

Applies to:

SAP BI 7.0 developers and support Users. For more information, visit the [EDW homepage](#).

Summary

This paper describes the rule type of the SAP BI transformation which determines whether and how a characteristic or key figure, or a data field or key field is updated into the target.

The various rule types discussed in this paper are as follows:-

- Direct Assignment
- Constant
- Formula
- Read Master Data
- Routine

Author: Dinesh V

Company: IGate Global Solutions Limited

Created on: 16 May 2011

Author Bio



Dinesh is working as a SAP BW/BI Consultant in IGate Global Solutions Limited, India. His expertise include SAP BW 3.5, SAP BI 7.0 and ABAP developments.

Table of Contents

| | |
|------------------------------------------------------------|----|
| Introduction | 3 |
| Rule Type - Direct Assignment: | 4 |
| Step 1: Assign the Rule Type – Rule Details | 4 |
| Step 2: Assign the Rule Type – Mapping..... | 4 |
| Rule Type - Constant..... | 5 |
| Step 1: Assign the Rule Type – Rule Details | 5 |
| Step 2: Value in the Transformation | 5 |
| Rule Type - Formula | 6 |
| Step 1: Map the info objects | 7 |
| Step 2: Assign the Rule Type – Rule Details | 7 |
| Step 3: Formula Builder | 7 |
| Rule Type - Read Master Data | 8 |
| Step 1: Map the source to the target info object | 8 |
| Step 2: Assign the Rule Type – Rule Details | 9 |
| Step 3: Assign the Rule Type – Rule Details | 9 |
| Rule Type - Routine..... | 10 |
| Step 1: Map the transformations and define Rule Type | 10 |
| Step 2: Write the logic using SAP ABAP Code..... | 11 |
| Related Content..... | 12 |
| Disclaimer and Liability Notice..... | 13 |

Introduction

The transformation process allows you to consolidate, cleanse, and integrate data. You can semantically synchronize data from heterogeneous sources.

When you load data from one BI object into a further BI object, the data is passed through a transformation. A transformation converts the fields of the source into the format of the target.

You create a transformation between a source and a target. The BI objects DataSource, InfoSource, DataStore object, InfoCube, InfoObject and InfoSet serve as source objects. The BI objects InfoSource, InfoObject, DataStore object and InfoCube serve as target objects.

A transformation consists of at least one transformation rule. Various rule types, transformation types, and routine types are available. These allow you to create very simple to highly complex transformations:

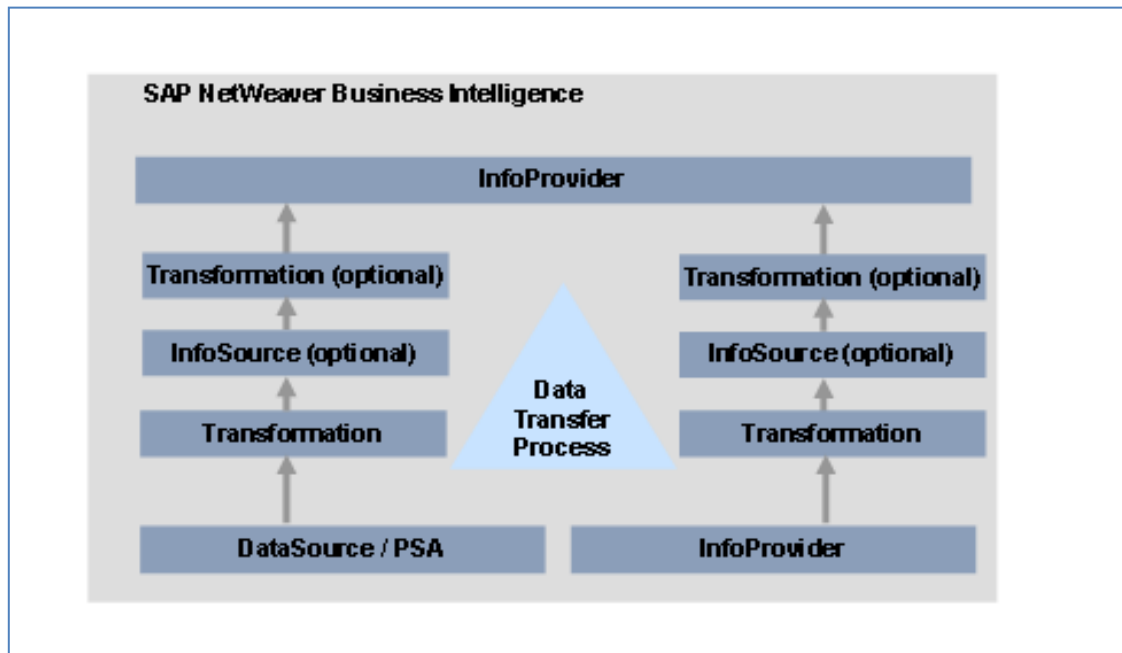



Figure 1:- SAP BI Data Flow

Rule Type - Direct Assignment:

The target field info object is filled directly from the selected source Info Object. When you create a new transformation between the source and target objects in SAP BI 7.0, the system proposes the default direct assignment between similar objects.

If the system does not propose a source Info Object, you can assign a source Info Object of the same type (amount, number, integer, quantity, float, time) through direct assignment.

Step 1: Assign the Rule Type – Rule Details

Double Click on the symbol  of the target field in the transformation to navigate to Rule Details window and assign the Rule Type as Direct Assignment as shown in below screenshot.

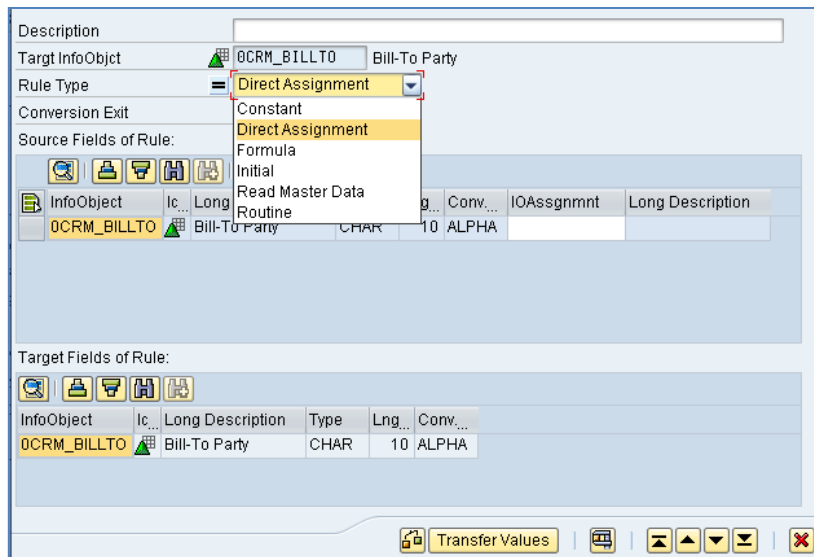


Figure 2:- Assign the Direct Assignment Rule Type

Step 2: Assign the Rule Type – Mapping

Click on the source info object and drag to the required target info object in the transformation to map the info objects via default direct assignment.

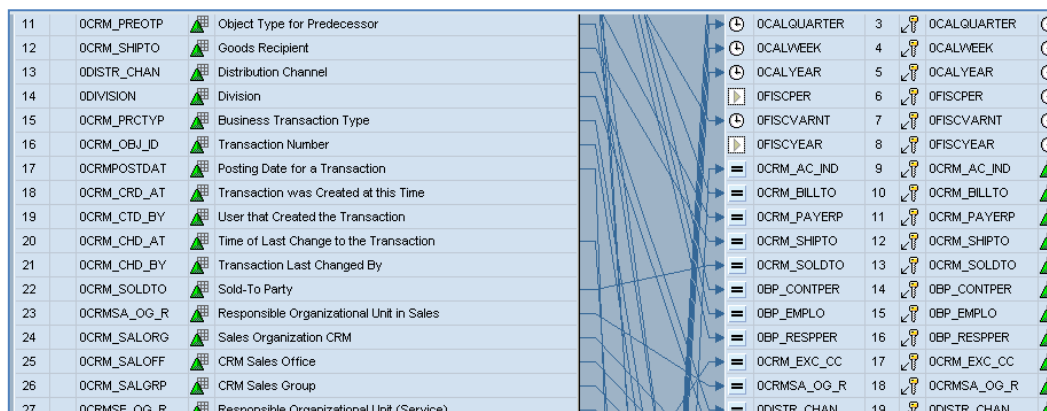



Figure 3:- Transformation – Direct Assignment

Rule Type - Constant

The field is not filled by the Info Object; it is filled directly with the value specified while defining the constant rule type.

For example (as per the screenshot): - the 0FISCYEAR target info object is been populated with constant value '2011' in the transformation.

Step 1: Assign the Rule Type – Rule Details

Double Click on the symbol  of the target field in the transformation to navigate to Rule Details window. Select the Rule Type as Constant and update the Constant Value as per the business requirement.

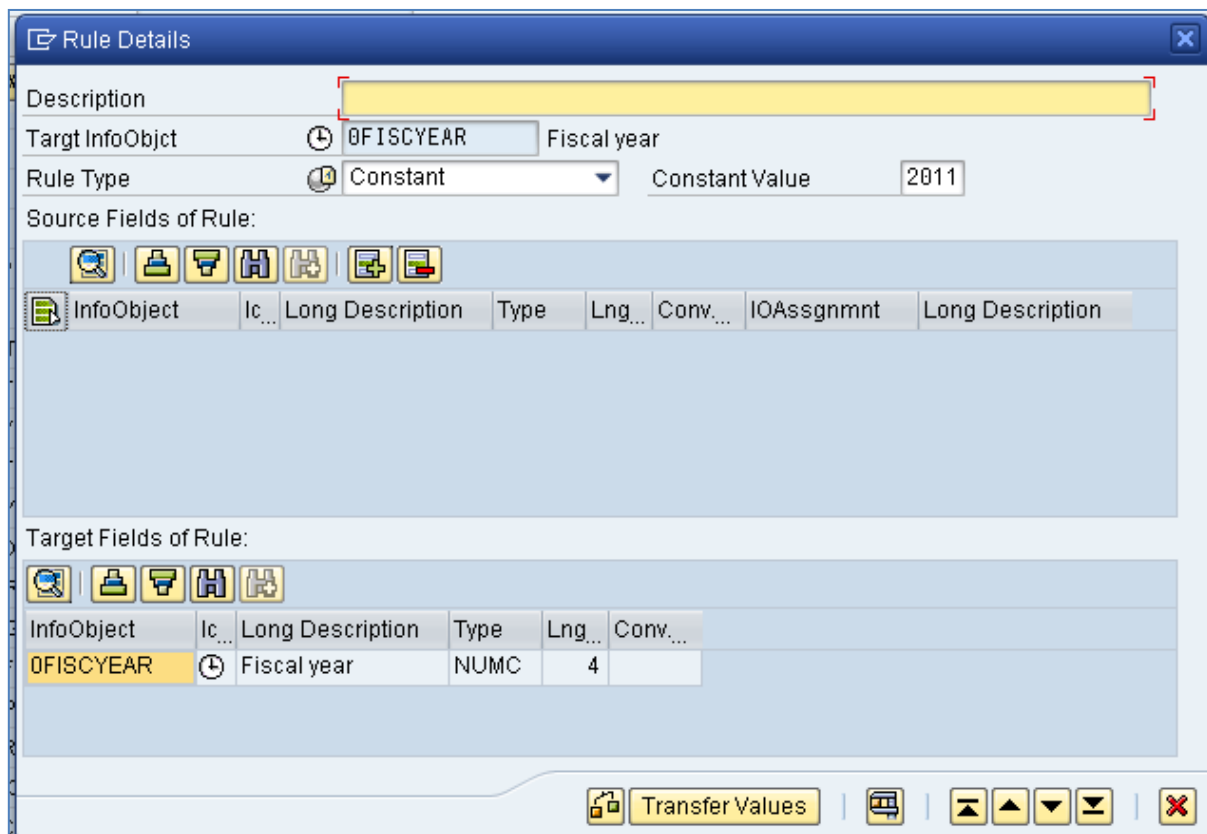


Figure 4:- Assign the Constant Rule Type

Step 2: Value in the Transformation

Once the constant is assigned in the Rule Details, the value will appear in the transformation as shown in below screenshot.

| Transaction Type | Transaction Number | Transaction Description | InfoObject | Value | InfoObject | Description |
|------------------|--------------------|-------------------------|-------------|-------|-------------|----------------------|
| | | | 0FISCPER | 6 | 0FISCPER | Fiscal year / period |
| | | | 0FISCVARNT | 7 | 0FISCVARNT | Fiscal year variant |
| | | | 2011 | 8 | 0FISCYEAR | Fiscal year |
| | | | 0CRM_AC_IND | 9 | 0CRM_AC_IND | Accounting Indicator |

Figure 5:- Transformation – Constant

Rule Type - Formula

The target Info Object is updated with a value determined using a formula which might need mapping of the source info objects to the target info object as parameters for the formula.

A transformation library having the standard functions is available for transformation rules to be utilized in our formulae as per the business need.

The transformation library, in collaboration with the formula builder, enables you to easily create formulas, without using ABAP coding.


The transformation library has over 70 pre-defined functions, in the following categories with some commonly used functions:-

- Basic Functions
 - & (Concatenate)
 - * (Multiplication)
 - >= (Greater than or equal to)
 - ^ (Raise to a power)
- Date Functions
 - DATE_DIFF (Date Difference)
 - DATE_MONTH (Calculate Month from Date)
 - DATE_TO_WEEK (DATE-> WEEK)
 - LAST_WORKINGDATE_MONTH (Calculate Last Work Day for Month)
- Functions for character strings
 - LEFT (First N Chars)
 - SUBSTRING (Part of Character Strings)
 - STR_LEN (Character String Length)
- Mathematical functions
 - COS (Cosine)
 - DIV (Quotient)
 - MOD (Remaining)
- Suitable functions
 - EXP (Exponential Function)
- User Defined Functions
 - Customer defined functions using BAdI (Refer the link in the Related Content)
- Other functions
 - ABORT_PACKAGE (Cancel Package)
 - SKIP_RECORD (Skip Record)

Step 1: Map the info objects

If any parameters are required in the formula, map the required source info objects to the target info object via direct assignment

Step 2: Assign the Rule Type – Rule Details

Double Click on the  target field in the transformation to navigate to Rule Details window. Select the Rule Type as Formula to navigate to the Formula Builder.

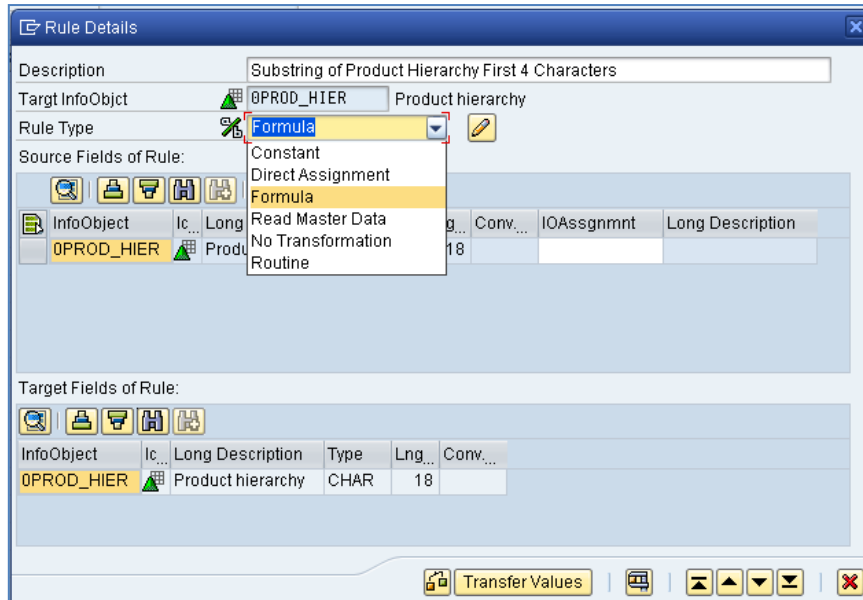


Figure 6:- Rule Type – Formula

Step 3: Formula Builder

Build the formula based on the business requirement using the functions, parameters and the values from the formula builder.

For example: - I have created a formula to select first 4 characters from the product hierarchy and populate the target info object.

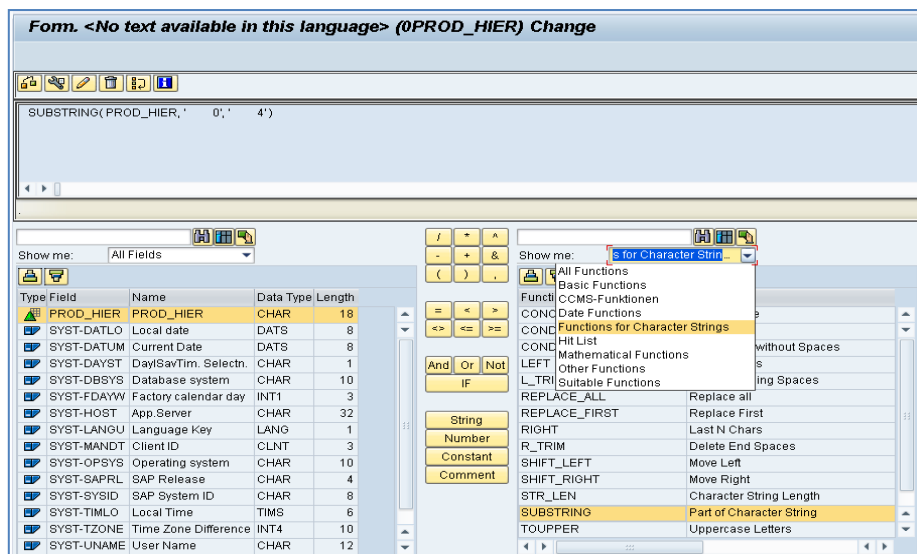


Figure 7:- Formula Builder

Rule Type - Read Master Data

The Info Object is updated by reading the master data table of a characteristic that is included in the source with a key and a value and that contains the corresponding Info Object as an attribute. The attributes and their values are read using the key and are then returned.

It is not possible to read recursively, that is, to read additional attributes for the attribute. To do this, you have to use routines.

If you have changed master data, you have to execute the change run. By reading the master data, the active version is read. If this is not available, an error occurs.

For Example:

The *Company Code (0COMP_CODE)* characteristic is included in the target but does not exist in the source as a characteristic.

However, the source contains a characteristic *Cost Center (0COSTCENTER)* that has the *Company Code (0COMP_CODE)* characteristic as an attribute.

You can read the *Company Code* attribute from the master data table and use it to fill the *Company Code* characteristic in the target.

Step 1: Map the source to the target info object

Connect the target info object (Company Code as per the above example) to the source info object (Cost Center) which has the desired attribute using direct assignment.

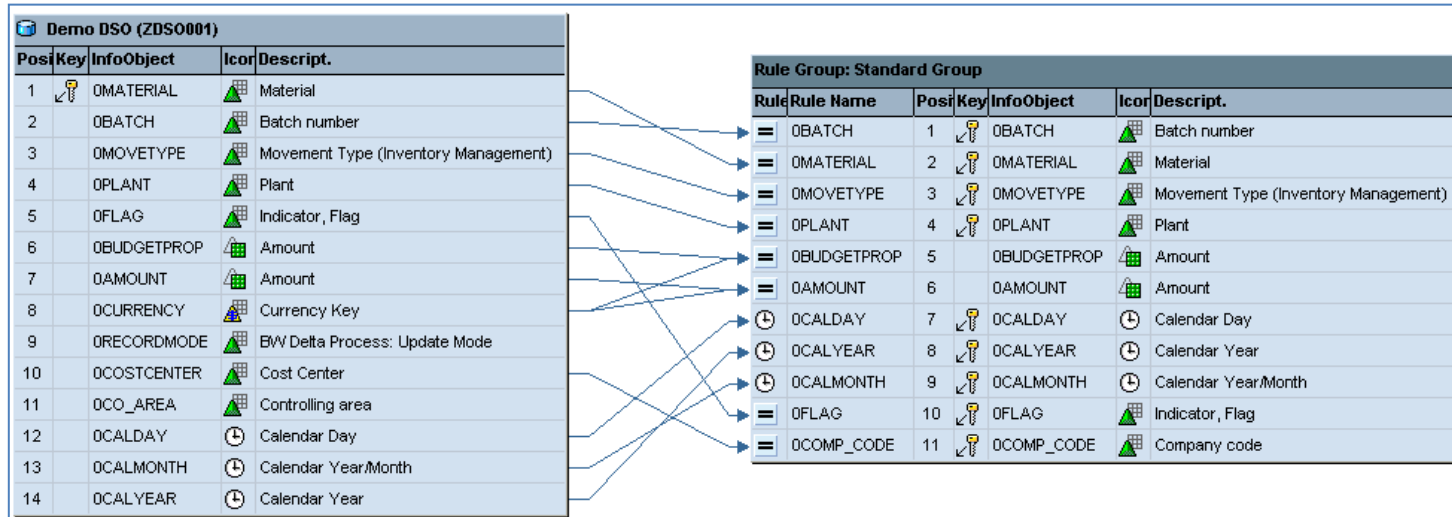



Figure 8:- Map the info objects

Step 2: Assign the Rule Type – Rule Details

Double Click on the  target field in the transformation to navigate to Rule Details window. Select the Rule Type as “Read Master Data” and assign the source info object which has the target info object as the attribute.

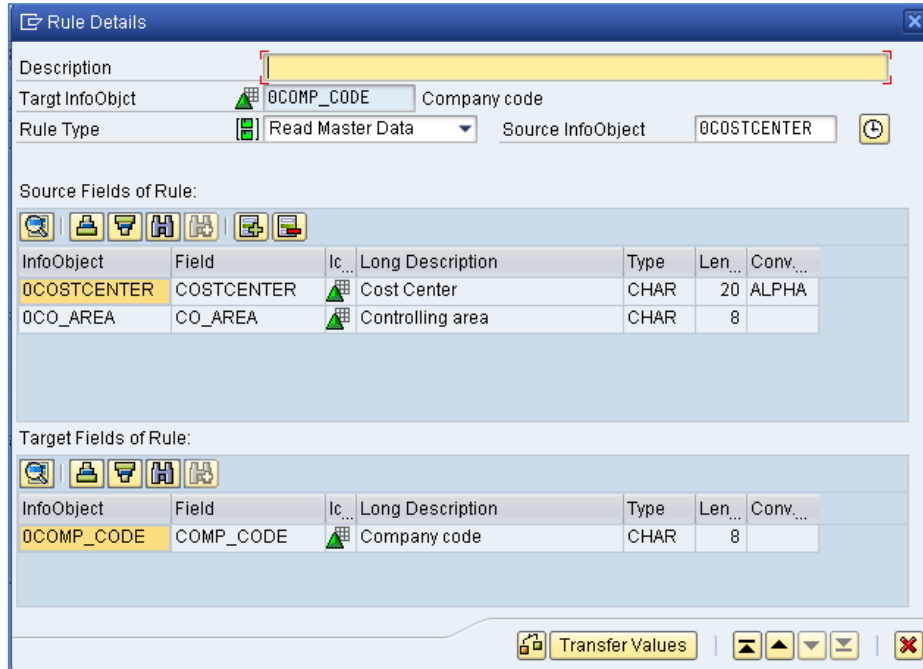


Figure 8:- Rule Type – Read Master Data

Step 3: Assign the Rule Type – Rule Details

Once the source info object is assigned, check and click on Transfer Values to complete the transformation as shown in the figure below.

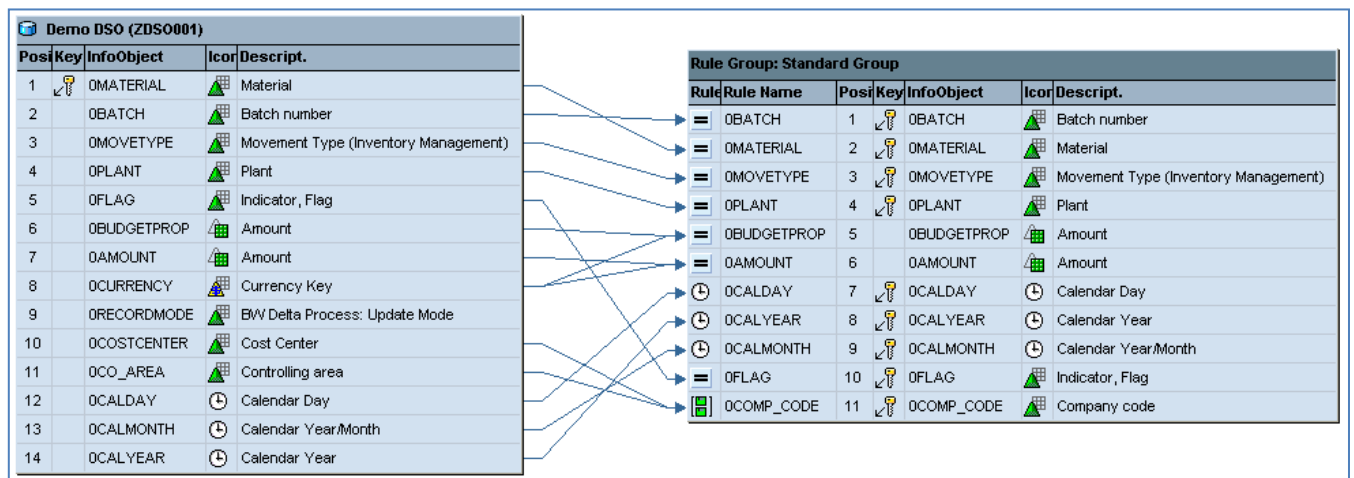


Figure 9:- Read Master Data in the transformation

Rule Type - Routine

The target field is filled by the start or field routine using the source fields as the parameters in the ABAP Code. We have to use the Routine only when there is no scope of using direct assignment, formula or read master data.

For Data Store objects and Info Objects: you cannot use the return code in the routine for data fields that are updated by being overwritten. If you do not want to update specific records, you can delete these from the start routine.

If, for the same characteristic, you generate different rules for different key figures or data fields, a separate data record can be created for each key figure from a data record of the source.

With Info Cubes: You can also select *Routine with Unit*. The return parameter 'UNIT' is then also added to the routine. You can store the required unit of the key figure, such as 'ST', in this parameter. You can use this option, for example, to convert the unit KG in the source, into tons in the target.

For Example:

Based on the value of the controlling area (OCO_AREA) in the source, the indicator flag (OFLAG) data in the target is populated. The logic should be defined to populate the Flag only if the Controlling Area is “CH01” or “PT01”.

Step 1: Map the transformations and define Rule Type

Map the required source fields to the target object via direct assignment. Double click on the target field and assign the Rule Type as “Routine” which will navigate to the routine screen.

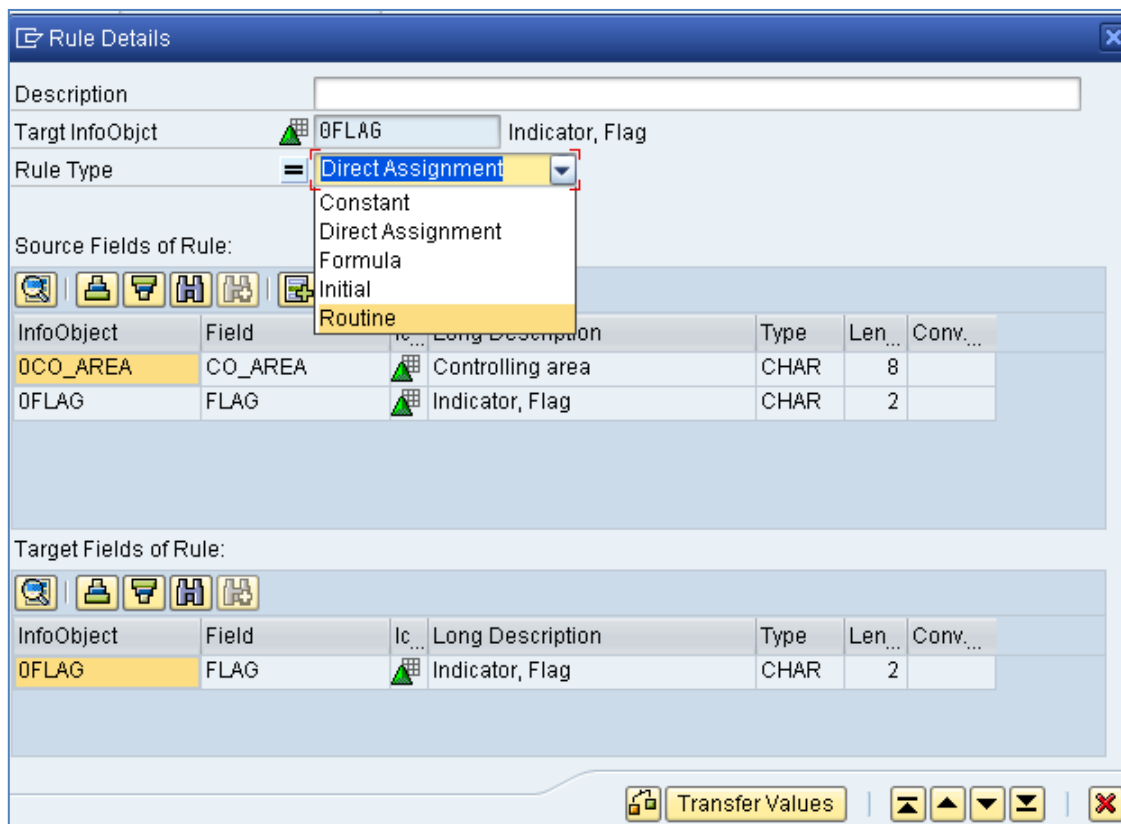


Figure 10:- Rule Type - Routine

Step 2: Write the logic using SAP ABAP Code

All the source fields in the routine Rule Type are declared in the standard types `_ty_s_SC_1` and the target fields in the types `_ty_s_TG_1`.

```

TYPES:
  BEGIN OF _ty_s_SC_1,
*   InfoObject: OFLAG Indicator, Flag.
    FLAG          TYPE /BIO/OIFLAG,
*   InfoObject: OCO_AREA Controlling area.
    CO_AREA       TYPE /BIO/OICO_AREA,
  END   OF _ty_s_SC_1.
TYPES:
  BEGIN OF _ty_s_TG_1,
*   InfoObject: OFLAG Indicator, Flag.
    FLAG          TYPE /BIO/OIFLAG,
  END   OF _ty_s_TG_1.

```

Figure 11:- Declaration of Source (`_ty_s_SC_1`) and Target Types (`_ty_s_TG_1`) in routine

The source fields can be accessed using the standard data declaration field "SOURCE_FIELDS". Hence the source fields can be used in the logic as `SOURCE_FIELDS-<field_name>`.

```

METHODS
  compute_OFLAG
  IMPORTING
    request          type rsrequest
    datapackid       type rsdatapid
    SOURCE_FIELDS    type _ty_s_SC_1
  EXPORTING
    RESULT           type _ty_s_TG_1-FLAG
    monitor          type rstr_ty_t_monitor
  RAISING
    cx_rsout_abort
    cx_rsout_skip_record.
METHODS
  invert_OFLAG
  IMPORTING
    i_r_sset_outbound      TYPE REF TO cl_rsmds_set
    i_th_fields_outbound   TYPE HASHED TABLE
    i_r_sset_outbound_complete TYPE REF TO cl_rsmds_set
    i_r_universe_inbound   TYPE REF TO cl_rsmds_universe
  CHANGING
    c_r_sset_inbound       TYPE REF TO cl_rsmds_set
    c_th_fields_inbound    TYPE HASHED TABLE
    c_exact                TYPE rs_bool.
ENDCLASS.           "routine DEFINITION

```

Figure 12:- Declaration of SOURCE_FIELDS in routine

Finally, the logic is written in standard SAP ABAP Code to populate the value of the flag to the target fields if the controlling area is either "CH01" or "PT01".

```

*   result value of the routine
IF SOURCE_FIELDS-CO_AREA = 'CH01' OR
  SOURCE_FIELDS-CO_AREA = 'PT01'.
  RESULT = SOURCE_FIELDS-FLAG.
ELSE.
  RESULT = '0'.
ENDIF.

```

Figure 13:- Routine Result Value

Related Content

[SAP BI Transformation](#)

[SAP BI Transformation - Rule Type](#)

[Transformation Library and Formula Builder](#)

[Example for Using the Formula Editor](#)

[BAI: Customer-Defined Functions in the Formula Builder](#)

[How To...Routines within Transformations](#)

Related Articles from the Author:

[Using Error Stack and Error DTPs in SAP BI 7.0](#)

[Publish a SAP BI Report in Portal](#)

[Generate Export Data Source](#)

For more information, visit the [EDW homepage](#)

Disclaimer and Liability Notice

This document may discuss sample coding or other information that does not include SAP official interfaces and therefore is not supported by SAP. Changes made based on this information are not supported and can be overwritten during an upgrade.

SAP will not be held liable for any damages caused by using or misusing the information, code or methods suggested in this document, and anyone using these methods does so at his/her own risk.

SAP offers no guarantees and assumes no responsibility or liability of any type with respect to the content of this technical article or code sample, including any liability resulting from incompatibility between the content within this document and the materials and services offered by SAP. You agree that you will not hold, or seek to hold, SAP responsible or liable with respect to the content of this document.