How to... Master Data Governance for Material:
Enterprise Material Services Enhancement Guide

Applicable Releases:
From MDG 6.1

Version 2
September 2018
Document History

<table>
<thead>
<tr>
<th>Document Version</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.00</td>
<td>First official release of this guide</td>
</tr>
<tr>
<td>2.00</td>
<td>Update chapter 3</td>
</tr>
</tbody>
</table>
# TABLE OF CONTENTS

1. BUSINESS SCENARIO ........................................................................................................................................... 4
2. GENERAL INFORMATION ........................................................................................................................................ 4
3. BACKGROUND INFORMATION .......................................................................................................................... 4
4. SCENARIOS FOR MATERIAL SERVICE ENHANCEMENTS .................................................................................. 5
   4.1. Scenario 1: Extend with Existing Reuse Attributes ......................................................................................... 7
       4.1.1. Create a Product, Product Version, Software Unit, SWC, and EnSWCV in SLD ........................................... 7
       4.1.2. Backend Table/Structure Level Changes .................................................................................................... 14
       4.1.3. ESR/Proxy Level Changes .......................................................................................................................... 14
       4.1.4. Data Model Enhancements (If Applicable) ................................................................................................... 19
       4.1.5. BAdI Implementation .................................................................................................................................. 19
   4.2. Scenario 2: Extend with Customer Reuse Attributes ....................................................................................... 20
       4.2.1. Creation of Product, Product Version, Software Unit, SWC, and EnSWCV in SLD ................................. 20
       4.2.2. Backend Table/Structure Level Changes .................................................................................................... 20
       4.2.3. ESR/Proxy Level Changes .......................................................................................................................... 20
       4.2.4. Data Model Enhancements ........................................................................................................................ 20
       4.2.5. BAdI Implementation .................................................................................................................................. 20
   4.3. Scenario 3: Extend with Existing Reuse Entity ................................................................................................. 20
       4.3.1. Create product, product version, software unit, SWC, and EnSWCV in SLD .............................................. 20
       4.3.2. Backend Table/Structure Level Changes .................................................................................................... 20
       4.3.3. ESR/Proxy Level Changes .......................................................................................................................... 20
       4.3.4. Data Model Enhancement .......................................................................................................................... 20
       4.3.5. BAdI Implementation .................................................................................................................................. 21
   4.4. Scenario 4: Extend with Customer Reuse Entity ............................................................................................... 21
       4.4.1. Creation of product, product version, software unit, SWC, and EnSWCV in SLD ................................. 21
       4.4.2. Backend table/Structure Level Changes .................................................................................................... 21
       4.4.3. ESR/Proxy level changes ............................................................................................................................ 21
       4.4.4. Data model enhancement ........................................................................................................................... 21
       4.4.5. BADI implementation .................................................................................................................................. 21
1. BUSINESS SCENARIO
SAP Master Data Governance for Material (MDG-M) provides business processes to find, create and change material master data, and to mark it for deletion. It supports the governance of material master data on a central hub and the distribution of material master data to connected operational and business intelligence systems. The processes are workflow-driven and can include several approval and revision phases, including collaboration between all users participating in master data maintenance.

Enterprise Services allow you to leverage SAP solutions to include a wide range of composite applications that are provided by customers and partners to build new, flexible, and innovative solutions. Enterprise Services are highly-integrated Web services combined with business logic and semantics that can be accessed and used repeatedly to support a particular business process.

This guide explains how to enhance asynchronous SAP Material Enterprise Services to accommodate your business requirements.

2. GENERAL INFORMATION
Note: The screenshots in this guide are from the SAP NetWeaver PI 7.0 platform and may differ from the SAP NetWeaver PI 7.1 screens.

You define a service operation enhancement in the Enterprise Service Repository (ESR). Based on this, you generate an enhancement proxy and implement the required business logic at your back-end system. You implement the business logic in an Enterprise Service-specific Business Add-In (BAdI). Advanced knowledge of SAP XI 3.0 is necessary to understand this document. Note that you can use the ESR which resides within SAP NetWeaver Process Integration (SAP NetWeaver PI 7.0) or within SAP NetWeaver Composite Environment (SAP NetWeaver CE 7.1). This document was written using an ESR of SAP NetWeaver PI 7.0.

3. BACKGROUND INFORMATION
From MDG6.1 on, the Inbound Service for Material Replication and outbound confirmation is delivered. The Inbound Service is capable to store Materials with options to Active Area, to Staging or to Active Area and in case of errors to Staging. In all cases the service considers key and value mapping definitions. Outbound Service and Inbound Confirmation are not available.

Note: Outbound Service and Inbound Confirmation are not available.
Note: The SAP Online Help for configuration of the SOA Manager can be found here: https://help.sap.com/viewer/5df72ff3c1664986b19a10e3721fe297/MDG92_SP00/en-US/e4d13754377d024be10000000a44176d.html
4. SCENARIOS FOR MATERIAL SERVICE ENHANCEMENTS

This document illustrates how to enhance the asynchronous material services MaterialERPBulkReplicateRequest_In with relevance to the scenarios mentioned below.

For general information about ESOA Service extensibility refer to the below link.

http://scn.sap.com/docs/DOC-18402

This service can be used to write data to Either Active Area or Staging Area depending upon Sender System configuration in the View Cluster MDGVC_BUS_SYS.
There are four possible scenarios for enhancing a material service explained in this document. The scenarios are:

**Scenario 1:** Extend with Existing Reuse Attributes (Example: fields FORMT, ZEIVO in MARA table but not in the data model MM or service definition)

**Scenario 2:** Extend with Customer Attributes from Reuse (Example: A customer enhances MARA with own attributes using APPEND structure say new field ZZLOB as mentioned in figure 1 above)

**Scenario 3:** Extend with an Existing Reuse Entity (Example: Storage Location table)

**Scenario 4:** Extend with Reuse Customer Entity (Example: ZMAT_BUPA table for material nickname)

Each of the above four scenarios have five main steps with relevance to active area and staging enhancements.

The main steps in each scenario are:

- Create a product, product version, Software unit, SWC (Software Component) & EnSWCV (Enhancement Software component version) in SLD.
- Backend table/structure level changes if any.
- ESR/Proxy level changes.
- Data modeling changes (Entity type, Attribute, Structure generation) if applicable.
- BAdI method implementation.

Below are detailed descriptions of each scenario and its associated steps.
4.1. Scenario 1: Extend with Existing Reuse Attributes

In this scenario, fields are available in the SAP standard tables but these fields are not modeled either in the data model definition or proxy interface definition. In the figure 1, the fields FORMAT and ZEIVO are part of the standard MARA table and are not modeled in the data model definition/proxy interface definition. The following steps outline how to extend these fields.

4.1.1. Create a Product, Product Version, Software Unit, SWC, and EnSWCV in SLD

Create a Product and a Product Version

Create a non-SAP product with a name that describes the enhancement. Follow the steps below to create a new product and product version.


2. Under Software Catalog choose Products.


5. Choose Create. The product and new Product version are created in SLD.

*Create a Software Unit*

Enter the Name for the new Software unit and choose Create. The new Software Unit is created in SLD.

*Create SWC and EnSWCV in SLD*

If a product exists, you can create a SWC and an EnSWCV during the maintenance of the product, or independently during the maintenance of the SWC. The procedure for creation during the maintenance of products is described here.

1. Create a non-SAP product and its version as shown in Step 1. Under Software Catalog choose Software Components.

3. Choose the non-SAP product version and software unit for which the EnSWCV must be created. For information on creating a non-SAP product version and software unit, see Step 1 and Step 2.

4. Enter the Vendor, Name, and Version of the Product. Note: To import the EnSWCV into the ESR immediately after its creation in the SLD, use the production state Released.

5. Choose Create. The new EnSWCV is created.

*Define Dependencies between SWCV and EnSWCV in SLD*

SAP NetWeaver PI allows you to create dependencies between one or more components. If you enhance an interface object, you must create a “based-on” dependency between the EnSWCV and the underlying SWCV, because one component can be a prerequisite for another component and some objects from the underlying SWCV can be reused in the original SWCV. For example, SWCV MDG APPLICATIONS 606 is the underlying component version for MDG APPLICATIONS 607 because it uses some MDG Applications 606 classes, functions, and data types.
The process described above is described in the following figure:

If dependencies exist in the ESR, you see an additional component, *Basis Objects*, in the navigation tree for the based-on SWCV. These objects contain all the namespaces from the underlying SWCV.

Once you create a dependency between the EnSWCV (as based-on SWCV) and the SWCV from which you need to enhance the service operation (underlying SWCV), you can reference objects from the based-on SWCV and also from underlying SWCV. This allows you to use the enhanced objects from namespaces of the underlying SWCV.

The dependency between the EnSWCV and underlying SWCV described above is illustrated in the following figure:

To define the dependencies between an EnSWCV and an underlying SWCV:

2. Choose the EnSWCV that you created in the previous section.
3. On the dependencies tab, choose Installation Time in the dropdown menu for context.


5. In the list that appears, choose the SWCV for which you want to define the enhancement.
6. In the Filter field enter the name of the SWC and choose Go to get to the underlying SWCV.

7. Once the system retrieves your value, select it, and choose Define Prerequisite Software Component.
8. The system displays the dependencies between the prerequisite SWCV (that contains the services to be enhanced) and the dependent EnSWCV.

4.1.2. **Backend Table/Structure Level Changes**
Since the fields FORMAT and ZEIVO are already part of the standard table definition of MARA, no further changes are needed in this step for this scenario.

4.1.3. **ESR/Proxy Level Changes**
In this step, the fields FORMAT and ZEIVO are added to the proxy definition. The following sub-steps explain how this is achieved using enhancement data type in SWCV.

1. Choose the Integration Repository in the XI system.

2. Choose the objects tab.

4. Select the EnSWCV from the list and choose Import.

5. The system displays an Import Protocol after it successfully imports the SWCVs.

6. Choose the Switch between Display and Edit Modes button to go to the change mode in the EnSWCV editor.

7. Insert the new namespace and save it (in the example, http://xiTestComponent/xi/mdg is the new namespace)

8. In the Change Lists tab right-click on the namespace and choose Activate.

9. Choose the Details tab to see the EnSWCV that depends on the underlying SWCV.
Now you can use the namespace to create enhancements for your enterprise service.

11. Enter the Name and Description of the Data type Enhancement and choose Create.

12. On the Enhancement Definition tab, choose the Data-type to be enhanced.
13. In the popup, select the data type to be enhanced (Example: MatlERPRplctReqCom) and select OK. The selected data type will then be entered into the field Name as shown below.

![Data Type Enhancement Popup](image)

14. Add the new field ZFORMT to the enhancement structure. Choose Save.

15. Activate the enhancement structure.

![Activate Change List](image)
16. Release the transport.

Regenerate the proxy using the sproxy transaction in the backend system after transports reach the system. The ABAP system generates an append structure for the service node that is to be enhanced. The system automatically adds the fields from the enhancement to the Web Services Description Language (WSDL) as optional fields.

### 4.1.4. Data Model Enhancements (If Applicable)

Refer to the MDG-M extensibility guide link below for more information on the extending data model. [http://scn.sap.com/docs/DOC-29042](http://scn.sap.com/docs/DOC-29042)

### 4.1.5. BAdI implementation

The BAdI shown below enables you to extend the mapping and the service functions.

Refer to example BAdI implementation: `MDG_SE_MAT_RPLCTROQ_IN`, which is available in MDG7. In MDG6.1, SAP notes 1808293 and 1806531 are required.

```plain
CL_MDG_BS_MAT_DS_REQ_IN_EXT =>
IF_MDG_SE_MAT_ERPBULKRPLCTRQ~INBOUND_PROCESSING
IF_MDG_SE_MAT_ERPBULKRPLCTRQ~DO_POST_APPLICATION_PROCESSING
```

The above methods contain different code segments for different scenarios.
4.2. Scenario 2: Extend with Customer Reuse Attributes

In this scenario, customer fields are added to the SAP standard tables along with modeling of these fields both in MDG data model definition and Service proxy interface definition. In figure 1, the field ZZLOB is added to the MARA table along with addition to the proxy and data model definition. The 5 main steps for enhancement are described in scenario 1 above. See below for scenario-specific changes to the steps.

4.2.1. Creation of Product, Product Version, Software Unit, SWC, and EnSWCV in SLD

This step needs to be implemented only once for each system. If this step is already completed as part of any other scenarios it should not be repeated. If this is not implemented, follow the steps under the scenario Extend with Existing Reuse Attributes.

4.2.2. Backend Table/Structure Level Changes

Refer to the extensibility guide for MDGM link below for more information on backend table/structure enhancements.

http://scn.sap.com/docs/DOC-47988

4.2.3. ESR/Proxy Level Changes

In this step, the new field in customer namespace needs to be added in the proxy structure definition. This process is similar to the steps followed in Scenario 1 under section 4.1.3.

Ignore the steps from 1 to 9 of Scenario 1 under section 4.1.3. if they are already performed and follow the steps from 10 onwards till Step 13.

At step 14, add field zzlob to the enhancement structure similar to addition of zformat field in Scenario 1 and follow the steps till the end of section 4.1.3. for Scenario 1.

4.2.4. Data Model Enhancements

http://scn.sap.com/docs/DOC-47988

4.2.5. BAdI Implementation

Refer to section 4.1.5 of Scenario 1.

4.3. Scenario 3: Extend with Existing Reuse Entity

In this scenario, a new entity will be created for standard SAP tables that are not part of the data model or proxy interface definition.

In the figure 1, the fields XCHAR and DISL are part of the standard MARC table but assuming that they are not used in the data model definition and a new entity is created for this. The following steps describe how this can be achieved with the standard 5 main steps.

4.3.1. Create product, product version, software unit, SWC, and EnSWCV in SLD

This step needs to be implemented only once per system. If this step is already completed as part of any other scenarios it should not be repeated. If this is not implemented, follow the steps that are mentioned under section (1) of the scenario Extend with Existing Reuse Attributes.

4.3.2. Backend Table/Structure Level Changes

This step is not relevant as the fields already exist in the standard SAP tables.

4.3.3. ESR/Proxy Level Changes

In this step, the new node in customer namespace needs to be added in the proxy structure definition. This process is similar to the steps followed in Scenario 1 under section 4.1.3.

Ignore the steps from 1 to 9 of Scenario 1 under section 4.1.3. as they need not be repeated again if these steps are already performed and follow the steps from 10 onwards.

At step 13, choose the data type where the new node needs to be added. Add the field’s ZDISL and ZXCHAR to the new node and follow the steps till the end of section 4.1.3. for Scenario 1.

4.3.4. Data Model Enhancement

Refer to the MDG-M extensibility guide link below for more information on the extending data model.

http://scn.sap.com/docs/DOC-31869
4.3.5. **BADI Implementation**  
Refer to section 4.1.5 of Scenario 1.

4.4. **Scenario 4: Extend with Customer Reuse Entity**  
In this scenario, a new customer-defined database table is defined along with the new definition of these details in the data model definition and the proxy interface definition.  
In figure 1, this scenario is mentioned where there is a customer defined table ZMAT_BUPA used to maintain nicknames for materials.  
The table consists of fields MATNR (that refers to MATNR of MARA) and BUPA_ID (a unique Business Partner ID) as key fields, and a non-key field NICKNAME that has the nickname of the material. There is corresponding enhancement to the proxy interface definition with this data and data model definition.  
The 5 main steps for enhancement are described in scenario 1 above. See below for scenario-specific changes to the steps.

4.4.1. **Creation of product, product version, software unit, SWC, and EnSWCV in SLD**  
This step needs to be implemented only once per system. If this step is already completed as part of any other scenarios, this step shouldn’t be repeated again. If this is not implemented, follow the steps that are mentioned under Step (1) of scenario Extend with Existing Reuse Attributes.

4.4.2. **Backend table/Structure Level Changes**  
In this step, a new database table, ZMAT_BUPA, with fields MATNR, BUPA_ID, and NICKNAME is created.  
Refer to the MDG-M extensibility guide link below for more information on the extending backend table/structure.  
http://scn.sap.com/docs/DOC-27859

4.4.3. **ESR/Proxy level changes**  
In this step, the new node in customer namespace needs to be added in the proxy structure definition. This process is similar to the steps followed in scenario 1 under section 4.1.3. Ignore the steps from 1 to 8 of scenario 1 under section 4.1.3 as they need not be repeated again if these steps are already performed and follow the steps from 9 onwards.

At step 13 choose the data type where the new node needs to be added. Add field’s zcost and zbrand to the new node and follow the steps till the end of section 4.1.3 for scenario 1.

4.4.4. **Data model enhancement**  
http://scn.sap.com/docs/DOC-27859

4.4.5. **BADI implementation**  
Refer to section 4.1.5 of Scenario 1.