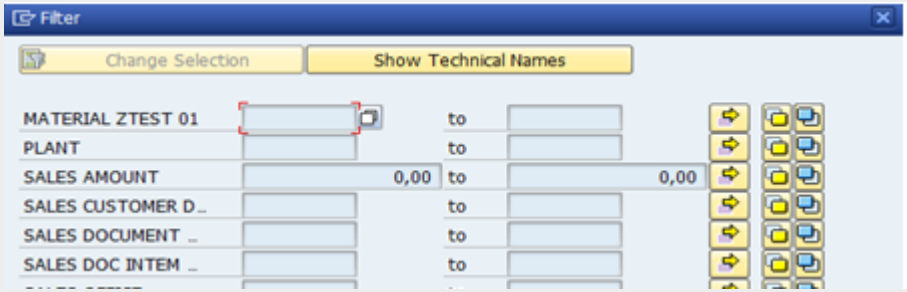
Context Menu of Data Transfer Process

→

Create Data Transfer Process

Maintain Data Transfer Process



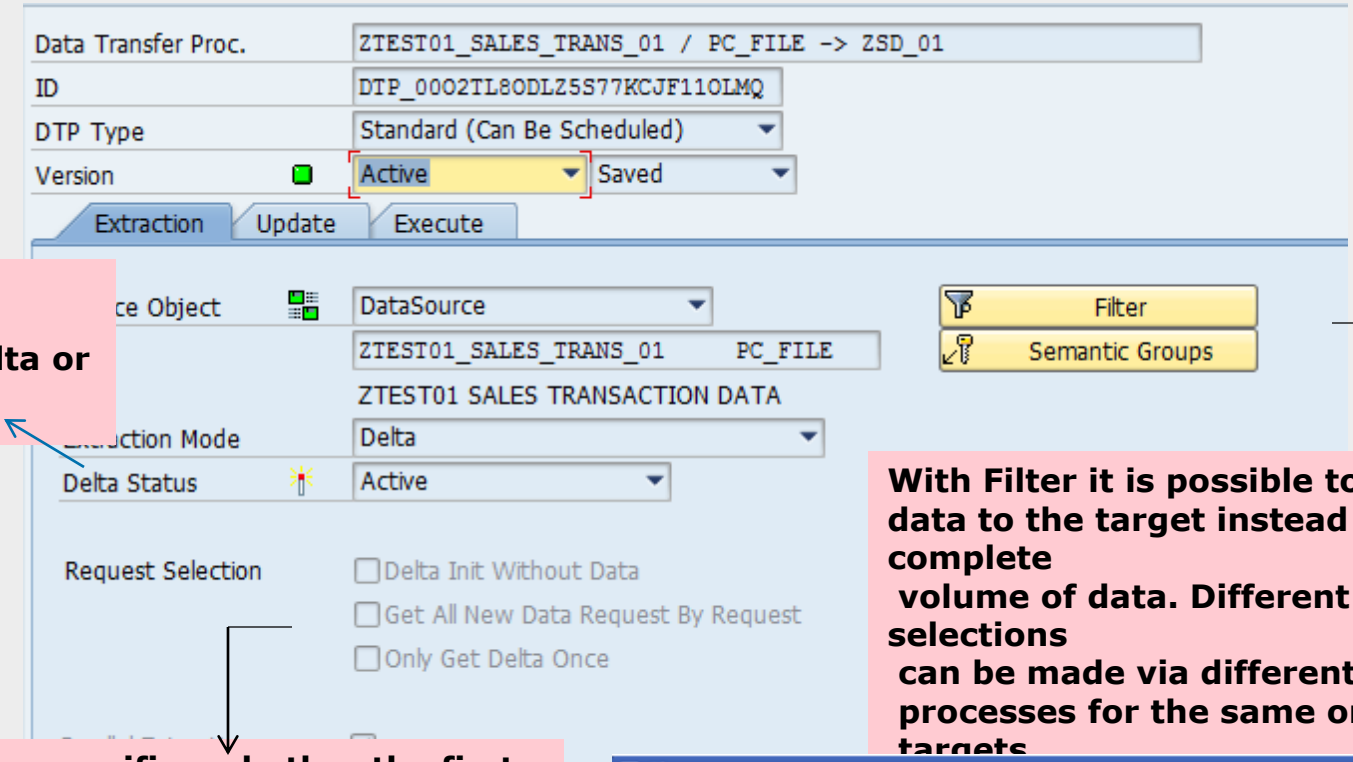


Extraction Mode :

Full : All the requests available in the source will be loaded

Delta :Only unloaded requests will be loaded

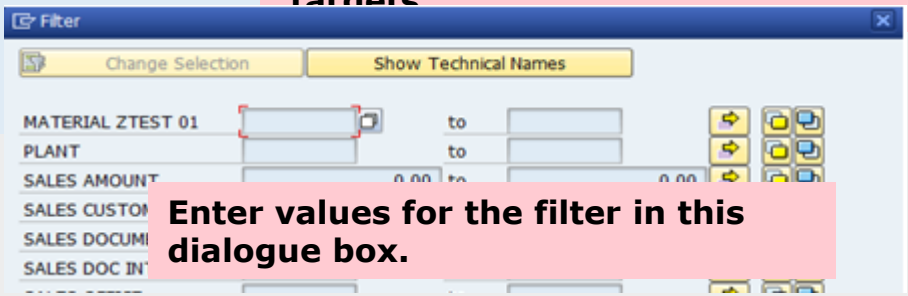
DTP: Filtering Data



Extraction Mode : Delta or Full

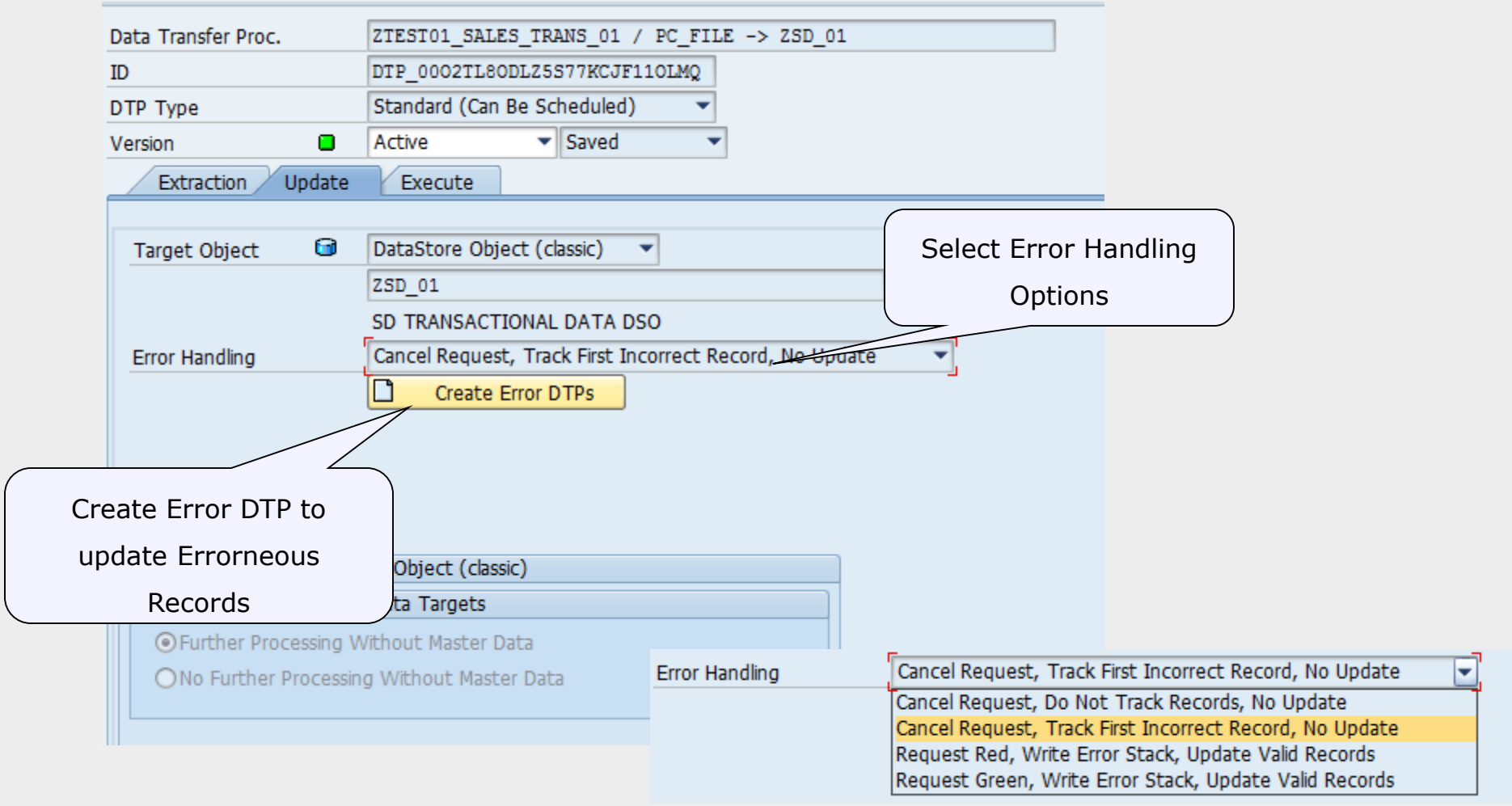
With Filter it is possible to load set of data to the target instead of the complete volume of data. Different data selections can be made via different data transfer processes for the same or for different targets

This option specifies whether the first request of delta DTP should be transferred without any data. This is same as "Initialization without data transfer" in Delta Info Package.

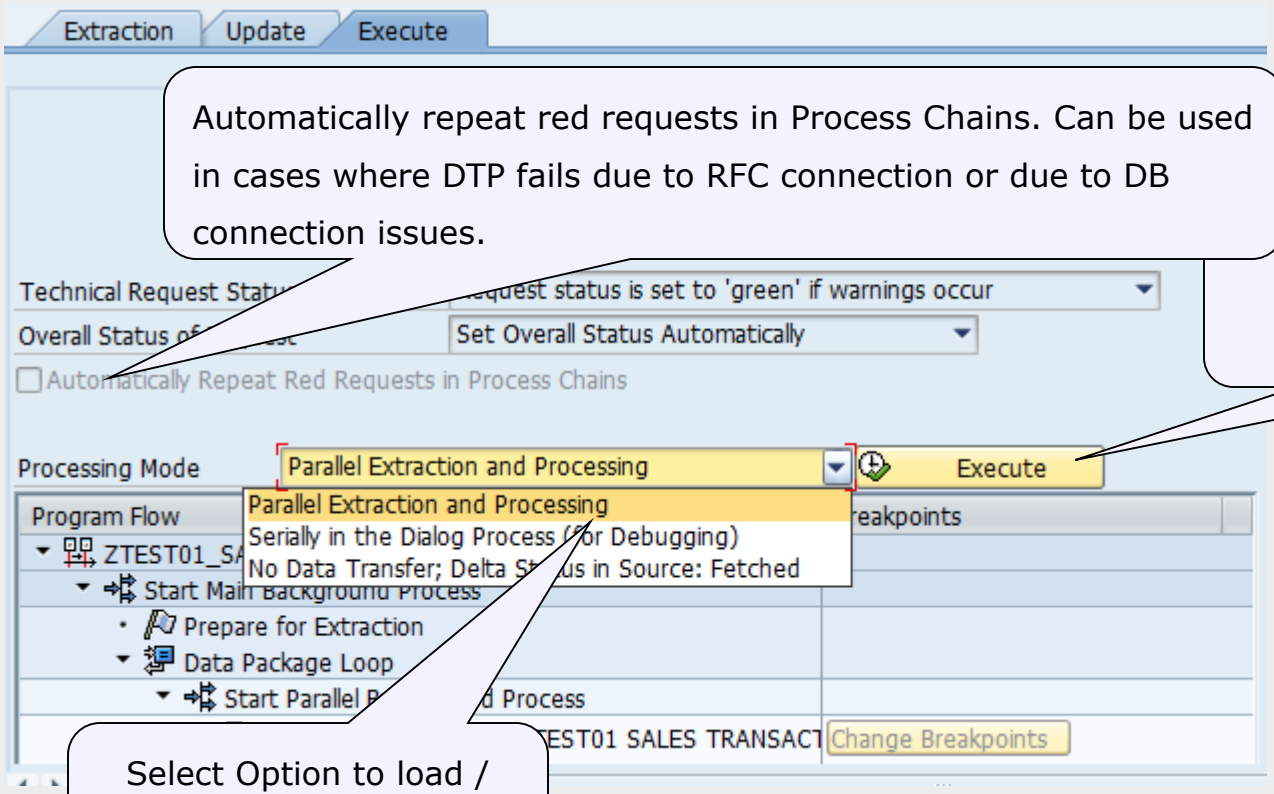


Enter values for the filter in this dialogue box.

Maintain Data Transfer Process



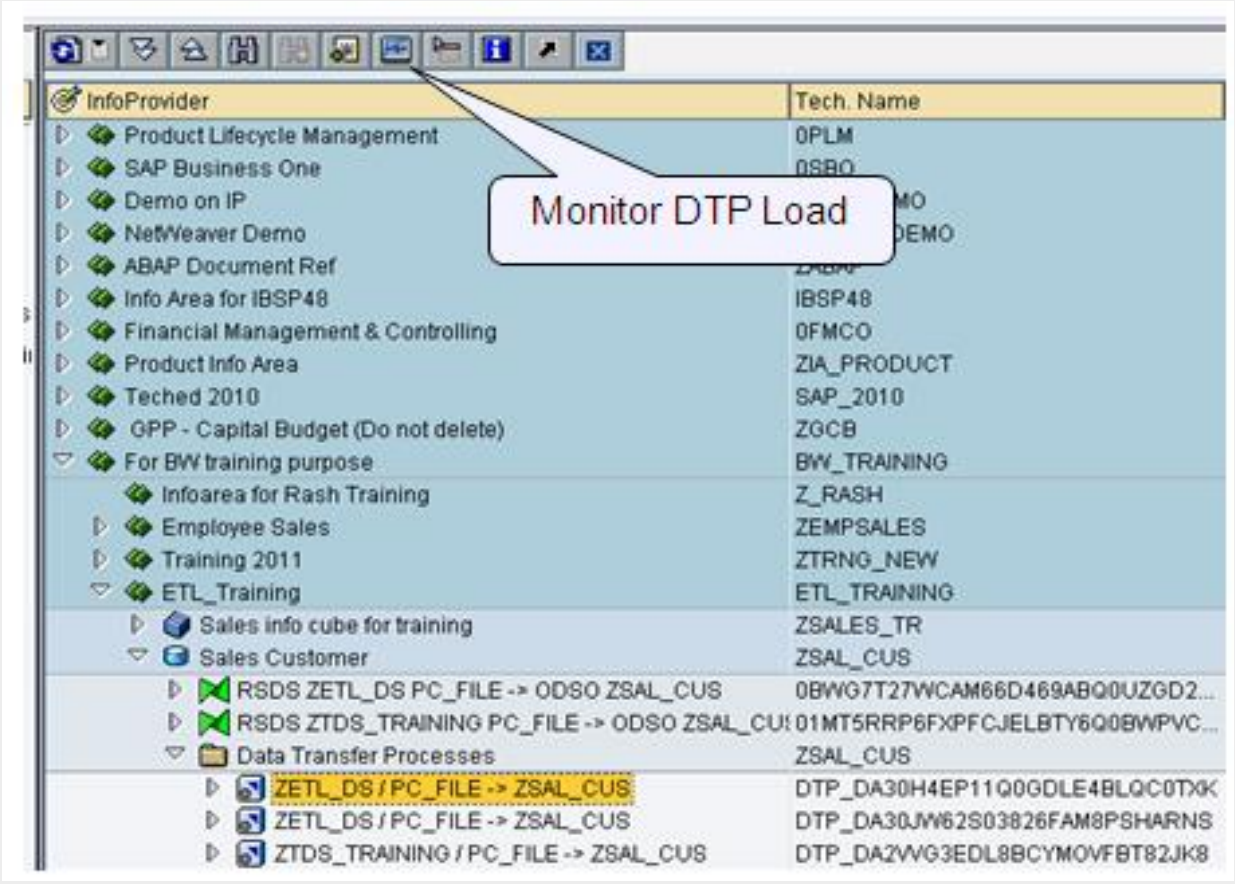
Maintain Data Transfer Process



Execute: To start data load from source to target

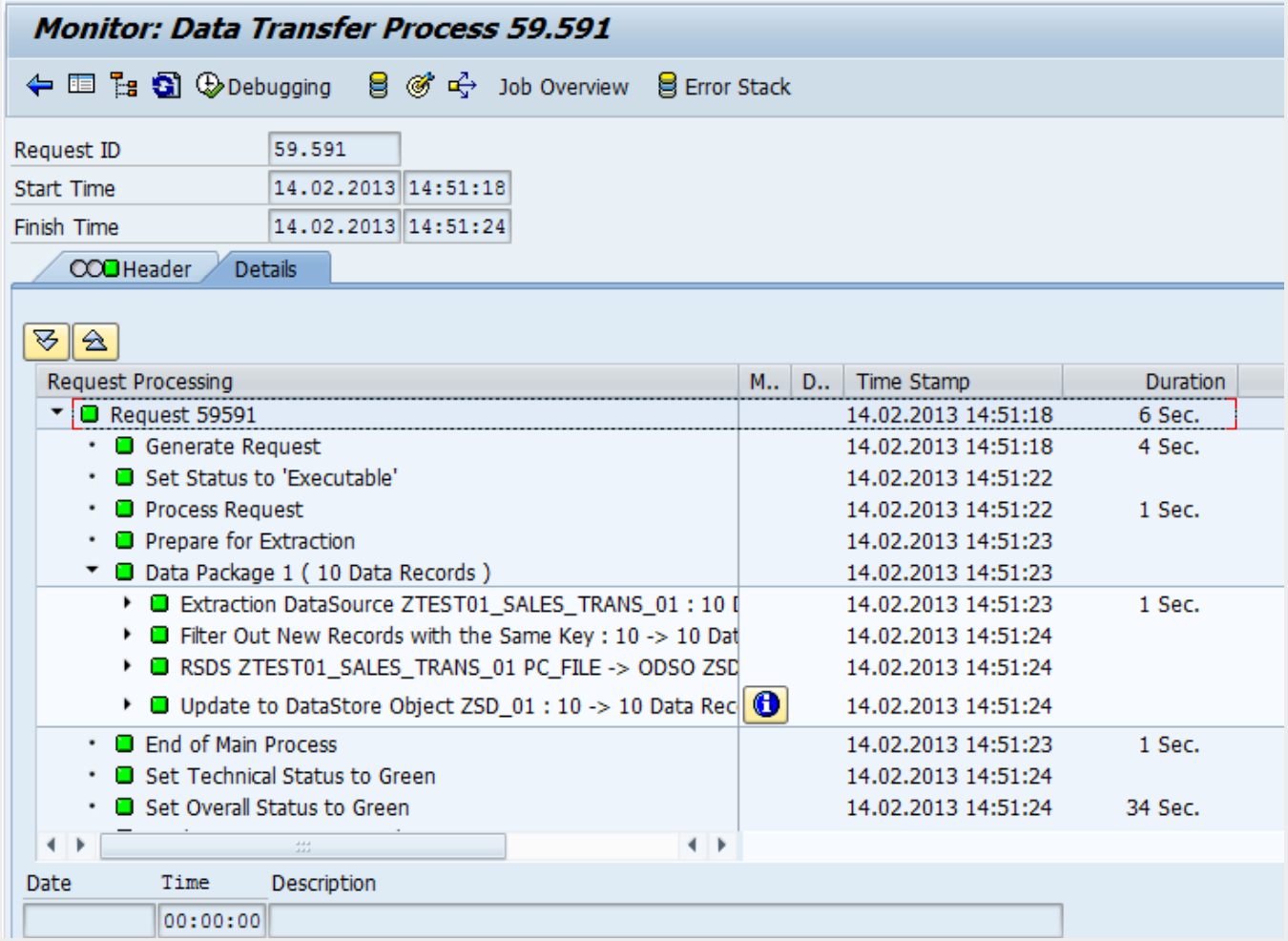
To Debug routines : Select "Serially in the Dialog Process (for Debugging)"

Monitor DTP Load



Select DTP and press Monitor

Monitor DTP Load



Green : Load Successful

Yellow: Dataload Running

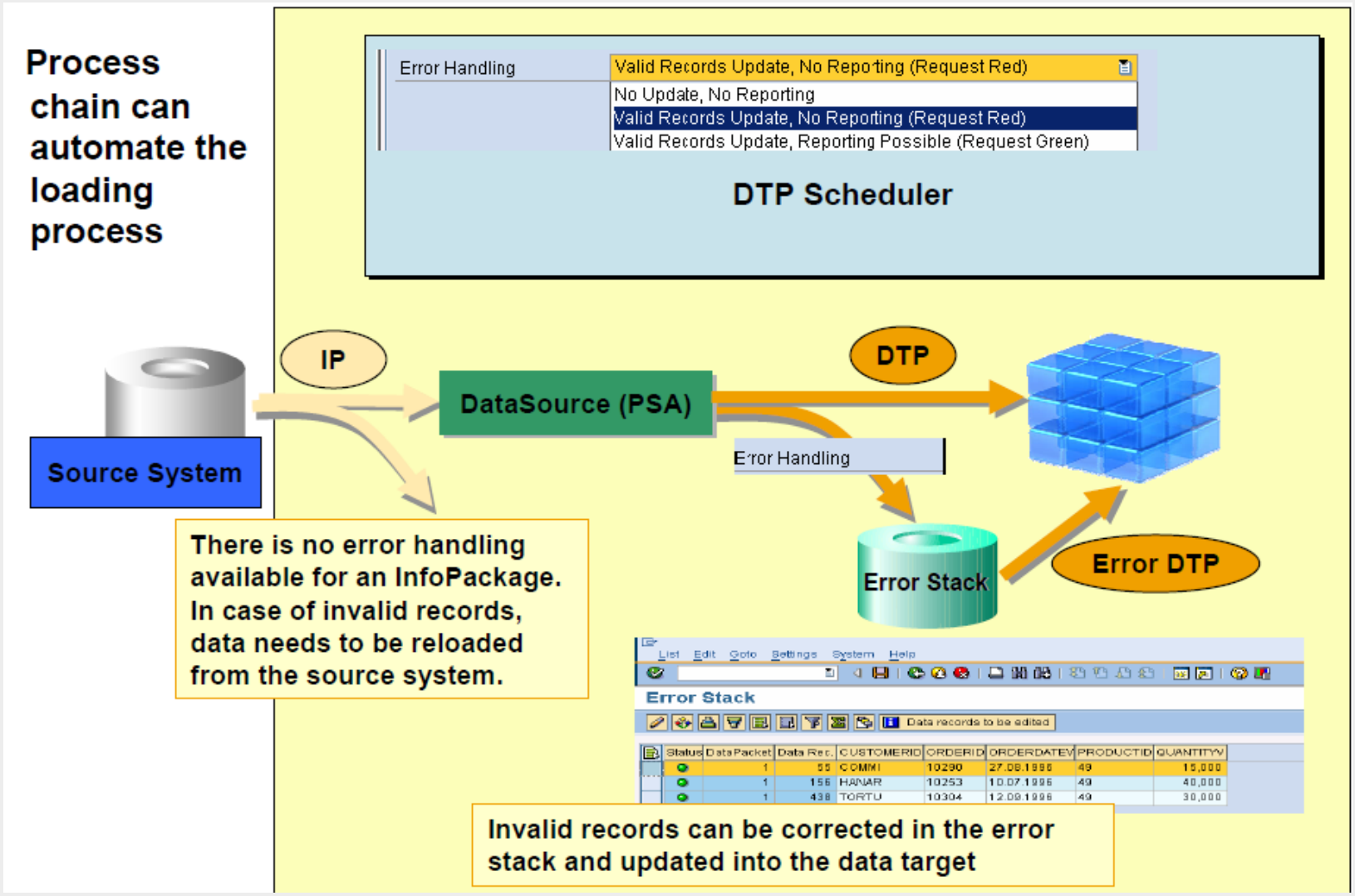
Red: Data Load Failed

Handling Data Records With Errors

➢Using the error handling settings on the Update tab page in the data transfer process, when data is transferred from a DTP source to a DTP target, you can specify how the system is to react if errors occur in the data records.

➢These settings were previously made in the Info Package. When using data transfer processes, Info Packages write to the PSA only. Error handling settings are therefore no longer made in the Info Package, but in the data transfer process

Error Handling Overview



Error Handling Features

- Possibility to choose in the scheduler to
  - Abort process when errors occur
  - Process the correct records but do not allow reporting on them
  - Process the correct records and allow reporting on them
- Number of wrong records which lead to a wrong request
- Invalid records can be written into an error stack
- Keys should be defined for error stack to enable the error handling of data store object
- Temporary data storage can be switched on/off for each sub step of the loading process
- Invalid records can be updated into data records after their correction

Error Stack

➢Stores erroneous records

- Keeps the right sequence of records → for consistent data store handling.
- Key of error stack defines which data should be detained from the update after the erroneous data record.
- After Correction, Error-DTP updates data from error stack to data target.
- Note: Once the request in the source object is deleted, the related data records in error stack area automatically deleted.

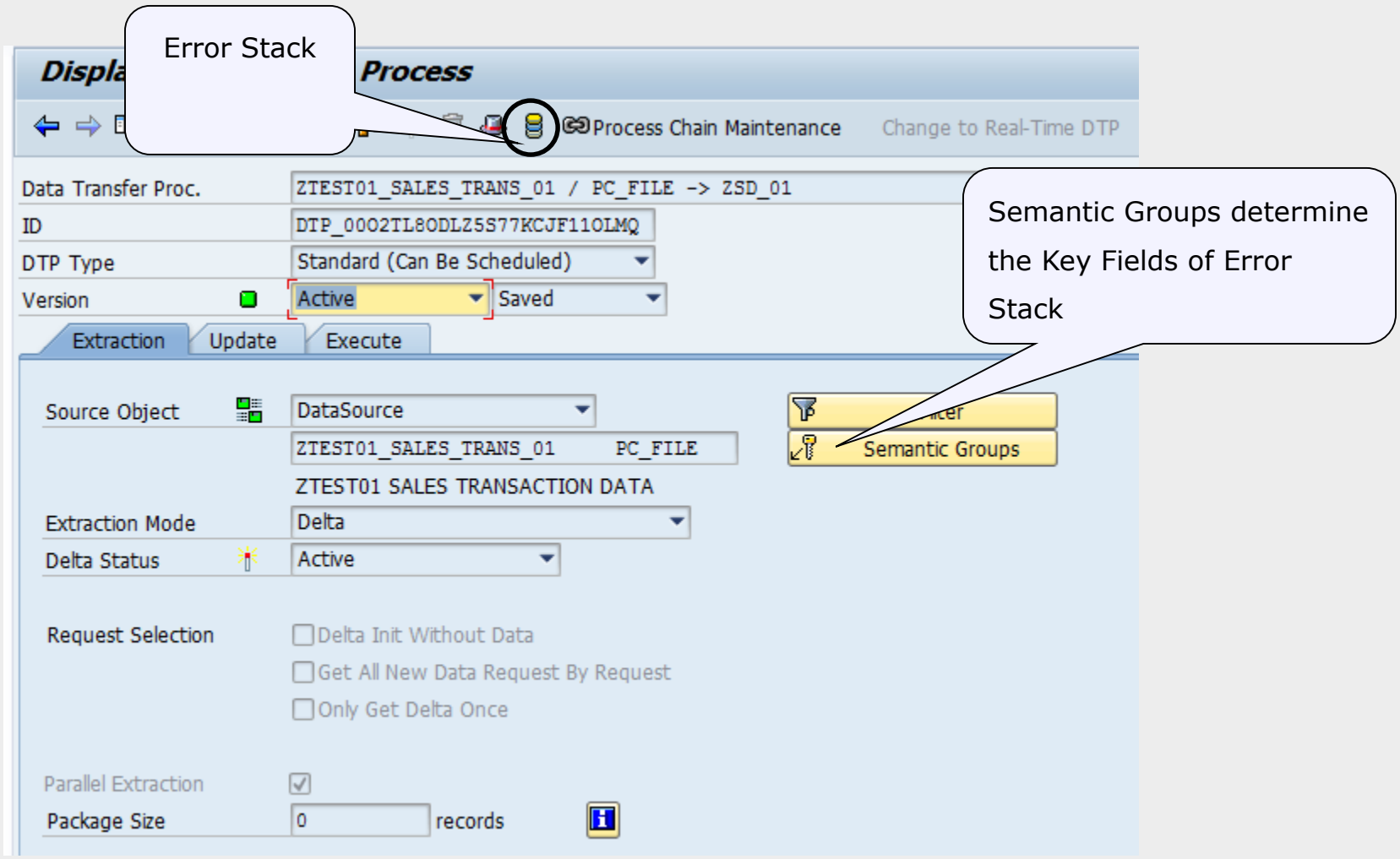
Error Stack

➢Key of Error Stack = Semantic Groups.

➢Subset of the key of the target object.

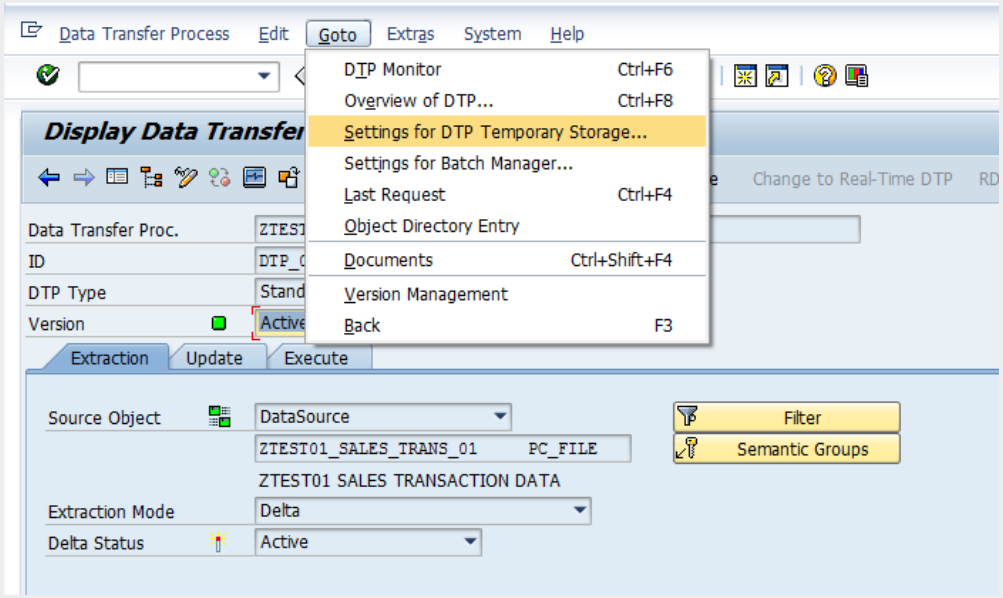
- Max. 16 fields
- Defines which data should be detained from
- the update of erroneous data record (for data store object)
- The bigger the key, the fewer records will be written to the error stack.

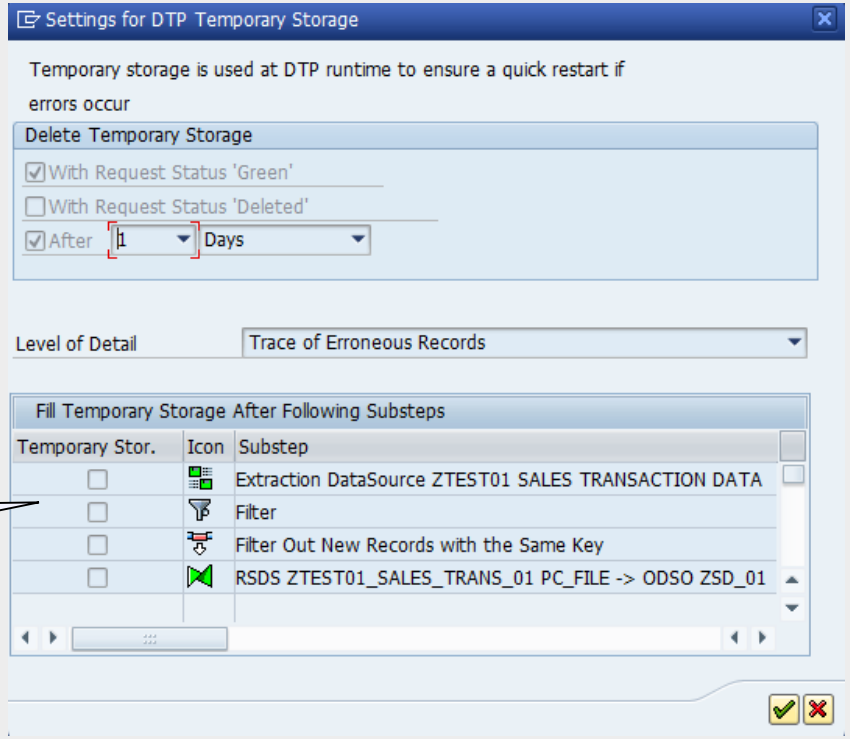
Error Stack


Error Stack
Process
Process Chain Maintenance
Change to Real-Time DTP
Data Transfer Proc.
ZTEST01_SALES_TRANS_01 / PC_FILE -> ZSD_01
ID
DTP_0002TL8ODLZ5S77KCJF110LMQ
DTP Type
Standard (Can Be Scheduled)
Version
Active
Saved
Extraction
Update
Execute
Source Object
DataSource
ZTEST01_SALES_TRANS_01 PC_FILE
ZTEST01 SALES TRANSACTION DATA
Semantic Groups
Extraction Mode
Delta
Delta Status
Active
Request Selection
Delta Init Without Data
Get All New Data Request By Request
Only Get Delta Once
Parallel Extraction
Package Size
0
records
Semantic Groups determine the Key Fields of Error Stack

Temporary Data Storage

- In order to analyze the data at various stages you can activate the temporary storage in the DTP
- This allows you to determine the reasons of error





Switch on/off the temporary data storage for data loading steps