Reference Data Management 2311

for SAP Master Data Governance

Technical Documentation

MDF Configuration Management



Version: 19.02.2024



Content

| 1 | Introd | luction: Master Data Framework | 3 |
|---|--------|--|----|
| 2 | Introd | luction: MDF Configuration Management | 4 |
| 3 | Comp | onents 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 | 13 |
| 4 | Examp | ple | 14 |
| 5 | Techn | ical Information | 17 |
| | 5.1 B | Adl Implementations | 17 |
| | 5.2 | Configuration Tables | 20 |



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

This document covers the "Configuration Management".



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.

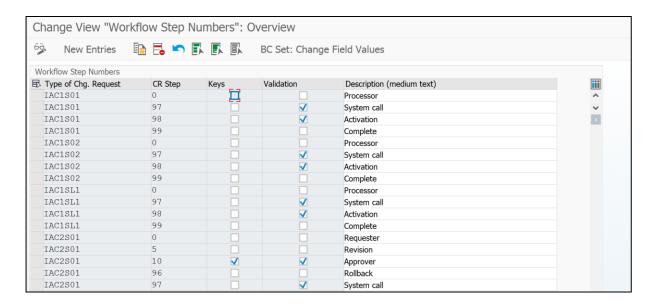
Note: The MDF Configuration Management works based on the validation framework which is provided by SAP MDG. This especially means that these configurations are activated by the activation of Business Add-Ins (BAdIs). Please check chapter 5.1 "BAdI Implementations" for additional information.

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.

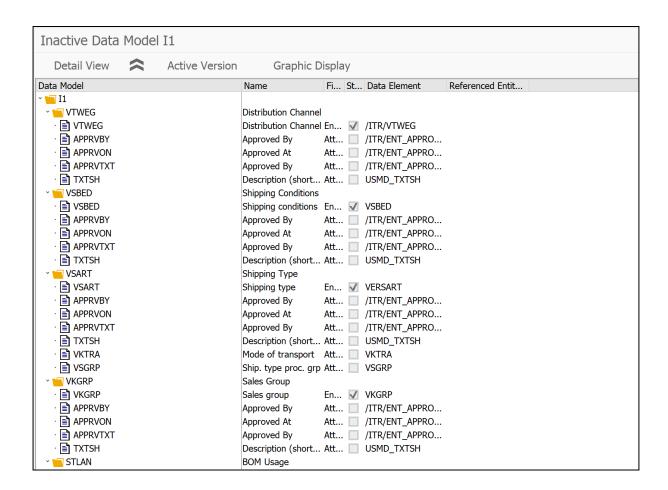
To find out which change requests are available within your system, execute transaction MDGIMG and expand the following nodes: "General Settings \rightarrow Process Modelling \rightarrow Workflow \rightarrow Rule Based Workflow \rightarrow Define Change Request Steps for Rule-Based Workflow" and execute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





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:





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

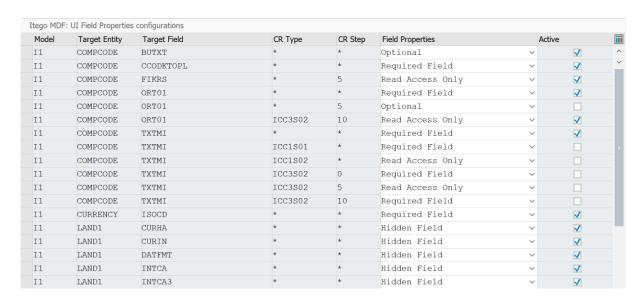
3.1.1 Field Properties

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):





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:

- Model: "I1" (for reference data objects)
- Target Entity: The entity type of the object of the field, you want to configure
- Target Field: The ID of the field that is supposed to be affected
- CR Type 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 Properties: You get to choose between several options on how the field will behave:
 - o Optional: The field is optional, no entry necessary
 - Read Access Only: Read only, no entry possible
 - o 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):



Press on "New Entries" and maintain:

- Data Model: "I1" (for reference data objects)
- Target Entity: ID of the entity type, that is supposed to be affected.
- CR Type: ID of the change request, that is supposed to be affected. Or insert "*" for all change requests.
- CR 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 can 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 | on | | | |
|-----------|------------------------|--------------|---------|-----------------|----------|
| Model | Target Entity | Target Field | Rule ID | RuleType | Active |
| I1 | COMPCODE | CCODECURR | 0001 | Validation/Limi | ta ∨ 🗸 |
| I1 | COMPCODE | CCODECURR | 0003 | Default | √ |
| I1 | COMPCODE | CCODECURR | 1003 | Default | √ |
| I1 | COMPCODE | CCODKTOPL | 0001 | Derivation | √ |
| I1 | COMPCODE | CCODKTOPL | 0004 | Derivation | √ |
| I1 | COMPCODE | CCODLAND1 | 0003 | Default | √ |
| I1 | COMPCODE | CCODLAND1 | 1003 | Default | √ |
| I1 | COMPCODE | CCODSPRAS | 0003 | Default | √ |
| I1 | COMPCODE | CCODSPRAS | 1003 | Default | √ |
| I1 | COMPCODE | CC_PERIV | 0001 | Derivation | √ |
| I1 | COMPCODE | CC_PERIV | 0004 | Derivation | √ |
| I1 | COMPCODE | ORT01 | 0005 | Default | √ |

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

- Model: "I1" (for reference data objects)
- Target Entity: ID of the entity type that is supposed to be affected
- Target Field: ID of the field that is supposed to be affected
- Rule ID: The Rule ID used in Step 1
- Rule Type: "Validation/Limitation"
- 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 Source Fields for Rules". The following table shows up (entries may differ):

| Model | Target Entity | Target Field | Counter | Source Entity | Source Field | Active |
|-------|---------------|--------------|---------|---------------|--------------|----------|
| I1 | COMPCODE | BUVAR | 1 | COMPCODE | DTTDSP | ✓ |
| I1 | COMPCODE | CCODECURR | 1 | COMPCODE | CCODLAND1 | V |
| I1 | COMPCODE | CCODKTOPL | 1 | COMPCODE | CCODLAND1 | |
| Ι1 | COMPCODE | CCODKTOPL | 2 | COMPCODE | CCODECURR | |
| Ι1 | COMPCODE | CC_PERIV | 1 | COMPCODE | CCODLAND1 | V |
| Ι1 | COMPCODE | CC_PERIV | 2 | COMPCODE | CCODECURR | √ |
| I1 | COMPCODE | CC_PERIV | 3 | COMPCODE | CCODSPRAS | V |
| I1 | COMPCODE | ORT01 | 1 | COMPCODE | CCODLAND1 | V |
| Ι1 | CURRENCY | ISOCD | 1 | CURRENCY | ISOCD | V |
| I1 | EKORG | BUKRS | 1 | EKORG | BUKRS | V |
| I1 | EKORG | EKOTX | 1 | EKORG | EKOTX | V |
| Ι1 | KTOPL | DSPRA | 1 | KTOPL | DSPRA | V |
| I1 | KTOPL | TXTLG | 2 | KTOPL | TXTLG | V |
| I1 | LAND1 | CURHA | 1 | LAND1 | CURIN | V |

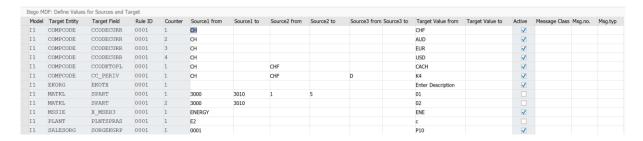


In this table, you can define fields, on which the validation of the input field, entered in the previous step, will depend on. E.g. if a company code is only allowed with currency "Euro" if the country Germany is entered, you would enter the country-field and the currency-field. Press on "New Entries"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

- Model: "I1" (for reference data objects)
- Target Entity: ID of the dependent entity type
- Target Field: 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"
- Source Entity: ID of the conditional entity type (which is used as a dependency)
- Source Field: 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 Values for Sources and Target". The following table shows up (entries may differ):



In this table, the valid values for the dependent field entered in "Define Values for Sources and Target" are defined:

- Model: "I1" (for reference data objects)
- Target Entity: ID of the dependent entity type
- Target Field: ID of the dependent field
- Rule ID: ID, defined/used in "Rule Definition"
- Counter: Multiple conditions can be defined. Each condition needs its own unique number (start with "1")
- Source[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



- Target Value From/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: R | ule Type Definition | | | | | | | |
|--------------|----------------------------|--------------|---------|--------------|--------|--|--|--|
| Model | Target Entity Target Field | Target Field | Rule ID | RuleType | Active | | | |
| I1 | MATKL | BKLAS | 0001 | Derivation v | ✓ | | | |

In activity "Define Source Fields for Rules" you can 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: D | efine Source Fields fo | r Rules | | | | |
|--------------|------------------------|--------------|---------|---------------|--------------|--------|
| Model | Target Entity | Target Field | Counter | Source Entity | Source Field | Active |
| I1 | MATKL | BKLAS | 1 | MATKL | SPART | ✓ |

In activity "Define Values for Sources and Target" you can set the value for the derived field in "Target Value from". 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.

| Model | Target Entity | Target Field | Rule ID | Counter | Source1 from | Source1 to | Source2 from | Source2 to | Source3 from Source | e3 to Target Value from | Target Value to | Active |
|-------|---------------|--------------|---------|---------|--------------|------------|--------------|------------|---------------------|-------------------------|-----------------|----------|
| I1 | MATKL | BKLAS | 0001 | 1 | 01 | | | | | 0701 | | ✓ |
| I1 | MATKL | BKLAS | 0001 | 2 | A1 | | | | | 3030 | | ✓ |
| Ι1 | MATKL | BKLAS | 0001 | 3 | A2 | | | | | 3040 | | ✓ |
| I1 | MATKL | BKLAS | 0001 | 4 | B1 | | | | | 3003 | | ✓ |

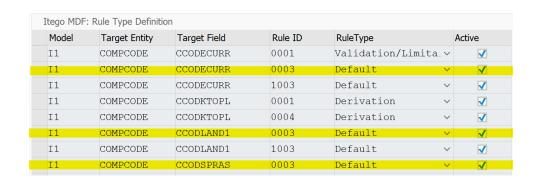
3.1.3.1 Default Values

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".

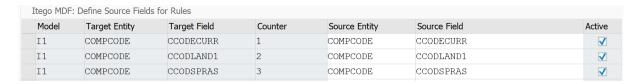
To set up such a default value, execute transaction /ITU/MDFIMG and maintain your configuration as described in section "Validation", but:

Choose Rule Type: "Default" in activity "Rule Type Definition"





In activity "Define Source Fields for Rules" maintain the field which should be defaulted.



After this define in activity "Define Values for Sources and Target" the default value without specifying Source fields in "Target Value from".





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"
- Target Entity: LGORT (Storage Location)
- CR Type: "*"
- CR 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 | Target Entity | CR Type | CR Step | Rule ID | Class/Interface | Active | | |
| I1 | LGORT | * | * | 0002 | | √ | | |

Then open "Rule Type Definition". Press new entries.

Please maintain

- Model: "I1"
- Target Entity: LGORT (Storage Location, this is the dependent entity type)
- Target Field: VSTEL (Shipping Location, this is the dependent field, it will be filled automatically with a specific value when PLANT is filled)
- 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)
- Active: Check for the rule to be active

| I | tego MDF: R | ule Type Definition | | | | |
|---|-------------|---|--------------|---------|--------------|--------|
| | Model | , | Target Field | Rule ID | RuleType | Active |
| | I1 | LGORT | VSTEL | 0002 | Derivation v | ✓ |

Open "Define Source Fields for Rules". Press new entries.

Please maintain:

- Model: "I1"
- Target Entity: LGORT (the dependent entity type)
- Target Field: VSTEL (the dependent field)
- Counter: "1", it's only depended on one field.
- Source Entity: LGORT (the conditional entity type)



- Source Field: PLANT (the conditional field)
- Active: Check for the rule to be active

| Itego MDF: | Define Source Fields fo | r Rules | | | | |
|------------|-------------------------|--------------|---------|---------------|--------------|----------|
| Model | Target Entity | Target Field | Counter | Source Entity | Source Field | Active |
| I1 | LGORT | VSTEL | 1 | LGORT | PLANT | ✓ |

Open "Define Values for Sources and Target". Press new entries.

Please maintain:

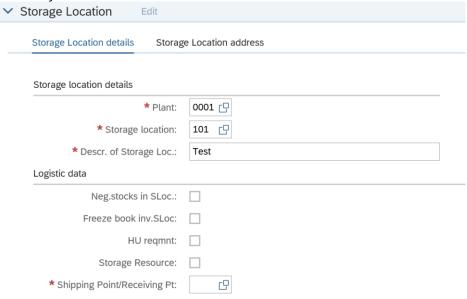
- Model: "I1"
- Target Entity: LGORT (the dependent entity type)
- Target Field: VSTEL (the dependent field)
- Rule ID: 0002
- Counter: "1", there is only one conditional field
- Source[n] From: 0001 (when this value is used in PLANT, the value for VSTEL will be set automatically)
- Target Value from: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

| Itego MDF: Define Values for Sources and Target Model Target Entity Target Field Rule ID Counter Source2 to Source3 from Source3 to Target Value from Target Value to | | | | | | | | | | | |
|--|-------|---------------|--------------|---------|---------|------------|--------------|------------|-------------------|-----------------|--------|
| | Model | Target Entity | Target Field | Rule ID | Counter | Source2 to | Source3 from | Source3 to | Target Value from | Target Value to | Active |
| | I1 | LGORT | VSTEL | 0002 | 1 | | | | 0001 | | ✓ |



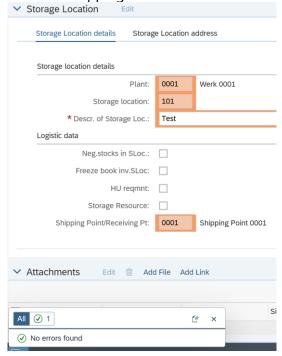
Now create a new Storage Location.

Fill every field with values. Choose 0001 for Plant.



Click "Check".

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

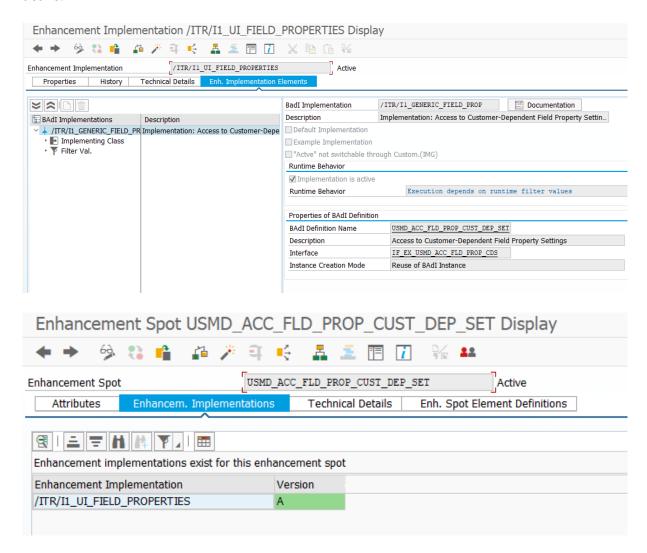




5 Technical Information

5.1 BAdI Implementations

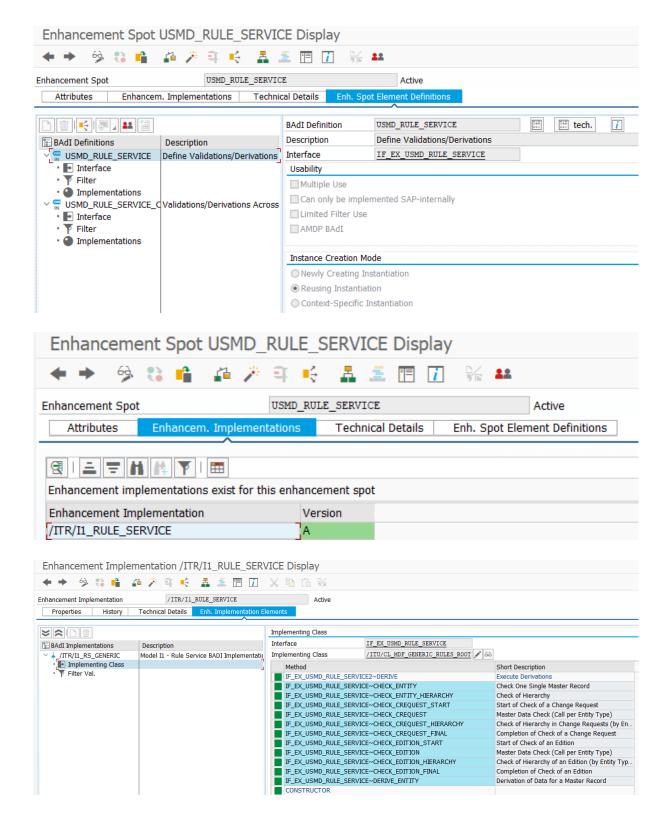
For field properties the Enhancement Spot USMD_ACC_FLD_PROP_CUST_DEP_SET needs to be active:



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

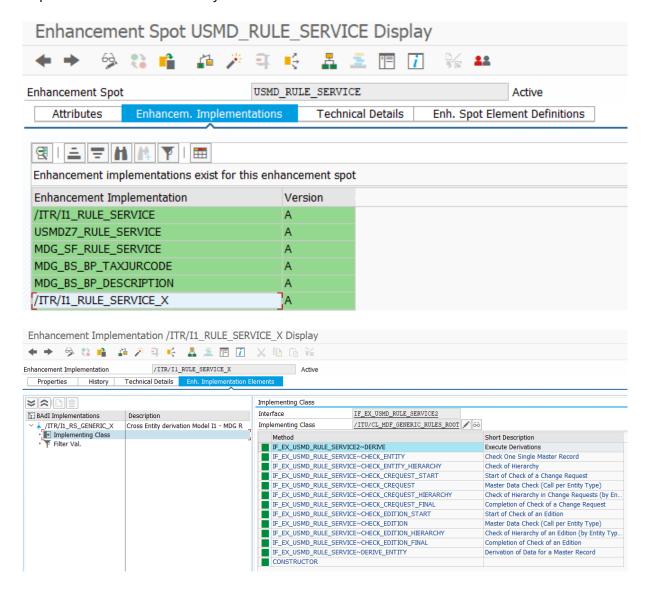
Implementation 1: Validations/Derivations







Implementation 2: Cross Entity Derivation





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