SAP How-To Guide: Extend MDG-M Data Model by a New Entity Type (Reuse Option)

Applies to:
SAP MDG-M running from SAP ECC 6 EhP 6 Master Data Governance. For more information, visit the Master Data Governance homepage. (http://scn.sap.com/community/mdm/master-data-governance)

Summary
SAP Master Data Governance provides an out-of-the-box solution for the central management of various master data objects such as financial objects, supplier and material. But SAP Master Data Governance also provides the flexibility to customize the solution, in cases where the pre-delivered content does not fully match customer requirements. You can use this guide to extend the MDG-M data model by a new entity. The attribute values of the new entity type will be copied to the corresponding customer tables (reuse option) after activation of the change request.

Created on: April 2014
Version: 2.0
Table of Contents

Introduction ......................................................................................................................... 3
Prerequisites ......................................................................................................................... 3
Scenario ................................................................................................................................. 3
  High Level Requirements .................................................................................................. 3
  Governance Process .......................................................................................................... 4
Introduction ......................................................................................................................... 4
  Data Modeling .................................................................................................................. 4
    Basic Data Model for Material ....................................................................................... 4
    Data Modeling Concepts in MDG .................................................................................... 5
User Interface Configuration ............................................................................................... 11
  Adaptation Options in Floorplan Manager .................................................................. 12
  Handling Previous Enhancements Using the “UI BAdI” ................................................. 12
  Removing Customizing or Personalization ...................................................................... 13
Implementation .................................................................................................................... 14
  Create DDIC Objects ....................................................................................................... 14
MDG Data Model Extension ............................................................................................... 19
  Extend MDG BS MAT (Material Master Data Structure) .............................................. 20
  Extend MDG Data Model ................................................................................................. 23
  Generate Model-Specific Structures ............................................................................. 30
Create SMT-Mapping .......................................................................................................... 32
  Create Mapping Entries in Customizing ......................................................................... 32
  Map the Active Area to the Staging Area ....................................................................... 34
  Map the Staging Area to the Active Area ....................................................................... 37
Adjust Staging Area of Linked Change Requests ............................................................. 41
BAdI: Extension of the API with Customer-Specific Segments ....................................... 42
  Creating a Selection-Range Table .................................................................................. 44
  BAdI Implementation: READ Method ............................................................................ 45
Extend User Interface .......................................................................................................... 46
  Create FPM List UIBB ...................................................................................................... 46
  Add List UIBB to Material UI ......................................................................................... 54
Testing Your Data Model Extension ................................................................................... 59
Related Content .................................................................................................................. 60
Copyright ............................................................................................................................. 61
Introduction

SAP Master Data Governance (MDG) is used for embedded MDM, that is, out-of-the-box, domain-specific master data governance to centrally create, change and distribute master data with a focus on SAP Business Suite.

Domain specific content (data models, user interfaces, workflows) is provided as part of the standard for several application areas. It is a common requirement from customers to adapt the MDG data models to their specific needs.

This guide explains how to extend the MDG-M data model by a new entity type. The attribute values of the new entity type will be copied to the corresponding customer tables (reuse option) after activation of the change request.

Prerequisites

You must have access to an ERP 6 EhP6 system with the MDG-M business functions enabled. You need access to the system used for cross-client maintenance. You must have basic understanding of the following areas:

- SAP ERP
- SAP MDG
- SAP Floorplan Manager (FPM)

Scenario

High Level Requirements

The business requires the new entity type *Business Partner Details* to save Business Partner Nicknames for materials as part of the MDG-M Material data model.

![Data Model Diagram](image-url)

*Figure: Data Model – Material (Scope of 2011 Delivery) with custom entity type “Business Partner Details”*
Governance Process

The default governance process delivered with MDG will be used. No changes to the governance process are necessary as part of this scenario.

Figure: Material Processing in Master Data Governance

Introduction

The implementation steps in this document are easier to understand if you are familiar with the basic concepts; in particular data modeling and user interface customization. In this introduction section, you can learn more about these concepts. Alternatively you can skip this section and move straight to the implementation part.

Data Modeling

In MDG, the data model is a central part of the application. SAP delivers several preconfigured data models that you can start using with little configuration.

Basic Data Model for Material

Looking at the Material object type and its related data in an abstract way, you can distinguish the following categories of data fields:

- Identifying Material Data ~ 10 fields
- Descriptive Material Data ~ 100 fields
- Process Controlling Material Data ~ 1,000 fields

MDG for material data focuses on the main identifying, descriptive attributes and process controlling data of the Material Master.

It is important to understand that MDG not only delivers the data fields in a model, but also comes with the standard business rules to check for completeness and consistency. These checks are only enforced when necessary in the process.

You can either centralize the maintenance of process controlling data on the MDG hub by using the standard backend transactions for material maintenance or you can decentralize the maintenance of process controlling data.

If customers require centralized governance of more process controlling data, they can enhance the delivered model (for example, they can add plant data or sales organization data).

The delivered standard data model is **MM**. (This model is linked to **Material** business object ID: **194**. It is also linked to the **Change Request** business object type: **BUS2550**.)

The **MM** data model for the different releases can be found here: [http://www.sdn.sap.com/irj/scn/index?rid=/library/uuid/e03cb2f0-3e03-3010-cca5-d4cd11592c28](http://www.sdn.sap.com/irj/scn/index?rid=/library/uuid/e03cb2f0-3e03-3010-cca5-d4cd11592c28)
Data Modeling Concepts in MDG

The meta-model below shows the basic elements making up a MDG data model. When you extend the data model by a new Entity Type you must also define its relationship to other data model elements and decide on a Storage and Use Type for the new Entity Type. In the following sections you will find more details regarding these topics.

Figure: The meta model is an entity-relationship-model (ERM)

The preconfigured data model for the business object types Material is MM. You can view the SAP delivered data model in Customizing for Master Data Governance under General Settings > Data Modeling > Edit Data Model. Alternatively, you can enter view cluster VC_USMD001 in transaction SM34.
### Storage and Use Types

You assign a storage and use type to specify whether and how master data can be changed in Master Data Governance. The storage and use type also indicates which database tables are generated by the system.

| Changeable via Change Request; Generated Database Tables (Type 1) | The master data of this storage and use type can be changed in Master Data Governance with a change request. The system generates all necessary database tables: check and text tables as well as additional tables, for example, for attachments and sets. The common key fields of these tables are:
| | • The entity type itself
| | • The edition – if you previously specified in the data model that the validity of master data changes is restricted to editions
| | • The entity types that are assigned to the entity type through leading relationships
| | Furthermore, all tables contain a checkbox that indicates whether the master data record is active. Depending on the workflow template used, it is possible that a master data record is not set to Active until the change request in which the record was created or changed is released.
| | The settings you make for the entity type (such as language dependency) result in additional key fields in the text table and the tables for attachments and sets.
| | The non-key fields contained in the text table are the entity texts. The non-key fields contained in the check table are the attributes of the entity type. The attachment and set tables contain predefined non-key fields. Furthermore, all database tables contain a checkbox that indicates whether the master data record was deleted. The check table also contains attributes that record which user created or changed the data records and when this was done. |
| Changeable w/o Change Request; Generated Check/Text Tables (Type 2) | The master data of this storage and use type can be changed in Master Data Governance without a change request. The system generates only the check and text tables with the entity type as well as with the entity types assigned to the entity type through leading relationships as fixed key fields. The non-key fields contained in the text table are the entity texts. The check table does not contain non-key fields. |
| Not Changeable via MDG; No Generated Tables (Type 3) | The master data of this storage and use type cannot be changed in Master Data Governance. Therefore, the system does not generate database tables. Instead, the system derives the available values from the domain that is assigned to the data element – either from the assigned value table or from the domain fixed values. |
| Changeable via Other Entity Type; Generated Database Tables (Type 4) | The master data of this storage and use type can be changed in Master Data Governance only with a change request of an entity type with storage and use type 1. The entity type needs to be in a relationship with the relationship type leading and assigned as the To-entity type to an entity type with storage and use type 1. The system generates the check table as described for storage and use type 1, but also generates the entity types that are assigned through qualifying relationships as key fields. The system does not generate a text table, attachments, or sets since entity texts are not allowed for entity types with this storage and use type. |
You can view the settings for Storage and use Type for existing Entity Types in Customizing for Master Data Governance under *General Settings > Data Modeling > Edit Data Model*. You select the MM data model and double click on *Entity Types* (view cluster VC_USMD001). In the list of entity types you can double click an entity type to view its details as shown below for Entity Type *MATERIAL*.

![Figure: Selection box showing the different storage/use types in MDG](image)

**Relationship Type**

If you have defined multiple entity types, you can determine what type of relationship should link them (leading, referencing, qualifying, or foreign key relationship). For each relationship, you specify a relationship type and cardinality.

<table>
<thead>
<tr>
<th>Relationship Type</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>Referencing</td>
<td>Specifies the <em>From-Entity type</em> as an attribute of the <em>To-Entity type</em>.</td>
</tr>
<tr>
<td>Leading</td>
<td>Specifies the <em>From-Entity type</em> on a higher level than the <em>To-Entity type</em>. The <em>From-Entity type</em> is automatically taken as the key in the generated tables. A <em>Leading</em> relationship type is identical to a <em>Qualifying</em> relationship type, except when the <em>To-Entity type</em> has a <em>Storage and Use Type</em> of 4. Master data for <em>To-Entity types</em> in <em>Leading</em> relationships is processed in the context of the entity type that is assigned using the leading relationship.</td>
</tr>
<tr>
<td>Qualifying</td>
<td>Specifies the <em>From-Entity type</em> on a higher level than the <em>To-Entity type</em>. The <em>From-Entity type</em> is automatically taken as the key in the generated tables.</td>
</tr>
</tbody>
</table>

The following options are possible for the relationship between two entity types:

<table>
<thead>
<tr>
<th>Cardinality</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>1:N</td>
<td>This cardinality represents a mandatory relationship in which one or more <em>To-Entity Types</em> can be assigned to a <em>From-Entity Type</em>. This cardinality is valid for relationships with the relationship types <em>Leading, Qualifying, and Referencing</em>. In relationships with the relationship type <em>referencing</em>, the <em>From-Entity Type</em> is a required attribute of the <em>To-Entity</em></td>
</tr>
</tbody>
</table>
Type.

| 0:N | This cardinality represents an optional relationship in which any number To-Entity Types can be assigned to a From-Entity Type. |

**Note**

Which relationship types are permitted depends on the storage and use types of the entity types. For a table with detailed information refer to help.sap.com.

**Important**

The general design assumption is that there is a 1:N relationship between a database table and its entity types. This means one entity type does not bundle several database tables.

**Reuse Area versus the Flexible Option**

When you extend the SAP delivered data model by a new entity type you must decide where to store data after the activation of a change request. During the processing of the change request the system stores data in the MDG staging area. After the activation of a change request, you can move the data to tables outside of MDG or keep the data in the MDG tables.

For optimal integration into SAP Business Suite, MDG provides the following two persistence modes:

- **Generated active area (flex mode)** – Tables as defined in the MDG data model are used to store active data.
- **Reuse active area (re-use mode)** – Existing structures of applications are used. For example, MDG for material makes use of the MARA table in ECC.

![Flex Mode](image1.png) ![Re-Use Mode](image2.png)

*Figure: Flexibility Option (left) versus Reuse Option (right)*

Where the data is stored is specified by the **Reuse Area** setting on the **Data Model** or **Entity Type** level as shown in the screenshots below.

![Change View](image3.png)

*Figure: Assignment of Reuse Area for the Data Model MM*
The MDG model MM is preconfigured with one reuse area called MATERIAL. This reuse area points to the access class CL_MDG_BS_MAT_ACCESS, which can handle all fields of the pre-delivered data model and some more.

If you extend the data model by a new entity type and want the data of that entity type to remain in the MDG tables after activation you can choose MDG as a reuse area.
Input Help (Accessible Using the F4 Key)

The system applies the following rules of precedence when assigning input help:
1. Search help assignment in data model definition
2. Backend structure MDG_BS_MAT_S_MARA, MDG_BS_MAT_S_* (not existing for Flex Entities)
   a. Search help assignment in the structure
   b. Value table on domain with foreign key association
3. Search help assignment on data element (for flex entities)
4. Fixed values or value table on domain

Note: Value table on domain without foreign key assignment (for flex entities) is not supported out of the box

Code Lists

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

Data Modeling Considerations for List-UIBBs

If you want the Material UI to use two or more list UIBBs (User Interface Building Blocks), you must adjust the data model. You can implement independent list UIBBs or interdependent list UIBBs. For more information, see the table below.

<table>
<thead>
<tr>
<th>Desired System Behavior</th>
<th>Action</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Independent List UIBBs</strong></td>
<td>Changes to one list UIBB have no impact on the other list UIBB.</td>
</tr>
<tr>
<td></td>
<td>Create separate entity types and assign one to each List-UIBB.</td>
</tr>
<tr>
<td><strong>Interdependent List UIBBs</strong></td>
<td>If you create a new row for the same key in one UIBB, the system creates a new row in the other UIBB.</td>
</tr>
<tr>
<td></td>
<td>Either assign the same entity type for both List-UIBBs or implement a derivation.</td>
</tr>
</tbody>
</table>

Example Implementation of Independent and Interdependent UIBBs

You have created **interdependent** list UIBBs for the following user interfaces using the ZZMARC entity type:

- MRP1/MRP2
- Foreign Trade Export

In the Component Configuration for MRP1/MRP2 and for Foreign Trade Export, you have maintained the ZZMARC entity type.

In addition, you have created an **independent** list UIBB for Foreign Trade Export (separate) user interface, using a separate entity type ZZMARCFTE. The reason for this decision is that a particular customer wants only to enter data relevant to foreign trade export, and does not want to enter MRP data. In the Component Configuration for Foreign Trade Export (separate), you have maintained the ZZMARCFTE entity type. The ZZMARCFTE entity type is not affected when you enter a new row in ZZMARC component configurations.
User Interface Configuration

The User Interface in MDG is configured using the Floorplan Manager. The Floorplan Manager (FPM) is a Web Dynpro ABAP application that provides a framework for developing new Web Dynpro ABAP application interfaces consistent with SAP UI guidelines.

The entry point you need for starting an application is the application configuration, which is tied to a single Web Dynpro application. The necessary information needed to start the application is divided between the following two entities:

- **Web Dynpro ABAP Application**: Contains the information about the main component and window of the application
- **Web Dynpro ABAP Application Configuration**: Contains the information about the configuration used for starting the main component

There are only 3 different main components used in FPM-based applications. Each one corresponds to one of the supported floorplans:

- OIF (Object Instance Floorplan): component `FPM_OIF_COMPONENT`
- GAF (Guided Activity Floorplan): component `FPM_GAF_COMPONENT`
- OVP (Overview Page Floorplan): component `FPM_OVP_COMPONENT`

In the screenshot below application `MDG_BS_MAT_OVP` is using component `FPM_ADAPTABLE_OVP` with configuration `BS_MAT_OVP_CBA` to provide the context-based adaptation (CBA) functionality. The layout is provided by `FPM_OVP_COMPONENT` in combination with configuration `BS_MAT_OVP_LAYOUT`. 
Adaptation Options in Floorplan Manager

A Floorplan Manager UI can be adapted using different techniques. The figure below shows the relationship between configuration, customizing, and personalization. Context-Based-Adaptation is another way the user interface can be customized for specific use cases.

In the context of MDG, you typically choose to customize the SAP delivered configuration. Only if customizing is not feasible do you copy the SAP delivered UI configuration to the customer namespace and change the copy.

In the following cases the UI should be copied rather than customized:

- Code changes are required
- The UI needs to be changed for all users in the system and not only client-specific
- The changes to the UI are extensive

Note
For more details regarding options for Floorplan manager user interface adaptation, advantages, disadvantages and steps required please familiarize yourself with SAP Note 1619534.

See also How To Guide for the UI: http://scn.sap.com/docs/DOC-30192

Handling Previous Enhancements Using the “UI BAdI”

With EhP6, the Business Add-In previously available to adjust the user interface for single processing of material is no longer available. Instead, a new UI provides greater flexibility while retaining and refining the functionality provided with EhP5. Below is the full list of the functions previously available through the single processing UI by implementing the BAdI in EhP5, that are now provided under the EhP6 UI paradigm:
- Initialize the displayed data (when creating a new entity, for example)
- Restrict the values displayed in a dropdown list field or selection field group
- Restrict the values displayed in the input help
- Dynamically control the visibility of fields on the user interface and of the property that determines if fields are required or display-only
- Define navigation destinations of UI elements of the type hyperlink (or pushbutton)
- Check if the lead selection of a table may be changed

For more information, see the MDG consulting SAP Note 1606341.

Removing Customizing or Personalization

If required, a system administrator can delete customizing or personalization from a central place using the following Web-Dynpro applications, which must be used with caution:

- WD_ANALYZE_CONFIG_USER
- WD_ANALYZE_CONFIG_COMP
- WD_ANALYZE_CONFIG_APPL
Implementation

Two major building blocks make up the implementation of the entity type extension. In the first phase, you extend the data model. In the second phase, you extend the user interface to include the new entity type.

The flow diagram below shows the detailed implementation steps. We recommend you use it as an orientation. Each box in the diagram below corresponds to a section in this guide in which you find detailed execution instructions.

Create DDIC objects

Extend MDG BS MAT

BApi: Extension of the API with Customer-Specific Segments

Extend MDG Data Model

Generate MDG data model specific structures

SMT Mapping

Map staging to primary persistence

Map primary persistence to staging

Create custom List-UIBB

Copy template List-UIBB

Adapt custom List-UIBB

Add custom List-UIBB to Material UI using CBA

Test Configuration

Figure: Implementation steps for re-use Entity-Type extension

Create DDIC Objects

In this section you create the table and structure that you use to store master data after activation in MDG. After activation data will be transferred from the MDG staging area to the table that you define in the following steps.
1. In transaction SE11 create table YMDGM_BUPA00 as shown.

![Dictionary: Display Table](image1)

2. In transaction SE11 create structure YMDGM_BUPA01_S as shown.

![Dictionary: Display Structure](image2)
3. In transaction SE11 create table type YMDGM_BUPA01_T as shown.

4. Make sure table type YMDGM_BUPA01_T is a Sorted Table.
5. Make sure table type `YMDGM_BUPA01_T` uses the keys as shown.

![Dictionary: Display Table Type](image)

6. In transaction SE11 create structure `YMDGM_BUPA01_S_X` as shown.

   Note: `DELETE_ROW` indicates that this row shall be deleted.

![Dictionary: Display Structure](image)
7. In transaction SE11 create table type YMDGM_BUPA01_T_X as shown.

8. Make sure table type YMDGM_BUPA01_T_X is a sorted table.
9. Make sure table type YMDGM_BUPA01_T_X has the key fields as shown.

**MDG Data Model Extension**

In this section you will extend the MDG data model with the entity type YBUPA01. The entity type has one attribute ZZNICHN and one qualifying relationship to entity type YBUPA. YBUPA01 will have two key fields MATERIAL and YBUPA. For details of the data model extension refer to the figure below.
Figure: Data Model details for extension

Extend MDG BS MAT (Material Master Data Structure)

1. Start transaction SE11.
   Enter MDG_BS_MAT_S_MAT_DATA as shown.
   Choose Display.
2. Create a new **Append**.

3. Enter the **Append Name** as shown.
4. Enter the append component details as shown.

5. Save your changes and make sure the append appears in structure `MDG_BS_MAT_S_MAT_DATA` as shown.
Extend MDG Data Model

1. Start the MDG customizing with transaction MDGIMG and start the Configuration Workbench as shown.
2. In the Configuration Workbench open the **MM Data Model**.

3. Choose the **Edit** pushbutton.
4. To create a new entity type, choose the **New** pushbutton.

5. Create the new entity type YBUPA01 (**SU Type 4**) with the details shown in the screenshot.
6.  Choose Save.
Choose OK in the confirmation dialog.

7.  Create the new Entity Type **YBUPA** *(SU Type 3)* with the details shown in the screenshot.
Choose **Save**.

Choose **OK** in the confirmation dialog.

---

**9.**

Go back to Entity Type YBUPA01.

Switch the **Attributes** tab.

Choose the **New** pushbutton.

---

**10.**

Create the new attribute **NICKNAME** as shown.
11. Choose **Save**. Choose **OK** in the confirmation dialog.

12. For **Entity Type YBUPA01** open the **Incoming Relationships** tab. Choose the **New** pushbutton.

13. Create the two relationships as shown.
<table>
<thead>
<tr>
<th>Step</th>
<th>Action</th>
</tr>
</thead>
<tbody>
<tr>
<td>14.</td>
<td>Choose the <strong>Activate</strong> pushbutton to activate your data model.</td>
</tr>
<tr>
<td>15.</td>
<td>To check the activated data model open the <strong>Edit Data Model</strong> Customizing activity as shown.</td>
</tr>
<tr>
<td>16.</td>
<td>Choose the <strong>Visualize Data Model</strong> pushbutton.</td>
</tr>
</tbody>
</table>
17. Make sure that your extension of the MDG Data Model looks similar to the details shown in the screenshot.

Generate Model-Specific Structures

Every time you change the MDG data model, you must regenerate the structures. In this Customizing activity, for each data model and entity type you generate technical structures in the ABAP Dictionary. The system uses these structures internally for implementing the staging area. To generate these Data Model-Specific structures follow the steps below.

Note

In general if you change a data model (for example, if you change attributes of entity types or relationships); you must regenerate the structures.
1. In MDG customizing start **Generate Data Model-Specific Structures**.

2. Select Data Model **MM**. Open the **Structures** view.

3. Create the tree entries in the structures table for entity type **YBUPA01**.
4. Save your changes.

Create SMT-Mapping

You extend mappings by creating new transformations (complex transformations, field mappings) and field checks for them or by editing them.

Important

When the mappings are saved the corresponding coding is generated. Make sure that all relevant structures are ready before you start.

Create Mapping Entries in Customizing

1. In MDG customizing start **Generate Data Model-Specific Structures**.
2. Select the MM data model.  
   Open the **Mapping** view.

![Change View "Data Models": Overview](image)

3. Create a new entry for Entity Type YBUPA01 / Active Area Mapping.  
   Mapping from the active area to the staging area:  
   **Z_MAP_YBUPA01_2_STA**  
   Mapping from the staging area to the active area:  
   **Z_MAP_YBUPA01_2_PP**

![Mapping Table](image)
Map the Active Area to the Staging Area

4. In MDG customizing start **Create and Edit Mappings**.

5. Select **Mapping -> New**.

6. Enter the name of the mapping:

   **Z_MAP_YBUA01_2_STA**
7. Create a new mapping step as shown.

8. Select your mapping step and choose the **Details** pushbutton.
9. Open the **Transformations** tab.

10. Add a new **Field Mapping**.

11. Enter the field mappings as shown. Save your changes.
Map the Staging Area to the Active Area

1. In MDG customizing start **Create and Edit Mappings.**

2. Select **Mapping -> New** to create the next mapping.

3. Enter the following mapping name:

   **Z_MAP_YBUPA01_2_PP**
4. Choose **Add** to create new mapping step using the details shown in the screenshot.

5. Choose the **Change Structure Keys** pushbutton. Then add the key fields of the change structure, as shown.
6. Add the key fields as shown.

7. Select your mapping step and choose the Details pushbutton.

8. Open the Transformations tab.
9. Choose the **Add** pushbutton to create a new **Field Mapping** entry.

10. Create the field mapping as shown. Save your changes.
### Adjust Staging Area of Linked Change Requests

This step is necessary to adjust any open change requests after you have changed the data model.

1. **Start Customizing for** [Master Data Governance](https://scn.sap.com) (transaction MDGIMG).
   
   Go to **General Settings -> Data Modeling -> Edit Data Model**.
   
   Select data model **MM**.
   
   Double click on **Entity Types**.
   
   Choose the pushbutton **Adjust staging area of linked change requests**.

2. Choose the **Yes** pushbutton.

3. The following message appears.

   **Note:** Make sure that user DDIC exist in all relevant clients.

---

### Conversion Program

*Conversion program USMD_ADJUST_STAGING scheduled in client 405.*
### BAAdI: Extension of the API with Customer-Specific Segments

1. In MDG customizing start the selected entry as shown.

![Image of Display IMG](image1)

2. Create a new BAAdI implementation.

![Image of BAAdI Implementations](image2)

3. Enter the name of the Enhancement Implementation as shown.

![Image of Create Enhancement Implementation](image3)
<table>
<thead>
<tr>
<th>Step</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>4.</td>
<td>Enter further details for the BAdI as shown in the screenshot.</td>
</tr>
<tr>
<td>5.</td>
<td>Check your Enhancement Implementation details look similar to the screenshot.</td>
</tr>
<tr>
<td>6.</td>
<td>Create a new filter entry.</td>
</tr>
</tbody>
</table>
7. Enter the filter value **YMDGM_BUPA01_S**

Creating a Selection-Range Table

1. The **IS_SELECTION** parameter contains the material selection criteria for which the database select must be executed. The selection criteria always contain the **MATNR** and possibly additional key fields of the customer-specific database table.

   In case you have key fields in your customer-specific table you must append **MDG_BS_MAT_S_MAT_SELECTION** by your key-fields.

   This is shown in the next step.

2. Start transaction
SE11.
Open structure MDG_BS_MAT_S_MAT_SELECTION.
Create an append called BUPA_ID_RANGE as shown.

It is vital that the component name of the append adheres to the naming standard <KEY-FIELDNAME OF BACKEND TABLE>_RANGE.

In our case the backend table is YMDGM_BUPA00 and the key-fieldname is BUPA_ID therefore the component name is BUPA_ID_RANGE.

3.
Here are the details for structure YBUPA_GUID_T_RANGE that you need in the previous step.

**BAdI Implementation: READ Method**

1. `method IF_MDG_BS_MAT_API_SEGMENTS_EXT.READ.
   DATA:`
Extend User Interface

Create FPM List UIBB

1. Navigate to the Component Configuration as shown in the screenshot.
2. Open configuration FPM_LIST_UIBB_TEMPLATE.

3. Choose the **Start Configuration** pushbutton.
4. Choose the **Copy** icon.

5. Enter a name for the copy as shown.
6. Refresh the navigation tree and open the new configuration.
<table>
<thead>
<tr>
<th>Step</th>
<th>Instruction</th>
</tr>
</thead>
<tbody>
<tr>
<td>7.</td>
<td>Choose the <strong>Start Configuration</strong> pushbutton.</td>
</tr>
<tr>
<td>8.</td>
<td>Choose the pushbutton <strong>Continue in Change Mode</strong>.</td>
</tr>
<tr>
<td>9.</td>
<td>Under <strong>General Settings</strong> choose the <strong>Feeder Class</strong> pushbutton.</td>
</tr>
</tbody>
</table>
10. Enter the name of the feeder class.

11. To confirm, choose Yes.

12. Enter the feeder parameters as shown.

   Make sure you enter a value for the number of default rows.
13. Choose the **Column** pushbutton.

![Component Configuration](image)

14. View the list of available columns.

![Edit Columns](image)
Select the columns from the list of available columns as shown.
Add List UIBB to Material UI

1. Navigate to the Application Configuration as shown.
2. Open the configuration BS_MAT_OVP.

3. Choose the Start Configuration pushbutton.

4. Choose the Continue in Display Mode pushbutton.
5. Navigate to the configuration BS_MAT_OVP_LAYOUT.

6. Select Additional Functions -> Enhance.

7. Provide a name for your enhancement.
8. Add a new List Component to the Section: MAT_SECTION element as shown.

9. Enter the details for the new list component.
10. Create a new wire entry with the details shown in the screenshot.

11. Save your changes.
Testing Your Data Model Extension

To test your configuration, start the MDG Material UI using the following URL (replace the parameters host, port and client-id to match your landscape):

https://<host>:<port>/sap/bc/webdynpro/sap/mdg_bs_mat?ACTION=CREATE&WDCONFIGURATIONID=BS_MAT_INIT&sap-client=<client-id>

Alternatively, start transaction PFCG, enter role name SAP_MDGM_MENU and click the Display pushbutton. Select the Menu tab. In the hierarchy window, navigate to Role Menu -> Material Governance -> Material Processing. Right click on Create Material and select Execute from the drop-down.
Related Content

1. For more information, visit the Configuration and Enhancement of SAP Master Data Governance homepage. ([http://scn.sap.com/docs/DOC-785](http://scn.sap.com/docs/DOC-785))
Copyright

© Copyright 2014 SAP AG. All rights reserved.

No part of this publication may be reproduced or transmitted in any form or for any purpose without the express permission of SAP AG. The information contained herein may be changed without prior notice.

Some software products marketed by SAP AG and its distributors contain proprietary software components of other software vendors.

Microsoft, Windows, Excel, Outlook, and PowerPoint are registered trademarks of Microsoft Corporation.

IBM, DB2, DB2 Universal Database, System i, System i5, System p, System p5, System x, System z, System z10, System z9, z9, iSeries, pSeries, xSeries, zSeries, eServer, z/VM, z/OS, i5/OS, S/390, OS/390, OS/400, AS/400, S/390 Parallel Enterprise Server, PowerVM, Power Architecture, POWER6+, POWER6, POWER5+, POWER5, POWER, OpenPower, PowerPC, BatchPipes, BladeCenter, System Storage, GPFS, HACMP, RETAIN, DB2 Connect, RACF, Redbooks, OS/2, Parallel Sysplex, MVS/ESA, AIX, Intelligent Miner, WebSphere, Netfinity, Tivoli and Informix are trademarks or registered trademarks of IBM Corporation.

Linux is the registered trademark of Linus Torvalds in the U.S. and other countries.

Adobe, the Adobe logo, Acrobat, PostScript, and Reader are either trademarks or registered trademarks of Adobe Systems Incorporated in the United States and/or other countries.

Oracle is a registered trademark of Oracle Corporation.

UNIX, X/Open, OSF/1, and Motif are registered trademarks of the Open Group.

Citrix, ICA, Program Neighborhood, MetaFrame, WinFrame, VideoFrame, and MultiWin are trademarks or registered trademarks of Citrix Systems, Inc.

HTML, XML, XHTML and W3C are trademarks or registered trademarks of W3C®, World Wide Web Consortium, Massachusetts Institute of Technology.

Java is a registered trademark of Oracle Corporation.

JavaScript is a registered trademark of Oracle Corporation, used under license for technology invented and implemented by Netscape.

SAP, R/3, SAP NetWeaver, Duet, PartnerEdge, ByDesign, SAP Business ByDesign, and other SAP products and services mentioned herein as well as their respective logos are trademarks or registered trademarks of SAP AG in Germany and other countries.

Business Objects and the Business Objects logo, BusinessObjects, Crystal Reports, Crystal Decisions, Web Intelligence, Xcelsius, and other Business Objects products and services mentioned herein as well as their respective logos are trademarks or registered trademarks of Business Objects S.A. in the United States and in other countries. Business Objects is an SAP company.

All other product and service names mentioned are the trademarks of their respective companies. Data contained in this document serves informational purposes only. National product specifications may vary.

These materials are subject to change without notice. These materials are provided by SAP AG and its affiliated companies ("SAP Group") for informational purposes only, without representation or warranty of any kind, and SAP Group shall not be liable for errors or omissions with respect to the materials. The only warranties for SAP Group products and services are those that are set forth in the express warranty statements accompanying such products and services, if any. Nothing herein should be construed as constituting an additional warranty.