



PUBLIC

## How-To: Maintain Check and Derivation Rules in MDG for Material

Applicable Releases:  
All

Version 4.0  
October 2024

## Document History

Document Version	Description
1.00	First official release of this guide
1.20	Rule for Backend Check for Net weight
1.30	Additional notes, General Implementation
1.40	EhP6: BAdI USMD_RULE_SERVICE_CROSS_ET Chapter 6.1.2
1.50	New Chapter: Derivation with BAdI USMD_RULE_SERVICE_CROSS_ET
1.60	New function in MDG6.1 (see chapter 3.4 and Note 1701437)
1.70	Chapter 4.4, 5.2 and 5.3
1.80	Small corrections in chapter 5.1 and 5.2
1.90	New Chapter: Cross Entity Read Access
2.00	Small corrections
2.10	Small corrections 4.2.4; new chapters 4.4.4 and 7.7
3.0	Layout update and new chapter for MDQ Validation (November 2021)
4.0	Layout update (October 2024)

<b>1. BUSINESS SCENARIO .....</b>	<b>5</b>
<b>2. DEFINITION AND OVERVIEW .....</b>	<b>6</b>
2.1 DEFINITION.....	6
2.2 OVERVIEW CHECKS AND DERIVATIONS.....	6
<b>3. CHECKS .....</b>	<b>7</b>
3.1 MODEL DRIVEN CHECKS.....	7
3.1.1 <i>Code lists</i> .....	7
3.1.2 <i>Cardinality check</i> .....	7
3.1.3 <i>Format check</i> .....	7
3.1.4 <i>Required fields</i> .....	7
3.2 RE-USE OF EXISTING CHECK LOGIC IN ERP BACKEND SYSTEM OR S/4HANA BACKEND SYSTEM .....	7
3.2.1 <i>Customizing OMT4</i> .....	7
3.2.2 <i>Code list</i> .....	8
3.2.3 <i>Business checks (Material API)</i> .....	8
3.2.4 <i>Required fields as defined in material master customizing (T130F) and hardcoded (Material API)</i> .....	8
3.3 CUSTOM CHECKS .....	9
3.4 CUSTOMIZING CHANGE REQUEST TYPE FOR CHECK MESSAGES.....	9
3.4.1 <i>Customizing setting: Issue Error as Warnings</i> .....	10
3.4.2 <i>Customizing setting: Standard</i> .....	11
3.4.3 <i>Configure Properties of Change Request Step</i> .....	11
3.4.4 <i>Enforce check at any roundtrip</i> .....	11
<b>4. MDG, DATA QUALITY MANAGEMENT VALIDATION RULES.....</b>	<b>13</b>
4.1 EXAMPLE FOR VALIDATION RULE TO CHECK IF MATERIAL STATUS IS ALLOWED .....	13
4.2 EXAMPLE FOR VALIDATION RULE TO CHECK IF A DIVISION IS PROVIDED FOR FINISHED GOODS .....	19
4.3 EXAMPLE FOR VALIDATION RULE WITH EXTRA DATA PROVISIONING .....	19
<b>5. CHECKS WITH BRF+ .....</b>	<b>20</b>
5.1 EXAMPLE FOR CHECK_MATERIAL .....	20
5.1.1 <i>Simple Checks</i> .....	20
5.2 EXAMPLE FOR CHECK_MEAN_GTIN .....	21
5.2.1 <i>Simple Checks</i> .....	21
5.3 EXAMPLE FOR CHECK_UNITOFMSR .....	22
5.3.1 <i>Simple Checks</i> .....	22
5.3.2 <i>Rule to compare Net and Gross weight for Base Unit of Measure</i> .....	23
5.4 EXAMPLE FOR CHECK_MARBASIC.....	25
5.4.1 <i>Warning, if Serial Number Profile is changed</i> .....	25
<b>6. DERIVATIONS .....</b>	<b>28</b>
6.1 CUSTOM DERIVATIONS.....	28
6.1.1 <i>Derivation using BRF+/BAPI USMD_RULE_SERVICE</i> .....	28
6.1.2 <i>EhP6: Derivation using BAPI USMD_RULE_SERVICE_CROSS_ET</i> .....	28
6.1.3 <i>Comparison of Derivations</i> .....	28
6.2 CHANGE REQUEST ENRICHMENT BADI (MDG_BS_MAT_API_ENRICH_BADI).....	29
6.3 SMT-MAPPING STAGING → ACTIVE AREA.....	29
<b>7. DERIVATIONS WITH BRF+.....</b>	<b>30</b>
7.1 CROSS-ENTITY RULES .....	30
7.2 DEFAULTING OF VALUES .....	31
7.3 EXAMPLE FOR DERIVE_MATERIAL .....	32
7.3.1 <i>Execution time</i> .....	32
7.3.2 <i>Simple Rules</i> .....	32
7.3.3 <i>Example: Rules with Expression</i> .....	33
7.3.4 <i>Example: Decision Table with multiple results</i> .....	35
7.3.4.1 <i>Create a Structure Data Object</i> .....	35

---

7.3.4.2	Create a Decision Table .....	37
7.3.4.3	Create a Rule.....	40
7.3.4.4	Create a Ruleset.....	42
7.4	EXAMPLE FOR DERIVE_MARCMRPFC.....	43
7.4.1	<i>Default the Period Indicator to 'M'</i> .....	43
7.5	EXAMPLE FOR DERIVE_MARCMRPSP.....	44
7.5.1	<i>Default the Procurement Type</i> .....	44
7.6	EXAMPLE: USING VARIABLES IN DERIVATION RULES (LIKE WORKFLOW STEP) .....	47
7.6.1	<i>Set the values for the variables</i> .....	47
7.6.1.1	EhP5: Set the values (EhP5 UI BADI) .....	47
7.6.1.2	EhP6: Set the values for the variables.....	49
7.6.1.3	MDG6.1: New variables .....	49
7.6.2	<i>Create BRF+ Ruleset</i> .....	49
7.6.3	<i>Create BRF+ Function</i> .....	52
7.6.4	<i>EhP5 screenshots of the process</i> .....	52
7.7	EXAMPLE: BRF VALIDATIONS/DERIVATIONS FOR STAGING ONLY .....	54
7.7.1	<i>Background</i> .....	54
7.7.2	<i>Approach</i> .....	54
7.7.3	<i>Step by Step</i> .....	54
8.	CROSS-ENTITY READ ACCESS FOR DERIVATION BRF+ .....	58
8.1	EXAMPLE: CHANGE THE MRP TYPE DEPENDING ON THE PLANT-SPECIFIC MATERIAL STATUS .....	58
8.1.1	<i>Prerequisite: Create Data Object</i> .....	58
8.1.2	<i>Prerequisite: Create Post-Exit</i> .....	60
8.1.3	<i>Prerequisite: Create Z-Class</i> .....	61
8.1.4	<i>Create Derivation for MARCMRPPP</i> .....	66
8.1.5	<i>Add method to the Post-Exit</i> .....	93
8.1.6	<i>Test the BRF+ Rule</i> .....	94
9.	CROSS-ENTITY DERIVATION WITH BADI USMD_RULE_SERVICE_CROSS_ET .....	96
10.	TECHNICAL ENVIRONMENT BRF+ LANDSCAPE AND RULE MAINTENANCE .....	98
10.1	USE CASE .....	98
10.2	EXAMPLE FOR APPLICATION EXIT GET_CHANGEABILITY .....	98
11.	ADDITIONAL INFORMATION .....	100
11.1	FURTHER READING.....	100
11.1.1	<i>Information on SAP MDG on SAP S/4HANA</i> .....	100
11.1.2	<i>SAP Roadmap Explorer</i> .....	100
11.1.3	<i>Related Information</i> .....	100
11.2	SAP NOTES .....	100

---

## **1. BUSINESS SCENARIO**

SAP Master Data Governance (MDG) provides business processes to find, create, change, and mark master data for deletion. It supports the governance of master data in a central hub and the distribution to connected operational and business intelligence systems.

The processes are workflow-driven and can include several approval and revision phases, and the collaboration of all users participating in the master data maintenance.

MDG offers change request (CR)-based processing of master data with integrated workflow, staging, approval, activation, and distribution.

To support these processes, check and derivation rules can be created for the MM data model. Checks ensure that the master data is consistent. You can use derivations to calculate values for attributes from other resolved attributes, thereby simplifying data entry.

This How-To guide provides an overview of the checks (validation) and derivations used in MDG-M, as well as examples.

## 2. DEFINITION AND OVERVIEW

### 2.1 Definition

- Checks/Validation raise messages. Message with severity *info*, *warning*, *error*, and *abort* are possible.
- Derivations calculate values for attributes and can also raise messages. Only messages with severity *info* are possible.

### 2.2 Overview Checks and Derivations

- Model driven checks
  - Code list check
  - Cardinality check
  - Format check
  - Required fields
- Re-use of existing check logic in ERP backend system or S/4HANA backend system
  - Check messages configured in OMT4
  - Code lists
  - Business checks (Material API)
  - Required fields as defined in material master customizing (T130F) and hardcoded (Material API)
- Custom checks and validation (modeled and programmed)
  - MDG, Data Quality Management validation rules
  - Rules from the Business Rules Framework (BRF+)
  - Coded rules using the BAdI USMD\_RULE\_SERVICE
- Custom derivations (modeled and programmed)
  - Rules from the Business Rules Framework (BRF+)
  - Coded rules using BAdI USMD\_RULE\_SERVICE
  - Coded rules using BAdI USMD\_RULE\_SERVICE\_CROSS\_ET
  - Coded enrichments using the BAdI MDG\_BS\_MAT\_API\_ENRICH\_BADI
  - Derivation with SMT-Mapping

Checks are usually carried out at *Check*, *Run Validation*, *Save*, *Submit* and *Activate* while derivations are carried out each roundtrip.

In addition, input errors found by basic technical checks (like for Units of Measure, Dates, and Currencies) are always executed after the next round trip and issued as errors.

### 3. CHECKS

#### 3.1 Model driven checks

##### 3.1.1 Code lists

The considered code list for the check comes from the Fixed Values or Value Range Table which is assigned to the domain of the data element.

In the customizing for the MM data model, you can use the 'No Existence Check' to deactivate the standard existence check for the value of the attribute.

Path: Master Data Governance -> General Settings -> Data Modeling -> Edit Data Model (VC\_USMD001)

Display View "Attributes": Overview						
Visualize Data Model						
Dialog Structure Inactive Data Models Entity Types Attributes Business Object Entity Types for H Hierarchy Attr Relationships Reuse Active Areas						
Attribute	Data Element	Required Field	Currency/Unit	Search Help	No Existence Check	Description
BEGRU	BEGRU	<input type="checkbox"/>			<input type="checkbox"/>	Authorization Group
BISMT	BISMT	<input type="checkbox"/>			<input type="checkbox"/>	Old Material Number
EXTWG	EXTWG	<input type="checkbox"/>			<input checked="" type="checkbox"/>	External Material Group
FERTH	FERTH	<input type="checkbox"/>			<input type="checkbox"/>	Production/Inspection Memo
GEWEI_MAT	GEWEI	<input type="checkbox"/>		GEWEI_F4_MASS	<input type="checkbox"/>	Weight Unit
GROES	GROES	<input type="checkbox"/>			<input type="checkbox"/>	Size/Dimensions

##### 3.1.2 Cardinality check

A field is mandatory if the referencing relationship has cardinality 1:1.

##### 3.1.3 Format check

Data element is used for format check.

##### 3.1.4 Required fields

Attributes can be defined as required fields.

Display View "Attributes": Overview						
Visualize Data Model						
Dialog Structure Inactive Data Models Entity Types Attributes Business Object Entity Types for H Hierarchy Attr Relationships Reuse Active Areas						
Attribute	Data Element	Required Field	Currency/Unit	Search Help	No Existence Check	Description
BEGRU	BEGRU	<input type="checkbox"/>			<input type="checkbox"/>	Authorization Group
BISMT	BISMT	<input type="checkbox"/>			<input type="checkbox"/>	Old Material Number
EXTWG	EXTWG	<input type="checkbox"/>			<input checked="" type="checkbox"/>	External Material Group
FERTH	FERTH	<input type="checkbox"/>			<input type="checkbox"/>	Production/Inspection Memo
GEWEI_MAT	GEWEI	<input type="checkbox"/>		GEWEI_F4_MASS	<input type="checkbox"/>	Weight Unit
GROES	GROES	<input type="checkbox"/>			<input type="checkbox"/>	Size/Dimensions
LABOR	LABOR	<input type="checkbox"/>			<input type="checkbox"/>	Laboratory/Design Office
LVORM_MAT	LVOMA	<input type="checkbox"/>			<input type="checkbox"/>	Flag Material for Deletion at Client Level

### 3.2 Re-use of existing check logic in ERP backend system or S/4HANA backend system

#### 3.2.1 Customizing OMT4

Only messages with severity E from the material API are shown in MDG. These can be messages for the required fields or additional messages defined in the customizing OMT4. (See Customizing, under Logistics -> General -> Material Master -> Basic Settings -> Define Attributes of System Messages).

With the customizing of the change request type, you can decide if these E messages are shown as errors or as warning messages during the governance processes. But at activation there is no conversion of the messages.

New function in MDG6.1 (see also Note 1701437, Chapter Message configuration in OMT4): Transaction OMT4 provides configuration for message severity. Messages can be raised as errors, warnings, or not at all. MDG-M only supports the severity configuration for a subset of these messages.

- If a message from this subset is configured as a warning, it is also shown as warning in MDG-M
- If a message not in this subset is configured as a warning, it is not shown in MDG-M
- Messages configured as errors are always shown as errors.

Supported subset:

- M3, configuration for messages 132, 159, 285, 347, and 348 is supported
- MM, configuration for messages 189, 312, and 657 is supported
- MH and WE, configuration for messages is not supported

### 3.2.2 **Code list**

The Material API checks the Fixed Values or Value Range Tables which are assigned to the domain of the data elements in the material master tables.

### 3.2.3 **Business checks (Material API)**

The Material API is called at *Check*, *Run Validation*, *Save*, and *Activate*

- *Check and Save* only check the changed material
- *Run Validation* checks the complete change request
- In Single Item Maintenance UI and Mass Maintenance UI, File import, etc.

Some of the settings of the Material Master Customizing are considered in MDG through the Material API.

All settings from the section *Basic Settings* are considered. For example, the output format of the material number and the material types. All settings from the section *Settings for Key Fields* are considered. For example, the definition of material groups and basic material and the settings for EAN's.

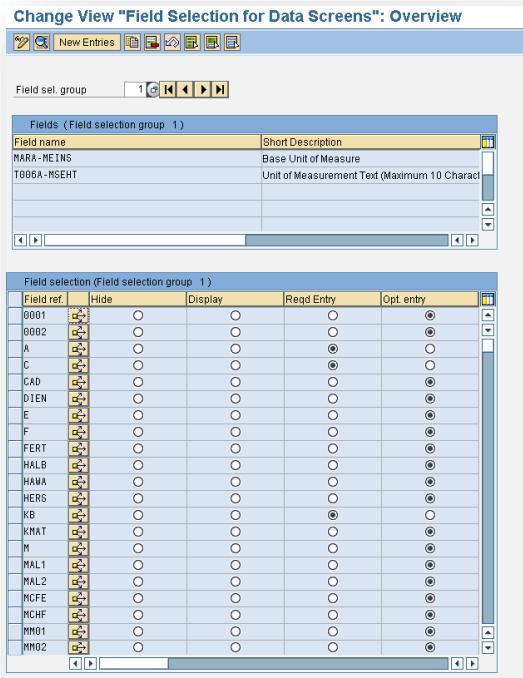
In addition, User Exit (SMOD) MGA00001 and BADI\_MATERIAL\_CHECK are considered in MDG-M.

### 3.2.4 **Required fields as defined in material master customizing (T130F) and hardcoded (Material API)**

The field properties for required fields in the Material Master Customizing are considered in the MDG checks. The section *Field Selection* is considered. Changes affect the field's properties in MDG-M. There, you define if a field is required (table T130F).

The field properties are determined by the field selection group. For the specification of the properties of a field selection group, there are several criteria. These are called field selection references. Six criteria (field selection references) are relevant for the definition of the field selection attributes (see consulting note 199165).

The six field selections which are relevant for MDG\_M are: Industry sector, Material type, transaction MM01/MM02 and field selection control KB. Also considered are SAP1 and SAP2, but must not be changed by customer. As MDG can't distinguish between the backend material create and change transaction both MM01 and MM02 checks are considered in the Check.



For the determination of UI field property values, see chapter 5.5. in <http://scn.sap.com/docs/DOC-30192>. Here only the settings for MM01 are considered for performance reasons, not MM02. Therefore, SAP recommends keeping the field control for MM01/MM02 in sync to avoid nontransparent messages.

Also, BADI\_MAT\_F\_SPEC\_SEL is considered in MDG-M.

### 3.3 Custom Checks

**MDG, Data Quality Management validation rules** are called at Check, Run Validation, Save, and Activate.

**Check Entities** from the BRF+ Rules and the Check BAdI are called at *Check, Run Validation, Save, and Activate*.

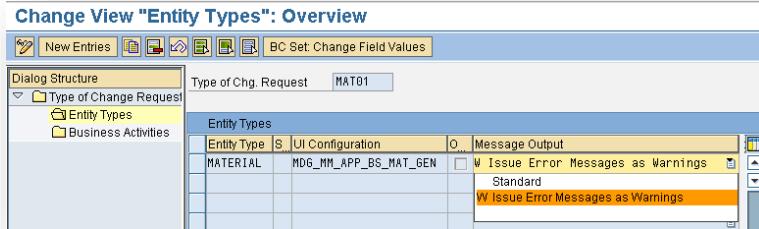
**Check Change Request** from the Check BADI and the BRF+ Rules is called at *Run Validation and Activate*.

- *Check/Save* only check the changed Material
- *Run Validation* checks the complete Change Request
- Valid for Single Item Maintenance UI and Mass Maintenance UI, File upload, Data import ...

### 3.4 Customizing CHANGE REQUEST Type for Check messages

Only messages with type error from the backend Material API are shown in MDG-M. Additional messages for checks can be created with BRF+ or with a BADI. These can be shown as abort, errors, warnings, or info. Within the customizing of the change request type, you can decide if these are shown as errors or as warning messages.

Note: Input errors found by basic technical checks (like for Units of Measure, Dates, and Currencies) are always issued as Errors.



### 3.4.1 Customizing setting: Issue Error as Warnings

#### EhP5 and \* Issue Error as Warnings MAT01

	Round trip	Check	Submit/Save	Activate	Excel upload
Derive Entity BRF+ *	A→I; E→I; W→I; I→I	Only on Screen of the Entity: A→I; E→I; W→I; I→I	hidden	hidden	A→I; E→I; W→I; I→I
Derive BADI	A→I; E→I; W→I; I→I	Only on Screen of the Entity: A→I; E→I; W→I; I→I	hidden	hidden	A→I; E→I; W→I; I→I
Check Entity BRF+ *	n. a.	A→E; E→W; W→W; I→I	A→E; E→W; W→W; I→I	n. a.	A→E; E→W; W→W; I→I
Check Entity BADI *		A→E; E→W; W→W; I→I	A→E; E→W; W→W; I→I		A→E; E→W; W→W; I→I
Check Material API (only A, E messages) *	n. a.	E → W	E → W	E→E	E → W
CR check's (only E messages)	n. a.	E → E	E→E	Before Validation: E->E	n. a.
Check CR BRF+	n. a.	n. a.	n. a.	A→E; E→E; W→W; I→I	n. a.
Check CR BADI	n. a.	n. a.	n. a.	A→E; E→E; W→W; I→I	n. a.
Duplicate Check	n. a.	W	W	W	n. a.

Messages from Derives are allowed converted into info messages.

### 3.4.2 Customizing setting: Standard

EhP5 and Standard ZZMAT01					
	Round trip	Check	Submit/Save	Activate	Excel upload
Derive BRF+	A→I; E→I; W→I; I→I	Only on Screen of the Entity: A→I; E→I; W→I; I→I	hidden	hidden	A→I; E→I; W→I; I→I
Derive BADI	A→I; E→I; W→I; I→I	Only on Screen of the Entity: A→I; E→I; W→I; I→I	hidden	hidden	A→I; E→I; W→I; I→I
Check Entity BRF+ *	n. a.	A→E; E→E; W→W; I→I	A→E; E→E; W→W; I→I	n. a.	A→E; E→E; W→W; I→I
Check Entity BADI *		A→E; E→E; W→W; I→I	A→E; E→E; W→W; I→I		A→E; E→E; W→W; I→I
Check Material API (only A, E messages) *	n. a.	E → E	E → E	E→E	E→E
CR check's (only E messages)	n. a.	E → E	E→E	Before Validation: E→E	n. a.
Check CR BRF+	n. a.	n. a.	n. a.	A→E; E→E; W→W; I→I	n. a.
Check CR BADI	n. a.	n. a.	n. a.	A→E; E→E; W→W; I→I	n. a.
Duplicate Check	n. a.	W	W	W	n. a.

Messages from Derives are allowed converted into info messages.

### 3.4.3 Configure Properties of Change Request Step

In this Customizing activity, you determine settings for the execution of a change request step for a change request type.

#### Enrichment Spots and Checks per Change Request Step View

In this view, you can complete the following actions for a change request step:

- Specify which enrichment spots and **checks** are relevant.
- View the execution sequence for enrichment spots and **checks**.
- Control the display of messages by specifying a message output. For example, you can ensure some messages display only as warnings.
- Determine whether a check is always executed or only executed when changes occur.

#### Entity Types per Change Request Step View and Attributes per Change Request Step View

You can complete the following actions for a change request step:

- Specify which fields are relevant, and which relevant fields are required, by setting field properties. For example, you can make a required field optional.
- Reduce the number of **checks** applied to fields by specifying a Check Logic for an entity type.
- You can only define properties for entity types, attributes, or relationships that are governed. You can apply this setting in Customizing for *Master Data Governance* under *General Settings -> Process Modeling -> Define Governance Scope*.

For the field properties topic please see also the How To Guide for the UI. You can find it here: <http://scn.sap.com/docs/DOC-30192>. See chapter for Field Properties.

### 3.4.4 Enforce check at any roundtrip

You can enforce a check by any roundtrip in the material UI by enhancing the method CHECK\_DATA of the class CL\_MDG\_BS\_BOL\_TRANSACTION.

You can enhance with an overwrite exit the IF statement checking the event ID or you can implement a post exit enhancement, copying the core code of the IF statement. In any case, you should care for notes or enhancements of this method in future to transfer the corrections to your enhancement in an appropriate manner.

The event ID 'XROUNDTRIP' is raised, if the user presses the Enter key, and if a value help is closed (to refresh the description of the value). If necessary, you can also check for the event ID 'ROUNDTrip' additionally. Depending on your intension, you can check only for one of these event IDs or for both. Check in independent scenarios, which combinations provides the right number of checks in your case.

### 4. MDG, DATA QUALITY MANAGEMENT VALIDATION RULES

You can collaboratively describe, catalog, and implement rules for data quality using a central rule repository. These validation rules can be used for data quality evaluations and for checking data in change requests, in consolidation and in mass processing for products and business partners.

More information:

- Blog:  
<https://blogs.sap.com/2019/11/08/master-data-quality-management-with-sap-master-data-governance-on-sap-s-4hana-1909/>
- How-To Guides:  
<https://wiki.scn.sap.com/wiki/display/SAPMDM/Documentation%3A+Data+Quality+Management+with+SAP+Master+Data+Governance>
- SAP Online Help:  
[https://help.sap.com/docs/SAP\\_S4HANA\\_ON-PREMISE/6d52de87aa0d4fb6a90924720a5b0549/47cd340e7449473698b6f604c977ae2b.html](https://help.sap.com/docs/SAP_S4HANA_ON-PREMISE/6d52de87aa0d4fb6a90924720a5b0549/47cd340e7449473698b6f604c977ae2b.html)

#### 4.1 Example for validation rule to check if material status is allowed

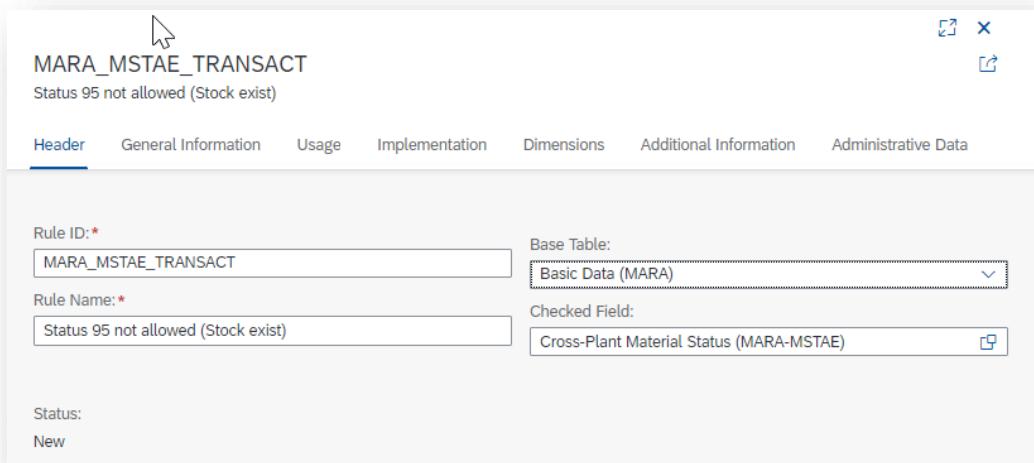
##### Scenario:

The material status 95 is only allowed if no stock is available anymore.

##### Solution:

Create a validation rule with procedure call.

Create a validation rule, enter a Rule ID and a Rule Name, then save the rule. Choose Basic Data (MARA) as the Base Table and choose MARA-MSTAE as the Checked Field. Click Create.



MARA\_MSTAE\_TRANSACT

Status 95 not allowed (Stock exist)

Header General Information Usage Implementation Dimensions Additional Information Administrative Data

Rule ID: \*

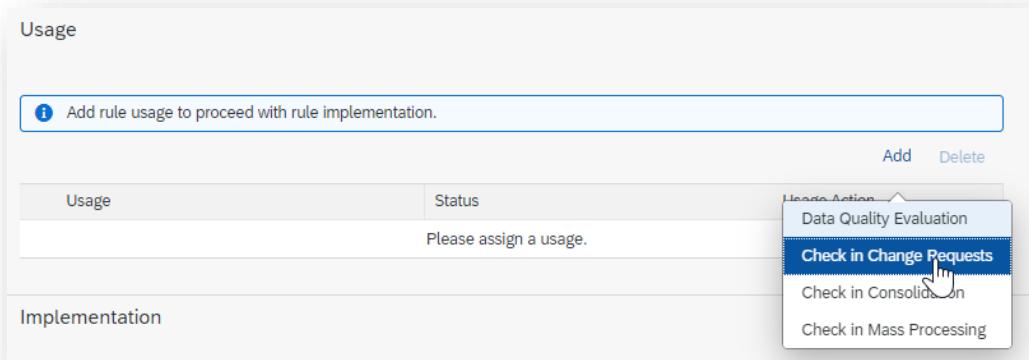
Base Table:

Rule Name: \*

Checked Field:

Status: New

In the Usage section, select Add to add a usage. Select Check in Change Requests and click Prepare.



Usage

Add rule usage to proceed with rule implementation.

Usage	Status
	Please assign a usage.

Usage Action

Data Quality Evaluation

Check in Change Requests

Check in Consolidation

Check in Mass Processing

## How-To: Maintain Check and Derivation Rules in MDG for Material

---

In the Implementation section, select the link for the scope expression. The BRFplus Workbench is launched and an initial expression is displayed that needs to be adapted to the desired scope of the rule. Maintain the Scope Expression as shown in the screenshot:

The screenshot shows the SAP BRFplus Workbench interface. The title bar says "BRFplus Workbench". The left sidebar has "Repository" selected under "Catalog". The main area shows a rule named "Boolean: MARA\_MSTAE\_TRANSACT\_SCP". The "Detail" tab is selected. The logic schema is defined as follows:

```
Boolean Logic Schema: <1>
If
  MARA Key...X-Plant Status is equal to 95 (No transactional data)
Then Boolean is true, else it is false
```

In the Implementation section, select the link for the condition expression. The BRFplus Workbench is launched, and an initial expression is displayed that needs to be adapted to the desired condition of the rule. Create a

Procedure Call Expression (SAP Online Help: <https://help.sap.com/viewer/9d5c91746d2f48199bd465c3a4973b89/latest/en-US/5461974e8ac74e6a875836ea721added.html> ).

The screenshot shows the SAP BRFplus Workbench interface. The title bar says "BRFplus Workbench". The left sidebar has "Repository" selected under "Catalog". The main area shows a rule named "Boolean: MARA\_MSTAE\_TRANSACT\_CON". The "Detail" tab is selected. The logic schema is defined as follows:

```
Boolean Logic Schema: <1>
If
  not(
    ZEM Check Trans Dat
  )
Then Boolean is true, else it is false
```

## How-To: Maintain Check and Derivation Rules in MDG for Material

Assign a method of a ABAP class previously created, which check the stock, to the procedure call expression.

The screenshot shows the SAP Class Builder interface. In the Repository Browser, the class **ZCL\_EM\_CHECK\_TRANS\_DATA** is selected. Under the **Methods** section, the method **CHECK\_TRANSACTIONAL\_DATA** is highlighted. The code editor displays the ABAP code for this method:

```
1 METHOD check_transactional_data.
2   DATA: lv_matnr      TYPE matnr.
3
4   DATA: lt_mard      TYPE mard_tab.
5
6   "Get material number
7   lv_matnr = cl_mdg_bs_mat_settings->get_matnr( ).
8
9
10  * MARD
11  cl_mdg_bs_mat_stock_service->read_mard_stock_inventory(
12    EXPORTING
13      iv_matnr      = lv_matnr
14      iv_filter     = abap_true
15    IMPORTING
16      et_mard_stock = lt_mard ).
17
18  IF lines( lt_mard ) GT 0.
19    r_boolean = abap_true.
20  ELSE.
21    r_boolean = abap_false.
22  ENDIF.
23
24
25
26
27
28 ENDMETHOD.
```

For more examples of transaction data methods, see class **CL\_MDG\_BS\_MAT\_MSG\_DET\_MMAM**. Map the parameter.

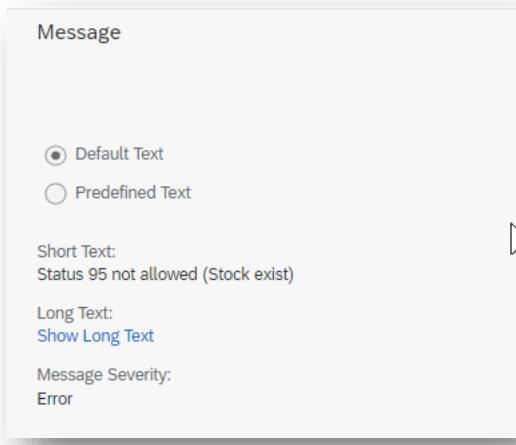
The screenshot shows the **Procedure Call** dialog for the method **CHECK\_TRANSACTIONAL\_DATA**. The **General** tab is selected. The **Result Data Object** is set to **Boolean**. The **Call Type** is **Static Method**, and the **Class Name** is **ZCL\_EM\_CHECK\_TRANS\_DATA**. The **Description** is **Check transactional data for status 95**. The **Detail** tab is also visible, showing the **Interface Name**, **Method Name** (**CHECK\_TRANSACTIONAL\_D...**), and **Description** (**No Description**). The **Mapped Parameters** section shows a single parameter **R\_BOOLEAN** with a type of **Boolean**. The **Exception Handling** section indicates that no exceptions are defined.

Save and activate the BRFplus expression, then navigate back to display the validation rule. Check if the status of both the scope expression and the condition expression is Active.

## How-To: Maintain Check and Derivation Rules in MDG for Material

---

In the Message section you can choose between Default or Predefined Text.



Select Approve to switch the status of the rule to Approved. In the Usage section, select Enable to enable the usage.

MARA\_MSTAE\_TRANSACT  
Status 95 not allowed (Stock exist)

Base Table: Basic Data (MARA)  
Checked Field: Cross-Plant Material Status (MARA-MSTAE)      Status: **Approved**

General Information    Usage    Implementation    Message    Dimensions    Additional Information    Administrative Data

**Business Details**

Business Description:  
If the material has the status 95, the system checks whether stock exists for this material. If stock exists, a error message is raised. [Less](#)

Business Reason:  
—

Scope:  
MARA-MSTAE = 95

Link:  
—

Link Text to Display:  
—

Link URL:  
—

**Contacts**

Rule Owner:  
**MDQ\_FAST**

Implementation Expert:

Business Contact:  
—

Data Owner:  
—

**Usage**

Usage	Status	Usage Action
<input type="radio"/> Check in Change Requests	Enabled	<a href="#">Disable</a>

**Implementation**

Scope	Condition
Expression: <b>MARA_MSTAE_TRANSACT_SCP</b>	Expression: <b>MARA_MSTAE_TRANSACT_CON</b>
Status: <b>Active</b>	Status: <b>Active</b>

## How-To: Maintain Check and Derivation Rules in MDG for Material

Create a Change Request Type where Validation Rules are enabled.

Checks and Enrichment Spots	Sequence	Message Out...	Relevant	Execution
#00 Basic Check	0	Standard	<input checked="" type="checkbox"/>	1 Always executed
#01 Authorization Check	0	Standard	<input checked="" type="checkbox"/>	1 Always executed
#02 Duplicate Check	99	Standard	<input type="checkbox"/>	2 Executed when data changes
#03 Validation Rules (BRF+)	0	Standard	<input checked="" type="checkbox"/>	1 Always executed
#04 BADI Validations	0	Standard	<input checked="" type="checkbox"/>	1 Always executed
#05 Existence Check	0	Standard	<input checked="" type="checkbox"/>	1 Always executed
#06 Reuse Area Check	0	Standard	<input checked="" type="checkbox"/>	1 Always executed
#07 Validation Rules (Data Quality)	0	Standard	<input checked="" type="checkbox"/>	1 Always executed

Create a material change request for a material for which inventory exists. Change the Cross-Plant Material Status to 95. If you choose Check, Send, or Save, the message is displayed.

Material: Pump PRECISION 100

Edit Expand all Collapse all Switch to Active Version Refresh Check

Change Request Edit

General Notes Attachments

General Data

Change Request: 462

\* Description: Change Material to Status 95

Priority:

Due Date:

Reason:

Change Management

Basic Data Edit Details Replication Status

General Data

Material: EM-100

\* Base Unit of Measure: PC Piece

\* Material Type: ZMDQ MDQ Testing

Industry Sector: M Mechanical engineering

Material Group: 01 Tools

Old Material Number:

Authorization Group:

Cross-Plant Material Status: 95 No transactional data

Batch Management: No

Serialization Level: Serialization within the stock material number

Approved Batch Record Required Indicator: No

Division: 01 DIVISION

Dimension

Net Weight: 250,000 KG Kilogram

Gross Weight: 280,000 KG

Length: 0,000

Width: 0,000

All 2 [Info] Messages refer to Material EM-100 Status 95 not allowed (Stock exist)



## 4.2 Example for validation rule to check if a division is provided for finished goods

### Scenario:

This example describes how to create a validation rule to check if a division is provided for finished goods.  
Rule: For all products of type Finished Product (MARA-MTART = FERT), the field Division (MARA-SPART) must be filled.

### Solution:

See SAP Online Help <https://help.sap.com/viewer/6d52de87aa0d4fb6a90924720a5b0549/latest/en-US/b6ae47a1766b4f7090b85c673e47618f.html>. In the Usage section, add usage “Check in Change Requests”.

## 4.3 Example for validation rule with Extra Data Provisioning

### Scenario:

This example describes how to create a validation rule where you want to access data which is not in the context objects and that resides in a different table of the data model. You can use one of the predelivered procedure calls in BRF+ in the application ZMDQ\_194.

### Solution:

See SAP Online Help <https://help.sap.com/viewer/6d52de87aa0d4fb6a90924720a5b0549/latest/en-US/6f1ab68657f7418cb5f40fe48d69f871.html>. In the Usage section, add usage “Check in Change Requests”.

### 5. CHECKS WITH BRF+

Path: MDGIMG ->General Settings -> Data Quality and Search -> Define Validation and Derivation Rules.  
In this Customizing activity, you define the validations and derivations for a data model. This activity calls the Web Dynpro application *Definition of Rules for Validations and Derivations* (USMD\_RULE). You can define validation and derivation rules for the data model MM. The system automatically generates the data objects used for the rule for the selected data model. For a precise description of the procedure, choose *Help* on the initial screen of this Web Dynpro application

#### Important

There is a Naming Convention for Trigger function nodes in the catalog structure.

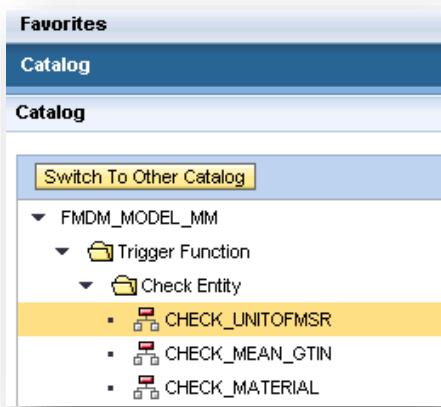
The naming convention for check trigger function nodes of a catalog structure is CHECK\_<name of entity type>, for example, CHECK\_MATERIAL

The naming convention for derivation trigger function nodes of a catalog structure is DERIVE\_<name of entity type>, for example, DERIVE\_MATERIAL

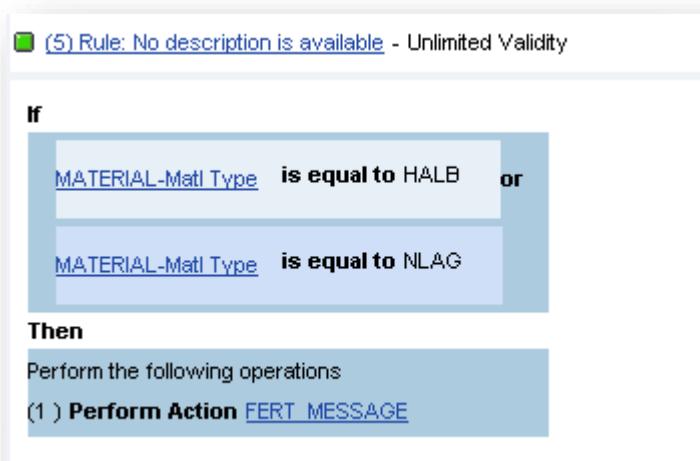
#### Important

If you update/change the data model MM, the data objects are updated. But if you used a data object already in a function as a signature, it is not automatically updated. Please see SAP Note 1731488 for how to proceed.

#### 5.1 Example for CHECK\_MATERIAL



##### 5.1.1 Simple Checks



The screenshot shows the BRF+ editor interface. At the top, it says '(5) Rule: No description is available - Unlimited Validity'. The 'If' section contains two conditions separated by 'or': 'MATERIAL-Matl Type is equal to HALB' and 'MATERIAL-Matl Type is equal to NLAG'. The 'Then' section contains the instruction 'Perform the following operations' followed by '(1) Perform Action FERT\_MESSAGE'.

**Log Message (Act) FERT\_MESSAGE** | Change Mode |  Active

Back | Display | Check | Save | Activate | Mark As Obsolete | Delete

**General**

Name:	FERT_MESSAGE	Short Text:	FERT_MESSAGE
Application:	FMDM_MODEL_MM	Access Level:	Application

Show More

Action Followup Actions

**Detail**

External Identification Mode: No External Identification in Log

**Log Message:**

(1) Abort message with text: Please use material type 'FERT' 

You can also use Message with message class USMD5 a message number 000, where you can specify variables.

## 5.2 Example for CHECK\_MEAN\_GTIN

### 5.2.1 Simple Checks

**Rules**

Insert Rule | Insert Exit Condition

(1) Rule: No description is available - Unlimited Validity

If  
Europäis...-EAN/UPC is initial

Then  
Perform the following operations  
(1) Perform Action MESSAGE\_EAN

**Log Message (Act) MESSAGE\_EAN** | Change Mode | Active

Back | Display | Check | Save | Activate | Mark As Obsolete | Delete

**General**

Name:	MESSAGE_EAN	Short Text:	MESSAGE_EAN
Application:	FMDM_MODEL_MM	Access Level:	Application

Show More

Action | Followup Actions

**Detail**

External Identification Mode: No External Identification in Log

**Log Message:**

(1) Error message with text: Please specify at least one EAN  

## 5.3 Example for CHECK\_UNITOFMSR

### 5.3.1 Simple Checks

(2) Rule: No description is available - Unlimited Validity

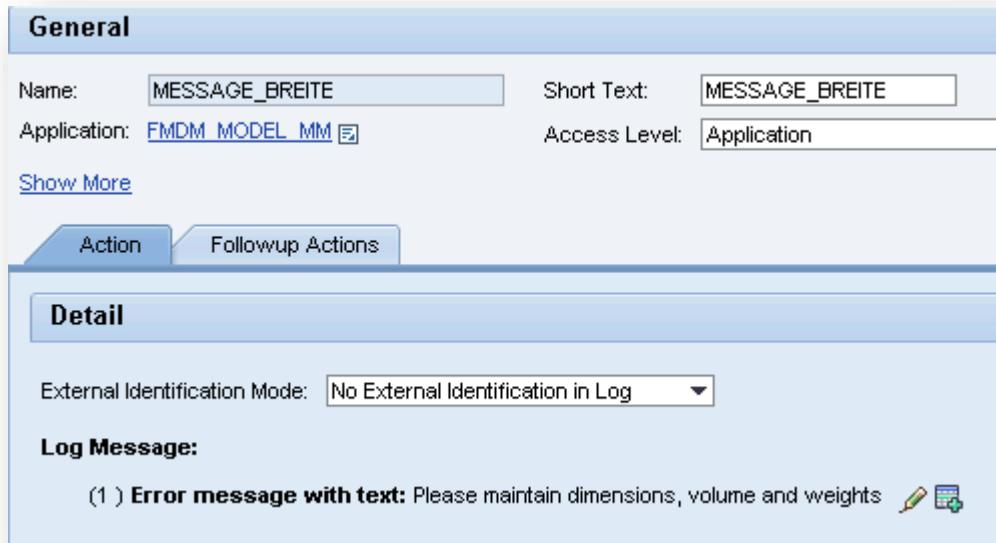
If

- Mengenei...Width      is initial  
or  
Mengenei...Length      is initial  
or  
Mengenei...Height      is initial  
or

Then

Perform the following operations

(1) Perform Action MESSAGE\_BREITE



### 5.3.2 Rule to compare Net and Gross weight for Base Unit of Measure

There is a check in the backend transactions for Material which compares the Net and Gross weight for the Base Unit of Measure. If the net weight is greater than the gross weight, the message 'The net weight is greater than the gross weight' is raised (see TA SE91 Message class M3 and message 176).

The backend check does not show up in MDG, because message M3 176 is not in the supported subset (see chapter4.2.1 Customizing OMT4). Therefore, this message is only shown in MDG if the severity is an error message.

**Solution:**

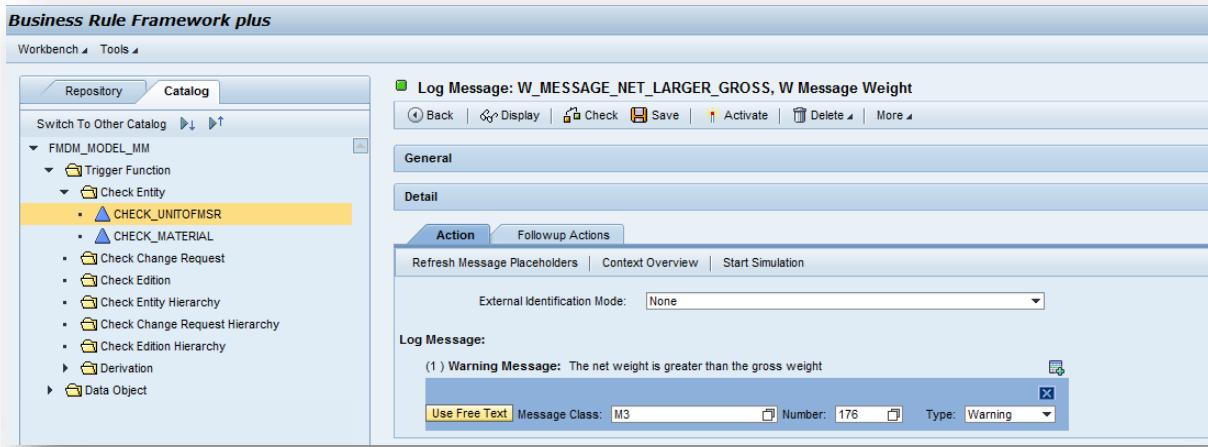
## How-To: Maintain Check and Derivation Rules in MDG for Material

You can create a BRF+ rule if the check is also necessary in MDG.

The screenshot shows the 'Business Rule Framework plus' interface. In the left sidebar, under 'FMDM\_MODEL\_MM / Trigger Function / Check Entity', the 'CHECK\_UNITOFMSR' function is selected and highlighted with a yellow background. The main panel displays the 'Function: CHECK\_UNITOFMSR' configuration screen. The 'General' tab is active, showing the function name and a 'Start Simulation' button. Below it, the 'Mode' dropdown is set to 'Event Mode'. The 'Assigned RuleSets' tab is selected, showing a single rule named 'CHECK\_GROSS\_AND\_NETWEIGHT' with the text 'Check Weight'. The 'Code Generation' tab is also present. The 'Context' section lists components: 'MATERIAL' (Text, Structure) and 'UNITOFMSR' (Units of Measure, Structure). The 'Result Data Object' section shows a 'Data Object' named 'Actions'.

This screenshot shows the same interface after creating a new rule. The 'Assigned RuleSets' tab now contains two rules: 'CHECK\_GROSS\_AND\_NETWEIGHT' (selected) and 'CHECK\_WEIGHT'. Both rules have the text 'Check Weight' and are marked as executable. The other tabs ('Signature' and 'Code Generation') remain visible.

This screenshot provides a detailed view of the 'Ruleset: CHECK\_GROSS\_AND\_NETWEIGHT, Check Weight' configuration. The 'General' tab shows the rule is enabled, uses function 'CHECK\_UNITOFMSR', has no preconditions, and is assigned to 'Check Weight'. The 'Detail' tab shows the rule has one variable: 'Number of Rules' (1), 'Number of Variables' (0), and 'Priority' (00). The 'Rules' tab displays the rule logic: 'If' 'Units of ... Unit' is equal to 'MaterialUnit', 'and' 'MaterialNet' is greater than 'Units of ... Gross', 'and' 'Units of ... Gross' is not initial. The 'Then' section shows a single action: '(1) Perform W Message Weight'.



## 5.4 Example for CHECK\_MARCBASIC

### 5.4.1 Warning, if Serial Number Profile is changed

#### Scenario:

There should be a warning message when you change the Serial Number Profile (SERNP) for a material which has stock.

#### Solution:

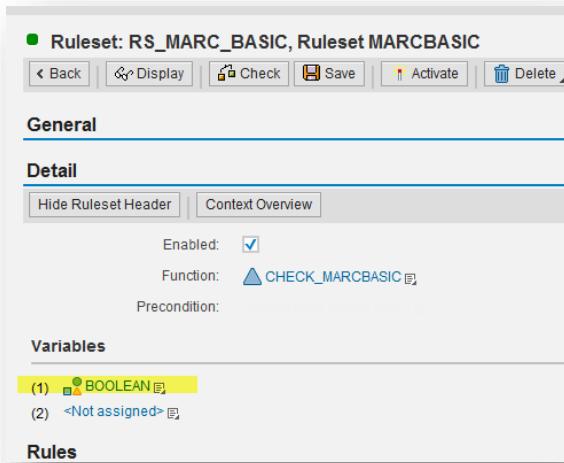
Create a ruleset for function CHECK\_MARCBASIC.

The first rule should check if stock exists for the material after checking that the serial number profile was changed.

The second rule should raise a message if first rule is true.

#### Step by Step:

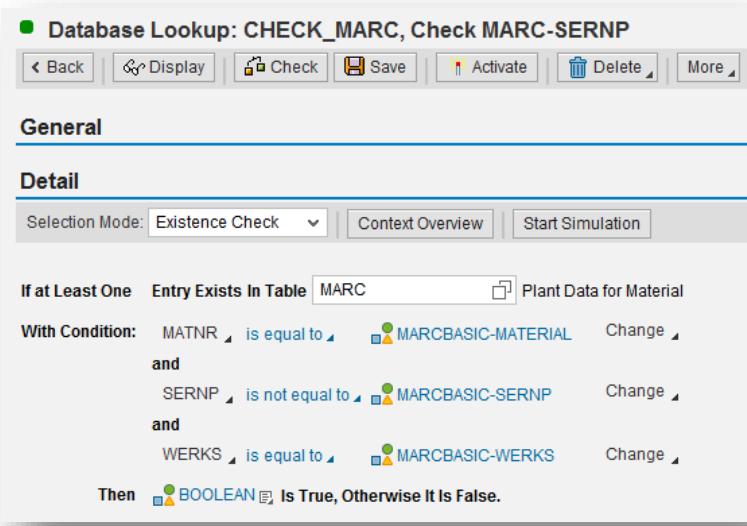
- Create ruleset
- Assign variable BOOLEAN



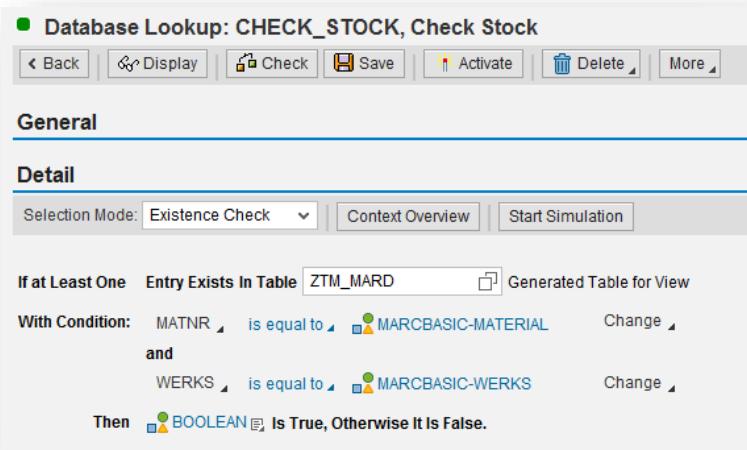
## How-To: Maintain Check and Derivation Rules in MDG for Material

---

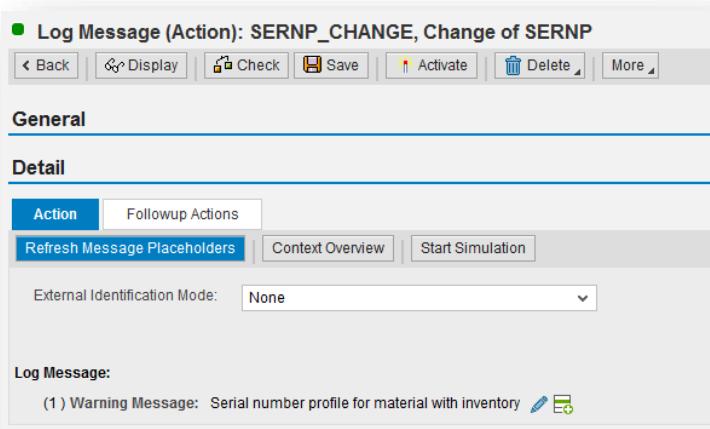
- Create database lookup CHECK\_MARC



- Create database lookup CHECK\_STOCK



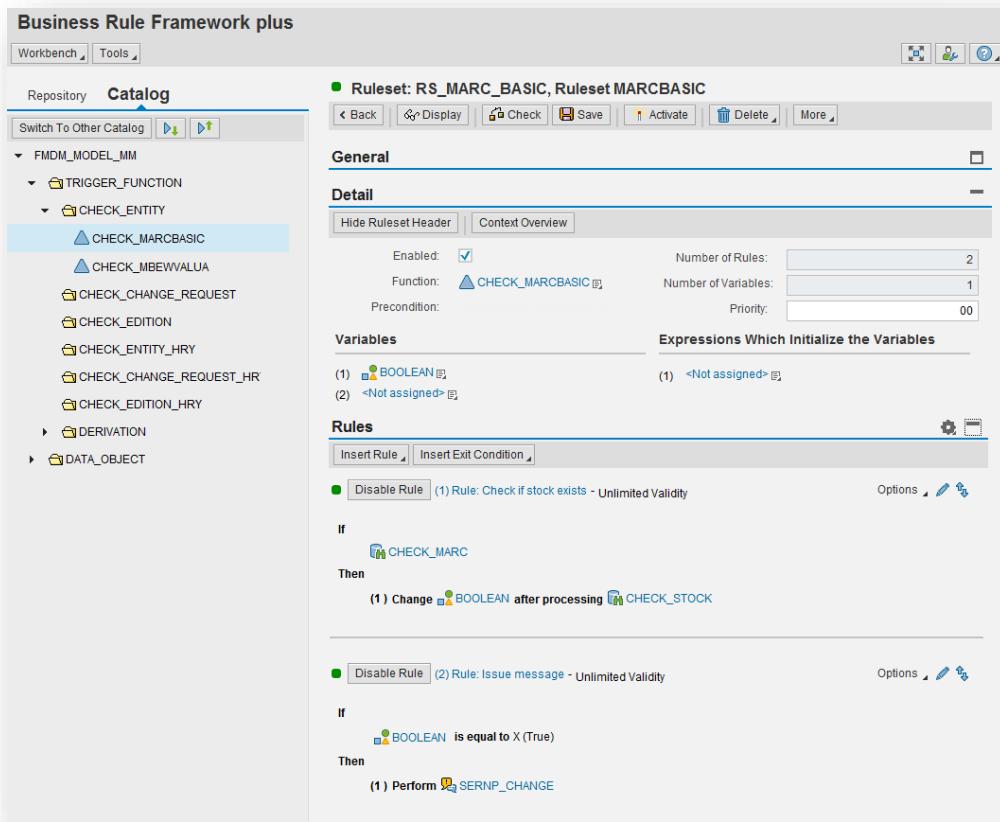
- Create Log Message



## How-To: Maintain Check and Derivation Rules in MDG for Material

---

- Create Rules



## 6. DERIVATIONS

### 6.1 Custom derivations

When creating a derivation, ensure that it is only executed if needed because the data is marked as changed and if you close the UI before saving it a confirmation popup appears. See also customer message 10181 2012: "Unsaved data confirmation popup after request submission".

#### 6.1.1 *Derivation using BRF+/ BAdI USMD\_RULE\_SERVICE*

Derive Entity from BRF+ Rules and Derivation BADI are called at each Entity (per roundtrip).

- Valid for Single Item Maintenance and UI Multi-Record Processing, File upload, Data import etc.
- Not valid for application USMD\_MASS\_CHANGE. There you select a set of entities of one type, and change single attributes for all these entities to the same value. The derivation in that application is not supported, as it would change other attributes than the specified ones and would break the logic of that application, especially the visualization of the mass change result.
- Messages from Derives are always converted into info messages.

#### 6.1.2 *EhP6: Derivation using BAdI USMD\_RULE\_SERVICE\_CROSS\_ET*

The BAdI is called at least once at every roundtrip (Technical note: the BAdI is triggered in the Flush Method of the Governance API).

- Valid for Single Item Maintenance UI, Multi-Record Processing, application USMD\_MASS\_CHANGE and Data import
- Messages from Derives are always converted into info messages.
- The BAdI gets only the attributes that are changed since the last BAdI call. Unchanged attributes are not transferred by the interface of this BAdI. You need to use the IO\_MODEL parameter (method READ\_CHAR\_VALUE or RETRIEVE\_CHAR\_VALUE) to determine their values. For more information, see the documentation of the interface **IF\_USMD\_MODEL\_EXT**.

Hint: Set parameter IF\_NO\_FLUSH to ABAP\_TRUE to avoid additional call of the Flush Method.

For more Information see chapter 9 Cross-Entity Derivation with BAdI USMD\_RULE\_SERVICE\_CROSS\_ET

#### 6.1.3 *Comparison of Derivations*

It is possible to use all 3 types of Derivations in parallel.

Recommendation for the BAdIs is to use USMD\_RULE\_SERVICE for derivation if possible for reasons of transparency (self-contained change of entity) and USMD\_RULE\_SERVICE\_CROSS\_ET only if this is necessary. The table below gives suggestions for the use of the different custom derivations:

	<b>BRF+</b>	<b>BAdI USMD_RULE_SE RVICE</b>	<b>BAdI USMD_RULE_SERVICE_CROS S_ET</b>
<b>Create new entity</b>	-	X	X
<b>Change one existing entity</b>	X	X	X
<b>Change across entities</b>	-	-	X
<b>Delete entity</b>	-	-	X
<b>Defaulting in UI (only if entity doesn't yet exist)</b>	-  X (from MDG6.1; Notes 1879515 and 1876796)  BRF+ is also called during start of the OVP.	-  X (from MDG6.1; Notes 1879515 and 1876796)  BAdI is also called during start of the OVP.	X  The BAdI is not called at start of the OVP. Therefore, no initial defaulting is done. It is called if data was changed.

<b>Material description (TXTMI) in log on language</b>	-	X	X
<b>Material description (TXTMI) in multiple languages</b>	-	-	X

## 6.2 Change Request Enrichment BADI (MDG\_BS\_MAT\_API\_ENRICH\_BADI)

You can use this BAdI to automatically fill in fields from the material master that are not subject to master data governance.

- The changes to the data have no impact on the UI
- This BAdI will only be called during activation, when the data is being transferred back to the active area. Therefore, use only technical error messages, no business checks.

## 6.3 SMT-Mapping Staging → Active Area

You can use SMT mapping to automatically fill in fields from the material master that are not subject to master data governance.

- The changes to the data have no impact on the UI
- These defaults are considered during activation and the various checks
- No messages possible
- Use case: To fill in mandatory fields those are not part of the MM data model. For example, to automatically fill in the cross-plant material status with a fixed value (EhP6)

### 7. DERIVATIONS WITH BRF+

#### Note:

Derivation for Unit of Measures: If you want to derive Unit of Measures you should use the BAdI USMD\_RULE\_SERVICE instead of BRF+, as there are technical limitations by using BRF+ for referenced fields.

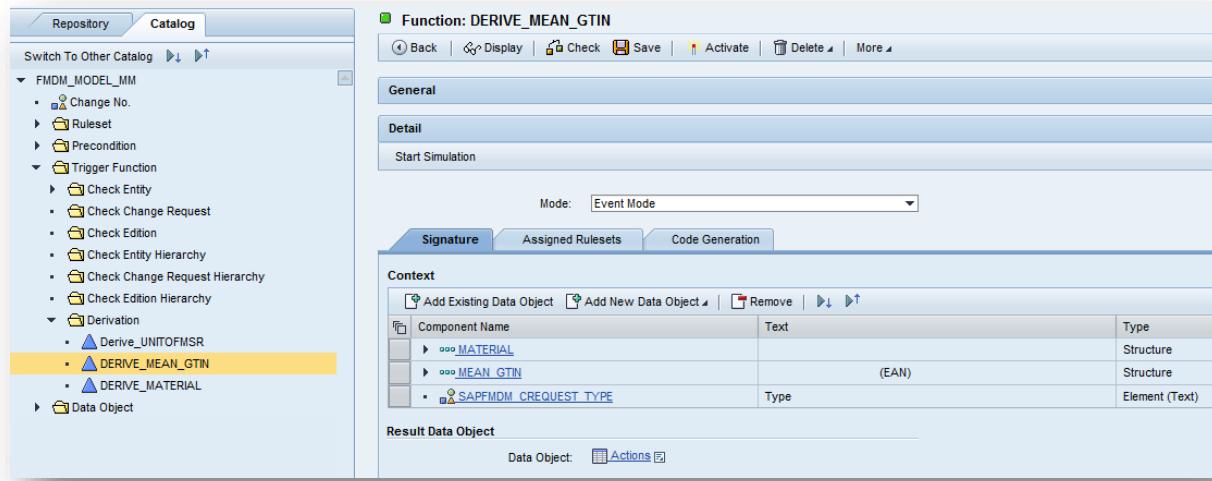
#### 7.1 Cross-Entity Rules

MDG supports certain cross-entity validations and derivations. Use cases for this include where checks and derivations span across 2 entities, one being a leading entity of storage and use type 1 and one dependent entity of storage and use type 4.

Example for entity MEAN\_GTIN with a 1: n relation:

Entities MATERIAL and MEAN\_GTIN for the derivation DERIVE\_MEAN\_GTIN.

Add the MATERIAL to the signature in addition to MEAN\_GTIN as you seen on the following picture.



The derivation is triggered during a roundtrip when a WRITE\_ENTITY is called for this entity.

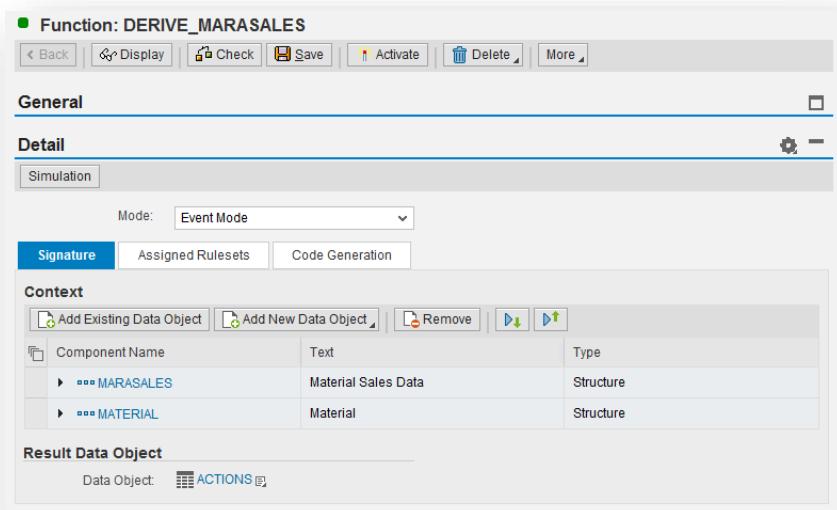
#### Note:

In the entity MATERIAL, of storage and use type 1, you also find the attribute of the 1:1 related dependent entities of type 4 (such as MARASALES and MARAPURCH), but these dependent entities cannot be used for reading or writing. If you want to have derivations for the attributes of these type 4 entities, you should create a derivation like for the 1: n related entities.

Example for entity MARASALES with 1:1 relation:

Entities MATERIAL and MARASALES for the derivation DERIVE\_MARASALES.

The derivation is triggered during a roundtrip when a WRITE\_ENTITY is called for this entity.



With EhP6 we support cross entity validation and derivation with the BAdl definition USMD\_RULE\_SERVICE\_CROSS\_ET. It is also possible to read the classification data and update other entity values with this BAdl (see chapter 9 [Cross-Entity Derivation with BAdl](#)). With MDG7.0 it is possible to update the data of an entity (called a target entity) based on the data of another entity (called a source entity) (see chapter 8 [Cross-Entity Read Access](#)).

### 7.2 Defaulting of values

With SAP Note 1876796 (prerequisite SAP Note: 1879515) it is possible to default values during material creation/change. BRF+ application is called during start of the OVP for 'Create/Change Material'. As no user input was possible until this point in time, the rule for defaulting cannot have any conditions. Expressions like if field A has value 'X', then default field B with 'Y' fail because the staging area is empty for this material. Only data like, for example, user or change request type can be used in these conditions. Be aware that these rules are also executed during the normal maintenance. If the user should be able to overwrite the default values; you should check for initial fields first.

Defaulting will only be executed if entity doesn't exist yet and all conditions of rules are valid. That means defaulting is executed at this point in time when entity is created.

## How-To: Maintain Check and Derivation Rules in MDG for Material

Example: Defaulting for the transportation group on MARASALES

The screenshot shows the SAP Ruleset editor interface. The title bar says "Ruleset: RS\_DERIVE\_MARASALES". The "General" tab is selected. Under "Detail", there are fields for "Enabled" (checked), "Function" (DERIVE\_MARASALES), "Precondition" (Current User = SMITH), and "Number of Rules" (1). The "Variables" section shows a variable "(1) <Not assigned>" and its expression "(1) <Not assigned>". The "Rules" section contains a single rule: "Rule: Defaulting Transportation Group - Unlimited Validity". This rule has an "If" condition: "MATERIAL-TRAGR is initial" and an "Then" action: "Change value of MARASALES-TRAGR to 0001 (On pallets)".

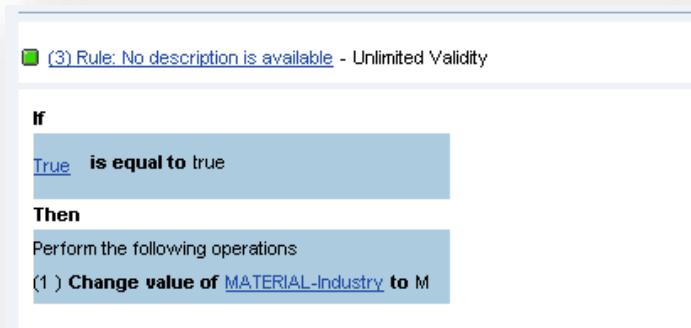
### 7.3 Example for DERIVE\_MATERIAL

#### 7.3.1 Execution time

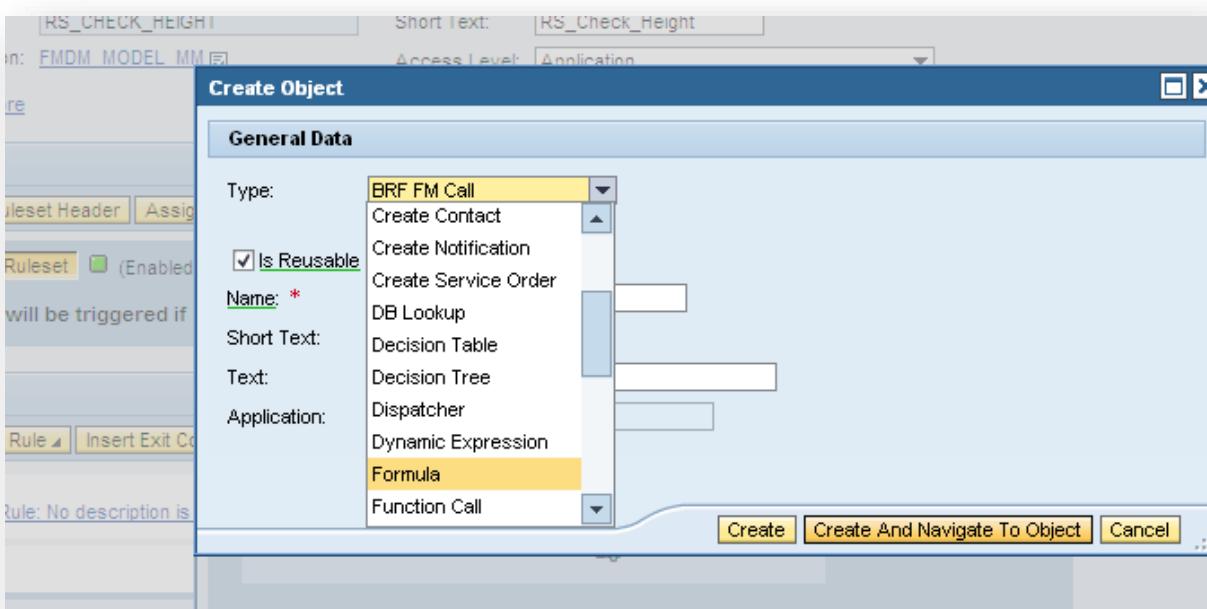
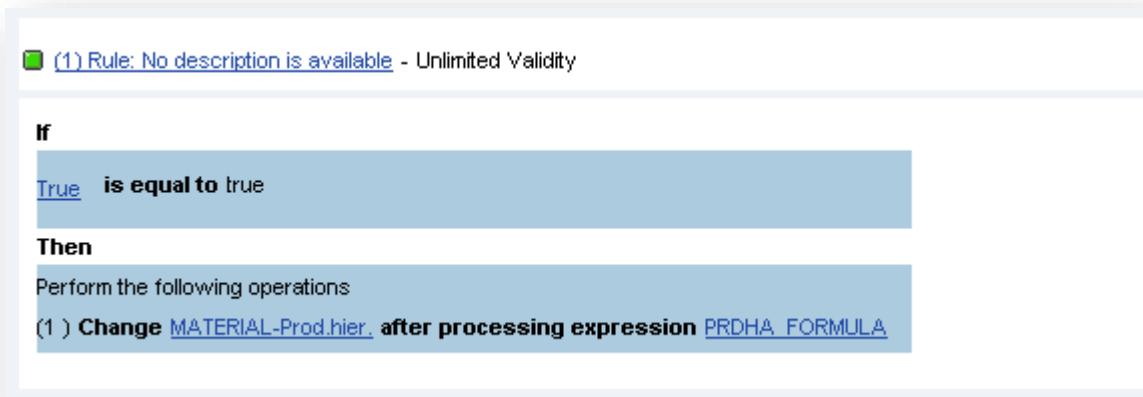
Derive Entity from BRF+ Rules and Derivation BADI are called at each Entity (per roundtrip). Make sure that the relevant field is maintained accordingly in your UI Configuration.

The screenshot shows the "Component Configuration Z01\_MDG\_BS\_MAT\_MARC\_OVERVIEW" dialog. It displays a grid of MRP parameters. Row 1: General Data. Row 2: MRP Type. Row 3: Availability check. Row 4: Plant-sp matl status. Row 5: New Group. Row 6: SchedMargin key. Rows 7-9: Empty. Row 10: In-house production (0 days). Rows 11-13: Empty. Rows 14-16: Empty. Rows 17-19: Empty. Rows 20-22: Empty. Rows 23-25: Empty. Rows 26-28: Empty. Rows 29-31: Empty. Rows 32-34: Empty. Rows 35-37: Empty. Rows 38-40: Empty. Rows 41-43: Empty. Rows 44-46: Empty. Rows 47-49: Empty. Rows 50-52: Empty. Rows 53-55: Empty. Rows 56-58: Empty. Rows 59-61: Empty. Rows 62-64: Empty. Rows 65-67: Empty. Rows 68-70: Empty. Rows 71-73: Empty. Rows 74-76: Empty. Rows 77-79: Empty. Rows 80-82: Empty. Rows 83-85: Empty. Rows 86-88: Empty. Rows 89-91: Empty. Rows 92-94: Empty. Rows 95-97: Empty. Rows 98-100: Empty. The "Action Assignment" section at the bottom shows "FPM Event ID for onEnter: ROUNDTRIP (Roundtrip)".

#### 7.3.2 Simple Rules



### 7.3.3 Example: Rules with Expression



## How-To: Maintain Check and Derivation Rules in MDG for Material

Move Token ➔ Delete Token

/	*	+	-
^	&	(	)
=	<>	,	
<	<=	>	>=
AND	OR	NOT	IF
String			
Number			
Amount			
Quantity			
Unit			
Currency			
Expression			
Comment			

**Formula Functions**

Filter by Category: Show All Functions | Filter by Text: Press Enter

Name	Description
ABS	Date and Time Functions
ARCCOS	Mathematical Functions
ARCSIN	Miscellaneous Functions
ARCTAN	System Functions
CONCATENATE	Table Functions
CONDENSE	Trims off leading and trailing spaces
CONVERT_AMOUNT	Converts an amount into the specified currency
CONVERT_QUANTITY	Converts a quantity into the specified unit
COS	Cosine

Formula PRDHA\_FORMULA | Change Mode | Active

Back | Display | Check | Save | Activate | Mark As Obsolete | Delete

**General**

Name: PRDHA\_FORMULA | Short Text: PRDHA\_FORMULA  
 Application: FMDM\_MODEL\_MM | Access Level: Application

Show More

**Detail**

Switch to Expert Mode

Result Data Object: Prod.hier.

Prod.hier. = CONCATENATE ( CONCATENATE ( 'EXXX' , MATERIAL-EZUG ) , CONCATENATE ( 'XXXXXXXX' , MATERIAL-Brand ) ) ||

Click to select the formula token and position the cursor

Move Cursor ➔ Move Token ➔ Move Token ➔ Delete Token

Selected Element: Data Object 'MATERIAL-Brand'

Name	Description
ACTIONS	Actions
MATERIAL	MATERIAL

/	*	-	+
^	&	(	)
=	<>	,	
<	<=	>	>=
AND	OR	NOT	IF
String			
Number			
Amount			
Quantity			
Comment			
Expression			

**Formula Functions**

Filter by Category: Show All Functions | Filter by Text: Press Enter

Name	Description
ABS	Amount
ARCCOS	Arc Cosinus
ARCSIN	Arc Sinus
ARCTAN	Arc Tangent
CONCATENATE	Concatenates two character strings
CONDENSE	Trims off leading and trailing spaces
CONVERT_AMOUNT	Converts an amount into the specified currency
CONVERT_QUANTITY	Converts a quantity into the specified unit
COS	Cosine
COSH	Hyperbola Cosinus

## How-To: Maintain Check and Derivation Rules in MDG for Material

The screenshot shows two overlapping windows. The top window is titled '(4) Rule: No description is available - Unlimited Validity' and contains a rule definition:

**If**  
**True is equal to true**

**Then**  
 Perform the following operations  
**(1) Change MATERIAL-Base type after processing expression FORMULA\_BASE\_TYPE**

The bottom window is titled 'Formula FORMULA\_BASE\_TYPE' and shows the formula configuration:

**General**

Name: FORMULA_BASE_TYPE	Short Text: FORMULA_BASE_TYPE
Application: FMDM_MODEL_MM	Access Level: Application

**Detail**

Result Data Object: Base type

Base type = CONCATENATE ( SUBSTRING ( MATERIAL-Material , 3 , 5 ) , 0 )

Context pane shows 'ACTIONS' and 'MATERIAL' selected.

Formula Functions pane lists various functions like ABS, ARCCOS, ARCSIN, ARCTAN, CONCATENATE, etc., with their descriptions.

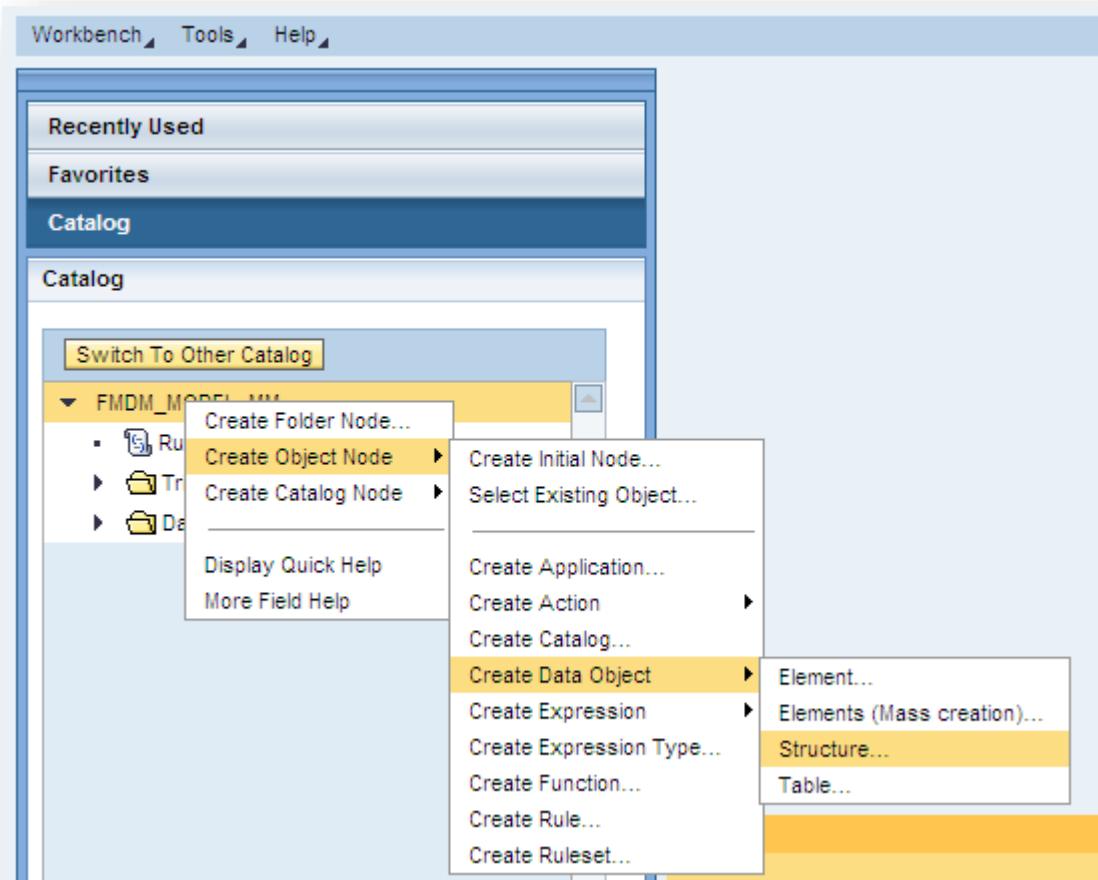
### 7.3.4 Example: Decision Table with multiple results

You want to return multiple results from one decision table. This requires that you pass back the results to the rule using a structure data object.

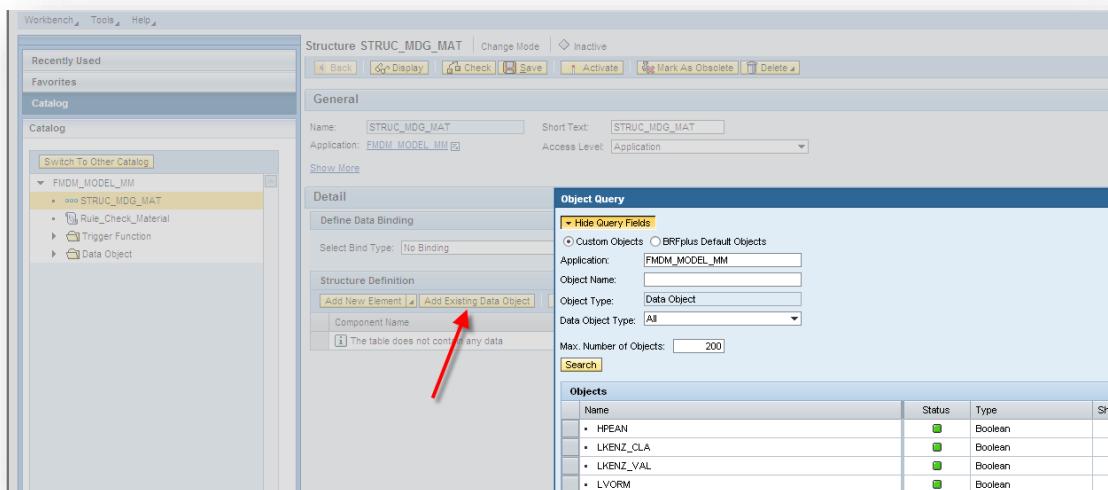
Here you find an example with BRF+ Decision Table for derivation with 2 output fields.

#### 7.3.4.1 Create a Structure Data Object

[http://help.sap.com/erp2005\\_ehp\\_05/helpdata/EN/00/da13e29b4e4fb4ace5990f212a05da/frameset.htm](http://help.sap.com/erp2005_ehp_05/helpdata/EN/00/da13e29b4e4fb4ace5990f212a05da/frameset.htm)

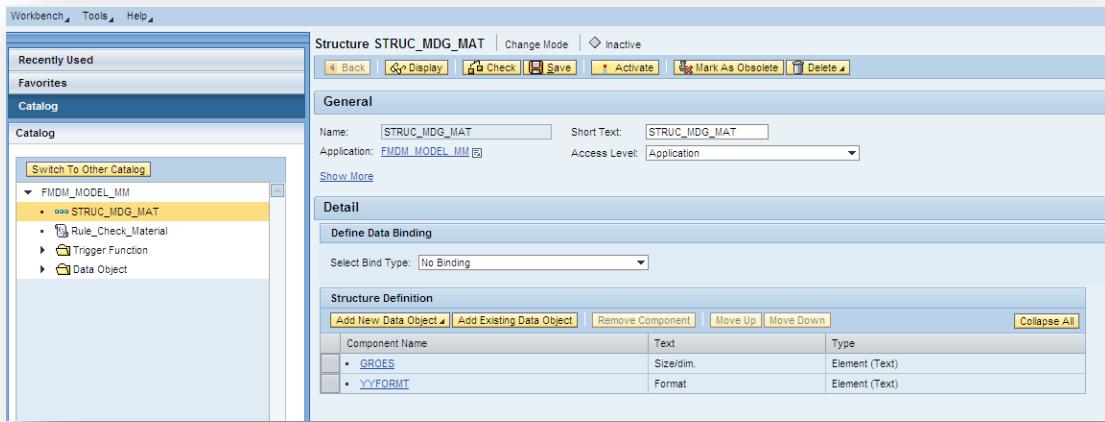


Add Elements to the created Structure. Under the *Detail* section, choose *Add Existing Data Object*.



## How-To: Maintain Check and Derivation Rules in MDG for Material

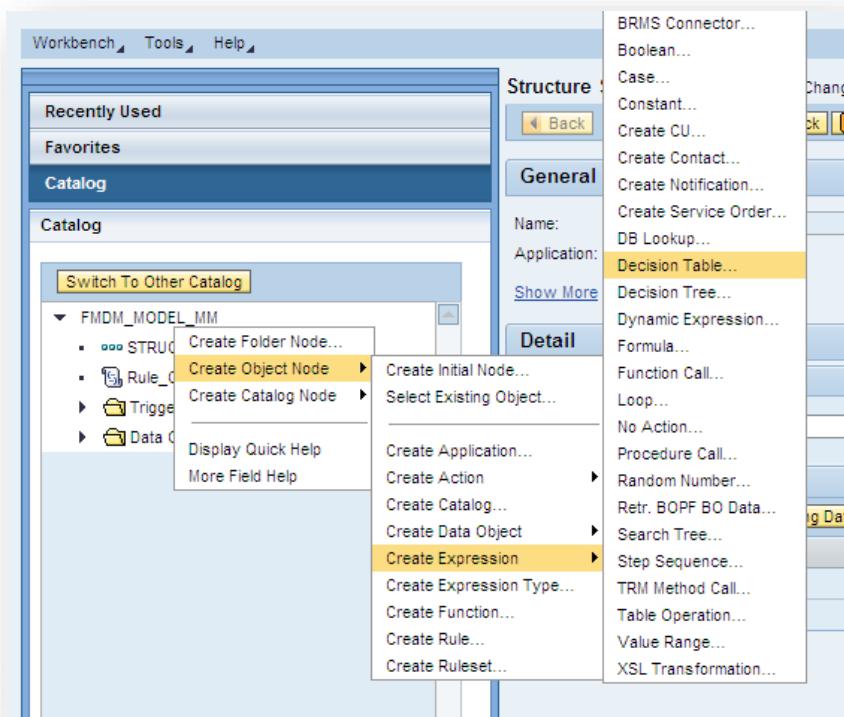
---



Choose *Activate* and save the data object.

### 7.3.4.2 Create a Decision Table

[http://help.sap.com/erp2005\\_ehp\\_05/helpdata/EN/00/da13e29b4e4fb4ace5990f212a05da/frameset.htm](http://help.sap.com/erp2005_ehp_05/helpdata/EN/00/da13e29b4e4fb4ace5990f212a05da/frameset.htm)

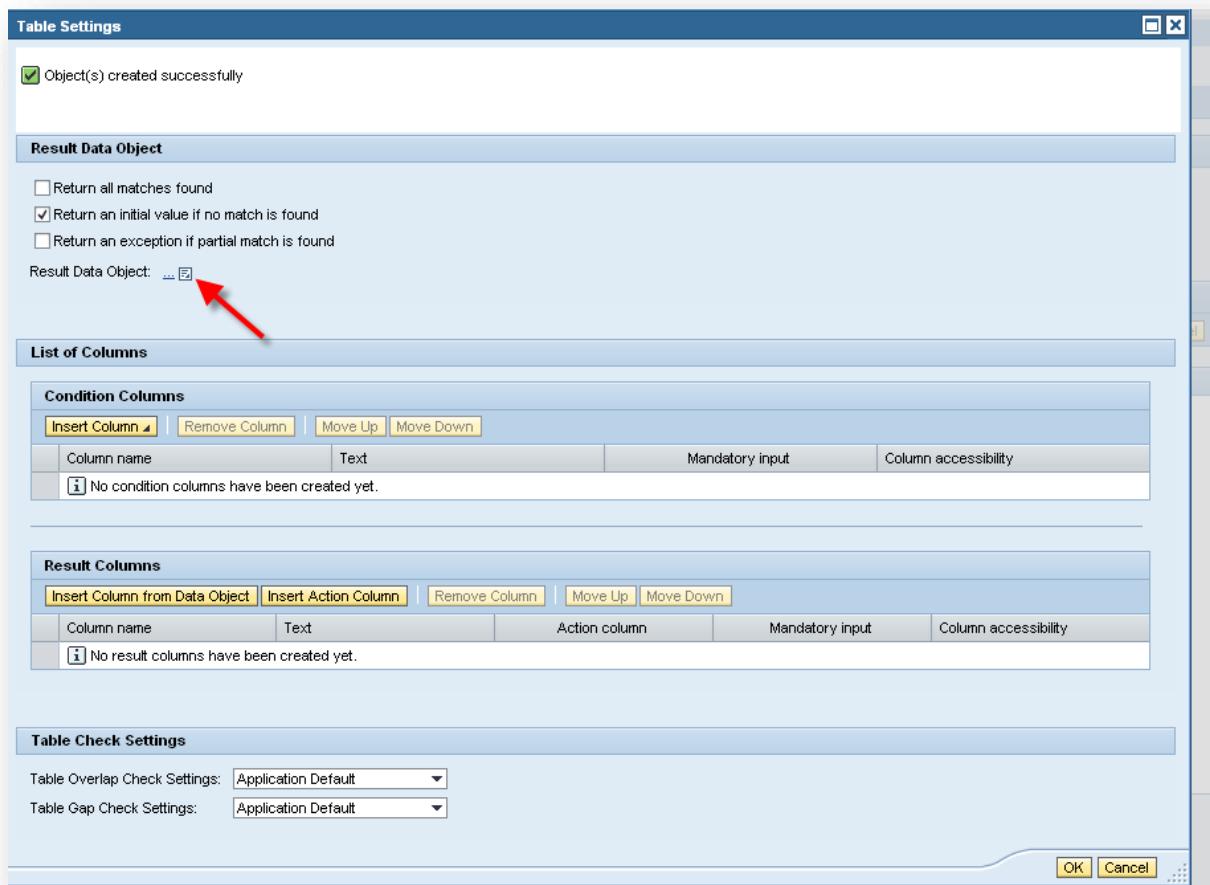


Add Result to the Decision Table

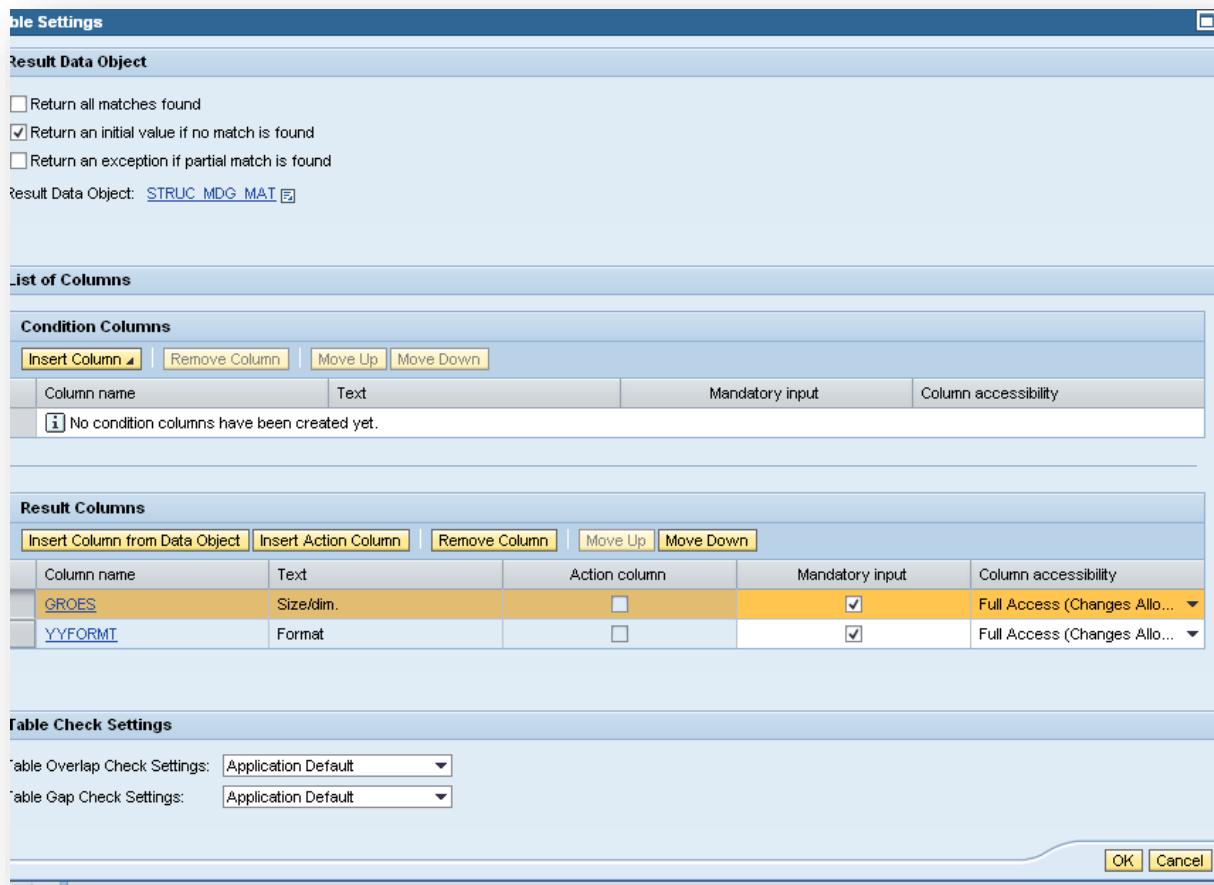
Choose next to Result Data Object.

## How-To: Maintain Check and Derivation Rules in MDG for Material

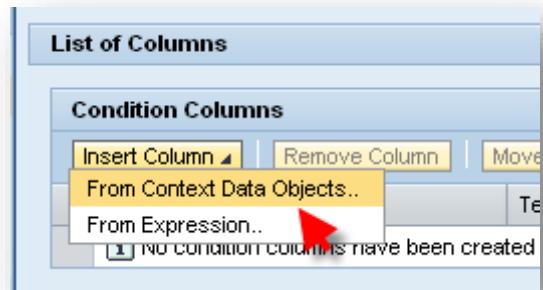
---



## How-To: Maintain Check and Derivation Rules in MDG for Material

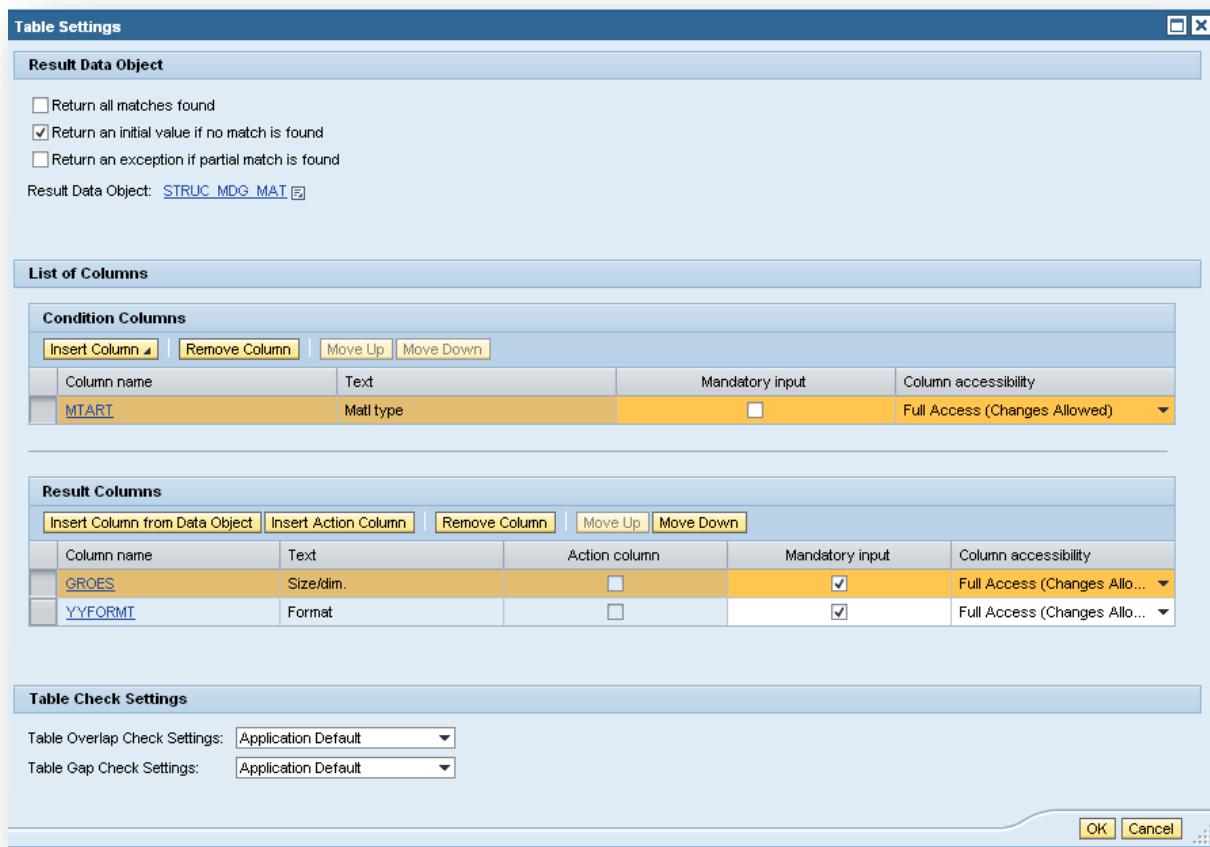


### Add Context to the Decision Table



## How-To: Maintain Check and Derivation Rules in MDG for Material

---

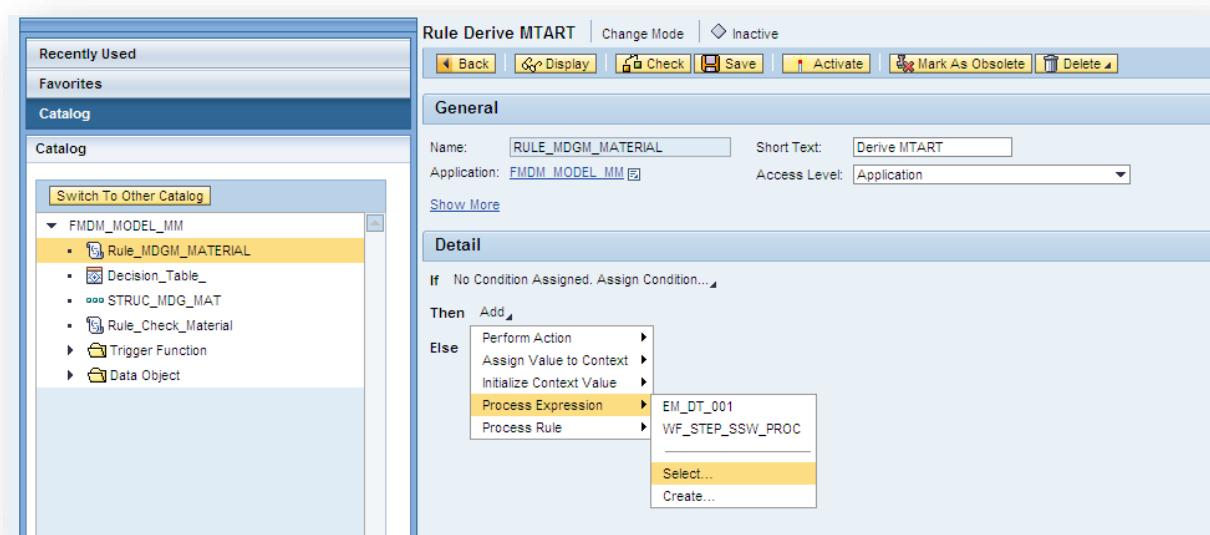
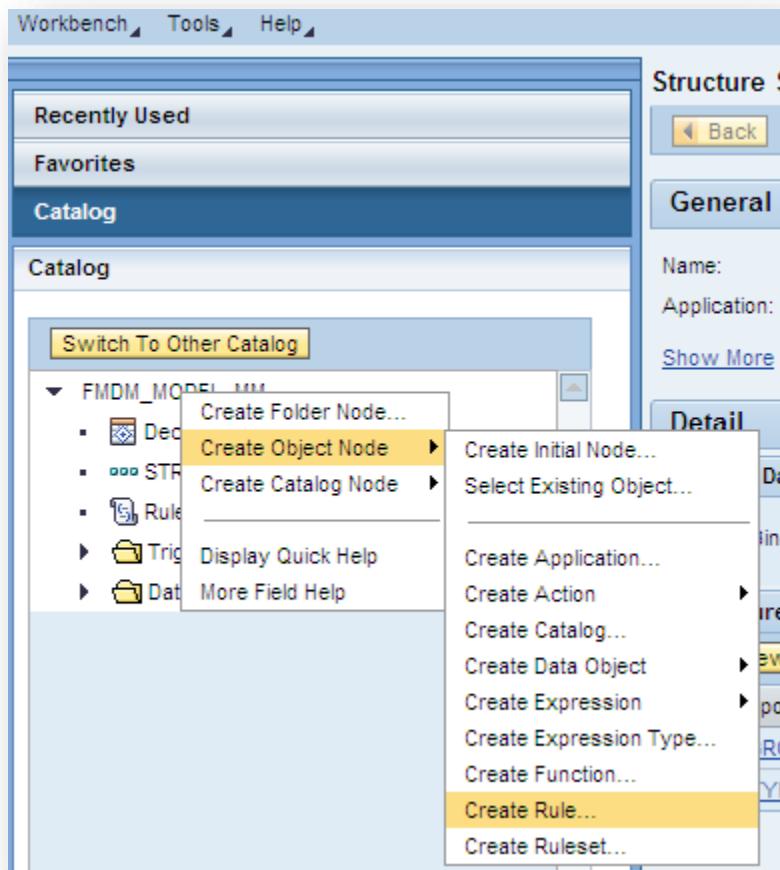


Fill table and activate.



### 7.3.4.3 Create a Rule

[http://help.sap.com/erp2005\\_ehp\\_05/helpdata/EN/00/da13e29b4e4fb4ace5990f212a05da/frameset.htm](http://help.sap.com/erp2005_ehp_05/helpdata/EN/00/da13e29b4e4fb4ace5990f212a05da/frameset.htm)

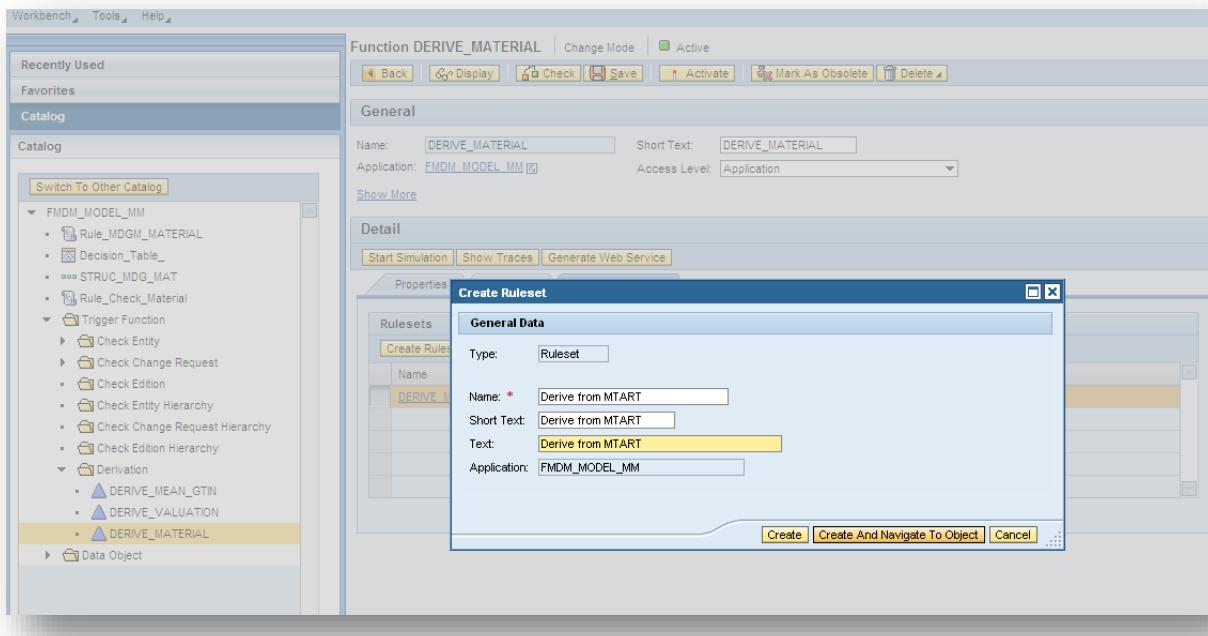


## How-To: Maintain Check and Derivation Rules in MDG for Material

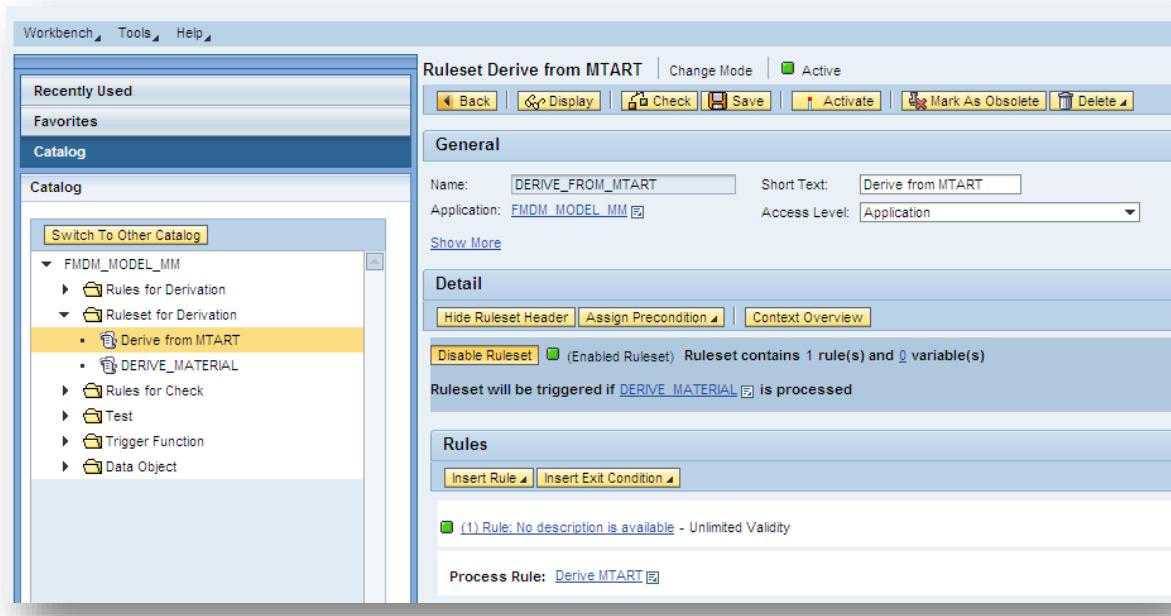
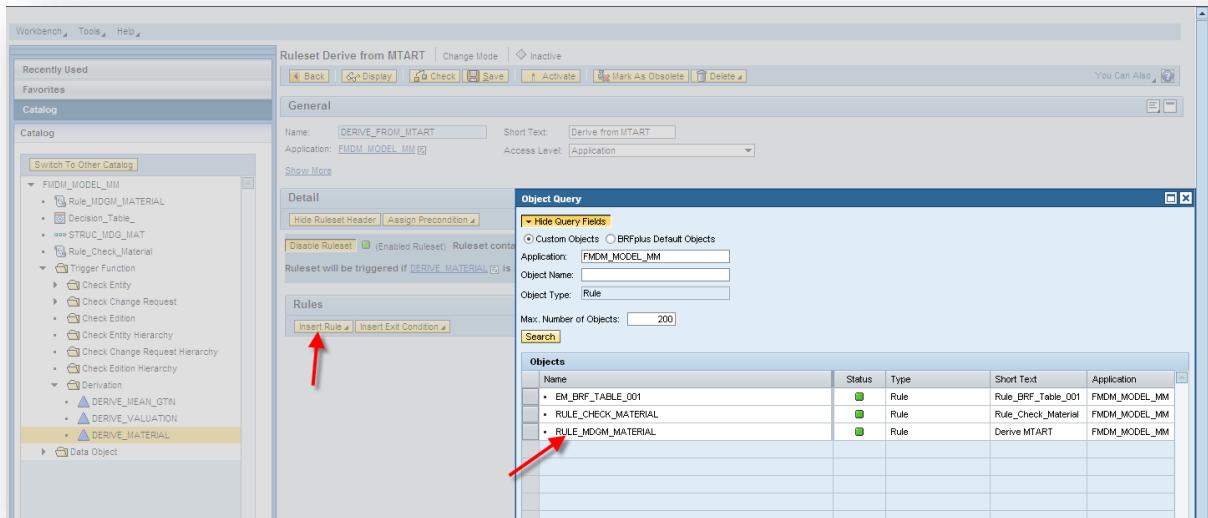


### 7.3.4.4 Create a Ruleset

[http://help.sap.com/erp2005\\_ehp\\_05/helpdata/EN/00/da13e29b4e4fb4ace5990f212a05da/frameset.htm](http://help.sap.com/erp2005_ehp_05/helpdata/EN/00/da13e29b4e4fb4ace5990f212a05da/frameset.htm)



## How-To: Maintain Check and Derivation Rules in MDG for Material



Now the system returns the multiple results Size/dim. and Format from the decision table depending on the maintained material type.

### 7.4 Example for DERIVE\_MARCMRPFC

#### 7.4.1 Default the Period Indicator to 'M'

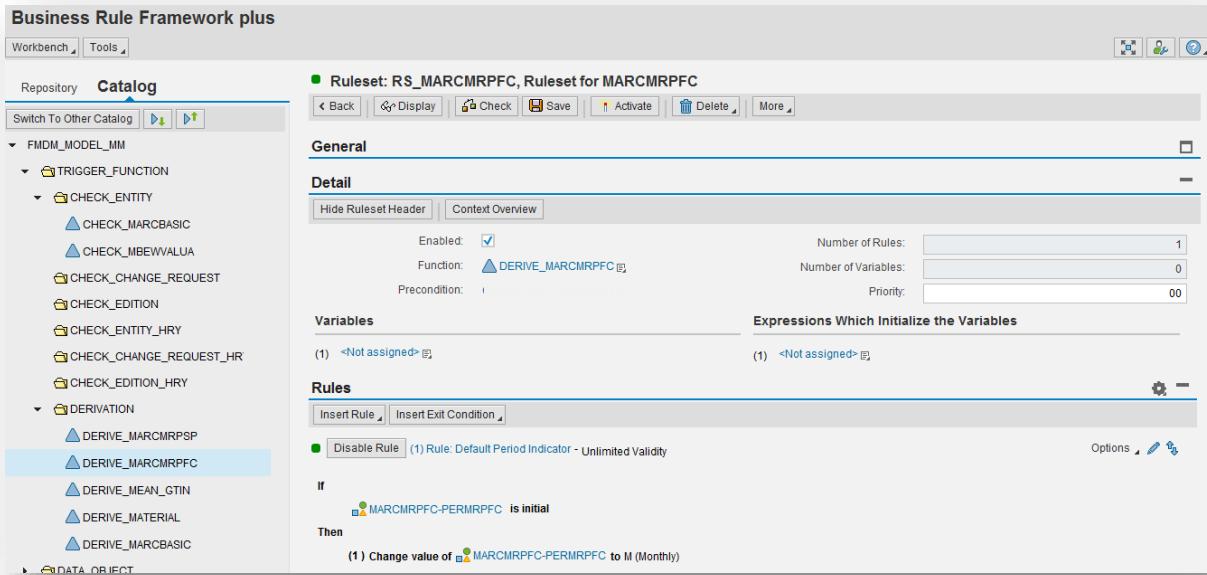
##### Scenario:

The Period Indicator (MARC-PERKZ) should be defaulted as in the backend (FM MARC\_ADD\_REFERENCE\_AFTER).

##### Solution:

Create a ruleset for function DERIVE\_MARCMRPFC.

Create a rule for if period indicator (MARCMRPFC-PERMRFPC) is initial set it to 'M'.



## 7.5 Example for DERIVE\_MARCMRPSP

### 7.5.1 Default the Procurement Type

#### Scenario:

The Procurement Type (MARC-BESKZ) should be defaulted as in the backend.

T134-BSEXT	T134-BSINT	MARC-BESKZ
0	0	blank
0	1 or 2	E
1 or 2	0	F
1 or 2	1 or 2	X

#### Solution:

Create a ruleset for function DERIVE\_MARCMRPSP.

Create rules for exit conditions and create the following rule for defaulting: if material type is not initial then change procurement type according to decision table.

#### Step by Step:

- Create ruleset
- Create variable for structure with BSEXT, BSINT, and MTART and expression to initialize the variable.

## How-To: Maintain Check and Derivation Rules in MDG for Material

**Structure: IS\_T134**

General

**Detail**

**Define Data Binding**

Binding Type: No Binding

**Structure Definition**

Add Existing Data Object	Add New Data Object	Remove	Up	Down	Up	Down
Component Name	Type					
BSEXT	External POs					
BSINT	Internal POs					
MTART	Material type					

**Database Lookup: READ\_T134**

General

**Detail**

Selection Mode: Data Retrieval

Select: Single Entry From: T134 Material Types

With Condition: MTART is equal to MATERIAL-MTART Change

Into: [IS\\_T134](#)

**Field Mapping**

Source Field	Text	Target Field	Text	Aggregation	Group By
BSEXT	External Purchase Orders Allowed	BSEXT	External POs	Select	
BSINT	Internal purchase orders allowed	BSINT	Internal POs	Select	
MTART	Material type	MTART	Material type	Select	

**Ruleset: RS\_MARCMRPSP, RS MARCMRPSP**

General

**Detail**

Hide Ruleset Header Context Overview

Enabled:

Function: [DERIVE\\_MARCMRPSP](#)

Precondition:

Number of Rules: 4

Number of Variables: 1

Priority: 00

Variables	Expressions Which Initialize the Variables
(1) <a href="#">IS_T134</a>	(1) Update variables after processing <a href="#">READ_T134</a>
(2) <Not assigned>	(2) <Not assigned>

- Create rules for exit conditions (material type is initial; procurement type not initial in staging; procurement type in MARC active area is 'blank') and a database lookup to check if MARC has the value 'blank' for the procurement type.

## How-To: Maintain Check and Derivation Rules in MDG for Material

---

**Rules**

Insert Rule | Insert Exit Condition

■ Disable Condition (1) Exit Condition: Material type is initial - Unlimited Validity

Exit ruleset when MATERIAL-MTART is initial Change Change Exit Condition is true

■ Disable Condition (2) Exit Condition: Procurement is not initial - Unlimited Validity

Exit ruleset when MARCMRPSP-BESKZ is not initial Change Change Exit Condition is true

■ Disable Condition (3) Exit Condition: Read MARC-BSKZ - Unlimited Validity

Exit ruleset when MARC\_SELECT is true

■ Database Lookup: MARC\_SELECT, Read MARC-BSKZ

Back | Display | Check | Save | Activate | Delete | More

**General**

**Detail**

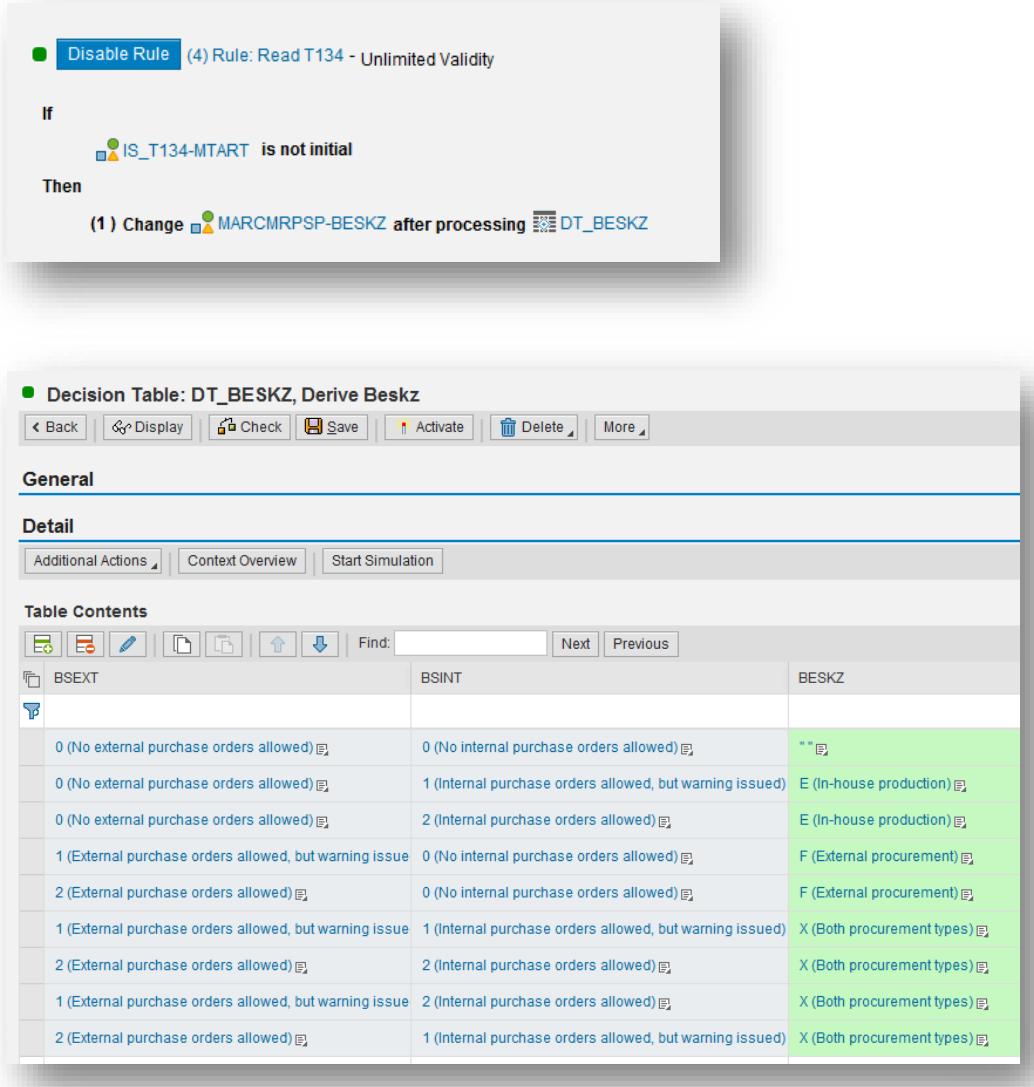
Selection Mode: Existence Check | Context Overview | Start Simulation

If at Least One Entry Exists In Table MARC Plant Data for Material

With Condition: BESKZ is initial Change  
and  
MATNR is equal to MARCMRPSP-MATERIAL Change  
and  
WERKS is equal to MARCMRPSP-WERKS Change

Then BOOLEAN Is True, Otherwise It is False.

- Create rule for defaulting with decision table



The screenshot shows two SAP Fiori screens. The top screen displays a rule configuration titled '(4) Rule: Read T134 - Unlimited Validity'. It includes an 'If' condition: 'IS\_T134-MTART is not initial' and an 'Then' action: '(1) Change MARCMRPSP-BESKZ after processing DT\_BESKZ'. The bottom screen shows the 'Decision Table: DT\_BESKZ, Derive Beskz' interface. It has tabs for 'General' and 'Detail', with 'Detail' selected. The 'Table Contents' grid has columns for BSEXT, BSINT, and BESKZ. The rows show various combinations of values for BSEXT and BSINT, leading to different BESKZ values: E (In-house production), F (External procurement), or X (Both procurement types). The BESKZ column is highlighted in green.

BSEXT	BSINT	BESKZ
0 (No external purchase orders allowed)	0 (No internal purchase orders allowed)	" "
0 (No external purchase orders allowed)	1 (Internal purchase orders allowed, but warning issued)	E (In-house production)
0 (No external purchase orders allowed)	2 (Internal purchase orders allowed)	E (In-house production)
1 (External purchase orders allowed, but warning issue)	0 (No internal purchase orders allowed)	F (External procurement)
2 (External purchase orders allowed)	0 (No internal purchase orders allowed)	F (External procurement)
1 (External purchase orders allowed, but warning issue)	1 (Internal purchase orders allowed, but warning issued)	X (Both procurement types)
2 (External purchase orders allowed)	2 (Internal purchase orders allowed)	X (Both procurement types)
1 (External purchase orders allowed, but warning issue)	2 (Internal purchase orders allowed)	X (Both procurement types)
2 (External purchase orders allowed)	1 (Internal purchase orders allowed, but warning issued)	X (Both procurement types)

## 7.6 Example: Using variables in Derivation Rules (like Workflow step)

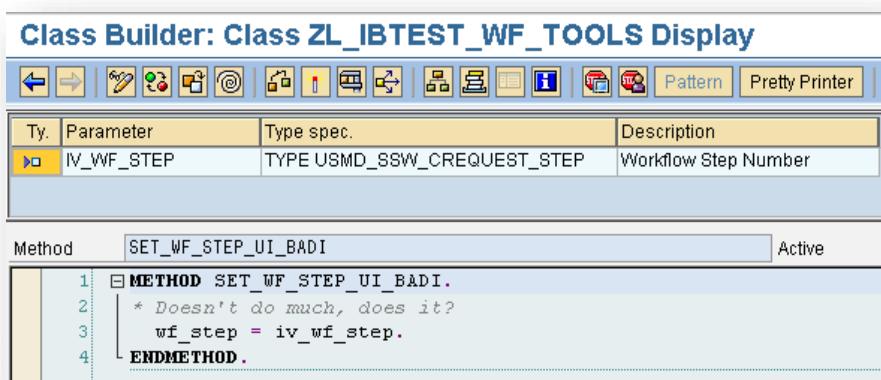
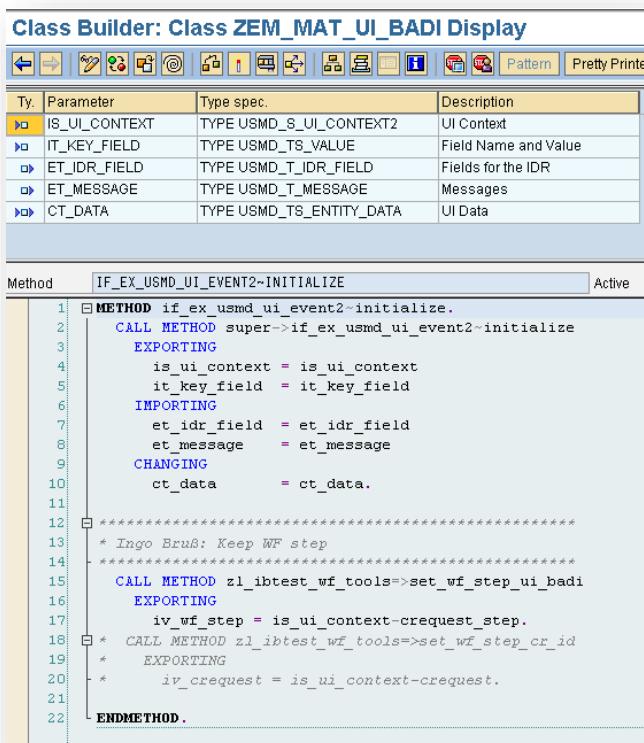
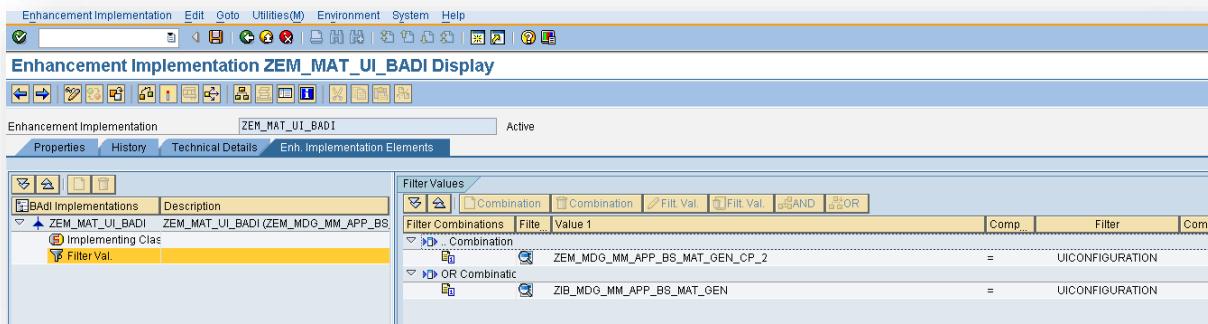
You want to derive values or messages dependent on a workflow step. The workflow step is not in the context of the BRF+ (until MDG6.1). But you can use variables, and you need to get the values. In the UI BAdI (EhP5) or enhanced METHOD process\_event (EhP6) the value is set and in the BRF rule the value is retrieved.

### 7.6.1 Set the values for the variables

#### 7.6.1.1 EhP5: Set the values (EhP5 UI BADI)

In the process, you should choose a Change Request Type that uses the UI configuration that is the filter value in the BADI.

## How-To: Maintain Check and Derivation Rules in MDG for Material



## How-To: Maintain Check and Derivation Rules in MDG for Material

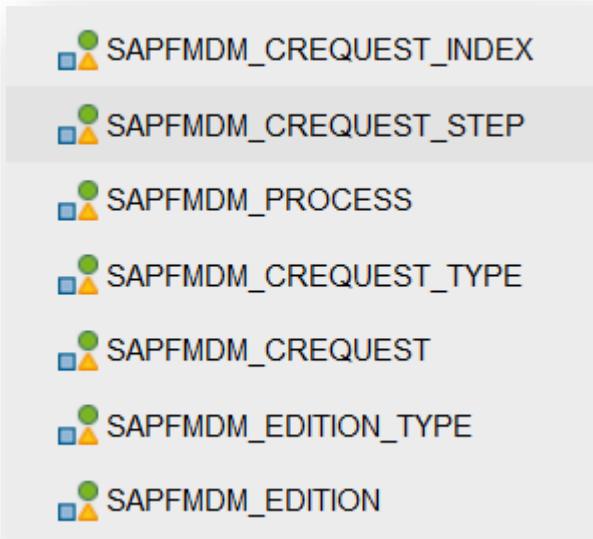
### 7.6.1.2 EhP6: Set the values for the variables

Read the WF container using CL\_USMD\_APP\_CONTEXT and use the step\_action to fill your variable into a BADI.

Enhance METHOD process\_event (CL\_MDG\_BS\_CR\_UIBB\_EVENTHANDLER). With IO\_EVENT->MV\_EVENT\_ID you will get the action and you can fill your variable.

### 7.6.1.3 MDG6.1: New variables

Now the workflow step is added in the context of the BRF+.



### 7.6.2 Create BRF+ Ruleset

The screenshot shows the SAP BRF+ Ruleset configuration interface. On the left, the catalog tree under 'FMDM\_MODEL\_MM' shows several nodes, with 'Derive\_material\_02' highlighted. The main panel displays the 'General' tab for the ruleset 'Derive\_material\_02'. The 'Detail' section shows the rule definition: 'If CR Step is equal to 90 Then Perform the following operations (1) Change value of MATERIAL-Size/dim. to Fixed Size (2) Perform Action Message\_Cr\_Step'. A green callout box highlights the message 'Ruleset contains 1 rule(s) and 1 variable(s)'.

## How-To: Maintain Check and Derivation Rules in MDG for Material

---

The screenshot displays two overlapping SAP GUI windows.

The top window is titled "Maintain Ruleset Variables". It shows a ruleset containing one variable, "(1) CR Step", which is initialized by the expression "WF STEP SSW PROC".

The bottom window is titled "Element CR Step". It shows the configuration for a step named "WF\_STEP\_SSW" under application "FMDM\_MODEL\_MM". The step is active and has a short text of "CR Step". The "Detail" tab is selected, showing the following properties:

- Define Data Binding:** Select Binding Type: "Bind to DDIC element (Data Dictionary)", DDIC Element: "USMD\_SSW\_CREQUEST\_STEP", Change Request Step: "Refresh Binding".
- Define Element Properties:** Element Type: "Text", Allowed Comparisons: "No Restriction".
- Element Attributes:** Length: "2".

## How-To: Maintain Check and Derivation Rules in MDG for Material

Procedure Call WF\_STEP\_SSW\_PROC | Change Mode | Active

Back | Display | Check | Save | Activate | Mark As Obsolete | Delete

**General**

Name: WF\_STEP\_SSW\_PROC Short Text: WF\_STEP\_SSW\_PROC  
Application: FMDM\_MODEL\_MM Access Level: Application

Show More

**Detail**

Result Data Object: CR Step  
Call Type: Static Method  
Class Name: ZL\_IBTEST\_WF\_TOOLS zl\_ibtest\_wf\_tools  
Use Interface Method:  
Interface Name:  
Method Name: GET\_WF\_STEP No Description

**Mapped Parameters**

1\_EV\_WF\_STEP - Optional, Exporting ( Change Request Step Used in Rule-Based Workflow ) Show Details

**Exception Handling**

No exceptions are defined in this sub-routine

Class Builder: Class ZL\_IBTEST\_WF\_TOOLS Display

Enhancement Info System  
MIME Repository  
Repository Browser  
Repository Information System  
Tag Browser  
Transport Organizer  
Test Repository

Class / Interface ZL\_IBTEST\_WF\_TOOLS

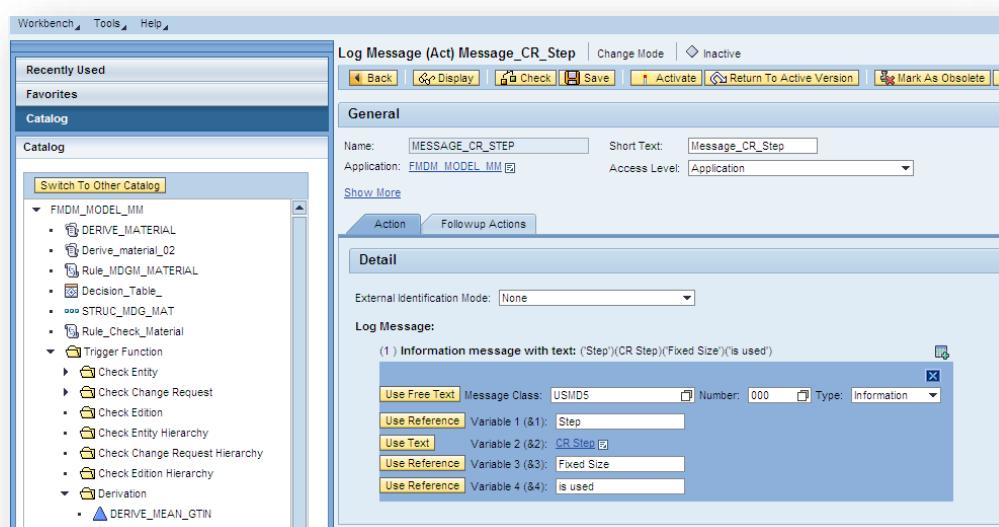
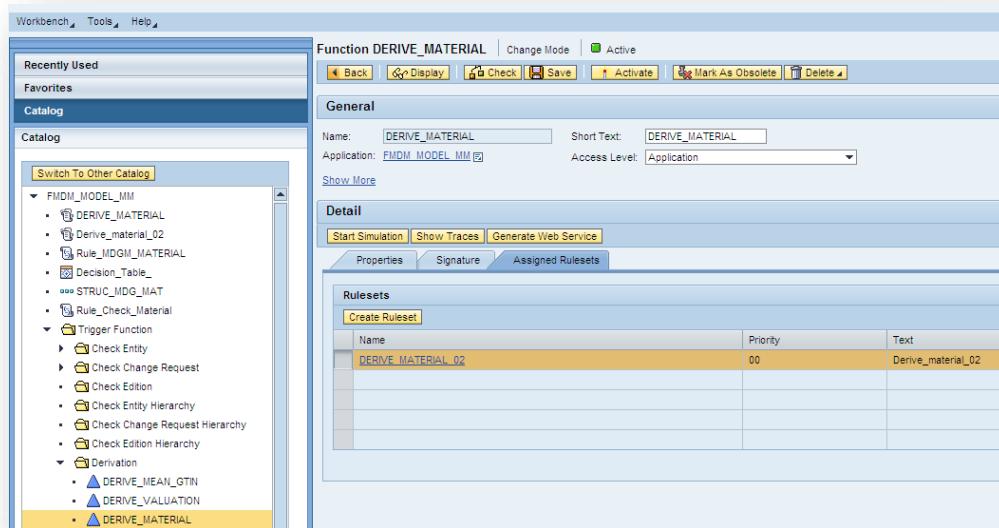
Object Name ZL\_IBTEST\_WF\_TOOLS Description zl\_ibtest\_wf\_tools  
WF\_STEP Workflow Step Number

Methods

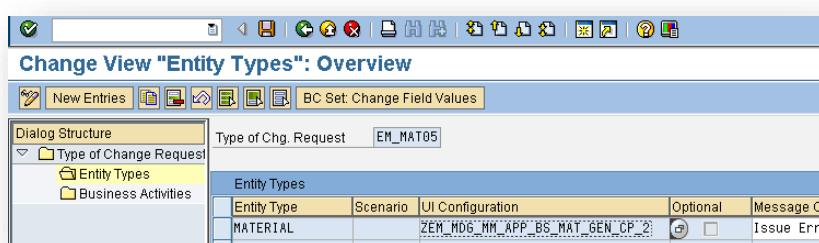
Method	Type	Description
GET_WF_STEP	method	method GET_WF_STEP. 1: VALUE(EV_WF_STEP) TYPE USMD_SSW_CREQUEST_STEP Change Request Step Used in Rule-Based Workflow

## How-To: Maintain Check and Derivation Rules in MDG for Material

### 7.6.3 Create BRF+ Function



### 7.6.4 EhP5 screenshots of the process



Create Material with CHANGE REQUEST-Type EM\_MAT05.

## How-To: Maintain Check and Derivation Rules in MDG for Material

---

**Create Change Request: Create Material**

Start Close

Entity  
Material: p-6400

Change Request  
Type: \* EM-Create Material with new UI

---

**Create Change Request: Material**

Entity Type: Material

Submit Close Check Save Draft Duplicate Check

Change Request

General Data Notes Attachments

Description: \* P-6400 Due Date:

Priority: Reason:

General Data Descriptions Classification Units of Measure EAN/UPC Basic Text Internal C

Material: \* P-6400

Material Description: P-6400

Material type: Semifinished Product Semifinished Product

Industry Sector: M Mechanical engineering

Base Unit of Measure: \* EA each Net weight: 1,000 Weight unit: KG Kilogr

Basic Data

Material Group: Ext. Material Group: Lab/Office: E1 Labor 1 for MDG

Product Hierarchy:

### 7.7 Example: BRF Validations/Derivations for staging only

#### 7.7.1 Background

Checks and derivations are called during change request processing and records are checked against these rules. In MDGIMG you can define on change request type and step level if the checks are executed and if the checks should be handled as errors or warnings only.

The rules are executed for all entities and organization units; independently which organization unit is in the staging of the corresponding change request.

Example: a material is in active area and opened for 2 plants (plant 1000 and 2000). A change request is started and only 1 plant (1000) is modified; hence only plant 1000 is in staging. If a user clicks on CHECK in the MDG UI, the system will execute the check for plant 1000 and 2000. This might lead into business process issues, if the user executing the CHECK is only responsible for plant 1000 and has no authorization to change/correct plant 2000.

This chapter explains a solution to be able to execute the BRF+ Checks only for the plant which is in the staging area.

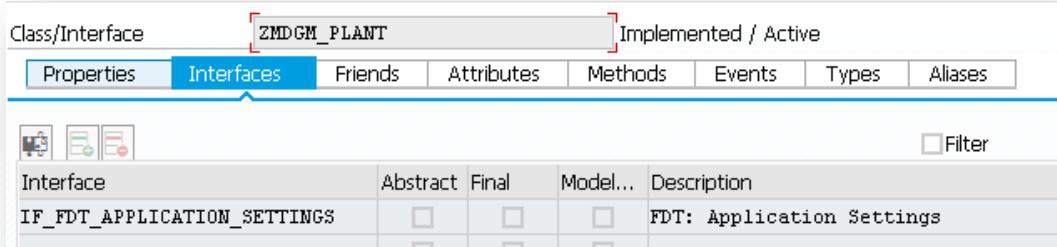
#### 7.7.2 Approach

The approach is to provide a solution without a modification. The solution will add an additional BRF+ rule or precondition to the BRF+ ruleset on the specific entity. The additional rule executes a BRF+ procedure call. This procedure call calls an ABAP class in the backend to retrieve from an ABAP coding the plant which is currently in the staging area of the corresponding change request. Afterwards the value will be saved into the context of the BRF function and can be used for filters in every other rule.

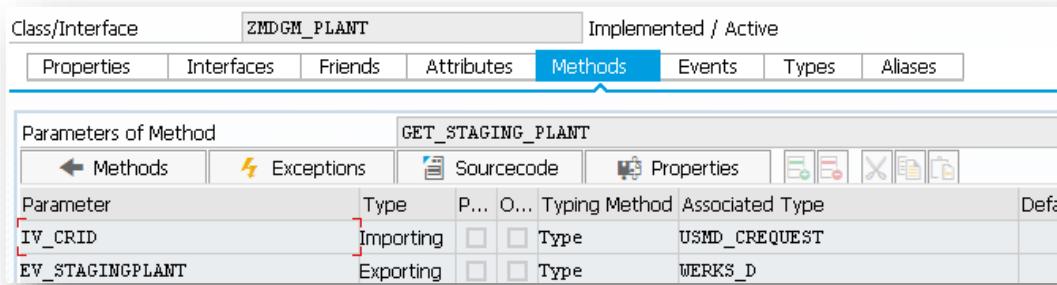
#### 7.7.3 Step by Step

##### 1. Implement ABAP class

- a. Implement an ABAP class in se80 which implements the interface IF\_FDT\_APPLICATION\_SETTINGS like a class ZMDGM\_PLANT.



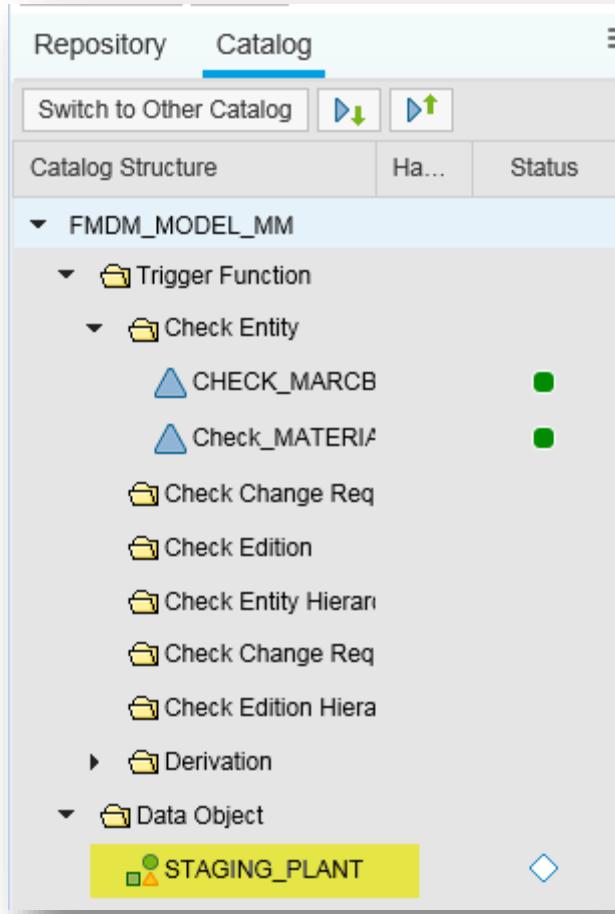
- b. Create a new method on class ZMDGM\_PLANT like "GET\_STAGING\_PLANT". The method needs at least two parameters:
  - i. Importing: CHANGE REQUEST ID
  - ii. Exporting: single plant



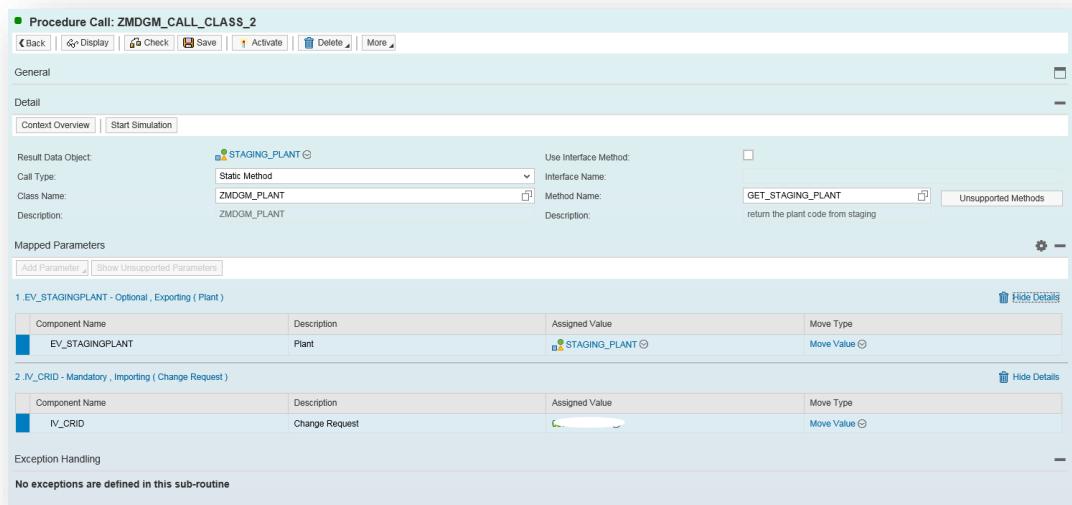
- c. Implement the details of the method. For this you can use MDG API and the method read\_char\_value. In this method, you should use the readmode if\_usmd\_model\_ext=>gc\_readmode\_no\_act which means that only staging area will be read.

## 2. BRF+

- a. Create Data object „STAGING PLANT“ of Type WERKS\_D and activate.



- b. Navigate to the Check (like CHECK: MARCBASIC) and to the corresponding ruleset.
- c. Add the Change Request ID to the context/signature.
- d. Add a new rule (position 1) to fill the context element STAGING\_AREA by calling the ABAP class you created earlier.



- e. Use in Rule (or Precondition) to check.

■ Ruleset: CHECK\_MARCBASIC

Back | Display | Check | Save | Activate | Delete | More

General

Detail

Show Ruleset Header | Context Overview

Rules

Insert Rule | Insert Exit Condition

■ Disable Rule (1) Rule: No description is available - Unlimited Validity

(1) Change STAGING\_PLANT after processing ZMDGM\_CALL\_CLASS\_2

■ Disable Rule (2) Rule: No description is available - Unlimited Validity

If MARCBASIC-Plant is equal to STAGING\_PLANT

Then

(1) Perform ZSULTEST

## 8. CROSS-ENTITY READ ACCESS FOR DERIVATION BRF+

With MDG7.0 it is possible to update the data of an entity (called a target entity) based on the data of another entity (called a source entity).

In addition, it is possible, by implementing a post-exit, to enforce the BRF+ derivation calls for a target entity, if only the data of the source entity is changed.

Note, that the BRF+ entity derivation is only called if data already exists for the target entity.

### 8.1 Example: Change the MRP Type Depending on the Plant-Specific Material Status

#### Scenario:

You want to create a BRF+ derivation to update the field *MRP Type (DISMM)* and field *MRP Group (DGRMRPPP)* of the target entity *MARCMRPPP* depending on the value of the field *Plant-Specific Material Status (MMSTA)* of the source entity *MARBASIC* according to the following logic:

```
IF      value(MMSTA) = '40' and value(DISMM) ='ND' and value(DGRMRPPP) = 'initial'  
THEN  set value(DISMM) = 'PD' and set value(DGRMRPPP) = '0001'  
ENDIF
```

#### Solution:

On the UI, the first field (MMSTA) appears in the assignment block Plant: General Data and the target fields (DISMM and DGRMRPPP) appear in the assignment block Plant: Material Requirements Planning.

#### Realization:

1: Create a BRF+ derivation with a ruleset containing the following rules:

- A rule to generate a table variable containing the key values of the source entity
- A rule containing a procedure call having as input parameter the table generated in the previous rule

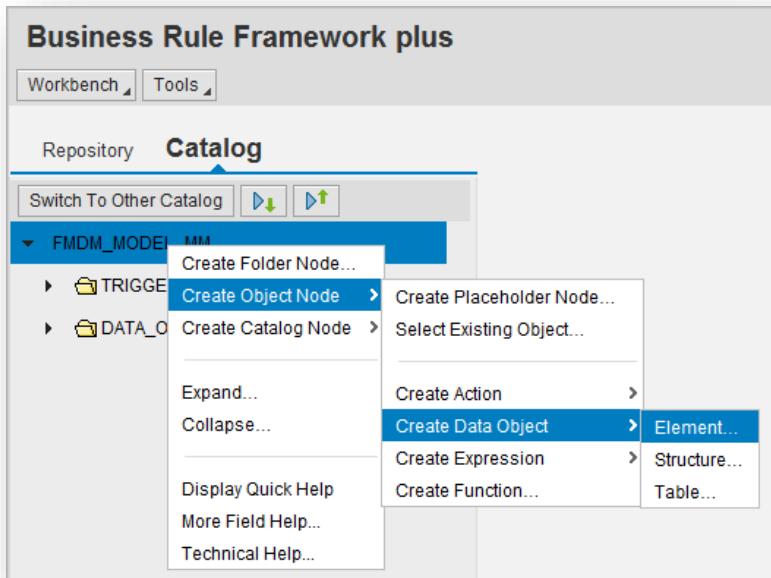
2: Enforce the BRF+ derivation if only data of the source entity will be changed and there is data for the target entity through post-exit.

#### 8.1.1 Prerequisite: Create Data Object

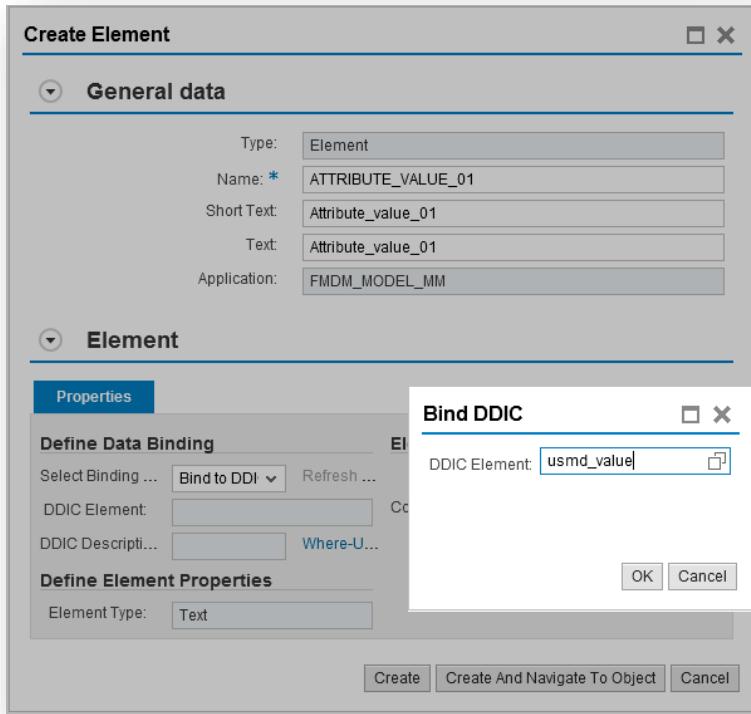
First following prerequisite needs to be done only once for data model MM.

Open the transaction USMD\_RULE for data model MM. Create a data object (call it *ATTRIBUTE\_VALUE\_01*) of type *Element* to be able to save an attribute value. Right click on the node

*DATA\_OBJECT* and navigate to the entry *Element* as shown in the next screen:

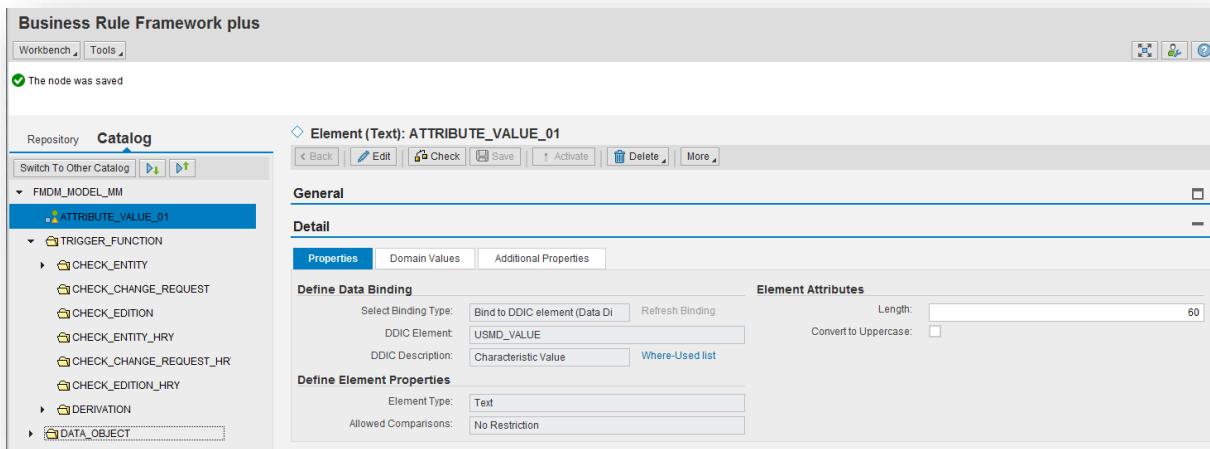


On the screen enter a *Name* of your element, for example *ATTRIBUTE\_VALUE\_01*, a *Short Text* and a *Text*. For the data binding select the entry *Bind to DDIC element (Data Dictionary)*. On the new screen enter the value *usmd\_value* in the field *DDIC Element*. Choose *OK*.



## How-To: Maintain Check and Derivation Rules in MDG for Material

Choose *Create* and accept your decisions on the screens.

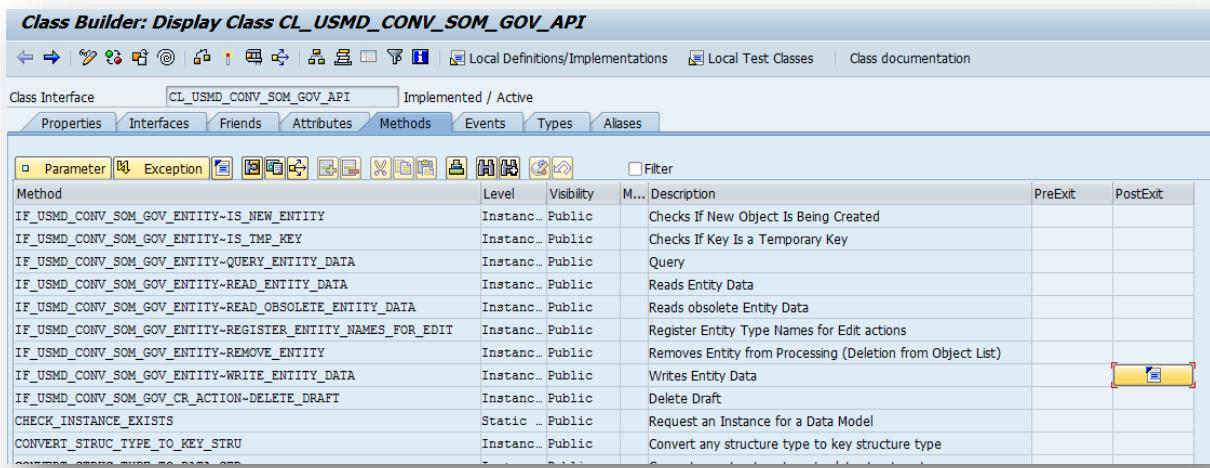


### 8.1.2 Prerequisite: Create Post-Exit

First following prerequisite needs to be done.

Create a post-exit for the method

*IF\_USMD\_CONV\_SOM\_GOV\_ENTITY~WRITE\_ENTITY\_DATA(CL\_USMD\_CONV\_SOM\_GOV\_API)*. The relevant coding depends on the rules. For this example, you can find the code in chapter 8.1.5, Add method to the Post-Exit.

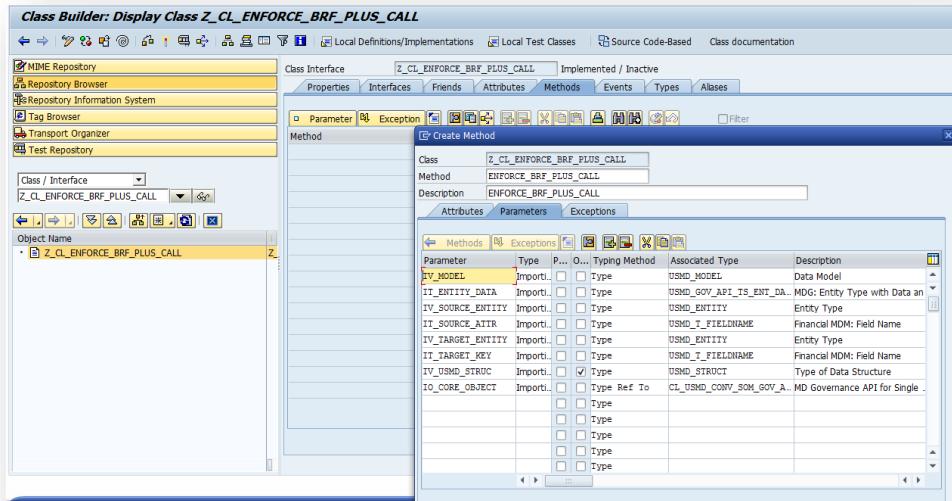


A quick guide on how to enhance methods can be found on the SDN. The following document follows this approach:

<http://wiki.sdn.sap.com/wiki/display/ABAP/Enhancement+Framework+-+Class+Enhancements+-+Pre-exit,+Post-exit+and+Overwrite-exit+methods+-+Concept+and+Simple+Scenarios>

### 8.1.3 Prerequisite: Create Z-Class

Create a Z-class, for example `Z_CL_ENFORCE_BRF_PLUS_CALL`, containing a static/public method, for example `ENFORCE_BRF_PLUS_CALL`.



The method needs to provide the following:

#### 1. Input parameters

Ty.	Parameter	Type spec.	Description
»	IV_MODEL	TYPE USMD_MODEL	Data Model
»	IT_ENTITY_DATA	TYPE USMD_GOV_API_TS_ENT_DATA	Data referring to the source entity
»	IV_SOURCE_ENTITY	TYPE USMD_ENTITY	Entity Type
»	IT_SOURCE_ATTR	TYPE USMD_T_FIELDNAME	Attributes of the source entity
»	IV_TARGET_ENTITY	TYPE USMD_ENTITY	Target entity, for which BRF+ rule will be called
»	IT_TARGET_KEY	TYPE USMD_T_FIELDNAME	Field Names of keys
»	IV_USMD_STRUCT	TYPE USMD_STRUCT OPTIONAL	Usmd structure type
»	IO_CORE_OBJECT	TYPE REF TO CL_USMD_CONV_SOM_GOV_API	MD Governance API for Single Object Maint. (Convenience)

#### 2. Method code

```
*****
* Method enforces the BRF+ rules for the target entity, if the value of * the
* source attribute is changed.
*****



data:
    ls_entity_data           type usmd_gov_api_s_ent_data,
    lt_entity_keys            type usmd_gov_api_ts_ent_tabl,
    ls_entity_keys            type usmd_gov_api_s_ent_tabl,
    lt_changes                type usmd_t_changed_entities,
    ls_changes                type usmd_s_changed_entities,
    lo_model                  type ref to if_usmd_model_ext,
    lo_so_gov_api             type ref to if_usmd_conv_som_gov_api,
    lv_model                  type usmd_model,
    ltr_data                  type ref to data,
```

```

lo_instance           type ref to if_usmd_model,
lo_app_context       type ref to if_usmd_app_context,
lv_edition          type usmd_edition,
lv_entity            type usmd_entity,
lv_entity_fieldname type usmd_fieldname,
lv_fieldname         type usmd_fieldname,
lt_entity_fieldname type usmd_ts_entity_fieldname,
ls_entity_fieldname type usmd_s_entity_fieldname,
lt_data              type usmd_gov_api_ts_ent_data,
ls_data              type usmd_gov_api_s_ent_data,
lv_enforce           type boole_d,
lt_attr              type usmd_ts_field,
ls_attr              type usmd_s_field,
lv_attr              type usmd_attribute,
lt_sel               type usmd_ts_sel,
ls_sel               type usmd_s_sel,
lv_lines             type int4,
lv_usmd_value        type usmd_value,
lv_usmd_struct       type usmd_struct,
lt_message           type usmd_t_message.

field-symbols:
<lt_entity_data>      type index table,
<ls_entity_data>      type any,
<ls_changed_entity>   type usmd_s_changed_entity,
<ls_key_value>        type usmd_s_attr_value,
<lt_data>              type sorted table,
<ls_data>              type any,
<lv_value>            type any.

clear: lv_edition.
if iv_source_entity is initial or it_source_attr is initial or
  iv_target_entity is initial.
  return.
endif.
"-----"
"                               Preparation
"-----"
" Determine the application context and the change request of the
" context.
lo_app_context = cl_usmd_app_context->get_context( ).
"Get model-instance and the edition
if lo_app_context is bound.
  lo_app_context->get_attributes(
    importing
      ev_edition      = lv_edition
      eo_model        = lo_model ).
  assert lo_model is bound.
  lv_model = lo_model->if_usmd_model_metadata_ext~d_usmd_model.
  assert lv_model is not initial.
endif.

```

```

" Belong the BRF_plus-rule to the right model? If not return.
if iv_model <> lv_model.
    return.
endif.

" Get a suitable modelinstance to get the field name of the entity
" (the previous calculated model-instance cannot be needed)
cl_usmd_model=>get_instance(
    exporting
        i_usmd_model = lv_model
    importing
        eo_instance = lo_instance
        et_message   = lt_message .
assert lo_instance is not initial.

"-----
"                               Checks
"-----


"Get fieldname of the source entity
lv_entity_fieldname = cl_usmd_services=>entity2fieldname(
    io_model      = lo_instance
    i_entity       = iv_source_entity .
lv_entity = lv_entity_fieldname.

" Do entity data refer to the source entity?
loop at it_entity_data into ls_entity_data where entity = lv_entity.
    clear ls_entity_keys.
    ls_entity_keys-entity = lv_entity.
    ls_entity_keys-tabl = ls_entity_data-entity_data.
    insert ls_entity_keys into table lt_entity_keys.
endloop.

" If the entity data do not refer to the source entity => don't
" enforce a rule
if lt_entity_keys is initial.
    return.
endif.

" Check whether an attribute from the given list IT_SOURCE_ATTR
" has been changed: If yes then enforce the BRF_plus rule for the
" target entity, otherwise return...

" Get changed attributes of the source entity
call method io_core_object->get_entity_field_changes
    exporting
        iv_struct          = 'KATTR'
        it_entity_keys     = lt_entity_keys
        iv_contained_changes = abap_false
    receiving
        rt_changes         = lt_changes.

```

```

" Check
loop at lt_changes into ls_changes where entity_type = lv_entity.
  if lv_enforce = abap_true.
    exit.
  else.
    loop at it_source_attr into lv_fieldname.
      if lv_enforce = abap_true.
        exit.
      else.
        loop at ls_changes-changed_entities assigning <ls_changed_entity>.
          read table <ls_changed_entity>-
changed_attributes with key fieldname = lv_fieldname transporting no fields.
          if sy-subrc = 0.
            lv_enforce = abap_true.
            exit.
          endif.
        endloop.
      endif.
    endloop.
  endif.
endloop.

" No attribute from the given list has been changed=> return
if lv_enforce = abap_false.
  return.
endif.

"-----
"                         Enforce rule
"-----


" Get fieldname of the target entity
clear lv_entity_fieldname.
lv_entity_fieldname = cl_usmd_services->entity2fieldname(
  io_model      = lo_instance
  i_entity      = iv_target_entity ).
if lv_entity_fieldname is initial.
  return.
endif.

" Get reference for target entity data
if iv_usmd_struct is supplied.
  lv_usmd_struct = iv_usmd_struct.
else.
  lv_usmd_struct = if_usmd_model_ext->gc_struct_key_attr.
endif.
lo_model->create_data_reference(
  exporting
    i_fieldname  = lv_entity_fieldname
    i_struct     = lv_usmd_struct
    if_table     = abap_true

```

```

        i_tabtype      = if_usmd_model_ext->gc_tabtype_sorted
        importing
            er_data      = ltr_data .
        assert ltr_data is bound.
        assign ltr_data->* to <lt_data>.

        "Selection options
        clear lt_sel.
        loop at it_entity_data into ls_entity_data where entity = lv_entity.
            assign ls_entity_data-entity_data->* to <lt_entity_data>.
            loop at <lt_entity_data> assigning <ls_entity_data>.
                loop at it_target_key into lv_fieldname.
                    assign component lv_fieldname of structure <ls_entity_data>
                    to <lv_value>.
                    if sy-subrc = 0.
                        clear ls_sel.
                        ls_sel-sign      = 'I'.
                        ls_sel-option    = 'EQ'.
                        ls_sel-fieldname = lv_fieldname.
                        ls_sel-low       = <lv_value>.
                        insert ls_sel into table lt_sel.
                    endif.
                endloop.
            endloop.
        endloop.

        " Add edition into it_sel (it is not necessary for material/BP)
        if lv_edition is not initial.
            clear ls_sel.
            ls_sel-sign      = 'I'.
            ls_sel-option    = 'EQ'.
            ls_sel-fieldname = usmd0_cs_fld-edition.
            ls_sel-low       = lv_edition.
            insert ls_sel into table lt_sel.
        endif.

        " Read target entity data
        lo_model->read_char_value(
            exporting
                i_fieldname = lv_entity_fieldname
                it_sel      = lt_sel
            importing
                et_data     = <lt_data> .

            if <lt_data> is not initial.
                ls_data-entity = lv_entity_fieldname.
                ls_data-entity_data = ltr_data.
                insert ls_data into table lt_data.

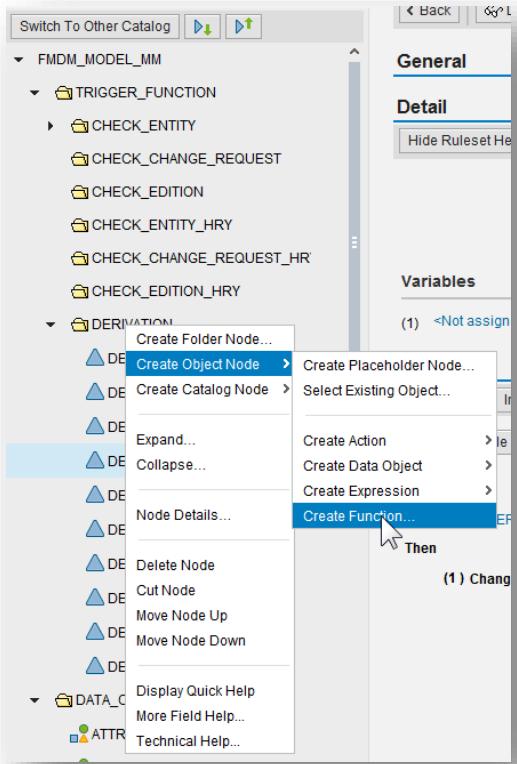
        " Enforce call of the BRF+ rule for the target entity...a write
        " is enough!
        call method io_core_object->write_entity_data
            exporting

```

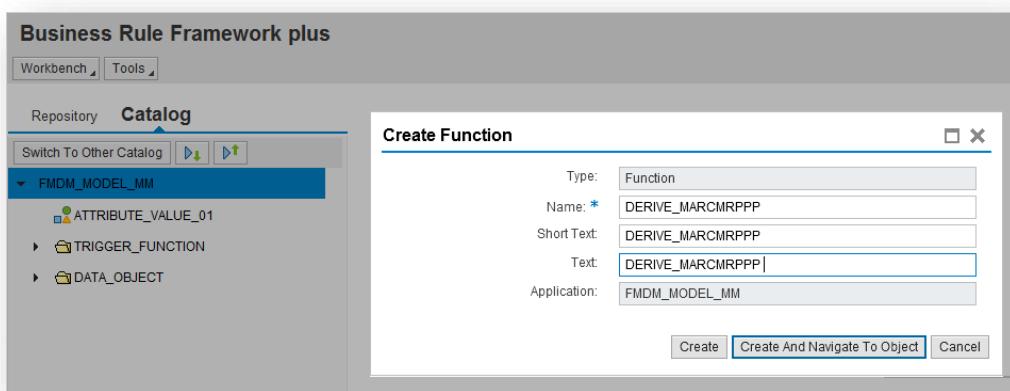
```
it_entity_data = lt_data.  
endif.
```

### 8.1.4 Create Derivation for MARCMRPPP

Open transaction USMD\_RULE and create a derivation with the name *DERIVE\_MARCMRPPP*. Right click on the node *DERIVATION* and navigate to *Create Function* as shown below.



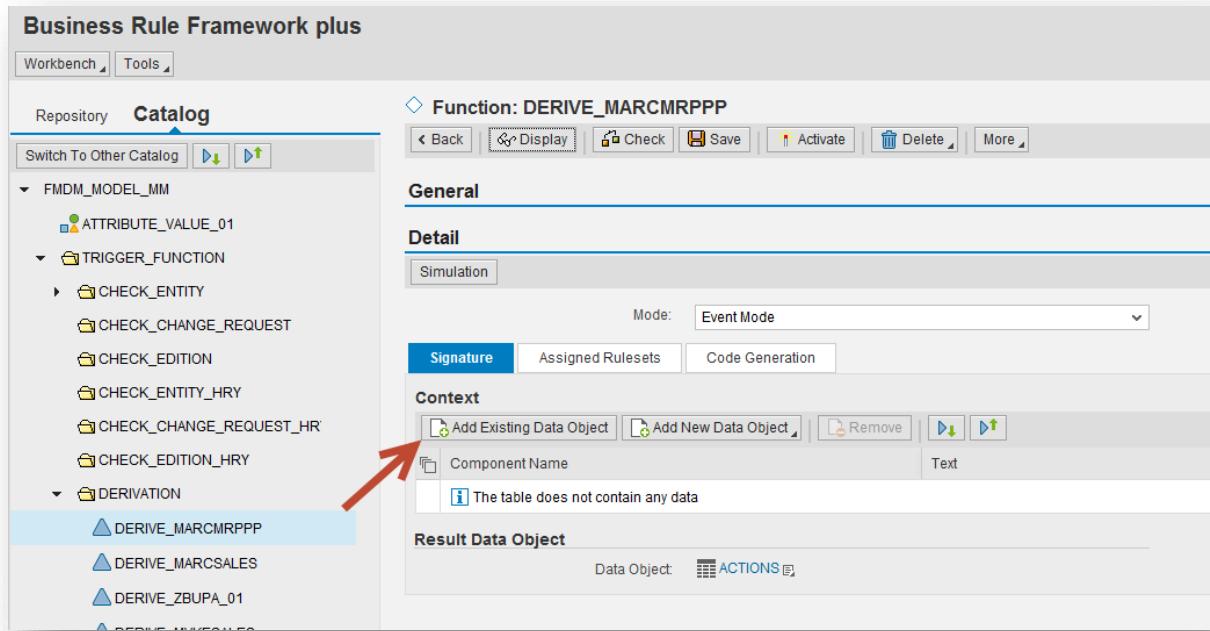
Enter the name *DERIVE\_MARCMRPPP*, a *Short Text*, and a *Text*.



## How-To: Maintain Check and Derivation Rules in MDG for Material

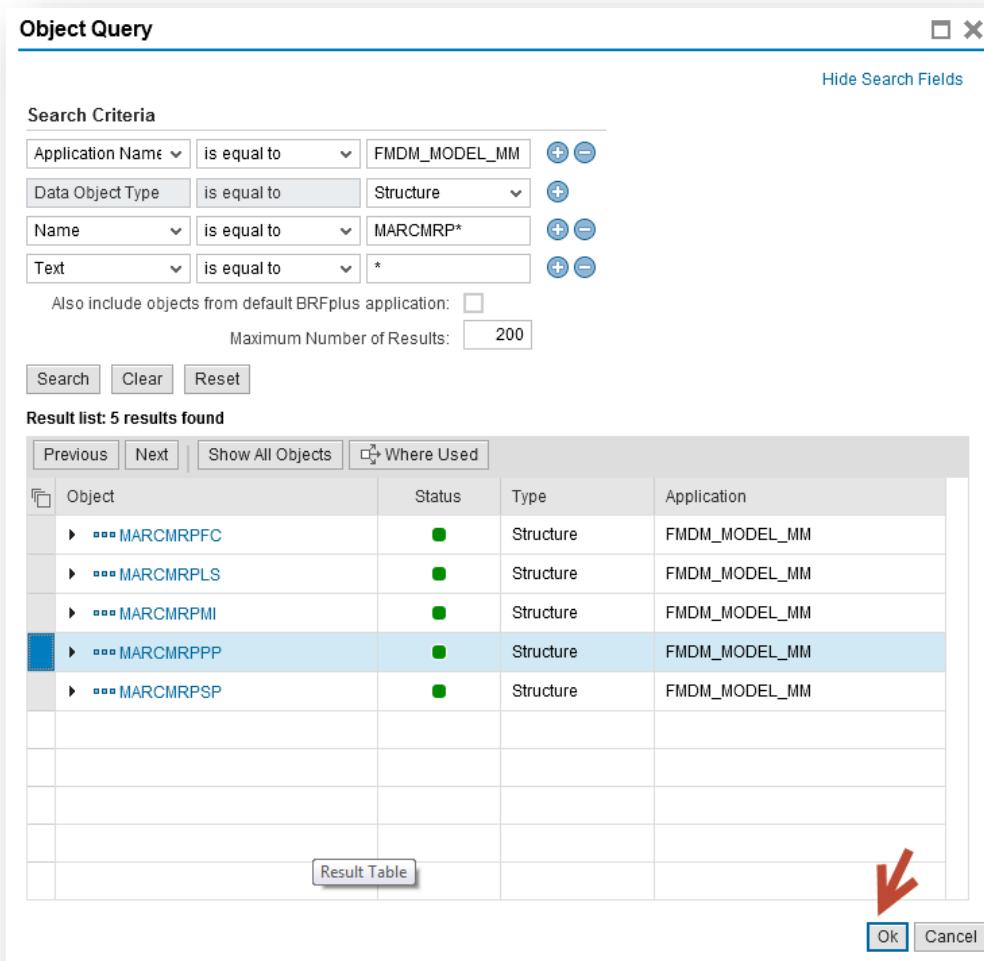
---

Choose *Create* and accept your decision to get the following screen. Click on *Edit* and *Add Existing Data Object*.

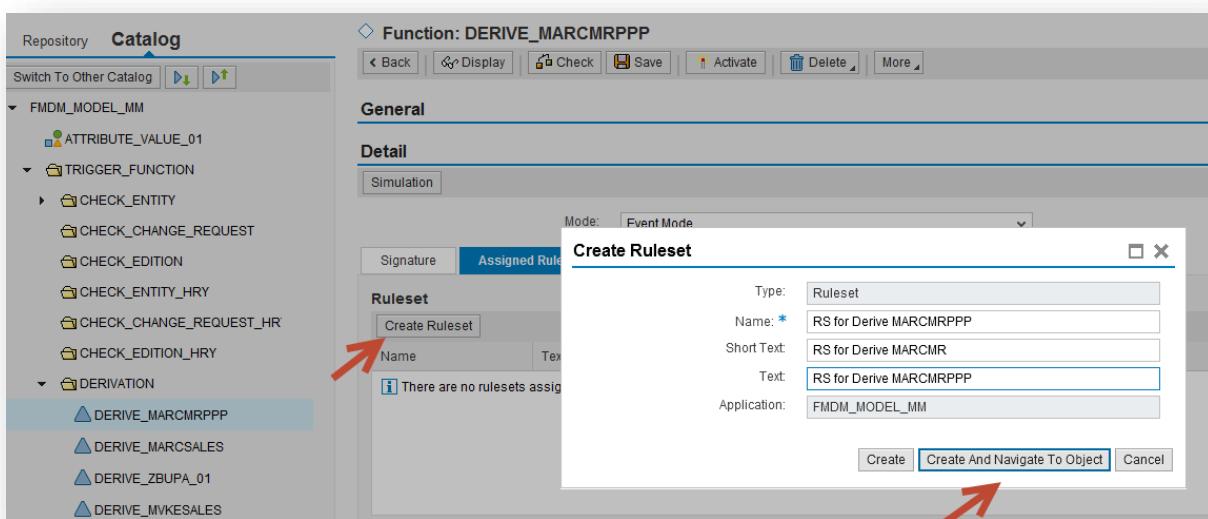


## How-To: Maintain Check and Derivation Rules in MDG for Material

Search for structure MARCMRPPP and choose OK.



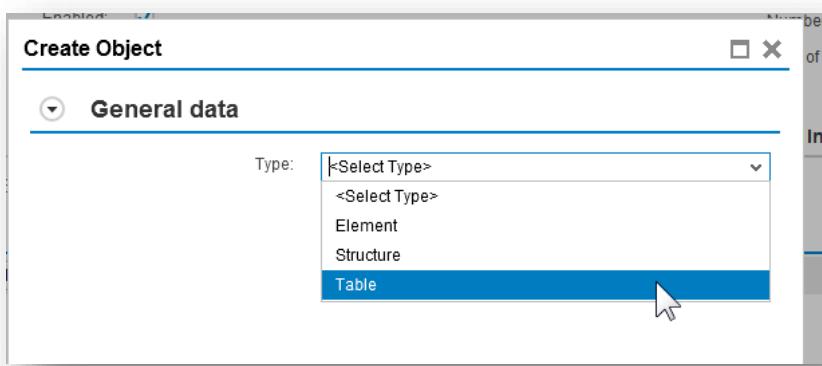
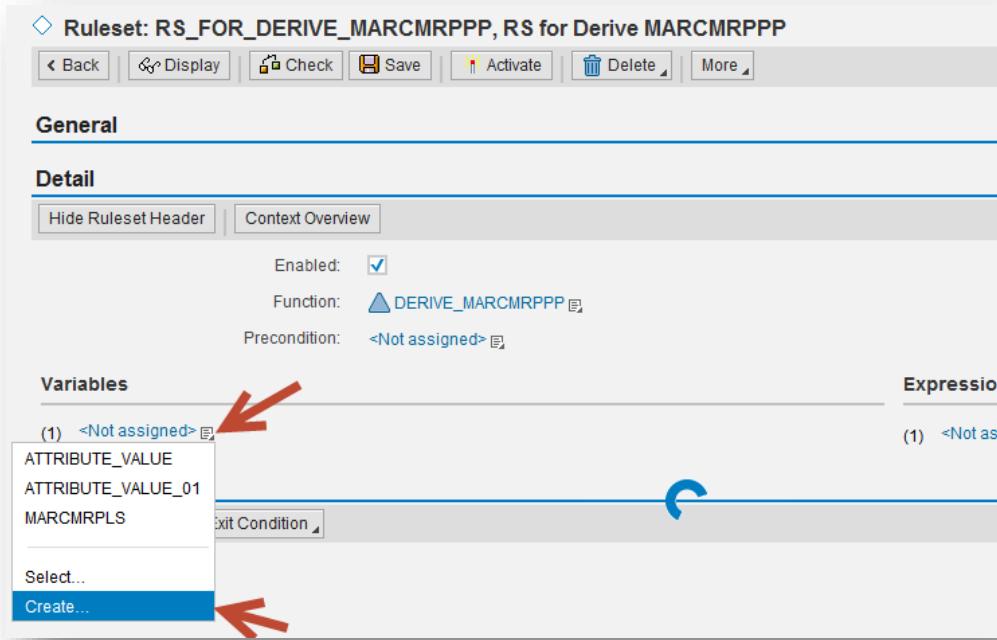
Create the ruleset: Choose *Assigned Rulesets* and then choose *Create Ruleset* as shown on the following screen.



Enter a *Name*, a *Short Text*, and a *Text*. Choose *Create And Navigate To Object*.

## How-To: Maintain Check and Derivation Rules in MDG for Material

You need a table containing the key-values of the source entity so create a variable, *TABLE\_OF\_KEY\_VALUES*, of type *Table* by choosing the icon to the right of the field <Not assigned> and navigate to *Create* as shown below.

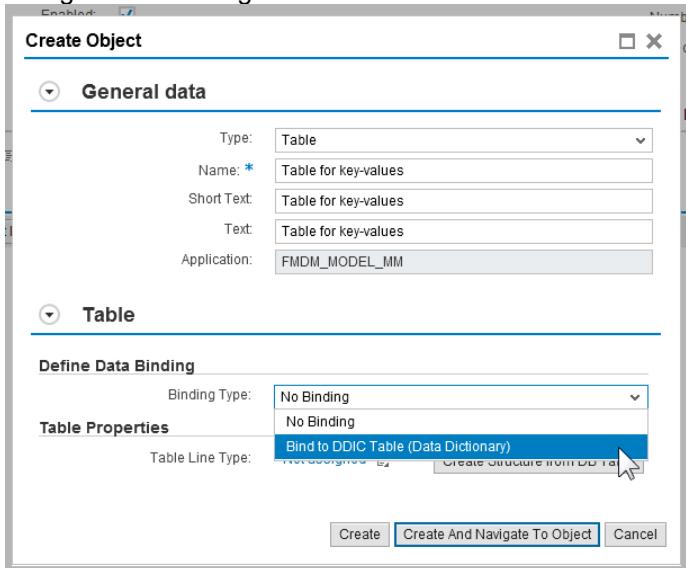


On the screen select the *Type Table*:

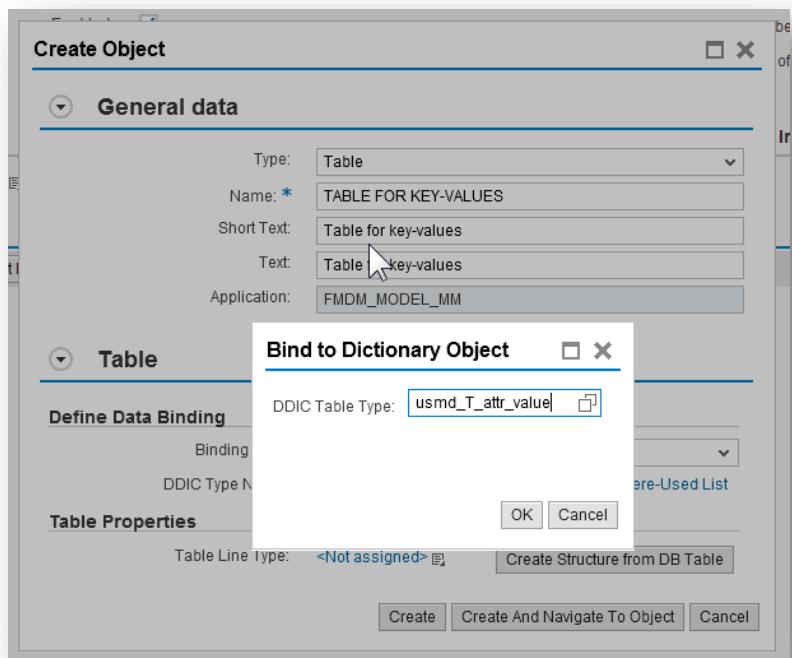
## How-To: Maintain Check and Derivation Rules in MDG for Material

---

You get the following screen:



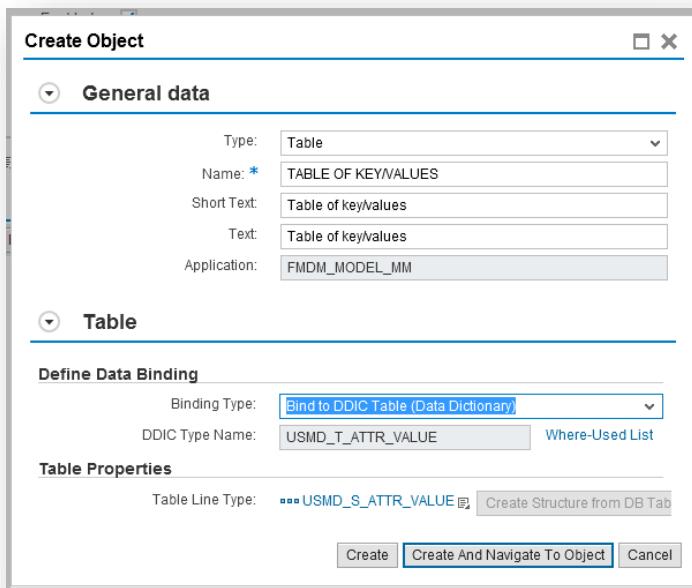
Enter a *Name*, for example Table of keys/values, a *Short Text*, and a *Text*. Enter in the field *Binding Type* the value *Binding to DDIC Table (Data Dictionary)* to get the following screen.



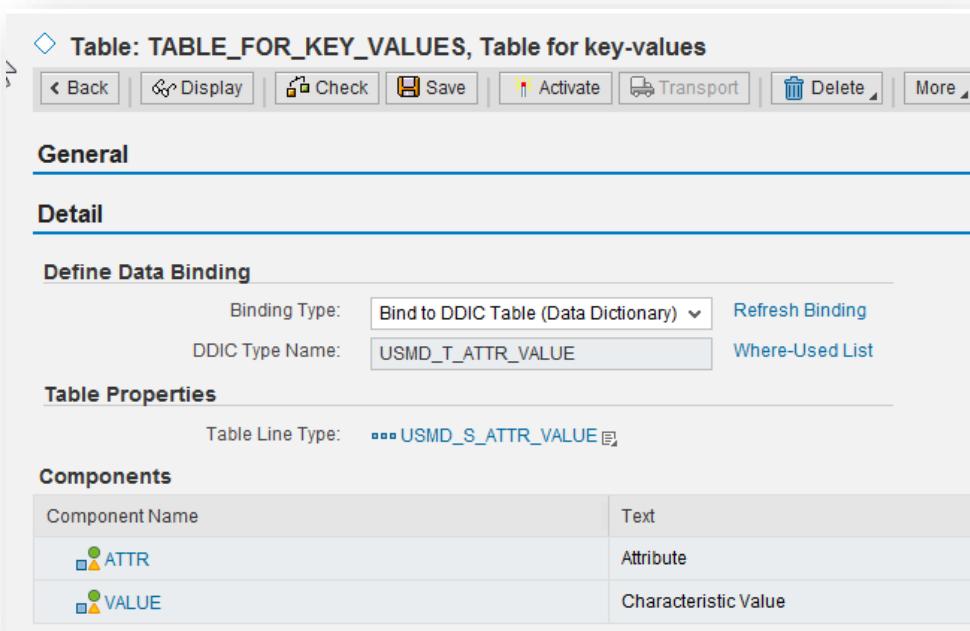
## How-To: Maintain Check and Derivation Rules in MDG for Material

---

Maintain *USMD\_T\_ATTR\_VALUE* in the field *DDIC Table Type* and choose *OK* to get the following screen:

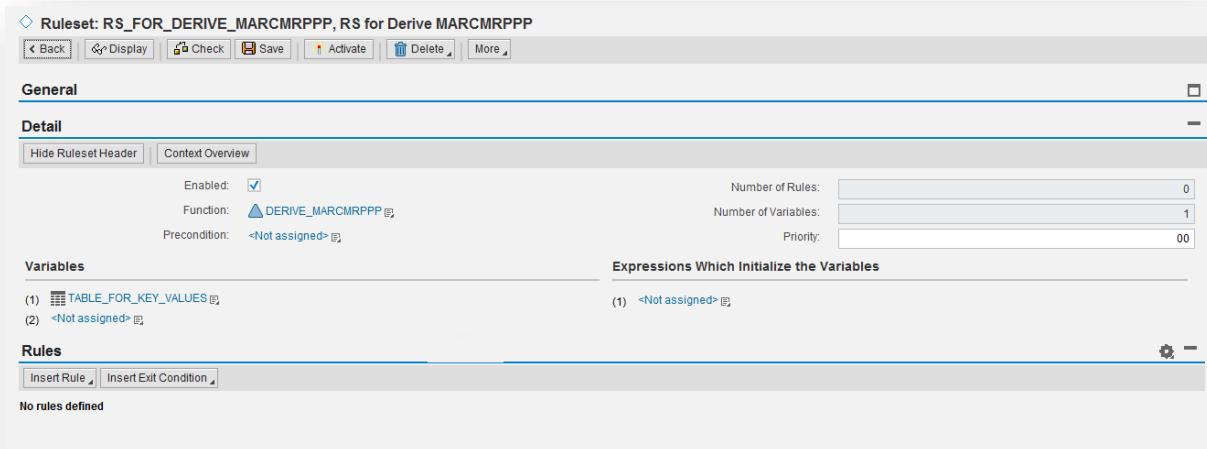


Choose *Create And Navigate To Object*. Choose *OK* to get the following screen:

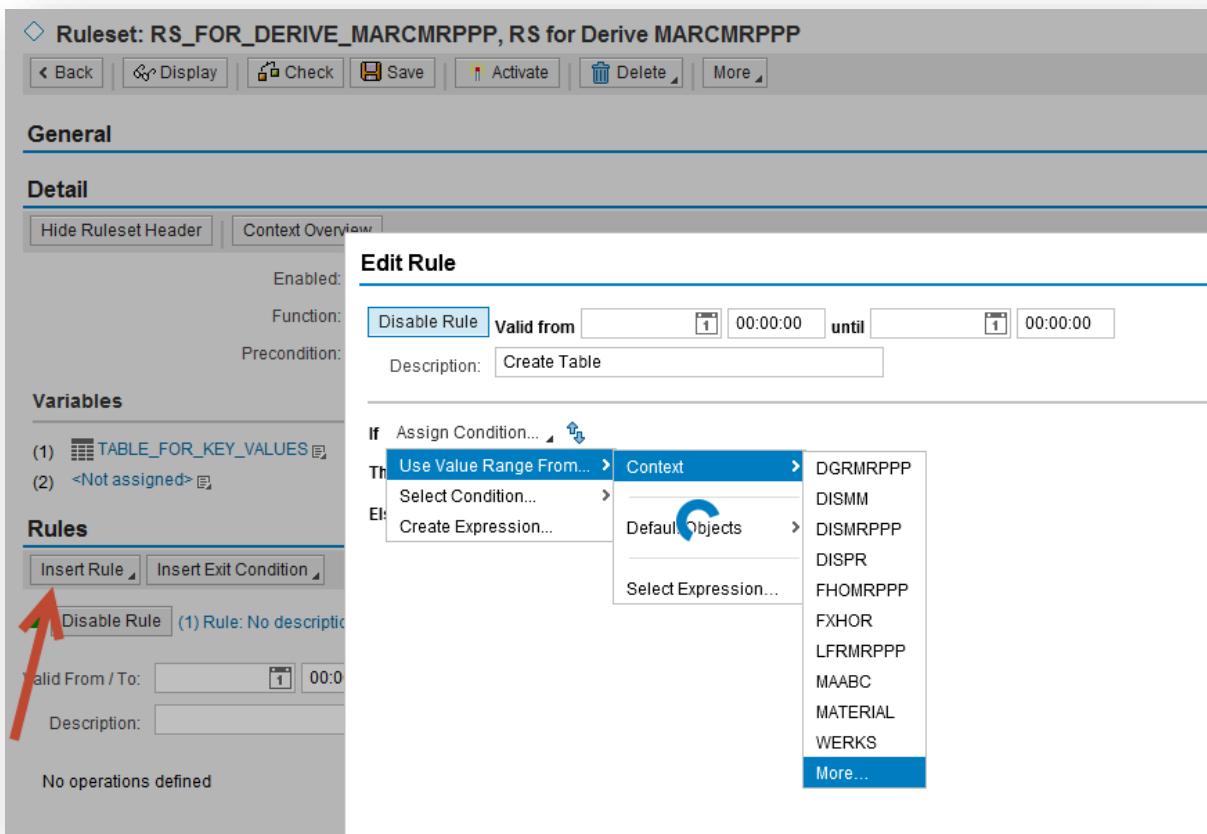


## How-To: Maintain Check and Derivation Rules in MDG for Material

Activate, choose *Back*, and accept your decision to get the following screen:

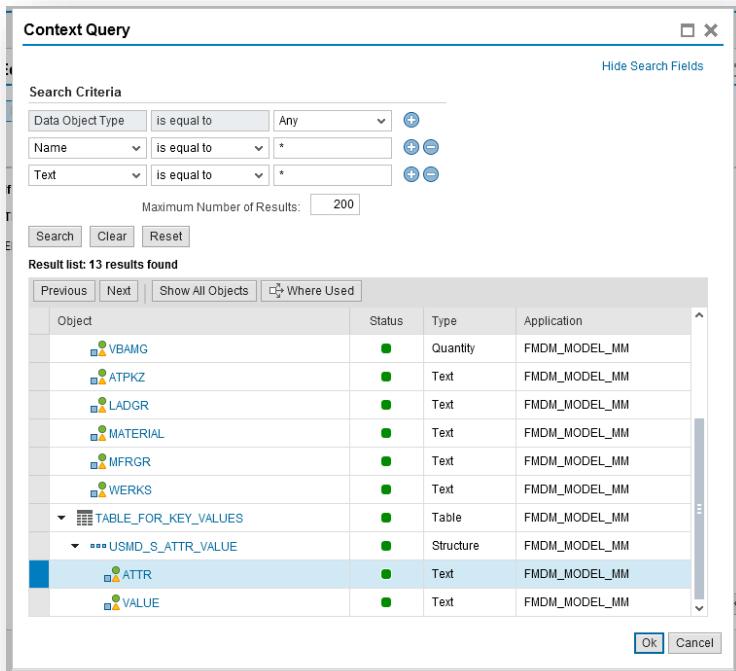


Create the first rule. Choose *Insert Rule* and navigate to *Create*. Enter a *Description*, for example *Create Table*, choose the first *Add* (then-branch), and navigate to *More*.

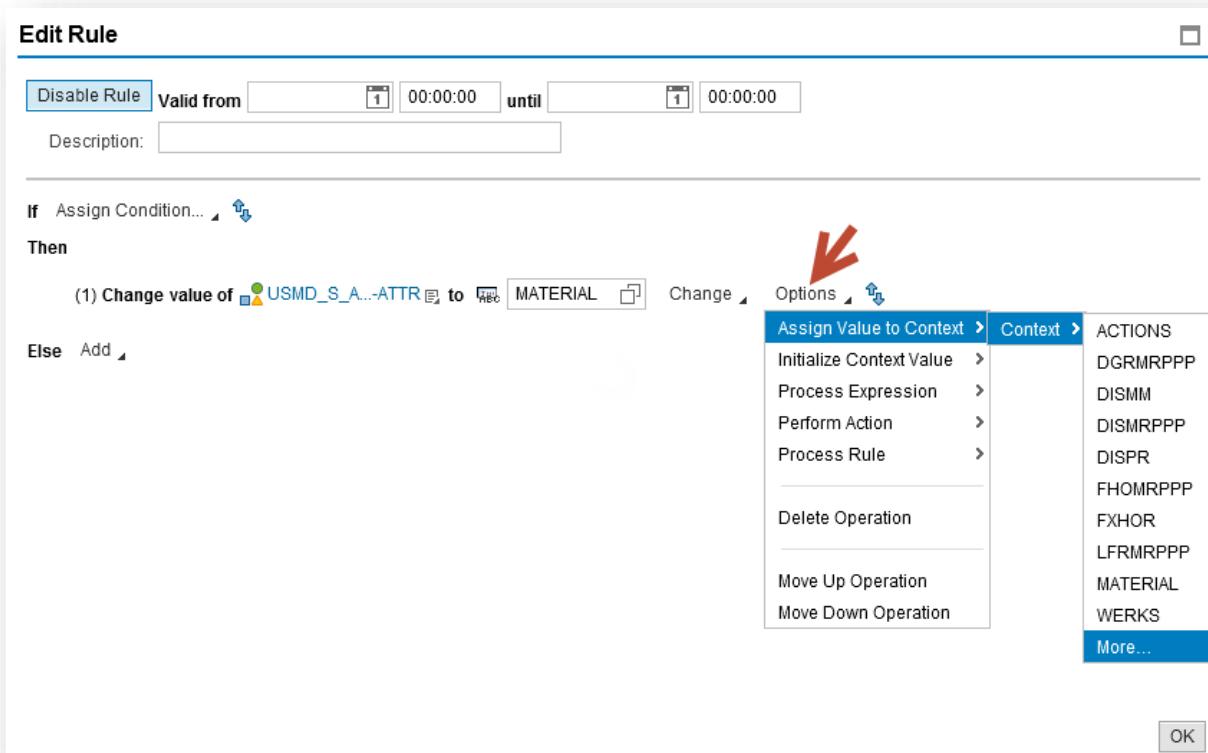


## How-To: Maintain Check and Derivation Rules in MDG for Material

On the obtained screen, navigate to the first component *ATTR* of the structure *USMD\_S\_ATTR\_VALUE*.



Choose *OK* to get the following screen.



Maintain the value *MATERIAL* in the empty box, choose *Options*, and navigate to *More*. Select the second component *VALUE* of the structure *USMD\_S\_ATTR\_VALUE* and choose *OK* to get the following screen:

## How-To: Maintain Check and Derivation Rules in MDG for Material

**Edit Rule**

Disable Rule Valid from [ ] 00:00:00 until [ ] 00:00:00  
Description: [ ]

If Assign Condition... ▾ ↕  
Then

(1) Change value of USMD\_S\_A...-ATTR to MATERIAL □ Change Options ↕  
(2) Change value of USMD\_S\_A...-VALUE to [ ] □ Change Options ↕

Else Add ▾

Choose *Change* and navigate to *MATERIAL* as shown on the following screen:

**Edit Rule**

Disable Rule Valid from [ ] 00:00:00 until [ ] 00:00:00  
Description: [ ]

If Assign Condition... ▾ ↕  
Then

(1) Change value of USMD\_S\_A...-ATTR to MATERIAL □ Change Options ↕  
(2) Change value of USMD\_S\_A...-VALUE to [ ] □ Change Options ↕

Else Add ▾

A red arrow points to the "Change" button in the context menu.

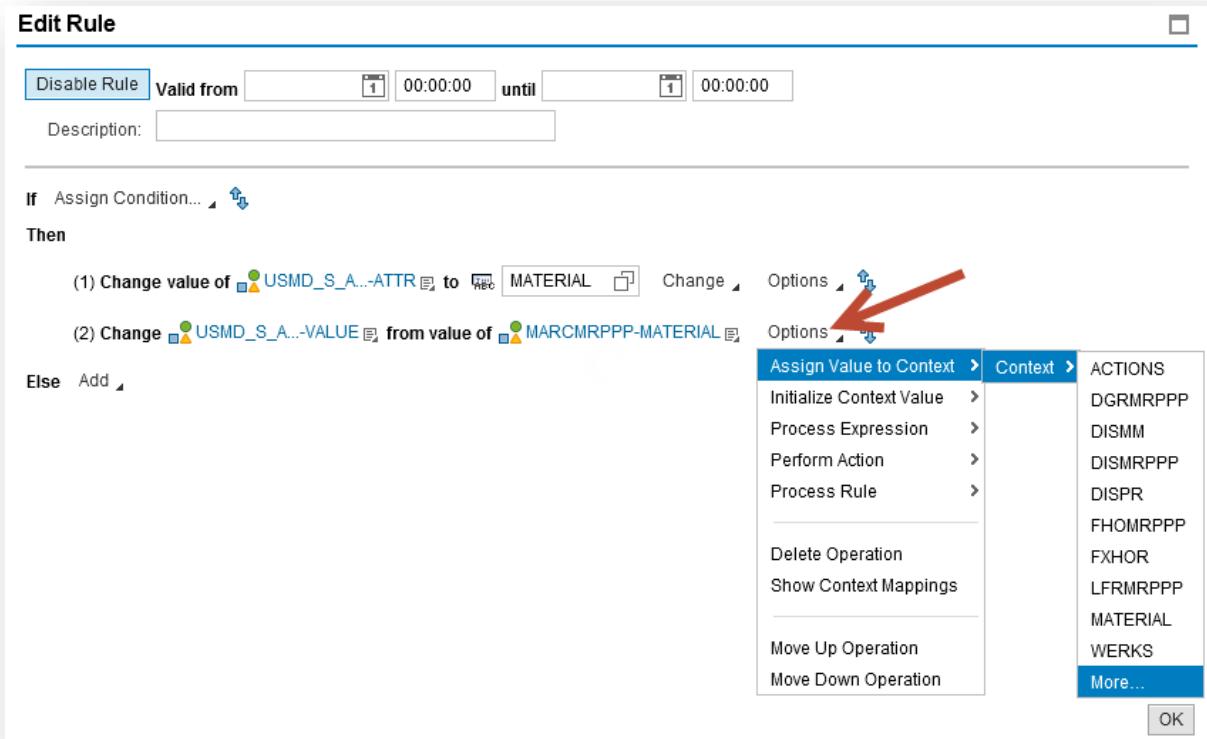
The context menu is open over the "Change" button, showing the following options:

- Context > DGRMRPPP
- Context > DISMM
- Context > DISMRPPP
- Context > DISPR
- Context > FHOMRPPP
- Context > FXHOR
- Context > LFRMRPPP
- Context > MAABC
- MATERIAL** (highlighted in blue)
- WERKS
- More...

OK

## How-To: Maintain Check and Derivation Rules in MDG for Material

Choose *Options* and navigate to *More* as shown below:



## How-To: Maintain Check and Derivation Rules in MDG for Material

On the next screen select the object *Table\_for\_key\_values* and choose OK.

Search Criteria

Data Object Type	is equal to	Any	+/-
Name	is equal to	*	+/-
Text	is equal to	*	+/-

Maximum Number of Results: 200

Search Clear Reset

Result list: 17 results found

Object	Status	Type	Application
DISPR	■	Text	FMDM_MODEL_MM
FHOMRPPP	■	Text	FMDM_MODEL_MM
FXHOR	■	Text	FMDM_MODEL_MM
LFRMRPPP	■	Text	FMDM_MODEL_MM
MAABC	■	Text	FMDM_MODEL_MM
MATERIAL	■	Text	FMDM_MODEL_MM
STDPD	■	Text	FMDM_MODEL_MM
WERKS	■	Text	FMDM_MODEL_MM
TABLE_FOR_KEY_VALUES	■	Table	FMDM_MODEL_MM
USMD_S_ATTR_VALUE	■	Structure	FMDM_MODEL_MM

Ok Cancel

Edit Rule

Disable Rule Valid from [ ] 00:00:00 until [ ] 00:00:00

Description: [ ]

If Assign Condition... ↕  
Then

- (1) Change value of **USMD\_S\_ATTR** to **MATERIAL** ↕ Change Options ↕
- (2) Change **USMD\_S\_ATTRIBUTE** from value of **MARCMRPPP-MATERIAL** ↕ Options ↕
- (3) Change **TABLE\_FOR\_KEY\_VALUES** from value of <Not assigned> ↕ Options ↕

Else Add ↕

Choose *Change* and select the entry *Insert* as shown on the following screen:

## Edit Rule

Disable Rule   Valid from  00:00:00 until  00:00:00

Description:

If Assign Condition... 

Then

(1) Change value of  USMD\_S\_A...-ATTR  to  MATERIAL  Change Options 

(2) Change  USMD\_S\_A...-VALUE  from value of  MARCMRPPP-MATERIAL  Options 

(3) Change  TABLE\_FOR\_KEY\_VALUES  from value of <Not assigned>  Options 

 Else Add 

You get the following screen:

## Edit Rule

Disable Rule   Valid from  00:00:00 until  00:00:00

Description:

If Assign Condition... 

Then

(1) Change value of  USMD\_S\_A...-ATTR  to  MATERIAL  Change Options 

(2) Change  USMD\_S\_A...-VALUE  from value of  MARCMRPPP-MATERIAL  Options 

(3) Insert values into  TABLE\_FOR\_KEY\_VALUES  from <Not assigned>  Options 

Else Add 

Choose the icon next to <Not assigned> and select the structure *USMD\_S\_ATTR\_VALUE*.

The screenshot shows the SAP Fiori interface for maintaining check and derivation rules. At the top, there's a header "Edit Rule" with fields for "Disable Rule", "Valid from" (1/00:00:00), "until" (1/00:00:00), and a "Description" input field. Below this, the "If" section contains an "Assign Condition..." button. The "Then" section lists three steps:

- (1) Change value of `USMD_S_ATTR` to `MATERIAL`
- (2) Change `USMD_S_ATTRIBUTE` from value of `MARCMRPPP-MATERIAL`
- (3) Insert values into `TABLE_FOR_KEY_VALUES` from <Not assigned>

Red arrows point from the "Options" buttons in steps (2) and (3) to the "Ok" button in the "Context Query" dialog.

**Context Query**

Object	Status	Type	Application
<code>FXHOR</code>	■	Text	FMDM_MODEL_MM
<code>LFRMRPPP</code>	■	Text	FMDM_MODEL_MM
<code>MAABC</code>	■	Text	FMDM_MODEL_MM
<code>MATERIAL</code>	■	Text	FMDM_MODEL_MM
<code>STDPD</code>	■	Text	FMDM_MODEL_MM
<code>WERKS</code>	■	Text	FMDM_MODEL_MM
<code>TABLE_FOR_KEY_VALUES</code>	■	Table	FMDM_MODEL_MM
<code>USMD_S_ATTR_VALUE</code>	■	Structure	FMDM_MODEL_MM
<code>ATTR</code>	■	Text	FMDM_MODEL_MM
<code>VALUE</code>	■	Text	FMDM_MODEL_MM

At the bottom right of the dialog are "Ok" and "Cancel" buttons, with a red arrow pointing to the "Ok" button.

Choose *Options* and in the same way as before, create a new entry of the structure `USMD_S_ATTR_VALUE` with `ATTR = WERKS` and the value `MARCMRPPP-WERKS`. Then insert this entry into the table `TABLE_FOR_KEY_VALUES` to get the following result:

**Edit Rule**

Disable Rule   Valid from  00:00:00 until  00:00:00  
Description:

If Assign Condition...

Then

- (1) Change value of USMD\_S\_A...-ATTR to MATERIAL Change Options
- (2) Change USMD\_S\_A...-VALUE from value of MARCMRPPP-MATERIAL Options
- (3) Insert values into TABLE\_FOR\_KEY\_VALUES from USMD\_S\_ATTR\_VALUE Options
- (4) Change value of USMD\_S\_A...-ATTR to WERKS Change Options
- (5) Change USMD\_S\_A...-VALUE from value of MARCMRPPP-WERKS Options
- (6) Insert values into TABLE\_FOR\_KEY\_VALUES from USMD\_S\_ATTR\_VALUE Options

Else Add

Choose OK and Activate and accept your decisions.

**General**

Enabled:

Function: DERIVE\_MARCMRPPP

Precondition: <Not assigned>

**Variables**

- (1) TABLE\_FOR\_KEY\_VALUES
- (2) <Not assigned>

**Rules**

Insert Rule | Insert Exit Condition

Disable Rule (1) Rule: No description is available - Unlimited Validity

- (1) Change value of USMD\_S\_A..-ATTR to MATERIAL
- (2) Change USMD\_S\_A..-VALUE from value of MARCMRPPP-MATERIAL
- (3) Insert values into TABLE\_FOR\_KEY\_VALUES from USMD\_S\_ATTR\_VALUE
- (4) Change value of USMD\_S\_A..-ATTR to WERKS
- (5) Change USMD\_S\_A..-VALUE from value of MARCMRPPP-WERKS
- (6) Insert values into TABLE\_FOR\_KEY\_VALUES from USMD\_S\_ATTR\_VALUE

Create the second rule.  
Choose *Options* and choose *Insert Rule* and select *Create*.

Rules

Insert Rule | Insert Exit Condition

Disable Rule (1) Rule: Create Table - Unlimited Validity

(1) Change value of USMD\_S\_A..-ATTR to MATERIAL  
 (2) Change USMD\_S\_A..-VALUE from value of MARCMRPPP-MATERIAL  
 (3) Insert values into TABLE\_FOR\_KEY\_VALUES from USMD\_S\_ATTR\_VALUE  
 (4) Change value of USMD\_S\_A..-ATTR to WERKS  
 (5) Change USMD\_S\_A..-VALUE from value of MARCMRPPP-WERKS  
 (6) Insert values into TABLE\_FOR\_KEY\_VALUES from USMD\_S\_ATTR\_VALUE

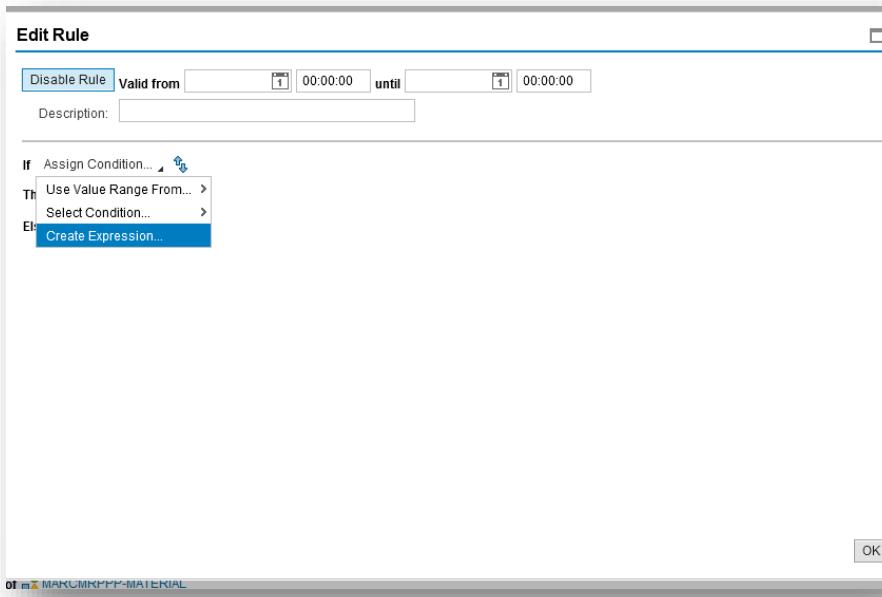
Options

- Create... | Insert Next Rule
- Select
- Remove
- Copy
- Move Down
- Show Header

## How-To: Maintain Check and Derivation Rules in MDG for Material

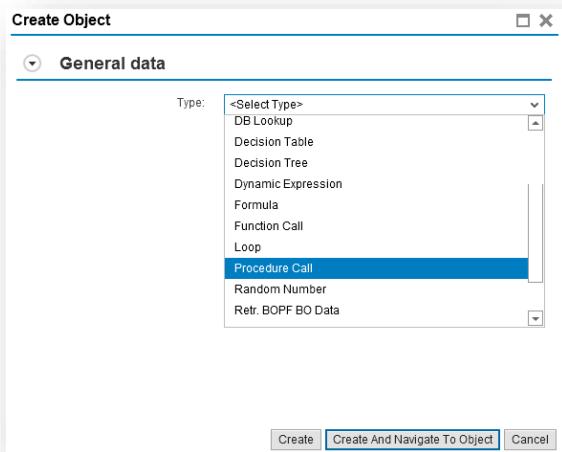
---

On the next screen choose *Assigned Condition* and navigate to *Create Expression* as shown below.



Enter a *Description*, for example *Get\_attribute value*.

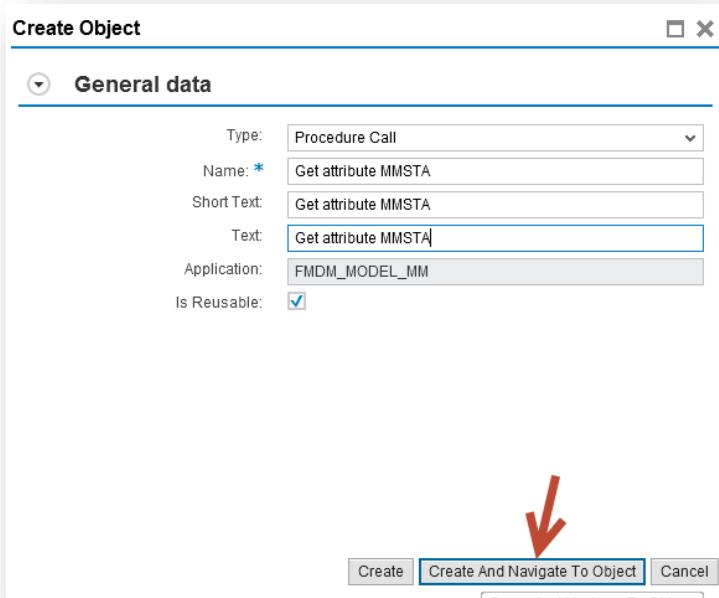
Choose *Assign Condition* and navigate to *Create Expression* and select type "Procedure Call" to get the following screen:



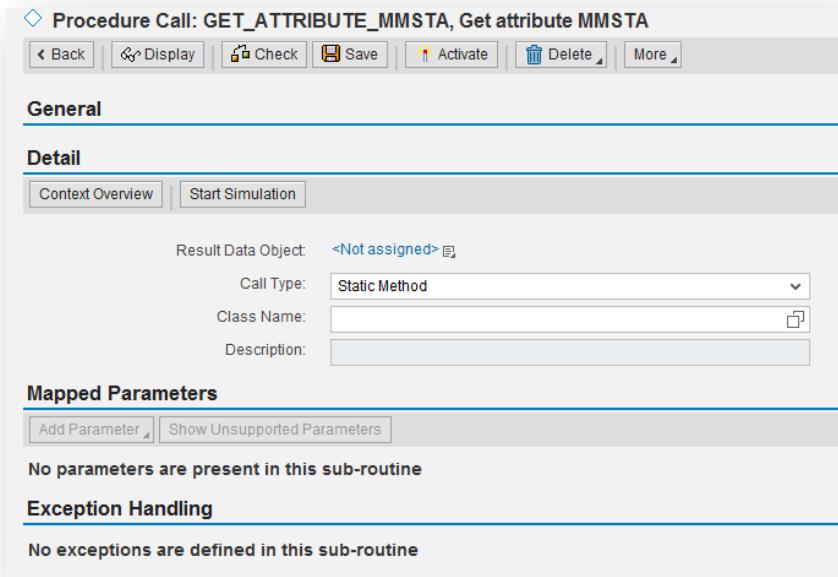
## How-To: Maintain Check and Derivation Rules in MDG for Material

---

On the next screen enter a name, for example *Get attribute MMSTA*, a *Short Text* and a *Text*, as shown below.



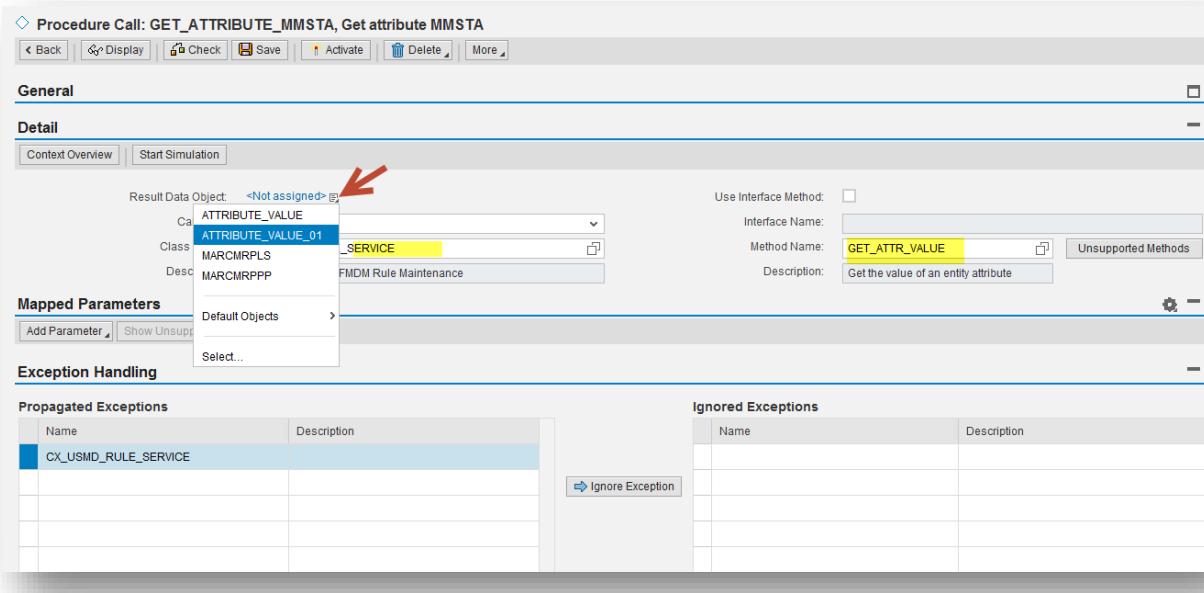
Choose *Create And Navigate To Object* and choose Yes to get the following result:



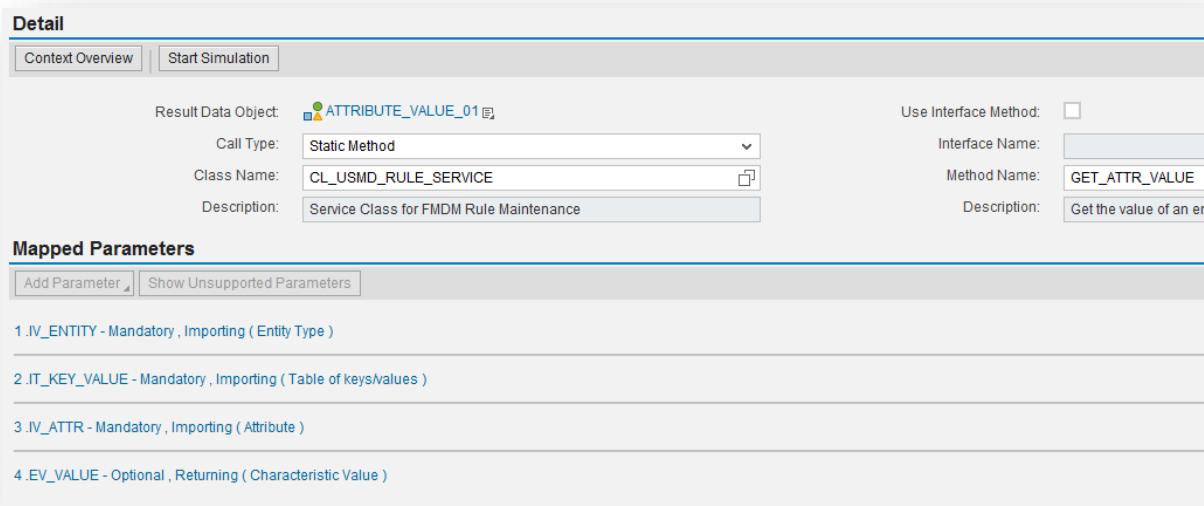
Enter *CL\_USMD\_RULE\_SERVICE* in the field *Class Name* and *GET\_ATTR\_VALUE* in the field *Method Name*. In the field *Result Data Object* assign the value *Attribute Value*: Choose the icon right to <Not

## How-To: Maintain Check and Derivation Rules in MDG for Material

assigned> and choose the value ATTRIBUTE\_VALUE\_01 to get the following screen:

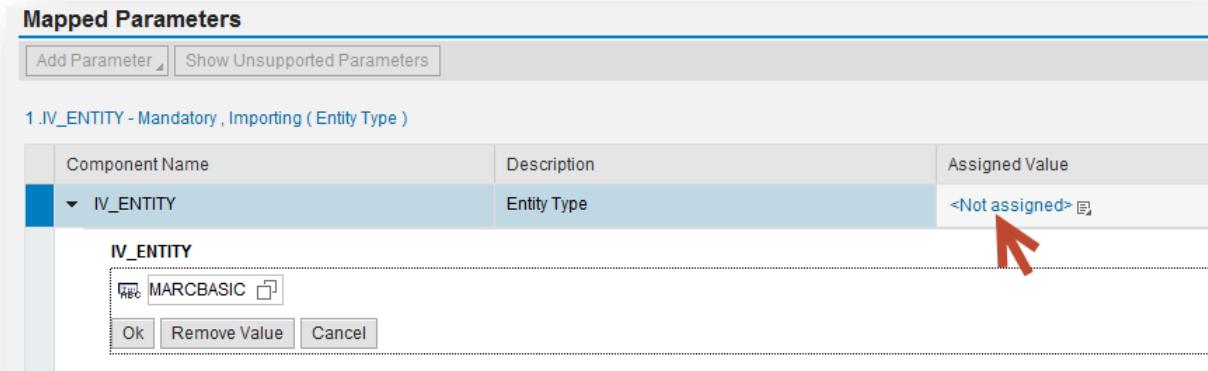


Choose “Add Parameter” and select all importing and exporting parameters one after the other. After these steps, you get the following result:



## How-To: Maintain Check and Derivation Rules in MDG for Material

Choose the parameter *IV\_ENTITY*, choose <Not assigned> and enter the name of your source entity, in this case *MARCBASIC*.



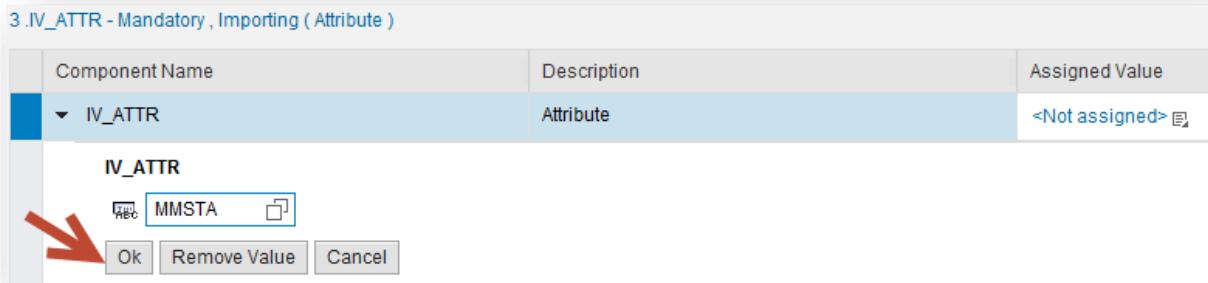
Choose *OK*. Select the parameter *IT\_KEY\_VALUE* and then choose the icon right to <Not assigned> and navigate to *Select Context Parameter* as shown below:



Select the entry *TABLE\_FOR\_KEY\_VALUE* to get the following screen:



Choose the parameter *IV\_ATTR* and then choose <Not assigned>. On the next screen enter the value of your attribute, in this case *MMSTA*:



## How-To: Maintain Check and Derivation Rules in MDG for Material

Choose OK to get the following screen:

3.IV_ATTR - Mandatory , Importing ( Attribute )				
Component Name	Description	Assigned Value	Move Type	
IV_ATTR	Attribute	MMSTA	Move Value	

Choose on the last parameter *EV\_VALUE* and then choose the icon right to <Not Assigned> and select the entry *ATTRIBUTE\_VALUE\_01* as shown below:

4.EV_VALUE - Optional , Returning ( Characteristic Value )				
Component Name	Description	Assigned Value		
EV_VALUE	Characteristic Value	<Not assigned>	ATTRIBUTE_VALUE_01	Select Context Parameter...
Exception Handling				

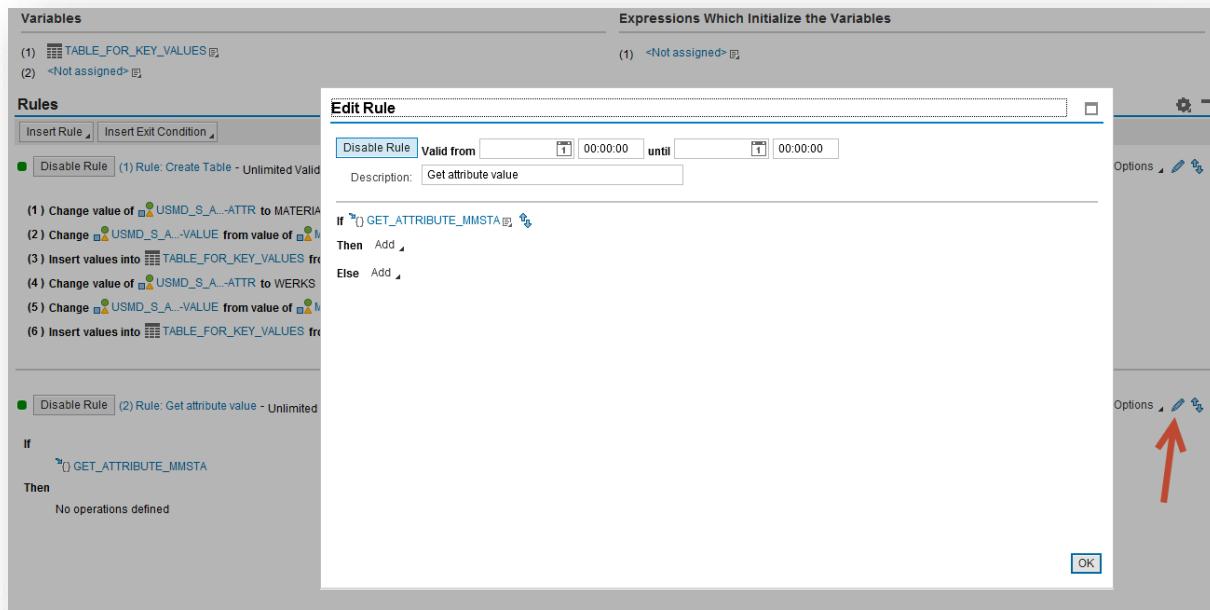
After this step, you get the following screen. Activate and choose Yes.

◇ Procedure Call: GET_ATTRIBUTE_MMSTA, Get attribute MMSTA				
<a href="#">Back</a> <a href="#">Display</a> <a href="#">Check</a> <a href="#">Save</a> <a href="#">Activate</a> <a href="#">Delete</a> <a href="#">More</a>				
<b>General</b>				
<b>Detail</b>				
<a href="#">Context Overview</a> <a href="#">Start Simulation</a>				
Result Data Object: <a href="#">ATTRIBUTE_VALUE_01</a>				
Call Type:	Static Method	Use Interface Method:	<input type="checkbox"/>	
Class Name:	CL_USMD_RULE_SERVICE	Interface Name:	<input type="text"/> GET_ATTR_VALUE	
Description:	Service Class for FMDM Rule Maintenance	Description:	Get the value of an entity attribute	
<b>Mapped Parameters</b>				
<a href="#">Add Parameter</a> <a href="#">Show Unsupported Parameters</a>				
1.EV_VALUE - Optional , Returning ( Characteristic Value )				
Component Name	Description	Assigned Value	Move Type	
EV_VALUE	Characteristic Value	ATTRIBUTE_VALUE_01	Move Value	
2.IT_KEY_VALUE - Mandatory , Importing ( Table of keys/values )				
Source Table for Column Update: <Not assigned>				
Component Name	Description	Assigned Value	Move Type	Map Element to
IT_KEY_VALUE	Table of keys/values	TABLE_FOR_KEY_VALUES	Move Corresponding Fields	<input type="checkbox"/>
3.IV_ATTR - Mandatory , Importing ( Attribute )				
Component Name	Description	Assigned Value	Move Type	
IV_ATTR	Attribute	MMSTA	Move Value	
4.IV_ENTITY - Mandatory , Importing ( Entity Type )				
Component Name	Description	Assigned Value	Move Type	
IV_ENTITY	Entity Type	MARC BASIC	Move Value	

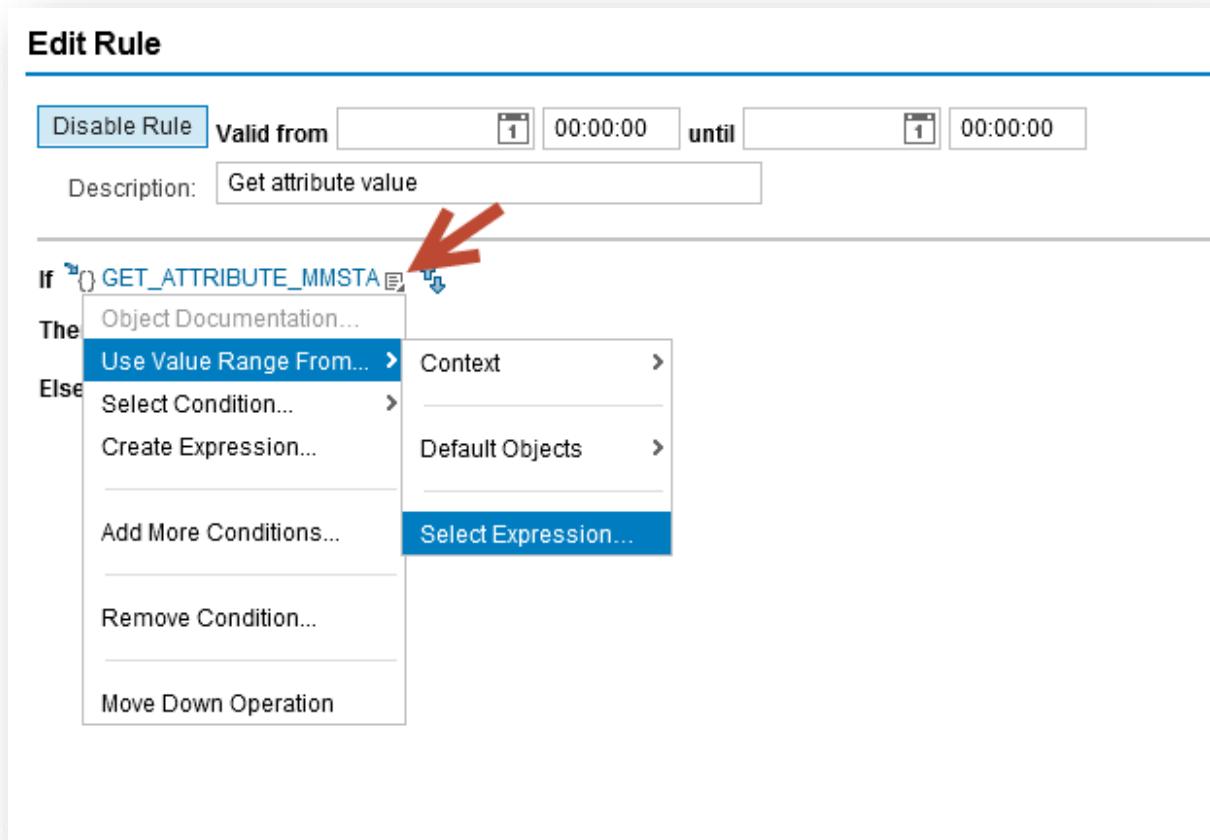
Go back to the Ruleset. Make sure that the sequence of the rules is correct.

## How-To: Maintain Check and Derivation Rules in MDG for Material

Choose *Edit* in the second rule to get the following screen:



Choose the icon right of the action *GET\_ATTRIBUTE\_MMSTA* and navigate to *Select Expression*, as shown below:



## How-To: Maintain Check and Derivation Rules in MDG for Material

On the next screen, select your Object *GET\_ATTRIBUTE\_MMSTA* created before as shown below:

**Object Query** X

[Hide Search Fields](#)

**Search Criteria**

Application Name	is equal to	FMDM_MODEL_MM	<a href="#">+</a> <a href="#">-</a>
Expression Type	is equal to	Any	<a href="#">+</a> <a href="#">-</a>
Name	is equal to	get_attribute_m*	<a href="#">+</a> <a href="#">-</a>
Text	is equal to	*	<a href="#">+</a> <a href="#">-</a>

Also include objects from default BRFplus application:

Maximum Number of Results:

[Search](#) [Clear](#) [Reset](#)

**Result list: 1 results found**

[Previous](#) [Next](#) | [Show All Objects](#) [Where Used](#)

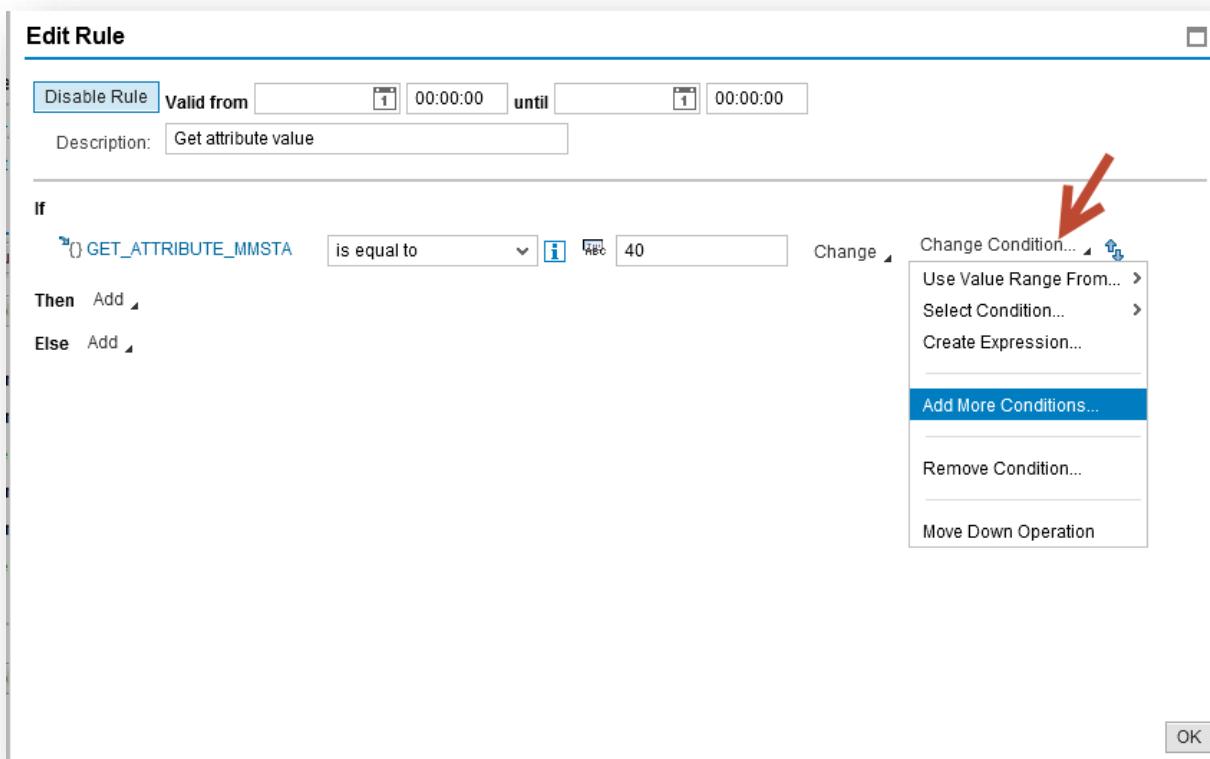
Object	Status	Type	Application
GET_ATTRIBUTE_MMSTA	■	Procedure Call	FMDM_MODEL_MM

[Ok](#) [Cancel](#)

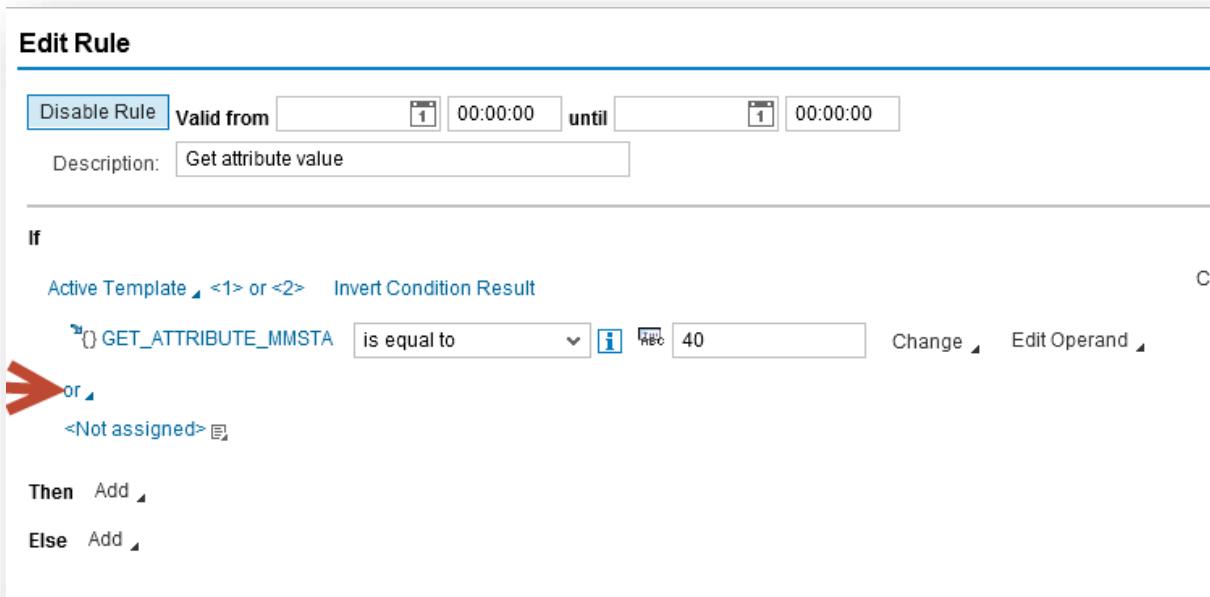


## How-To: Maintain Check and Derivation Rules in MDG for Material

Choose **OK** to get the next screen:

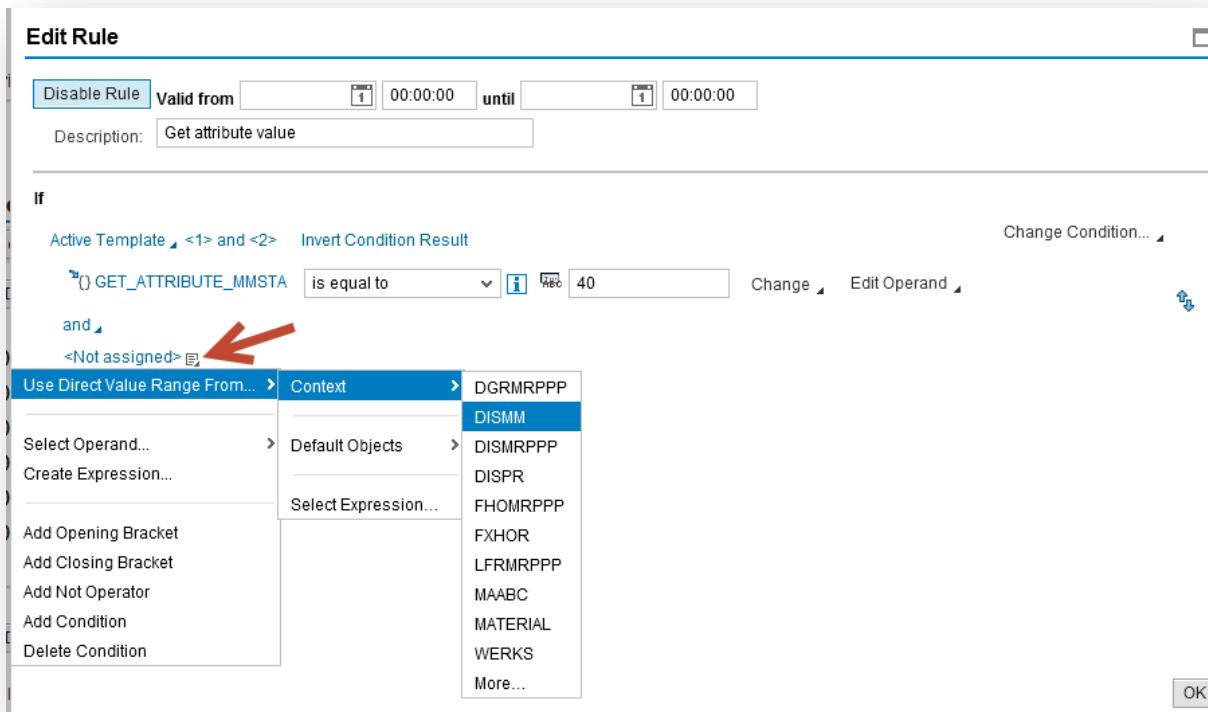


Enter your value "40" in the empty field. Choose *or* and change it to *and*.

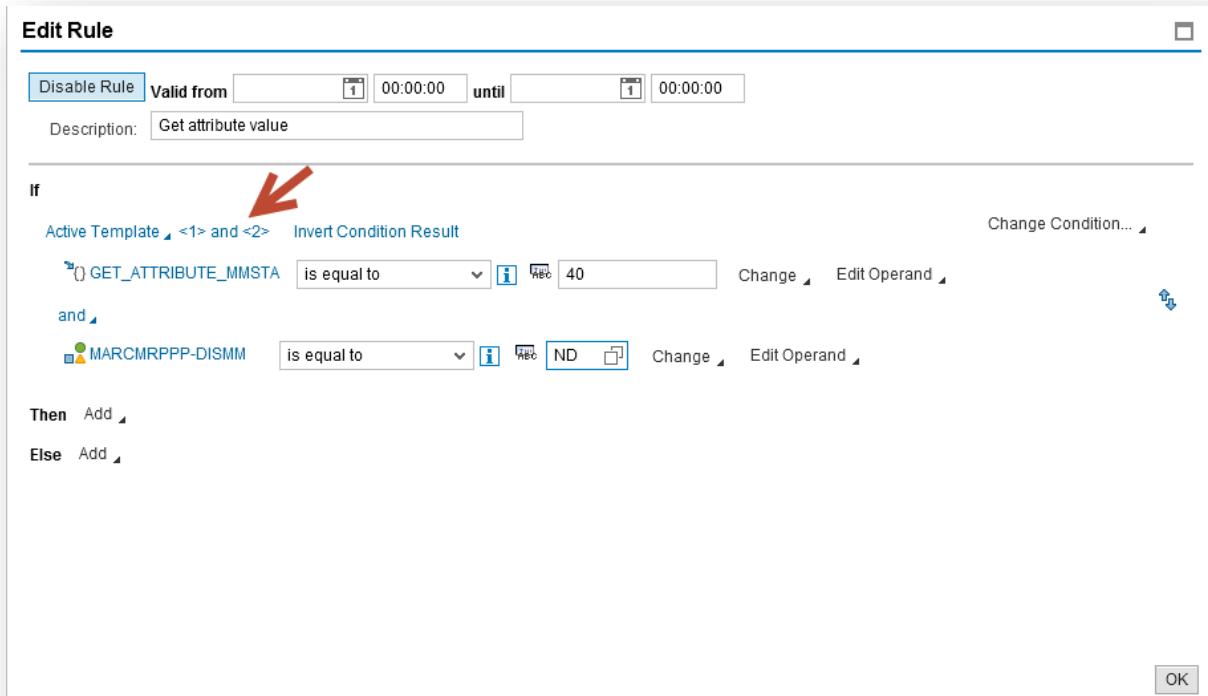


## How-To: Maintain Check and Derivation Rules in MDG for Material

Choose the icon right of <Not assigned> and navigate to *DISMM*, as shown below:

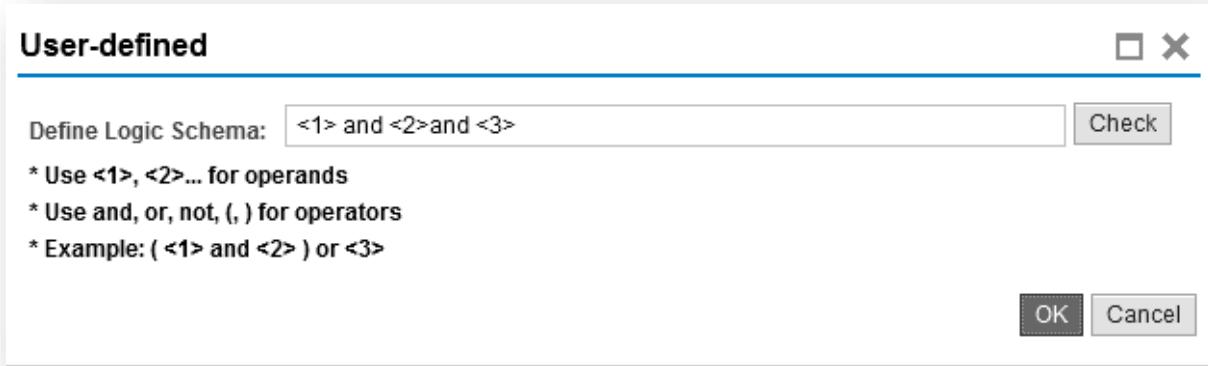


Enter your value “ND” in the empty field. Then change the Active Template. Click on <1>and<2>.

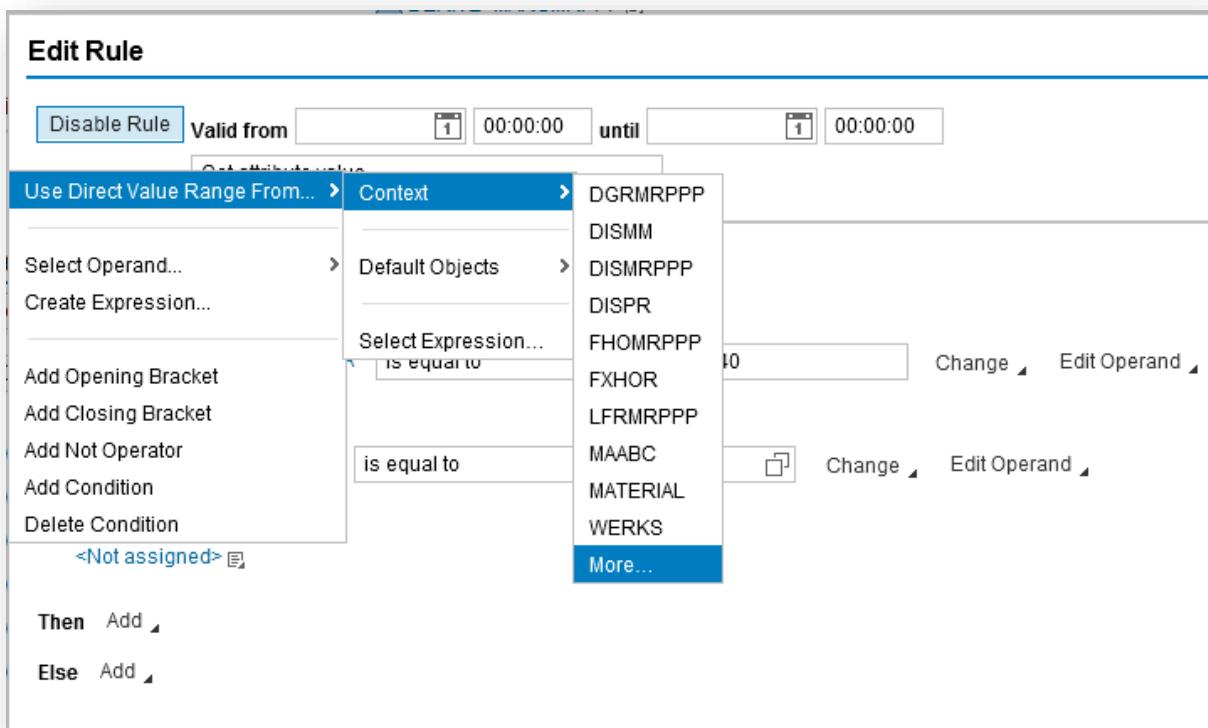


## How-To: Maintain Check and Derivation Rules in MDG for Material

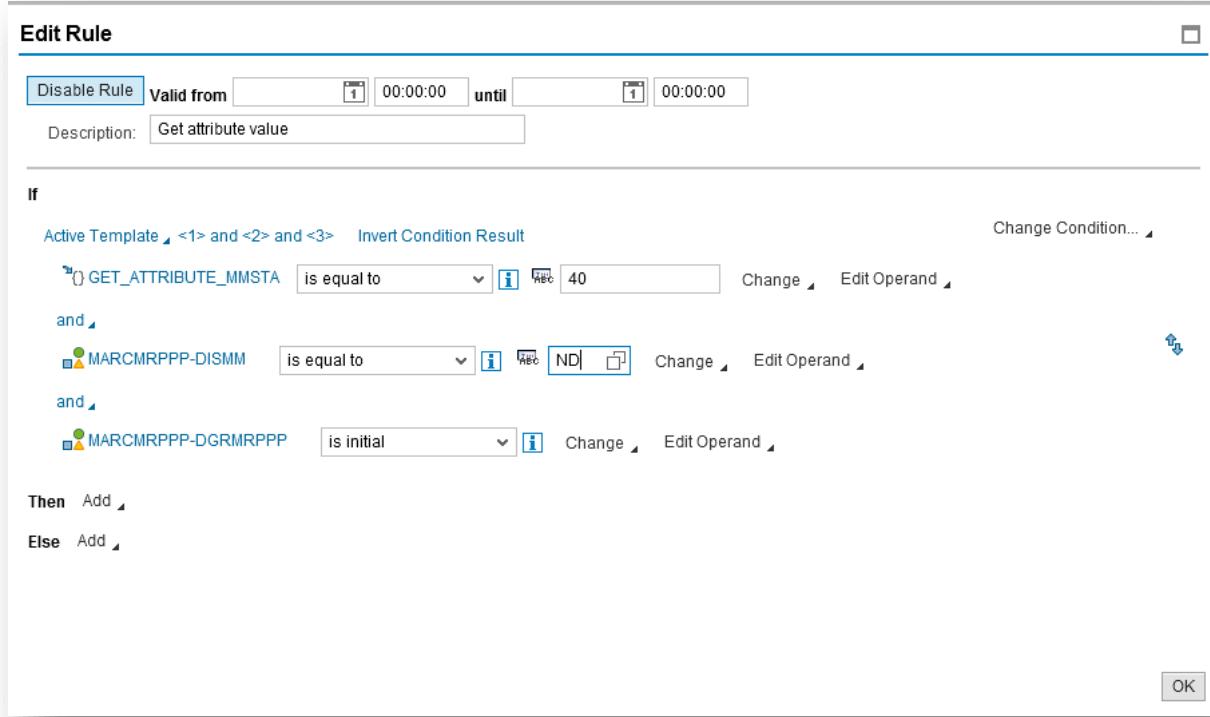
Enhance the logic schema with value “and <3>”.



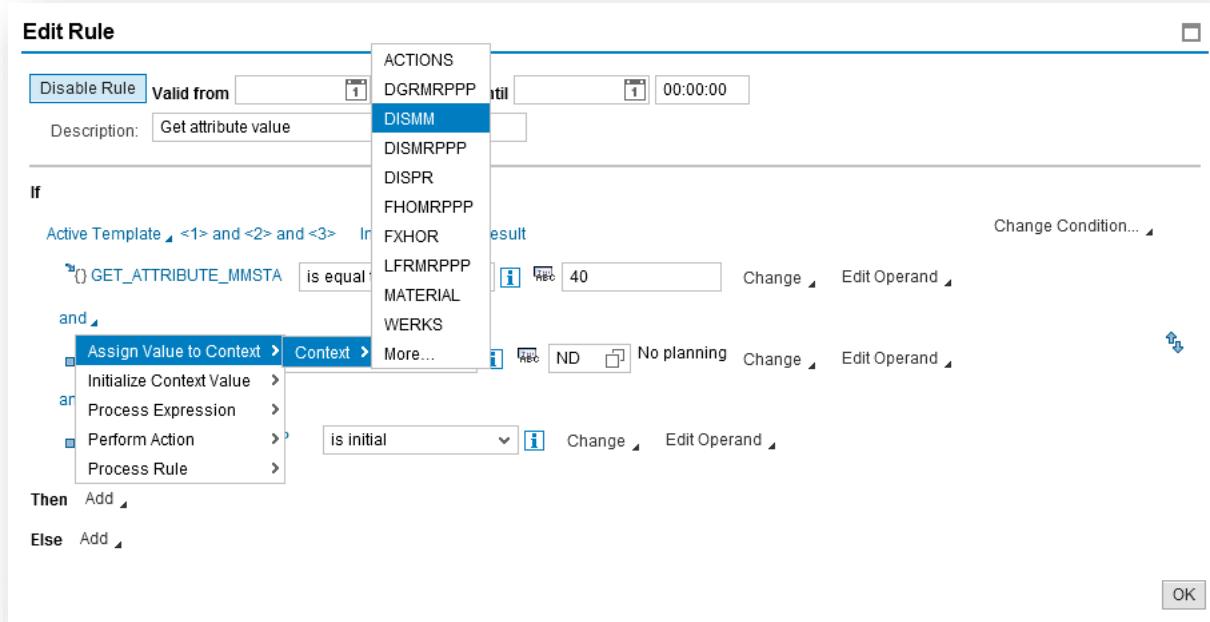
Add new condition stating that DGRMRPPP should be initial.



## How-To: Maintain Check and Derivation Rules in MDG for Material

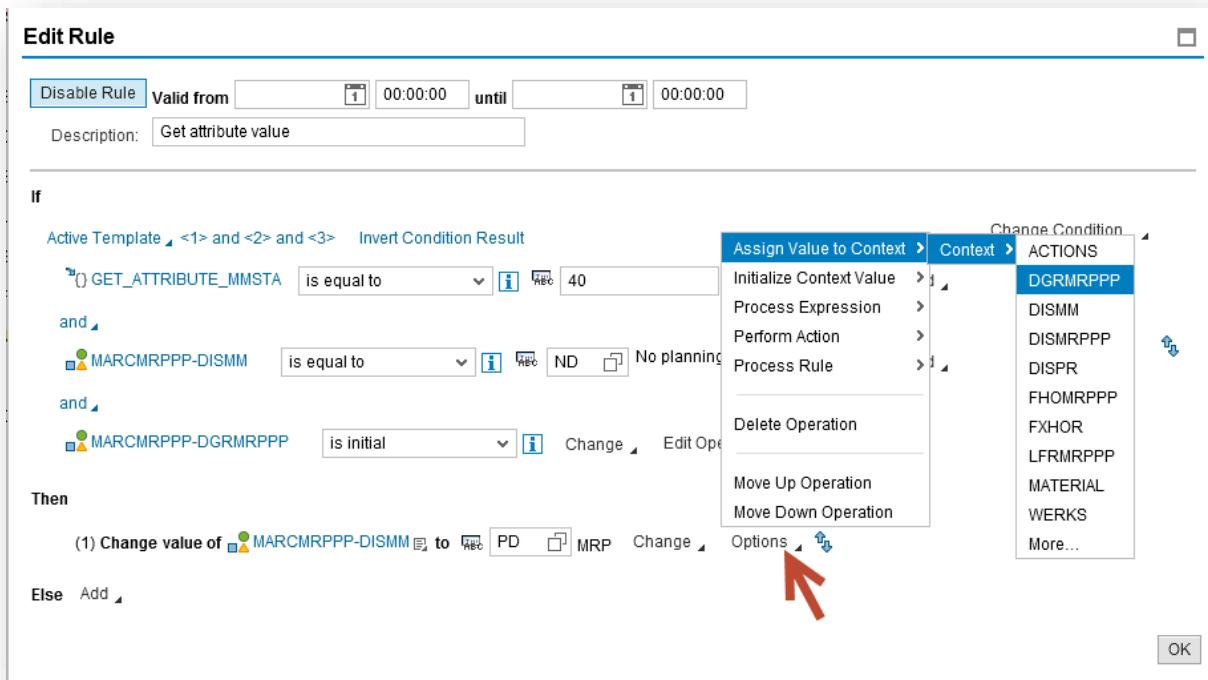


Choose the first *Add* (then-branch) and navigate to your attribute of the target entity, in this case *DISM*M as shown below:



## How-To: Maintain Check and Derivation Rules in MDG for Material

Assign value 'PD'. Then click on Options and to the same for DGRMRPPP and enter your value for it.



Enter your value, in this case '0001', in the empty rectangle and choose OK.

## How-To: Maintain Check and Derivation Rules in MDG for Material

Activate and then accept your decision. Choose *Back* and activate.

The screenshot shows the SAP MDG interface for maintaining check and derivation rules. The title bar indicates the ruleset is for deriving MARCMRPPP from material. The toolbar includes Back, Display, Check, Save, Activate (highlighted by a red arrow), Delete, and More. The General tab is selected, showing the rule is enabled, its function is DERIVE\_MARCMRPPP, and its precondition is <Not assigned>. The Detail tab is also visible. The Variables tab lists two variables: TABLE\_FOR\_KEY\_VALUES and <Not assigned>. The Expressions Which Initialize the Variables tab shows the expression <Not assigned>. The Rules tab contains two sections: one for creating a table with six steps involving USMD\_S\_ATTR and MARCMRPPP, and another for getting attribute values with conditions on GET\_ATTRIBUTE\_MMSTA, MARCMRPPP-DISMM, and MARCMRPPP-DGRMRPPP, followed by changes to MARCMRPPP-DISMM and MARCMRPPP-DGRMRPPP.

Go back and activate the function.

Now the BRF+ derivation with ruleset containing a rule to generate the key values table and a rule containing a procedure is available.

### 8.1.5 Add method to the Post-Exit

Run transaction SE24. Open class interface CL\_USMD\_CONV\_SOM\_GOV\_API. Go to the Post-Exit in method IF\_USMD\_CONV\_SOM\_GOV\_ENTITY~WRITE\_ENTITY\_DATA.

To enforce the BRF+ call for the target entity, add the following code:

```
data:  
    lv_field          type usmd_fieldname,
```

```
lt_source_attr           type usmd_t_fieldname,  
lt_target_key            type usmd_t_fieldname.  
* Append the name of the corresponding source entity attribute on which  
* the update of the target entity data depends  
clear lt_source_attr.  
lv_field = 'MMSTA'.  
append lv_field to lt_source_attr.  
  
* Append the keys of your target entity  
clear lt_target_key.  
lv_field = 'MATERIAL'.  
append lv_field to lt_target_key.  
lv_field = 'WERKS'.  
append lv_field to lt_target_key.  
  
Z_CL_ENFORCE_BRF_PLUS_CALL=>ENFORCE_BRF_PLUS_CALL(  
    exporting  
        iv_model      = 'MM' "your model"  
        it_entity_data = it_entity_data  
        iv_source_entity = 'MARCBASIC' "your source entity"  
        io_core_object = core_object  
        it_source_attr = lt_source_attr  
        iv_target_entity = 'MARCMRPPP' "your target entity"  
        it_target_key = lt_target_key  
).
```

Note:

- For the creation of the procedure call Get attribute value you can use the method CHECK\_ATTR\_VALUE(CL\_USMD\_RULE\_SERVICE) instead of the method GET\_ATTR\_VALUE(CL\_USMD\_RULE\_SERVICE).
- Method IS\_ENTITY\_EMPTY(CL\_USMD\_RULE\_SERVICE) can also be useful to check whether the data of an entity is empty.

#### 8.1.6 Test the BRF+ Rule

Change an existing material. Go to the plant details. Open UIBB for Plant: General Data and Plant: Material Requirement Planning.

## How-To: Maintain Check and Derivation Rules in MDG for Material

Check the existing values for Plant-Specific Material Status, MRP Group, and MRP Type.

Material: P-260514-01; Plant: 0001

Done | Expand all | Collapse all | Side Panel

Plant: General Data

General Data	Forecast Requirements
Logistics Handling Group:	Splitting Indicator:
Plant-Specific Material Status:	Fiscal Year Variant:
Profit Center:	Period Indicator:
Serial Number Profile:	Total Replenishment Lead Time:
Documentation Required:	* Availability Check Group:
Negative Stocks in Plant Allowed:	Cross-Project Material:

Plant: Sales

Plant: Foreign Trade

Plant: Purchasing

Plant: Material Requirement Planning [Details]

General Data	MRP Procedure
MRP Profile:	* MRP Type: ND
MRP Group:	MRP Controller: No planning
ABC Indicator:	Planning Time Fence:
	Planning Cycle:
	Reorder Point: 0,000

Now, enter the value '40' for Plant-Specific Material Status and press return. The result should be the following.

Material: P-260514-01; Plant: 0001

Done | Expand all | Collapse all | Side Panel

Plant: General Data

General Data	Forecast Requirements
Logistics Handling Group:	Splitting Indicator:
Plant-Specific Material Status: 40	Fiscal Year Variant:
Profit Center:	Period Indicator:
Serial Number Profile:	Total Replenishment Lead Time:
Documentation Required:	* Availability Check Group:
Negative Stocks in Plant Allowed:	Cross-Project Material:

Plant: Sales

Plant: Foreign Trade

Plant: Purchasing

Plant: Material Requirement Planning [Details]

General Data	MRP Procedure
MRP Profile:	* MRP Type: PD
MRP Group: 0001	MRP Controller: MRP
ABC Indicator:	Planning Time Fence:
	Planning Cycle:
	Reorder Point: 0,000

## 9. CROSS-ENTITY DERIVATION WITH BADI USMD\_RULE\_SERVICE\_CROSS\_ET

The BAdI USMD\_RULE\_SERVICE\_CROSS\_ET is called at least once at every round trip if data was changed. Technical note: the BAdI is triggered in the Flush Method of the Governance API.

- Valid for Single Item Maintenance UI, Multi-Record Processing, application USMD\_MASS\_CHANGE and Data import
- Messages from derives are always converted into information messages.
- The BAdI gets only the attributes that are changed since the last BAdI call. Unchanged attributes are not transferred by the interface of this BAdI. You need to use the IO\_MODEL parameter (method READ\_CHAR\_VALUE or RETRIEVE\_CHAR\_VALUE) to determine their values. For more information, see the documentation of interface **IF\_USMD\_MODEL\_EXT**.

Hint: Set parameter IF\_NO\_FLUSH to ABAP\_TRUE to avoid additional calls of the Flush method.

### Example Scenario:

You want to derive the material description from the characteristic valuation of the characteristics AAD268002 and AAD126067 (classification).

### Example Solution:

SAP Notes 1806103 and 1815882 provide the example BAdI implementation MDG\_BS\_MAT\_CLF2DESCRIPTION for the cross entity derive BAdI USMD\_RULE\_SERVICE\_CROSS\_ET.

SAP Note 1818020 is also required.

In SE18, enhancement spot USMD\_RULE\_SERVICE, copy the example implementation

Implementations		
BAdI Definition	USMD_RULE_SERVICE_CROSS_ET	
	Enhancement Implementation	BAdI Implementation
	MDG_BS_MAT_CLF2DESCRIPTION	MDG_BS_MAT_CLF2DESCRIPTION
	ZEM_MAT_DESC_DERIVE	ZEM_MAT_DESC_DERIVE
	Description	
	USMD_RULE_SERVICE_EXT Implementation for Test Model	
	Example implementation: Derive description from classification	
	ZEM_MAT_DESC_DERIVE from Classification	

Change the characteristics in the attributes of the class interface to your values:

Attribute	Level	Visibility	R...	Typing	Associated Type	Description	Initial value
GC_CLASSTYPE	Constant	Private	<input type="checkbox"/>	Type	KLASSENART	Class Type	'001'
GC_CHARNAME1	Constant	Private	<input type="checkbox"/>	Type	ATNAM	Characteristic Name	'AAD268002'
GC_CHARNAME2	Constant	Private	<input type="checkbox"/>	Type	ATNAM	Characteristic Name	'AAD126067'
GV_CHARID1	Static Att.	Private	<input type="checkbox"/>	Type	MDG_BS_CLF_CHAR...	Internal characteristic	
GV_CHARID2	Static Att.	Private	<input type="checkbox"/>	Type	MDG_BS_CLF_CHAR...	Internal characteristic	

### Example Characteristics:

Char.	Description	Dat...	N...	D...	Unit	R..	Org. Area
AAD268002	Guideline for conformity decla	CHAR	20	0			
BAH609003	Max. speed of the drive	NUM	6	0	1/min		
BAJ196003	Design temperature	NUM	3	0	K		
BAI188003	Max. overpressure of the shaft	NUM	3	2	bar		
AAD375002	Load moment of the drive	NUM	6	0			
AAD126067	Text for Centrifugal pump	CHAR	13	0			

Result on the material UI: The BAdl implementation concatenates two characteristic valuations to a new material description in two languages.

Language Key	Language	Material Description
EN	English	4711:SUBMERSIBLE
FR	French	4711:SUBMERSIBLE

Characteristic Values		
Max. speed of the drive:	33.000 <input type="button"/> 1/min <input type="button"/> 1/minute	<a href="#">Details</a>
Design temperature:	230 <input type="button"/> K <input type="button"/> Kelvin	<a href="#">Details</a>
Max. overpressure of the shaft:	6.00 <input type="button"/> bar <input type="button"/> bar	<a href="#">Details</a>
Load moment of the drive:	4.000 <input type="button"/>	<a href="#">Details</a>
Text for Centrifugal pump:	SUBMERSIBLE <input type="button"/> Submersible	<a href="#">Details</a>
Guideline for conformity decla:	4711 <input type="button"/> Guideline Name 4711	<a href="#">Details</a>

### Limitation:

The material description is updated in the staging buffer and can be saved. But on the UI, the updated data is only displayed after the next roundtrip. That means if you change only the classification values the description is not immediately updated on the UI. You need to do another user action (such as choosing the Check button or changing other values) to see the updated data.

### Technical background:

The reason for this behavior is that the classification UIBB (reuse component) is not embedded into the FPM phase model. The event CLF\_CHANGED of Web Dynpro component /PLMU/WDC\_CLF is raised in component controller method RETRIEVE\_CLASSES\_VALUES to inform consumers that need to synchronize data. This event is actual part of the Web Dynpro phase WDDOMODIFYVIEW that does not allow triggering an additional roundtrip in FPM. FPM needs to request an additional FPM roundtrip during processing of a Web Dynpro action handler method. Consumers of the Web Dynpro event for classification changed cannot trigger any FPM roundtrip in WDC phase modify view.

## 10. TECHNICAL ENVIRONMENT BRF+ LANDSCAPE AND RULE MAINTENANCE

### 10.1 Use case

The application FMDM\_MODEL\_MM is a customizing object, client dependent, and transportable.

The Customer wants to edit the content of decision tables for Rules locally in the production system. The definition of Validation and Derivation Rules (USMD\_RULE) is customizing. Therefore, the maintenance depends on the client settings (SCC4 - Client Maintenance). However, some customers want to change transported BRF+ customizing objects in the production system independent from the change settings for the client customizing in transaction SCC4.

Solution:

Customer needs to decide what he wants to transport across the landscape OR transport once and change in the production system. A possible solution is provided by SAP Note 1462550. It describes how the change settings can be overwritten in an application exit (follow-up SAP Note 1501679 might be also relevant).

### 10.2 Example for Application Exit GET\_CHANGEABILITY

The changeability exit allows overruling the client settings for the changeability of client-dependent objects. Also, the automatic recording of changes on transport requests can be adapted. However, when transportable customizing objects become editable in this way, changes may get lost with the next transport from a customizing system or a client copy. See document BRF+ Application Exits:

<http://scn.sap.com/docs/DOC-4564>. In this document, you can also find methods for additional authorization checks.

#### IF\_FDT\_APPLICATION\_SETTINGS

General Implementation

The interface IF\_FDT\_APPLICATION\_SETTINGS defines a set of static methods – the actual Application Exits. For each method/exit a corresponding Boolean attribute GV\_<method name> is defined by the interface. The attribute is used to indicate whether the interface-method is implemented and the exit shall be used. An exit-method is called by the BRF+ framework only if the corresponding Boolean attribute has been set to true. Setting the attributes to true is best done inside a static class constructor, as in the following example:

```
METHOD class_constructor.  
  if_fdt_application_settings~gv_authority_check = abap_true.  
  if_fdt_application_settings~gv_get_element_values = abap_true.  
ENDMETHOD.
```

Depending on the triggering event and the called method's interface, the application exit may influence the behavior of BRF+' object handling. Since the currently executed program exits the BRF+ at this point, it is in the responsibility of the programmer to avoid any kind of runtime errors.

```
METHOD if_fdt_application_settings~get_changeability.  
  
  DATA: l_expression_type_id TYPE if_fdt_types=>id.  
  
  *--Entire BRF+ application is allowed to be modified in Dev system.  
  IF sy-sysid+0(1) EQ 'D'.           " Dev system  
    cv_changeable = abap_true.  
    RETURN.  
  ENDIF.  
  *--Import the expression type  
  cl_fdt_factory=>get_id_information( EXPORTING iv_id = iv_id  
                                       IMPORTING ev_expression_type_id = l_expression_type_id ).  
  
  *--If expression type is decision table  
  if l_expression_type_id EQ if_fdt_constants=>gc_ex_ty_decision_table.  
    cv_changeable = abap_true.  
  ELSE.  
    cv_changeable = abap_false.  
  ENDIF.
```

ENDMETHOD.

## 11. ADDITIONAL INFORMATION

### 11.1 Further Reading

#### 11.1.1 Information on SAP MDG on SAP S/4HANA

- Exchange knowledge: [SAP Community](#) | [Q&A](#) | [Blog](#)
- Try SAP Master Data Governance on S/4HANA for free: [Trial Version](#)
- Try SAP Master Data Governance on S/4HANA on the SAP Cloud Appliance Library: [S/4HANA 2022 FPS1](#)
- Learn more: [Latest Release](#) | [Help Portal](#) | [How-to Information](#) | [Key Presentations](#)

#### 11.1.2 SAP Roadmap Explorer

- Please see the [roadmap for SAP Master Data Governance](#)

#### 11.1.3 Related Information

- Learn more: [Floorplan Manager for Web Dynpro ABAP](#) | [How to Adapt FPM](#) | [FPM Blog](#) | [How-to Information](#) | [Service Mapping Tool](#) | [SAP S/4HANA Cookbook CVI](#)

### 11.2 SAP Notes

In addition to the detailed explanations written in this document, please see the following SAP Notes for further important information.

Note	Description
<a href="#">3372801</a>	Upgrade or Conversion for Master Data Governance, Central Governance
<a href="#">3043582</a>	MDG Customer Connection 2020
<a href="#">3194967</a>	MDG Customer Connection 2021 for S/4HANA 2022
<a href="#">3311039</a>	MDG Customer Connection 2023
<a href="#">3428179</a>	Master Data Governance: Continuous Influence
<a href="#">3134600</a>	MDG-M: Supported fields in Data Model MM
<a href="#">1806108</a>	Functional restrictions in MDG-M in MDG7 (incl. SP02)
<a href="#">2129261</a>	Functional restrictions in MDG-M in MDG8
<a href="#">2284745</a>	Functional Restrictions in MDG for Material with SAP Master Data Governance 9.0
<a href="#">2461516</a>	Functional Restrictions in MDG for Material with SAP Master Data Governance 9.1
<a href="#">2656693</a>	Functional Restrictions in MDG for Material in SAP Master Data Governance 9.2 and on SAP S/4HANA 1809
<a href="#">2816571</a>	Functional Restrictions in MDG for Material on SAP S/4HANA 1909
<a href="#">2948873</a>	Functional Restrictions in MDG for Material on SAP S/4HANA 2020
<a href="#">3070012</a>	Functional Restrictions in MDG for Material on SAP S/4HANA 2021
<a href="#">3219945</a>	Functional Restrictions in MDG for Material on SAP S/4HANA 2022
<a href="#">3374998</a>	Functional Restrictions in MDG for Material on SAP S/4HANA 2023

<a href="#"><u>2950412</u></a>	Functional restrictions in MDG Process Analytics on SAP S/4HANA 2020
<a href="#"><u>3066855</u></a>	Functional restrictions in MDG Process Analytics on SAP S/4HANA 2021
<a href="#"><u>3225098</u></a>	Functional restrictions in MDG Process Analytics on SAP S/4HANA 2022
<a href="#"><u>3381795</u></a>	Functional restrictions in MDG Process Analytics on SAP S/4HANA 2022
<a href="#"><u>2479869</u></a>	Usage of Lean Classification with SAP Master Data Governance
<a href="#"><u>1619534</u></a>	How to Create, Enhance and Adapt FPM Applications
<a href="#"><u>1637249</u></a>	MDG: Information for efficient message processing
<a href="#"><u>2105467</u></a>	MDG Performance
<a href="#"><u>2561461</u></a>	Scope of support for SAP Master Data Governance (MDG)
<a href="#"><u>1637249</u></a>	MDG: Information for efficient message processing