Reference Data Management 2204

for SAP Master Data Governance

Technical Documentation

MDF Configuration Management

Itego

Trusted data. Delivered.

Version: 18.05.2022

Content

1 Introduction: Master Data Framework 3

2 Introduction: MDF Configuration Management 4

3 Components of the MDF Configuration Management 4

3.1.1 Field Properties 6

3.1.2 Validations 7

3.1.3 Derivations 11

3.1.4 Search Configuration 14

4 Example 15

5 Technical Information 18

5.1 BAdI Implementations 18

5.2 Configuration Tables 21

1 Introduction: Master Data Framework

The Itego Master Data Framework (MDF) builds the foundation for Itego Reference Data Management (RDM) and covers the following components

- MDF for SAP MDG
    - Reference Data Processing
    - Standard Enhancements
    - Customer Specific Objects
    - Configuration Management
    - Generic Data Replication
- MDF for SAP ERP and S/4HANA
    - Local Staging Area
    - Generic Data Replication
- MDF for non-SAP
    - WebService Connect (planned / PoC version available)

2 Introduction: MDF Configuration Management

Using the MDF Configuration Management, a user is able to configure user interfaces, check data and derive values based on business rules. The main activities in this component are:

- Field Properties: Define fields as optional, mandatory or hidden
- Validations: Validate user input
- Derivations: Derive values for input fields
- Default Values: Set default values for input fields
- Search Configuration: Configure search attributes and the result list

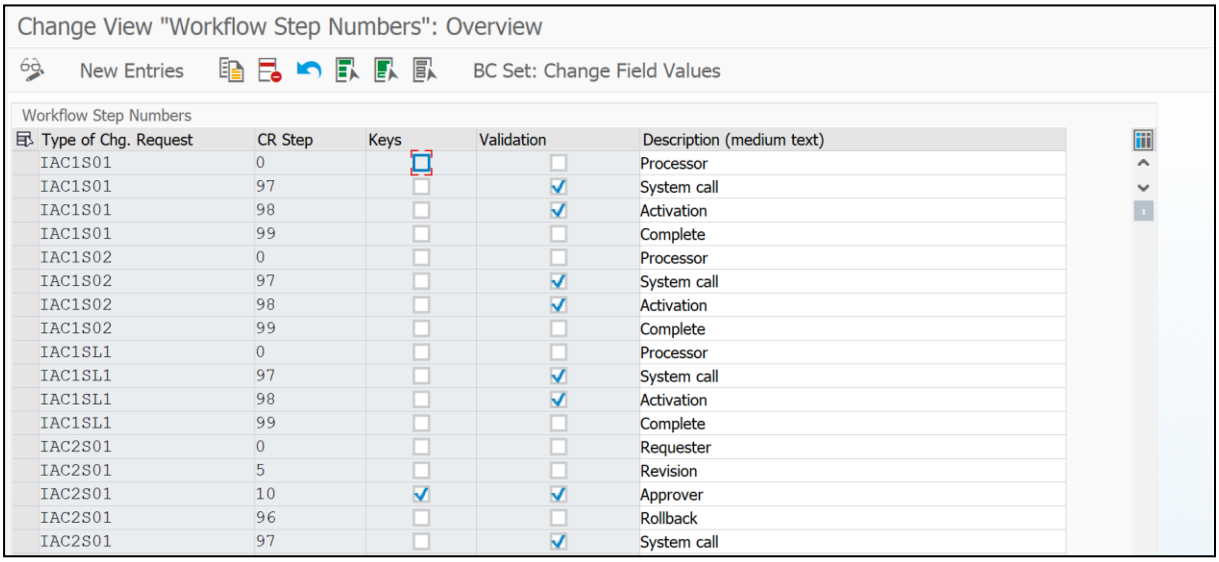
These functionalities are generic and therefore available for every object in RDM. They will be explained in the following sections.

Note: The MDF Configuration Management works based on the validation framework which is provided by SAP MDG. This especially means that these configurations are activated by the activation of Business Add-Ins (BAdIs). Please check chapter 5.1 "BAdI Implementations" for additional information.

3 Components of the MDF Configuration Management

Before you can get started with configuration activities, you need to acquire information about available change request types and the IDs of the related user interfaces.

To find out which change requests are available within your system, execute transaction MDGIMG and expand the following nodes: "General Settings → Process Modelling → Workflow → Rule Based Workflow → Define Change Request Steps for Rule-Based Workflow" and execute it. Here you can identify which change request types and which steps are available:

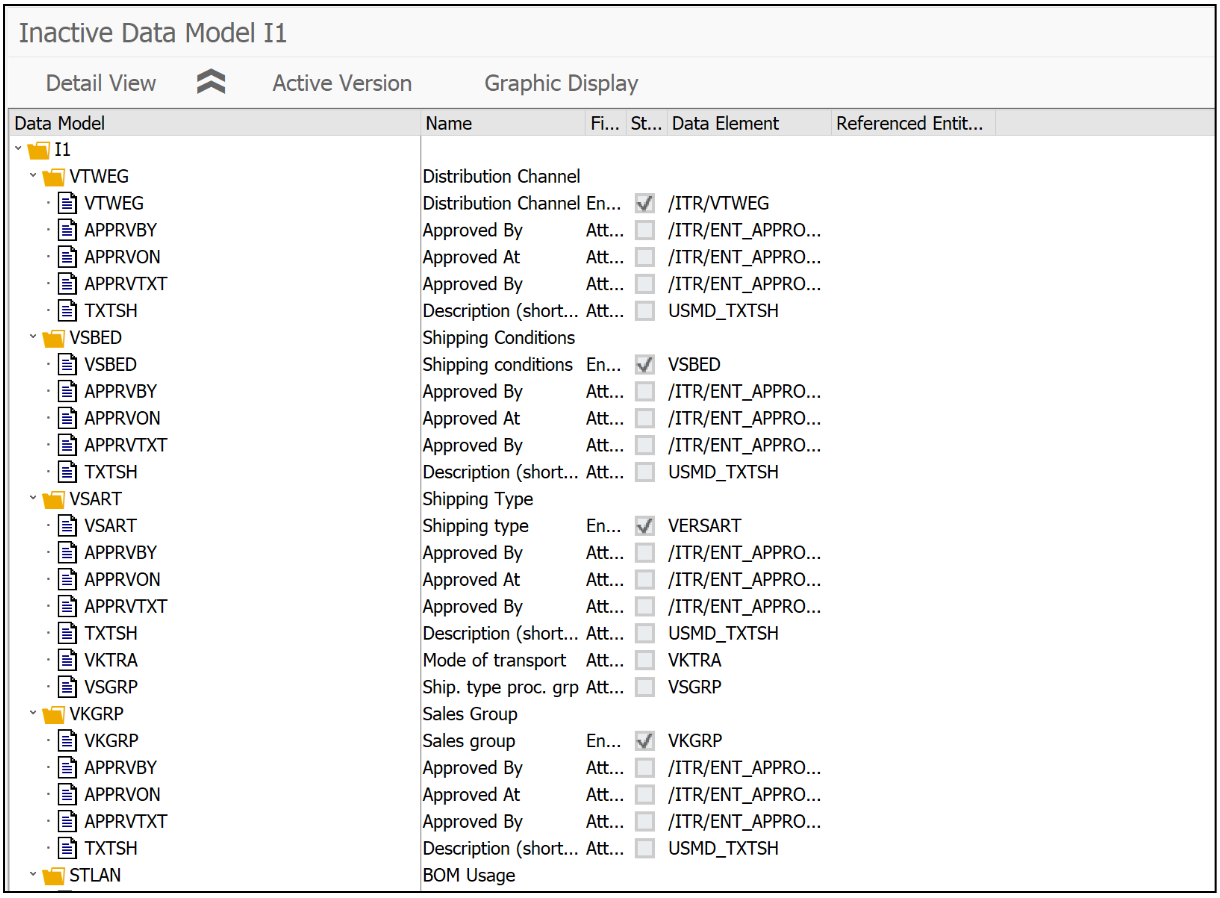


Change View "Workflow Step Numbers": Overview

Workflow Step Numbers

Type of Chg. Request	CR Step	Keys	Validation	Description (medium text)
IAC1S01	0	□	□	Processor
IAC1S01	97	□	☑	System call
IAC1S01	98	□	☑	Activation
IAC1S01	99	□	□	Complete
IAC1S02	0	□	□	Processor
IAC1S02	97	□	☑	System call
IAC1S02	98	□	☑	Activation
IAC1S02	99	□	□	Complete
IAC1SL1	0	□	□	Processor
IAC1SL1	97	□	☑	System call
IAC1SL1	98	□	☑	Activation
IAC1SL1	99	□	□	Complete
IAC2S01	0	□	□	Requester
IAC2S01	5	□	□	Revision
IAC2S01	10	☑	☑	Approver
IAC2S01	96	□	□	Rollback
IAC2S01	97	□	☑	System call

You also need to acquire some information about the input fields used in these change request types. This can be done by executing the transaction MDGIMG and expanding the path “General Settings → Data Modeling” and executing “Edit Data Model”. In the next screen, select data model I1 and press on “Visualize Data Model”. You will see a list of available entity types including fields:



Inactive Data Model I1

Data Model	Name	Fi...	St...	Data Element	Referenced Entit...
I1					
VTWEG	Distribution Channel				
VTWEG	Distribution Channel	En...	☑	/ITR/VTWEG	
APPRVBY	Approved By	Att...	□	/ITR/ENT_APPRO...	
APPRVON	Approved At	Att...	□	/ITR/ENT_APPRO...	
APPRVTXT	Approved By	Att...	□	/ITR/ENT_APPRO...	
TXTSH	Description (short...	Att...	□	USMD_TXTSH	
VSBED	Shipping Conditions				
VSBED	Shipping conditions	En...	☑	VSBED	
APPRVBY	Approved By	Att...	□	/ITR/ENT_APPRO...	
APPRVON	Approved At	Att...	□	/ITR/ENT_APPRO...	
APPRVTXT	Approved By	Att...	□	/ITR/ENT_APPRO...	
TXTSH	Description (short...	Att...	□	USMD_TXTSH	
VSART	Shipping Type				
VSART	Shipping type	En...	☑	VERSART	
APPRVBY	Approved By	Att...	□	/ITR/ENT_APPRO...	
APPRVON	Approved At	Att...	□	/ITR/ENT_APPRO...	
APPRVTXT	Approved By	Att...	□	/ITR/ENT_APPRO...	
TXTSH	Description (short...	Att...	□	USMD_TXTSH	
VKTRA	Mode of transport	Att...	□	VKTRA	
VSGRP	Ship. type proc. grp	Att...	□	VSGRP	
VKGRP	Sales Group				
VKGRP	Sales group	En...	☑	VKGRP	
APPRVBY	Approved By	Att...	□	/ITR/ENT_APPRO...	
APPRVON	Approved At	Att...	□	/ITR/ENT_APPRO...	
APPRVTXT	Approved By	Att...	□	/ITR/ENT_APPRO...	
TXTSH	Description (short...	Att...	□	USMD_TXTSH	
STLAN	BOM Usage				

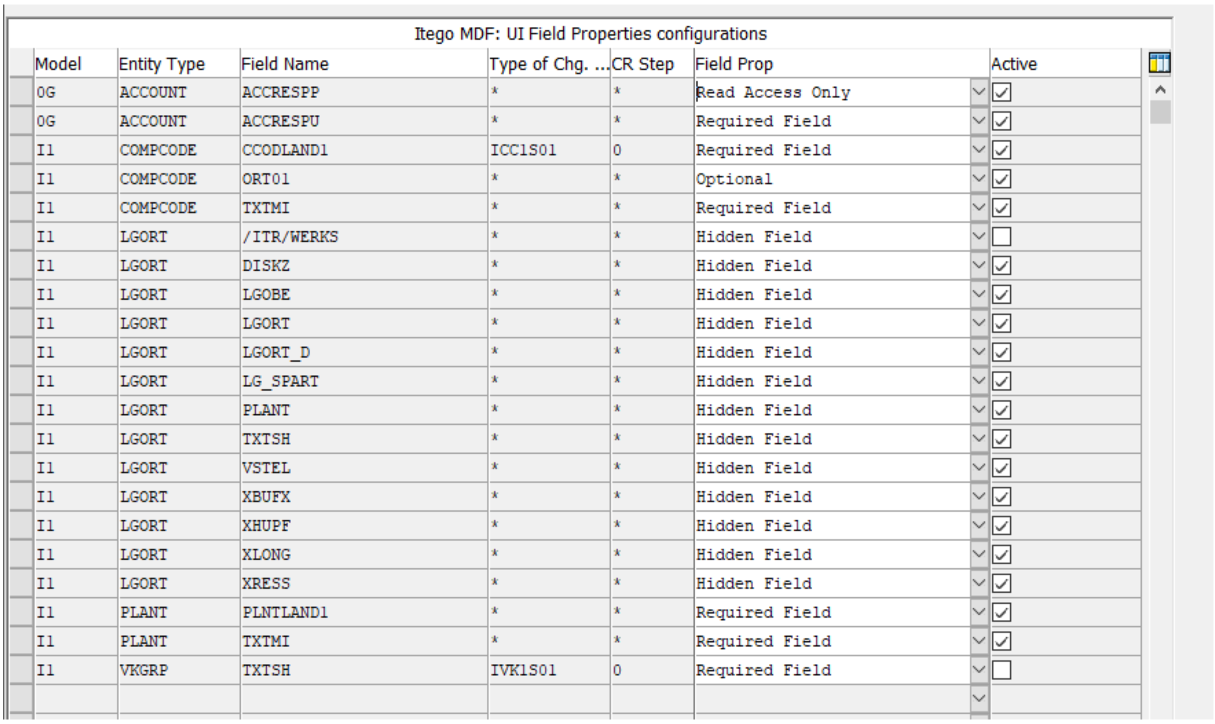
Based on this, configurations can be carried out using transaction /ITU/MDFIMG.

3.1.1 Field Properties

The Itego Master Data Framework includes the functionality of configuring input fields for the end user in order to guide his or her input, by marking fields as

- Required
- Hidden
- Optional
- Read Only

To configure the properties of a field, execute transaction /ITU/MDFIMG. Expand node “UI Field Properties” and execute the entry “Configure Field properties”. The following table will show up (entries may differ and note that currently Data Models other than I1 are not supported):



Itego MDF: UI Field Properties configurations

Model	Entity Type	Field Name	Type of Chg. ...	CR Step	Field Prop	Active
0G	ACCOUNT	ACCRESPP	*	*	Read Access Only	☑
0G	ACCOUNT	ACCRESPU	*	*	Required Field	☑
I1	COMPCODE	CCODLAND1	ICC1S01	0	Required Field	☑
I1	COMPCODE	ORT01	*	*	Optional	☑
I1	COMPCODE	TXTMI	*	*	Required Field	☑
I1	LGORT	/ITR/WERKS	*	*	Hidden Field	☐
I1	LGORT	DISKZ	*	*	Hidden Field	☑
I1	LGORT	LGOBE	*	*	Hidden Field	☑
I1	LGORT	LGORT	*	*	Hidden Field	☑
I1	LGORT	LGORT_D	*	*	Hidden Field	☑
I1	LGORT	LG_SPART	*	*	Hidden Field	☑
I1	LGORT	PLANT	*	*	Hidden Field	☑
I1	LGORT	TXTSH	*	*	Hidden Field	☑
I1	LGORT	VSTEL	*	*	Hidden Field	☑
I1	LGORT	XBUFX	*	*	Hidden Field	☑
I1	LGORT	XHUPF	*	*	Hidden Field	☑
I1	LGORT	XLONG	*	*	Hidden Field	☑
I1	LGORT	XRESS	*	*	Hidden Field	☑
I1	PLANT	PLNTLAND1	*	*	Required Field	☑
I1	PLANT	TXTMI	*	*	Required Field	☑
I1	VKGRP	TXTSH	IVK1S01	0	Required Field	☐

If you now want to maintain a new field property, you need to add an entry. Press on "New Entries" and fill out the following fields:

- Data Model: "I1" (for reference data objects)
- Entity Type: The entity type of the object of the field, you want to configure
- Type of Chg. Request: The type of change request, that is supposed to be affected by the rule (or insert "*" for all change request types)
- CR Step: The change request step, that is supposed to be affected by the rule (or insert "*" for all change request steps)
- Field Name: The ID of the field that is supposed to be affected
- Field Property: You get to choose between several options on how the field will behave:
    - Optional: The field is optional, no entry necessary
    - Read Access Only: Read only, no entry possible
    - Required Field: The field needs to be maintained
    - Hidden Field: The field is not visible and cannot maintained
- Active: Select if the configuration should be active, otherwise it is not

3.1.2 Validations

The Itego Master Data Framework offers the functionality of validating user input. For each field, you are able to provide a value or a range of values, which is valid. If the user proceeds to enter an invalid value, further processing will not be possible.

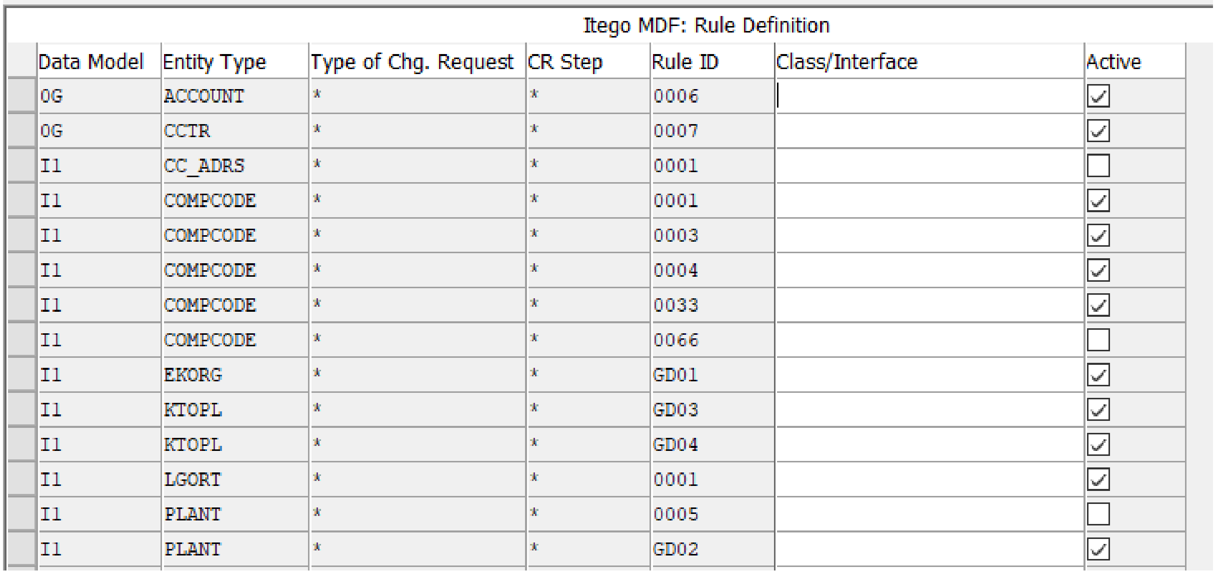
To configure the validation, execute transaction /ITU/MDFIMG and expand the “Rule Service Configuration” node. For maintaining a validation rule, you need to maintain the entries:

- Rule Definition
- Rule Type Definition

If the rules should also depend on other fields, you might need to maintain the entries:

- Define Conditional Fields for Rules
- Define Values for Conditional Rules

Start by maintaining the “Rule Definition” activity. Once you press on it, the following table shows up (entries may differ):



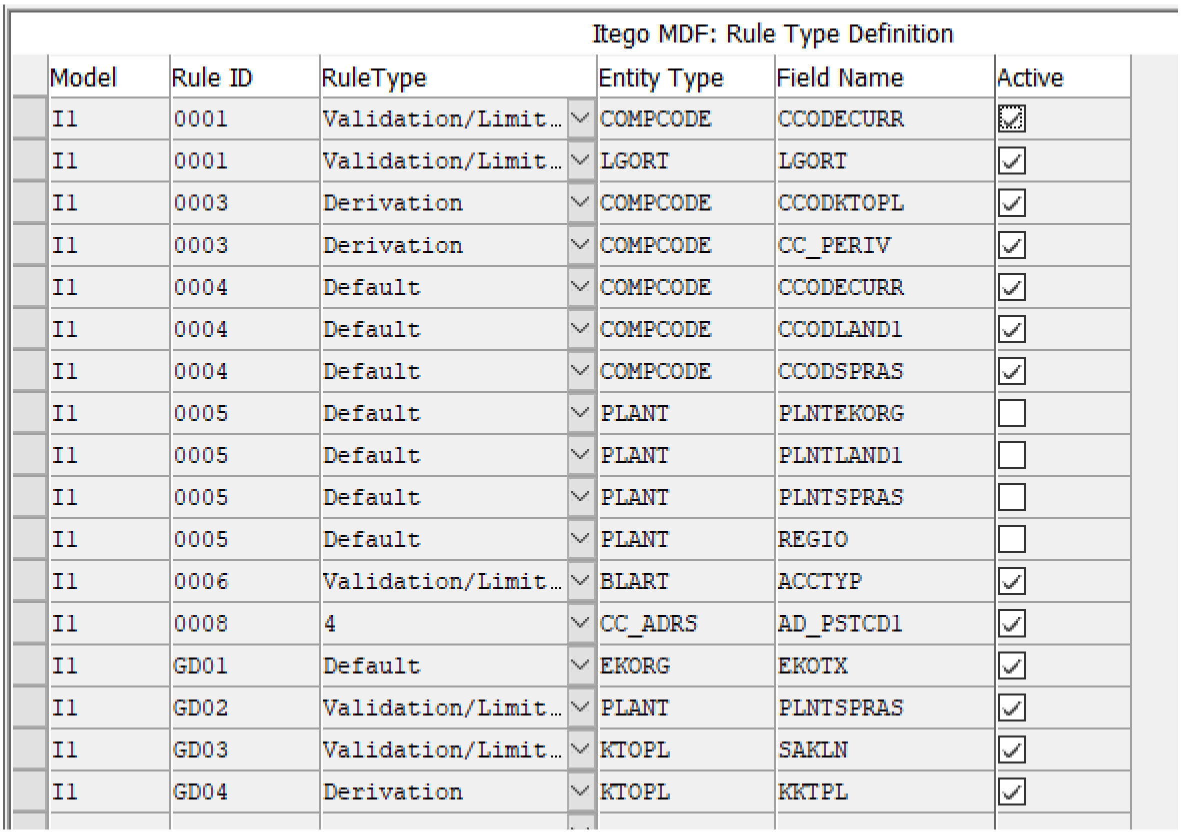
Itego MDF: Rule Definition

Data Model	Entity Type	Type of Chg. Request	CR Step	Rule ID	Class/Interface	Active
0G	ACCOUNT	*	*	0006		☑
0G	CCTR	*	*	0007		☑
I1	CC_ADRS	*	*	0001		□
I1	COMPCODE	*	*	0001		☑
I1	COMPCODE	*	*	0003		☑
I1	COMPCODE	*	*	0004		☑
I1	COMPCODE	*	*	0033		☑
I1	COMPCODE	*	*	0066		□
I1	EKORG	*	*	GD01		☑
I1	KTOPL	*	*	GD03		☑
I1	KTOPL	*	*	GD04		☑
I1	LGORT	*	*	0001		☑
I1	PLANT	*	*	0005		□
I1	PLANT	*	*	GD02		☑

Press on “New Entries” and maintain:

- Data Model: “I1” (for reference data objects)
- Entity Type: ID of the entity type, that is supposed to be affected.
- Type of Chg. Request: ID of the change request, that is supposed to be affected. Or insert “*” for all change requests.
- Step: ID of the change request step, that is supposed to be affected. Or insert “*” for all change request steps.
- Rule ID: A 4-digit ID, which is used for identifying the rule later in the process. You can either use a new ID or one that is already in use.
- Class/Interface: you usually leave this empty
- Active: Check if you want the rule to be active

Save your entries and navigate back to the IMG node, then execute the activity "Rule Type Definition". The following table shows up (entries may differ):



Itego MDF: Rule Type Definition

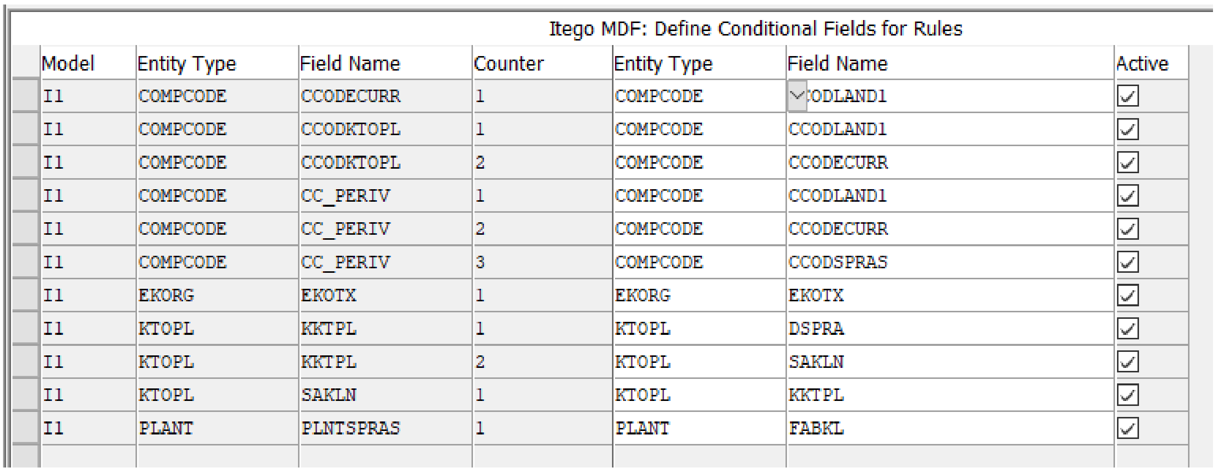
Model	Rule ID	RuleType	Entity Type	Field Name	Active
I1	0001	Validation/Limit...	COMPCODE	CCODECURR	☑
I1	0001	Validation/Limit...	LGORT	LGORT	☑
I1	0003	Derivation	COMPCODE	CCODKTOPL	☑
I1	0003	Derivation	COMPCODE	CC_PERIV	☑
I1	0004	Default	COMPCODE	CCODECURR	☑
I1	0004	Default	COMPCODE	CCODLAND1	☑
I1	0004	Default	COMPCODE	CCODSPRAS	☑
I1	0005	Default	PLANT	PLNTEKORG	□
I1	0005	Default	PLANT	PLNTLAND1	□
I1	0005	Default	PLANT	PLNTSPRAS	□
I1	0005	Default	PLANT	REGIO	□
I1	0006	Validation/Limit...	BLART	ACCTYP	☑
I1	0008	4	CC_ADRS	AD_PSTCD1	☑
I1	GD01	Default	EKORG	EKOTX	☑
I1	GD02	Validation/Limit...	PLANT	PLNTSPRAS	☑
I1	GD03	Validation/Limit...	KTOPL	SAKLN	☑
I1	GD04	Derivation	KTOPL	KKTPL	☑

Press on "New Entries" and maintain the input fields accordingly:

- Data Model: "I1" (for reference data objects)
- Rule ID: The Rule ID used in Step 1
- Rule Type: "Validation/Limitation"
- Entity Type: ID of the entity type that is supposed to be affected
- Field Name: ID of the field that is supposed to be affected
- Active: Check if you want the rule to be active

Save your entries and navigate back to the IMG node.

For conditional rule execute activity "Define Conditional Fields for Rules". The following table shows up (entries may differ):



Itego MDF: Define Conditional Fields for Rules

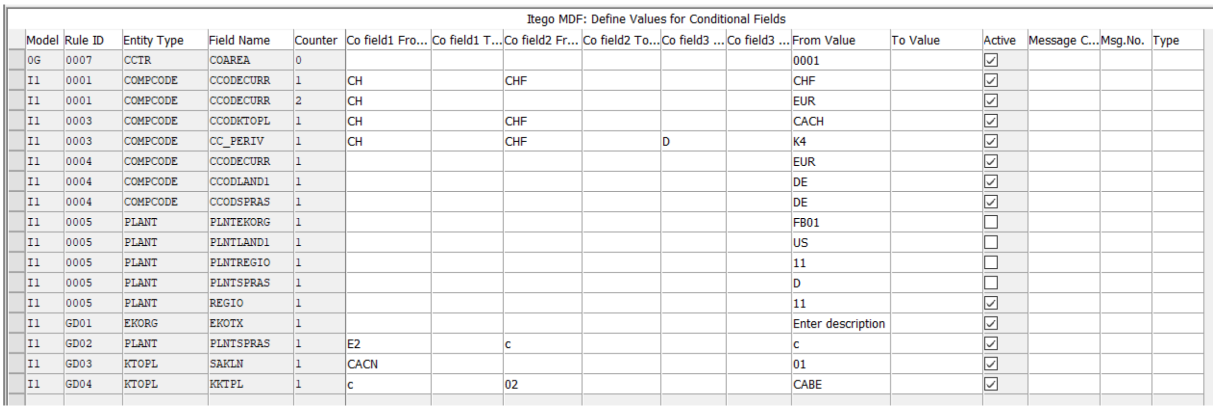
Model	Entity Type	Field Name	Counter	Entity Type	Field Name	Active
I1	COMPCODE	CCODECURR	1	COMPCODE	CODLAND1	☑
I1	COMPCODE	CCODKTOPL	1	COMPCODE	CCODLAND1	☑
I1	COMPCODE	CCODKTOPL	2	COMPCODE	CCODECURR	☑
I1	COMPCODE	CC_PERIV	1	COMPCODE	CCODLAND1	☑
I1	COMPCODE	CC_PERIV	2	COMPCODE	CCODECURR	☑
I1	COMPCODE	CC_PERIV	3	COMPCODE	CCODSPRAS	☑
I1	EKORG	EKOTX	1	EKORG	EKOTX	☑
I1	KTOPL	KKTPL	1	KTOPL	DSPRA	☑
I1	KTOPL	KKTPL	2	KTOPL	SAKLN	☑
I1	KTOPL	SAKLN	1	KTOPL	KKTPL	☑
I1	PLANT	PLNTSPRAS	1	PLANT	FABKL	☑

In this table, you can define fields, on which the validation of the input field, entered in the previous step, will depend on. E.g. if a company code is only allowed with currency “Euro” if the country Germany is entered, you would enter the country-field and the currency-field. Press on “New Entries” and maintain the table fields accordingly:

- Data Model: “I1” (for reference data objects)
- Entity Type: ID of the dependent entity type
- Field Name: ID of the dependent field
- Counter: Limitations can depend on more than one field. If you want to use this functionality, make sure, you assign each conditional field a different number, starting with “1”
- Entity Type: ID of the conditional entity type (which is used as a dependency)
- Field Name: ID of the conditional field (which is used as a dependency)
- Active: Check if you want the rule to be active

To define multiple conditional fields for a dependent field, simply add another entry to this table and increment the “Counter” for each new entry.

Save your entries and navigate back to the IMG node, then execute the entry “Define Conditional Rules”. The following table shows up (entries may differ):



Itego MDF: Define Values for Conditional Fields

Model	Rule ID	Entity Type	Field Name	Counter	Co field1 Fro...	Co field1 T...	Co field2 Fr...	Co field2 To...	Co field3 ...	Co field3 ...	From Value	To Value	Active	Message C...	Msg.No.	Type
0G	0007	CCTR	COAREA	0							0001		☑			
I1	0001	COMPCODE	CCODECURR	1	CH		CHF				CHF		☑			
I1	0001	COMPCODE	CCODECURR	2	CH						EUR		☑			
I1	0003	COMPCODE	CCODKTOPL	1	CH		CHF				CACH		☑			
I1	0003	COMPCODE	CC_PERIV	1	CH		CHF		D		K4		☑			
I1	0004	COMPCODE	CCODECURR	1							EUR		☑			
I1	0004	COMPCODE	CCODLAND1	1							DE		☑			
I1	0004	COMPCODE	CCODSPRAS	1							DE		☑			
I1	0005	PLANT	PLNTEKORG	1							FB01		□			
I1	0005	PLANT	PLNTLAND1	1							US		□			
I1	0005	PLANT	PLNTREGIO	1							11		□			
I1	0005	PLANT	PLNTSPRAS	1							D		□			
I1	0005	PLANT	REGIO	1							11		☑			
I1	GD01	EKORG	EKOTX	1							Enter description		☑			
I1	GD02	PLANT	PLNTSPRAS	1	E2		c				c		☑			
I1	GD03	KTOPL	SAKLN	1	CACN						01		☑			
I1	GD04	KTOPL	KKTPL	1	c		02				CABE		☑			

In this table, the valid values for the dependent field entered in “Define Conditional Fields for Rules” are defined:

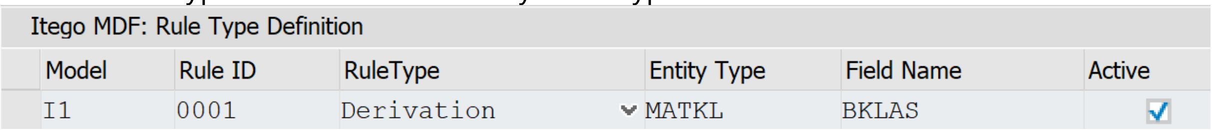
- Data Model: “I1” (for reference data objects)
- Rule ID: ID, defined/used in “Rule Definition”
- Entity Type: ID of the dependent entity type
- Field Name: ID of the dependent field
- Counter: Multiple conditions can be defined. Each condition needs its own unique number (start with “1”)
- Co field[n] From/To value: Enter a value or a range of values, for conditional field [n]. If one of these values is entered in the input field by the user, the validation rule will be in executed
- From Value/To Value: Define a value or a range of values that is valid for the dependent field
- Active: Check if you want the rule to be active

3.1.3 Derivations

The Itego Master Data Framework offers the functionality of deriving field values based on user input. E.g. if a user enters the Division “01” in a Material Group creation process, the Valuation Class “0710” could be derived from the user input and will be filled automatically.

To set up such a derivation rule, execute transaction /ITU/MDFIMG and maintain your configuration as described in section “Validation”, but:

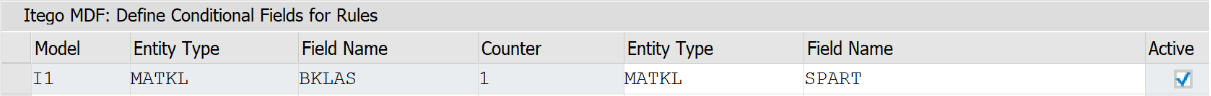
Choose Rule Type: “Derivation” in activity “Rule Type Definition”



Itego MDF: Rule Type Definition

Model	Rule ID	RuleType	Entity Type	Field Name	Active
I1	0001	Derivation	MATKL	BKLAS	☑

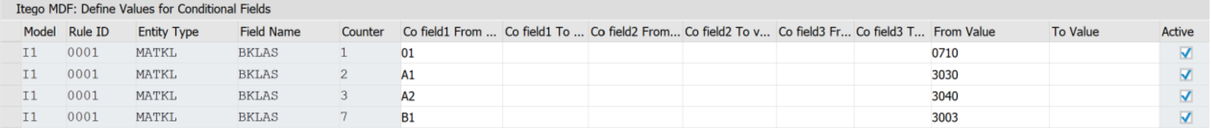
In activity “Define Conditional Rules” you are able to define the field which is the source field for the derivation. In the example below “SPART” (Division) is the source for the derived field “BKLAS” (Valuation Class).



Itego MDF: Define Conditional Fields for Rules

Model	Entity Type	Field Name	Counter	Entity Type	Field Name	Active
I1	MATKL	BKLAS	1	MATKL	SPART	☑

In activity “Define Values for Conditional Fields” you can set the value for the derived field in “From value”. All options should have a value, in case of changing to another option, which means that when using Derivation, you have to maintain values for every possible option.



Itego MDF: Define Values for Conditional Fields

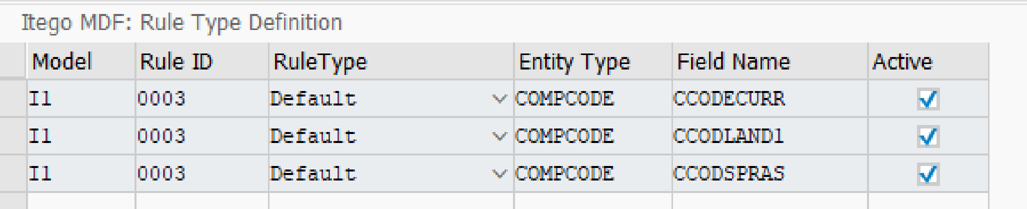
Model	Rule ID	Entity Type	Field Name	Counter	Co field1 From ...	Co field1 To ...	Co field2 From...	Co field2 To v...	Co field3 Fr...	Co field3 T...	From Value	To Value	Active
I1	0001	MATKL	BKLAS	1	01						0710		☑
I1	0001	MATKL	BKLAS	2	A1						3030		☑
I1	0001	MATKL	BKLAS	3	A2						3040		☑
I1	0001	MATKL	BKLAS	7	B1						3003		☑

3.1.3.1 *Default Values*

The Itego Master Data Framework offers the functionality of assigning default values to input fields. E.g. if a user creates a new Company Code, the currency field could already be filled out with “EUR”.

To set up such a default value, execute transaction /ITU/MDFIMG and maintain your configuration as described in section “Validation”, but:

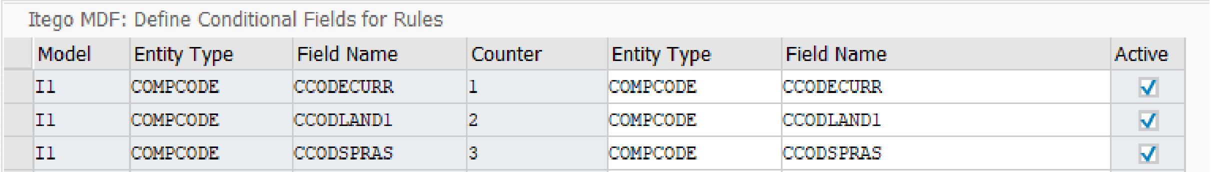
Choose Rule Type: “Default” in activity “Rule Type Definition”



Itego MDF: Rule Type Definition

Model	Rule ID	RuleType	Entity Type	Field Name	Active
I1	0003	Default	COMPCODE	CCODECURR	☑
I1	0003	Default	COMPCODE	CCODLAND1	☑
I1	0003	Default	COMPCODE	CCODSPRAS	☑

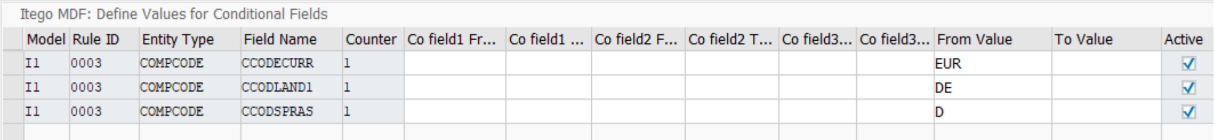
In activity “Define Conditional Fields for Rules” maintain the field which should be defaulted.



Itego MDF: Define Conditional Fields for Rules

Model	Entity Type	Field Name	Counter	Entity Type	Field Name	Active
I1	COMPCODE	CCODECURR	1	COMPCODE	CCODECURR	☑
I1	COMPCODE	CCODLAND1	2	COMPCODE	CCODLAND1	☑
I1	COMPCODE	CCODSPRAS	3	COMPCODE	CCODSPRAS	☑

After this define in activity “Define Values for Conditional Fields” the default value without specifying conditional fields in “From value”.



Itego MDF: Define Values for Conditional Fields

Model	Rule ID	Entity Type	Field Name	Counter	Co field1 Fr...	Co field1 ...	Co field2 F...	Co field2 T...	Co field3...	Co field3...	From Value	To Value	Active
I1	0003	COMPCODE	CCODECURR	1							EUR		☑
I1	0003	COMPCODE	CCODLAND1	1							DE		☑
I1	0003	COMPCODE	CCODSPRAS	1							D		☑

3.1.4 Search Configuration

The Itego Master Data Framework offers the functionality to configure the reference data search. Execute transaction /ITU/MDFIMG, expand the node "Search UI Configuration" and execute activity "Configure Search Criteria and Result List".

Please maintain the following entries:

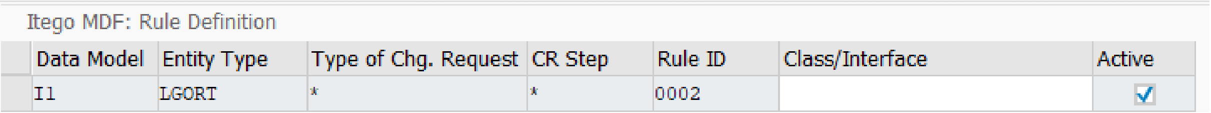
- Data Model: "I1" (for reference data objects)
- Entity Type: ID of the entity type
- Field Name: ID of the field
- Active: Check if you would like to add this field to the search criteria and the result list
- No: Specify the order of the fields by assigning a sequence (start with "1")

4 Example

This example shows the dependency between the field Shipping Point and Plant in the object Storage Location. When the value of the Plant field is set as “0001“, the field Shipping point will automatically be filled with the value “0001” and no other entry is possible.

First open “Rule Definition”. Press new entries.
Please maintain:

- Data Model: “I1”
- Entity Type: LGORT (Storage Location)
- Type of Chg. Request: “*”
- Step: “*”
- Rule ID: 0002 (since there already is a Rule 0001 for LGORT)
- Class/Interface: you usually leave this empty
- Active: Check for the rule to be active

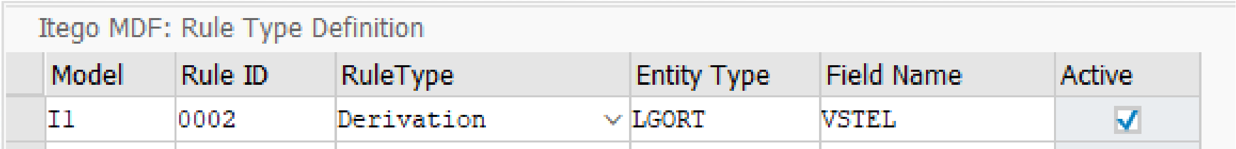


Itego MDF: Rule Definition

Data Model	Entity Type	Type of Chg. Request	CR Step	Rule ID	Class/Interface	Active
I1	LGORT	*	*	0002		☑

Then open “Rule Type Definition”. Press new entries.
Please maintain

- Data Model: “I1”
- Rule ID: 0002, since we set this as the rule one step before
- Rule Type: “3” (Derivation: When the field Plant is filled with a specific value then Shipping Location is automatically set)
- Entity Type: LGORT (Storage Location, this is the dependent entity type)
- Field Name: VSTEL (Shipping Location, this is the dependent field, it will be filled automatically with a specific value when PLANT is filled)
- Active: Check for the rule to be active



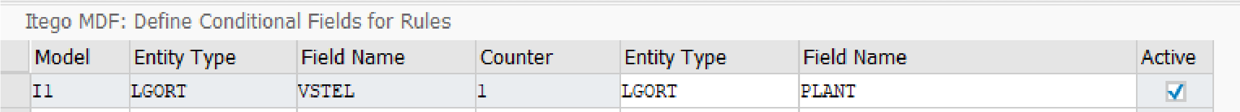
Itego MDF: Rule Type Definition

Model	Rule ID	RuleType	Entity Type	Field Name	Active
I1	0002	Derivation	LGORT	VSTEL	☑

Open “Define Conditional Fields for Rules”. Press new entries.
Please maintain:

- Data Model: “I1”
- Entity Type: LGORT (the dependent entity type)
- Field Name: VSTEL (the dependent field)
- Counter: “1”, it’s only depended on one field.

- Entity Type: LGORT (the conditional entity type)
- Field Name: PLANT (the conditional field)
- Active: Check for the rule to be active

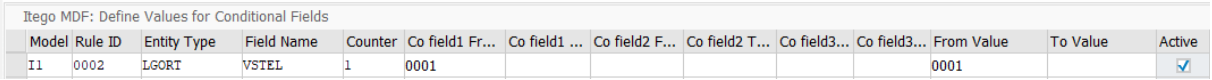


Itego MDF: Define Conditional Fields for Rules

Model	Entity Type	Field Name	Counter	Entity Type	Field Name	Active
I1	LGORT	VSTEL	1	LGORT	PLANT	☑

Open “Define Conditional Fields for Rules”. Press new entries.
Please maintain:

- Data Model: “I1”
- Rule ID: 0002
- Entity Type: LGORT (the dependent entity type)
- Field Name: VSTEL (the dependent field)
- Counter: “1”, there is only one conditional field
- Co field[n] From/To value: 0001 (when this value is used in PLANT, the value for VSTEL will be set automatically)
- From Value/To Value: 0001 (The dependent field VSTEL will be filled with this value)
- Active: Check for the rule to be active

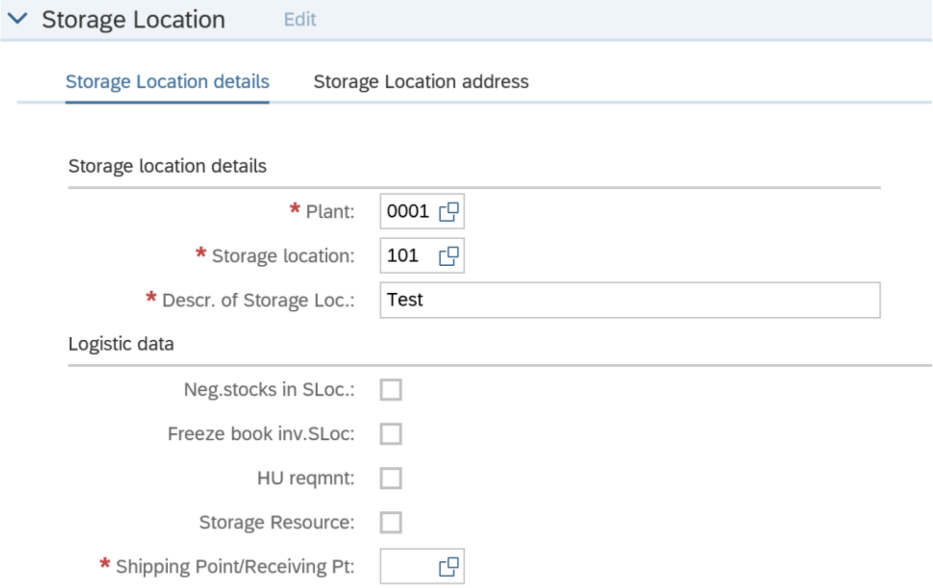


Itego MDF: Define Values for Conditional Fields

Model	Rule ID	Entity Type	Field Name	Counter	Co field1 Fr...	Co field1 ...	Co field2 F...	Co field2 T...	Co field3...	Co field3...	From Value	To Value	Active
I1	0002	LGORT	VSTEL	1	0001						0001		☑

Now create a new Storage Location.

Fill every field with values. Choose 0001 for Plant.



Storage Location Edit

Storage Location details | Storage Location address

Storage location details

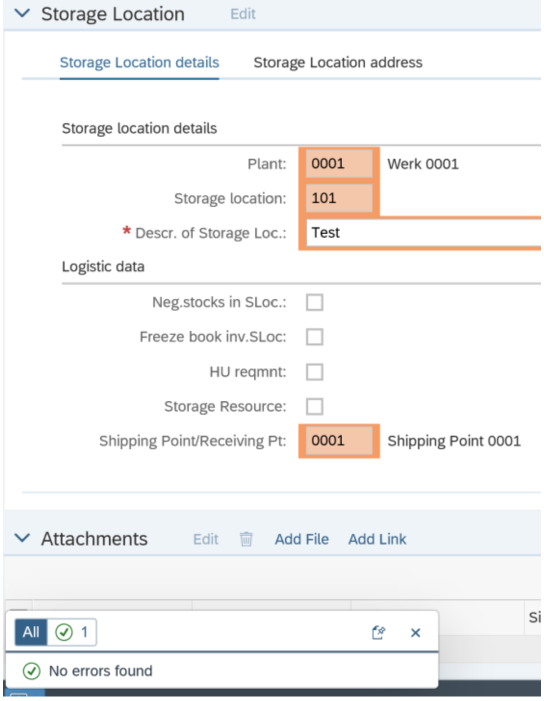
* Plant:	0001
* Storage location:	101
* Descr. of Storage Loc.:	Test

Logistic data

Neg.stocks in SLoc.:	□
Freeze book inv.SLoc:	□
HU reqmnt:	□
Storage Resource:	□
* Shipping Point/Receiving Pt:	

Click “Check”.

The field Shipping location is now filled automatically with the value “0001”.



Storage Location Edit

Storage Location details | Storage Location address

Storage location details

Plant:	0001	Werk 0001
Storage location:	101	
* Descr. of Storage Loc.:	Test	

Logistic data

Neg.stocks in SLoc.:	□	
Freeze book inv.SLoc:	□	
HU reqmnt:	□	
Storage Resource:	□	
Shipping Point/Receiving Pt:	0001	Shipping Point 0001

Attachments Edit Add File Add Link

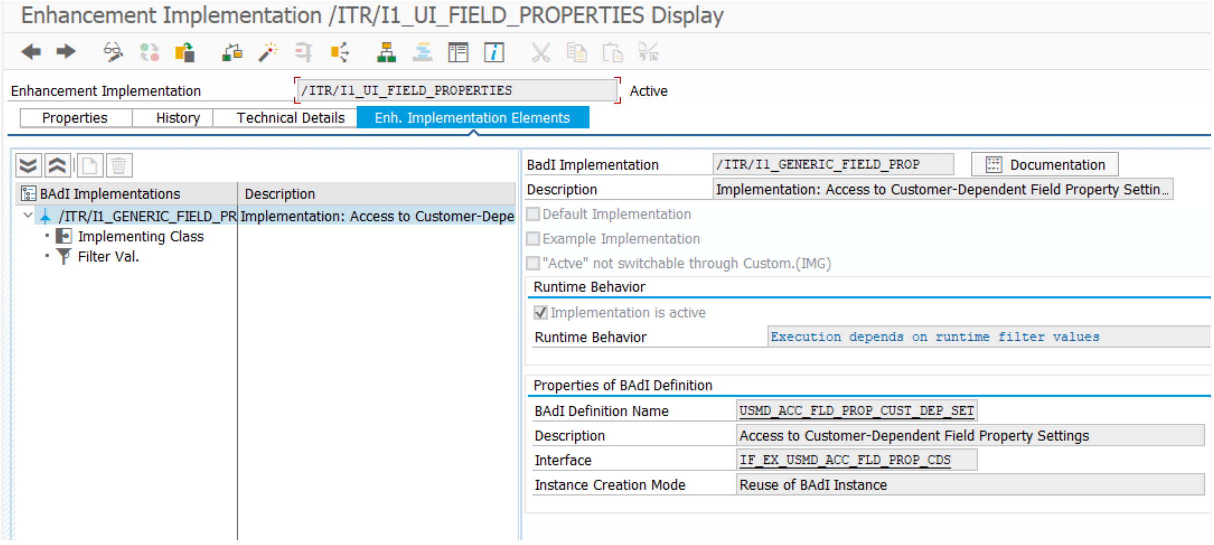
All ✓ 1

✓ No errors found

5 Technical Information

5.1 BAdI Implementations

For field properties the Enhancement Spot USMD_ACC_FLD_PROP_CUST_DEP_SET needs to be active:



Enhancement Implementation /ITR/I1_UI_FIELD_PROPERTIES Display

Enhancement Implementation /ITR/I1_UI_FIELD_PROPERTIES Active

BAdI Implementations	Description
/ITR/I1_GENERIC_FIELD_PR	Implementation: Access to Customer-Depe
Implementing Class	
Filter Val.	

BadI Implementation: /ITR/I1_GENERIC_FIELD_PROP

Description: Implementation: Access to Customer-Dependent Field Property Settin…

□ Default Implementation

□ Example Implementation

□ "Actve" not switchable through Custom.(IMG)

Runtime Behavior

☑ Implementation is active

Runtime Behavior: Execution depends on runtime filter values

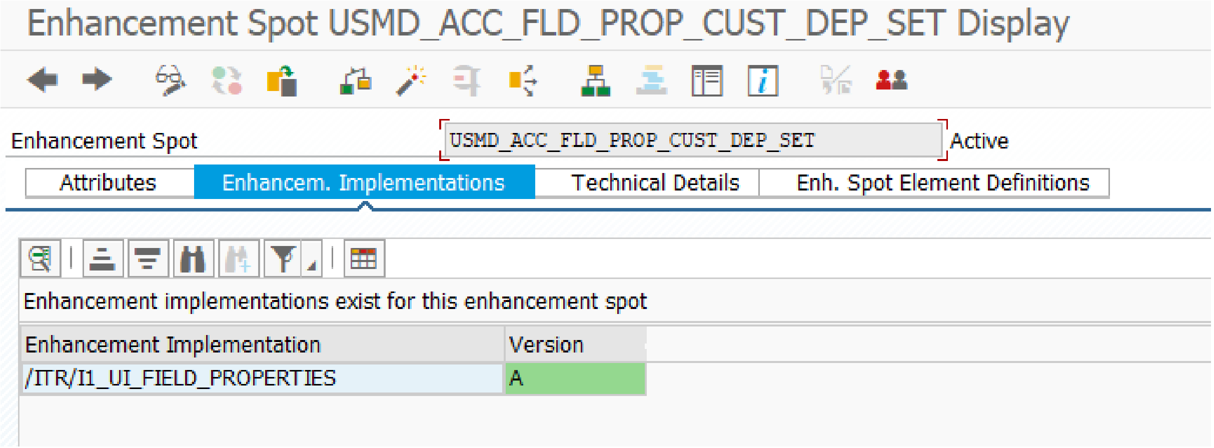
Properties of BAdI Definition

BAdI Definition Name: USMD_ACC_FLD_PROP_CUST_DEP_SET

Description: Access to Customer-Dependent Field Property Settings

Interface: IF_EX_USMD_ACC_FLD_PROP_CDS

Instance Creation Mode: Reuse of BAdI Instance



Enhancement Spot USMD_ACC_FLD_PROP_CUST_DEP_SET Display

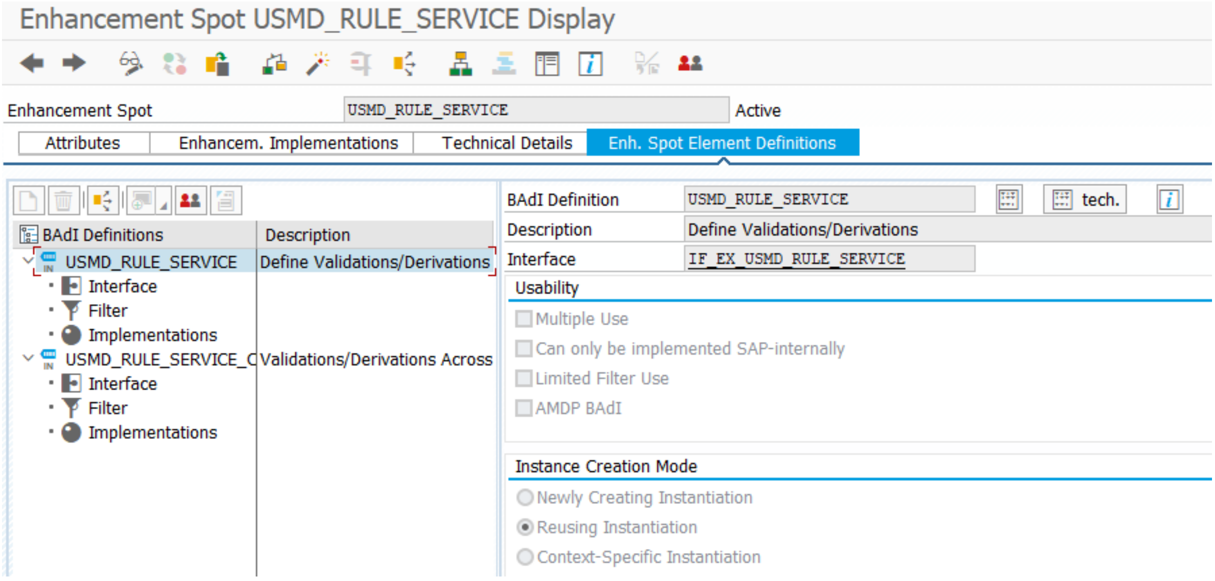
Enhancement Spot USMD_ACC_FLD_PROP_CUST_DEP_SET Active

Enhancement implementations exist for this enhancement spot

Enhancement Implementation	Version
/ITR/I1_UI_FIELD_PROPERTIES	A

For validations the Enhancement Spot USMD_RULE_SERVICE needs to be active (with two enhancement implementations):

Implementation 1: Validations/Derivations



Enhancement Spot USMD_RULE_SERVICE Display

Enhancement Spot	USMD_RULE_SERVICE	Active

BAdI Definitions	Description
USMD_RULE_SERVICE	Define Validations/Derivations
Interface	
Filter	
Implementations	
USMD_RULE_SERVICE_C	Validations/Derivations Across
Interface	
Filter	
Implementations	

BAdI Definition	USMD_RULE_SERVICE
Description	Define Validations/Derivations
Interface	IF_EX_USMD_RULE_SERVICE

Usability

□ Multiple Use

□ Can only be implemented SAP-internally

□ Limited Filter Use

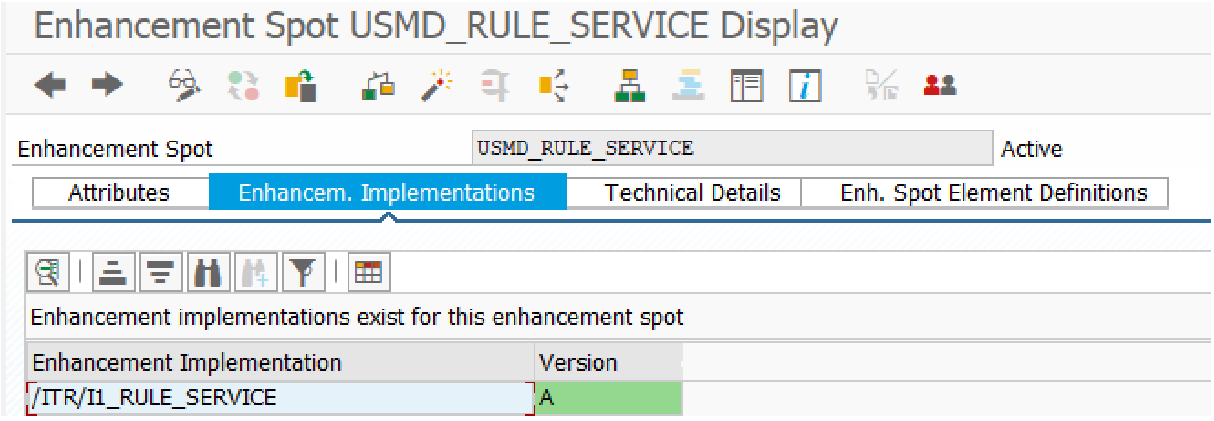
□ AMDP BAdI

Instance Creation Mode

○ Newly Creating Instantiation

● Reusing Instantiation

○ Context-Specific Instantiation

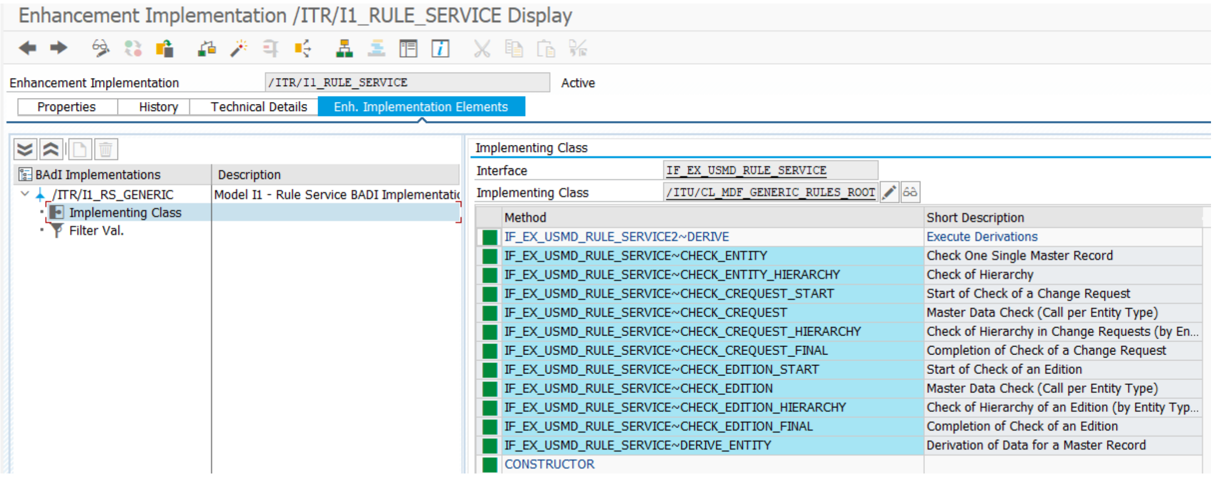


Enhancement Spot USMD_RULE_SERVICE Display

Enhancement Spot	USMD_RULE_SERVICE	Active

Enhancement implementations exist for this enhancement spot

Enhancement Implementation	Version
/ITR/I1_RULE_SERVICE	A



Enhancement Implementation /ITR/I1_RULE_SERVICE Display

Enhancement Implementation	/ITR/I1_RULE_SERVICE	Active

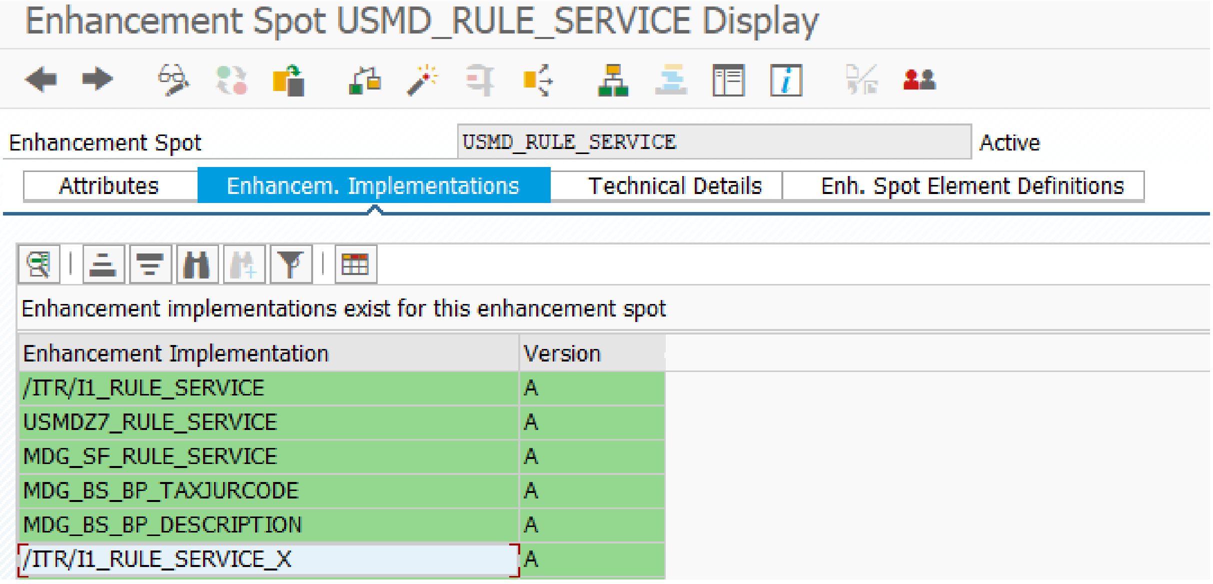
BAdI Implementations	Description
/ITR/I1_RS_GENERIC	Model I1 - Rule Service BADI Implementatic
Implementing Class	
Filter Val.	

Implementing Class

Interface	IF_EX_USMD_RULE_SERVICE
Implementing Class	/ITU/CL_MDF_GENERIC_RULES_ROOT

	Method	Short Description
🟩	IF_EX_USMD_RULE_SERVICE2~DERIVE	Execute Derivations
🟩	IF_EX_USMD_RULE_SERVICE~CHECK_ENTITY	Check One Single Master Record
🟩	IF_EX_USMD_RULE_SERVICE~CHECK_ENTITY_HIERARCHY	Check of Hierarchy
🟩	IF_EX_USMD_RULE_SERVICE~CHECK_CREQUEST_START	Start of Check of a Change Request
🟩	IF_EX_USMD_RULE_SERVICE~CHECK_CREQUEST	Master Data Check (Call per Entity Type)
🟩	IF_EX_USMD_RULE_SERVICE~CHECK_CREQUEST_HIERARCHY	Check of Hierarchy in Change Requests (by En...
🟩	IF_EX_USMD_RULE_SERVICE~CHECK_CREQUEST_FINAL	Completion of Check of a Change Request
🟩	IF_EX_USMD_RULE_SERVICE~CHECK_EDITION_START	Start of Check of an Edition
🟩	IF_EX_USMD_RULE_SERVICE~CHECK_EDITION	Master Data Check (Call per Entity Type)
🟩	IF_EX_USMD_RULE_SERVICE~CHECK_EDITION_HIERARCHY	Check of Hierarchy of an Edition (by Entity Typ...
🟩	IF_EX_USMD_RULE_SERVICE~CHECK_EDITION_FINAL	Completion of Check of an Edition
🟩	IF_EX_USMD_RULE_SERVICE~DERIVE_ENTITY	Derivation of Data for a Master Record
🟩	CONSTRUCTOR	

Implementation 2: Cross Entity Derivation

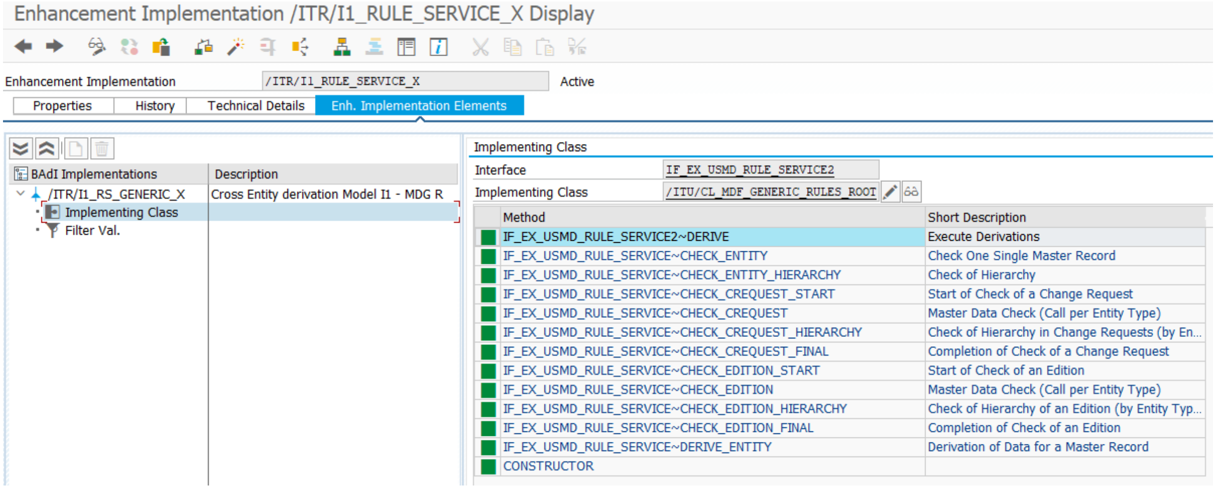


Enhancement Spot USMD_RULE_SERVICE Display

Enhancement Spot: USMD_RULE_SERVICE Active

Enhancement implementations exist for this enhancement spot

Enhancement Implementation	Version
/ITR/I1_RULE_SERVICE	A
USMDZ7_RULE_SERVICE	A
MDG_SF_RULE_SERVICE	A
MDG_BS_BP_TAXJURCODE	A
MDG_BS_BP_DESCRIPTION	A
/ITR/I1_RULE_SERVICE_X	A



Enhancement Implementation /ITR/I1_RULE_SERVICE_X Display

Enhancement Implementation: /ITR/I1_RULE_SERVICE_X Active

BAdI Implementations	Description
/ITR/I1_RS_GENERIC_X	Cross Entity derivation Model I1 - MDG R
Implementing Class	
Filter Val.	

Implementing Class

Interface: IF_EX_USMD_RULE_SERVICE2

Implementing Class: /ITU/CL_MDF_GENERIC_RULES_ROOT

	Method	Short Description
🟩	IF_EX_USMD_RULE_SERVICE2~DERIVE	Execute Derivations
🟩	IF_EX_USMD_RULE_SERVICE~CHECK_ENTITY	Check One Single Master Record
🟩	IF_EX_USMD_RULE_SERVICE~CHECK_ENTITY_HIERARCHY	Check of Hierarchy
🟩	IF_EX_USMD_RULE_SERVICE~CHECK_CREQUEST_START	Start of Check of a Change Request
🟩	IF_EX_USMD_RULE_SERVICE~CHECK_CREQUEST	Master Data Check (Call per Entity Type)
🟩	IF_EX_USMD_RULE_SERVICE~CHECK_CREQUEST_HIERARCHY	Check of Hierarchy in Change Requests (by En...
🟩	IF_EX_USMD_RULE_SERVICE~CHECK_CREQUEST_FINAL	Completion of Check of a Change Request
🟩	IF_EX_USMD_RULE_SERVICE~CHECK_EDITION_START	Start of Check of an Edition
🟩	IF_EX_USMD_RULE_SERVICE~CHECK_EDITION	Master Data Check (Call per Entity Type)
🟩	IF_EX_USMD_RULE_SERVICE~CHECK_EDITION_HIERARCHY	Check of Hierarchy of an Edition (by Entity Typ...
🟩	IF_EX_USMD_RULE_SERVICE~CHECK_EDITION_FINAL	Completion of Check of an Edition
🟩	IF_EX_USMD_RULE_SERVICE~DERIVE_ENTITY	Derivation of Data for a Master Record
🟩	CONSTRUCTOR	

5.2 Configuration Tables

RDM Configuration (see also transaction /ITR/RDMIMG) as delivered by Itego is stored here:

- Field Properties: /ITR/UI_FLD_PROP
- Rule Adapter Class: /ITR/RULE_CONFIG
- Default Values: /ITR/I1_RS_DEF
- Search Configuration: /ITR/SRCH_RS_CFG

Table /ITR/RULE_CONFIG contains the implementation classes which are provided for each reference data object type. The provided classes can be extended by the implementation of the validations that are additionally required.

MDF Configuration (see also transaction /ITU/MDFIMG) is stored in the following tables:

- Field Properties: /ITU/UI_FLD_PROP
- Rule Definition: /ITU/RF_ROOT
- Rule Type Definition: /ITU/RF_RULES
- Define Conditional Fields for Rules: /ITU/RF_CO_FLDS
- Define Values for Conditional Rules: /ITU/RF_CO_RULES
- Search Configuration: /ITU/SRCH_UI_CFG