Reference Data Management

for SAP Master Data Governance

Configuration



Version: 25.09.2026 RDM 2607 (Release 805 SP16)

Content

1 Introduction 5

2 Prerequisites 5

3 Configuration for Reference Data Governance 6

3.1 Activate Business Functions 6

3.2 Activate Data Model I1 6

3.3 Activate Business Configuration Sets 7

3.3.1 SAP MDG: BC Sets 8

3.3.2 SAP Business Systems: BC Sets 10

3.4 Configure User Roles and Authorizations 11

3.4.1 Reference Data Governance – MDG 11

3.4.2 Reference Data Governance – SAP Business System 11

3.5 Copy Rule Based Workflows from Client 000 11

3.5.1 Introduction 12

3.5.2 Automated Copy 13

3.5.3 Manual Copy 16

3.6 Configure Multiple Object Change Request 17

3.6.1 Workflow Settings 17

3.6.2 Governance Settings 20

3.6.3 UI Settings 21

3.6.4 Validation Settings 22

3.6.5 Data Replication and Local Staging Area 23

3.7 Adjust Governance Scope, Search Helps and Remote Validations 24

3.7.1 SAP MDG: Adjust Governance Scope 24

3.7.2 SAP Business System: Adjust Local Scope 24

3.7.3 SAP MDG: Configure Search Helps and Remote Validations 24

3.8 Adjust User Interfaces 35

3.9 Configure Rule Based Workflow 36

3.10 Configure Changeable Keys 36

3.11 Set Up Data Replication 37

3.11.1 Data Replication Framework settings 37
3.11.2 Data Replication using Application Link Enabling (ALE) 39
3.11.3 Data Replication using WebServices 42
3.11.4 Key Mapping 44
3.11.5 Value Mapping 46
3.12 Set Up Data Transfer 49
3.13 Configure Product Hierarchy 52
3.13.1 Verify the product hierarchy usage in your business applications 52
3.13.2 BC-Set Activation 53
3.13.3 Configure the product hierarchy usage in RDM. 54
3.13.4 Check Interlocking 55
3.13.5 Define Edition 56
3.13.6 Creation of a Product Hierarchy Name 57
3.13.7 Maintenance of number ranges 57
3.13.8 Define Replication Model 58
3.13.9 Data Export from business application system 59
3.13.10 Data Import (File Upload) 61
3.13.11 Data Download 66
3.13.12 Enabling 'Add to Change Request' Functionality 67
3.13.13 Delete Hierarchy 67
3.14 Configure Classification 68
3.14.1 Internal Numbering 68
3.14.2 Upload and Download of Classes and Characteristics 69
3.15 Configure Payment Term 78
3.15.1 Activation Report 78
3.15.2 Data Migration Report 79
3.16 Configure Public Holiday 81
3.17 Configure Work Breakdown Structure 82
3.17.1 Replication of WBS Elements 82
3.17.2 Export, Import and Upload WBS 85
3.18 Configure Initial Load of Exchange Rates 92

3.19 Set Up Local Staging Area 93
3.20 Solution Manager Integration 95
3.21 Project Specific Enhancements 95
4 Configuration for Reference Data Harmonization 96
4.1 Activate Business Configuration Sets 96
4.1.1 SAP MDG: BC Sets 96
4.2 Configure User Roles and Authorizations 96
4.2.1 Reference Data Harmonization – Sender 96
4.2.2 Reference Data Harmonization – Receiver 96
5 Configuration Examples 97
5.1 Data Replication based on SAP ALE 97
5.1.1 Prerequisite: RFC Destination 97
5.1.2 Define Logical System and Check or Create Business System 97
5.1.3 Check or Create BAdI for determination of local system name 99
5.1.4 Define Distribution Model 100
5.1.5 Business Governance: Data Replication to MDG client 108
5.2 Data Replication based on Webservices 116
5.2.1 Receiver: Technical Administration: Profiles and Provider Systems 116
5.2.2 Receiver: Service Administration: Business Context 118
5.2.3 Receiver: Service Administration: Local Integration Scenario 119
5.2.4 Receiver: Services Registry: Published Systems and Objects 119
5.2.5 Sender: Technical Administration: Profiles and Provider Systems 121
5.2.6 Sender: Service Administration: Business Context 122
5.2.7 Sender: Service Administration: Local Integration Scenario 123
5.2.8 Sender: Service Administration: Logon Data 124
5.2.9 Sender: Service Administration: Web Service 124
5.2.10 Sender: Services Registry: Published Systems and Objects 125

1 Introduction

Reference Data Management (RDM) for SAP MDG provides the governance of reference data using pre-delivered reference data objects, user interfaces, workflows, and the replication of the data to a local staging area in receiving SAP systems. If you encounter any problems with this guide do not hesitate to get in contact with us. Please use support@itego.de (subject: "Configuration Guide RDM: <topic>").

2 Prerequisites

RDM is built as an Add-On to SAP Master Data Governance (MDG) which means that it needs to be installed based on SAP MDG. See the RDM Installation Guide for more details.

SAP MDG for Custom Objects (MDG-CO) needs to be activated and certain configuration steps need to be performed. See section 3.1 "Activate Business Functions".

3 Configuration for Reference Data Governance

3.1 Activate Business Functions

Before you activate business functions, ensure that you have the administration authorization for MDG. The required authorization objects are delivered with the authorization role SAP_MDG_ADMIN. In transaction PFCG, we recommend creating a copy of this role and to assign the relevant authorizations. For authorization object USMD_DM Data Model you need to assign the value for field "USMD_MODEL": "I1" and the values for "ACTVT" (e.g. 01: Create or generate and 02: Change).

In transaction SFW5 "Activate Business Functions" activate the following business functions for MDG-CO:

- MDG_FOUNDATION
- MDG_FOUNDATION_2
- MDG_FOUNDATION_3
- MDG_FOUNDATION_4
- MDG_FOUNDATION_5
- MDG_FOUNDATION_6

And activate the following business function for RDM if you want to use the RDM Product Hierarchy:

- /ITR/FOUNDATION_01

Also activate the Web Dynpro Applications for MDG-CO (see Configuration Guide for SAP MDG Custom Objects 8.0 -> "Services to be activated for MDG Web Dynpro Applications"). This document can be found here http://help.sap.com/mdg (MDG based on SAP ERP -> Application Help -> Version 8.0 -> Configuration of SAP MDG -> Config. of SAP MDG Central Governance -> Configuration of MDG for Custom Objects -> "Services to be activated for MDG Web Dynpro Applications").

Note: These business functions only need to be checked or activated on the SAP MDG system. No action necessary on the SAP Business systems which are usually the receivers of the data maintained on the SAP MDG system.

3.2 Activate Data Model I1

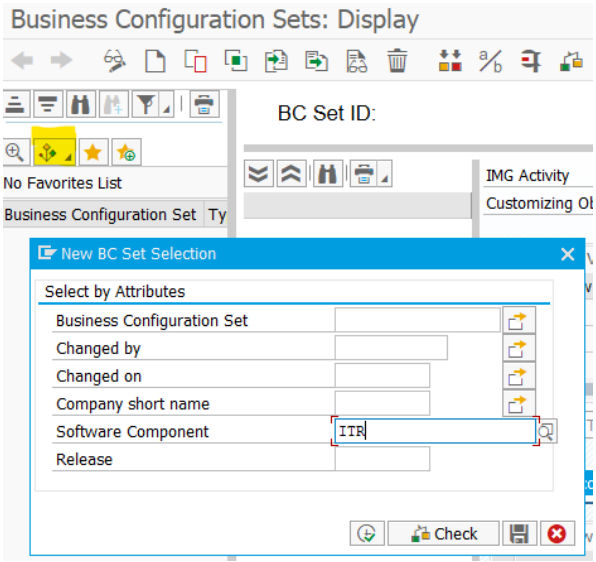
Check whether the data model I1 has been activated in transaction MDGIMG: General Settings -> Data Modeling -> Edit Data Model. If it has not been activated, select Data Model "I1" and click on  "Activate".

3.3 Activate Business Configuration Sets

The following Business Configuration Sets (BC Sets) might have to be imported on the SAP MDG system. Use transaction SCPR20 "Activate Business Sets" to activate them shown in the sequence below.

Please be aware that, as with any BC Set, you should check for conflicts before activating them. If there are conflicts, please investigate if you would like to activate anyways, partially, or not, as entries in your SAP MDG implementation might be affected. In doubt, please create a backup including the affected configuration tables in a transport, which can be used to restore your settings later if necessary. Please consider clarifying any remaining question by contacting support@itego.de (subject: "BC Set Usage")

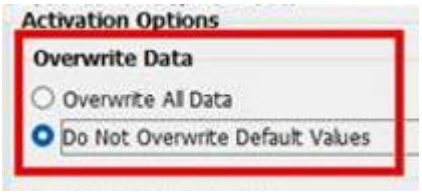
You can use transaction SCPR3 "Display and maintain BC Sets" to investigate the content of the BC Sets. Please use "Select by Attributes":



And search for Software Components ITR, ITO and ITG (on the MDG System) or ITO and ITG (on an SAP Business System).

Note: Starting with RDM 2407 the latest available BC Set includes the complete content from earlier versions. This means that for the installation of newer versions than RDM 2403 only the latest available BC Set (highest number <n>) needs to be activated. Please also be aware that new BC Sets (for the Fiori Launchpad) have been delivered which need to be activated as well. Please make sure that BC Sets are completely activated, in doubt repeat the activation step and if no complete activation is possible, contact support@itego.de (subject: "BC Set Activation").

Important Note for Upgrades: For the upgrade and the activation of new features, please make sure that "Do Not Overwrite Default Values" (see below) is used. If in doubt, please check with support@itego.de for instructions.



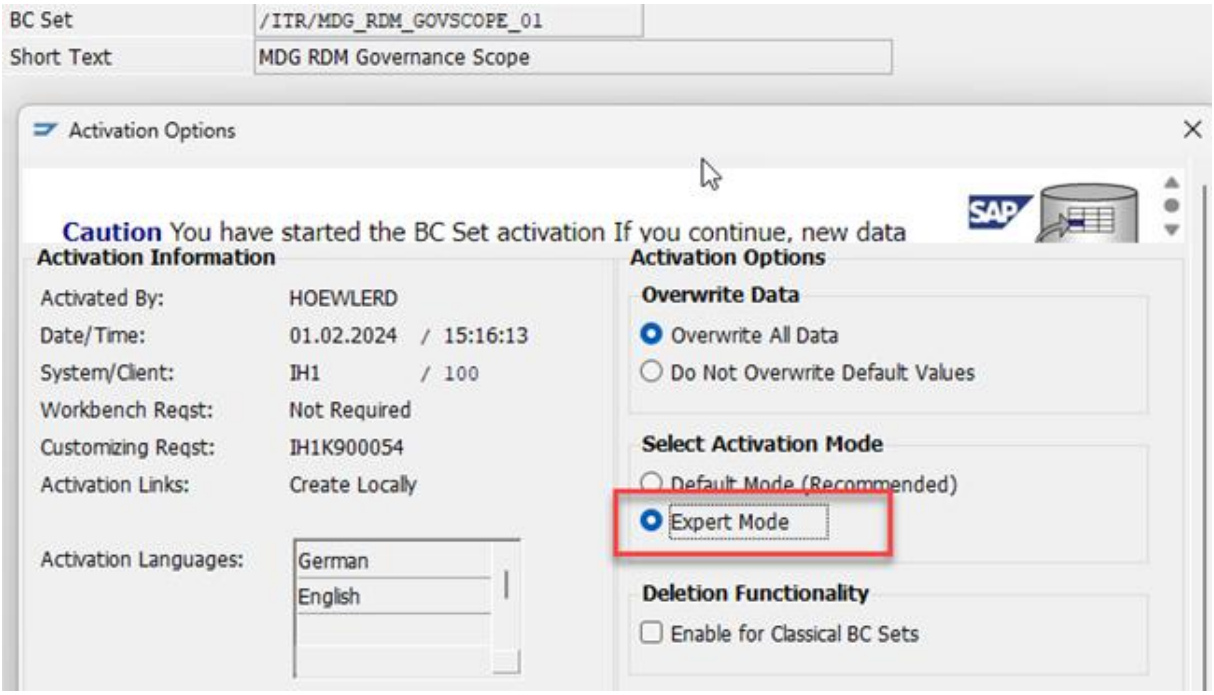
3.3.1 SAP MDG: BC Sets

3.3.1.1 Software Component ITR

The following BC Sets need to be activated on the MDG system:

- /ITR/MDG_RDM_BASIC_FRMWRK_<n> MDG RDM Framework Basic Settings
- /ITR/MDG_RDM_BUSACT_<n> MDG RDM Business Activities
- /ITR/MDG_RDM_DRF_BUS_ALT_<n> MDG RDM Replication Business Alternative
- /ITR/MDG_RDM_GOVSCOPE_<n> MDG RDM Governance Scope
- /ITR/RDM_SEARCH_HELP_<n> MDG RDM Search Help Selection
- /ITR/FLP_SEM_OBJC_<n> Fiori Launchpad Semantic Objects
- /ITR/FLP_SPACES_PAGES_<n> Fiori Launchpad Spaces and Pages

Note: /ITR/MDG_RDM_GOVSCOPE_<n> needs to be activated in "Expert Mode" which might need additional authorizations.



Besides the BC Sets listed above, the following BC Sets deliver Change Request types and Workflow configurations (**check the note below before activating**):

- /ITR/MDG_RDM_CREQUEST_<n> MDG RDM Change Request Types
- /ITR/MDG_RDM_WORKFLOW_<n> MDG RDM Workflow

Important Note: It is necessary to activate /ITR/MDG_RDM_WORKFLOW_01 as well if you want to use the delivered change request types. Please handle with care, as this BC Set delivers Step Types, Actions and Change Request Status that might already have been used in your system. These are:

- Step Types: A, E, H, P
- Actions: I1
- CR Status: 51, 53, 54, 55, 98, 99

Please make sure that you at least create a backup of the current settings and align them with the setting from the BC Sets after activation. In doubt **do not** activate this BC Set. If you do not activate it, you either must use your own change request types or you need to adjust the delivered change request types.

For Data Transfer please activate these BC Sets:

- /ITR/MDG_RDM_DT_<n> MDG RDM Data Transfer Settings

To get some predefined File Upload Variants please also consider activating the following BC Sets (please be aware that not all possible use cases are covered):

- /ITR/MDG_RDM_UPLOAD_VAR_<n> Upload Variants for RDM Objects

Note: In earlier versions of RDM the BC Set /ITR/MDG_RDM_BRF_DRF_<n> has been used to deliver templates for the configuration of the Data Replication. Do not activate these but review chapter 3.10 Set Up Data Replication for additional information.

3.3.2 SAP Business Systems: BC Sets

Note: It is quite common that MDG Systems have more than one client. One client (e.g. 100) could be used for the MDG system itself and another client or clients (e.g. 400 and 401) could be used to serve as "(Test) Business Systems". In this case also activate the BC Sets mentioned for Software Component ITO on the MDG system clients that you actually use as "(Test) Business System".

3.3.2.1 Software Component ITO

The following BC Set must be imported to all SAP business systems:

- /ITR/RDM_STAGING_<n> MDG RDM Local Staging Area

The following BC Sets should be imported to the SAP business system, which is used for the extraction of reference data (used for the initial load of the RDM system)

- /ITR/RDM_MDMGX_<n> MDG RDM Data Export

The following BC Sets must be imported to all SAP business systems which will be integrated to SAP Solution Manager ChaRM (not required for XLD):

- /ITR/MDG_RDM_SOL_MAN_<n> MDG RDM Solution Manager Integration

3.4 Configure User Roles and Authorizations

3.4.1 Reference Data Governance – MDG

The following roles are delivered for Reference Data Governance on the MDG system:

- /ITR/ITEGO_MDG_RDM_MENU_<n> NWBC Menu
- /ITR/ITEGO_MDG_RDM_DISP_<n> Display Role
- /ITR/ITEGO_MDG_RDM_REQ_<n> Requester
- /ITR/ITEGO_MDG_RDM_SPEC_<n> Data Specialist
- /ITR/ITEGO_MDG_RDM_STEW_<n> Data Steward

These Requester, Data Specialist and Data Steward Roles are reused in the BRF Workflow definitions for the Change Requests described in section 3.9 "Configure Rule Based Workflow". It is recommended to copy the roles to your own namespace and replace the roles in the Workflow definitions.

3.4.2 Reference Data Governance – SAP Business System

The following roles are delivered for Reference Data Governance on the SAP business systems:

- /ITR/ITEGO_LSA_MENU Local Staging Area Menu
- /ITR/MDG_LSA_ADMIN Local Staging Area Administration
- /ITR/MDG_STAGING_<n> Local Staging Area
- /ITR/MDG_LSA_SNAPSHOT_MNG_<n> Snapshot Management

Make sure to adjust the authorization objects of your roles to define which roles can access and work with which reference data object types. The delivered roles should be excluded from the authorization profiles for other roles.

After the adjustment of the roles, assign your users to the roles and make sure that data model "I1" is assigned to the user profile parameter R_FMDM_MODEL "SAP Master Data Governance".

The following role is delivered for the Master Data Framework on the MDG system and enables the user to maintain Business Rule configurations:

- /ITU/MDF_RF_MNT_<n> MDF Rule Framework Maintenance

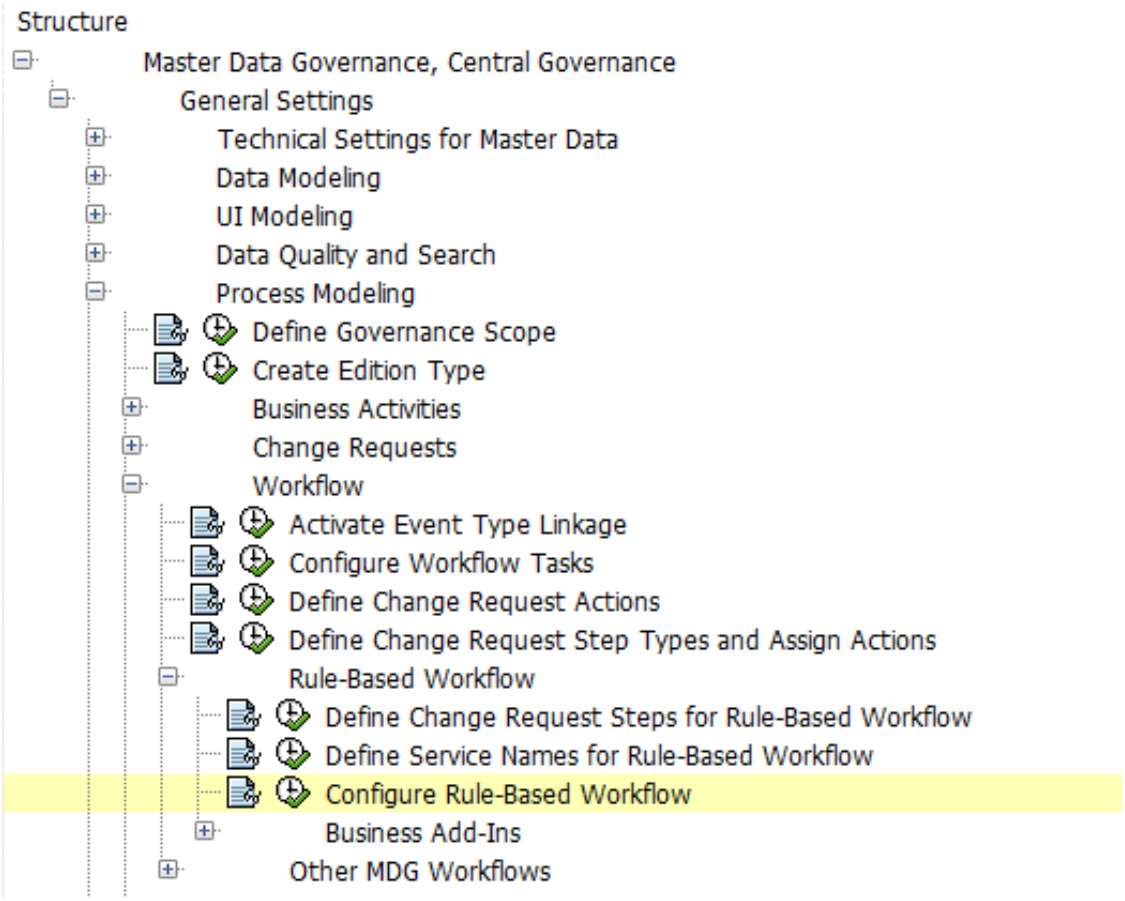
3.5 Copy Rule Based Workflows from Client 000

Rule based workflows delivered by RDM are delivered in client 000. To transfer these workflows to your operational client, you need to execute the following steps. Two alternatives are provided, the "automated copy" (which is recommended) or the "manual

copy". Please make sure that you make yourself familiar with the topic using the introduction provided below.

3.5.1 Introduction

Using transaction MDGIMG -> Process Modelling -> Workflow -> "Configure Rule-Based Workflow" provides access to the BRF+ (Business Rule Framework+) decision tables, that provide flexible processing of RDM change requests.



The BRF+ customizing used for this is delivered by SAP or SAP Partners like Itego to client 000 (using "C-tables"). This is how client specific data for BRF+ is delivered without overwriting data on target systems. For more information you can also check: BRFplus User Guide.

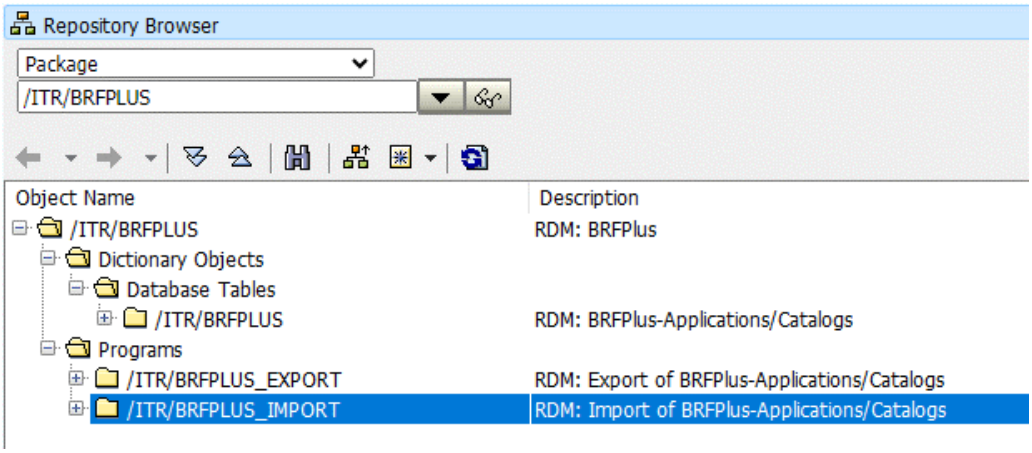
The customizing must be transferred from client 000 to the operational client where the change requests are processed. SAP standard provides a copy functionality (see also: How to copy BRF+ rules in your target client) and Itego implemented a convenient way to transfer

the customizing using the SAP XML Export/Import functionality of BRF+ (see also: Export and Import of BRFplus XML Data). This is called "Automated Copy" and described below.

3.5.2 Automated Copy

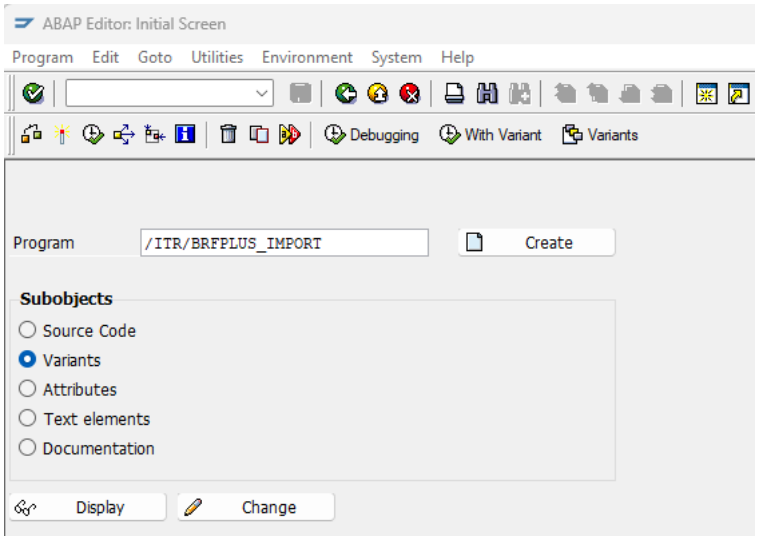
Itego RDM package /ITR/BRFPLUS includes two programs (reports)

- /ITR/BRFPLUS_EXPORT
- /ITR/BRFPLUS_IMPORT

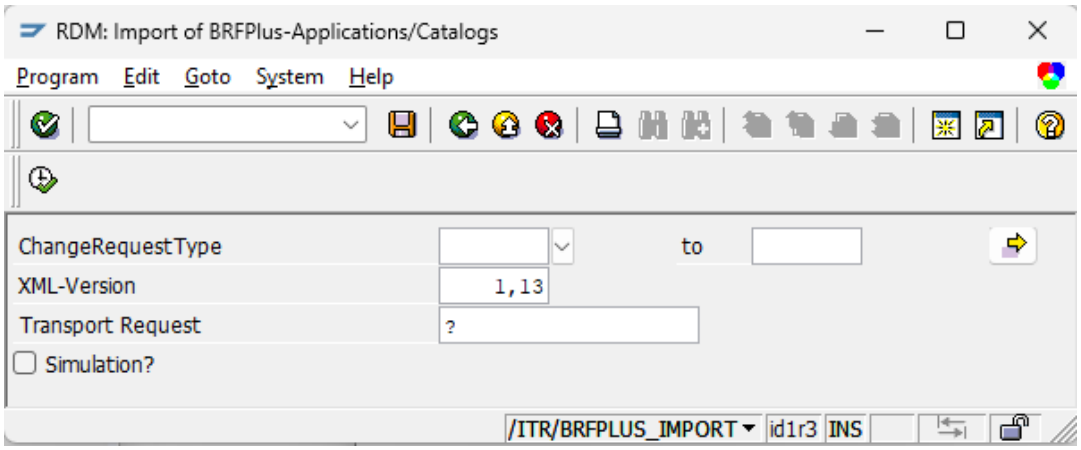


The XML export is already done by Itego and BRF+ data is delivered via table /ITR/BRFPLUS.

Before the XML Import (report /ITR/BRFPLUS_IMPORT) can be started a workbench request has to be created with transaction SE09 or SE10. After creating the request run transaction SA38 and start report /ITR/BRFPLUS_IMPORT:



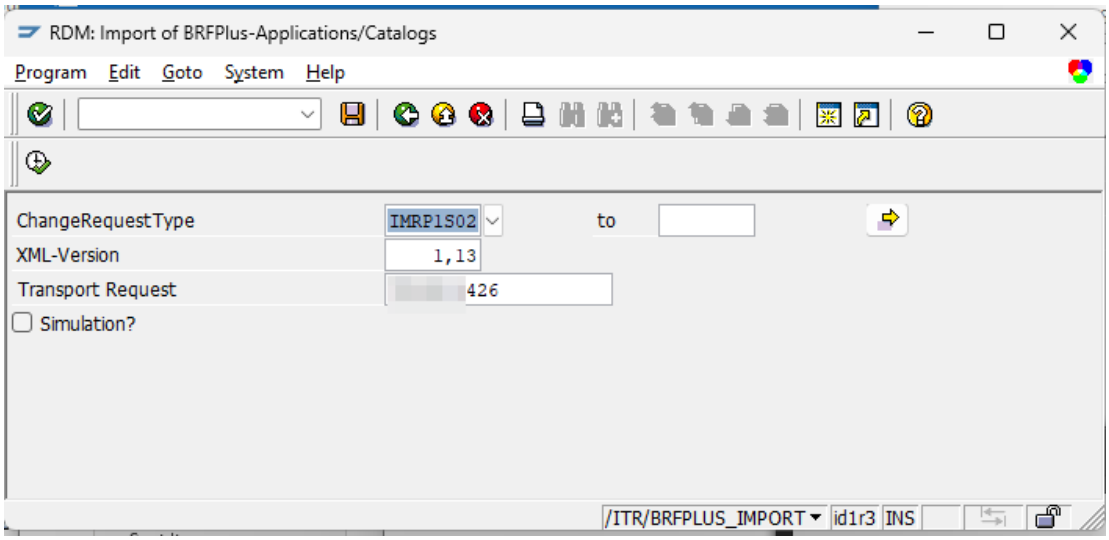
The report offers the following selection criteria:



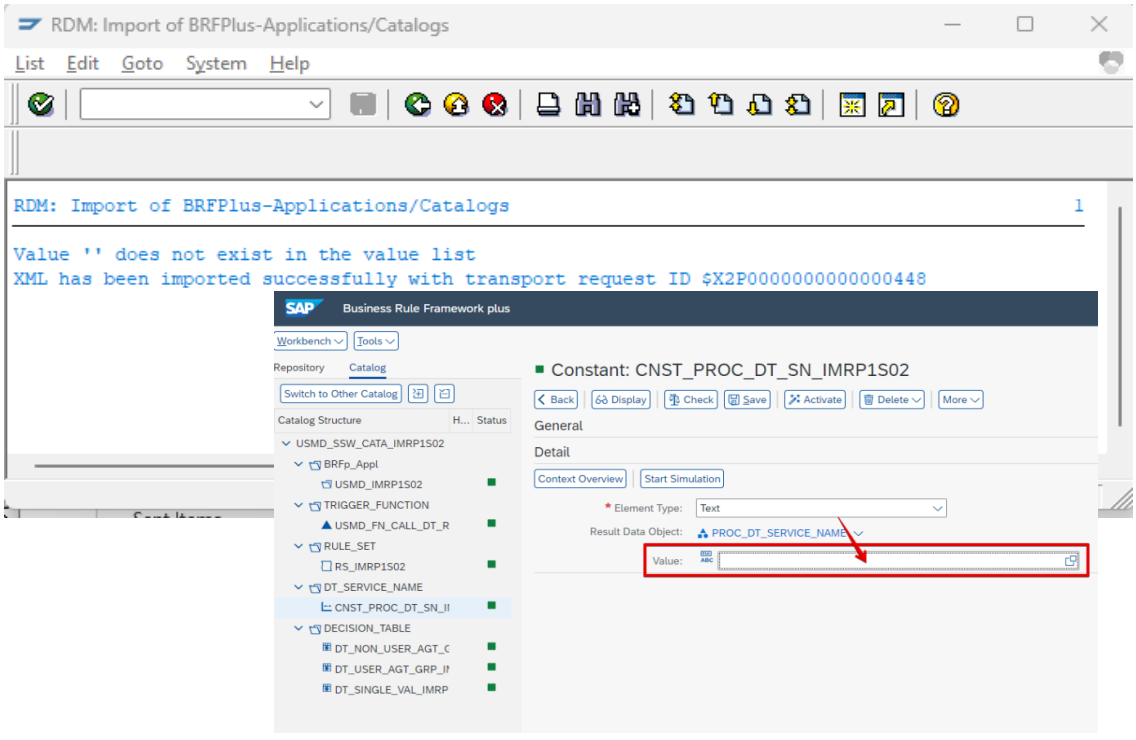
- Change Request Type
  Here you can select a range, use a generic entry like IC* or select explicit change request types to be written to BRF+.
- XML-Version
  Usually it is fine to take the default value.
  - Itego delivers version 1,11; 1,12 and 1,13.
  - A higher version e.g. 1,14 works with 1,13 as well
- Transport Request.
  During the program run, the imported BRF+ data records are written to the transport request (TR) you just created before. The TR is mandatory, as the SAP XML interface requires it. For RDM a TR is not mandatory, but you can use this TR to import the BRF+ data to other systems if necessary.

- Simulation
  Checking this check box means to execute a test run including logging.

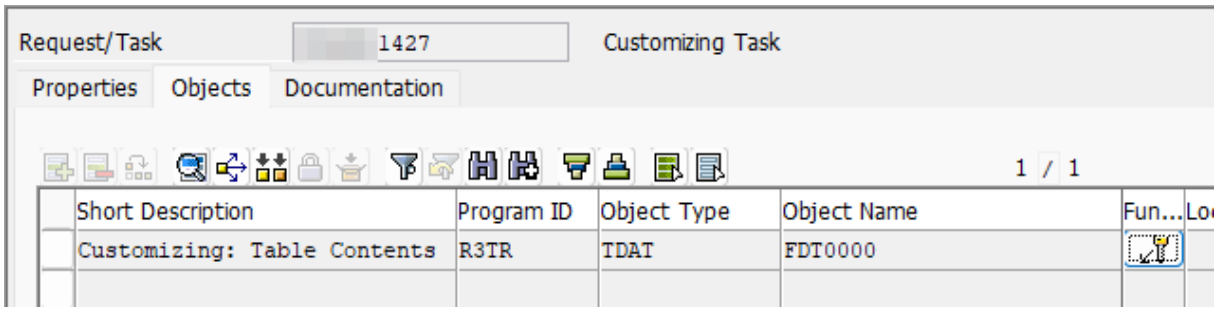
Example: Change request type IMRP1S02



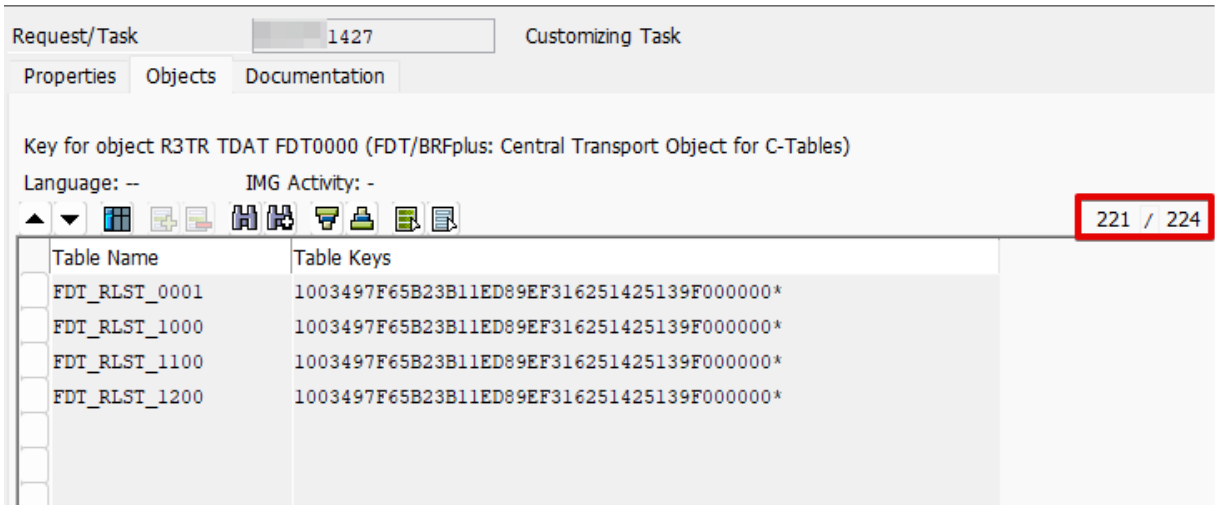
Please ignore the warning: Value '&1' does not exist in the value list (FDT_EXPRESSIONS-016)
In context of change request decision tables a not needed text field has no value, that's it – please see below:



Your created customizing request has one entry for TDAT FDT0000



and 224 entries for table keys in different tables:



This means a large amount of data will be written to different database tables with a long processing time especially when you copy many (or all) change request types. The XML processing also creates a lot of resource consumption and therefore processing in background is recommended.

3.5.3 Manual Copy

1. Create Transport: Log in to client 000 and create a customizing transport.
2. Identify Application IDs: In the Data Browser (transaction SE16), enter the table name "FDT_ADMN_0000". In the table-selection screen, search for the name(s) "DT_SINGLE_VAL_I*" and press execute. Then copy all APPLICATION_IDs and close the transaction. Please set the maximum of hits to 1,000.
   Hint: you can use System->List->Save->Save and save the table as a "Text with Tabs" and import this file into a spreadsheet format (like e.g. Excel). Then copy all APPLICATION_IDs.

3. Transfer Applications to Transport: Execute transaction SE38 and execute the report "FDT_TRANS". In the Workbench/Customizing field enter the transport, you created in step 1. For the Object ID field, press on multiple selection and paste the copied APPLICATION_IDs from step 3 into the "Single value" table (Hint: you can also save your list of APPLICATION_IDs into a text file and import it by clicking "Import from Text File" in the "multiple selection" dialog box). Then select the "Transport Whole Application(s)" checkbox and execute the transaction. Close the program once it's done.
4. Release Transport: Execute transaction SE10 and release the transport, you created in step 1.
5. Copy Applications to MDG client: Log on to your MDG client and copy the transport (using transaction SCC1, including the tasks of the request) you just released in step 4 (Source Client: 000). Then check the same table as in step 2 for the copied data.

3.6 Configure Multiple Object Change Request

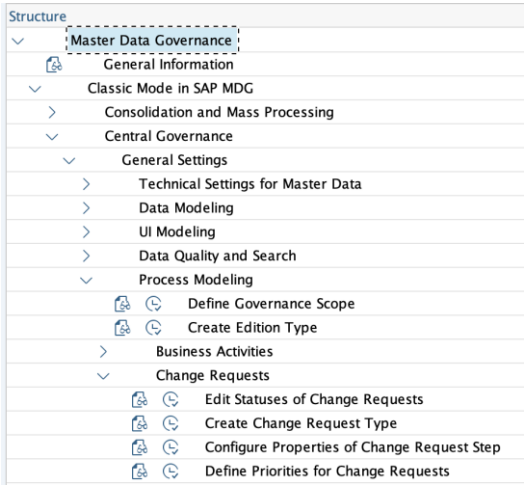
Implementing a multiple object change request allows to manage and process changes for several related master data objects simultaneously within a single workflow transaction. This functionality is recommended for scenarios requiring the coordinated maintenance of interdependent entities (e.g., Company and its Company Codes).

3.6.1 Workflow Settings

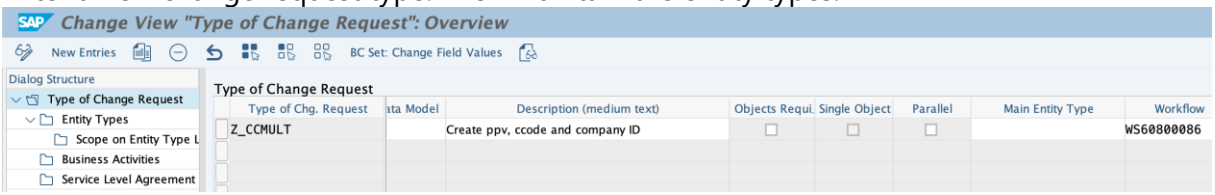
To create a multiple object, change request workflow, a new change request type must be created. It is recommended to copy an existing, similar change request type template and modify the settings to ensure consistency and to minimize the configuration effort.

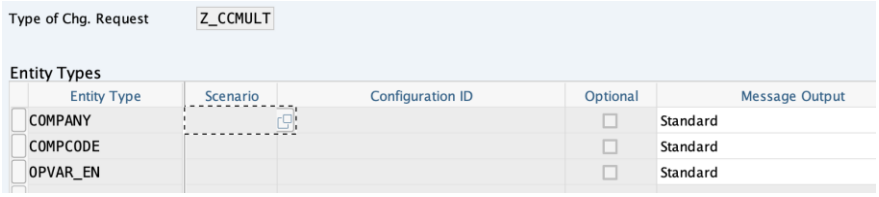
For detailed explanations and more information visit SAP Help Portal – Rule based Workflows
The following short guide shows relevant changes with the example Company Code, Company ID, and Posting Period Variant.

Open Tx MGDIMG and choose "Create Change Request Type".

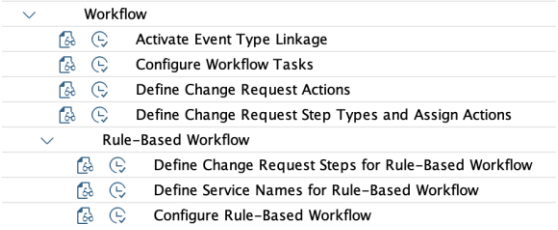


Enter a new change request type. Then maintain the entity types.

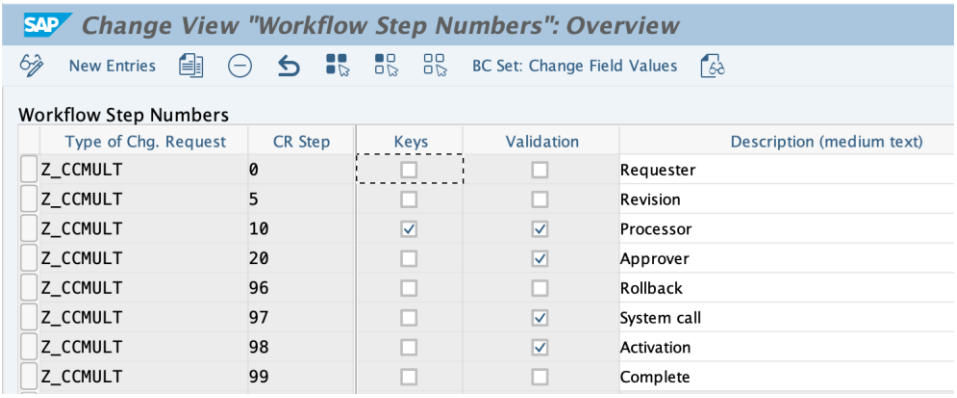




In the overview choose “Define Change Request Steps for Rule-Based Workflow”.

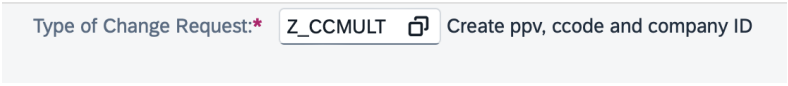


Enter the steps with description for the Change Request Type.

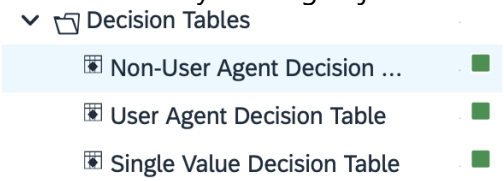


In the overview choose “Configure Rule-Based Workflow”.

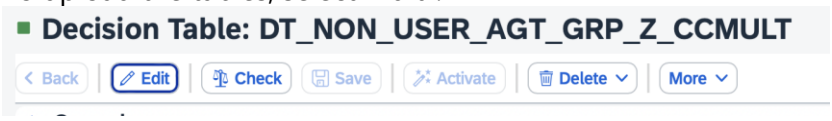




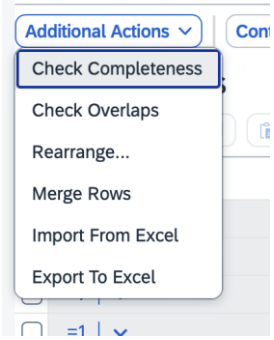
From an already existing object the decision tables should have been downloaded.



To upload the tables, select “Edit”.



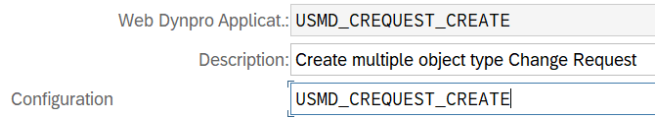
Import the excel for every table save, and activate,



After activation of the workflow can be used via the Fiori tile

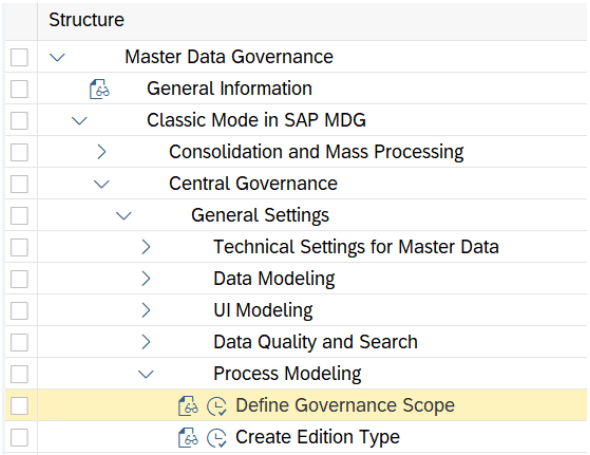


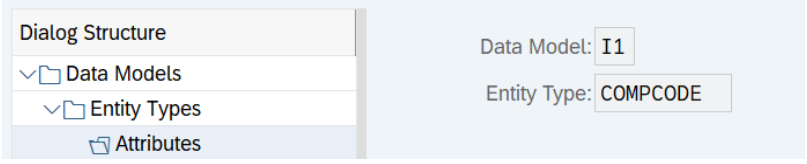
If you are using NWBC the Web Dynpro Application can also be added to a user role or the NWBC launchpad by using these settings:



3.6.2 Governance Settings

To make sure that the object types can be assigned to each other in the change request processing check the MDG governance scope and make sure that "yes" is configured. Use transaction MDGIMG:





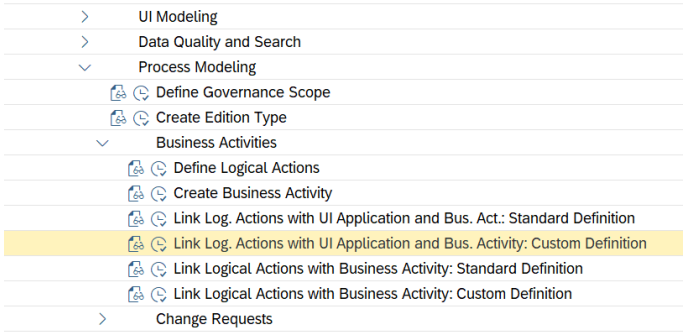
Example here: maintain attributes "Company" and "Posting Period Variant" from "Company Code".



☐	OPVAR	Posting Period Variant	Yes

3.6.3 UI Settings

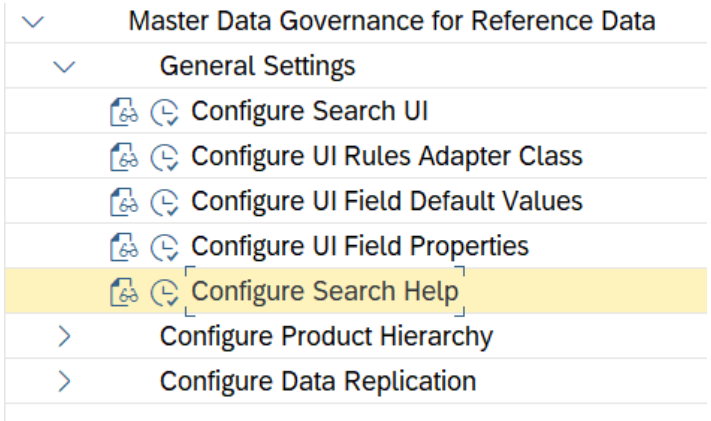
Configuration of User Interface using transaction MDGIMG:



Example: usage of enhanced "Company Code" UI with assignments for "Company" and "Posting Period Variant":

☐	I_COMPANY	*	*	*	USMD_OVP_GEN	/ITR/I1_COMPANY_OVP_GEN
☑	I_COMPCODE	*	*	*	USMD_OVP_GEN	/ITR/I1_COMPCODE_OVP_GEN_02
☐	I_CURRENCY	*	*	*	USMD_OVP_GEN	/ITR/I1_CURRENCY_OVP_GEN

Configuration of search helps using transaction: /ITR/RDMIMG



Example here: assign "Company" and "Posting Period Variant" from "Company Code" and assign "Company Code" from "Company":

☑	I1	COMPANY	/ITR/_S_I1_PP_COMPANY	CCODCOMP	Company Codes Address / Company Code	MDG-Staging
☐	I1	COMPCODE	/ITR/_S_I1_PP_COMPCODE	CCODECURR	Company Code / Currency	Customizing
☐	I1	COMPCODE	/ITR/_S_I1_PP_COMPCODE	CCODKTOPL	Company Code / Chart of Accts	Customizing
☐	I1	COMPCODE	/ITR/_S_I1_PP_COMPCODE	CCODLAND1	Company Code / Country/Reg.	Customizing
☐	I1	COMPCODE	/ITR/_S_I1_PP_COMPCODE	CCODSPRAS	Company Code / Language	Customizing
☑	I1	COMPCODE	/ITR/_S_I1_PP_COMPCODE	OPVAR	Company Codes / Posting Period Variants	MDG-Staging
☑	I1	COMPCODE	/ITR/_S_I1_PP_COMPCODE	RCOMP	Company Codes / Global Company Data (for KONS Ledger)	MDG-Staging

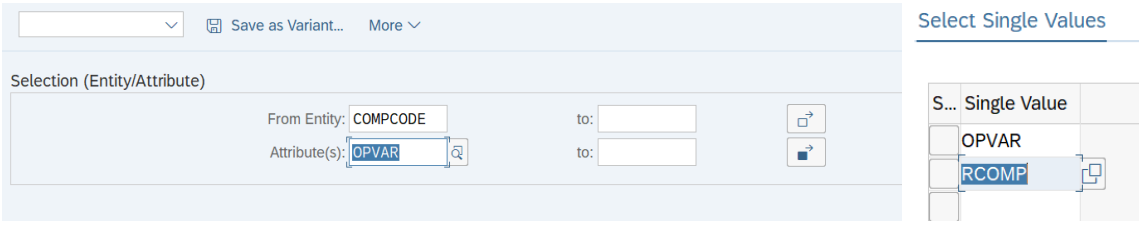
3.6.4 Validation Settings

In some use cases the data model-based validations for the so called "existence check" need to be deactivated. For this purpose, two reports are delivered which need to be executed after each data model generation.

- /ITU/CHG_NO_EXIST_CHECK (for data model attributes)
- /ITU/CHG_NO_EXIST_CHECK_RELA (for data model relations)

The example below is again handling the scenario of maintaining Company, Company Code and Posting Period Variant in a multiple object type change request.

Run report /ITU/CHG_NO_EXIST_CHECK (for data model attributes) with transaction SA38:

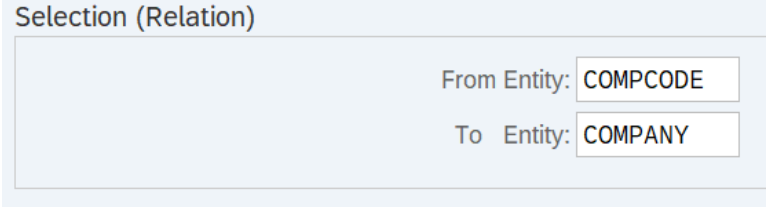


Result:

Entity	Attribute	from	to
COMPCODE	OPVAR	' '	'X'
COMPCODE	RCOMP	' '	'X'

This means: Posting Period Variant (attribute OPVAR) and Company (attribute RCOMP) can now be assigned from Company Code (entity COMPCODE) even though they have not been activated.

Run report /ITU/CHG_NO_EXIST_CHECK_RELA (for data model relations) with transaction SA38:

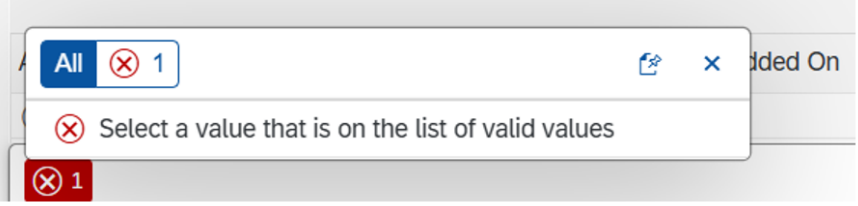


Result:

R_ENTITY	RELATION	TO_ENTITY	OLD_FLAG	NEW_FLAG
COMPCODE	CCODCOMP	COMPANY	' '	'X'

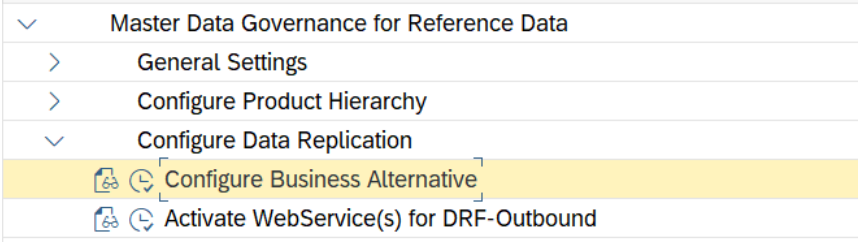
This means: Company Code (entity COMPCODE) can now be assigned from Company (entity COMPANY) can now be assigned from Company Code even though it has not been activated. This is achieved by deactivating the existence check on relation CCODCOMP.

The last two steps need to be executed if the system throws warnings like or like this after selecting already.



3.6.5 Data Replication and Local Staging Area

To be able to select the multiple objects which are maintained in this change request type for activation in the Local Staging Area in the receiving business system it needs to be checked that the change request information is transferred to this system. Check transaction /ITR/RDMIMG in the MDG system:



Example: business system "IH1 client 400":

Business System	Logical system	Entity Type	A
IH1400	IH1400	RDM_CR_TF	☑

3.7 Adjust Governance Scope, Search Helps and Remote Validations

Adjusting the Governance Scope on SAP MDG leads to a reduced set of attributes which are maintained for a specific object type. As the set of attributes might also differ from a local point of view also the local scope on SAP Receivers can be adjusted.

3.7.1 SAP MDG: Adjust Governance Scope

You can determine a set of governed attributes for each reference data object type. Fields which will be defined as "out-of-scope" are shown as read-only in change requests unless they are removed from the user interface. Prerequisite: You are aware of the consequences of changing the governance scope. See the help document in transaction MDGIMG "Customizing for Master Data Governance": General Settings -> Process Modeling -> Define Governance Scope before you execute this activity.

Most fields defined in this customizing activity will not be replicated but will be replaced by a "no data" sign which will allow to keep the local field values in the receiving system. Certain limitations apply, so please make sure to test the data replication after the definition of the Governance Scope.

3.7.2 SAP Business System: Adjust Local Scope

Based on the attributes which can be maintained globally on the SAP MDG Sender you can define specific scope adjustments on each SAP Receiver system. For this use transaction /ITR/RDM_LSA_FIELDS which maintains view /ITR/LSA_TABS_FV. Every attribute which is not within this scope definition will not be overwritten when data is activated on the business system.

3.7.3 SAP MDG: Configure Search Helps and Remote Validations

3.7.3.1 Configure Search Helps

This feature is mainly used to avoid additional configuration or development efforts when the Search Helps in an object type which is in the RDM scope refer to an object type which is not in the RDM scope.

Note: currently some object types are not fully covered. This includes all object types which in the Search Help refer to the object types Language, Country, and Currency (which could be loaded to RDM) but also Unit of Measure, Tax Code, MRP Controller and Holiday Calendar (which might need additional configuration for some Search Helps). Please consider clarifying remaining questions by contacting support@itego.de (subject: "Configure Search Helps")

Depending on the chosen Governance scope you should align the search help for each object type in scope. This can be done by calling transaction /ITR/RDMIMG -> General Settings -> Configure Search Helps. Here the following pattern should be applied:

- Configure Search Helps for each object type in scope.
- For each object type configure all search helps
- For each search help configure:
    - "Customizing" if the object type to be maintained in this attribute **is not** part of the Governance Scope
    - "MDG-Staging" if the object type to be maintained in this attribute **is** part of the Governance Scope

RDM will show

- Data from the ERP persistence (related ERP customizing tables) for each search help flagged with "Customizing"
- Data from the MDG persistence (generated MDG tables) for each search help flagged with "MDG-Staging"

Note: if you do not configure this, RDM will show data as defined in the data model. Means: sometimes data will be shown based on generated MDG tables, sometimes data will be shown based on related ERP customizing tables. As a data replication to the MDG client should be configured (see Chapter 3.10), data in the ERP customizing tables in general should be the same as the data in the generated MDG tables. Which means that the entry "Customizing" in most cases should be the right setting because:

- for the Business Governance Alternative (Direct Update) the data is written to the ERP persistence immediately after activating an approved Change Request
- for the IT Governance Alternative (via Local Staging Area) the data is written to the ERP persistency immediately after the data is activated in the Local Staging Area.

However, if you configure change request types which can be used to maintain more than one object type, the expected results could be different as the search help in such a change request type usually would show inactive data as well. In these cases, the setting "MDG-Staging" might make sense.

3.7.3.2 Remote Validations

3.7.3.2.1 Configure Remote Validations

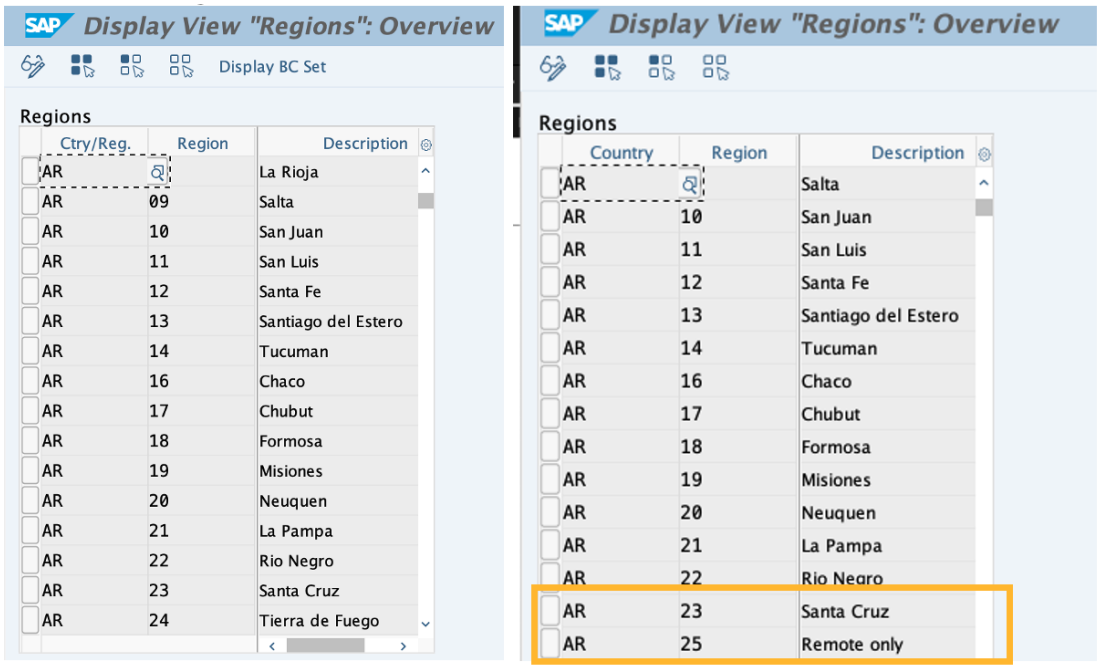
Remote validations allow to verify data against external systems in real-time to ensure it meets specific target system requirements before activation. The setup for Remote Validations is done by defining the entity (e.g. Plant), the field definition level, and the specification of the source for the value (a remote SAP Business System).

For each individual field, you can determine whether the value source is local or remote, ensuring that the system does not merely display information but actively performs a validation. An error message is triggered if a value does not exist in the remote system or if an inconsistency is detected between the local data and the remote destination, thereby maintaining high data integrity across the landscape.

3.7.3.2.1.1 Set Remote Validations

As an example, we use the attribute *Region* of the entity type *Plant Address*.

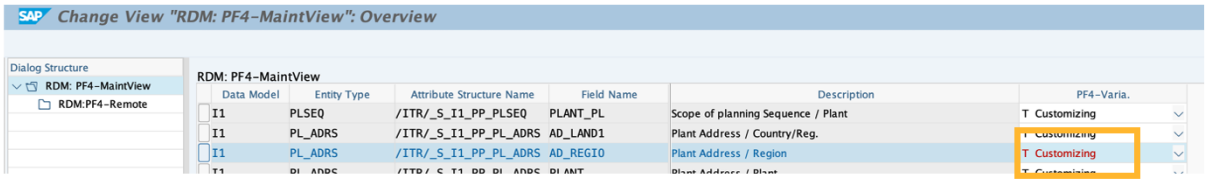
In the local system (table on the left) we see entries *1-24* for the regions in the country *Argentina*. In the remote system (table on the right) we do not have Region *24* and there is an additional Region *25*.



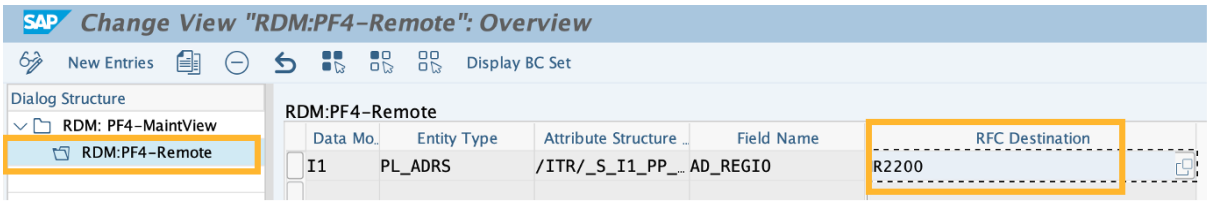
To use Remote Validations, start transaction */ITU/PF4_CHOICE_RE*.



You can set the origin of the values in column '*PF4-Variant*' from the customizing table or from the MDG-Staging tables. When using remote validations "*Customizing*" must be set!

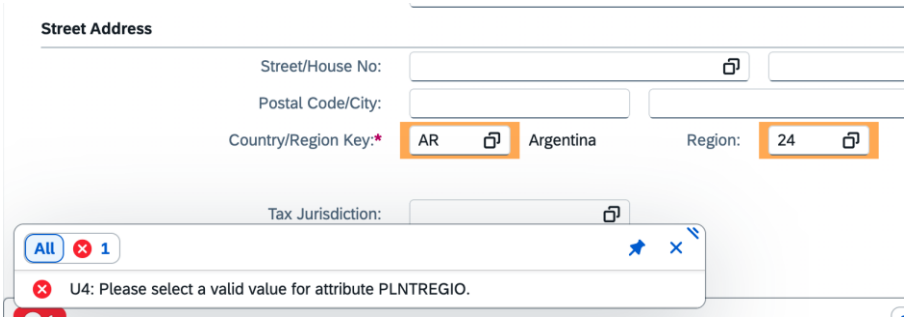


Selecting the field *Region (AD_REGIO)* of entity type *Plant-Address (PL_ADRS)* and then "*RDM:PF4-Remote*" allows to enter an RFC Destination (e.g. the ERP System).



Values are not only displayed but also undergo a formal validation process. An error message is triggered if the value does not exist in the remote system or if an inconsistency is detected between the local and the remote system.

E.g. we have *Region 24* in the local system that does not exist in the remote system, which leads to an error message.



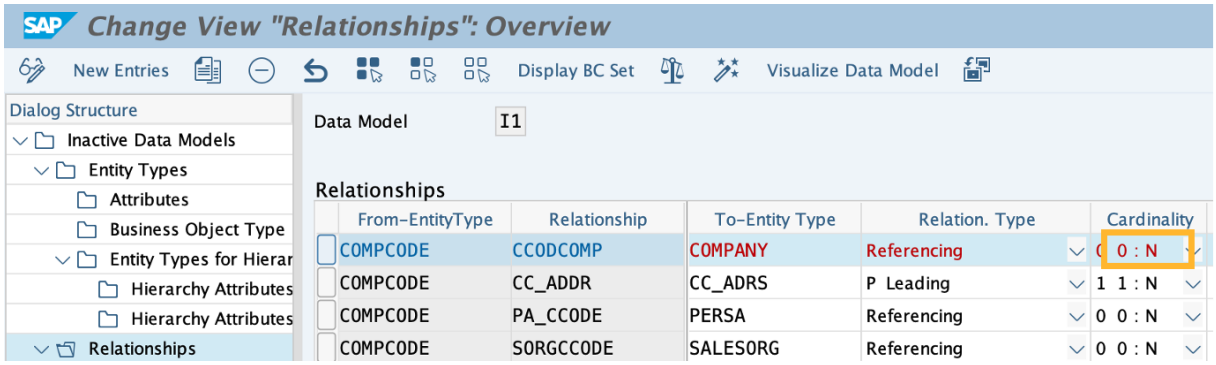
3.7.3.2.1.2 Deactivate Existence Check

The Existence Check typically validates data against local values by default. However, this causes issues when values exist only in a remote system. To resolve this, the local existence check must be deactivated. For this use report */ITU/CHG_NO_EXIST_CHECK* and/or */ITU/CHG_NO_EXIST_CHECK_RELA* via transaction *SA38*. Entries are governed by tables *USMD0022* and *USMD0023*. The choice of the report depends on how the attribute is linked to the entity:

1. Report */ITU/CHG_NO_EXIST_CHECK_RELA:*

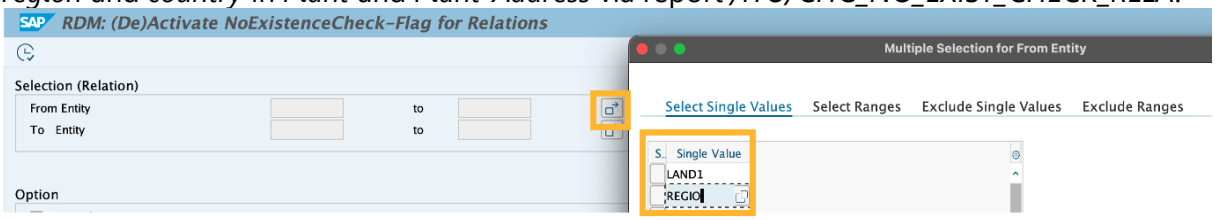
Use when attributes are derived via Entity Relationships.

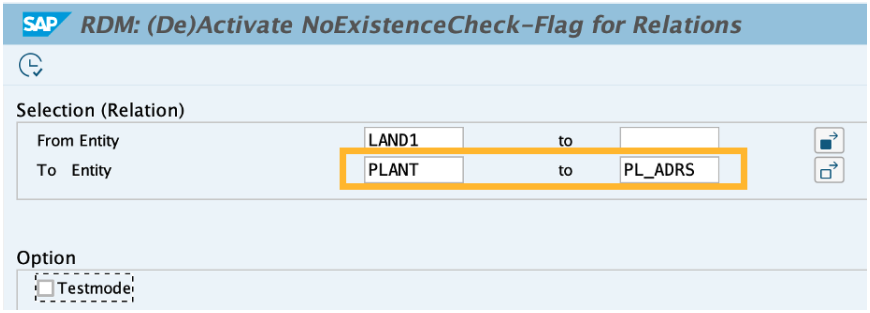
This function dynamically modifies table USMD0023 entries that are incorporated into validation checks through relationships. An example would be, if you need to validate a *Company* against a *Company Code* via a local or remote ERP system:



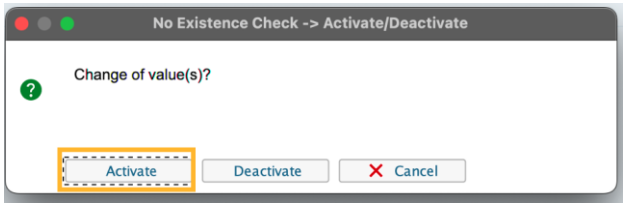
2. *Report /ITU/CHG_NO_EXIST_CHECK*
Use when attributes are defined directly within the entity type.
This function should be triggered when attributes are native to the entity type and are not assigned via relationships.

For the Plant-Address example we would deactivate the existence check for the attribute's *region* and *country* in *Plant* and *Plant-Address* via report */ITU/CHG_NO_EXIST_CHECK_RELA*.



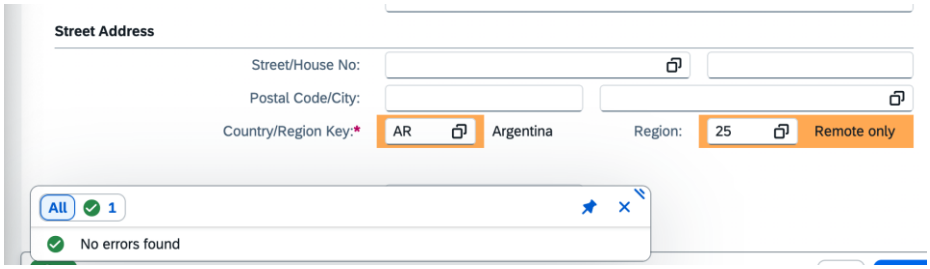


To disable the existence check, select activate.



Note: this is currently supported only for a specific list of objects.

For Region *25* that exists in the remote system only, no error message is thrown due to the deactivation of the existence check.



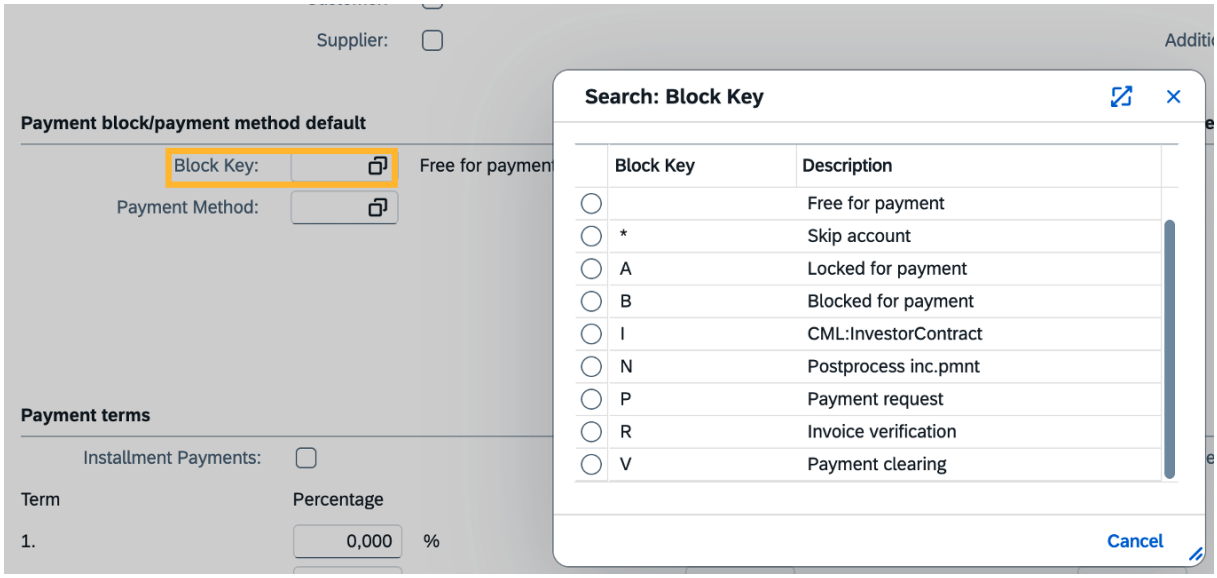
Note: When implementing Remote Validations, the usage of the Business Governance Alternative (also known as "Direct Table Update") for the validated object type must be strictly avoided. Performing a direct update to a target ERP system bypasses the critical Activation Logic and Foreign-Key checks. This creates a high risk of data inconsistency, as unapproved data from the MDG staging area could be committed to the ERP tables before completing the mandatory approval workflow. To ensure data integrity, all remote validations should be processed via the Local Staging Area that respect the validation rules.

3.7.3.2.1.3 Enable Custom Attributes for Remote Validations

To enable an attribute for remote validations a search help must be defined. This search help will use the function module *RFC_READ_TABLE* to fetch data from the remote system. This search help will be integrated into the UI configuration by adapting the corresponding feeder class. Below you will find the steps in detail.

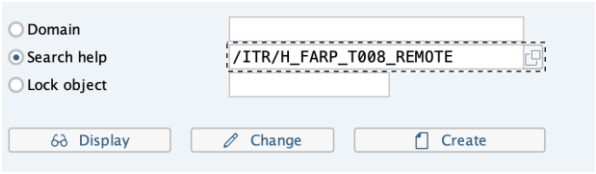
Example: Payment Block Key:
In this scenario, we want to fetch the *Payment Block Key* (data stored in *T008*) from a remote system for the entity type *Payment Term*.

1. Field: *ZSCHFDL (Payment Block Key)*
2. Entity type: *PAYTERM (Payment Term)*



Step 1: Create a new search help

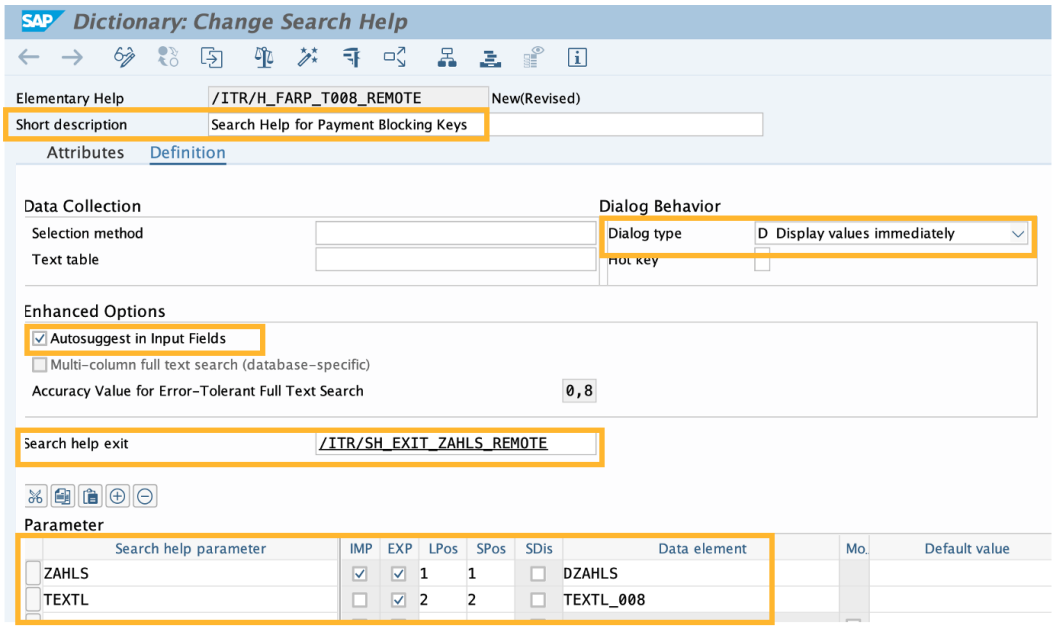
You need to create a specific search help that will handle the remote call with RFC_READ_TABLE. Open transaction *SE11* and enter the search help name */ITR/H_FARP_T008_REMOTE* and click create.



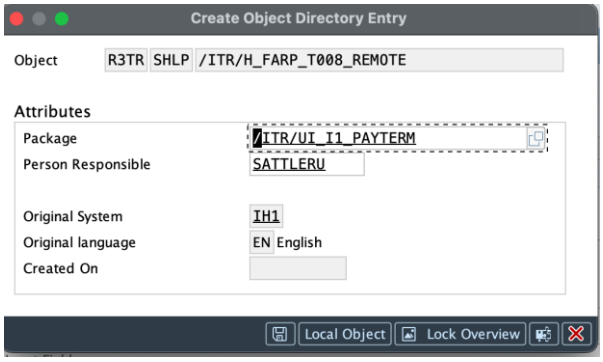
Select "*Elementary Search Help*".



- Enter a short description.
- Use Dialog type "*Display values immediately*" ("D").
- Check "*Autosuggest in Input Fields*".
- Enter */ITR/SH_EXIT_ZAHLS_REMOTE* as search help exit.
- Enter the search help parameters such as in the screenshot below.



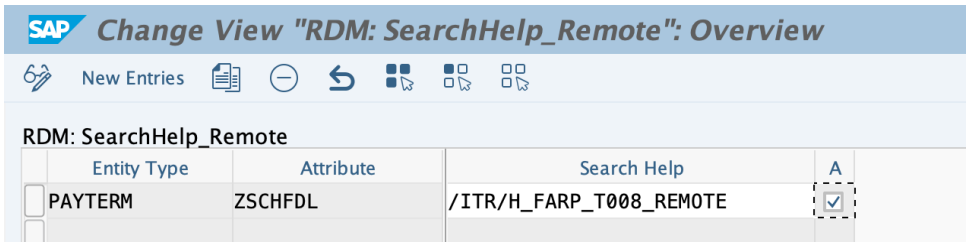
Save and add to a package.



Step 2: Register the search help

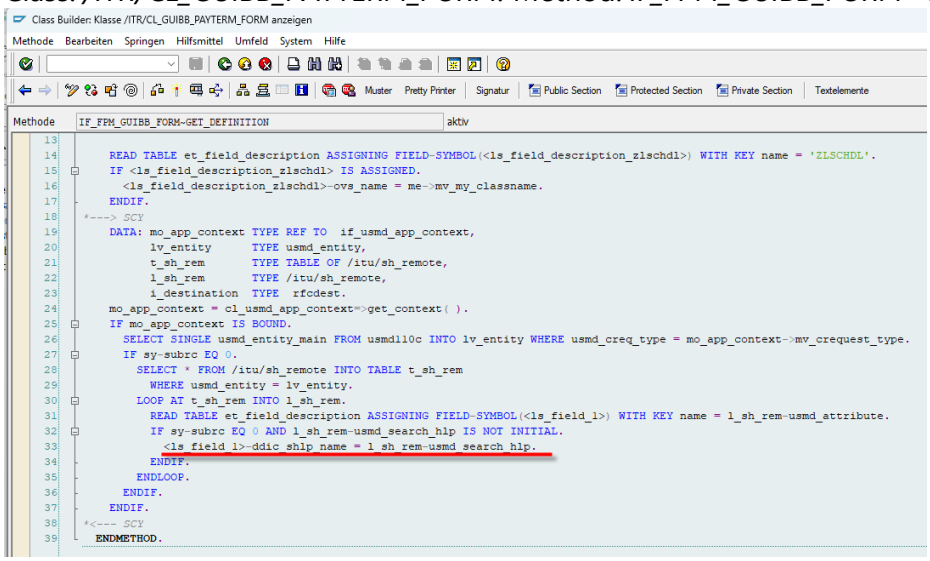
Once the search help */ITR/H_FARP_T008_REMOTE* is created, register it using transaction */ITU/PF4_SH_REMOTE*.

Map the field *ZSCHFDL* to the new search help and set the Activation flag.



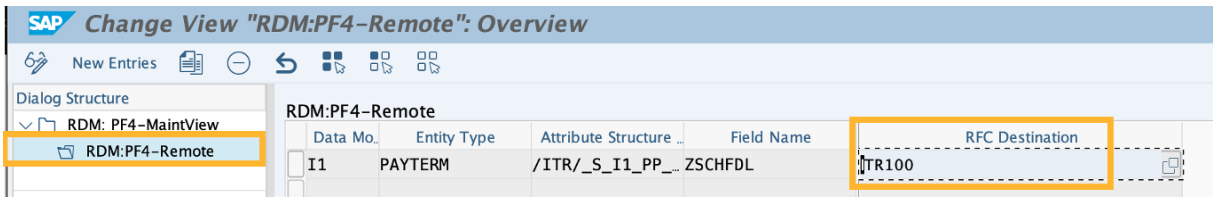
Step 3: Dynamic search help assignment in the feeder class

To ensure the UI recognizes the remote search help, you must modify the Feeder Class. Class: */ITR/CL_GUIBB_PAYTERM_FORM*. Method: *IF_FPM_GUIBB_FORM~GET_DEFINITION*



Step 4: Set Remote System

You must prepare the mapping table so that the system knows which RFC destination to call for a specific field. This ensures the value list matches the remote system's data.
Open transaction */ITU/PF4_CHOICE_RE*. Search for the Payment Term field *ZSCHFDL,* select PF4-Variant *Customizing* and click on *RDM:PF4-Remote* to enter the remote system.

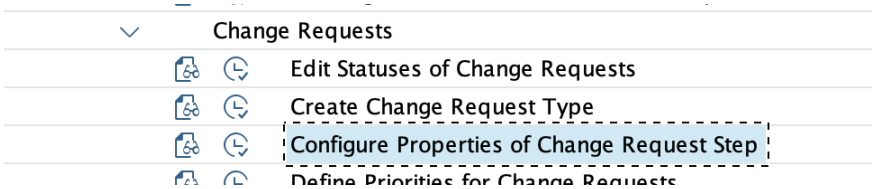


Step 5: Disable standard MDG validation logic

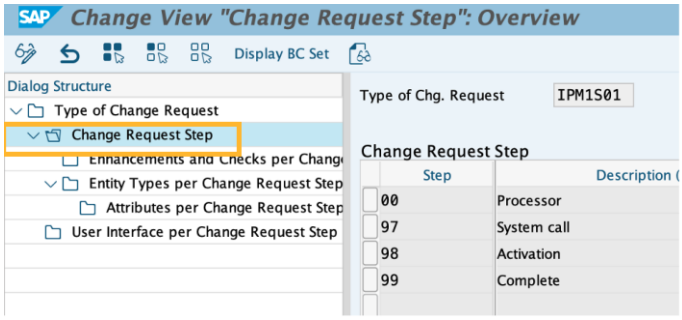
Because validation is now handled remotely via the new search help and dynamic assignment, you may need to bypass or suppress standard MDG check logic for these specific fields to prevent conflicts between local validation rules and remote data results.

To disable the standard field checks per change request type and step follow the steps below:

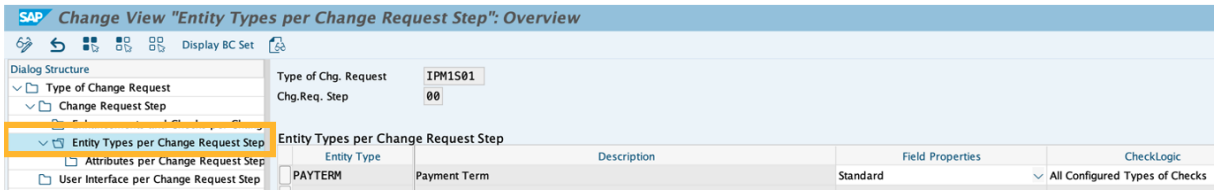
In *MDGIMG* go to Classic Mode in *SAP MDG → Central Governance → General Settings → Process Modeling → Business Activities → Change Requests → Configure Properties of Change Request Step*



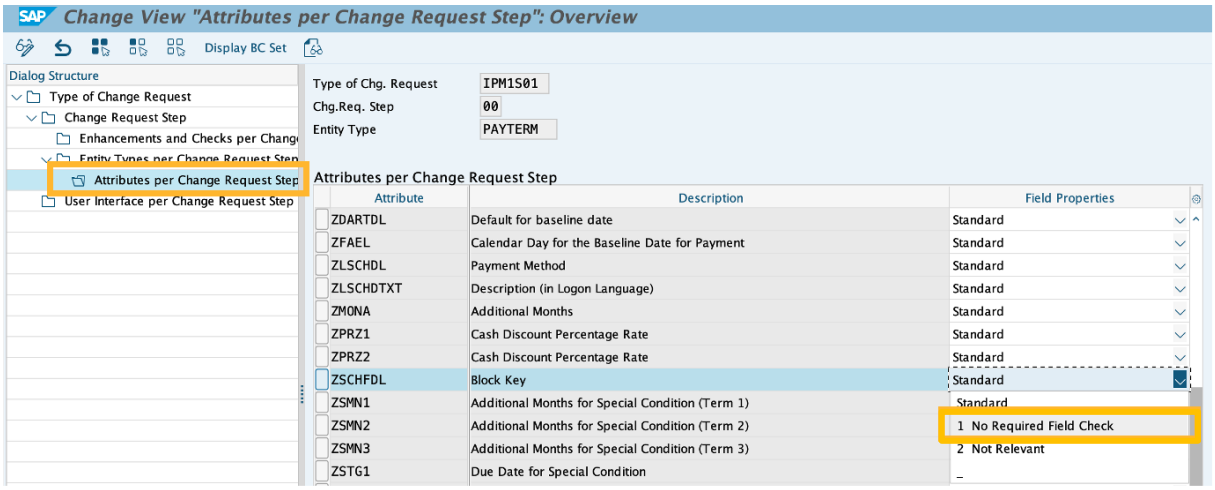
Select the corresponding change request type, in this example *IPM1S01*, then click on *Change Request Step*.



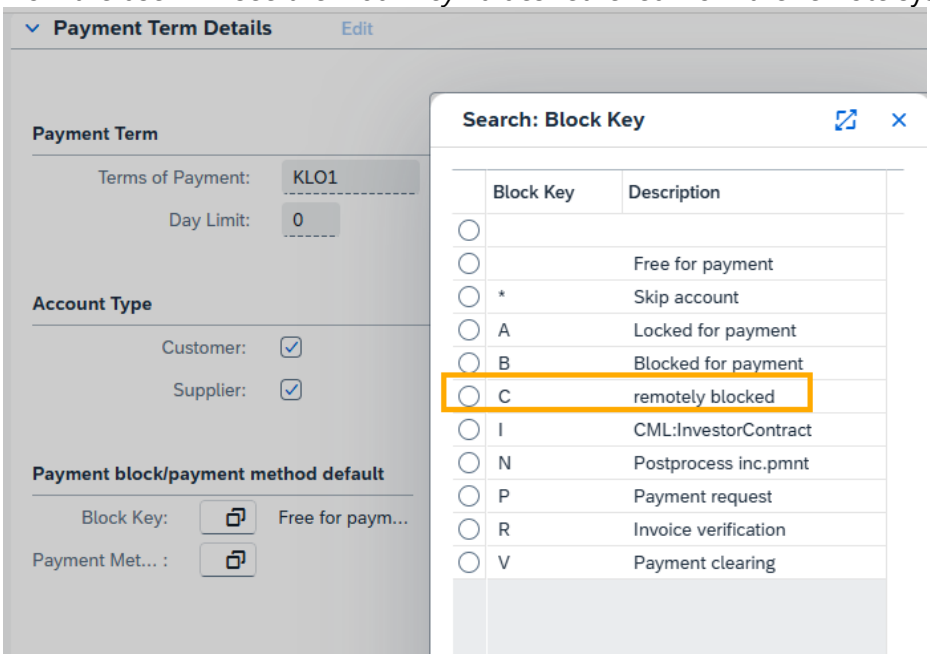
Select *step 00*, then go to *Entity Types per Change Request Step*.



Select entity type *PAYTERM* and go to *Attributes per Change Request Step*. Then for the attribute *ZSCHFDL Block Key* change the field property to *No Required Field Check*.



Now the user will see the *Block Key* values retrieved from the remote system:



3.7.3.2.1.4 List of supported Entity-Types and Attributes

The following table shows all supported entity-types and attributes for remote validations.

RDM-Object	Entity-Type	Attributes
Plant	PLANT	FABKL PLNTSPRAS PL-VKORG
	PL_ADRS	AD_LAND1

		AD_REGIO
Tax Code	KALSM_TAX	LAND1_TAX
Payment Term Details	DAYLIMIT	PAYMNTTRM
Company Code	COMPCODE	CCODECURR CCODKTOPL CCODLAND1 CCODESPRAS
	CC_ADRS	CC_LAND CC_REGIO
Document Type	BLART	STBLA

3.8 Adjust User Interfaces

User Interfaces should be adjusted based on the Governance Scope and the user requirements.

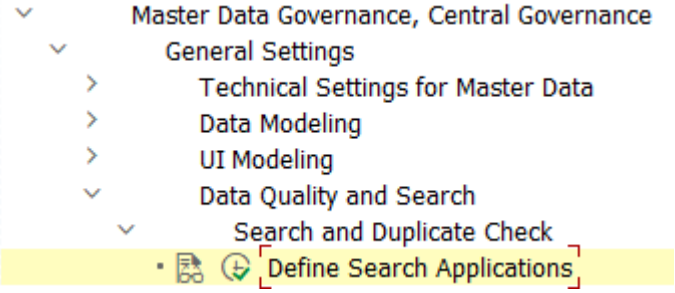
The Master Data Framework (MDF) provides capabilities for the definition of

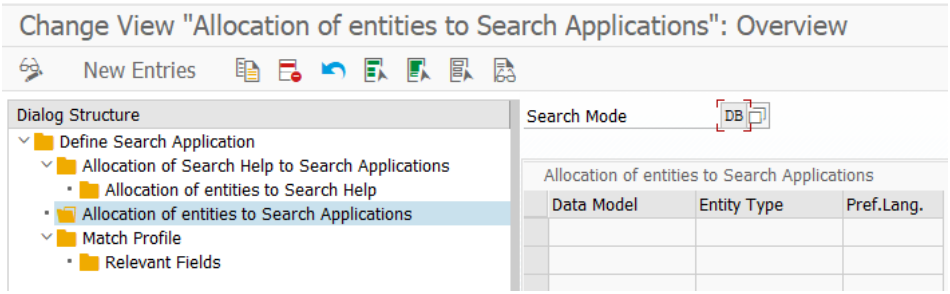
- Field properties
- Search configurations
- Default Values
- Validations and
- Derivations

Please check the document "Technical Documentation of the MDF Configuration Management" to learn more about this.

On top of these functionalities provided, the SAP Floorplan Manager (FPM) can be leveraged for additional adjustments.

Note: To define Search Modes for individual entities SAP provides the configuration option below. Make sure that you are aware of side effects to other entities described below.





Make sure that this configuration only is used when you define an allocation for every entity type used in and SAP MDG application. Otherwise, an entry for an RDM object might impact the search for other entities. For example, the search mode for an SAP MDG-F based object like Cost Center, which should be HA (HANA) based, might be changed to DB (Database) by the SAP MDG framework.

3.9 Configure Rule Based Workflow

Prerequisite: You have configured the general settings for SAP Business Workflow in Customizing for SAP NetWeaver under Application Server -> Business Management -> SAP Business Workflow.

The configuration of the rule-based workflow is done in transaction MDGIMG: General Settings -> Process Modeling -> Workflow -> Rule-based Workflow -> Configure Rule-Based Workflow.

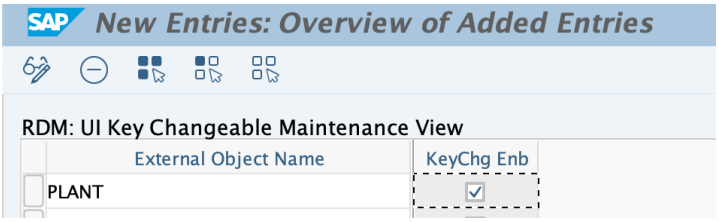
Consider the following BC Sets from section 3.3 "Activate Business Configuration Sets" as templates for your configuration (**check also the notes mentioned in this section**):

- /ITR/MDG_RDM_CREQUEST_<n> MDG RDM Change Request Types
- /ITR/MDG_RDM_WORKFLOW_<n> MDG RDM Workflow

3.10 Configure Changeable Keys

This functionality ensures that an object identifier can be adjusted after initial creation if required. The keys can only be changed if the processor of the current workflow step is allowed to make changes to the data. For more information refer to the Functional Documentation in section 2.1.7.

In order to allow changeable keys, open the view /ITR/UI_KEYCHG_V. Enter the object name and select the checkbox to activate changeable Keys.



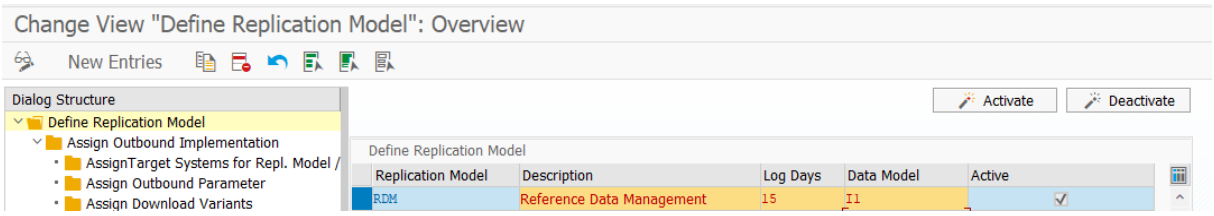
3.11 Set Up Data Replication

To configure the Data Replication for RDM the "Data Replication Framework" (DRF) needs to be configured as well as the technical settings for ALE or Webservices.

Note: Make sure that you also define the data replication to the MDG client which will make sure that related ERP customizing tables are synchronized. For the IT Governance Alternative this should be configured via the Local Staging Area in the MDG development system whereas for the Business Governance Alternative this should be configured by using the direct replication to the MDG production system (with optional connection to the test as well as the development system). See chapter 5.1.5 for a configuration example.

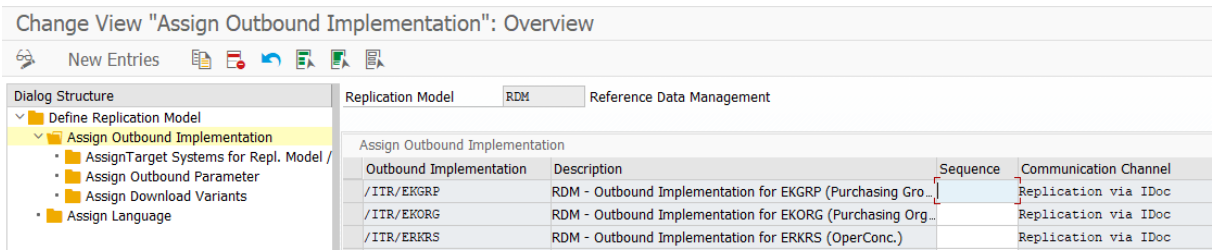
3.11.1 Data Replication Framework settings

Use transaction DRFIMG to maintain Business Systems and the Data Replication Model.



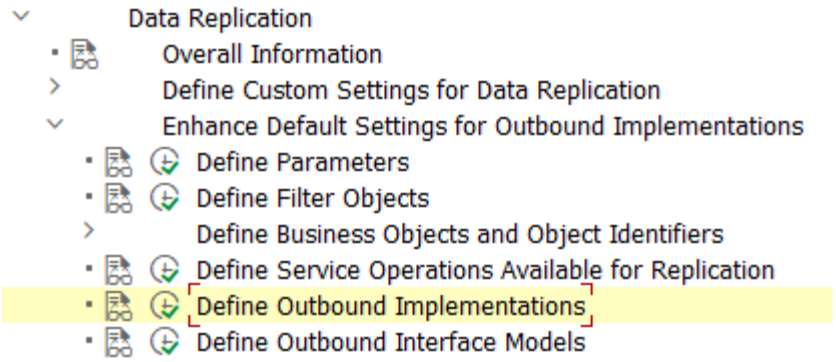
3.11.1.1 Configuration of Outbound Implementations

Configure Outbound Implementations for your model based on the scope of your RDM implementation and define a communication channel.



For some RDM object types the assigned outbound implementations follow an object specific implementation. If you do not have project specific requirements which require the usage of these, please change the implementation classes for the following objects to the generic implementation:

Transaction: DRFIMG



Object Type	Outbound Implementation	Outbound Implementation Class
Currency	/ITR/CURRC	/ITR/CL_CURRENCY_OUT_IDOC_GEN
Purchasing Group	/ITR/EKGRP	/ITR/CL_EKGRP_OUT_IDOC_GEN
Purchasing Organization	/ITR/EKORG	/ITR/CL_EKORG_OUT_IDOC_GEN
Account Group Customer	/ITR/KTOKD	/ITR/CL_KTOKD_OUT_IDOC_GEN
Account Group Vendor	/ITR/KTOKK	/ITR/CL_KTOKK_OUT_IDOC_GEN
Account Group Gen. Led.	/ITR/KTOKS	/ITR/CL_KTOKS_OUT_IDOC_GEN
Country	/ITR/LAND1	/ITR/CL_LAND1_OUT_IDOC_GEN
Material Group	/ITR/MATKL	/ITR/CL_MATKL_OUT_IDOC_GEN
Unit of Measure	/ITR/MSSIE	/ITR/CL_MSSIE_OUT_IDOC_GEN
Payment Term Day Limit	/ITR/PAYMD	/ITR/CL_DAYLIMIT_OUT_IDOC_GEN
Payment Term	/ITR/PAYMN	/ITR/CL_PAYMNTTRM_OUT_IDOC_GEN
Plant	/ITR/PLANT	/ITR/CL_PLANT_OUT_IDOC_GEN
Product Hierarchy	/ITR/PRODH	/ITR/CL_PRODH_OUT_IDOC_GEN
Product Hierarchy	/ITR/PRODL	/ITR/CL_PRODHL_OUT_IDOC_GEN
Product Hierarchy	/ITR/PRODN	/ITR/CL_PRODHN_OUT_IDOC_GEN
Region	/ITR/REGIO	/ITR/CL_REGIO_OUT_IDOC_GEN
Sales Org	/ITR/SAORG	/ITR/CL_SALESORG_OUT_IDOC_GEN

Note: You do not have to change the outbound implementation classes for other objects. They follow a generic approach even if the classes do not have a "_GEN" suffix.

3.11.1.2 Replication of change request related information

Change Request related information will be replicated to target systems by configuring the parameter RDM_CR_TF using /ITR/RDMIMG -> Configure Data Replication -> Configure Business Alternative. This configuration is target system dependent and needs to be configured for each target system if required for all systems. Example for a target system called "ID1400":

RDM: Activate Direct Table Update

Business System	Logical sy...	Entity Type	A
ID1400	ID1400	RDM_CR_TF	☑

3.11.1.3 Deletion of assignments

Deletion of assignments will be replicated to target systems by configuring the parameter RDM_DE_TF using /ITR/RDMIMG -> Configure Data Replication -> Configure Business Alternative. This configuration is target system dependent and needs to be configured for each target system if required for all systems. This for example allows to delete language dependent texts or an entry in the sequence of the object type "Planning Scope". Example:

RDM: Activate Direct Table Update

Business System	Logical s...	Entity Type	A
ID1400	ID1400	RDM_DE_TF	☑

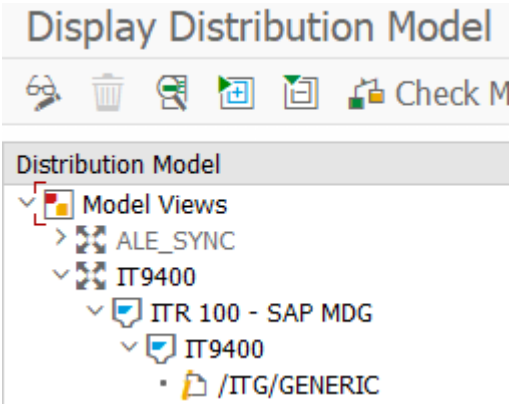
Note: Please be aware that these deletions are not supported for all object types in all cases. Please consider clarifying remaining questions by contacting support@itego.de (subject: "Deletion of assignments").

3.11.2 Data Replication using Application Link Enabling (ALE)

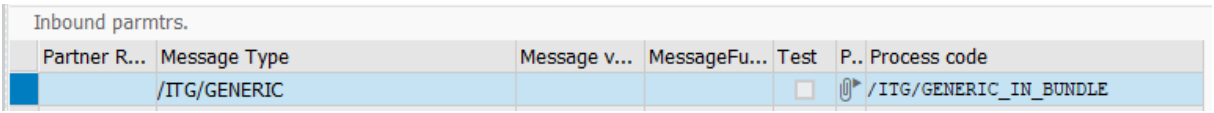
If the Data Replication for reference data objects shall be implemented using ALE, the communication needs to be setup between the RDM system and the receiving systems.

3.11.2.1 ALE Configuration for Generic Message Type

For most object types RDM uses the generic message type /ITG/GENERIC which needs to be configured using transactions BD64 and WE20. The figures below show the most important settings. For a more detailed description please have a look at the configuration example provided in chapter 5.1 Data Replication based on SAP ALE.
BD64 (example)



WE20 (example from receiving system)



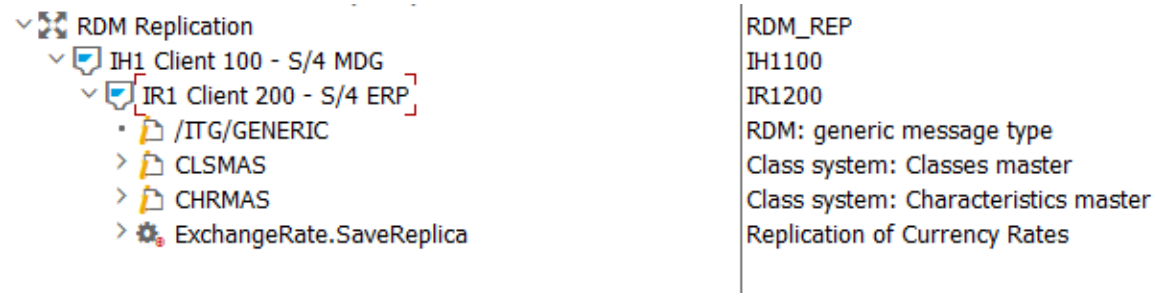
3.11.2.2 ALE Configuration for Object Specific Message Types

For some RDM object types SAP delivers standard message types which need to be configured. These are:

- Classification (Classes and Characteristics): CLSMAS and CHRMAS
- Exchange Rates: EXCHANGE_RATE

If these objects are part of you implementation scope, please add these message types as shown in the previous chapter.

Example from Transaction BD64 (Sender):



Transaction WE20 (Sender), after generation of partner profiles:

Outbound

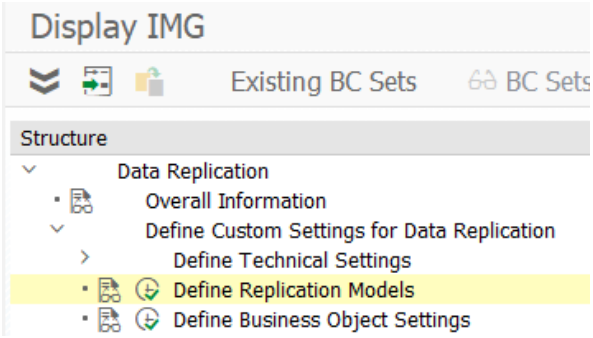
Partner R...	Message type	Message v...	Function	Test	Receiver ...	I...	Pa...	Basic type
	CHRMAS				A000000001		100	CHRMAS05
	CLSMAS				A000000001		100	CLSMAS04
	EXCHANGE_RATE				A000000001		100	EXCHANGE_RATE01

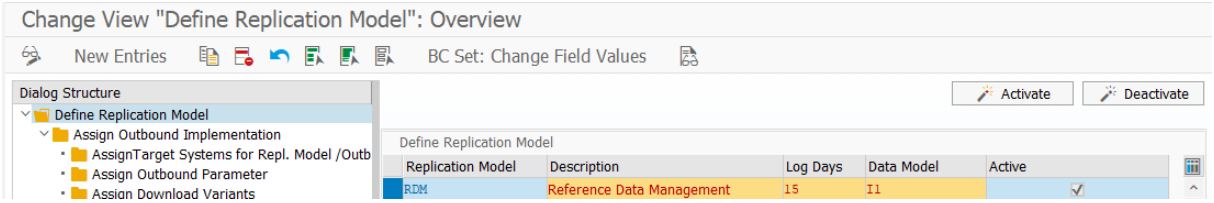
Transaction WE20 (Receiver), after generation of partner profiles:

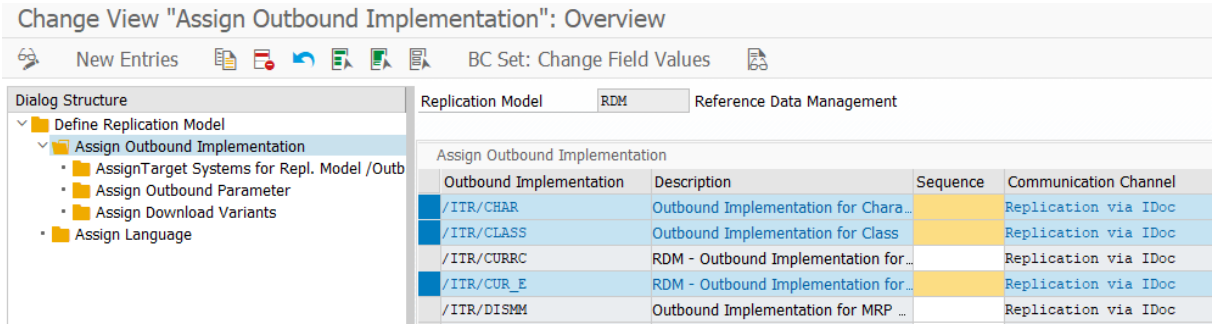
Inbound

Partner R...	Message Type	Message v...	Function	Test	P..	Process code
	CHRMAS					CHRM
	CLSMAS					CLSM
	EXCHANGE_RATE					BAPP

For these please also make sure that they have been added in the Replication Model (transaction DRFIMG):







3.11.3 Data Replication using WebServices

RDM implements SOAP as a network protocol, which uses XML to transfer the reference data. With this it enables an exchange of data between heterogeneous applications on different systems. The web services are described in WSDL files (Web Service Description Language) which are provided by the server. Using this information, the client application obtains information about the offered web services.

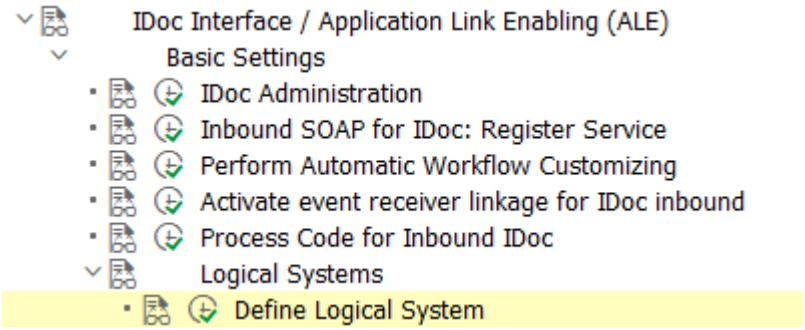
The web services offer two functionalities:

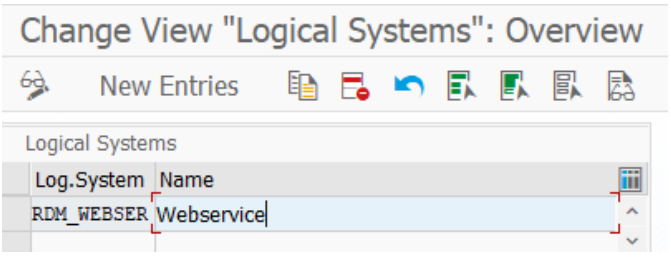
- Push (Send from RDM)
- Pull (Call from outside)

Examples:

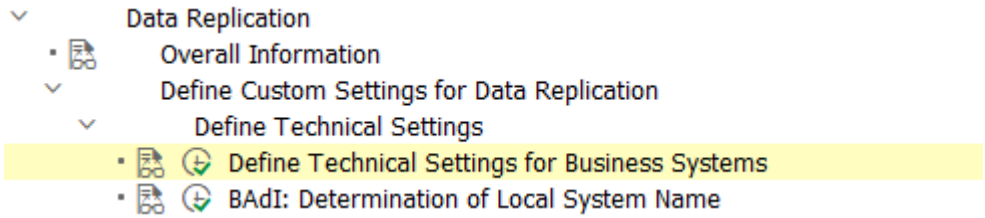
- Push (generic): /ITR/RDM_WS_SEND_OBJECT
- Pull (generic): /ITR/RDM_WS_GET_OBJECT
- Pull (object-specific): /ITR/RDM_WS_GET_PLANT

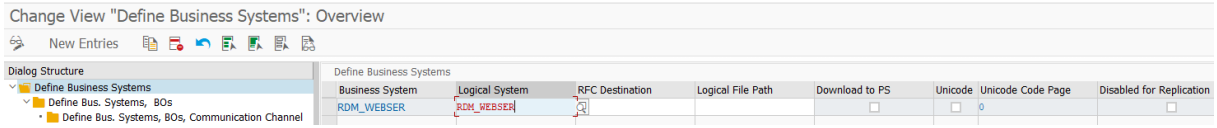
Transaction SALE: Define Logical System

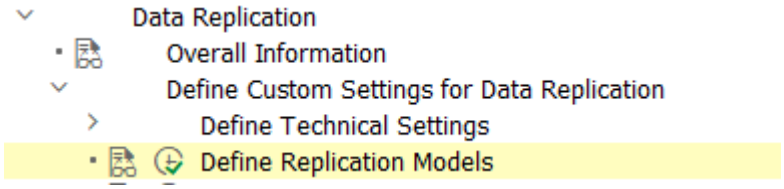


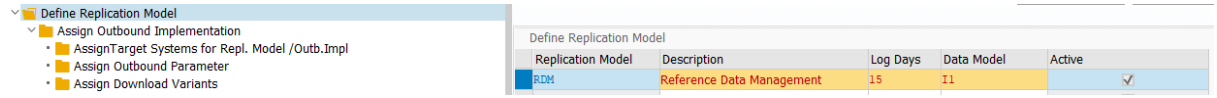


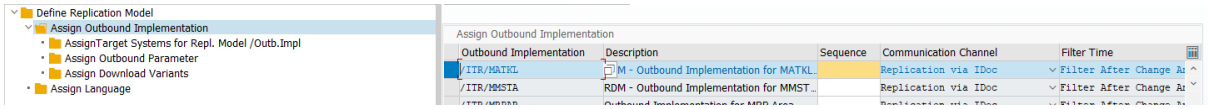
Transaction DRFIMG: Define Technical Settings and Replication Models

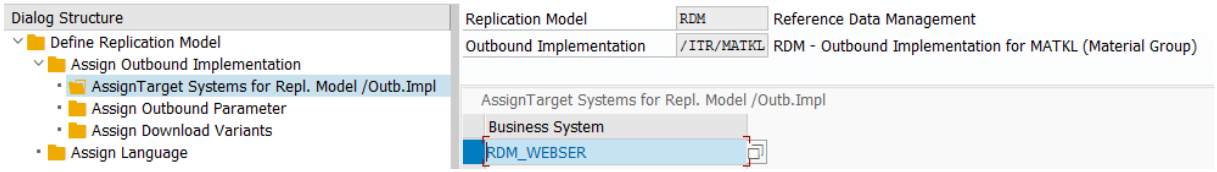


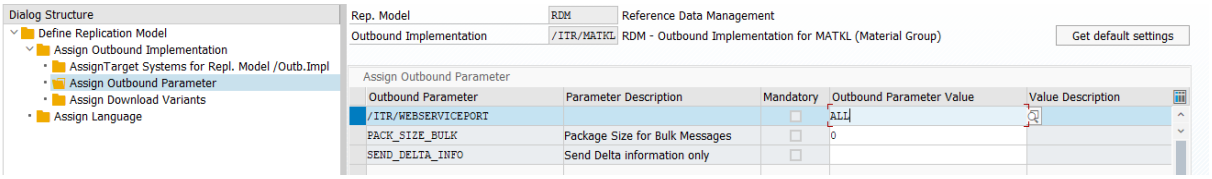






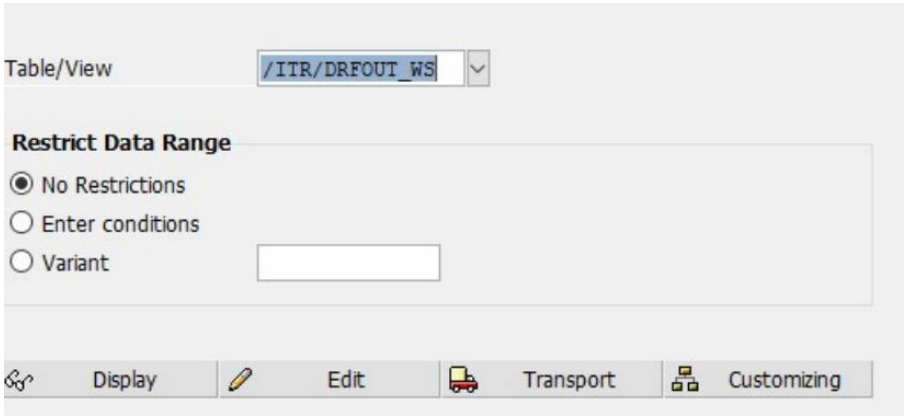


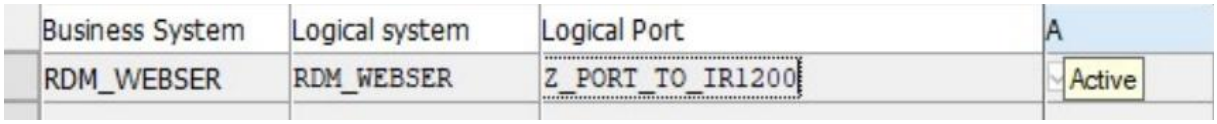




Using outbound parameter /ITR/WEBSERVICEPORT you can define which logical port will be used. If you define Value “ALL” all ports defined will be used.

Maintenance of ports via table /ITR/DRFOUT_WS





For necessary settings with regards to the WebService Environment using transaction SOAMANAGER please have a look at chapter 5.2 Data Replication based on Webservices.

For more information, please contact support@itego.de (subject: "Configuration Guide RDM: Webservices")

3.11.4 Key Mapping

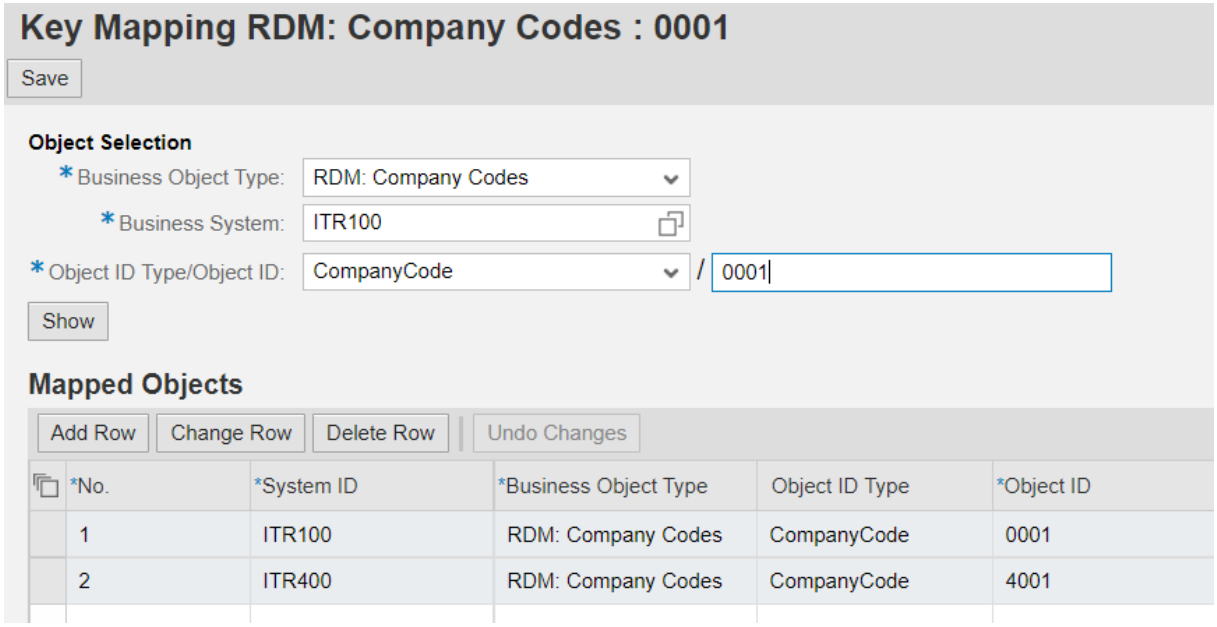
SAP MDG provides the following transactions to maintain and analyze Key Mapping:

- MDG_KM_MAINTAIN - Maintain Key Mapping
- MDG_ANALYSE_IDM - Search Key Mapping

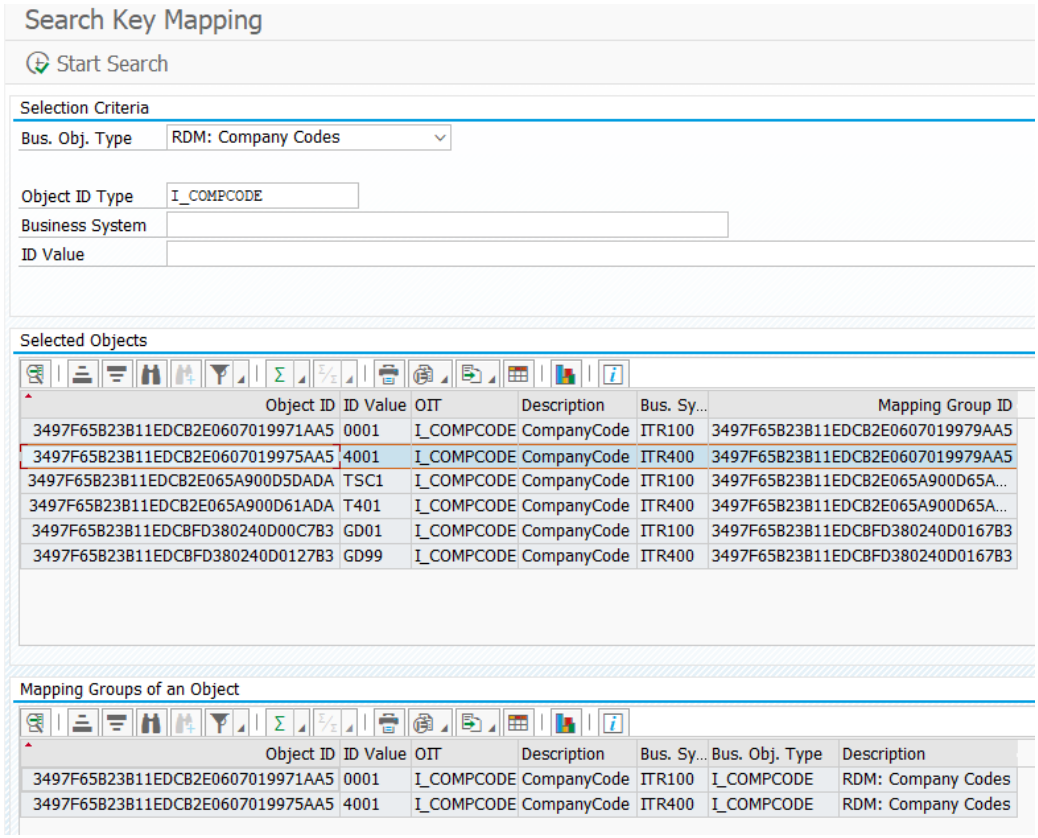
Use these transactions to maintain or search the Key Mapping for reference data object types by using:

- Business Object Type: "RDM: <Object Type>". Example: "RDM: Company Codes"
- Object ID Type: "I_<Entity Type> Description". Example: "I_COMPCODE CompanyCode"

Example for maintenance of Key Mapping:



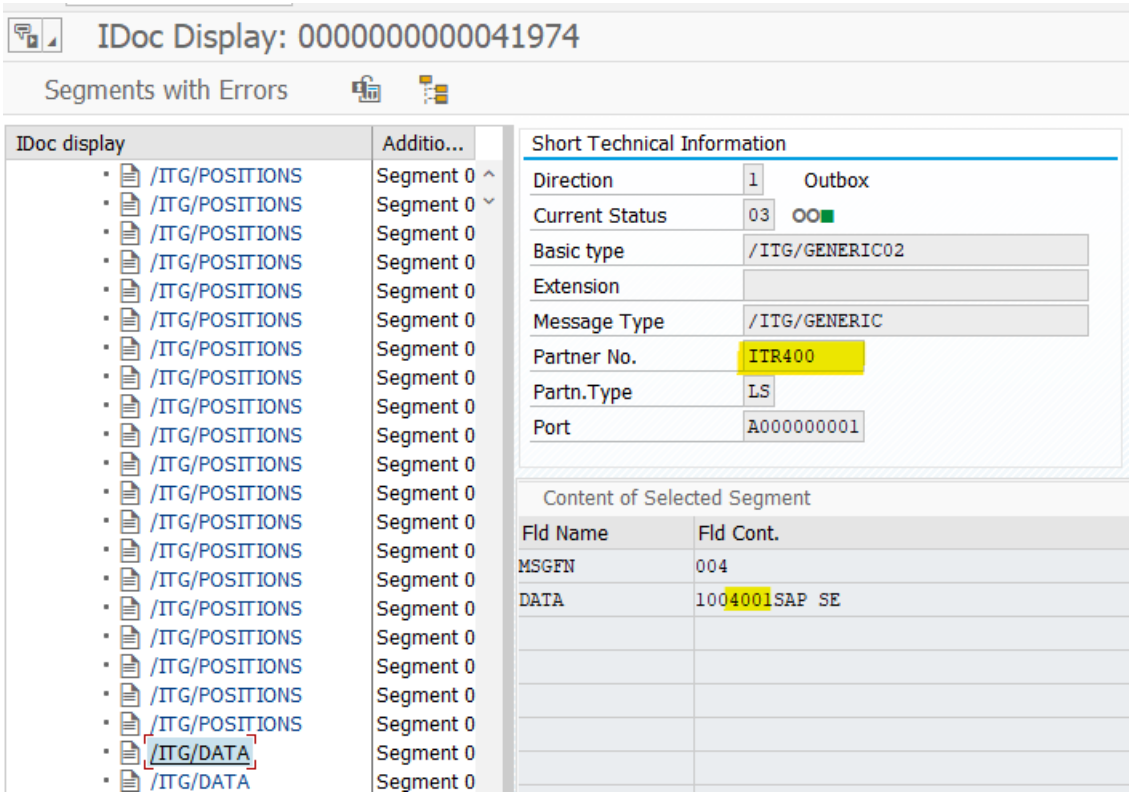
Example for Search Key Mapping:



Export to Spreadsheet:

	B	E	G
1	ID Value	Business System	Description of Business Object Type
2	0001	ITR100	RDM: Company Codes
3	4001	ITR400	RDM: Company Codes

In the data replication the maintained values will be mapped based on the receiver system. Example, based on the maintained mappings above (transaction BD87):



3.11.5 Value Mapping

SAP provides the transaction "VMIMG - Value Mapping Customizing" to maintain Value Mappings. Use this transaction to maintain Value Mapping for reference data object types by using:

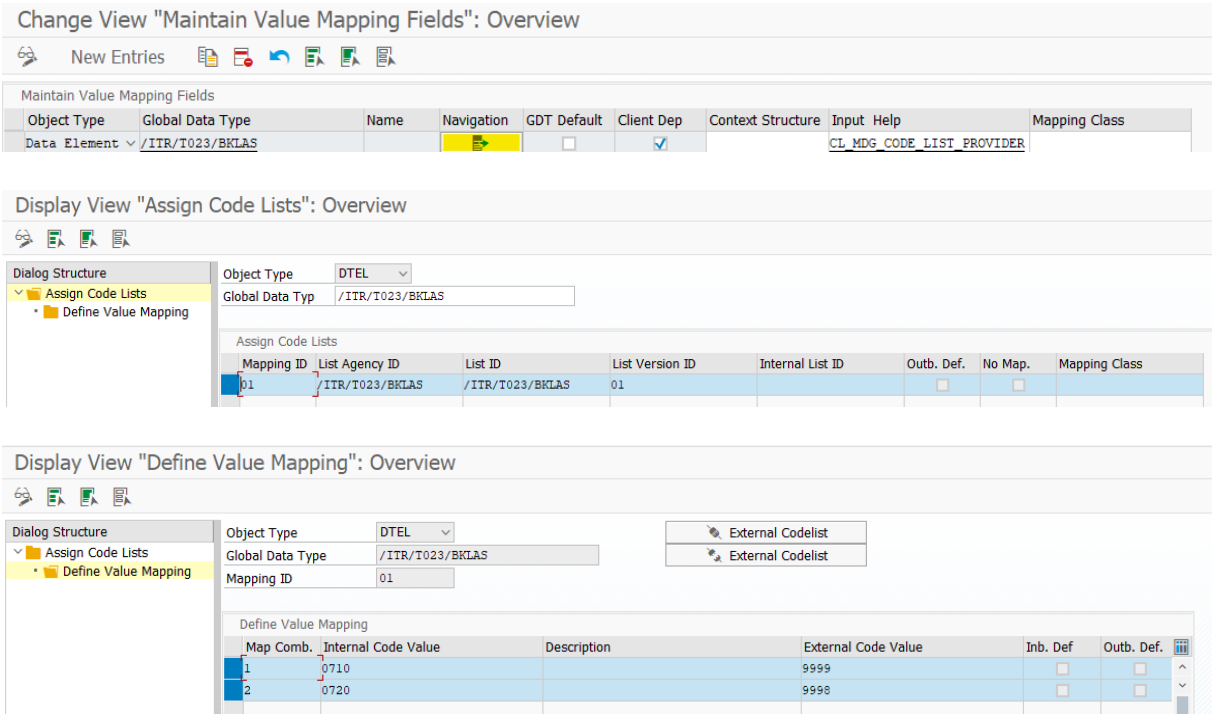
- Type: Data Element
- Global Data Type (GDT): "/ITR/<table>/<attribute>". Example: "/ITR/T023/BKLAS" (Valuation Class for Material Group).

Examples for other attributes:

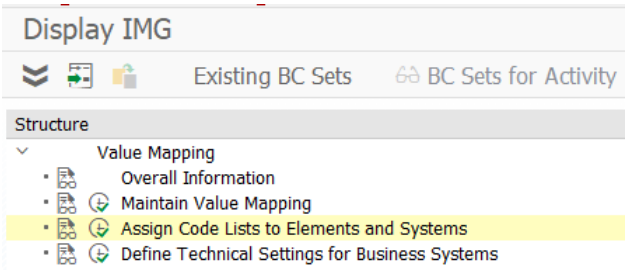
- List ID: "/ITR/T023/BKLAS"
- List Agency ID: "/ITR/T023/BKLAS"

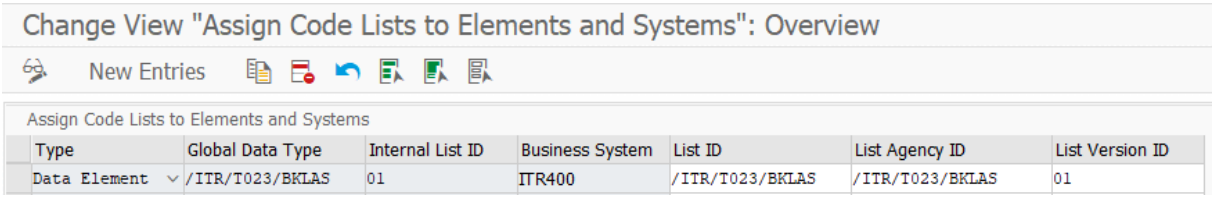
- List Version ID: 01

Maintained example:



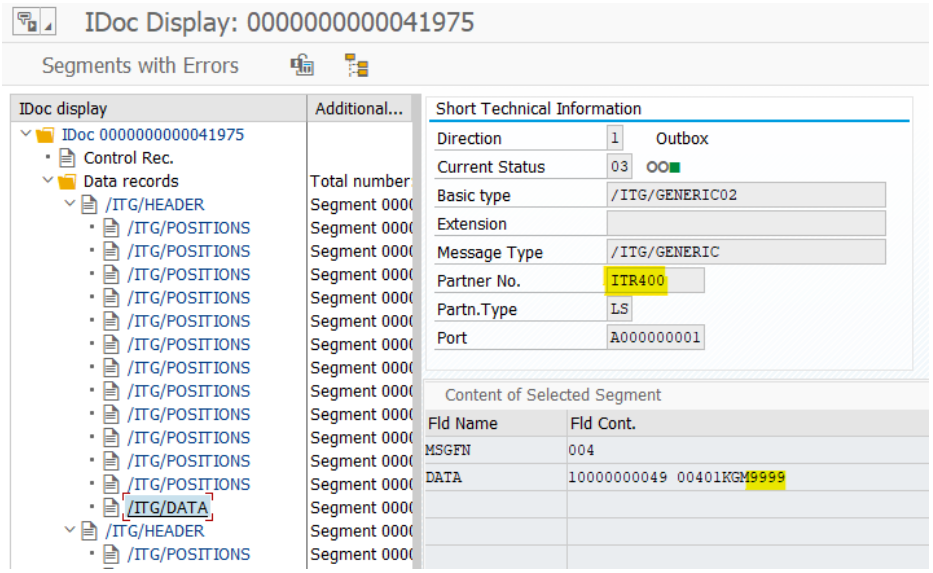
External codes (for system ITR400) for internal codes. Example: 9999 for 0710.



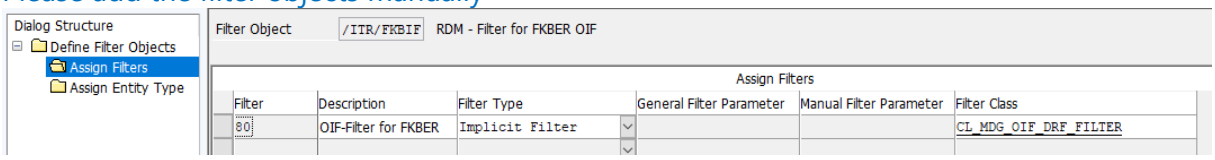


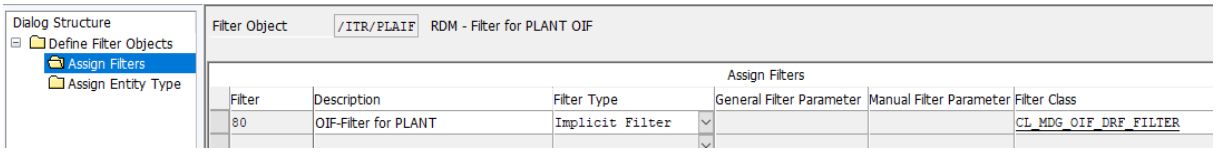
Code list are now in this example assigned for Business System ITR400. Make sure that this entry is added after you added the mapping in step “Maintain Value Mapping”. Otherwise, the system will not allow you to add this entry here.

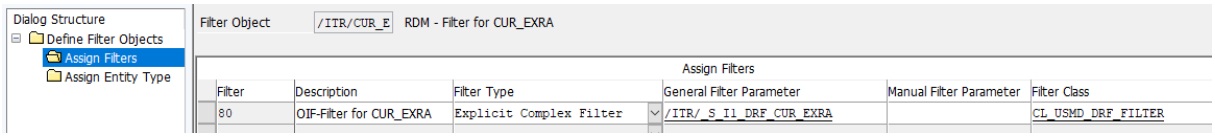
In the data replication the maintained values will be mapped based on the receiver system. Example, based on the maintained mappings above (transaction BD87):



Please add the filter objects manually







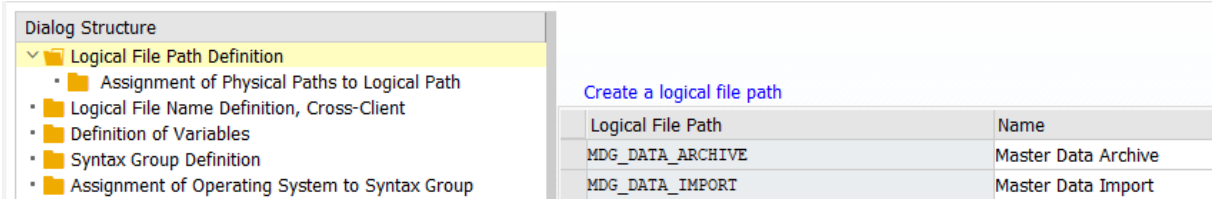
3.12 Set Up Data Transfer

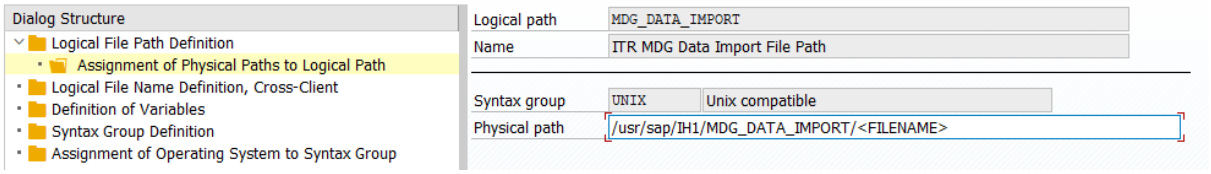
Data Transfer needs to be configured for initial load and consist of the configuration of MDMGX (on an SAP business system) and Data Import (on the MDG RDM system).

The configuration of MDMGX is done through the activation of the BC Sets /ITR/RDM_MDMGX_<n> (see: 3.3 "Activate Business Configuration Sets") on the SAP business system. This enables users to extract reference data objects from this system using transaction MDMGX.

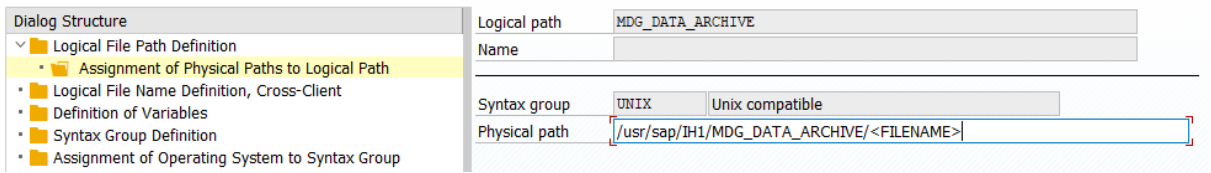
Please also check section 3.3 "Activate Business Configuration Sets" for the activation of the Data Transfer on the MDG system (BC Sets /ITR/MDG_RDM_DT_<n>). These BC Set deliver the necessary object type definitions. After this configure Data Transfer in transaction MDGIMG: Data Transfer -> Define File Source and Archive Directories for Data Transfer. For details check the IMG documentation for this activity and the additonal configuration example below.

Transaction FILE:

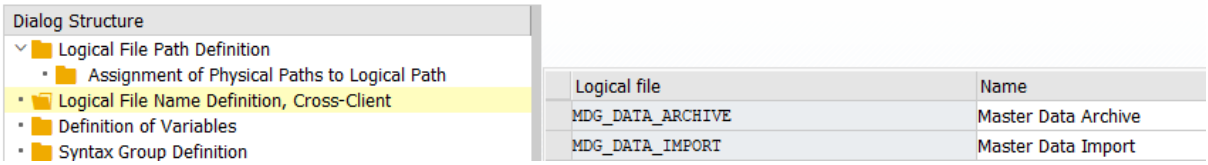


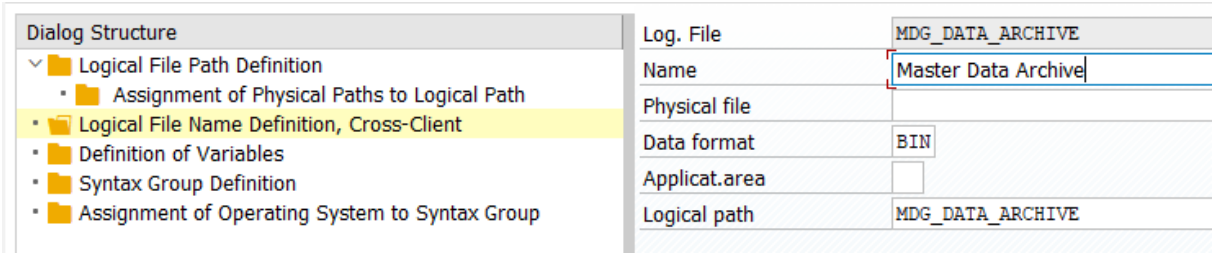


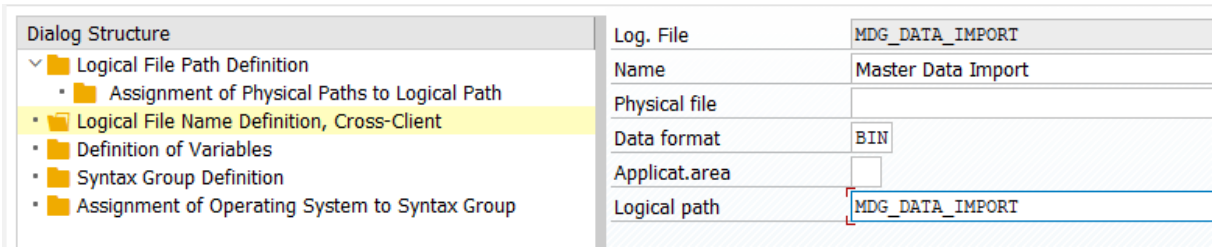
e.g.: /usr/sap/<systemID>/MDG_DATA_IMPORT/<FILENAME>



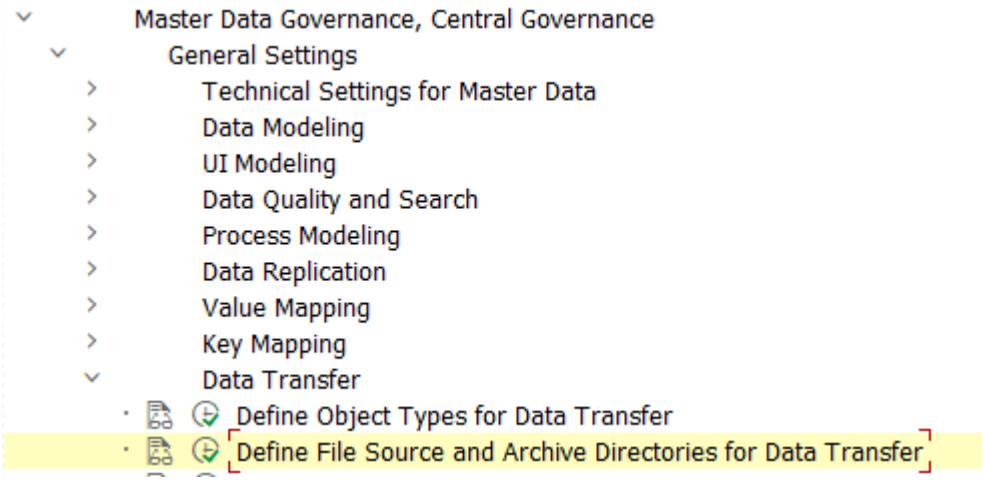
e.g.: /usr/sap/<systemID>/MDG_DATA_ARCHIVE/<FILENAME>

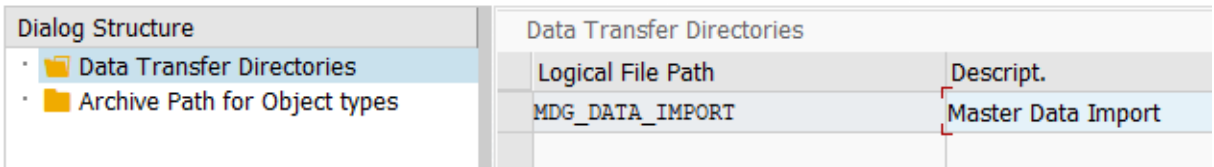


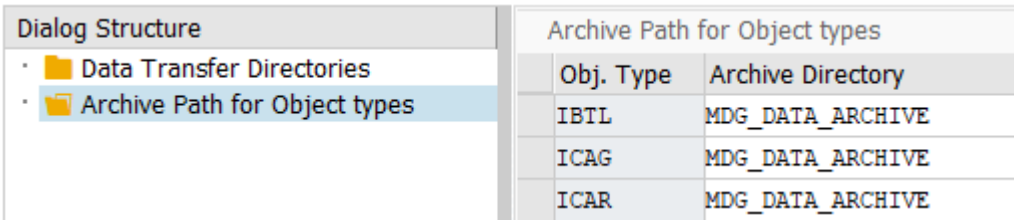




Using transaction MDGIMG configure the usage of these directories:







Using MDMGX and Data Import all reference data object types which do not belong to a hierarchy can be extracted. See "Reference Data Management for SAP MDG - Functional Documentation" for usage details and the next section for details about the configuration and load of a product hierarchy.

3.13 Configure Product Hierarchy

If the product hierarchy is within the scope of maintained reference data object types additional steps need to be performed on the MDG RDM system as well as on the SAP business application system.

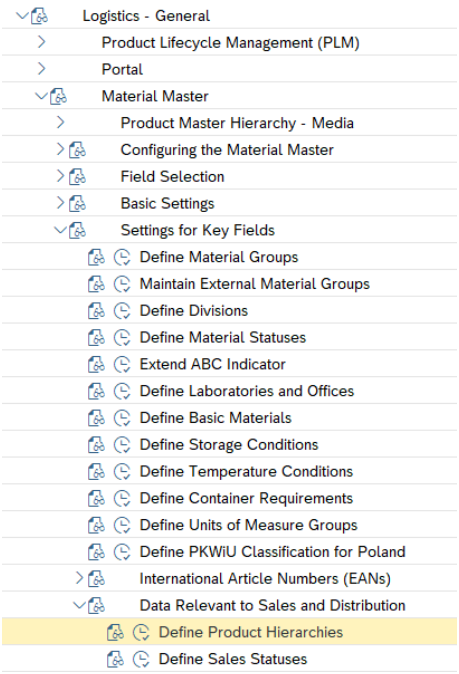
On the MDG RDM System the following activities need to be performed:

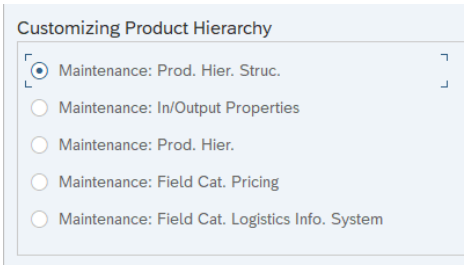
- Verify the product hierarchy usage in your business applications
- Configure the product hierarchy usage in RDM
- Check Interlocking
- Define Edition
- Creation of a Product Hierarchy Name
- Maintenance of number ranges
- Data Import (File Upload) after Data Export from the business application system

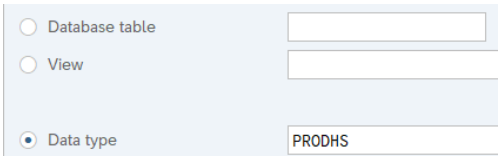
3.13.1 Verify the product hierarchy usage in your business applications

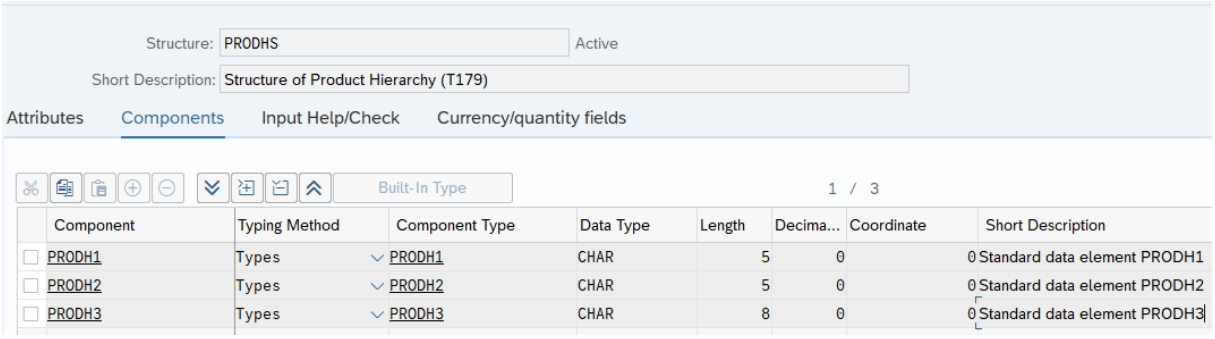
The RDM standard delivery assumes an SAP standard configuration of three levels and number ranges must be maintained according to the Product Hierarchy Set Up in the receiving business applications. The Set Up of the Product Hierarchy in an SAP business application is done through the definition of structure PRODHS.

To check the configuration use transaction SPRO and check the following entries:







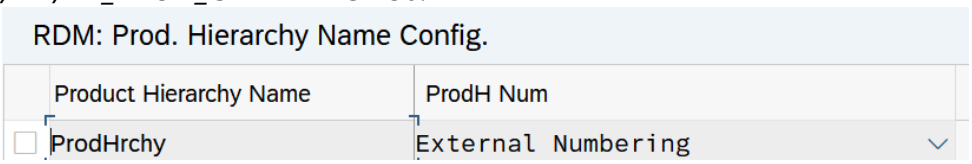


The figure above shows the SAP standard configuration. Please refer to the following SAP note for more information: [1070939 - Consulting: Changing the product hierarchy structure PRODHS - SAP for Me].

3.13.2 BC-Set Activation

Execute the following steps to activate the required BC Sets and establish the base configuration for the application.

Open Tx-SCPR20 and enter */ITR/MDG_RDM_APPL_<n>*. Click Execute and check table */ITR/PH_ENUM_GEN* in Tx-SM30.



Technical Note:
Due to the technical delivery class 'A' of the client-dependent tables *USMD020C* and USMD020T, they cannot be transported via standard BC Sets or R3TR/TABU entries.

To initialize the required control entries for the Product Hierarchy, you must manually execute the following program in each target client:

- Program: */ITR/INSERT_PRODH_DRF_REPL*

This step is mandatory to ensure the Data Replication Framework (DRF) settings for the product hierarchy are correctly populated.

3.13.3 Configure the product hierarchy usage in RDM.

3.13.3.1 Configure Product Hierarchy Level

Use transaction /ITR/RDMIMG and configure RDM Product Hierarchy level descriptions by selecting "Configure product Hierarchy".



Add Product Hierarchy Level and Description. Save the changes.

Product Hierarchy Level Config Table

Prod.H.Lvl	Description (medium text)
1	Division
2	Brand
3	Segment

Standard – 3 Level | Extended - 6 Lil.

1	Division
2	Brand
3	Segment
4	Type
5	Series
6	Model

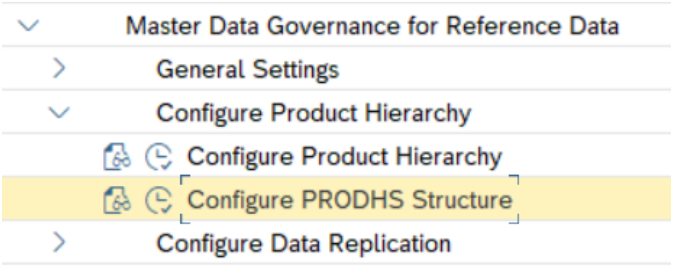
Note: In the standard delivery all levels are represented by Product Hierarchy Nodes. Optional configuration for Product Hierarchy Sub Nodes is possible and would be activated for the next level which is not defined in the configuration shown above. Example: a configuration of Division and Brand means that a level for "Segment" can be represented by Product Hierarchy

Nodes. This might be useful If additional attributes should be defined for the "leaf nodes" of the Product Hierarchy.

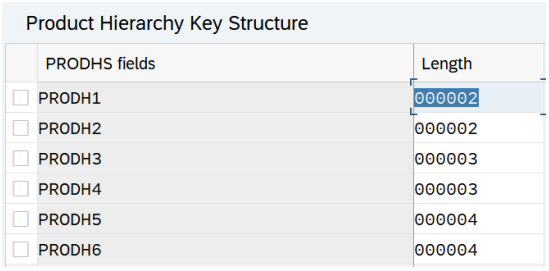
3.13.3.2 Configure Product Hierarchy Structure

It is possible to define up to 9 levels and the number should add up to not more than 18 (0000018) characters.

Use transaction /ITR/RDMIMG.
Select "Configure PRODHS Structure".

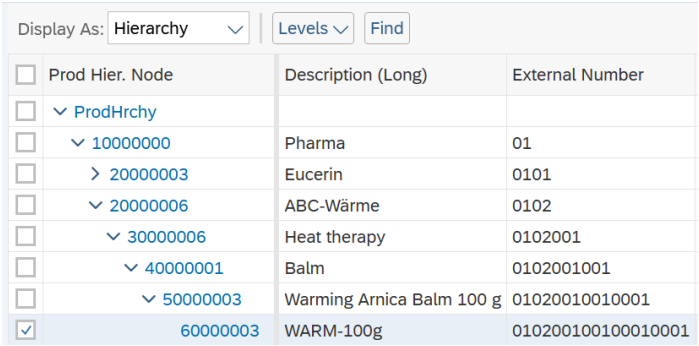


Enter PRODH Field and length e.g. 6 level Length: 2-2-3-3-4-4.



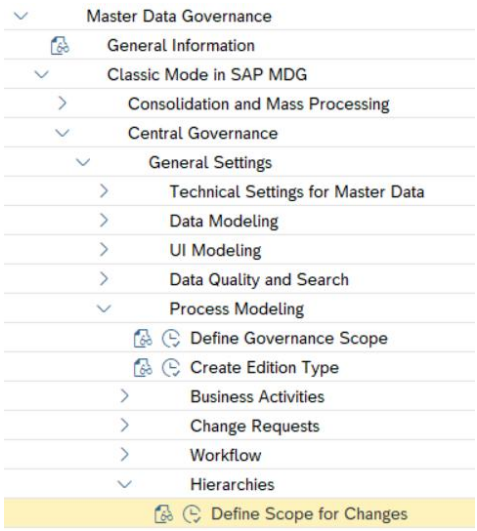
PRODH1	000002
PRODH2	000002
PRODH3	000003
PRODH4	000003
PRODH5	000004
PRODH6	000004

Example: Six-Level Hierarchy 2-2-3-3-4-4.



3.13.4 Check Interlocking

Execute transaction MDGIMG: (Classic Mode in SAP MDG) -> Central Governance -> Process Modelling -> Hierarchies -> Define Scope for Changes -> Data Model I1 -> Scope for Changes -> Hierarchy Type: Product Hierarchy: PRODH -> Interlocking. This needs to be defined as "Strict" and shall not be changed as inconsistencies in the Product Hierarchy might occur.

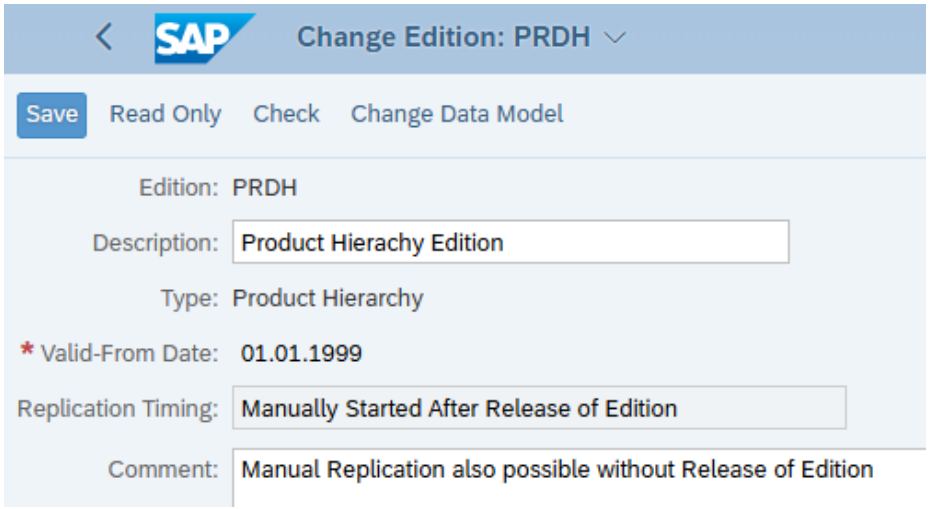


Hierarchy Type	Description	Interlocking
PRODH	Product Hierarchy	Strict

3.13.5 Define Edition

For the definition of an Edition start the Fiori Launchpad and select Analysis of Editions -> Create. Use Edition Type "Product Hierarchy (I1_PRODH)" and define the Data Replication Timing "Manually started after Release of Edition"

Example:



Note: Parallel editions are currently not supported.

3.13.6 Creation of a Product Hierarchy Name

The creation of the Product Hierarchy Name is done through a change request process. Start the Fiori Launchpad and select Change Requests -> Sales -> Product Hierarchy Name -> New -> Change Request Type: IPN01 (Create Product Hierarchy Name). Select your edition and continue.

The RDM system on default uses the Product Hierarchy Name ID "ProdHrchy". Submit and verify that the change request is finalized automatically.

3.13.7 Maintenance of number ranges

To ensure system compatibility and adequate capacity for different data sources, all number ranges must start at or above 10,000,000.

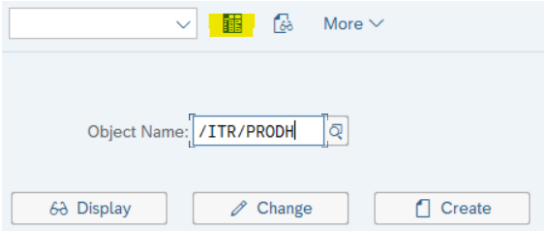
- Reserved Space: The range between 10,000,000 and 14,999,999 is specifically reserved for bulk uploads from external sources.
- Manual Entry Threshold: To prevent collisions and maintain a clear logical split, manual creation in the User Interface should always begin above this threshold (e.g., starting at 15,000,000).

For the first level of the hierarchy, the ranges are split as follows:

Data Source	Range Assignment	Logic

File Uploads	10,000,000 – 14,999,999	Provides 5 million slots for external data.
Manual Creation	15,000,000 – 19,999,999	Starts with digit '15' to signify Level 1 manual entry.

The maintenance of number ranges starts with definition of the "From Number" and the "To Number" for each level of the Product Hierarchy Nodes and the (optional) Product Hierarchy Sub Nodes. After the initial load of the Product Hierarchy also the Number Range Status needs to be maintained before new Nodes or Sub Nodes can be created. Use transaction SNRO (Object: /ITR/PRODH) and choose "Interval Editing" for the maintenance process.



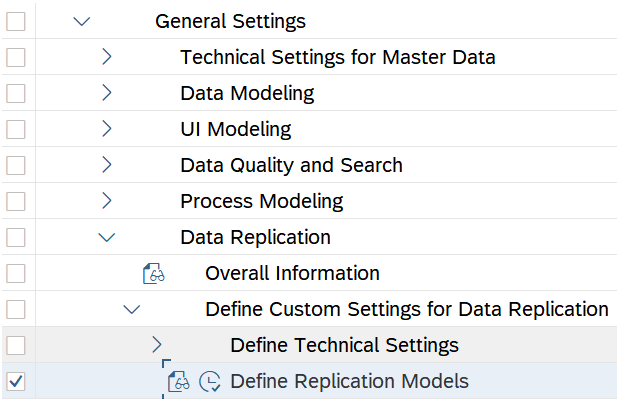
Create an interval. An example is shown below.

Number Range No.	From No.	To Number	NR Status	External
01	000000000015000000	000000000019999999	0	
02	000000000025000000	000000000029999999	0	
03	000000000035000000	000000000039999999	0	
04	000000000045000000	000000000049999999	0	
05	000000000055000000	000000000059999999	0	
06	000000000065000000	000000000069999999	0	

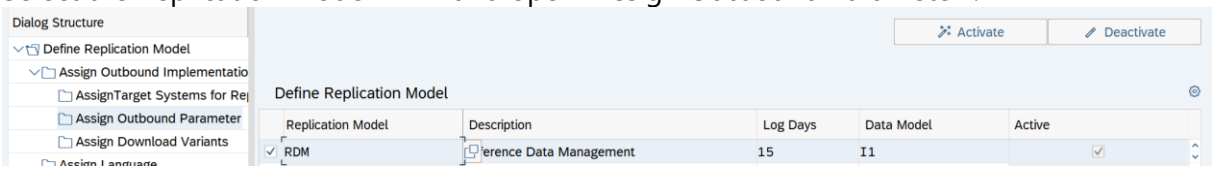
01	000000000015000000	000000000019999999
02	000000000025000000	000000000029999999
03	000000000035000000	000000000039999999
04	000000000045000000	000000000049999999
05	000000000055000000	000000000059999999
06	000000000065000000	000000000069999999

3.13.8 Define Replication Model

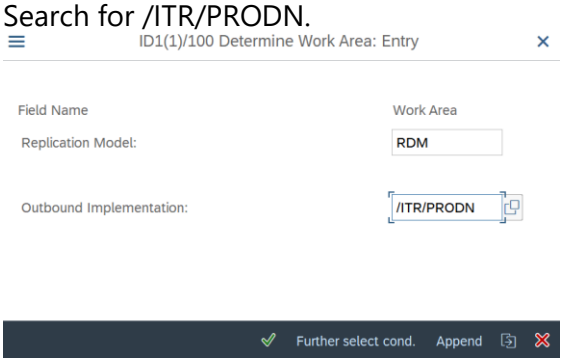
Open MDGIMG ➔Define Custom Settings for Data Replication ➔ Define Replication Models



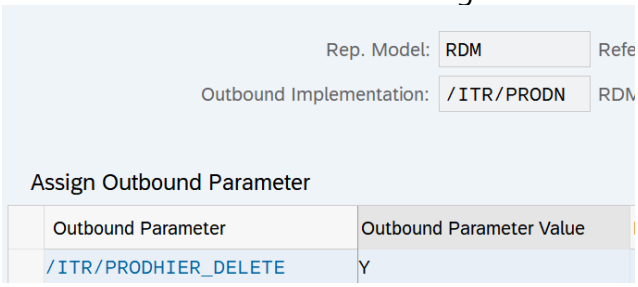
Select the Replication Model RDM and open "Assign Outbound Parameter".



Search for /ITR/PRODN.

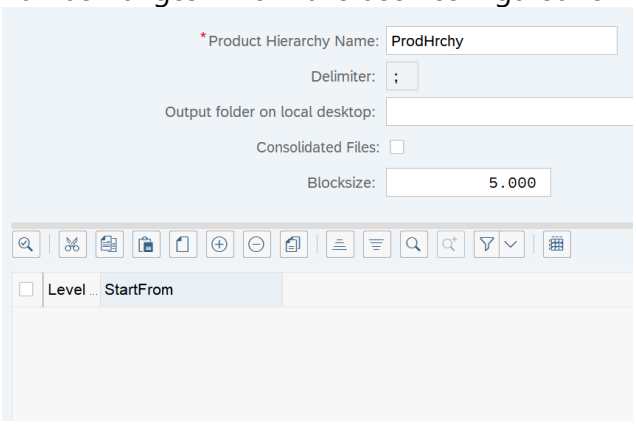


Enter "/ITR/PRODHIER_DELETE" as the Outbound Parameter and "Y" as the Outbound Parameter Value and save the changes.

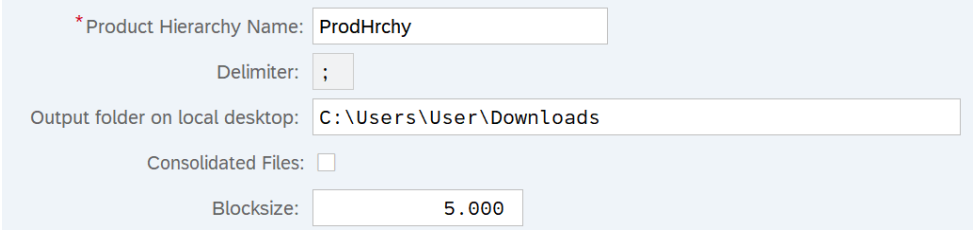


3.13.9 Data Export from business application system

For the data export from the SAP business application system execute transaction SA38 and choose report /ITR/MDG_PRODH_EXPORT to extract the product hierarchy using the defined number ranges which have been configured for MDG RDM.



Select an Output folder.



- **Consolidated Files**: Only check off Consolidated Files if you have a smaller 'Product Hierarchy' and want to output less consolidated files.

- **Block size**: Change the number of entries in a single file by editing the block size. Having fewer entries will increase the performance, but you may end up with more output files, depending on the size of the hierarchy.

Add level and Start From – use the levels you configured earlier. Then, execute the report.

Level ...	StartFrom
1	10000000
2	20000000
3	30000000
4	40000000
5	50000000
6	60000000

Files without extra letter contain the following attributes separated by ";"

Product Hierarchy Node; Description; External Number; Prod.Hier.Level; Prefix; Suffix.

Attribute	Explanation
Product Hierarchy Node	Unique internal MDG key
Description (long text)	Node Description
External Number	Unique external ERP key consisting of all external numbers (prefix) of the parent nodes and the current external number (suffix). The length of the current key is configured in Step 1.3 Configure Product Hierarchy Structure
Prod.Hier. Level	Current Node Level configured in 1.2 Configure Product Hierarchy Level
External Number Prefix	Keys of the parent nodes that have been attached to each other
External Number Suffix	Current external number

Hierarchy – H Files contain the following attributes separated by ";"
Higher Level Name; Higher Level Node; Lower Level Node

Attribute	Explanation
Higher Level Name	Product Hierarchy Name –only ProdHrchy is available as Name
Higher Level Node	Parent Node – the node above the current node
Lower-Level Node	Child Node - the node below the current node

3.13.10 Data Import (File Upload)

After the Data Export the created files must be loaded to the MDG RDM system using File Upload. This should be done in the following sequence for a product hierarchy with 3 levels:

- Level 1 to 3: attributes and texts
- Level 1: hierarchy assignments of level 1 nodes to the product hierarchy name
- Level 2: hierarchy assignments of level 2 nodes to level 1 nodes
- Level 3: hierarchy assignments of level 3 nodes to level 2 nodes

File Name	Type of Transfer
Only Level	Attribute

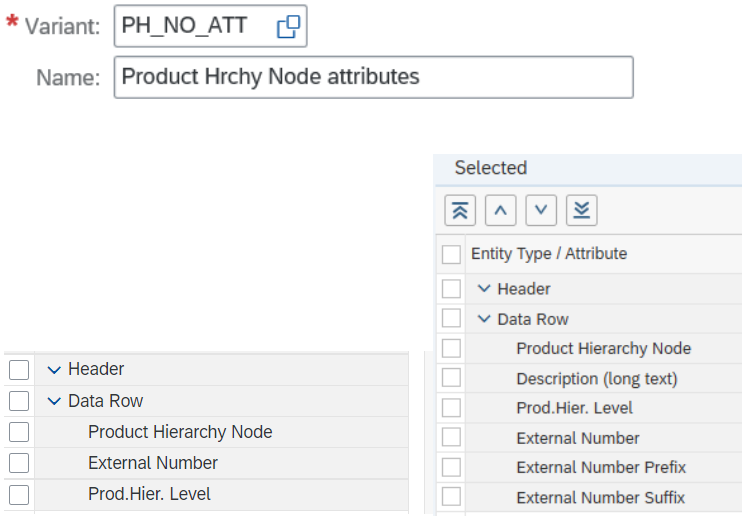
Level + H	Hierarchies
Level +T	Language-Dependent Texts

Please verify that all numbers have been generated according to the number ranges defined.

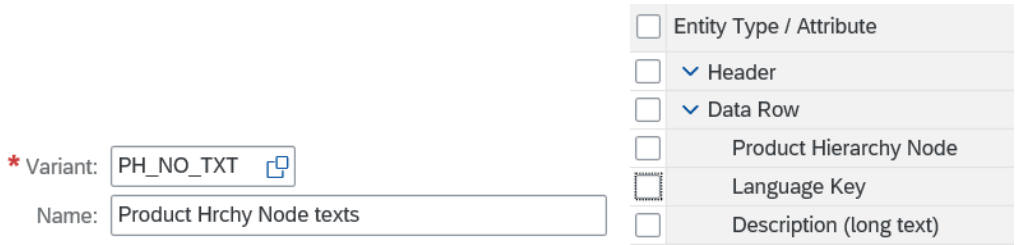
3.13.10.1 Create Variants

Defining upload variants is a prerequisite for the data import process, as they establish the technical mapping required for the system to interpret the hierarchy levels and language-dependent texts correctly:

- Level 1 attributes and texts (PH_NO_ATT and PH_NO_TXT)
  - Entity Type: Product Hierarchy Node
  - Attributes Data Row: Product Hierarchy Node, External Number, Prod.Hier. Level



  - Texts Data Row: Product Hierarchy Node, Language Key, Description (long text)



- Level 2 (and higher) attributes and texts
  - Same Entity Type and variants as for Level 1 Nodes

- Level 1 hierarchy assignments of level 1 nodes to the product hierarchy name

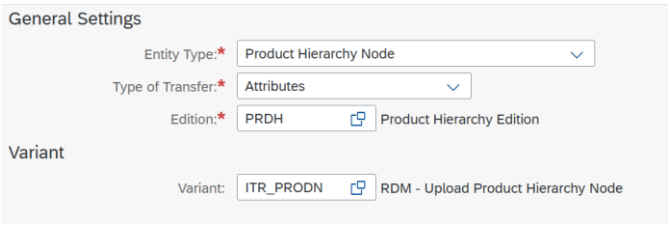
- Entity Type: Product Hierarchy Node
- Higher-level Node: Product Hierarchy Name
- Lower-level Node: Product Hierarchy Node

- Level 2 (and higher) hierarchy assignments of level 2 nodes to level 1 nodes
  - Entity Type: Product Hierarchy Node
  - Higher-level Node: Product Hierarchy Name, Product Hierarchy Node
  - Lower-level Node: Product Hierarchy Node

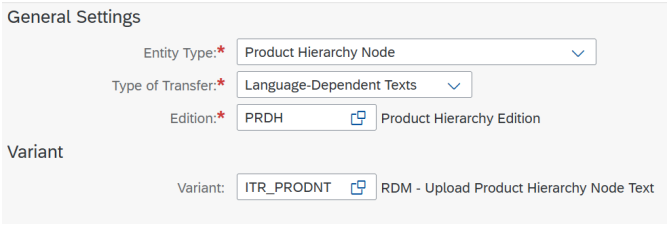
3.13.10.2 Initial Load

Use the variants above (for Nodes, Hierarchy and Texts) to load the data vie "File Upload".

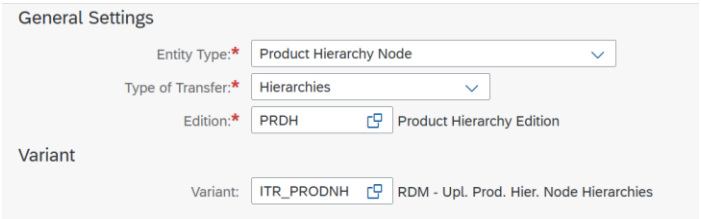
Nodes:



Texts:

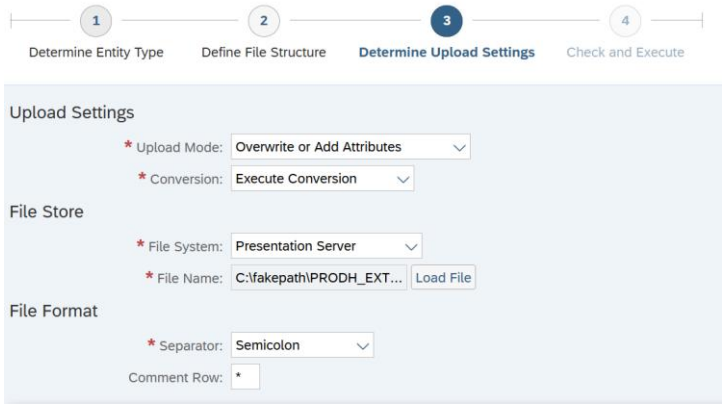


Hierarchy:

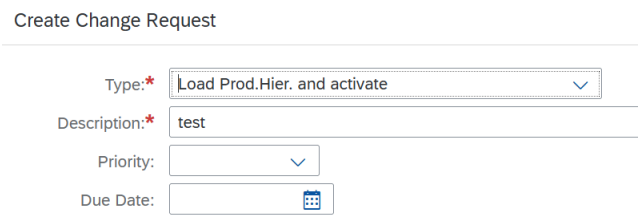


- Upload Mode: "Overwrite or Add"
- Conversion: "Execute Conversion"
- File System / File Name: <upload directory>

- Separator: “Semicolon”
- Comment Row: *



- Change Request Type: “Load Prod. Hier. And activate”
- Description: <any>



After each File Upload check the objects using the search application for the object: Change Requests -> Processing -> Sales -> Product Hierarchy Node.

After this the maintenance of the product hierarchy can start and the next created nodes (using the Single Object Maintenance) will get the next number. Nodes which are created by File Upload will get the numbers which are specified in the files. Aligned with the examples used above make sure that you use numbers from the following number ranges for the creation of new nodes (example for a three-level hierarchy):

- 1: 10000000 - 14999999
- 2: 20000000 - 24999999
- 3: 30000000 – 34999999

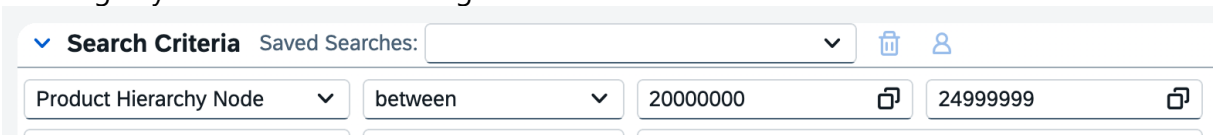
3.13.10.3 Upload new nodes

Before uploading new nodes, the last uploaded nodes must be checked.

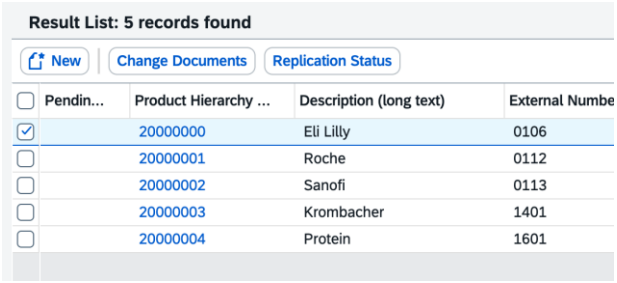
Number Range No.	From No.	To Number
01	000000000015000000	000000000019999999
02	000000000025000000	000000000029999999
03	000000000035000000	000000000039999999
04	000000000045000000	000000000049999999
05	000000000055000000	000000000059999999
06	000000000065000000	000000000069999999

Upload Nodes Level 1 are between 10.000.000 and 14999999.

Open Product Hierarchy Nodes in Fiori. Check for each level the latest created note outside the range by search criteria for Range.



The next available node for Range 2 is 20000005.



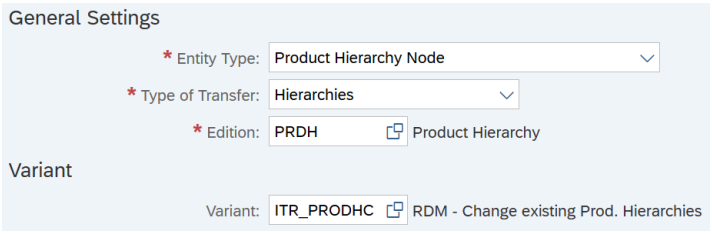
Upload the Nodes, Texts and Hierarchy as described in the Initial Load Section.

Note:
The upload file may only contain the new parts of the hierarchy. You can't change the nodes that are already part of the hierarchy with this upload option. To change an existing Hierarchy, see next chapter.

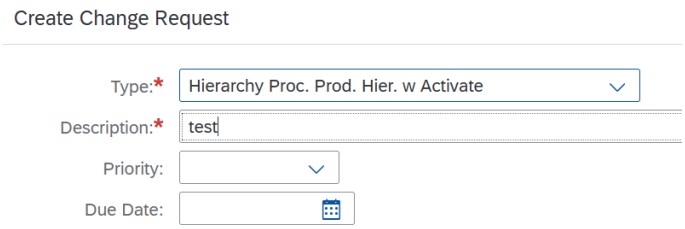
3.13.10.4 Change existing Hierarchy

The only way to change an existing hierarchy by upload is the Upload Mode - Delete All. This deletes all previous connections. Include all parts of the existing hierarchy you want to keep in the file.

Select Type of transfer Hierarchies and select Variant – ITR_PRODHC.



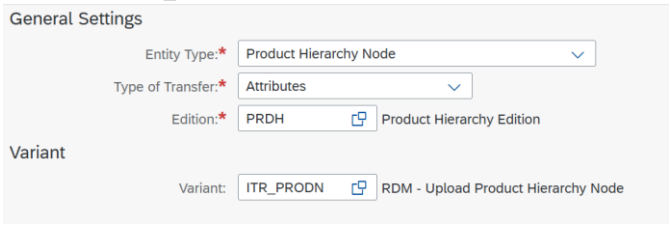
Load the Hierarchy File. Select Hierarchy Change Request and execute upload.



3.13.11 Data Download

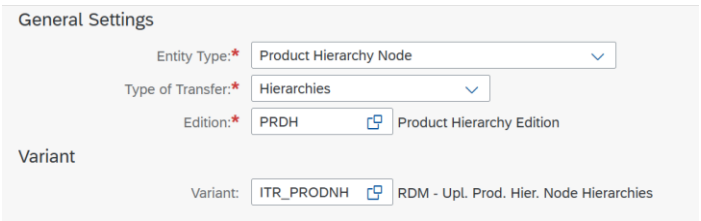
Open the File Download and select.

- Entity Type - Product Hierarchy Node,
- Type of Transfer - Attributes, Edition
- Variant - ITR_PRODN



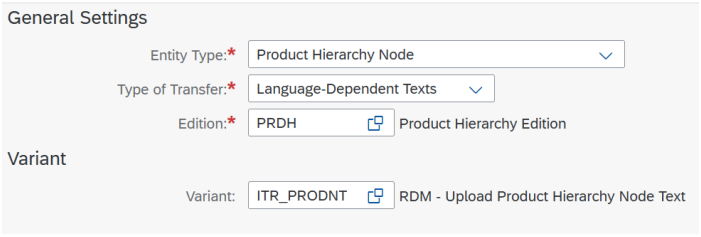
Press Next until your able to Execute, then save.

Go back and change Type of transfer to Hierarchies and select Variant – ITR_PRODNH.



Press Next until your able to Execute, then save.

Go back and select Type of Transfer - Language-Dependent Texts and Variant - ITR_PRODNT.



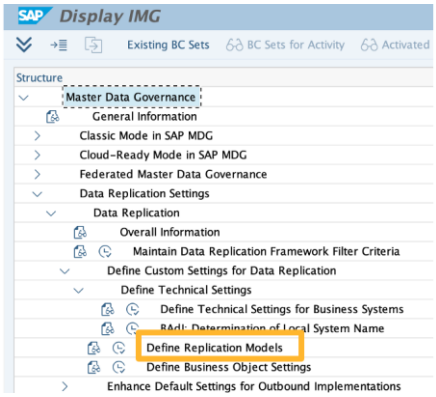
Press Next until your able to Execute, then save.

3.13.12 Enabling 'Add to Change Request' Functionality

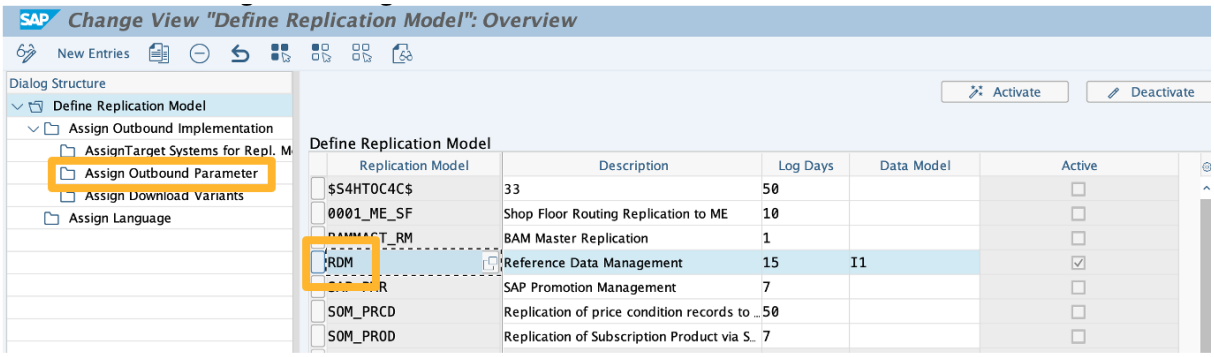
To enable the "Add to Change Request" functionality in the user interface, the corresponding SAP Note 3668887 must be implemented in the system. This note ensures that the required backend functionality is available and prevents dumps during processing. This functionality ensures that the description of existing hierarchy nodes can be changed without the need to navigate to the change request itself. Mark the node, add it to the change request and change the description. Without this functionality you need to navigate to the change request and add the hierarchy node manually before you can change the description.

3.13.13 Delete Hierarchy

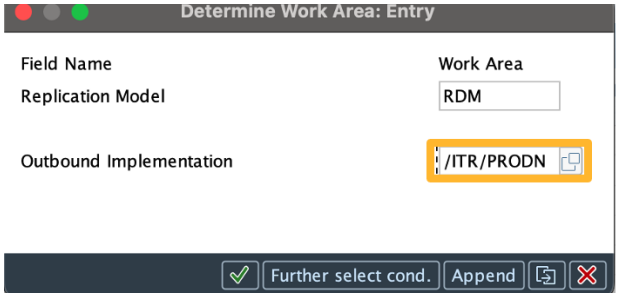
The replication of the hierarchy is always done by sending the complete hierarchy. To ensure a consistent replication in the receiving system the product hierarchy needs to be deleted first as part of the replication process. This needs to be configured: Open Tx *MDGIMG* and select „Define Replication Model ".



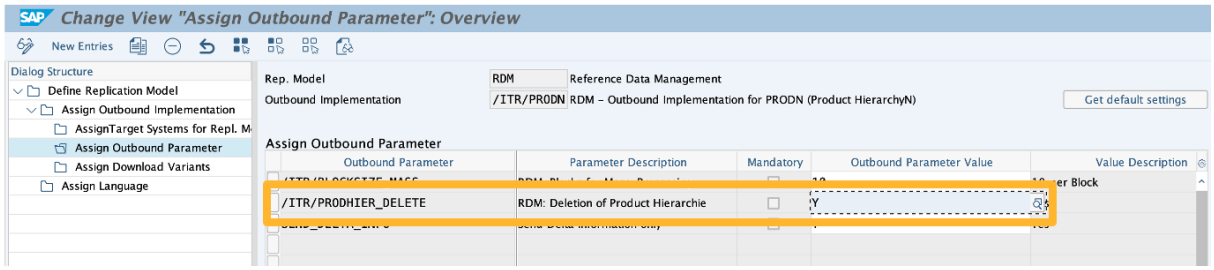
Select „RDM" and select „Assign Outbound Parameter ".



Enter „*/ITR/PRODN*“ as Outbound Implementation.



Set „Y“ to configure the deletion step.



3.14 Configure Classification

This section describes the configuration required for Classification, covering internal numbering, the prefix for temporary key assignment, and the upload/download of Classes and Characteristics.

3.14.1 Internal Numbering

Classes and Characteristics are implemented with an internal numbering. This means that number ranges need to be maintained using transaction SNRO (or SNUM). Examples:

Classes, Number Range Object /ITR/CLF:

Edit Intervals: Classification, Object /ITR/CLF

Number Range No.	From No.	To Number	NR Status	External
01	0000000000000000001	9999999999999999999	80	

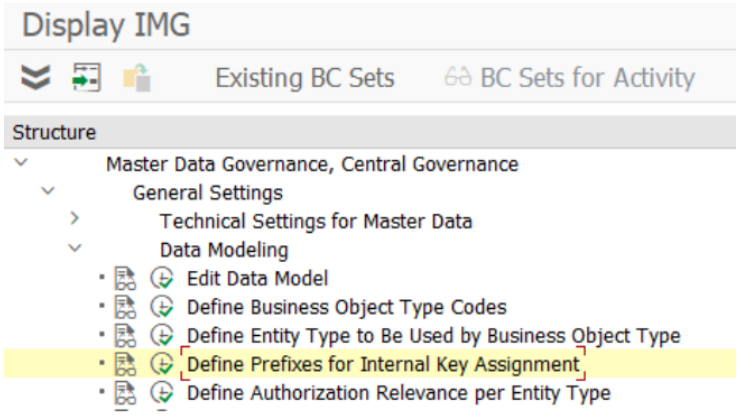
Characteristics, Number Range Object /ITR/CHAR:

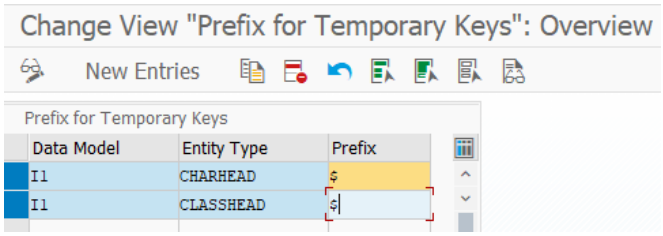
Edit Intervals: Characteristics, Object /ITR/CHAR

Number Range No.	From No.	To Number	NR Status	External
01	0000000000000000001	9999999999999999999	80	

Prefix for Internal Key Assignment:

Additionally, a prefix for the temporary keys needs to be defined via IMG: Master Data Governance, Central Governance → General Settings → Data Modeling → Define Prefixes for Internal Key Assignment:





3.14.2 Upload and Download of Classes and Characteristics

The file upload process for classes and characteristics is different from other processes because these object types use a temporary key (starting with a "$"). The definition of the temporary key as described above is mandatory before any process can be started.

Only after activation the final keys (technical names: CLASSHEAD, CHARHEAD) are assigned. This key will be used for all processes which change classes or characteristics. For classes, this key is identical to the field CLINT in table KLAH. For characteristics, this key is identical to the field ATINN in the table AUSP. Please note that these keys are not visible on the single object maintenance UI.

For processes which create data, the creation of the temporary key must be initiated by uploading the respective header data. This temporary key will then be used to assign descriptions, additional texts, and other related data (e.g. values, restrictions and keyword data). An example can be found in section 3.14.2.4 “Create Classes and Characteristics using file upload”.

3.14.2.1 Upload Variants for Classes and Characteristics

The next sections assume that you are familiar with the documentation provided by SAP which can be found for example here: https://help.sap.com/mdg -> Product Assistance -> Master Data Governance (Classic Mode) -> Working with Master Data Governance -> Working with SAP MDG, Central Governance -> General Functions -> Master Data Processing -> File Upload / File Download (Link for SAP S/4 2025 FPS01 (Feb 2026): Master Data Processing | SAP Help Portal).

Note: File Upload is not used for the initial load of classes and characteristics. Please use the SAP ALE standard scenario using IDoc-Types CLSMAS (for classes) and CHRMAS (for characteristics.

Prerequisite for using Upload Variants: the latest BC-Set /ITR/MDG_UPLOAD_VAR_* must be activated before these variants are available in your system. The delivered variants are templates only. Copy them to your own namespace so that they can be adjusted and fit into a consistent authorization concept.

Unlike most other entity types in our data model, Classes and Characteristics do not use the language-dependent texts transfer type. This is a deliberate modeling difference from other Itego RDM object types and is the reason why the load process for texts/descriptions works differently.

Entity Type: CHARHEAD (Characteristic):

Header for Characteristics (CHARHEAD) is the main entity type, Description for Characteristics (CHARDESC), Restrictions for Char (CHARREST), Char Values (CHARVAL) and Descriptions for Characteristics Values (CHVALDESC) are modeled as dependent entity types. For each of these entity types a variant for attributes transfer type is offered:

Entity Type	Attribute Variant
Header for Characteristics	ITR_CHARH
Description for Characteristics	ITR_CHARD
Restrictions for Char	ITR_CHARR
Char Values	ITR_CHARV
Descriptions for Characteristics Values	ITR_CHARVD

The language dependent long texts for characteristics are part of variant ITR_CHARD, the language dependent long texts for characteristic values are part of variant ITR_CHARVD. **Note:** Variant ITR_CHARHT is obsolete.

Entity Type: CLASSHEAD (Class):

Header for Class (CLASSHEAD) is the main entity type, Description for Class (CLASSDESC), Characteristics of a Class (CLASSCHAR), Class Text (CLASSTEXT) and Keywords Data for Class (CLKEYWORD) are modeled as dependent entity types. For each of these entity types a variant for attributes transfer type is offered:

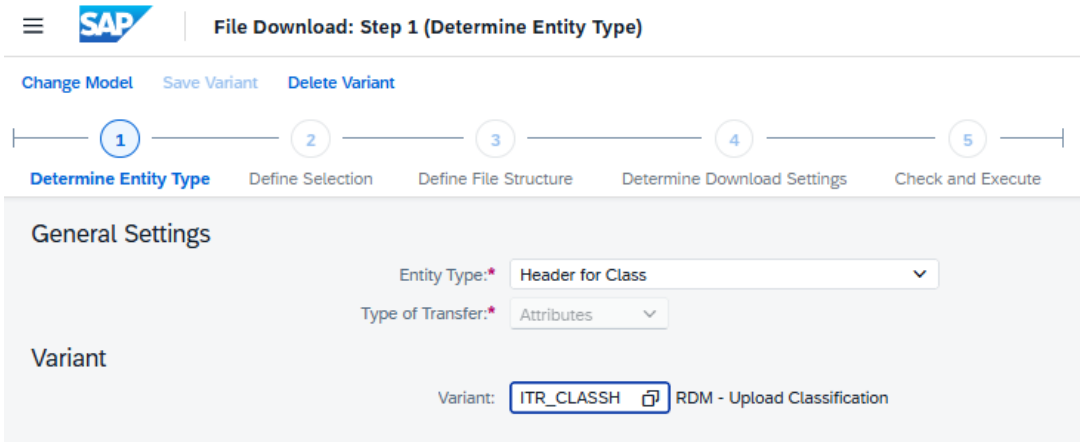
Entity Type	Attribute Variant
Header for Class	ITR_CLASSH
Descriptions for Class	ITR_CLASSD
Characteristics of a Class	ITR_CLASSC
Class Text	ITR_CLASST
Keywords Data for Class	ITR_CLASSK

Note: ITR_CHARHT and ITR_CLASHT/ITR_CLASSHT are obsolete language-dependent-text variants and should not be used. Use ITR_CHARD (for Characteristics) and ITR_CLASSD (for Classes) instead.

To down- or upload a complete class or characteristic, all related entities must be down- and uploaded in the order shown in the tables above (header first, after this related data).

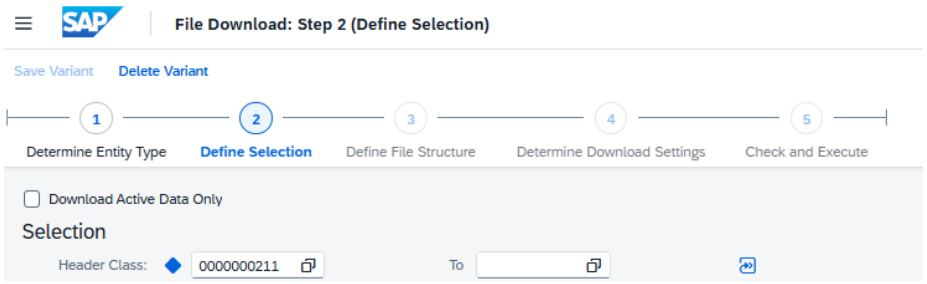
3.14.2.2 Download Classes and Characteristics

To download classes and their dependent entity types start with selecting entity type 'Header for Class' and variant ITR_CLASSH.

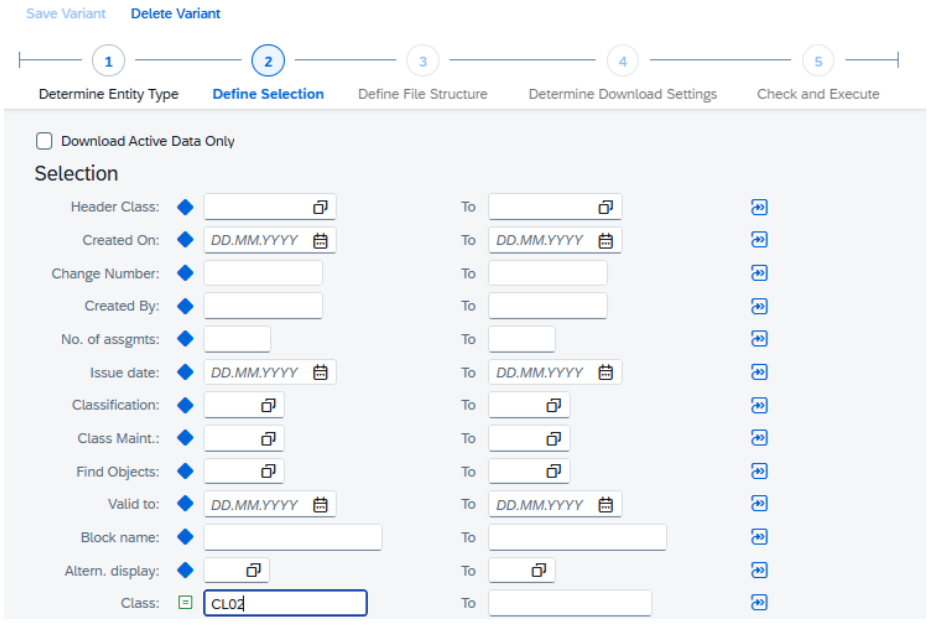


Select your class(es). Simple example here, just one class: key is “0000000211”, class is “CL02”.

Example: Selection by internal key (Header Class):



Example: Selection by Class:

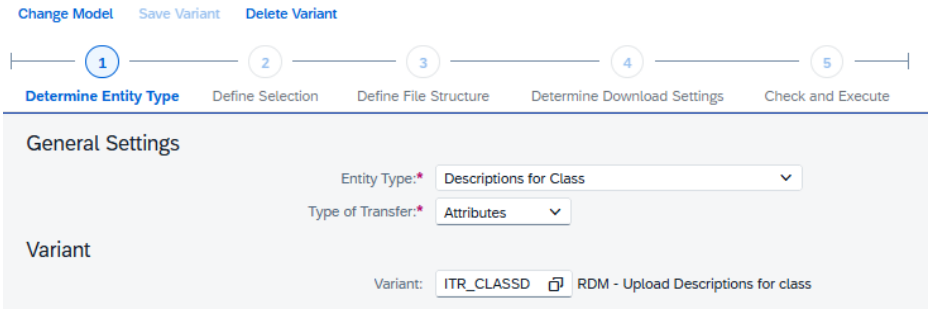


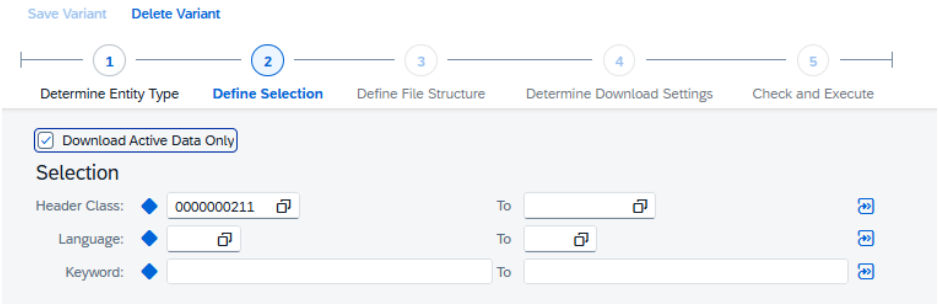
The file created shows the key in the first column (CLASSHEAD), the class in the second column and the description in the logon language of the user in the third column:

```
* Download
* Data Model: I1
* Entity Type: CLASSHEAD
* Variant: ITR_CLASSH
* User: IT_ALL_01
* Date: 20260818 / Time: 135409
* Selection:
* CLASSHEAD EQ | 0000000211 |
*CLASSHEAD;CLASS1;TXTMI;KLART;VWS
0000000211;CL02;Description CL02;
```

Use other variants for classes to load more data related to class "CL02". To down- or upload a complete class it is necessary to down- and upload all entities in the order shown in the list in the previous section 3.14.2.1 "Upload Variants for Classes and Characteristics".

Note: the language dependent variant ITR_CLASSHT is obsolete and should not be used. Use entity type 'Descriptions for Class' and variant ITR_CLASSD instead:






```
*CLASSHEAD;KSCHL1;SPRAS_CL
0000000211;Description CL02 DE;DE;
0000000211;Description CL02;EN;
```

To download characteristics and their dependent entity types proceed in the same way. Start with selecting entity type 'Header for Characteristics' and variant ITR_CHARH.

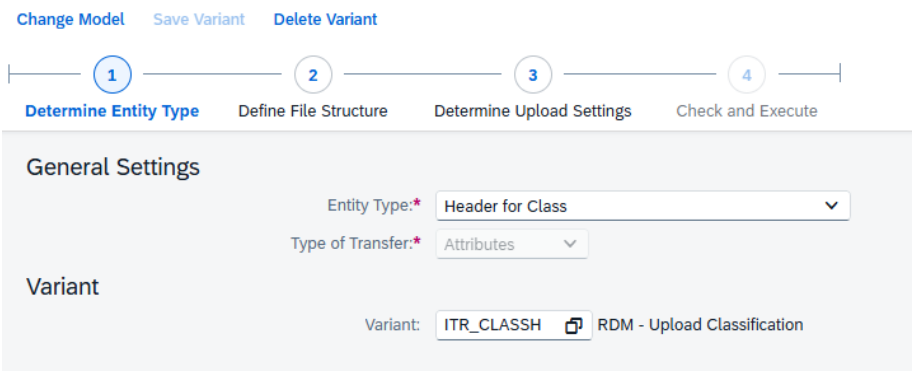
To down- or upload a complete characteristic it is necessary to down- and upload all entity types in the order shown in the list in the previous section 3.14.2.1 "Upload Variants for Classes and Characteristics".

3.14.2.3 Change Classes or Characteristics using file upload

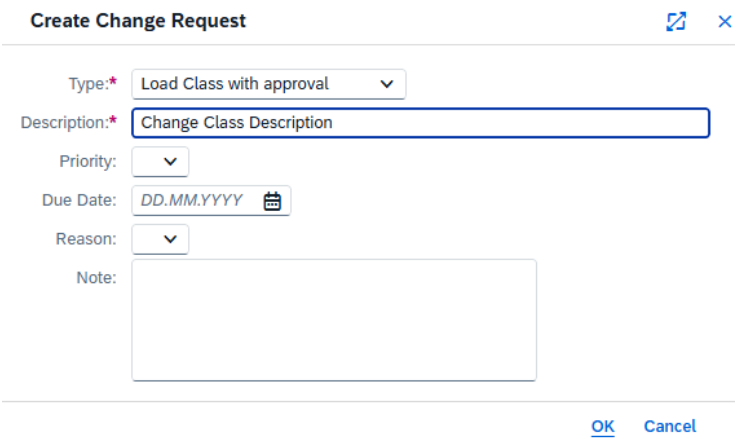
Changing data using the file upload works by using the created files with column CLASSHEAD and CHARHEAD filled, indicating that the provided class should be changed. Simple example: Change Class Description in the logon language of the user:

```
*CLASSHEAD;CLASS1;TXTMI;KLART;VWSTL
0000000211;CL02;Description CL02v2;
```

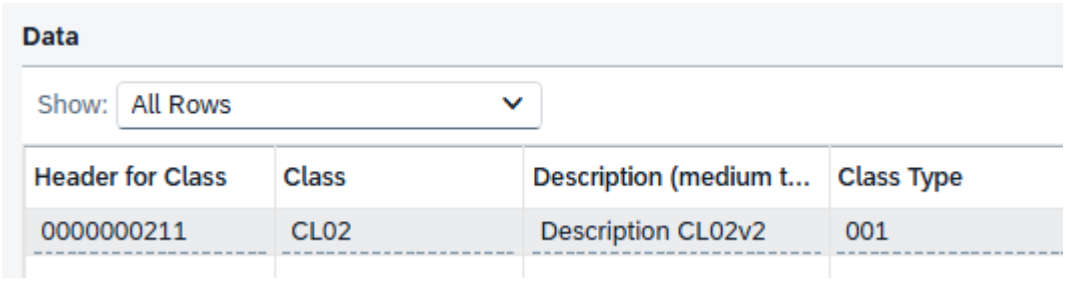
Use variant ITR_CLASSH because only one language should be changed:



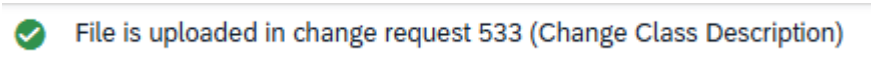
A change request will be created:



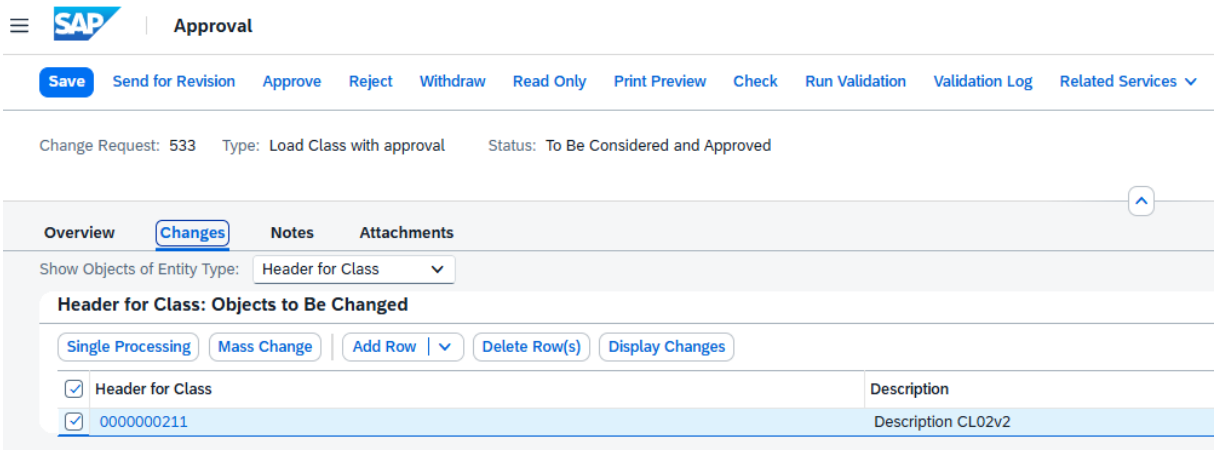
The data provided in the file will be shown for validation before the Upload is executed:



Created change request:



Validate and approve and the data will be changed:



3.14.2.4 Create Classes and Characteristics using file upload

Creating new classes and characteristics works with leaving the CLASSHEAD and CHARHEAD empty in the first step (to indicate that you would like to create new data) and using the obtained temporary key in the second step to add the related data to your change request.

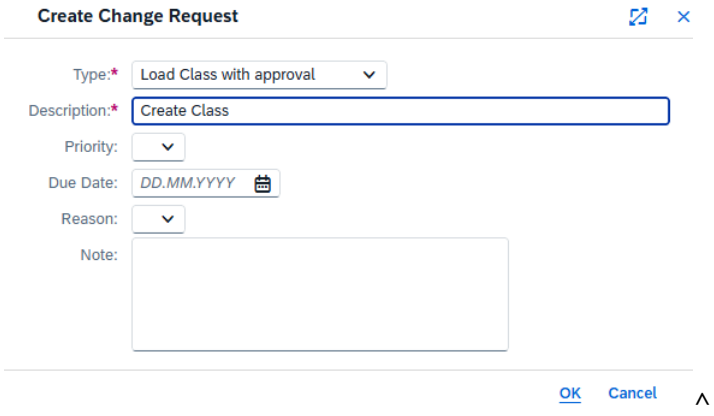
Note**:** Although the example below shows the creation of a class, the same pattern applies to the creation of a characteristic. If you are creating a new class as well as new characteristics which should be assigned to the new class, please make sure to create new characteristics first, otherwise you will not be able to assign the new characteristics to the new class.

First step: Use "Header for Class" or "Header for Characteristics". Example here: "Header for Class":

```
*CLASSHEAD;CLASS1;TXTMI;KLART;
;CL03;Description CL03;001;;;;
```

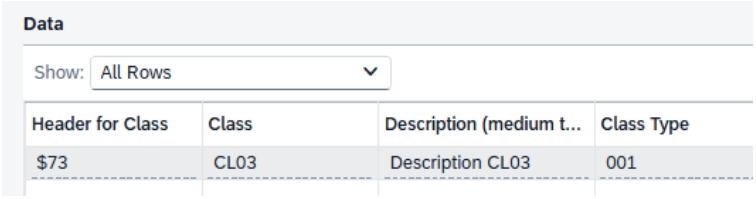
Note: Column CLASSHEAD is empty to indicate that this is the creation of a class

A change request will be created:



Make sure that you use change request types which at least do have two steps otherwise you will not be able to add the related data to be loaded in step 2.

Data shown during file upload for validation and to obtain the temporary key(s):

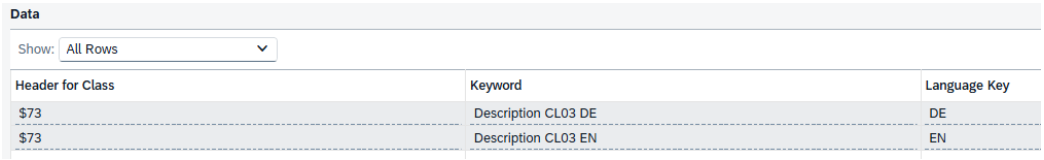


Second step: Upload more data related to the new class by using the dependent entity types and corresponding variants. The files need to contain the temporary key starting with $. Example: Descriptions for Class.

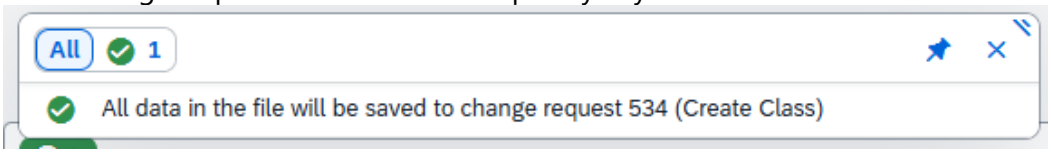
```
*CLASSHEAD;KSCHL1;SPRAS_CL
$73;Description CL03 DE;DE;
$73;Description CL03 EN;EN;
```

Be aware of the usage of temporary key "$73" from the first step.

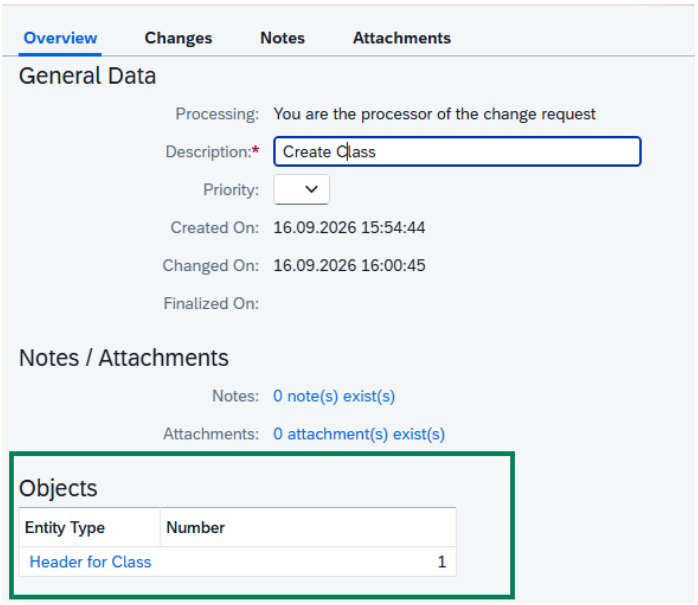
Data shown for validation:



Used change request based on the temporary key:

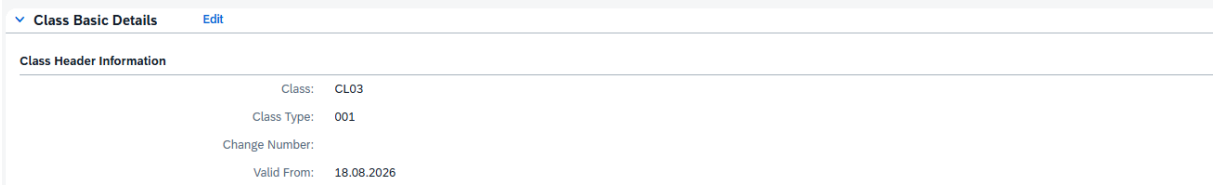


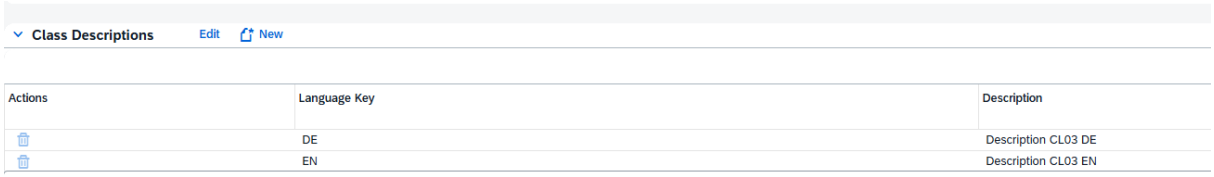
Note: When you access the change request you will see only entity type 'Header for Class'



To see the text which has been uploaded, you need to navigate to the single object maintenance UI.

Language dependent texts uploaded for the new class CL03:





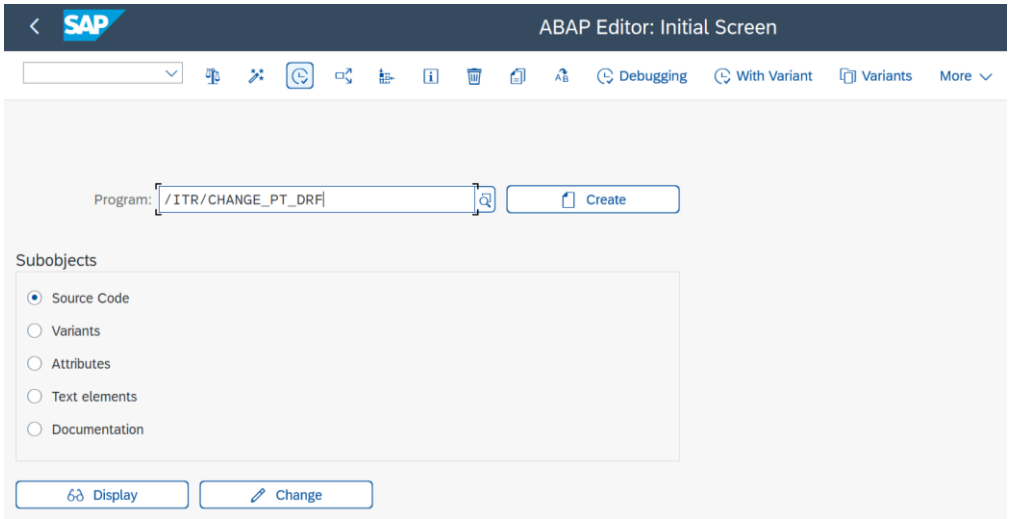
To finalize the process, approve the change request.

3.15 Configure Payment Term

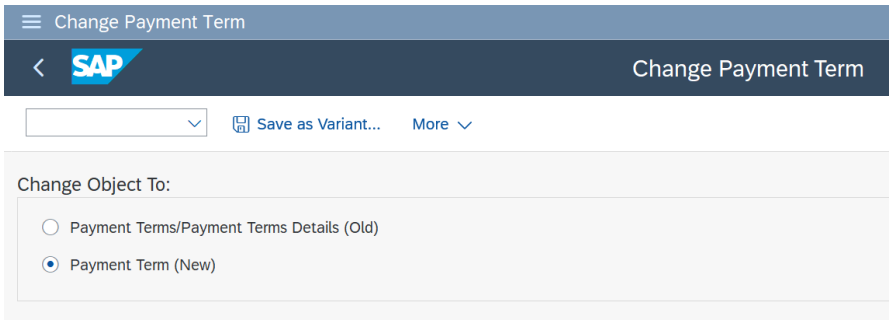
Since Release RDM 2603, the Payment Term object has been enhanced to combine Payment Terms and Payment Term Details (the Day Limit) into a single object type with a unified user interface. This reduces the maintenance effort associated with managing two separate objects and user interfaces.

3.15.1 Activation Report

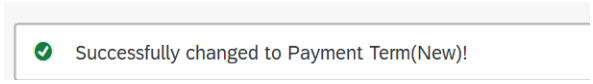
To migrate to the new version of Payment Term execute the program */ITR/CHANGE_PT_DRF* via Tx *SE38*.



Choose option '*Payment Term (New)*' and press "Execute".



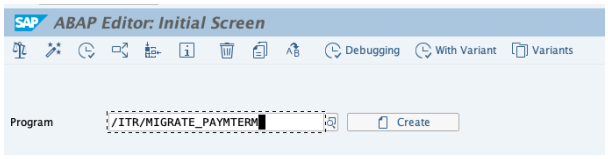
After execution it confirms the switch to the new Payment Term.



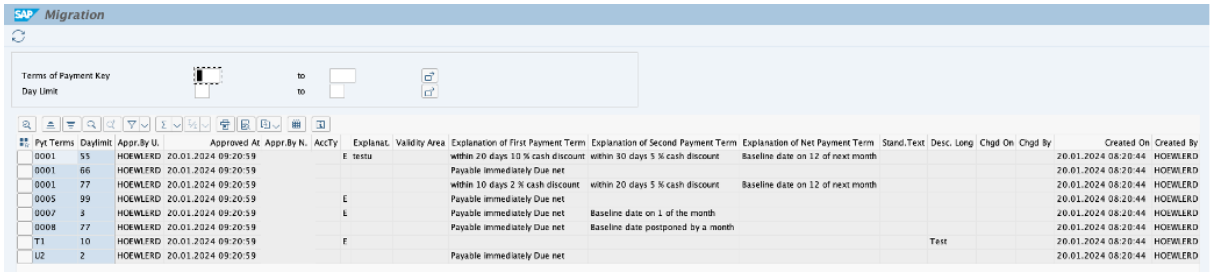
3.15.2 Data Migration Report

The migration report transfers existing Payment Term data, which was maintained in two separate user interfaces (Day Limit and Payment Term), into the new consolidated single-UI version.

Open transaction SE38 (ABAP Editor), enter /ITR/MIGRATE_PAYMTERM and execute it.

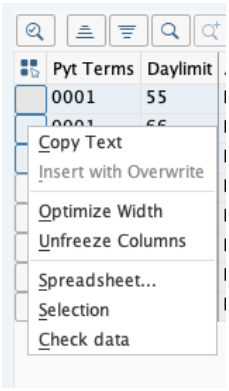


The screen lists all available Payment Terms together with their existing day limits, which will be merged into a single UI during the migration.



Pyt Terms	Daylimit	Appr.By U.	Approved At	Appr.By N.	AccTy	Explanat.	Validity Area	Explanation of First Payment Term	Explanation of Second Payment Term	Explanation of Net Payment Term	Stand.Text	Desc. Long	Chgd On	Chgd By	Created On	Created By
0001	55	HOEWLERD	20.01.2024 09:20:59		E	testu		within 20 days 10 % cash discount	within 30 days 5 % cash discount	Baseline date on 12 of next month					20.01.2024 08:20:44	HOEWLERD
0001	66	HOEWLERD	20.01.2024 09:20:59					Payable immediately Due net							20.01.2024 08:20:44	HOEWLERD
0001	77	HOEWLERD	20.01.2024 09:20:59					within 10 days 2 % cash discount	within 20 days 5 % cash discount	Baseline date on 12 of next month					20.01.2024 08:20:44	HOEWLERD
0005	99	HOEWLERD	20.01.2024 09:20:59		E			Payable immediately Due net							20.01.2024 08:20:44	HOEWLERD
0007	3	HOEWLERD	20.01.2024 09:20:59		E			Payable immediately Due net	Baseline date on 1 of the month						20.01.2024 08:20:44	HOEWLERD
0008	77	HOEWLERD	20.01.2024 09:20:59					Payable immediately Due net	Baseline date postponed by a month						20.01.2024 08:20:44	HOEWLERD
T1	10	HOEWLERD	20.01.2024 09:20:59		E							Test			20.01.2024 08:20:44	HOEWLERD
U2	2	HOEWLERD	20.01.2024 09:20:59					Payable immediately Due net							20.01.2024 08:20:44	HOEWLERD

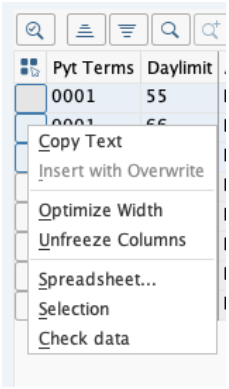
Select one or more Payment Terms and choose Right-click → Check data to check the selected entries for inconsistencies. All inconsistencies must be resolved before the migration can be performed.





Once the issues have been resolved, run the check again. No errors should be reported.

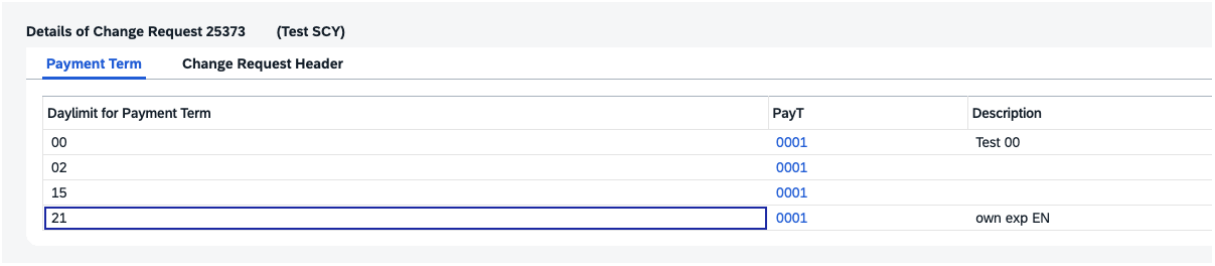
To perform the migration, select the Payment Terms and choose Right-click → Selection.



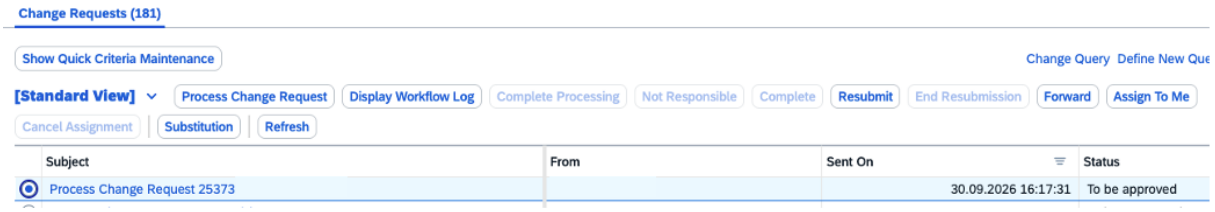
A new change request is created automatically and its number is displayed.

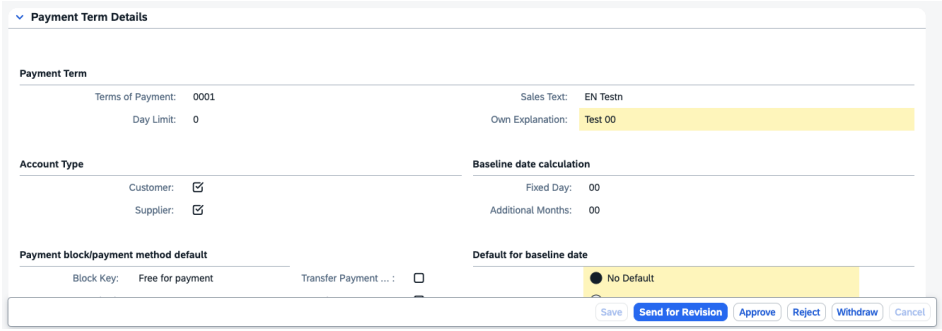


The “My Change Requests” view in the Fiori launchpad provides an overview of all Payment Terms included in the change request.

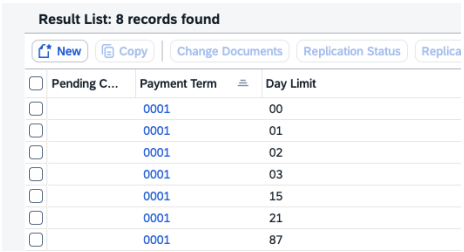


Open the change request from your inbox, then either approve it or make any necessary changes.





In the search overview of the new Payment Term all fields should now be visible.



3.16 Configure Public Holiday

If the Public Holiday is within the scope of maintained reference data object types of additional steps need to be performed on the MDG RDM system for the correct data import and number range settings.

Prerequisite:

- Data Export from SAP S/4 has been executed.

Step 1: Set Number Range

- Transaction: SM30
  - Table/View: /ITR/NRO_FTGID
  - Maintain
  - Entity Type: FTGID
  - Public hol. ID: 000

Step 2: Data Import

- Execute Data Import or File Upload
- Execute Search (with Maximum Number of Hits = 5000) and identify the last number

Step 3: Set Number Range Status

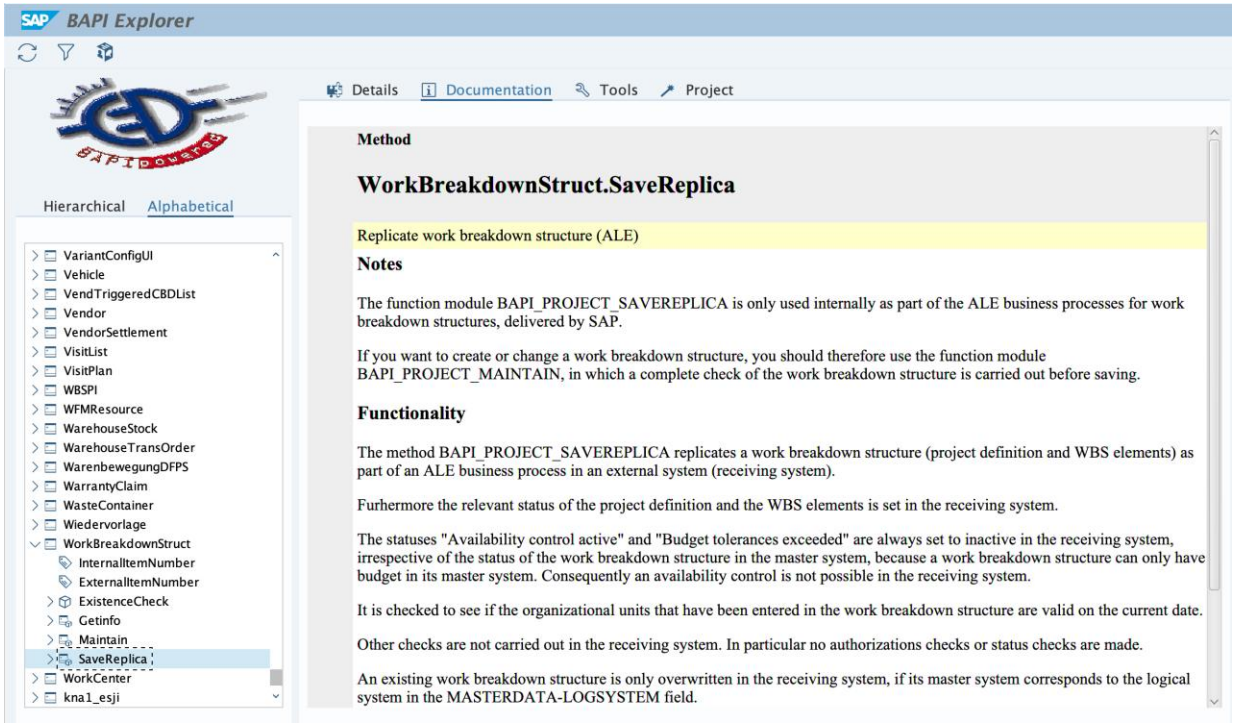
- Transaction SM30
  - Table/View: /ITR/NRO_FTGID
  - Maintain
  - Entity Type: FTGID
  - Set Public hol. ID to the last number identified from step 2.

3.17 Configure Work Breakdown Structure

The configuration of the Work Breakdown Structure (WBS) ensures that project hierarchies and elements are replicated accurately across the SAP landscape. This section describes the setup of the Application Link Enabling (ALE) framework using BAPI-based integration. Additionally, it covers the export and import procedures required for initial data loads or manual synchronization.

3.17.1 Replication of WBS Elements

To configure the ALE-based Replication of the Work Breakdown Structures, you need to establish a connection between the source and target SAP instances using the standard ALE framework. Unlike traditional IDoc-based transfers, this specific scenario utilizes a BAPI-based integration to ensure data integrity across systems. See Tx BAPI:

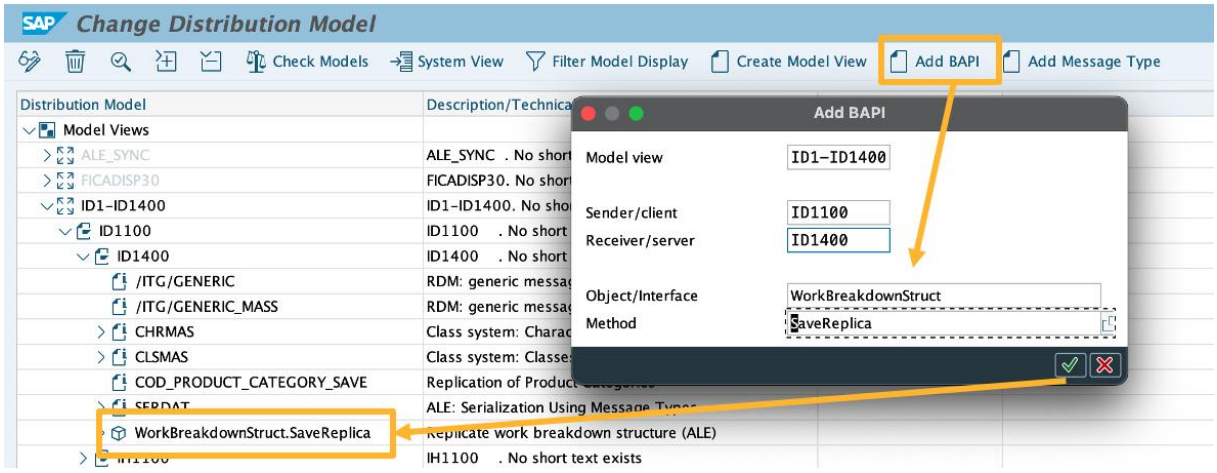


First, maintain the distribution model via Tx BD64. Click Edit and select “Add BAPI”.

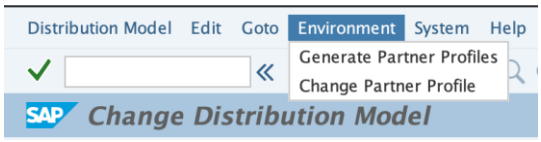
Enter the Model view, your sender and receiver system.

- For Object/Interface enter: *WorkBreakdownStruct*
- For Method enter: *SaveReplica*

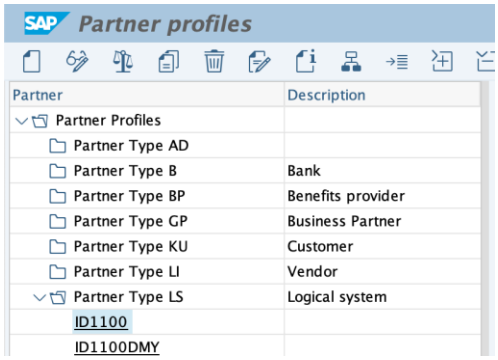
Confirm and save.



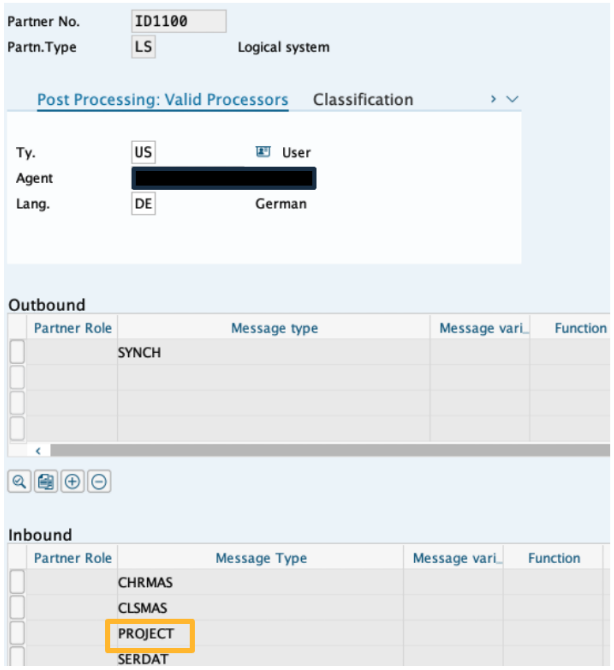
Save the model, generate the Partner Profiles (Environment -> Generate Partner Profiles), and distribute the model to the target system.



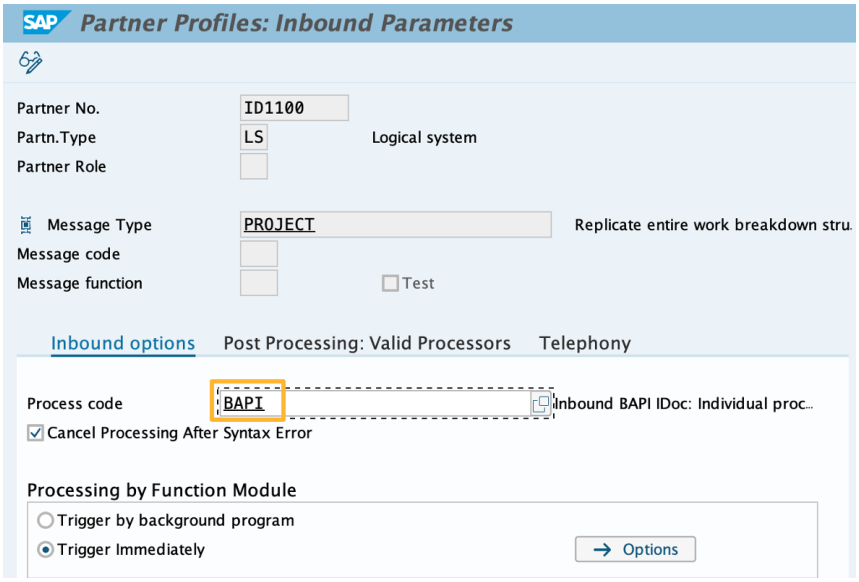
To ensure the target system can process the incoming data, you need to verify the settings in the target system. Open Tx WE20 and locate the Partner Type LS (Logical System) for your sender system.



Find the Inbound Parameter for Message Type *PROJECT* and select it.



Ensure the Process Code is set to BAPI.

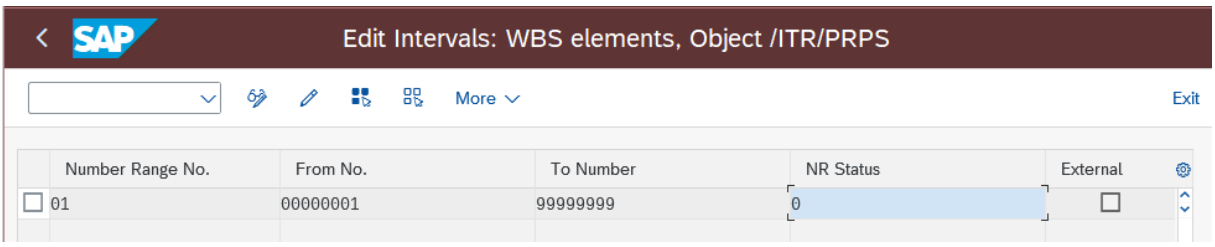


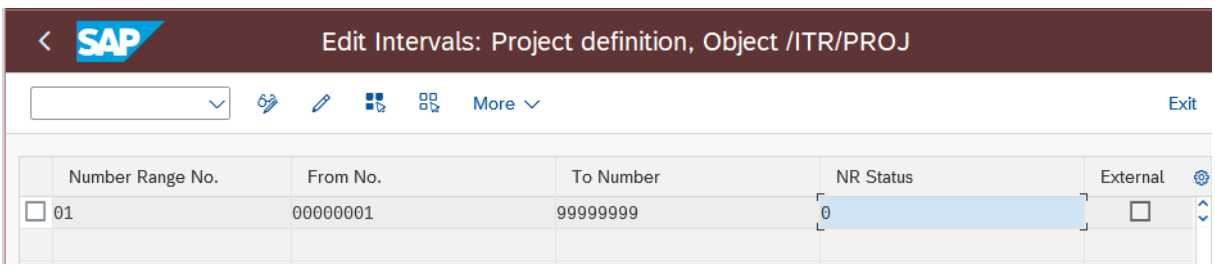
3.17.2 Export, Import and Upload WBS

The Work Breakdown Structure (WBS) provides a hierarchical breakdown of a project into manageable and clearly defined work elements. It consists of the Project Definition, its WBS-Elements and the hierarchy. This also means that the export will consist of three files (one for the Project Definition, one for the WBS Elements and one for the structure of the WBS Element Hierarchy).

3.17.2.1 Maintenance of number ranges

Use transaction SNRO (objects: /ITR/PRPS and /ITR/PROJ) and choose "Interval Editing" for the maintenance process.





3.17.2.2 Export

To perform a complete export of the Work Breakdown Structure, the MDMGX based extraction and an additional report must be executed. This dual approach is mandatory because the standard extraction and the hierarchy structure use different technologies and formats:

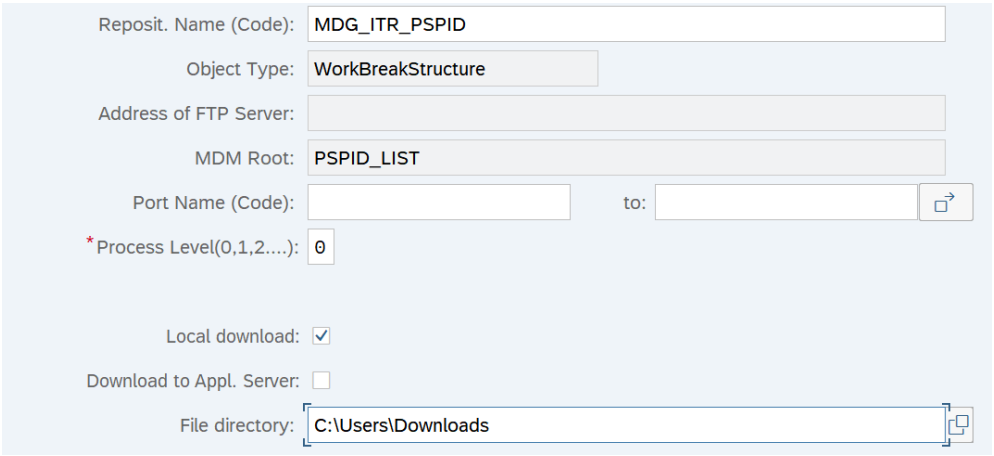
- MDMGX Extraction: Downloads master data as XML files. These files must be transferred to the application server (e.g., via Tx CG3Z) and then imported using the Data Transfer Tool (DTIMPORT).

- Report-Based Export: Generates TXT files specifically to capture the WBS hierarchy structure. These files are designed for direct upload via the File Upload application.

To export the Work Breakdown Structure, open Tx MDMGX, select "Start Extraction" and enter/select:

- Reposit. Name (Code): *MDG_ITR_PSPID*
- Local download

Then choose a File directory and execute.

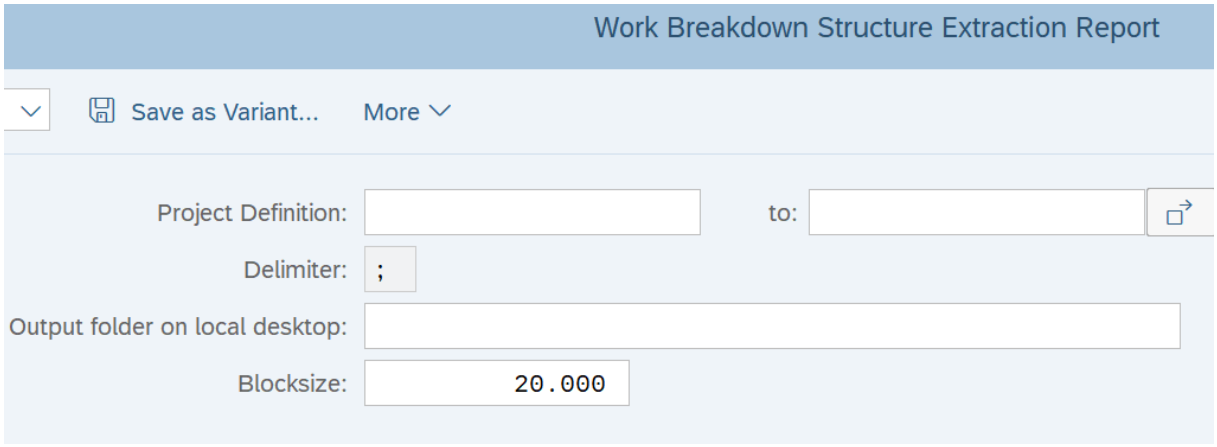


The WBS hierarchy is not included in the exported files. Use the following report */ITR/MDG_WBS_EXPORT* (via Tx SA38*)* to extract it.

Report attributes

- **Project Definition**: Select a specific Project Definition for the export
- **Output Folder:** Select an Output folder for the files.

- **Block size**: Change the number of entries in a single file by editing the block size. Having fewer entries will increase the performance, but you may end up with more output files, depending on the size of the hierarchy.



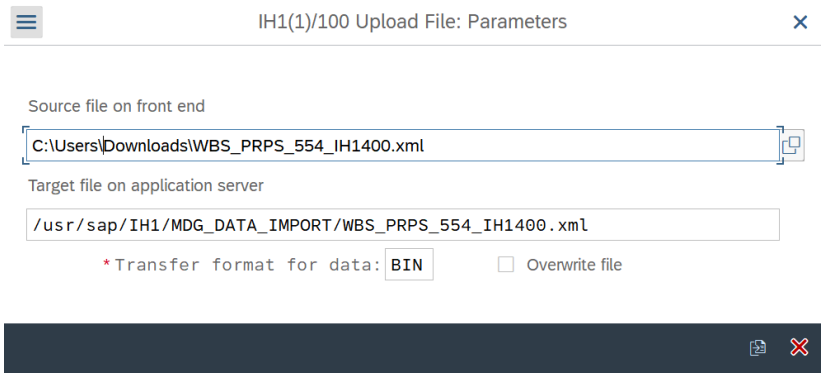
The report generates three files:

1. *PROJ_EXTRACT* (Project Definitions)
2. *WBS_EXTRACT* (WBS Elements)
3. *WBS_HIERARCHY_EXTRACT* (Hierarchy)

Only the WBS Hierarchy must be uploaded using the File Upload the other files can either be uploaded via File Upload or Data Import.

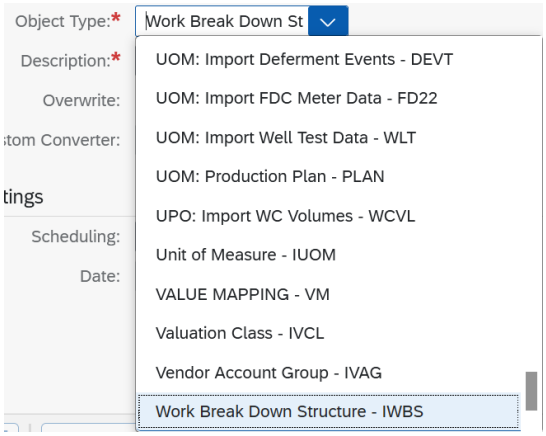
3.17.2.3 Import

If you want to use Data Import for the Project Definitions and/or the WBS Elements, transfer the files to the application server, open Tx CG3Z and select the source file. Enter the target file on the application server.

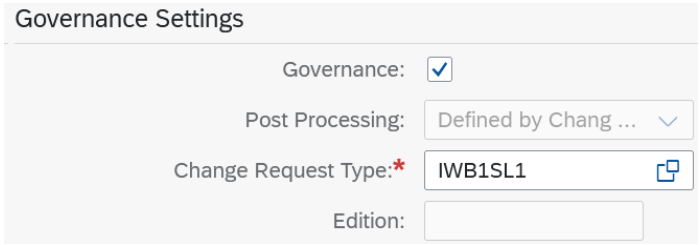


Repeat this for all xml files.

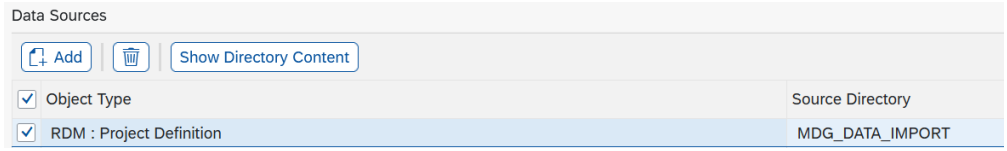
To import the files, open „Import Master Data“ in the Fiori Launchpad. Choose Object Type „Work Breakdown Structure“.



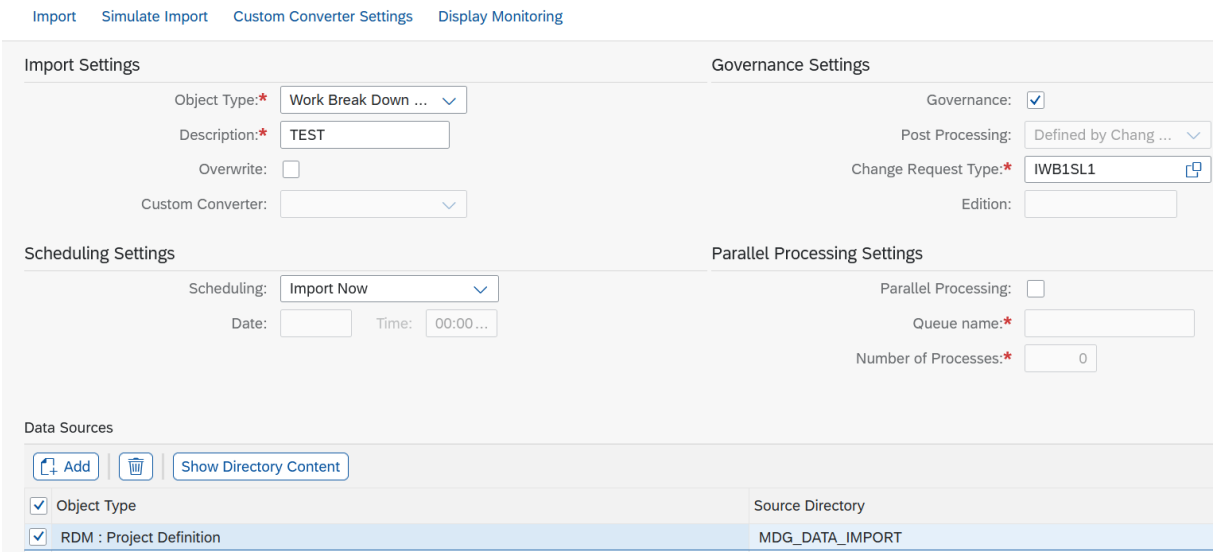
Select Governance and choose a Change-Request Type.



Add the Data Source and Source Directory.



Execute the import.



Display the monitoring to check if the import process was successfully completed.

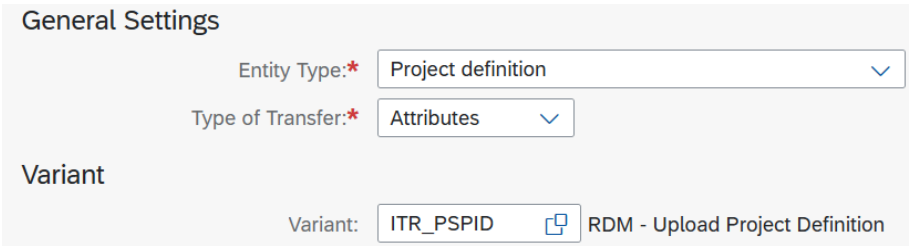
Object Type	Run Number	Start Date	Start Time	End Date	End Time	Duration	Status	Application	User
Work Break Down Structure	10000079	30.03.2026	15:23:16	30.03.2026	15:23:18	00:00:02		Data Load Import	IT_ALL_06

3.17.2.4 Upload

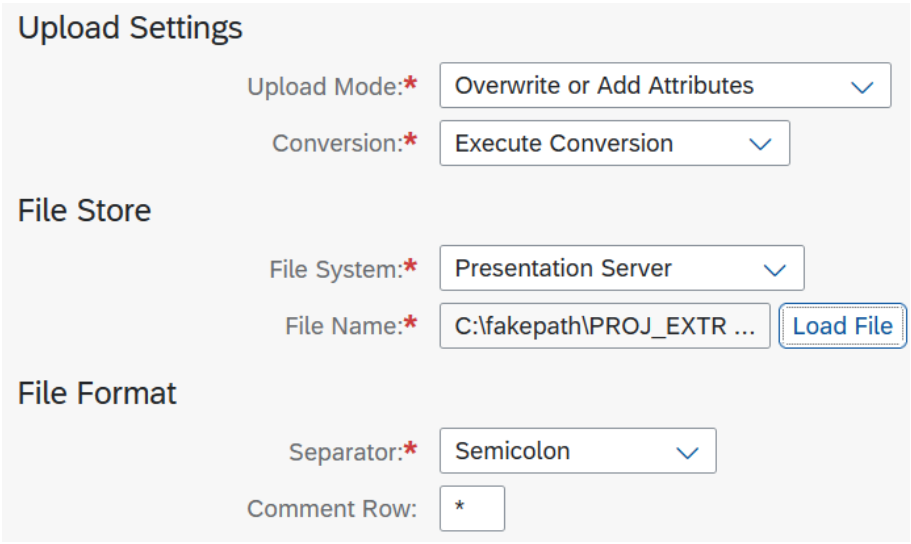
The WBS Element Hierarchy must be uploaded separately via File Upload but is also possible to upload all three files if all of them are downloaded as .txt files. This section describes how to upload all of them via File Upload. If the Project Definitions and the WBS Elements have been loaded via Data Import skip the next two sections.

3.17.2.4.1 Upload Project Definition

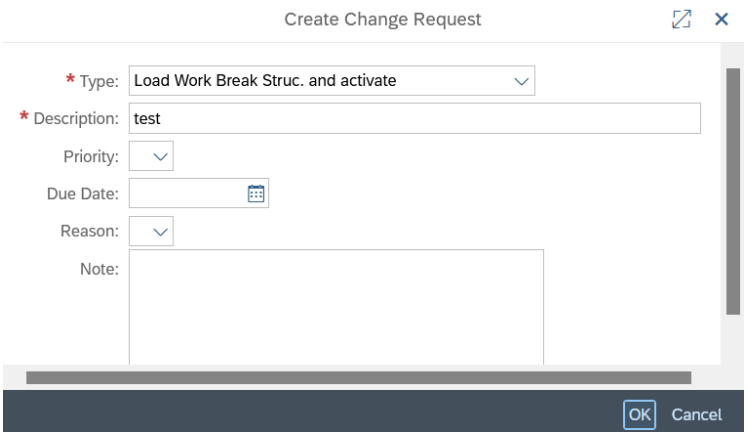
Open the „Upload File“ tile in the Fiori Launchpad and select the ITR_PSPID variant.



Load the *PROJ_EXTRACT* File in Step-3.

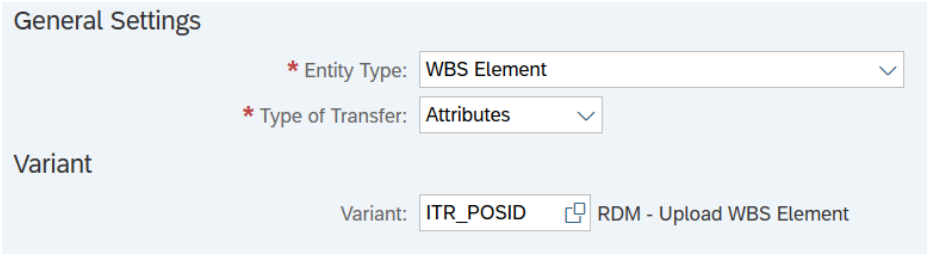


Press „Next“ and create a Change Request. Then, execute the upload.

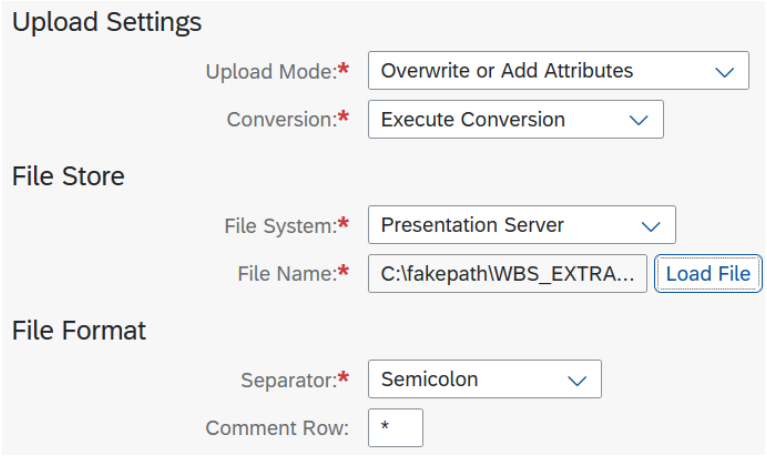


3.17.2.4.2 Upload WBS Elements

Open the „Upload File“ tile in the Fiori Launchpad and select the ITR_POSID variant.

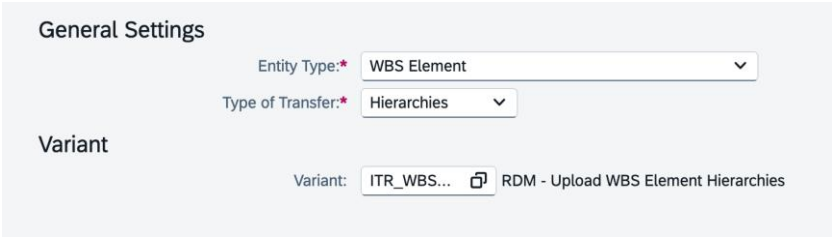


Load the *WBS_EXTRACT* File in Step-3. Then, execute the upload.

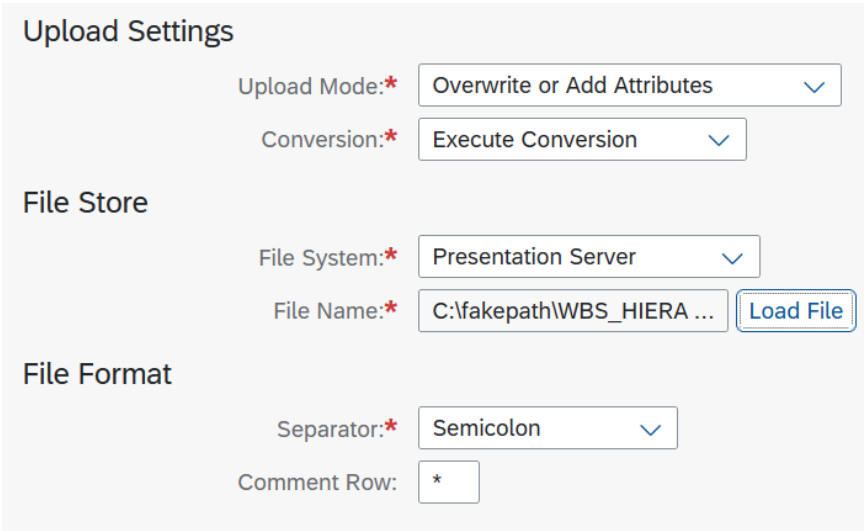


3.17.2.5 Upload WBS Hierarchy

Open the „Upload File" tile in the Fiori Launchpad and select the *ITR_WBSEH* variant.



Load the *WBS_HIERARCHY_EXTRACT* File in Step-3. Then, execute the upload.



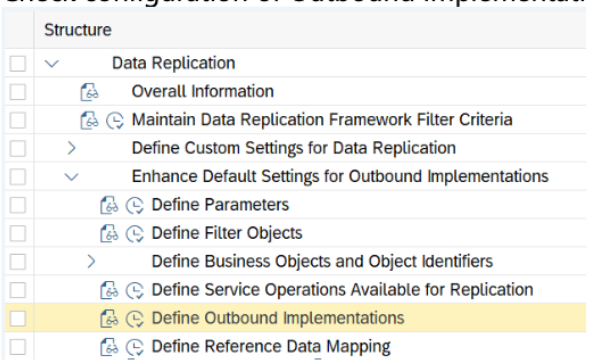
3.18 Configure Initial Load of Exchange Rates

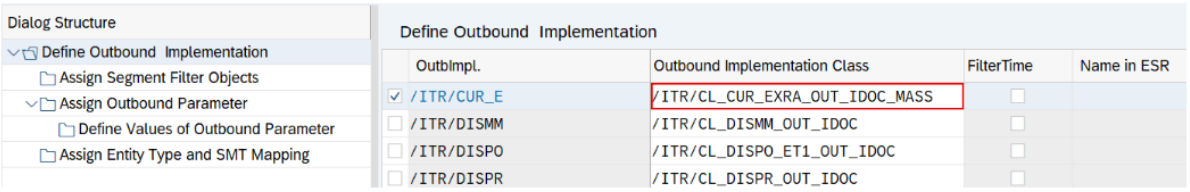
For customers processing a large volume of exchange rates during an initial mass load, a potential performance issue has been identified in the standard system setup, which can lead to application dumps depending on the memory size of the system to be loaded.

To resolve this issue and ensure stable performance during the initial mass load of exchange rates, you must apply a specific configuration with regards to the IDoc packaging size for the relevant outbound implementation.

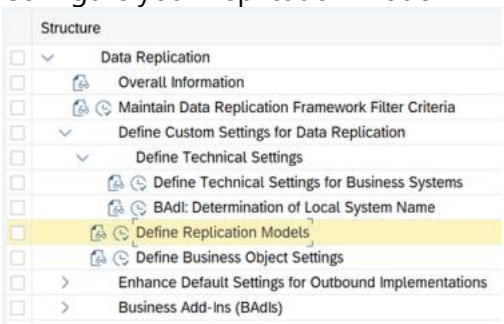
Start transaction DRFIMG.

Check or adjust the configuration of the Outbound Implementation.

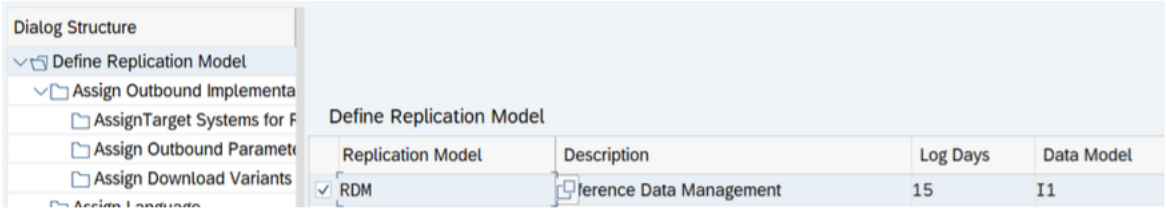




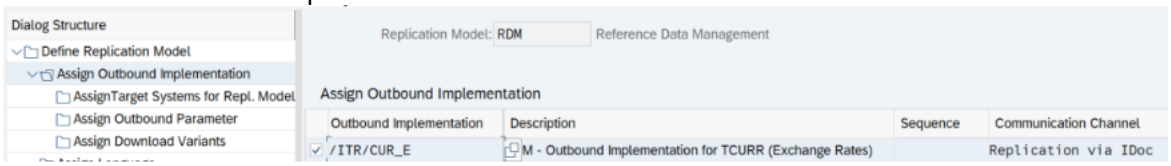
Configure your Replication Model



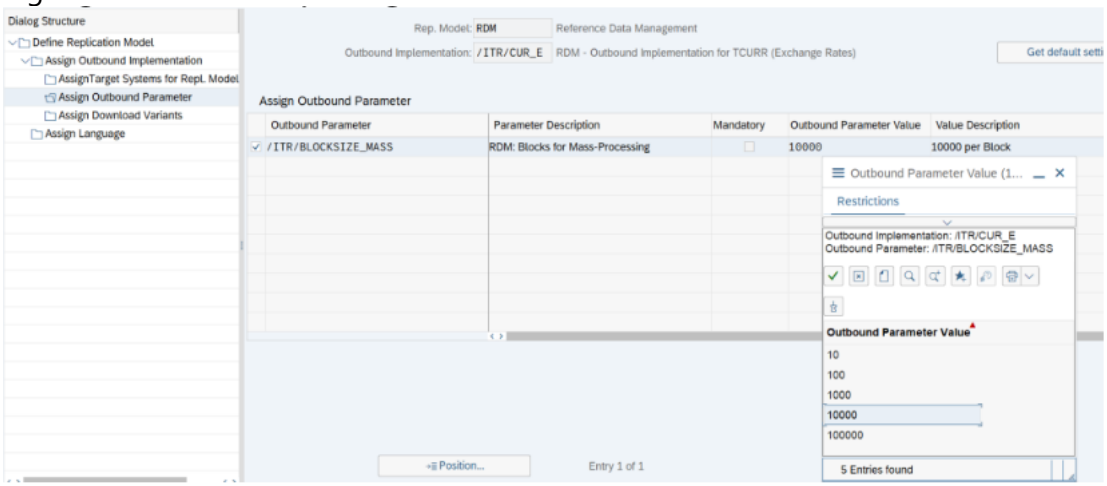
Select your Replication Model



Select the outbound implementation.



Within the configuration settings for the outbound implementation, configure a suitable package size.



It is recommended to set the IDoc package size such that the system creates IDocs in packages of 10,000. This package size ensures stable performance during the initial mass load and that the receiving system can process all packages efficiently. Please adjust if this setting does not meet the requirements of your SAP system. You might use higher or lower numbers depending on your system sizing.

3.19 Set Up Local Staging Area

The configuration of the Local Staging Area in the SAP receivers can be done by importing BC Set /ITR/RDM_STAGING_<n> "MDG RDM Staging Settings" (see: 3.3 "Activate Business Configuration Sets"). Make sure that you must do this for each SAP receiver.

Also make sure that you define users with role /ITR/MDG_STAGING_<n> "User for Local Staging Area". See section 3.4 "Configure User Roles and Authorizations".

Note: any user, who wants to run the transaction /ITR/STAGING needs write-access to the transport directory of the corresponding ERP system. In most cases, the directory usually is "/usr/sap/trans", but could also be called differently. It might therefore be necessary to check the folder path with SAP basis and provide access for the user.

3.20 Solution Manager Integration

Please refer to the separate document "MDF Solution Manager Integration".

To obtain more information about these and other project specific enhancements contact: support@itego.de - Subject: "RDM Solution Manager Integration".

3.21 Project Specific Enhancements

MDG RDM offers functionalities which are not activated and not supported in the standard delivery but can be activated based on project specific requirements. Examples are:

- Status Net (Controlled maintenance of an object status)
- Object Deletion (Governed deletion of objects)

To obtain more information about these and other project specific enhancements contact: support@itego.de - Subject: "RDM Project Specific Enhancements".

4 Configuration for Reference Data Harmonization

4.1 Activate Business Configuration Sets

4.1.1 SAP MDG: BC Sets

4.1.1.1 Software Component ITG

To get predefined Configuration Groups for Reference Data Harmonization please also consider activating the following BC Set (please be aware that these are needed only when you use the MDG system for the synchronization of reference data types which are not covered in your Reference Data Governance scenarios):

- /ITR/RDH_CONFIG_GROUPS_<n> RDH Configuration Groups

4.2 Configure User Roles and Authorizations

4.2.1 Reference Data Harmonization – Sender

The following roles are delivered for Reference Data Harmonization (on the sender system which might be in most case SAP MDG) and define which actions are allowed for which user.

- /ITR/ITEGO_MDG_RDH_DISP_<n>S Display Functions in the Sender-System
- /ITR/ITEGO_MDG_RDH_BUSINES_<n>S Business Functions in the Sender-System
- /ITR/ITEGO_MDG_RDH_EXPERT_<n>S Expert Functions in the Sender-System

4.2.2 Reference Data Harmonization – Receiver

The following roles are delivered for Reference Data Harmonization (on the receiver system which might be the SAP MDG system when data is consolidated in a first step in SAP MDG) and define which actions are allowed for which user.

- /ITR/ITEGO_MDG_RDH_DISP_<n>E Display Functions in the Receiver-System
- /ITR/ITEGO_MDG_RDH_BUSINES_<n>E Business Functions in the Receiver-System
- /ITR/ITEGO_MDG_RDH_EXPERT_<n>E Expert Functions in the Receiver-System

5 Configuration Examples

5.1 Data Replication based on SAP ALE

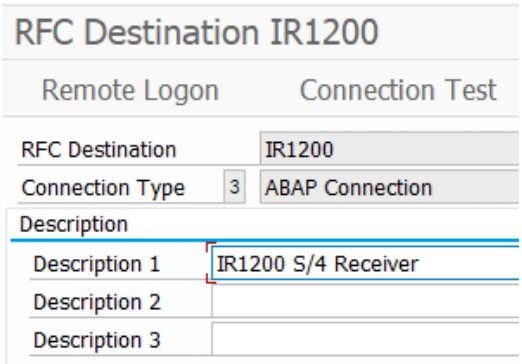
Most RDM objects will be replicated using a generic message type. This section shows how to set this up based on two systems:

- RDM for MDG (sender): System IH1, client 100
- SAP Business System (receiver): System IR1, client 200

Please adjust the examples below to your own system landscape.

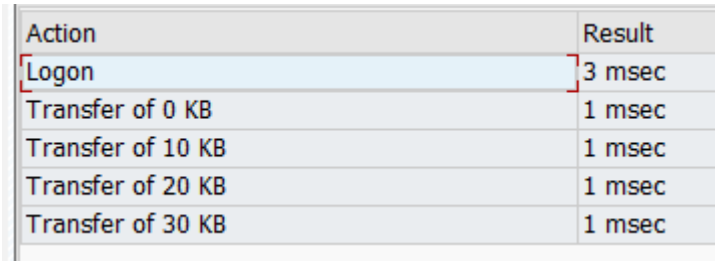
5.1.1 Prerequisite: RFC Destination

System IH1 100: Check or create an RFC destination to system IR1 200 using transaction SM59:





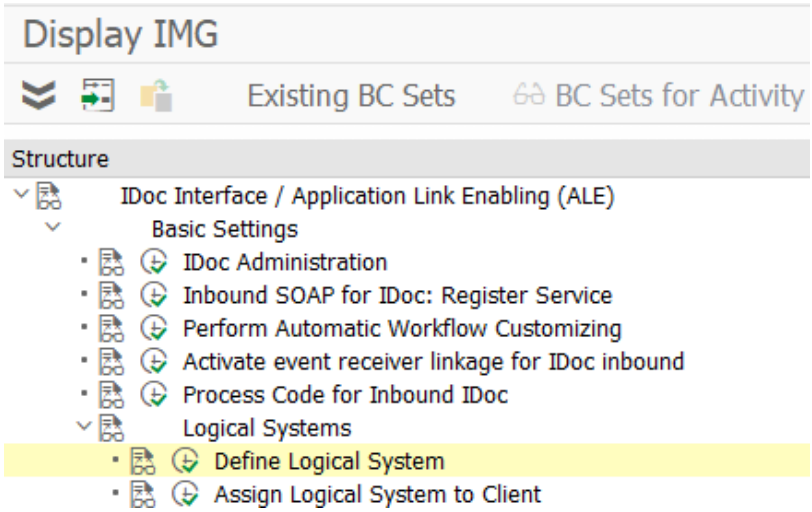
Connection Test should be successful.

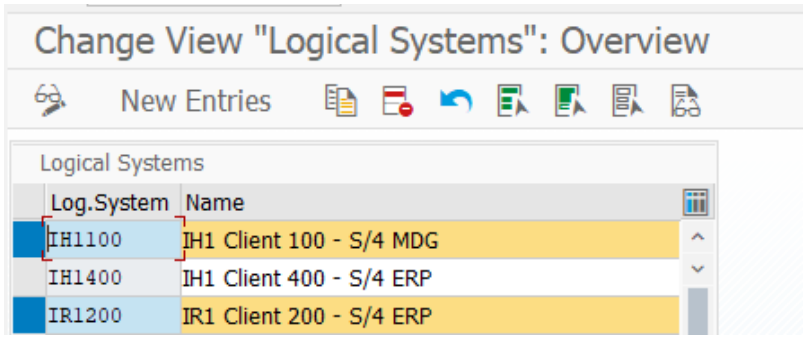


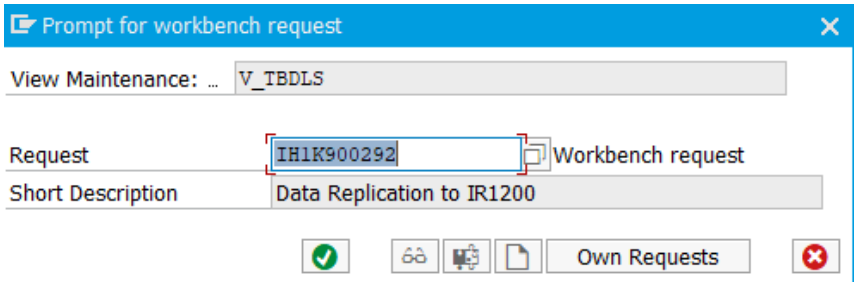
Action	Result
Logon	3 msec
Transfer of 0 KB	1 msec
Transfer of 10 KB	1 msec
Transfer of 20 KB	1 msec
Transfer of 30 KB	1 msec

5.1.2 Define Logical System and Check or Create Business System

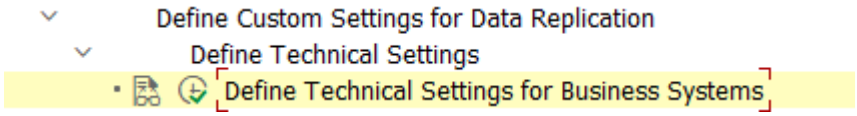
System IH1 100, transaction SALE:

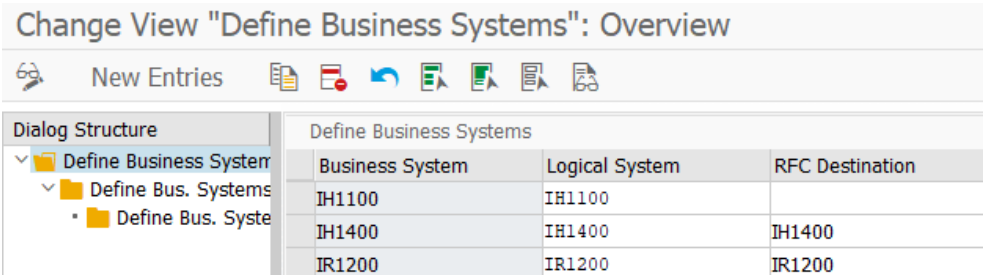






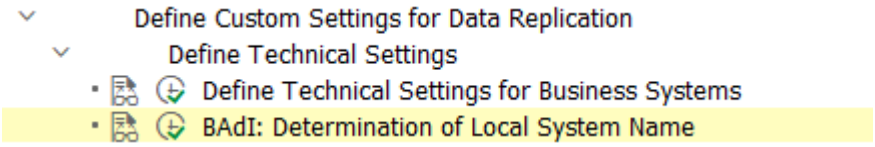
Check or Create Business System: System IH1 100, transaction DRFIMG.



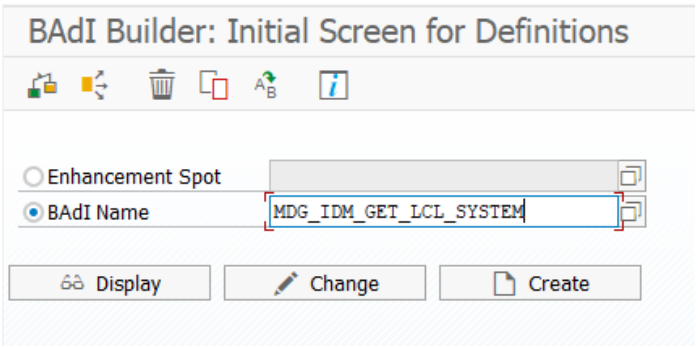


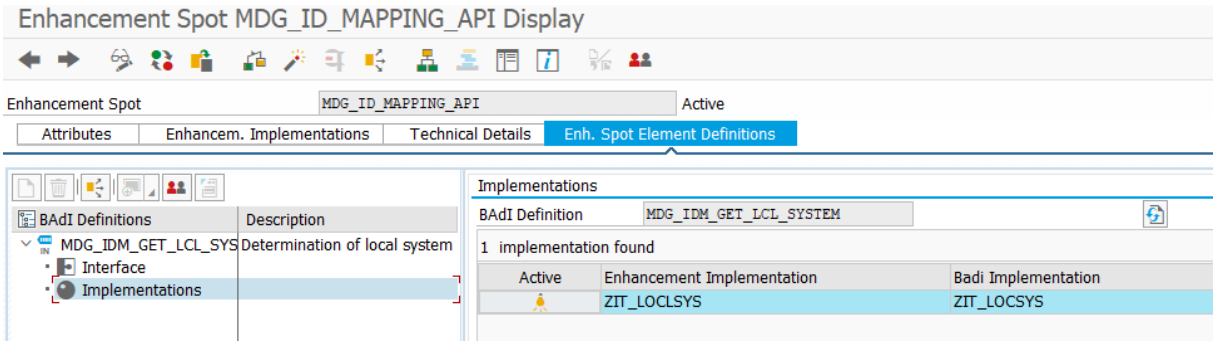
5.1.3 Check or Create BAdI for determination of local system name

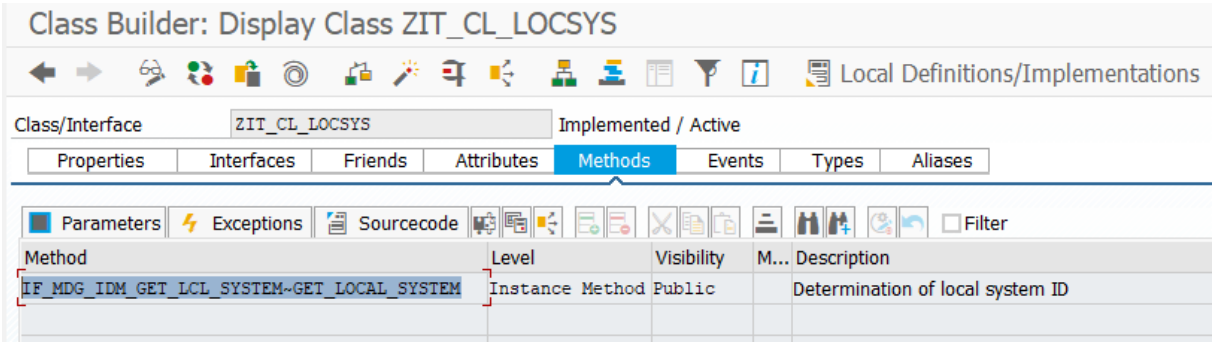
System IH1 100, transaction DRFIMG:



Or use transaction SE18, BAdI Name: MDG_IDM_GET_LCL_SYSTEM








```
method IF_MDG_IDM_GET_LCL_SYSTEM~GET_LOCAL_SYSTEM.
*! This method determines the local business system via the ALE logical sys
tem assigned to it.
* If no business system is maintained, it returns an empty value.

  DATA:
    lv_own_logical_system TYPE logsys,
    ls_bs_tech            TYPE mdg_s_bus_sys_tech,
    lv_not_found          TYPE boole_d.

  CALL FUNCTION 'OWN_LOGICAL_SYSTEM_GET'
    IMPORTING
      own_logical_system             = lv_own_logical_system
    EXCEPTIONS
      own_logical_system_not_defined = 1
      OTHERS                         = 2.

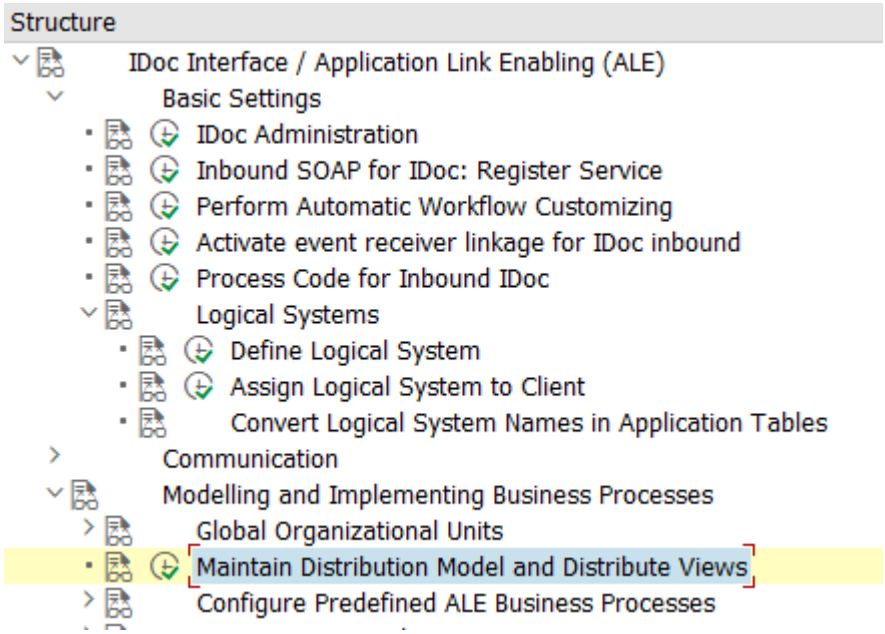
  cl_mdg_bs_access_cust_data=>select_bs_data_for_logsys(
    EXPORTING
      iv_logsys = lv_own_logical_system
    IMPORTING
     es_bs_tech   = ls_bs_tech
     ev_not_found = lv_not_found ).

  IF lv_not_found = abap_false.
    ev_local_system = ls_bs_tech-business_system.
  ENDIF.

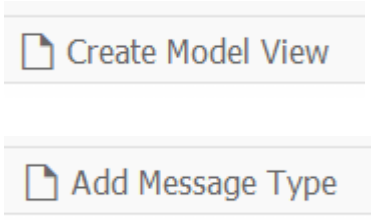
  endmethod.
```

5.1.4 Define Distribution Model

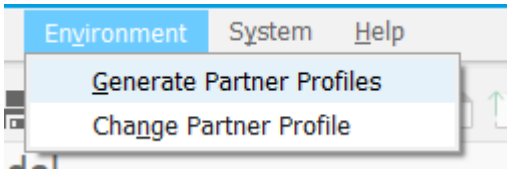
Use transaction SALE (or BD64):

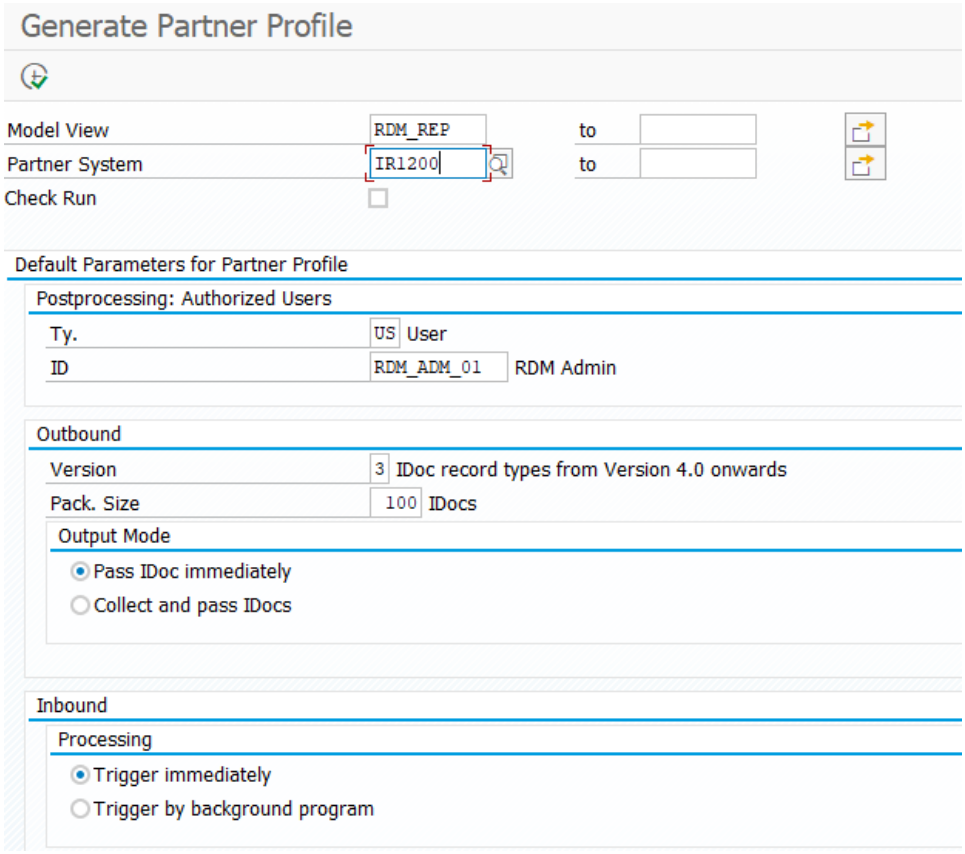


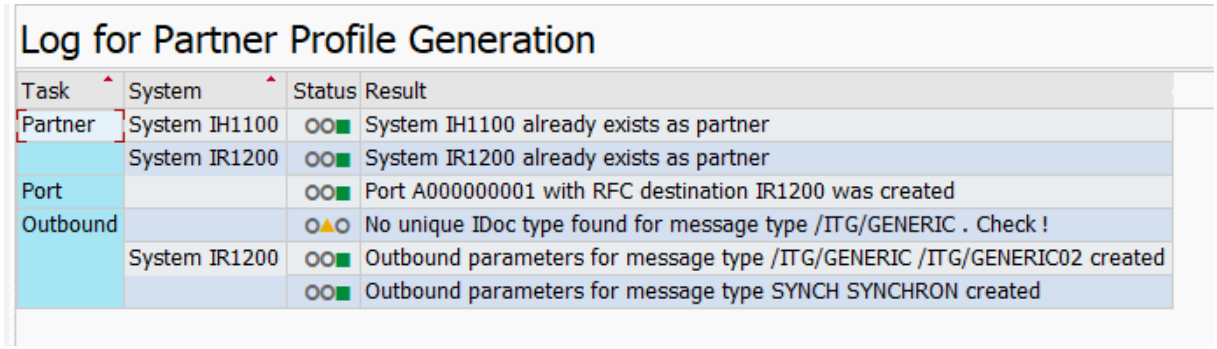
Create Model View and Add Message Type



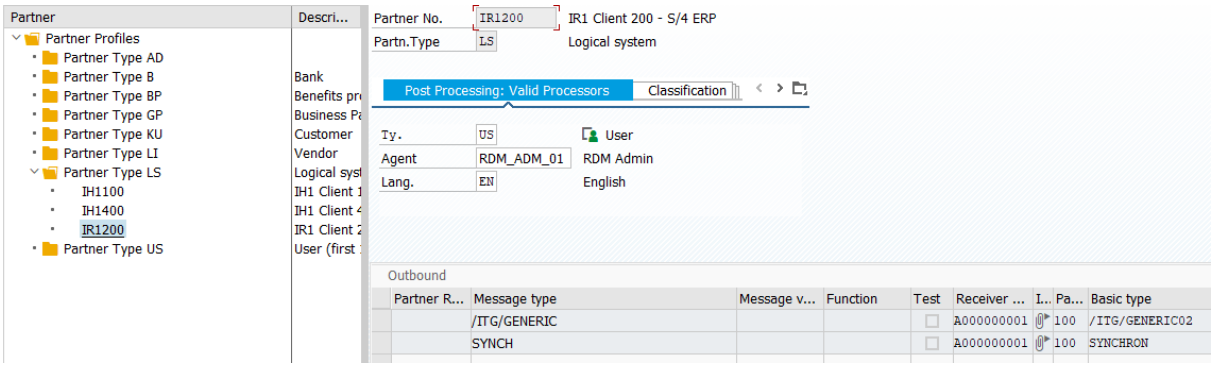




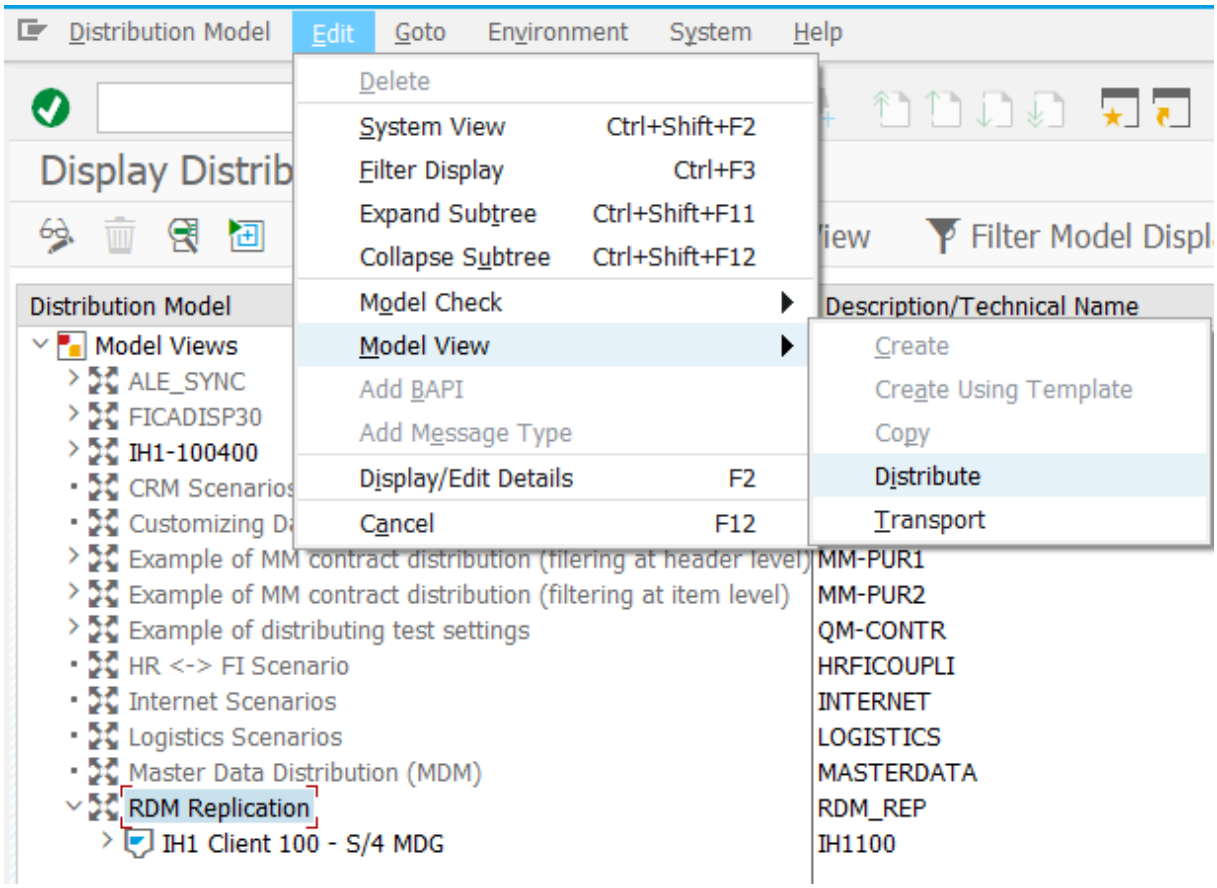


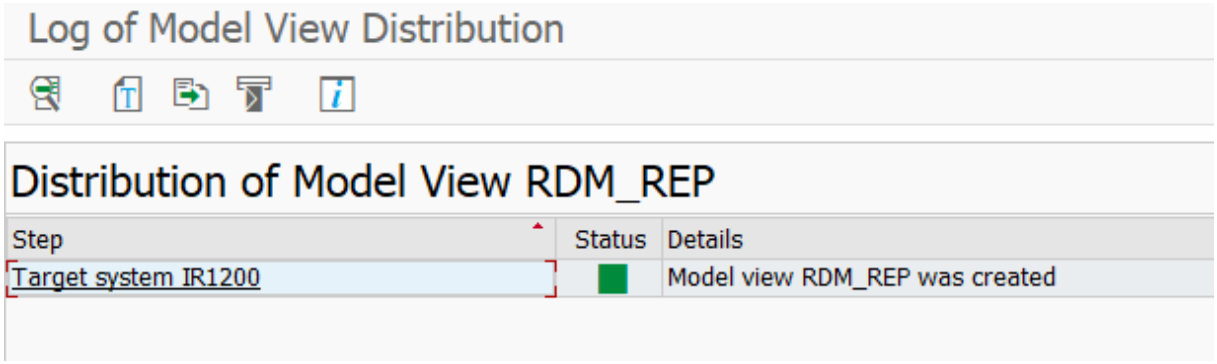


Check in transaction WE20:



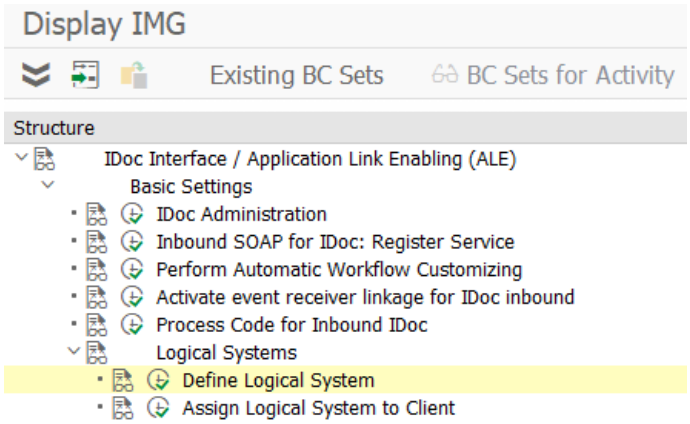
BD64: Distribute Model View to receiver (IR1200)

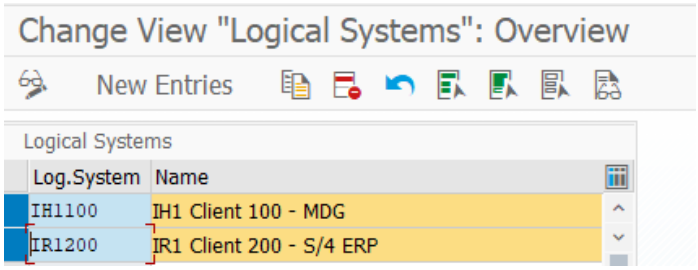


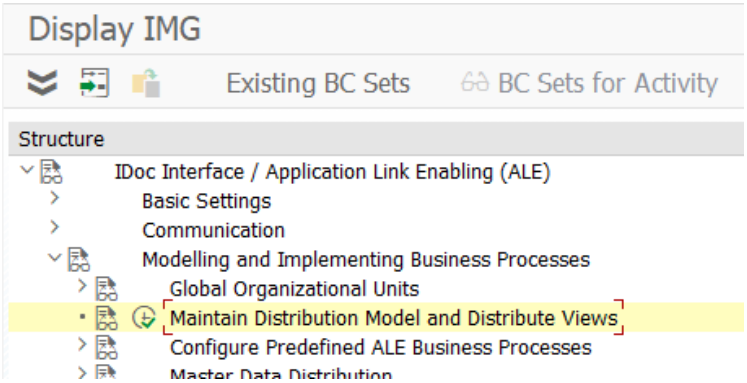


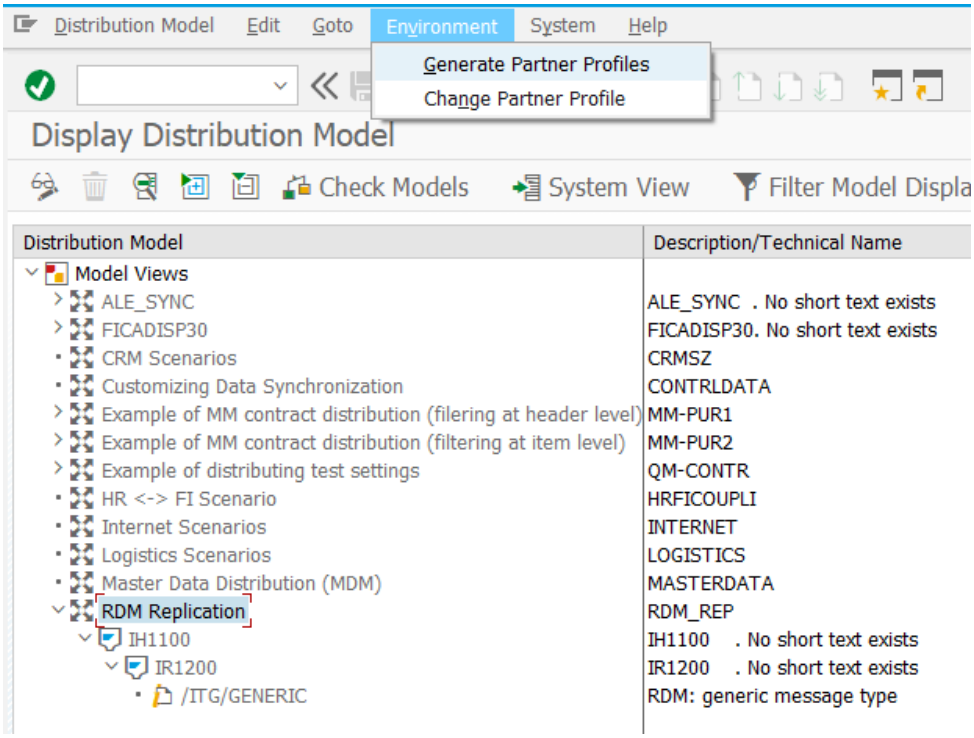
Log On to system IR1200 and execute transaction SALE:

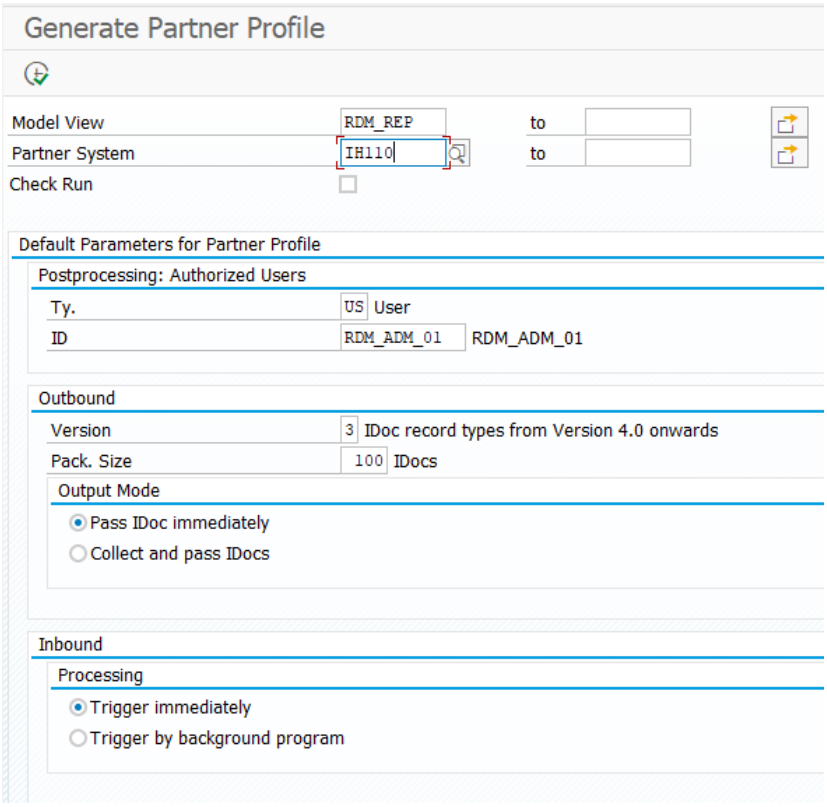
Check Logical Systems



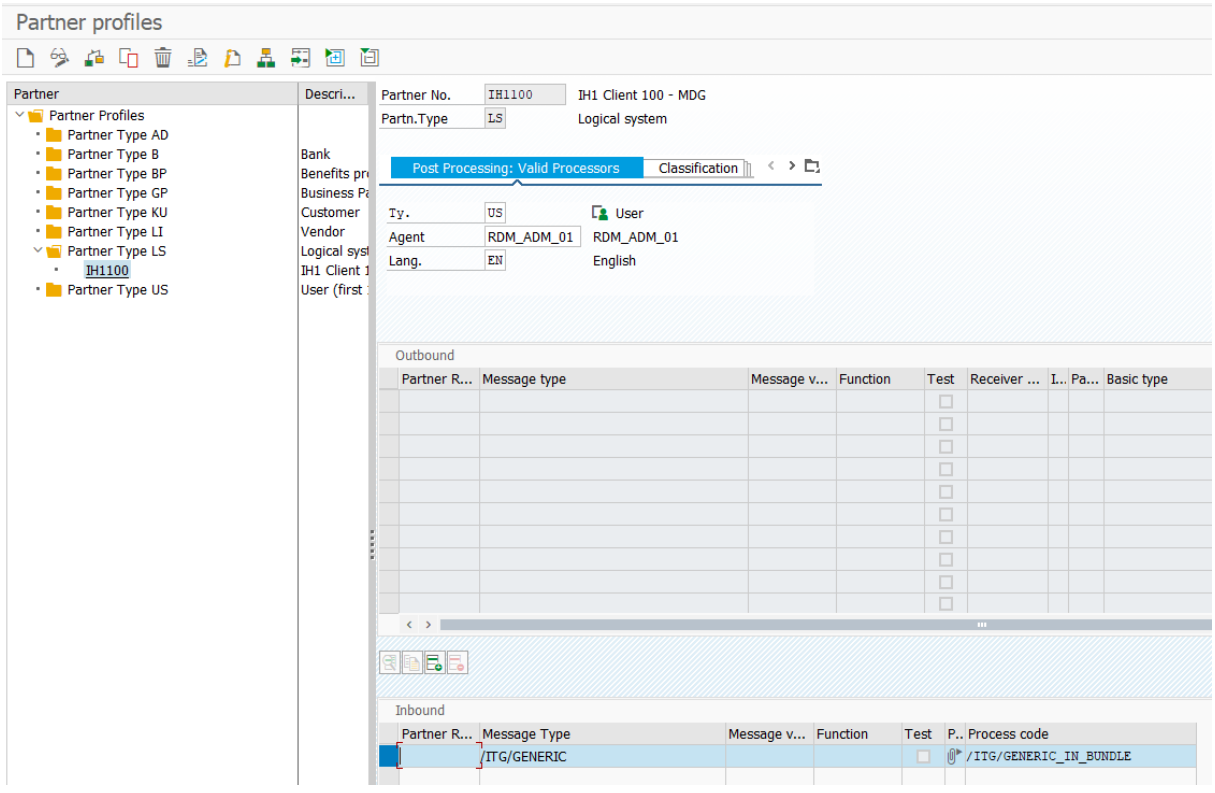


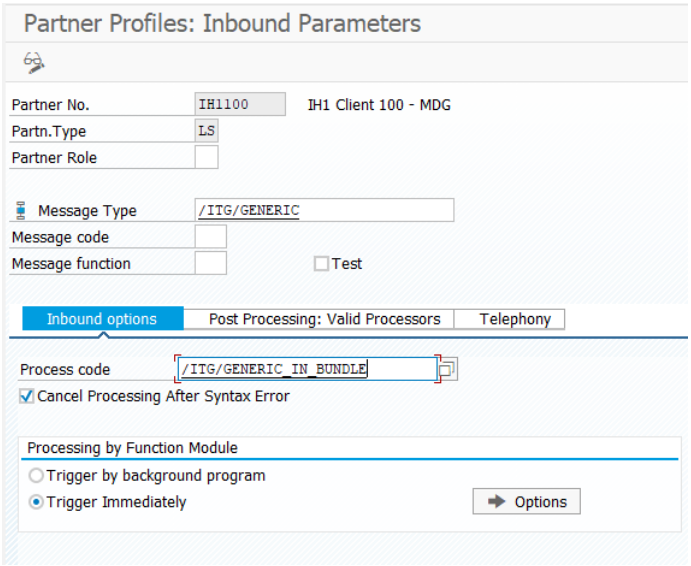

Display IMG
Existing BC Sets
BC Sets for Activity
Structure
IDoc Interface / Application Link Enabling (ALE)
Basic Settings
Communication
Modelling and Implementing Business Processes
Global Organizational Units
Maintain Distribution Model and Distribute Views
Configure Predefined ALE Business Processes


Distribution Model
Edit
Goto
Environment
System
Help
Generate Partner Profiles
Change Partner Profile
Display Distribution Model
Check Models
System View
Filter Model Displa
Distribution Model
Description/Technical Name
Model Views
ALE_SYNC
ALE_SYNC . No short text exists
FICADISP30
FICADISP30. No short text exists
CRM Scenarios
CRMSZ
Customizing Data Synchronization
CONTRLDATA
Example of MM contract distribution (filering at header level)
MM-PUR1
Example of MM contract distribution (filtering at item level)
MM-PUR2
Example of distributing test settings
QM-CONTR
HR <-> FI Scenario
HRFICOUPLI
Internet Scenarios
INTERNET
Logistics Scenarios
LOGISTICS
Master Data Distribution (MDM)
MASTERDATA
RDM Replication
RDM_REP
IH1100
IH1100 . No short text exists
IR1200
IR1200 . No short text exists
/ITG/GENERIC
RDM: generic message type

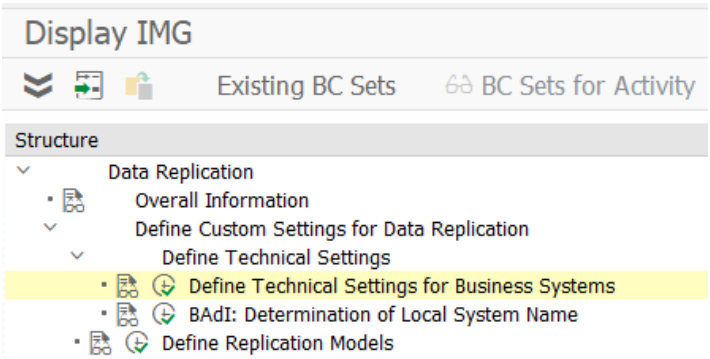


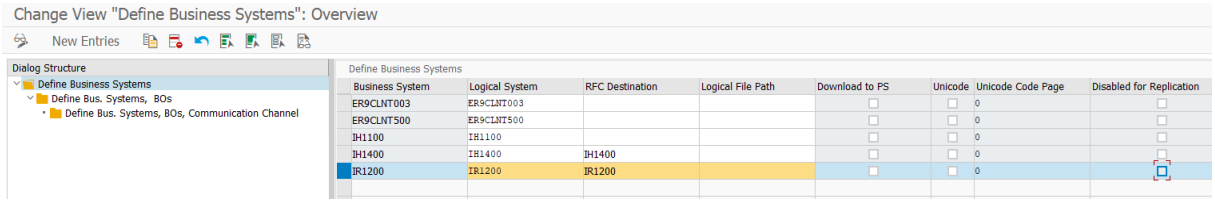
Check in WE20



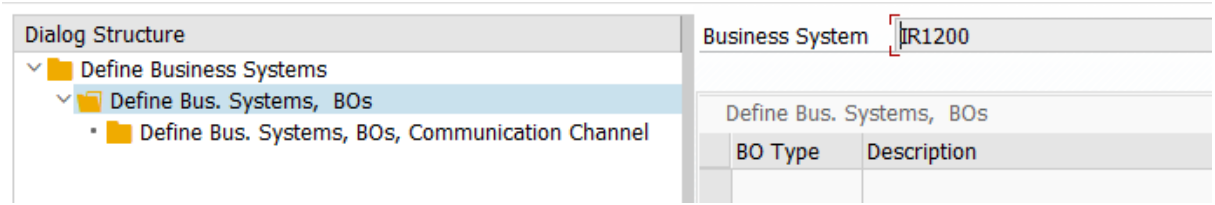


System IH1 100: Check DRFIMG:

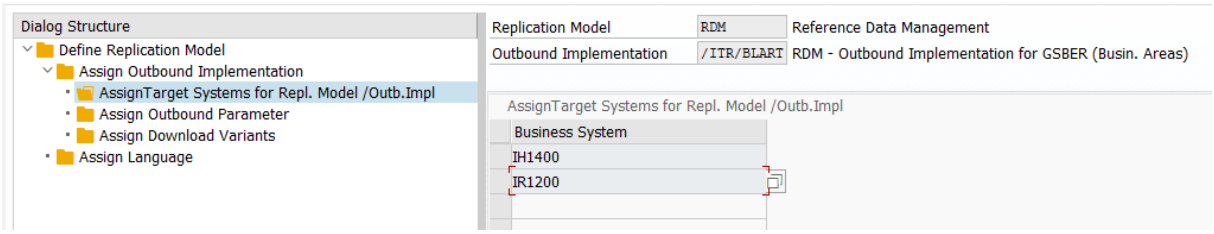


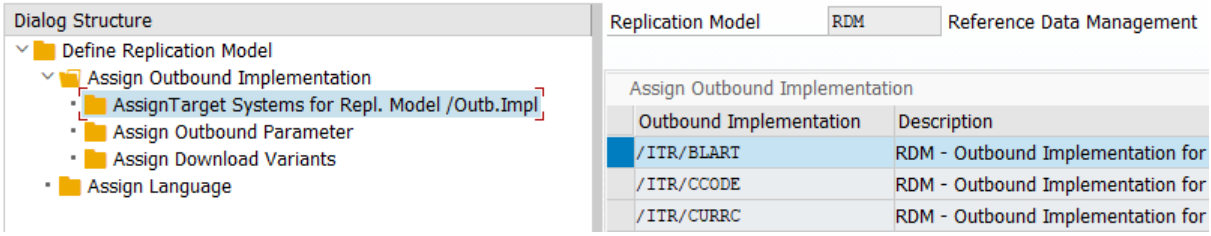


Nothing else to be configured:

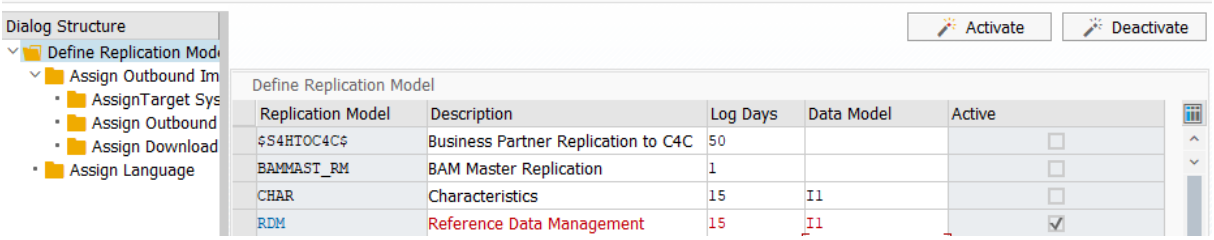


For every object in scope for this receiver system define target system and outbound implementation:





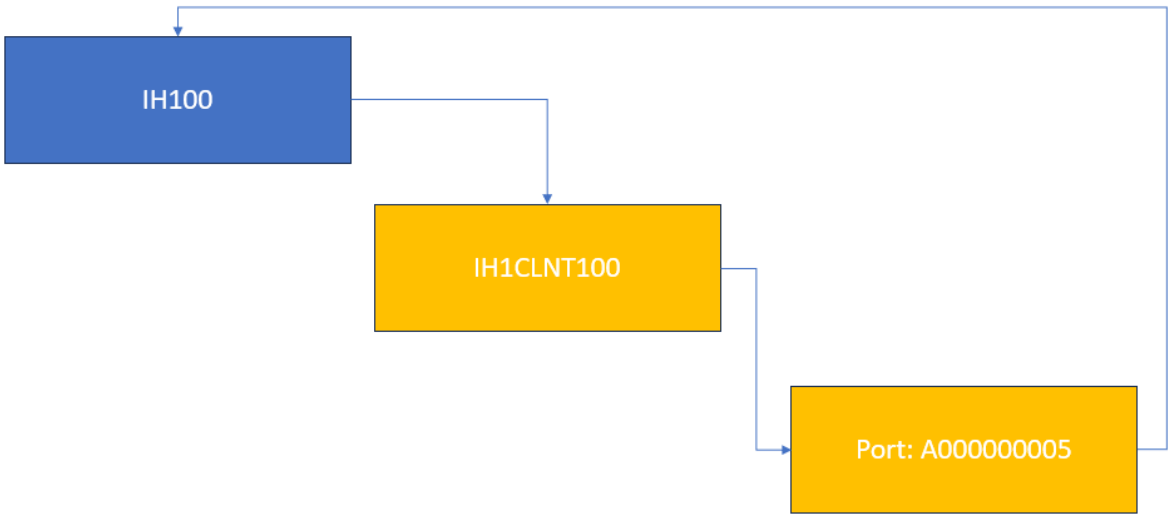
Data Replication Model needs to be active:



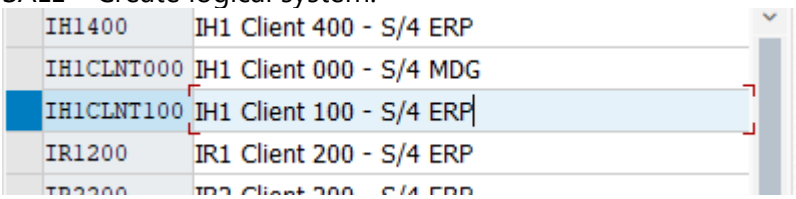
5.1.5 Business Governance: Data Replication to MDG client

This section shows a possible data replication configuration for the data replication from the MDG for RDM data persistence (based on the "Flex Mode") to the ERP tables within the same client. This is necessary to avoid inconsistencies for all object types that are in the scope for the Business Governance scenario. It is important to understand that (like in MDG for Financials) the data persistence is decoupled from the corresponding ERP tables. Without this replication scenario below the data between the ERP layer and the MDG layer will start to differ after the data has been loaded (from the ERP tables) and is changed for the first time in MDG for RDM.

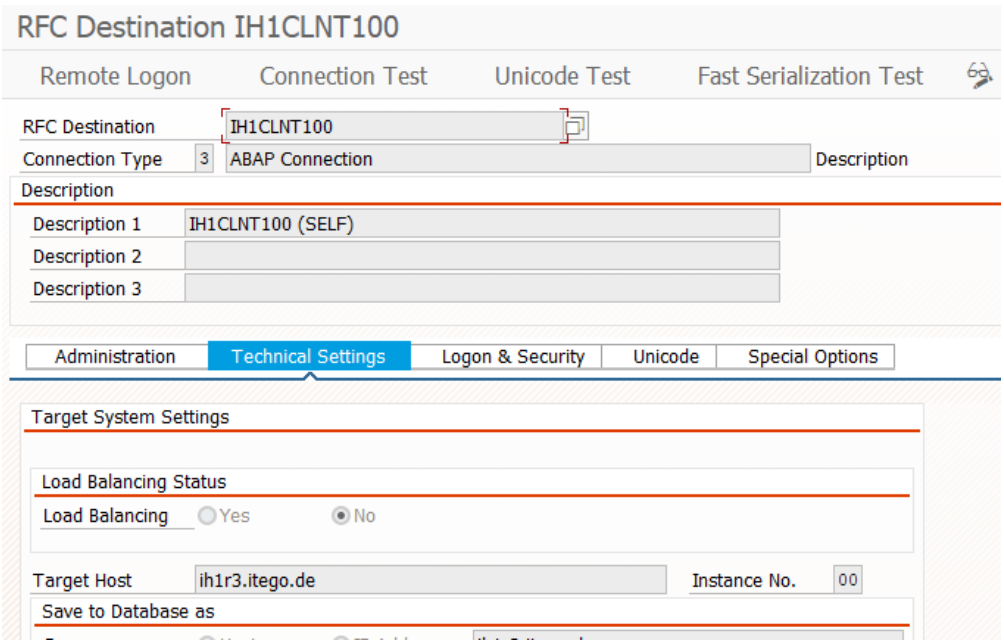
Create a new logical system as receiver: IH1CLNT100 and send back to IH1100.

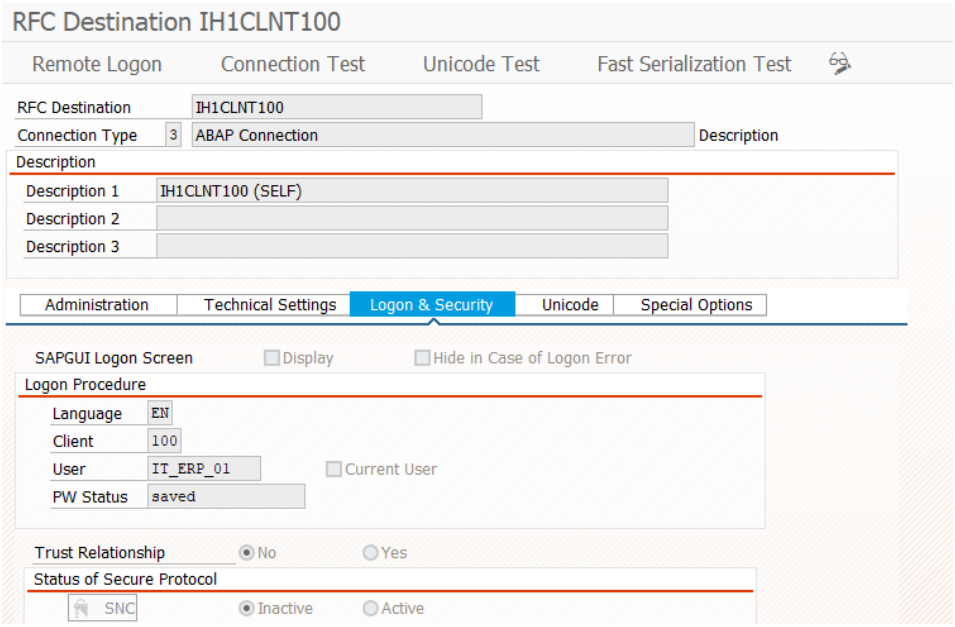


SALE – Create logical system.



SM59: Create RFC Destination



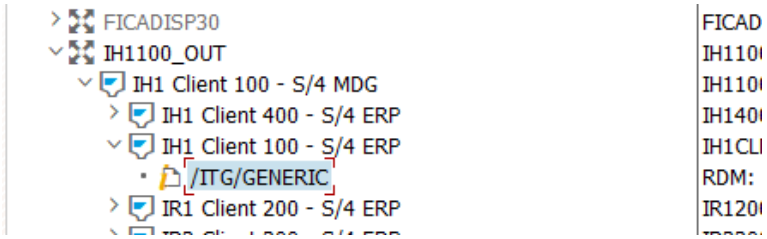


Note: A specific technical user needs to be defined and used. The screenshot only shows an example.

Connection Test

Action	Result
Logon	2 msec
Transfer of 0 KB	0 msec
Transfer of 10 KB	0 msec
Transfer of 20 KB	0 msec
Transfer of 30 KB	0 msec

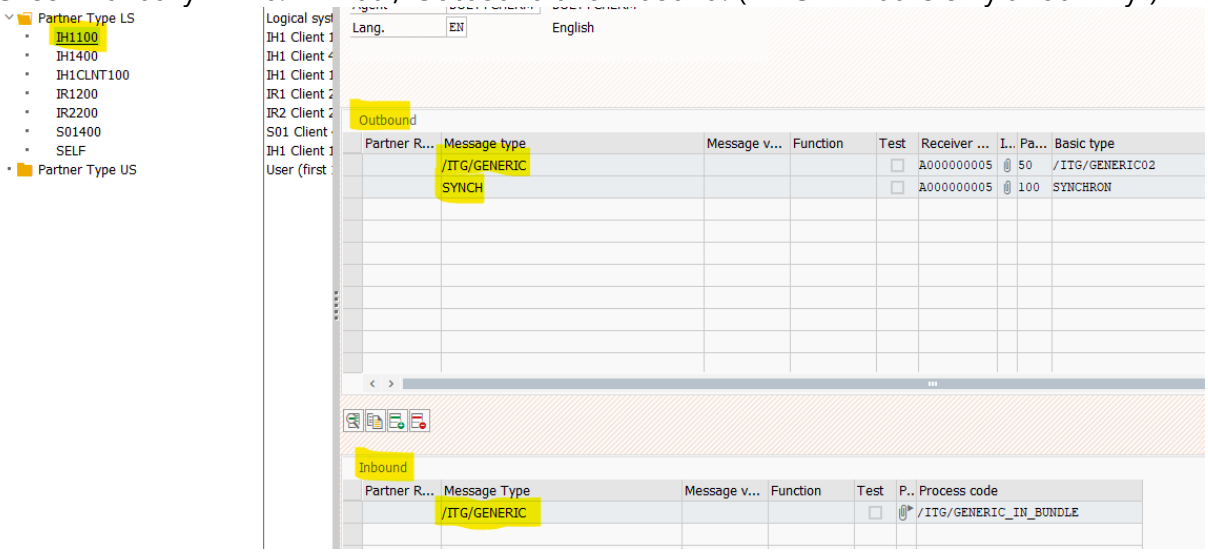
Add to Distribution Model

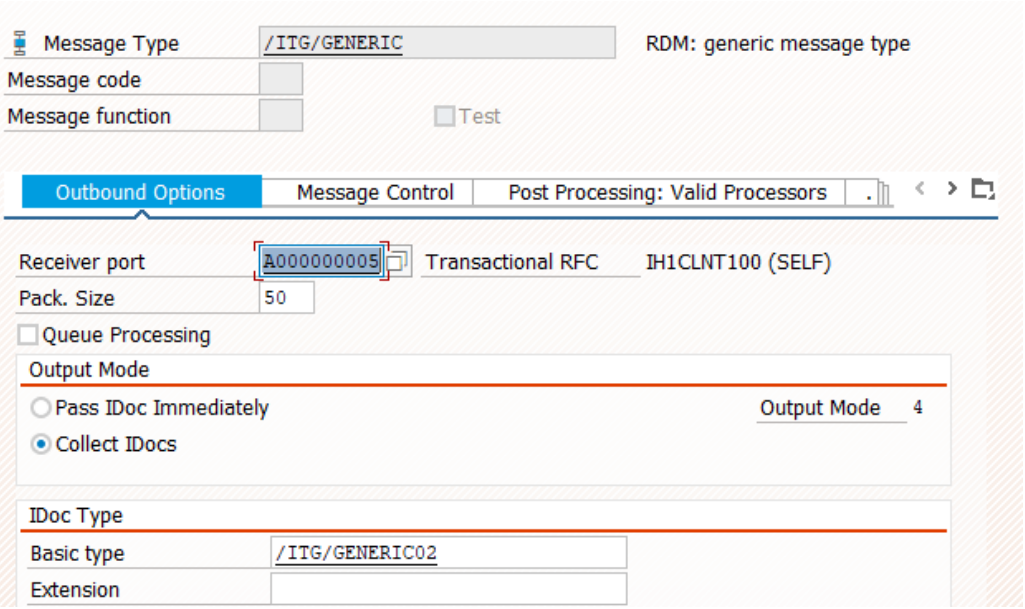


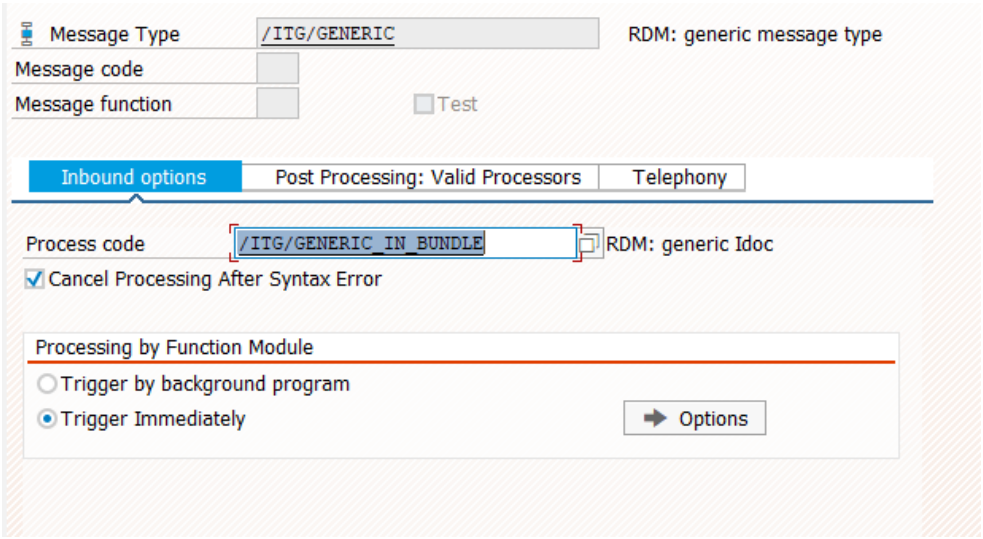
Generate Partner Profiles

Task	System	Status	Result
Partner	System IH1100		System IH1100 already exists as partner
	System IH1CLNT100		System IH1CLNT100 already exists as partner
Port	System IH1CLNT100		Port A000000005 with RFC destination IH1CLNT100 already exists
Outbound			No unique IDoc type found for message type /ITG/GENERIC . Check !
	System IH1CLNT100		Outbound parameters for message type /ITG/GENERIC /ITG/GENERIC02 already exist
			Outbound parameters for message type SYNCH SYNCHRON already exist

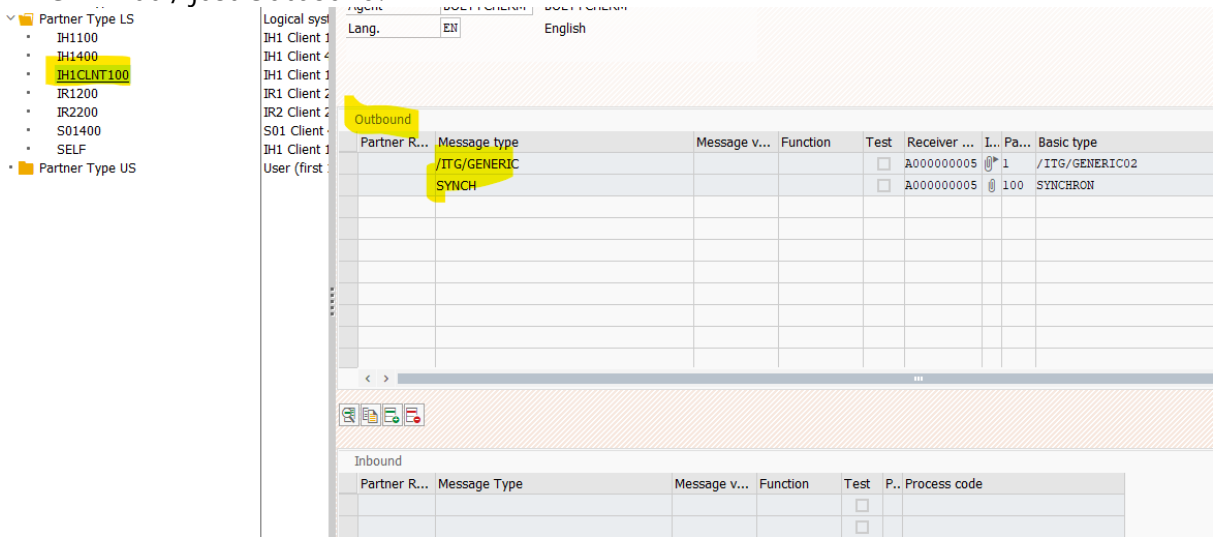
Check manually WE20: IH1100 / Outbound and Inbound! (IH1CLNT100 is only a “dummy”)

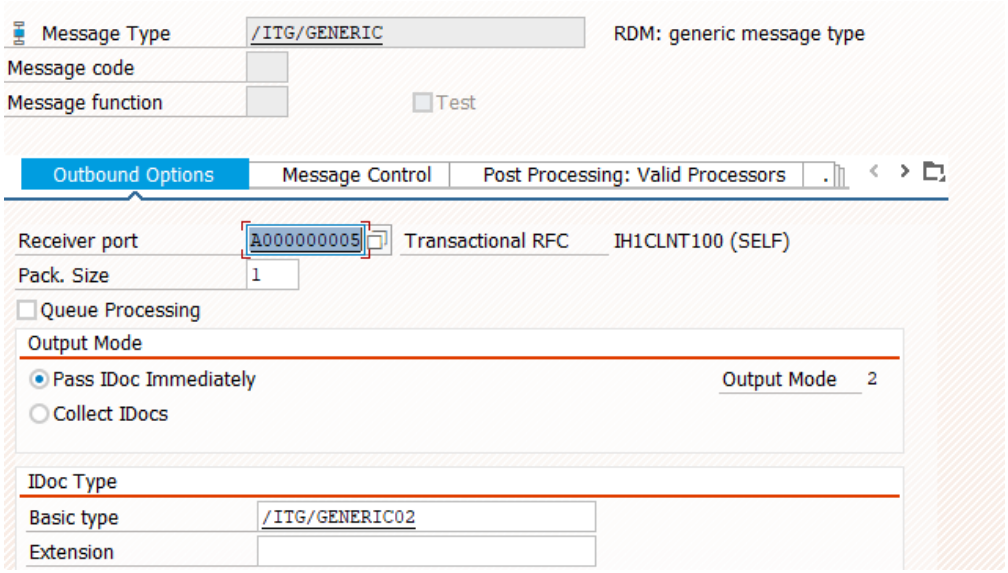




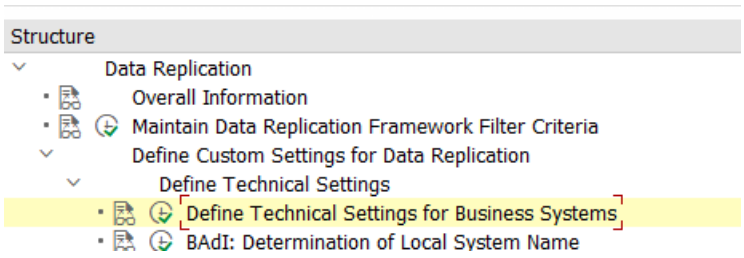


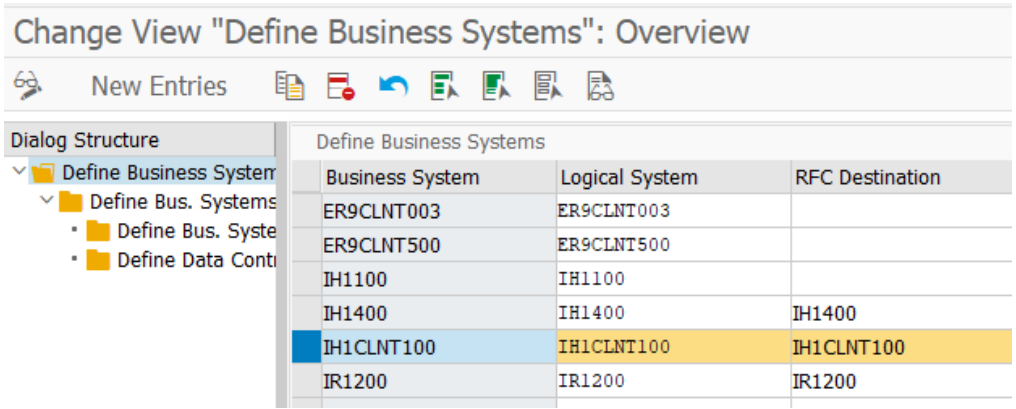
IH1CLNT100 / just Outbound:

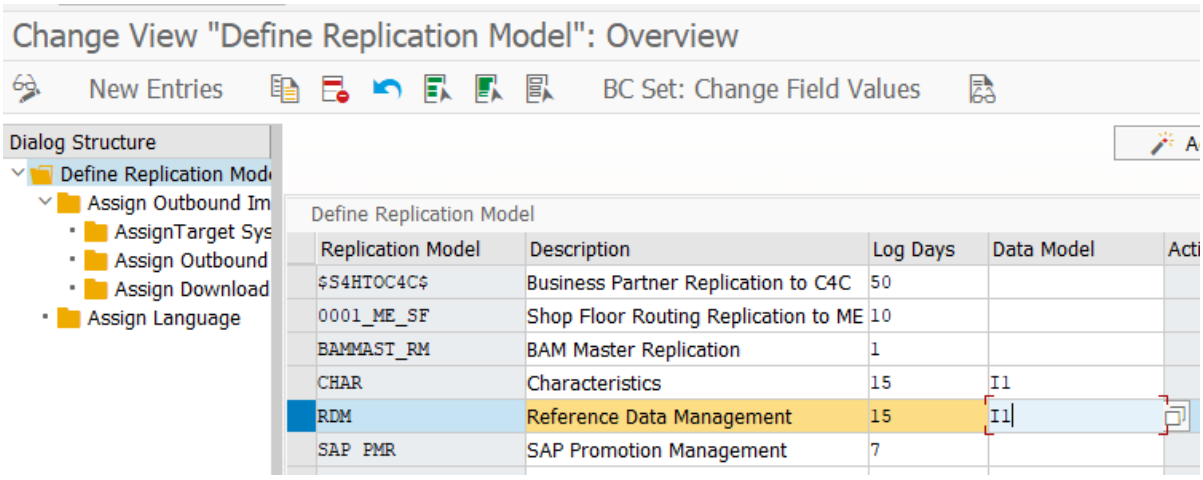



Message Type
/ITG/GENERIC
RDM: generic message type
Message code
Message function
Test
Outbound Options
Message Control
Post Processing: Valid Processors
Receiver port
A000000005
Transactional RFC
IH1CLNT100 (SELF)
Pack. Size
1
Queue Processing
Output Mode
Pass IDoc Immediately
Output Mode
2
Collect IDocs
IDoc Type
Basic type
/ITG/GENERIC02
Extension

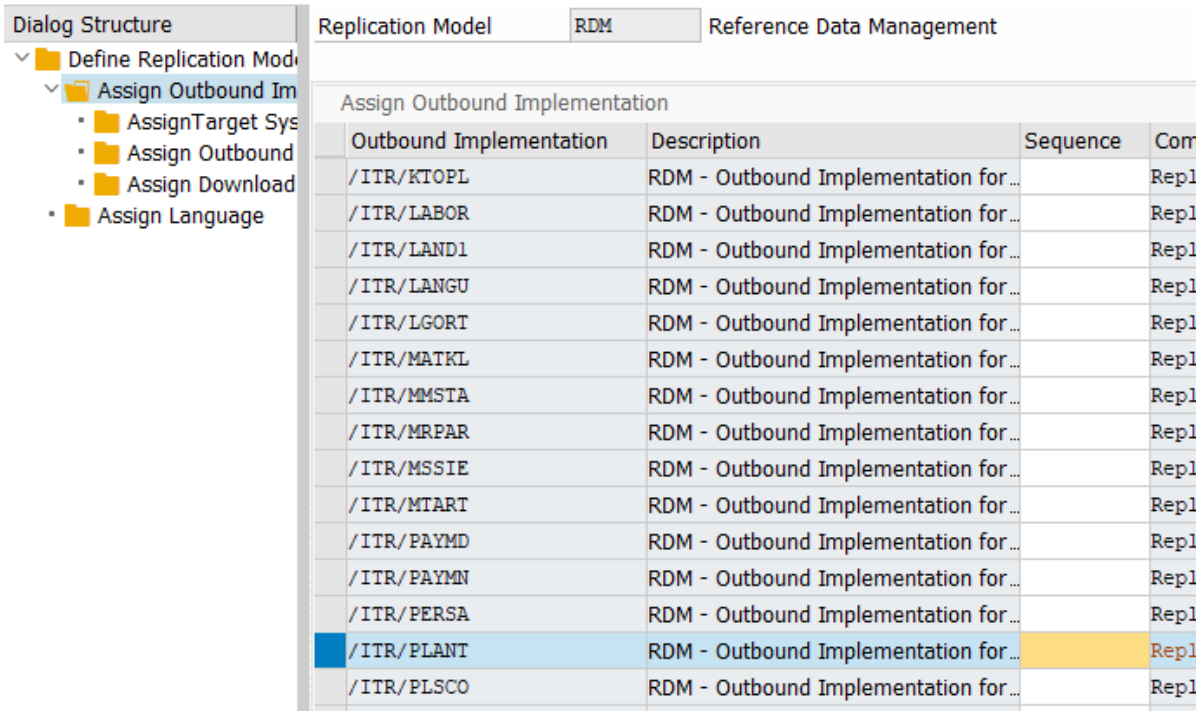
DRFIMG:

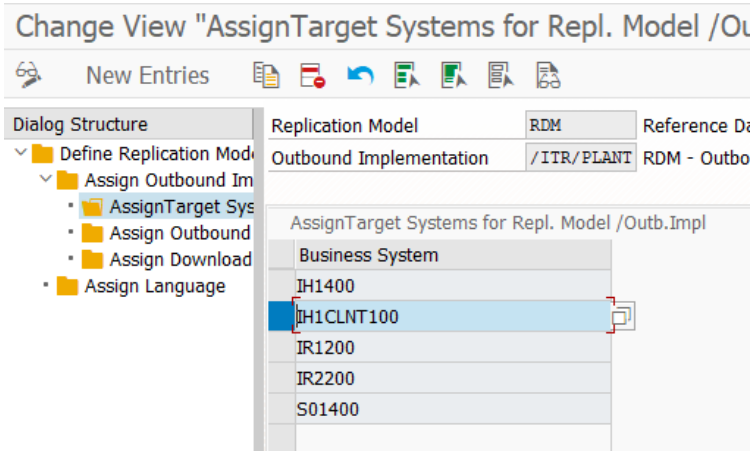

Structure
Data Replication
Overall Information
Maintain Data Replication Framework Filter Criteria
Define Custom Settings for Data Replication
Define Technical Settings
Define Technical Settings for Business Systems
BAdI: Determination of Local System Name


Change View "Define Business Systems": Overview
New Entries
Dialog Structure
Define Business System
Define Bus. Systems
Define Bus. Syste
Define Data Contr
Define Business Systems
Business System
Logical System
RFC Destination
ER9CLNT003
ER9CLNT003
ER9CLNT500
ER9CLNT500
IH1100
IH1100
IH1400
IH1400
IH1400
IH1CLNT100
IH1CLNT100
IH1CLNT100
IR1200
IR1200
IR1200



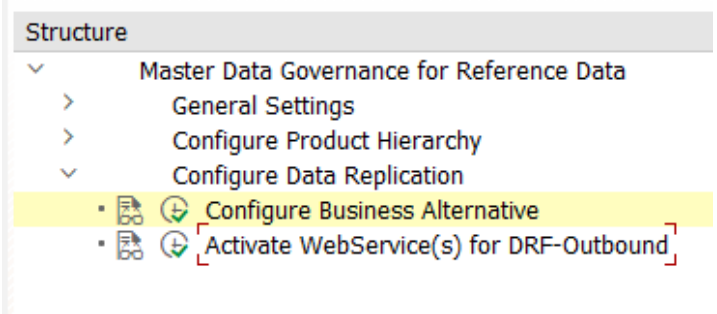
The next step needs to be done for each object type in scope for Business Governance ("Direct Update"). The below list is just an example.

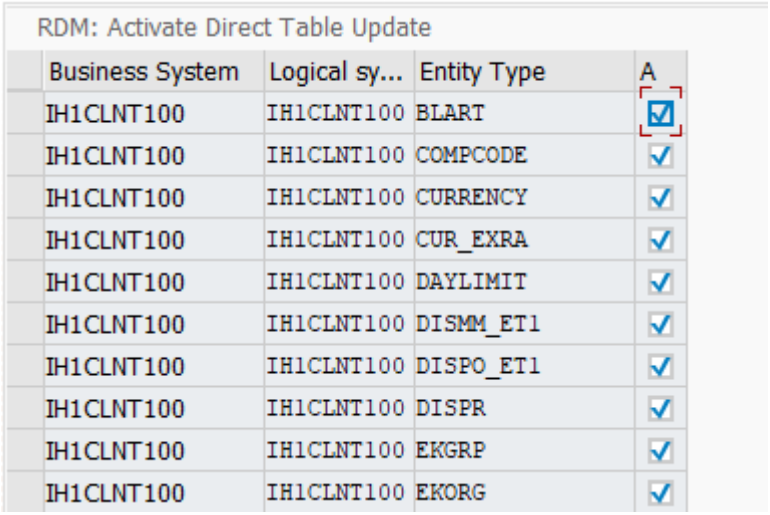




Last step: Activate the Business Governance for all object types in scope for Business Governance ("Direct Update"). The below list is again just an example.







5.2 Data Replication based on Webservices

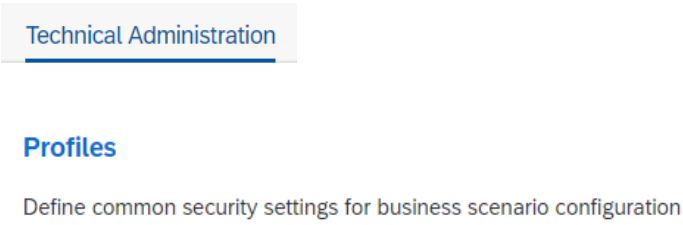
This section shows how to set up a Webservice based communication using the following systems:

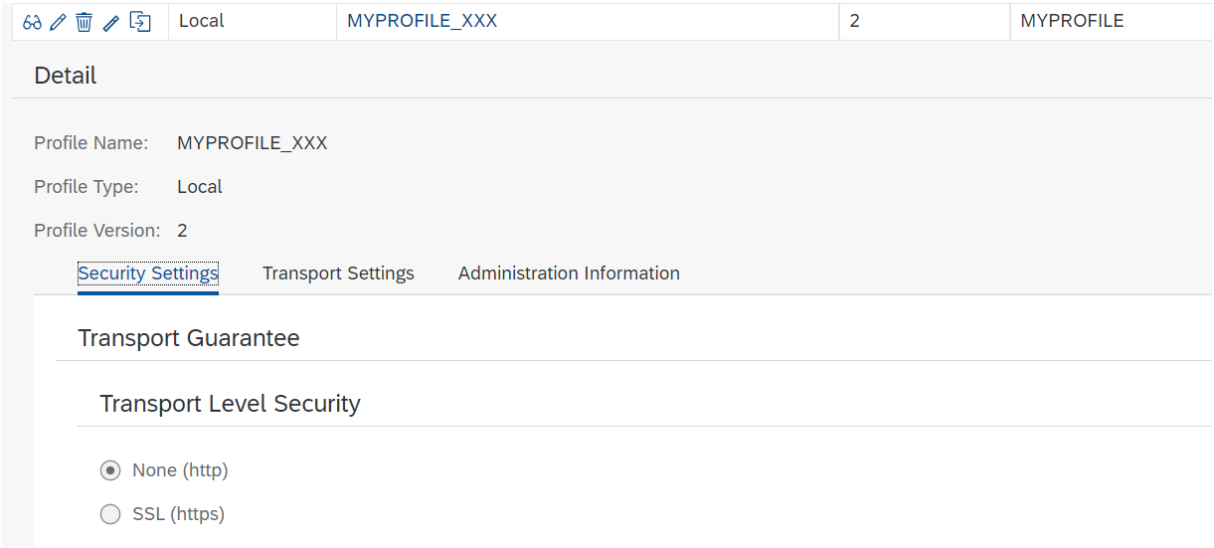
- RDM for MDG (sender): System IH1, client 100
- SAP Business System (receiver): System IR1, client 200

Please adjust the examples below to your own system landscape.

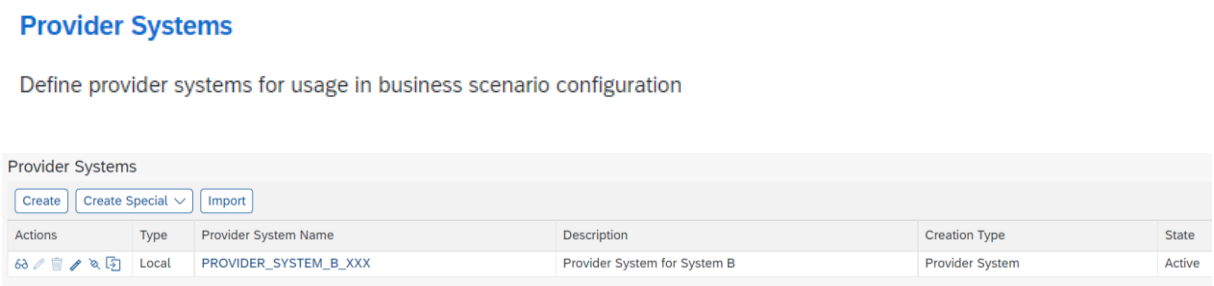
5.2.1 Receiver: Technical Administration: Profiles and Provider Systems

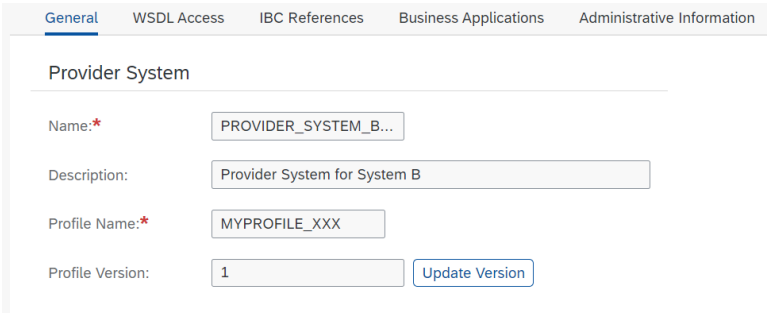
System IR1 200: transaction SOAMANAGER:

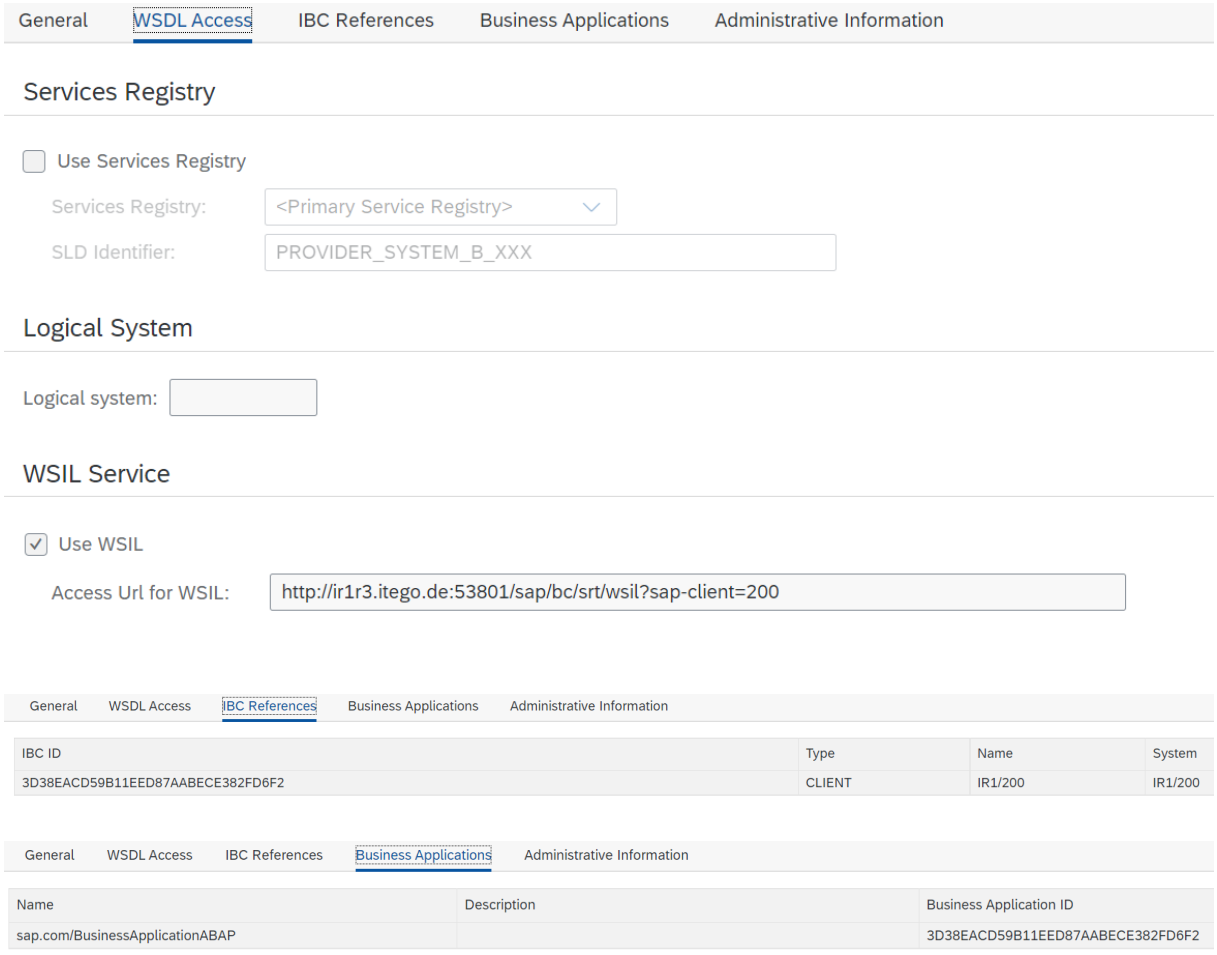




Or https based on your security requirements.

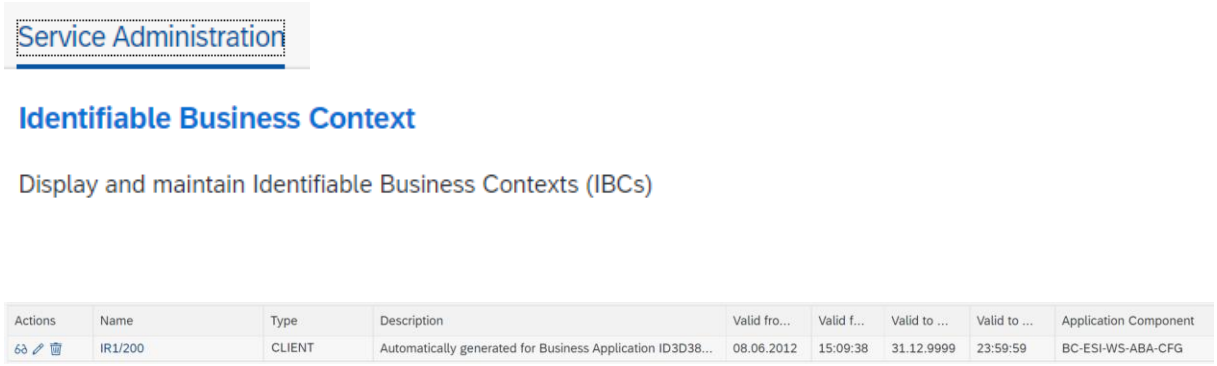






5.2.2 Receiver: Service Administration: Business Context

System IR1 200: transaction SOAMANAGER:



Identifiable Business Context Reference

Display and maintain Identifiable Business Contexts references (IBC reference)

Actions	Name	Type	Type Description	System	Description	Application ...	Is Assign...
	IR1/200	CLIENT	Client	IR1/200	Automatically generated for...	BC-ESI-WS-AB...	☑

5.2.3 Receiver: Service Administration: Local Integration Scenario

System IR1 200: transaction SOAMANAGER:

Service Administration

Local Integration Scenario Configuration

Configure multiple service definitions and service groups supporting change management

	Local	MYSCENARIO_B_XXX	Scenario B	Active

Detail

Scenario Name: MYSCENARIO_B_XXX

Scenario Type: Local

Service Definitions | Service Groups | Administrative Information

Internal Name	External Name	External Namespace	Description	Assigned Profiles	Is Configured
/ITR/RDM_WS_SEND_OBJECT	/ITR/RDM_WS_SEND_OBJECT	urn:sap-com:document:sap:soap...	No short text found	MYPROFILE_XXX	☑

5.2.4 Receiver: Services Registry: Published Systems and Objects

System IR1 200: transaction SOAMANAGER:

Services Registry

Publishing Systems

Display and maintain Publishing Systems in Services Registry

Actions	Name	Name and Client	Logical Key	Host Name	Publishing System Type	Application	Publ.	Orig.
	IR1	IR1(200)	200.SystemName.IR1.SystemNumber.0090257000.SystemHome.ir1r3#ABAP	ir1r3	ABAP	☑	☑	☑

Published Service Definitions

Display and maintain published Service Definitions in Services Registry

Actions	Internal Name	External Namespace	External Name	State	Description	Publishing System
	/ITR/RDM_WS_SEND_OBJECT	urn:sap-com:document:sap:soap:functions:mc-style	/ITR/RDM_WS_SEND_OBJECT	Configured	No short text found	IR1(200) on ir1r3

Published Bindings

Display and maintain published Bindings in Services Registry

Actions	Binding Name	Service Namespace	Internal Service Name	External Service Name	Publishing Sys
	BINDING__-ITR_-RDM_WS_SEND_OBJECT_MYPROFILE_XXX_L	urn:sap-com:document:sap:soap:functions:mc-style	/ITR/RDM_WS_SEND_OBJECT	/ITR/RDM_WS_SEND_OBJECT	IR1(200) on ir1
	BINDING__-ITR_-RDM_WS_SEND_OBJECT_MYPROFILE_XXX_L	urn:sap-com:document:sap:soap:functions:mc-style	/ITR/RDM_WS_SEND_OBJECT	/ITR/RDM_WS_SEND_OBJECT	IR1(200) on ir1
	Z_IR1200_WS_SEND_OBJECT	urn:sap-com:document:sap:soap:functions:mc-style	/ITR/RDM_WS_SEND_OBJECT	/ITR/RDM_WS_SEND_OBJECT	IR1(200) on ir1

Details of Service Definition: /ITR/RDM_WS_SEND_OBJECT

Overview | Configurations | Classifications | Details

Define Services and Bindings

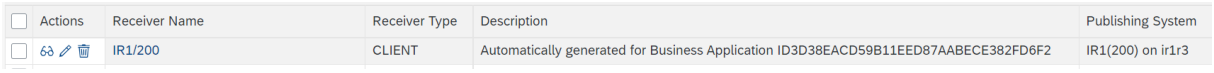
Create Service | Activate | Deactivate | Delete | Republish | Display as List

Service/Binding	Actions	State	Description
B330F036C1921EDD87C2B0AEB7C14D2B		Active	
BINDING__-ITR_-RDM_WS_SEND_OBJECT_MYPROFILE_XXX_L			
B330F036C1921EED87C87B02F24F87E6		Active	
BINDING__-ITR_-RDM_WS_SEND_OBJECT_MYPROFILE_XXX_L			
Z_IR1200_WS_SEND_OBJECT		Active	Z_IR1200_WS_SEND_OBJECT
Z_IR1200_WS_SEND_OBJECT			

Export WSDL to set up sender system in a later step.

Published IBCs

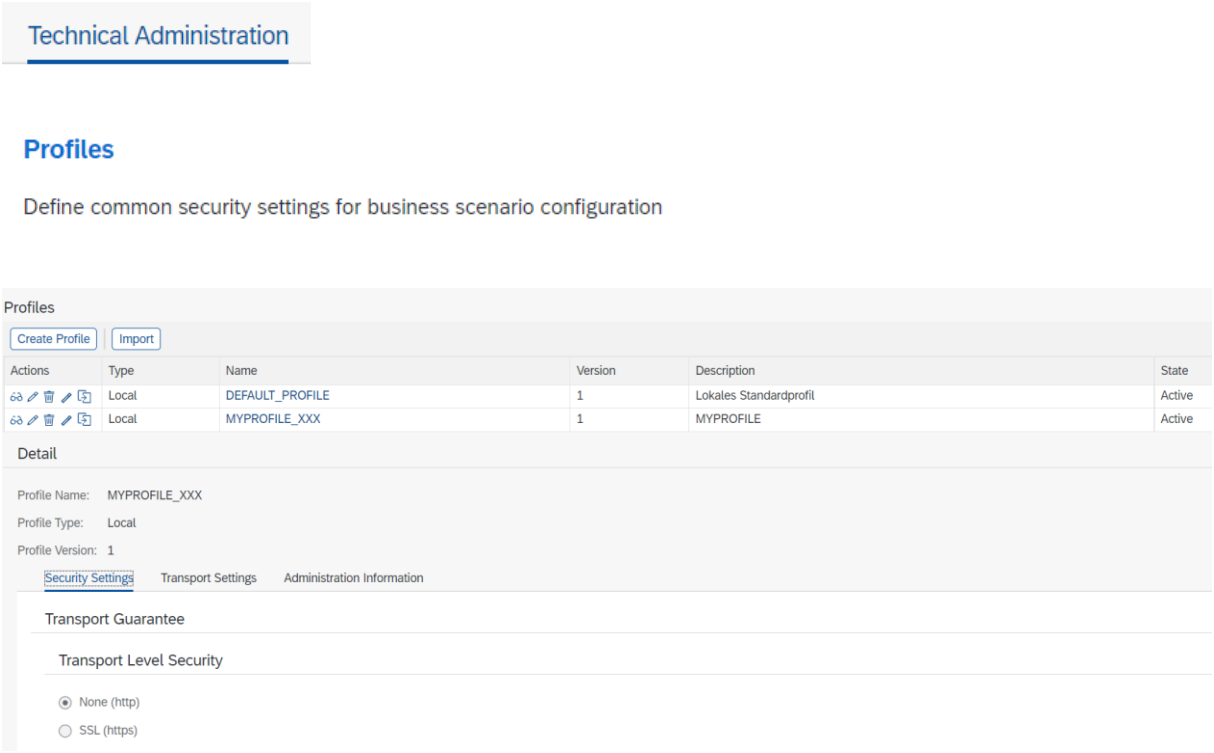
Display and maintain published Identifiable Business Contexts (IBCs) in Services Registry



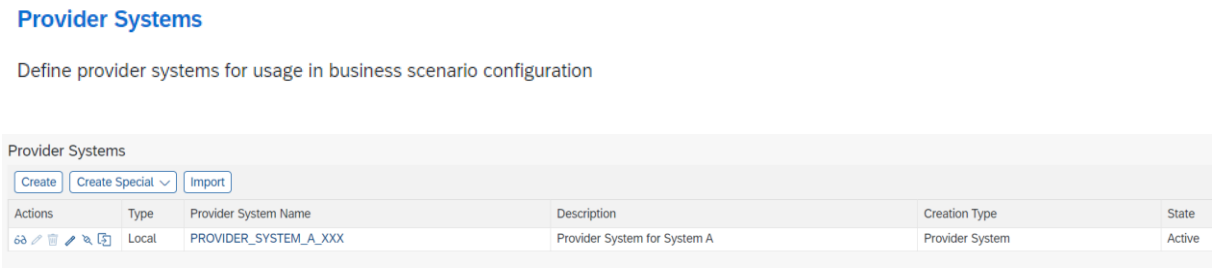
Actions	Receiver Name	Receiver Type	Description	Publishing System
	IR1/200	CLIENT	Automatically generated for Business Application ID3D38EACD59B11EED87AABECE382FD6F2	IR1(200) on ir1r3

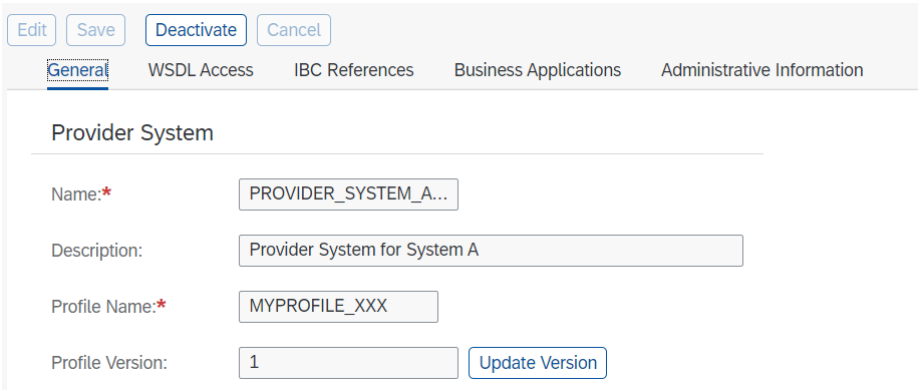
5.2.5 Sender: Technical Administration: Profiles and Provider Systems

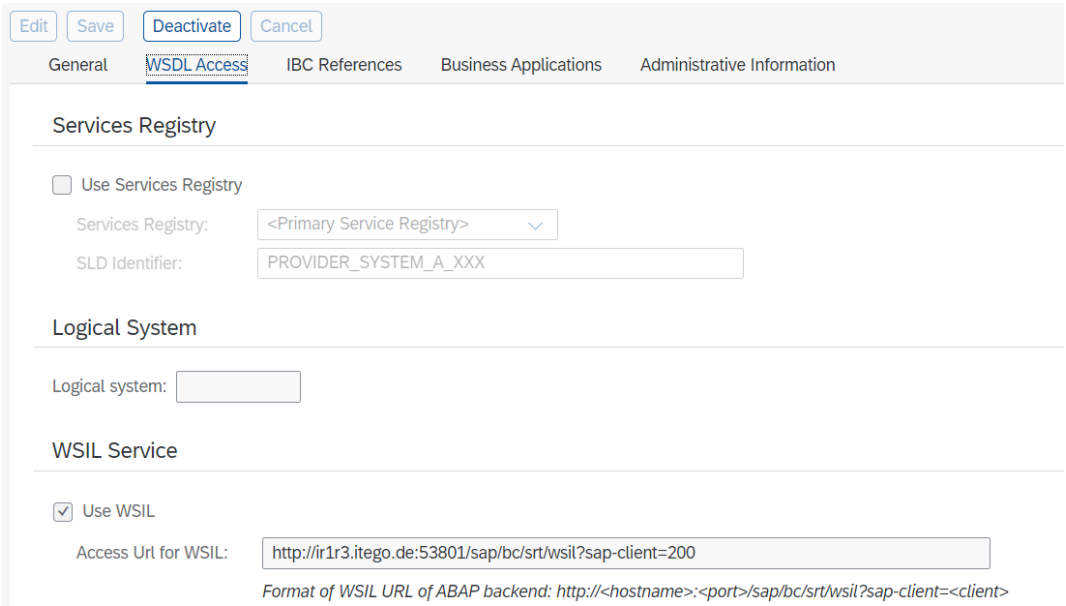
System IH1 100: transaction SOAMANAGER:

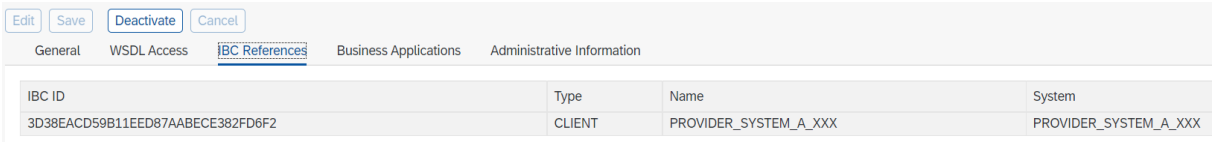


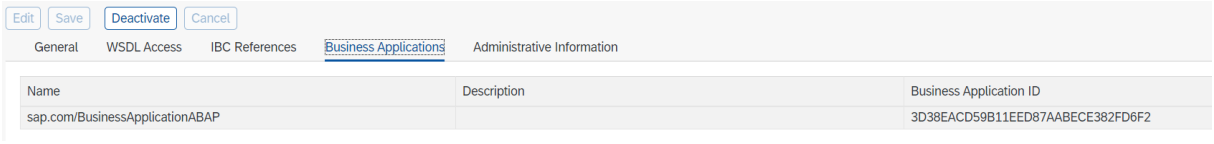
Or https based on your security requirements.





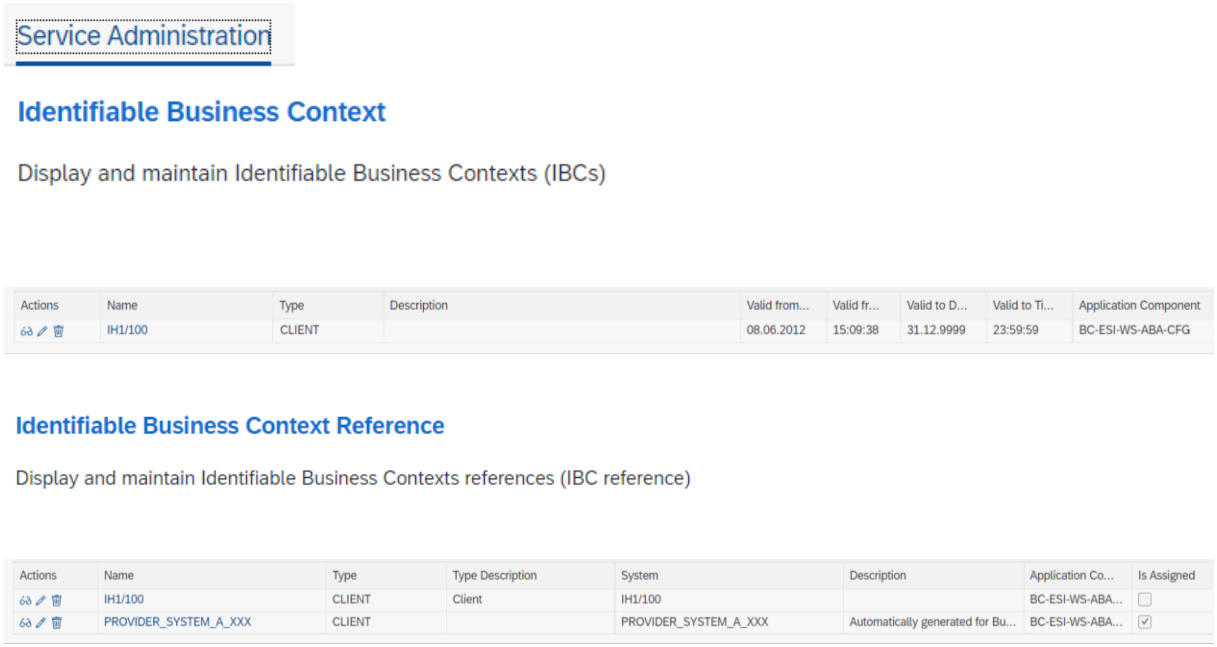






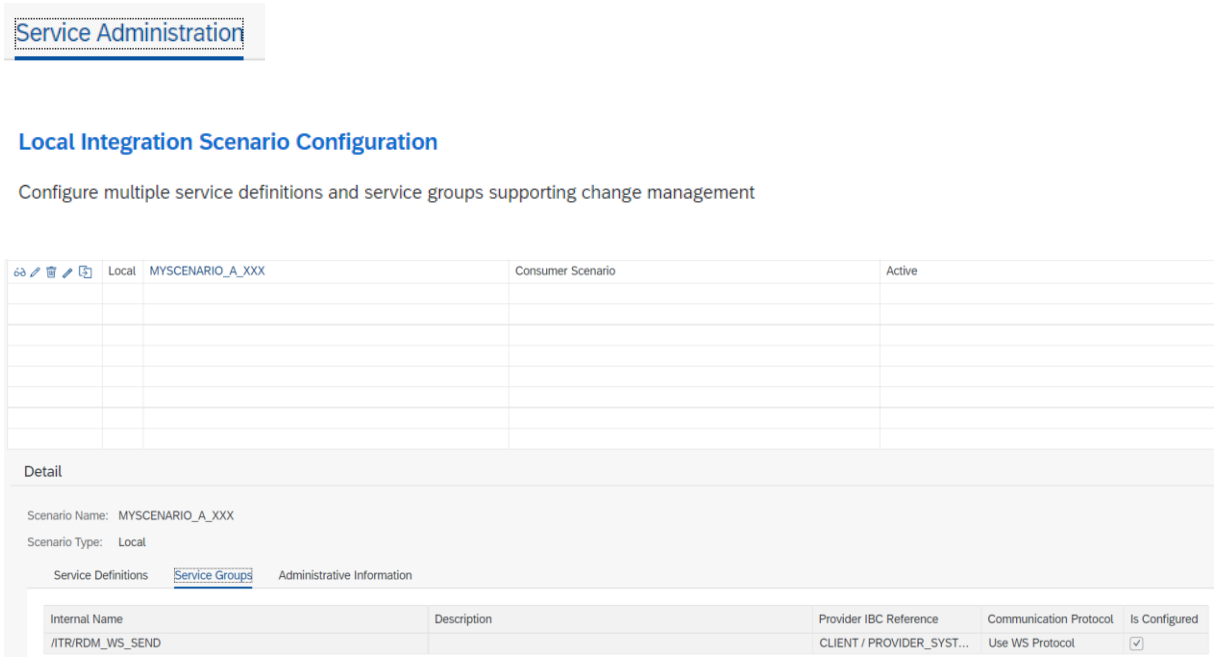
5.2.6 Sender: Service Administration: Business Context

System IH1 100: transaction SOAMANAGER:



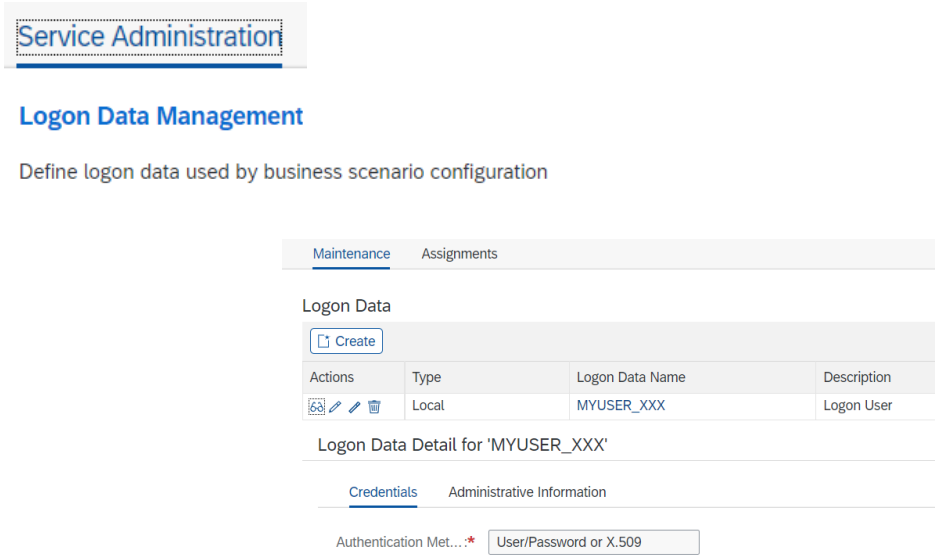
5.2.7 Sender: Service Administration: Local Integration Scenario

System IH1 100: transaction SOAMANAGER:

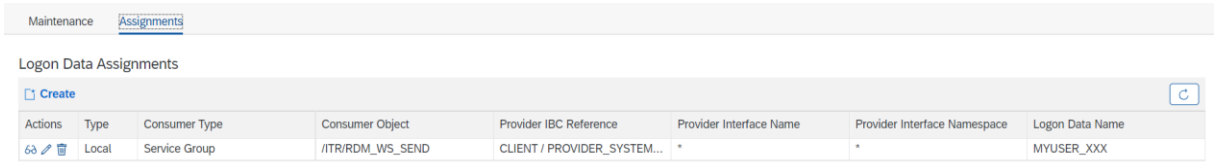


5.2.8 Sender: Service Administration: Logon Data

System IH1 100: transaction SOAMANAGER:



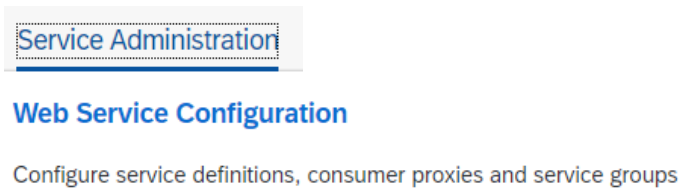
Provide User and Password

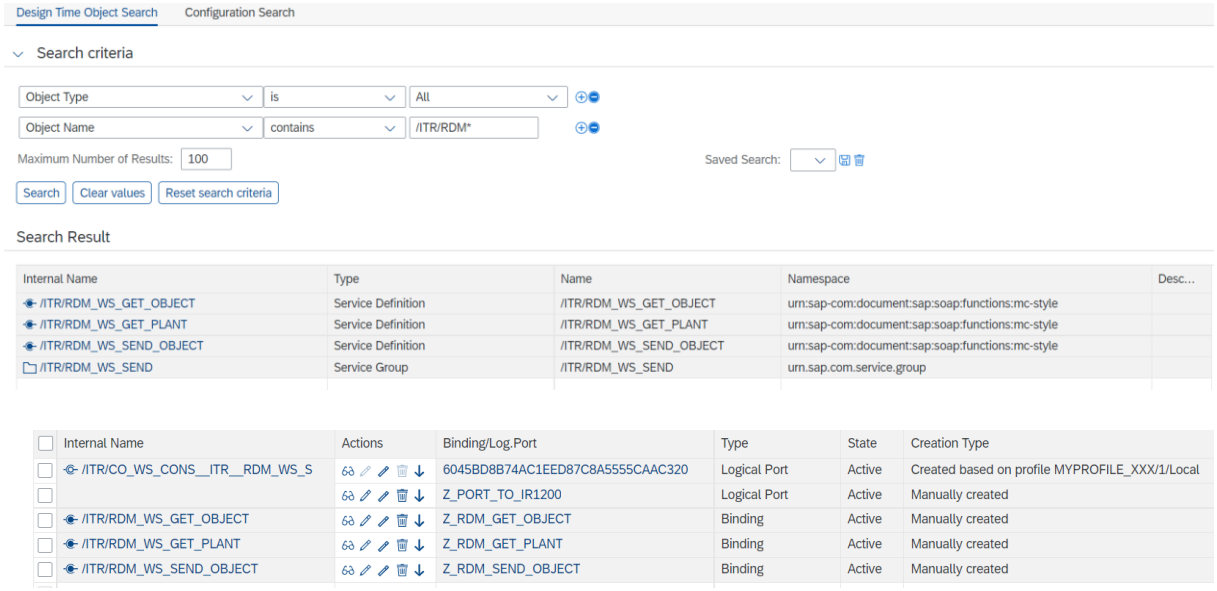


Assignment to Service Group

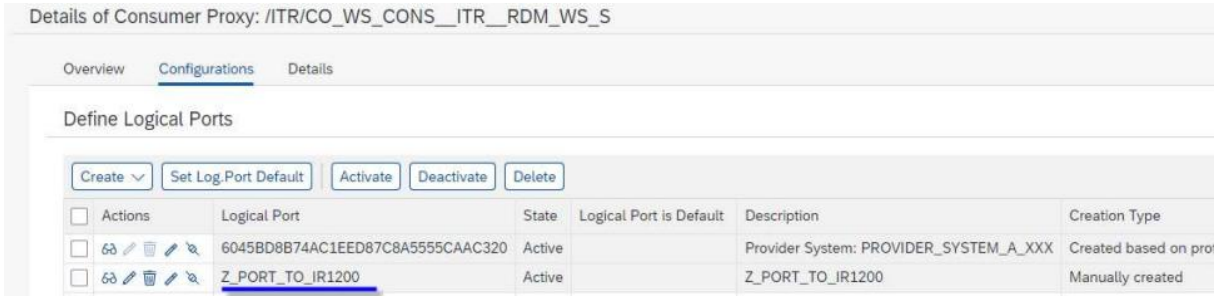
5.2.9 Sender: Service Administration: Web Service

System IH1 100: transaction SOAMANAGER:



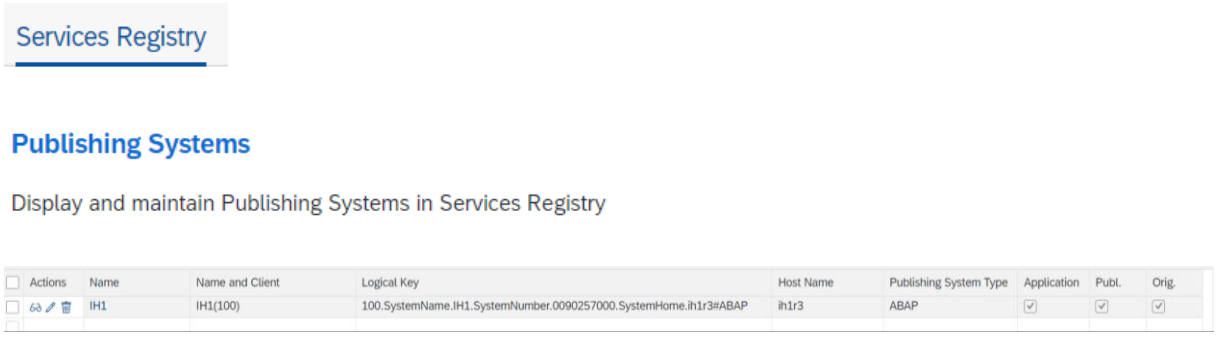


Logical port created based on exported WSDL from receiver system.



5.2.10 Sender: Services Registry: Published Systems and Objects

System IH1 100: transaction SOAMANAGER:



Published Service Definitions

Display and maintain published Service Definitions in Services Registry

Actions	Internal Name	External Namespace	External Name	State	Description	Publishing System
	/ITR/RDM_WS_GET_OBJECT	urn:sap-com:document:sap:soap:functions:mc-style	/ITR/RDM_WS_GET_OBJECT	Configured		IH1(100) on ih1r3
	/ITR/RDM_WS_GET_PLANT	urn:sap-com:document:sap:soap:functions:mc-style	/ITR/RDM_WS_GET_PLANT	Configured		IH1(100) on ih1r3
	/ITR/RDM_WS_SEND_OBJECT	urn:sap-com:document:sap:soap:functions:mc-style	/ITR/RDM_WS_SEND_OBJECT	Configured		IH1(100) on ih1r3

Published Bindings

Display and maintain published Bindings in Services Registry

Actions	Binding Name	Service Namespace	Internal Service Name	External Service Name	Publishing System
	Z_RDM_GET_OBJECT	urn:sap-com:document:sap:soap:functions:mc-style	/ITR/RDM_WS_GET_OBJECT	/ITR/RDM_WS_GET_OBJECT	IH1(100) on ih1r3
	Z_RDM_GET_PLANT	urn:sap-com:document:sap:soap:functions:mc-style	/ITR/RDM_WS_GET_PLANT	/ITR/RDM_WS_GET_PLANT	IH1(100) on ih1r3
	Z_RDM_SEND_OBJECT	urn:sap-com:document:sap:soap:functions:mc-style	/ITR/RDM_WS_SEND_OBJECT	/ITR/RDM_WS_SEND_OBJECT	IH1(100) on ih1r3

Published IBCs

Display and maintain published Identifiable Business Contexts (IBCs) in Services Registry

Actions	Receiver Name	Receiver Type	Description	Publishing System
	IH1/100	CLIENT		IH1(100) on ih1r3