How To...
Master Data Governance for Material
Extend Search

Applicable Releases:
EhP6; MDG6.1; MDG7.0

Version 1.80
May 2015
## Document History

<table>
<thead>
<tr>
<th>Document Version</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.20</td>
<td>First official release of this guide</td>
</tr>
<tr>
<td>1.30</td>
<td>Chapter 8, 9 and 10</td>
</tr>
<tr>
<td>1.40</td>
<td>Small corrections</td>
</tr>
<tr>
<td>1.50</td>
<td>New chapter 9.3</td>
</tr>
<tr>
<td>1.60</td>
<td>Small corrections, new chapter 9.2.1 Change Search Criteria from MDG7.0 on</td>
</tr>
<tr>
<td>1.70</td>
<td>New chapters 4 and 10.4; Updates to chapters 5.5.2.3 and 8</td>
</tr>
<tr>
<td>1.80</td>
<td>New chapter 8.7.1 and 8.7.2</td>
</tr>
</tbody>
</table>
## Typographic Conventions

<table>
<thead>
<tr>
<th>Type Style</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Example Text</strong></td>
<td>Words or characters quoted from the screen. These include field names, screen titles, pushbuttons labels, menu names, menu paths, and menu options.  &lt;br&gt;Cross-references to other documentation</td>
</tr>
<tr>
<td><strong>Example text</strong></td>
<td>Emphasized words or phrases in body text, graphic titles, and table titles</td>
</tr>
<tr>
<td><strong>Example text</strong></td>
<td>File and directory names and their paths, messages, names of variables and parameters, source text, and names of installation, upgrade and database tools.</td>
</tr>
<tr>
<td><strong>Example text</strong></td>
<td>User entry texts. These are words or characters that you enter in the system exactly as they appear in the documentation.</td>
</tr>
<tr>
<td><code>&lt;Example text&gt;</code></td>
<td>Variable user entry. Angle brackets indicate that you replace these words and characters with appropriate entries to make entries in the system.</td>
</tr>
<tr>
<td><strong>EXAMPLE TEXT</strong></td>
<td>Keys on the keyboard, for example, <strong>F2</strong> or <strong>ENTER</strong>.</td>
</tr>
</tbody>
</table>
Table of Contents

1. Business Scenario ........................................................................................................... 1
2. Background Information ................................................................................................. 1
3. Prerequisites ..................................................................................................................... 2
4. Hints for Enterprise Search .............................................................................................. 2
5. Step-by-Step Procedure: Extend with Existing Attributes (Reuse) ................................ 3
   5.1 Extend Model MM: Entity Material ............................................................................ 3
   5.2 Extend Structures for Data Model MM ...................................................................... 4
   5.3 SMT Mapping ............................................................................................................. 5
   5.4 Active Area: Connector for MATERIAL .................................................................... 5
      5.4.1 Template Modeler for active area ...................................................................... 5
         5.4.1.1 Create new Software Component ............................................................... 5
         5.4.1.2 Create Connector for Material .................................................................... 7
      5.4.2 Administration Cockpit ...................................................................................... 8
         5.4.2.1 Update Connector Material (optional) ......................................................... 8
         5.4.2.2 Schedule Indexing ....................................................................................... 9
   5.4.3 Test Connector Material ......................................................................................... 10
   5.5 Staging Area: Connector for MDG ........................................................................... 11
      5.5.1 Template Modeler .............................................................................................. 11
         5.5.1.1 Software component ................................................................................. 11
         5.5.1.2 Create new search connector template for MDG .................................. 11
      5.5.2 Administration Cockpit ...................................................................................... 13
         5.5.2.1 Re-create the associations, authorizations, request and response attributes (optional) ................................................................................................................................. 13
         5.5.2.2 Add the new attributes to ZMDG_MATERIAL template request fields ................................................................................................................................. 17
         5.5.2.3 Add the new attributes to the ZMDG_MATERIAL template response fields ................................................................................................................................. 17
         5.5.2.4 Schedule Indexing ....................................................................................... 18
      5.5.3 Test Connector ZMDG_MATERIAL .................................................................... 18
   5.6 Customizing ................................................................................................................. 18
6. Step-by-Step Procedure: Extend by custom attributes (from Reuse) .............................. 20
   6.1 Extend Backend: MARA with Append ....................................................................... 20
   6.2 Extend ES Structure for Material: ESO_S_MARA_MAW1 ........................................ 20
   6.3 Extend Model MM: Entity Material ........................................................................... 22
   6.4 Extend Structures for Model MM .............................................................................. 22
   6.5 SMT Mapping ............................................................................................................. 23
   6.6 Active Area: Connector for MATERIAL .................................................................... 23
      6.6.1 Template Modeler .............................................................................................. 23
         6.6.1.1 Enhance Template MATERIAL .................................................................. 23
         6.6.1.2 Add attributes to the Request Fields ......................................................... 27
6.6.2 Administration Cockpit ......................................................... 30
6.6.2.1 Update Connector Material ............................................. 30
6.6.2.2 Schedule Indexing ......................................................... 30
6.6.3 Test Connector Material.......................................................... 32
6.7 Staging Area: Connector for MDG ............................................. 32
6.8 Customizing ............................................................................. 32

7. Step-by-Step Procedure: Extend with Existing Entity (Reuse) .............. 33
7.1 Extend Model: Entity Plant (YMARC) ........................................ 33
7.2 Extend Structures for Model ..................................................... 33
7.3 SMT Mapping ........................................................................... 33
7.4 Active Area: Connector for MATERIAL ...................................... 33
7.5 Staging Area: Connector for MDG ............................................. 34
7.6 Customizing ............................................................................. 34

8. Step-by-Step Procedure: Extend with Custom Entity (Reuse) ............... 35
8.1 Backend: New Table ZMDGM_BU PA01 ..................................... 35
8.2 Create ES Structure for ZMDGM_BU PA01: ZESO_S_ZMDGM_BU PA01 35
8.3 Extend the Extraction Logic for the new ES node ......................... 36
8.4 Extend Model MM: Entity ZBU PA ............................................ 39
8.5 Extend Structures for Model MM ............................................... 39
8.6 SMT Mapping ........................................................................... 39
8.7 Active Area: Connector for MATERIAL ...................................... 39
8.7.1 BAdI Implementation: GET_ES_NODEINFO Method .................. 43
8.7.2 BAdI Implementation for handling Change Pointers .................. 43
8.8 Staging Area: Connector for MDG ............................................. 44
8.9 Customizing ............................................................................. 44

9. Step-by-Step Procedure: Extend with Existing Entity (Flex) ............... 45
9.1 Extend Model: Entity YMARC3 ................................................ 45
9.2 Extend Structures for Model ..................................................... 46
9.3 SMT Mapping ........................................................................... 46
9.4 Active Area: Connector for MATERIAL ...................................... 47
9.4.1 Enhance Template Material for the Flex Node ......................... 47
9.4.2 BAdI Implementation for Indexing Flex node data .................... 50
9.4.3 BAdI Implementation for handling Change Pointers for Flex node data 53
9.5 Staging Area: Connector for MDG ............................................. 53
9.5.1 Create new search connector template for MDG ...................... 53
9.5.2 Add the request fields to Z1MDG_MATERIAL template ............ 54
9.5.3 Schedule Indexing ............................................................... 55
9.5.4 Test Connector: Z1MDG_MATERIAL ..................................... 55
9.6 Customizing ............................................................................. 55

10. Step-by-Step Procedure: Extend Search UI .................................... 56
10.1 Enhance UIBB MDG_BS_MAT_SEARCH_RESULT_LIST .................. 56
10.2 Change Search Criteria Values ............................................................................................................. 57
  10.2.1 Change Search Criteria from MDG7.0 on ......................................................................................... 58
10.3 Default for class type .................................................................................................................................. 58
10.4 Add field descriptions to the result list ................................................................................................. 59
  10.4.1 POST-EXIT for method CL_MDG_BS_MAT_BO_SEARCH->
            GET_SEARCHABLE_ATTRIBUTES ......................................................................................... 59
  10.4.2 POST-EXIT for method CL_MDG_BS_MAT_SP_SEARCH->
            COMPLETE_DATA ................................................................................................................. 63

11. Multiple Languages ..................................................................................................................................... 64
1. Business Scenario

SAP Master Data Governance for Material (MDG-M) provides business processes to find, create, change, and mark material master data for deletion. It supports the governance of material master data on a central hub and the distribution to connected operational and business intelligence systems. The processes are workflow-driven and can include several approval and revision phases, and the collaboration of all users participating in the master data maintenance.

In EhP6 MDG for Material master data domain addresses only Basic Data, Descriptive Data, and Classification Data for Materials but it is possible to extend the data model. If you want to search with these new fields you have to extend the search.

2. Background Information

There are 5 possible scenarios how you can extend the standard MDG Material data model.

   a. **Extend with existing attributes (reuse):** Extend the data model with existing fields of the reuse table. An example can be found in Chapter 5: Fields FORMT, MSTAE of MARA table.

   b. **Extend with custom attributes (reuse):** Extend the data model with custom fields that are in reuse. An example can be found in Chapter 6: Two new fields ZZBRAND and ZZPRODTYPE defined by customer in backend MARA table.

   c. **Extend with existing entities (reuse):** Extend the data model with existing entities in reuse. An example can be found in Chapter 7: Plant Entity – YMARC.

   d. **Extend with custom entities (reuse):** Extend the data model with a custom entity that is defined in reuse. An example can be found in Chapter 8: A new table ZMDG_BUPA defined in reuse.

   e. **Extend with existing entities (based on flexible entities):** The data model is extended with a new entity (as in scenario D) but the data for this resides not in the underlying SAP database tables but in the MDG-generated tables. An example can be found in Chapter 9.

For all the scenarios mentioned above, you have to enhance the standard data model either with additional attributes or with additional entities. Also, the search must be extended, so that you are able to search with these fields. You have 2 search connectors for Enterprise Search of Material: MATERIAL (for the active area) and MDG_MATERIAL (for the staging area). The detailed steps on how to extend the data model and the enterprise search are covered from chapter 5 onwards.

If the ES is unavailable because the data and search model enhancement is not finished, you can do the following to mitigate the ES dependencies:

In the MDG IMG (General Settings -> Data Quality and Search -> Business Add-Ins -> BAdI: Search for Data in Reuse Active Area), deactivate temporarily the implementation MDG_BS_MAT_PP_SEARCH.

**Result:**

The QUERY method of the access class can search for materials by ID and description. That means, that the standard value help for materials still works, but Material search doesn’t work. After you finished your data and search model enhancements don’t forget to activate the implementation again.
3. Prerequisites

- Enhancement of Master Data Governance Content:
  [http://help.sap.com/erp2005_ehp_06/helpdata/EN/ab/05f68761744157b40f81af8689b700/frameset.htm](http://help.sap.com/erp2005_ehp_06/helpdata/EN/ab/05f68761744157b40f81af8689b700/frameset.htm)

- Extensibility Guides:

- SAP Notes:
  - 2157166 ES reuse data indexing
  - 2110371 SPI Metadaten
  - 2106708 SPI metadata: Removal of unused nodes
  - 2095659 SEARCH: Performance improvement
  - 2098606 SEARCH: Check boxes 'Fuzzy Search' and 'Search Active Data Only'
  - 2100398 SEARCH: Object characteristics are offered as request attributes
  - 2106680 SEARCH: Response fields are not determined correctly
  - 2108051 Search criteria / custom parameters get lost in case of error
  - 2123673 SelectOptions20: Default label for search term is wrong

4. Hints for Enterprise Search


Enterprise Search content is shipped via logical transport object SEST. For details concerning the Import in all clients please refer to note 1790318.

**Note:** It is not recommended to regenerate a template after go live. New attributes should be added manually.
5. Step-by-Step Procedure: Extend with Existing Attributes (Reuse)

Scenario:
You have extended the standard MDG Material data model MM with additional attributes that already exist on the underlying SAP reuse table MARA (For example, FORMT for Page Format or MSTAE for Cross Plant Material Status, which were not available in the data model MM in EHP5) and you want to search for materials based on these attributes.

5.1 Extend Model MM: Entity Material
Refer the standard extensibility guide on how to enhance/extend the data model.

The screenshot below illustrates this.

Include Search Helps for the new attributes. Then also F4 Helps on the fields are available in the Search screen.
5.2 Extend Structures for Data Model MM

Important for Search in staging area: Ensure that the generated ES structure `/MDGMM/_S_MM_ES_MATERIAL` has the new attributes added to the model.
5.3 SMT Mapping

If the data model is already enhanced and mappings have been defined, then this chapter can be skipped. If not, then you need to extend the mappings for the fields by editing each of the existing mappings MDG_BS_MAT_MAP_2PP and MDG_BS_MAT_MAP_2STA by mapping the corresponding fields in source and target structures.

The screenshot below illustrates how to extend the mappings from MDG Customizing.

5.4 Active Area: Connector for MATERIAL

Note: For this scenario (extension by existing standard SAP attributes of reuse), the attributes will already be part of the Enterprise Search 'MATERIAL' template. However, if the fields are not yet part of the standard search template 'MATERIAL' of SAP_APPL SWC, then the following steps must be performed.

5.4.1 Template Modeler for active area

If the fields are not yet part of the standard search template 'MATERIAL' please refer to 6.6.1 Template Modeler and perform the mentioned steps 6.6.1.1 to 6.6.1.4.

5.4.1.1 Create new Software Component

To open the Modeler, select the Modeler hyperlink. In the Software Component menu choose Maintain Software Components and choose Create. Enter the name for the new software component and assign a package. The software component type is Normal Software Component. Choose Create. Enter a description for the new software component and choose Save. Select a workbench request.
Include software components, here for example PLMWUI.

Note:
To transport the new software component and the new template to other systems please refer to SAP note 1483890.

5.4.1.2 Create Connector for Material

Mark your new software component and from the Software Component menu select Include Software Components. Include the highest software component for connector MATERIAL. You need to create the connector for MATERIAL in the highest software layer. If software layer hierarchy is PLMWUI / EA-APPL/ SAP-APPL then use SW Component PLMWUI. If software layer hierarchy is EA-APPL / SAP-APPL then use SW Component EA-APPL. The template MATERIAL is part of the software hierarchy (For example, PLMWUI/EA_APPL/SAP_APPL/LO/LO_MD/MM/MATERIAL).

Once you have located the template, create the connector for the Enterprise Search. To create a search connector for the corresponding template, from the Actions menu select Create Connector.
5.4.2 Administration Cockpit

5.4.2.1 Update Connector Material (optional)

The update is only necessary if the fields are not yet part of the standard search template MATERIAL.

Please refer to 6.6.2.1 Update Connector Material
5.4.2.2 Schedule Indexing

Schedule indexing for the object types of this connector and its dependent connectors. Enter values directly in the table or use the input form.

Options:
- **Start Time**
- **Recurrence Period**
- **Indexing Mode**
  - Full Indexing Mode
  - Copy Settings

Example Object Details:
- **Object**
  - ACCESS_CONTROL
  - Business Object
  - Prepared
  - Start Time: 02.11.2019 11:00:00

Additional Settings:
- **Start Date**
- **Start Time**
- **Full Indexing Mode**
- **Real-Time Indexing**
- **Months**
- **Weeks**
- **Days**
- **Hours**
5.4.3 Test Connector Material

Execute report ESH_TEST_SEARCH: & choose the connector for MATERIAL. Search based on any attribute by checking the Request Attribute-Based Search check box as shown in the screenshot below.
5.5 Staging Area: Connector for MDG

5.5.1 Template Modeler

5.5.1.1 Software component
Delete all the existing connectors for the MDG_MATERIAL template. Execute the transaction ESH_MODELER and select the software component that you created in Chapter 5.4.1.1 Create new Software Component.
Mark your new software component and in the Software Component menu select Include Software Components. Include the software component MDG_APPL (for connector MDG_MATERIAL). Choose Save and select a Workbench Transport.
To transport the new software component and the new template to other systems please refer to SAP note 1483890.

5.5.1.2 Create new search connector template for MDG
Create a new Enterprise Search connector template by using transaction MDG_ES_TEMPL or in the implementation guide under General Settings > Data Quality and Search > Create Search Object Connector Templates. Once the Enterprise Search template is generated, a relationship has to be maintained between the Object Type Code (OTC) and the Enterprise Search Template in the implementation guide under General Settings > Data Quality and Search > Assign Search Object Connector Templates to Object Types.

Create new search connector template for MDG under the new software component and assign the new connector to object type 194.
If you have already created a ZMDG_MATERIAL and you want to override the existing template then you have to delete all existing associations (example: to CLES_CLASSIFICATION), authorizations, and relevant request, and response attributes to allow for regeneration (please refer to chapter 5.5.2.1 Re-create the associations, authorizations, request and response attributes). Otherwise it is not possible to update the template.

Alternative: Generate another new template with this transaction.

Note: If you have problems with TXTMI updates please refer to note 1728230.
5.5.2  Administration Cockpit

5.5.2.1  Re-create the associations, authorizations, request and response attributes (optional)

Note: This step is only necessary if you had overridden the template ZMDG_MATERIAL.
How To... MDG-M: Extend Search

MDG Modeler for Search and Analytics

Define Model ZMDG_MATERIAL of Software Component ZEM_EXT_MDG_MAT Model Language: MDG

List of Nodes of ZMDG_MATERIAL:

- MATERIAL
- BSCDat
t
- CLAS
t
- INT
t
- T_MAT
- UNIT
- VALU
- YMAD
- ZMDG

Details: Attributes of Node MATERIAL

- PROPERTY
- DESCRIPTION
- DATA SOURCE
- DATA SOURCE DE
- ATT.
- NAME
- B#
Go to node Relations, Choose the row Material and click on Create Association.

Create an association from the material node to MDG_CLF_MATERIAL and choose the target node CLES_CLASSIFICATION in the pop-up.

Create another association from Material to MTART_BASIC. Choose the target node MTART.

Choose the newly created association (MDG_CLF_MATERIAL) and create a foreign key relationship as shown in the screen shot.
Choose Cardinality as Exactly One and Reverse Cardinality as Arbitrary.

Choose the newly created association (MTART_BASIC) and create a foreign key relationship as shown in the screen shot.

Choose the Cardinality as Exactly One and Reverse Cardinality as Arbitrary.
5.5.2.2 Add the new attributes to ZMDG_MATERIAL template request fields

The request fields determine the fields which you could see as search criteria. With the customizing in chapter 5.6 you also change the Field Properties, so that the fields are not shown as search criteria.

5.5.2.3 Add the new attributes to the ZMDG_MATERIAL template response fields

The response fields determine the fields you can see in the results list. With the customizing shown in chapter 5.6 you can also change the Field Properties, so that the fields are not used in the selection or in the search results list. You can only add fields to the response structure where the cardinality is 1:1 to the material. That means, for example, that MARC fields are not supported.

From MDG 7.0 onwards you do not need to enhance the response attributes. The fields which are available for the configuration are only defined with the customizing shown in chapter 5.6. For enhancing the UI, see also chapter 10.1 Enhance UIBB MDG_BS_MAT_SEARCH_RESULT_LIST.
5.5.2.4 Schedule Indexing

Refer to chapter 5.4.2.2 Schedule Indexing and use the your new template ZMDG_MATERIAL

5.5.3 Test Connector: ZMDG_MATERIAL

Refer to chapter 5.4.3 Test Connector Material and use the new template ZMDG_MATERIAL

5.6 Customizing

You can customize the fields in the drop down of the search attributes and the fields in the search result list using “Field Properties”. Check that the new fields are not disabled for search and result list in this customizing; the check boxes for No Selection and No Results List should be unchecked.
Note:
Ensure that an entity attribute is at least one time not flagged as ‘No Selection’ that this attribute is contained in the list of the material search criteria.
Prerequisite: SAP note 2147531 - SEARCH: Support entity attributes defined in more than one entity type. After having implemented this note, you need to flag an entity attribute as non-selectable in all entity types to define it as non-selectable in the search UI.
6. Step-by-Step Procedure: Extend by custom attributes (from Reuse)

Scenario:
You have extended the standard Material data model MM with custom attributes that are defined on the underlying SAP reuse table.

Example:
Backend table MARA is extended with two fields (Brand ZZBRAND and Product Type ZZPRODTYPE) and search for materials should be done with these attributes.

Note:
This is also valid if you extended other material tables like MARC, MARD. Then you have to extend the corresponding structures (like ESO_S_MAKT_LTX, ESO_S_MARC, ESO_S_MARD, ESO_S_MARM, ESO_S_MEAN, ESO_S_MVKE, and ESO_S_MVKE_LONGTEXT).

6.1 Extend Backend: MARA with Append
Refer to SAP note 44410 that describe the steps that have to be carried out in order to integrate customer specific fields into the Material Master.

6.2 Extend ES Structure for Material:
ESO_S_MARA_MAW1
ESO_S_MARA_MAW1 is used in the ES Template for Material (MARA). Therefore it must be extended with the Z*Fields.
How To... MDG-M: Extend Search

Use the include from MARA:
6.3 Extend Model MM: Entity Material

Refer chapter 5.1 Extend Model MM: Entity Material

Include Search Helps for the new attributes. Then also F4 Helps on the fields are available in the Search screen.

6.4 Extend Structures for Model MM

Refer chapter 5.2 and ensure that the custom attributes are part of the generated search structure for staging /MDGMM/_S_MM_ES_MATERIAL.
6.5 SMT Mapping

Refer chapter 5.3 and proceed similarly for the current custom attributes (ZZBRAND & ZZPRODTYPE) if the mapping is not already maintained.

6.6 Active Area: Connector for MATERIAL

6.6.1 Template Modeler

Check if a new Software Component and connector already exist for the MATERIAL template. If yes, then directly proceed.

Otherwise refer to Chapter 5.4.1 Template Modeler for active area and create a new software component and template.

6.6.1.1 Enhance Template MATERIAL

Enhance the template MATERIAL.
How To... MDG-M: Extend Search
How To... MDG-M: Extend Search

[Image of a detailed MDG-M interface showing search extend options and a table with attribute details]

THE BEST-RUN BUSINESSES RUN SAP™
6.6.1.2 Add attributes to the Request Fields

Until MDG6.1
The search result list only contains attributes defined in the entities MATERIAL, MARAPURCH or MARASALES. Attributes defined in other entities are not mapped between active and staging format. However, you can define an overwrite exit for method CL_MDG_BS_MAT_PP_SEARCH->IF_BADI_SDQ_PP_SEARCH~SEARCH_PP in order to consider attributes in custom defined entities having a 1:1 relationship to MATERIAL. Only in this case you have to add the attributes to the response fields (see 6.6.1.3). Then you have to do the customizing (see chapter 5.6) and the enhancement of the result list (see chapter 10.1).

From MDG7.0
Make sure that SAP Notes 2110371, 2106708, 2095659, 2098606, 2106680, 2108051 are implemented. Then there is no need for the overwrite exit anymore. Also you don’t need to enhance the response fields in the template. Only customizing (see chapter 5.6) and the enhancement of the result list (see chapter 10.1) is necessary.

6.6.1.3 Add attributes to the Response Fields

- Until MDG6.1
  The search result list only contains attributes defined in the entities MATERIAL, MARAPURCH or MARASALES. Attributes defined in other entities are not mapped between active and staging format. However, you can define an overwrite exit for method CL_MDG_BS_MAT_PP_SEARCH->IF_BADI_SDQ_PP_SEARCH~SEARCH_PP in order to consider attributes in custom defined entities having a 1:1 relationship to MATERIAL. Only in this case you have to add the attributes to the response fields (see 6.6.1.3). Then you have to do the customizing (see chapter 5.6) and the enhancement of the result list (see chapter 10.1).

- From MDG7.0
  Make sure that SAP Notes 2110371, 2106708, 2095659, 2098606, 2106680, 2108051 are implemented. Then there is no need for the overwrite exit anymore. Also you don’t need to enhance the response fields in the template. Only customizing (see chapter 5.6) and the enhancement of the result list (see chapter 10.1) is necessary.
How To... MDG-M: Extend Search
Save and Finish.

6.6.1.4 Create Connector for Material
6.6.2 Administration Cockpit

6.6.2.1 Update Connector Material

6.6.2.2 Schedule Indexing
6.6.3 Test Connector Material

Report ESH_TEST_SEARCH:

6.7 Staging Area: Connector for MDG

Refer chapters 5.5.1 Template Modeler to 5.5.3 Test Connector: ZMDG_MATERIAL

6.8 Customizing

Refer chapter 5.6 Customizing.
7. Step-by-Step Procedure: Extend with Existing Entity (Reuse)

Scenario:
You have extended the standard Material Data model MM with an existing underlying SAP reuse table like MARC. For example, you created a Plant Entity (YMARC) with attributes of the Plant and you want to perform searches based on these entity attributes.

7.1 Extend Model: Entity Plant (YMARC)
Refer to the standard extensibility guide on how to extend data model by additional entities. Screenshot below shows an example.

<table>
<thead>
<tr>
<th>Dialog Structure</th>
<th>Data Model</th>
<th>Entity Type</th>
<th>YMARC</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>Entity Types</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>StorageUseType</td>
<td>Changeable via Other Entity Type, Generated Database Tables</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Validity of Entity</td>
<td>No Edition</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Date Element</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Hierarchies</td>
<td>No Hierarchies</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Validity / Hierarchy</td>
<td>Hierarchy is Not Edition Dependent</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Key Assignment</td>
<td>Key Cannot be Changed, No Internal Key Assignment</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Language Dependent Texts</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Medium Text Length</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Short Text Length</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Attachments</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Sets</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Search Help</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Src. Fld Short Text</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Src. Fld Medium Text</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Src. Fld Long Text</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Temporary Keys</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Description</td>
<td>Plant Data for Material</td>
</tr>
</tbody>
</table>

7.2 Extend Structures for Model
Refer chapter 5.2 Extend Structures for Data Model and follow it for the YMARC entity.

7.3 SMT Mapping
Refer chapter 5.3 SMT Mapping and maintain the mapping if it is not already maintained.

7.4 Active Area: Connector for MATERIAL
Check if the connector exists for the MATERIAL template. The search MATERIAL template will also have the Plant node delivered as standard. If it does not exist, then refer chapters 6.6.1.1 Enhance
**Template MATERIAL** and *6.6.1.2 Add attributes to the Request Fields* to create a connector for MATERIAL that includes the PLANT node.

Next, refer to chapters *6.6.1.3* to *6.6.3* to add Plant attributes to the MATERIAL search template request and response fields, create and update connectors, schedule indexing, and test the connector.

**7.5 Staging Area: Connector for MDG**

Refer chapters *5.5.1 Template Modeler* to *5.5.3 Test Connector: ZMDG_MATERIAL* and follow the process for the Plant attributes.

**7.6 Customizing**

Refer chapter *5.6 Customizing*.
8. Step-by-Step Procedure: Extend with Custom Entity (Reuse)

Scenario:
You have introduced a new table (ZMDGM_BUPA01) to the reuse area (ECC) in order to maintain nicknames for materials. The table uses the 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 contains a nickname for the material. You also may have extended the standard material views with an additional view to maintain this information.

In order to govern this data with MDG for Material, you have extended the standard MDG Material data model (MM) with the new custom entity and its corresponding attributes. Now you want to perform searches for materials based on the new attributes.

8.1 Backend: New Table ZMDGM_BUPA01

8.2 Create ES Structure for ZMDGM_BUPA01: ZESO_S_ZMDGM_BUPA01

You must create a new Enterprise Search structure, ZESO_S_ZMDGM_BUPA01 because there is a 1:n relationship between MARA and ZMDGM_BUPA01.
How To... MDG-M: Extend Search

Note:
If there is a 1:1 relationship between the backend table and the <NEW_TABLE>, then the attributes can be added to the existing structures (such as ESO_S_MARA_MAW1, ESO_S_MARC, ESO_S_MARD, ESO_S_MARM, and ESO_S_MEAN, ESO_S_MVKE). Refer to chapter 6.2 and follow the process to extend the existing structure.

8.3 Extend the Extraction Logic for the new ES node
You need to enhance the extraction logic to get the data for the new Enterprise Search node created for ZMDGM_BUPA01 from the database into enterprise search.

Create a BADI implementation for BADI_ESH_IF_OBJECT_DATA_ENH (enhancement spot BADI_ESH_IF_OBJECT_DATA).

Use the following filter values:
- IV_OBJECT_TYPE_ID = ‘MATERIAL’
- IV_SOFTWARE_COMPONENT = ‘Your new software component’ (see chapter 5.4.1.1 Create new Software Component).
Provide implementation for the method GET_DATA. See sample code below.

```abap
METHOD if_badi_enh_if_object_data_enh-get_data.

DATA lc_bupa TYPE TABLE OF zmdgm_bupa01.
DATA ls_bupa TYPE zmdgm_bupa05.
DATA lt_bupa_map TYPE TABLE OF zmdgm_bupa05.

DATA ls_node_data TYPE KEY TO date.
FIELD-SymbolS : ls_extraction TYPE ash_if_extract_node.
FIELD-SymbolS : ls_object_id TYPE any.
FIELD-SymbolS : ls_node_material TYPE STANDARD TABLE.
FIELD-SymbolS : ls_node_material TYPE any.
FIELD-SymbolS : lt_material TYPE material.
FIELD-SymbolS : ls_bupa TYPE zmdgm_bupa01.

CLEAR et_messages.
READ TABLE ls_object_extraction ASSIGNING <ls_extraction> WITH TABLE KEY
node_type_id = cl_material_extraction->node_material.
IF sy-subrc <> 0.
  RETURN.
ENDIF.

ASSIGN <ls_extraction>-modify_table_ref-> TO <ls_node_material>.

LOOP AT lt_object_id ASSIGNING <ls_object_id>.
  ASSIGN COMPONENT 'MATERIAL' OF STRUCTURE <ls_object_id> TO <lt_material>.
  INSERT ls_bupa INTO TABLE lt_bupa_map.
ENDLOOP.

SELECT * INTO TABLE lt_bupa FROM zmdgm_bupa01 FOR ALL ENTRIES IN lt_bupa_map.
WHERE main = lt_bupa_map.main.
CREATE DATA ls_node_data LIKE LINE OF <lt_node_material>.
ASSIGN <ls_node_data>-data-> TO <ls_node_material>.

LOOP AT ls_bupa ASSIGNING <ls_bupa>.
  MOVE-CORRESPONDING <ls_bupa> TO <ls_node_material>.
  APPEND <ls_node_material> TO <ls_node_material>.
ENDLOOP.
ENDMETHOD.
```
How To... MDG-M: Extend Search

Provide implementation for the method NEXT. See sample code below.
8.4 Extend Model MM: Entity ZBUPA

Refer to the standard extensibility guide on how to extend data model with additional entities. Follow the process for the new entity ZBUPA.

8.5 Extend Structures for Model MM

Refer to chapter 5.2 Extend Structures for Data Model and follow the process for the entity ZBUPA.

8.6 SMT Mapping

Refer to chapter 5.3 SMT Mapping for general steps on how to extend the mappings and extend the mappings if not already done during data model extension.

8.7 Active Area: Connector for MATERIAL

The search MATERIAL template does not have the new custom entity/attributes by default. Extend the template for the MATERIAL with the node that corresponds to the new entity.

Refer to chapters 6.6.1.1 Enhance Template MATERIAL, 6.6.1.2 Add attributes to the Request Fields and 6.6.1.3 Add attributes to the Response Fields to first create the new software component and then the connector for MATERIAL that also includes the new custom node for ZMDGM_BUPA01.

Enhance the Template Material. Refer to chapter 6.6.1.1 and 6.6.1.2.
Provide the composition from the ROOT node (in this case MATERIAL) to the new node.

Go to the Node Relations step. Select the MATERIAL node and create a composition by choosing the Create button.
Create a composition for the MATERIAL_ZMDGM_BUPA0 node.

In the foreign keys section below, generate/maintain the foreign keys appropriately.
Add attributes to the Request Attributes. Refer to chapter 6.6.1.2 Add attributes to the Request Fields

Add to the Response Attributes. Refer to chapter 6.6.1.3 Add attributes to the Response Fields
If necessary, update your existing connector and schedule it again. Refer to chapters 6.6.1.4 to 6.6.3 to create and update connectors, schedule indexing, and test the connector.

8.7.1  BAdI Implementation: GET_ES_NODEINFO Method
You only need to implement this method if you want to search with the custom attributes of this Custom node. A new enhancement implementation has to be provided for the enhancement spot MDG_BS_MAT_API_SEGMENTS_EXT. You can also find this in the customizing Master Data Governance for Material -> Business Add-Ins-> BAdI: Extension of the API with Customer-Specific Segments.

Implement method GET_ES_NODEINFO if you have extended the enterprise search model MATERIAL. See also extensibility guide http://scn.sap.com/docs/DOC-27859.

With this method you get the node name in the Enterprise Search template for the given customer-defined database table with parameters ET_ES_NODENAME.

In case of Material Template extension (with new node) apart from the BADI the mapping should be based on SMT between Staging and Active.

```java
METHOD if_mdg_bs_mat_api_segments_ext-get_es_nodeinfo.

   DATA: lv_es_node TYPE tbnam.

   lv_es_node = 'MATERIAL_ZMDGM_BUPA0'
   APPEND lv_es_node TO et_es_nodename.
ENDMETHOD.
```

8.7.2  BAdI Implementation for handling Change Pointers
In order to handle change pointers for custom node, a new enhancement implementation has to be provided for the enhancement spot USMD_TRANSACTIONEVENTS.

SAP note 1669651 contains an example implementation (MDG_BS_MAT_ES_CP_FLEX_ENTITY) to create ES change pointers on activation. The example is for a flex entity, but will also work for custom node. You need to replace '<FlexET>' by your custom entity.

Create a new BAdI implementation and adapt the example implementation.
8.8 Staging Area: Connector for MDG
Refer to chapters from 5.5.1 Template Modeler to 5.5.3 Test Connector: ZMDG_MATERIAL and follow the process for the custom ZBUPA entity.

8.9 Customizing
Refer chapter 5.6 Customizing.
9. Step-by-Step Procedure: Extend with Existing Entity (Flex)

Scenario:
This scenario is similar to the scenario described in section (6) except that the entity for Plant is configured to be based on Flex than reuse.

9.1 Extend Model: Entity YMARC3

Refer the standard extensibility guide on how to extend data model by additional entities.

Note: Reuse area is specified as 'MDG' to indicate that this entity is a flex entity.
9.2 Extend Structures for Model

Refer to chapter 5.2 Extend Structures for Data Model

9.3 SMT Mapping

No mapping is required for flex nodes.
9.4 Active Area: Connector for MATERIAL

Extend the template for the MATERIAL with the node that corresponds to the flex entity (in this case for YMARC).

9.4.1 Enhance Template Material for the Flex Node

Enhance Template Material. Refer to chapter 6.6.1.1 and 6.6.1.2.

Provide the composition from ROOT node (in this case MATERIAL) to the flex node.

Go to Node Relations step. Click on the MATERIAL node and create a composition by clicking on the Create button. Create a composition to MATERIAL_PLANT_FLEX node. In the foreign keys section below, maintain the foreign keys appropriately.
Add attributes to Request Attributes. Refer to chapter 6.6.1.2 Add attributes to the Request Fields
How To... MDG-M: Extend Search

Add to Response Attributes. Refer to chapter 6.6.1.3 Add attributes to the Response Fields

If necessary, update your existing connector and schedule it again.
9.4.2 BADI Implementation for Indexing Flex node data

Enhancement Spot: BADI_ESH_IF_OBJECT_DATA

A new implementation has to be provided for the BADI BADI_ESH_IF_OBJECT_DATA_ENH.

As the BADI is filter based, the filter values for this should be based IV_OBJECT_TYPE_ID = 'MATERIAL' and IV_SOFTWARE_COMPONENT = <name of the software component under which the template is enhanced>.

As the implementation is not delivered as standard, sample code of the implementation is attached here.

Method: IF_BADI_ESH_IF_OBJECT_DATA_ENH~GET_DATA: (for Delta Indexing)

```
METHOD if_badi_esh_if_object_data_enh~get_data.

DATA:
  lt_sel TYPE usmd_ts_sel,
  lt_message TYPE usmd_t_message,
  lr_usmd_data_model TYPE REF TO if_usmd_model_ext,
  lr_data TYPE REF TO data.

FIELD-SYMBOLS:
```
**How To... MDG: Extend Search**

```verbatim
<fs_object_extraction>  TYPE esh_s_if_extract_node,
<ft_modify_table>      TYPE STANDARD TABLE,
<fs_modify_table>       TYPE any,
<fs_data>              TYPE any,
<fs_usmd_active>       TYPE usmd_active,
<ft_data>              TYPE ANY TABLE.

CLEAR et_messages.

CALL METHOD cl_usmd_model_ext=>get_instance
EXPORTING
  i_usmd_model = 'MM'
IMPORTING
  eo_instance = lr_usmd_data_model
  et_message = lt_message.

CALL METHOD lr_usmd_data_model->create_data_reference
EXPORTING
  i_fieldname = 'YMARC3'
  i_struct = if_usmd_model_ext=>gc_struct_key_attr
  if_table = 'X' " Financial MDM: General Indicator
  i_tabtype = if_usmd_model_ext=>gc_tabtype_standard
IMPORTING
  er_data = lr_data.

ASSIGN lr_data=>* TO <ft_data>.

CALL METHOD lr_usmd_data_model->read_char_value
EXPORTING
  i_fieldname = 'YMARC3'
  it_sel = it_sel
  i_readmode = if_usmd_model_ext=>gc_readmode_no_inact
IMPORTING
  et_data = <ft_data>
  et_message = lt_message.

LOOP AT it_object_extraction ASSIGNING <fs_object_extraction>
  WHERE node_type_id = 'MATERIAL_PLANT_FLEX' AND
      parent_node_type_id = 'MATERIAL'.
  ASSIGN <fs_object_extraction>-modify_table_ref=>* TO
      <ft_modify_table>.
  LOOP AT <ft_data> ASSIGNING <fs_data>.
      APPEND INITIAL LINE TO <ft_modify_table> ASSIGNING
      <fs_modify_table>.
      ASSIGN COMPONENT if_mdg_sdq_const=>usmd_active OF STRUCTURE
          <fs_modify_table> TO <fs_usmd_active>.
      IF sy-subrc = 0.
        <fs_usmd_active> = 1.
      ENDIF.
      MOVE-CORRESPONDING <fs_data> TO <fs_modify_table>.
  ENDLOOP.
```
Method **IF_BADI_ESH_IF_OBJECT_DATA_ENH~NEXT**: (for Initial Indexing)

```xml
METHOD if_badi_esh_if_object_data_enh~next.

DATA lt_sel TYPE usmd_ts_sel.
DATA lt_message TYPE usmd_t_message.
DATA lr_usmd_data_model TYPE REF TO if_usmd_model_ext.
DATA lr_data TYPE REF TO data.

FIELD-SYMBOLS: <fs_object_extraction> TYPE esh_s_if_extract_node,
            <ft_modify_table> TYPE STANDARD TABLE,
            <fs_modify_table> TYPE any,
            <fs_data> TYPE any,
            <fs_usmd_active> TYPE usmd_active,
            <ft_data> TYPE ANY TABLE.

CALL METHOD cl_usmd_model_ext=>get_instance
  EXPORTING
    i_usmd_model = 'MM'
  IMPORTING
    eo_instance = lr_usmd_data_model
    et_message = lt_message.

CALL METHOD lr_usmd_data_model->create_data_reference
  EXPORTING
    i_fieldname = 'YMARC3'
    i_struct = if_usmd_model_ext=>gc_struct_key_attr
    i_table = 'X' " Financial MDM: General Indicator
    i_tabtype = if_usmd_model_ext=>gc_tabtype_standard " Single-
                Character Indicator
  IMPORTING
    er_data = lr_data.

ASSIGN lr_data->* TO <ft_data>.

CALL METHOD lr_usmd_data_model->read_char_value
  EXPORTING
    i_fieldname = 'YMARC3'
    it_sel = lt_sel
    i_readmode = if_usmd_model_ext=>gc_readmode_no_inact " Read mode
  IMPORTING
    et_data = <ft_data>
    et_message = lt_message.

LOOP AT it_object_extraction ASSIGNING <fs_object_extraction>
  WHERE node_type_id = 'MATERIAL_PLANT_FLEX'
   AND parent_node_type_id = 'MATERIAL'.
```
9.4.3 BADI Implementation for handling Change Pointers for Flex node data

In order to handle change pointers for Flex node, a new enhancement implementation has to be provided for the enhancement spot USMD_TRANSACTION_EVENTS.

SAP note 1669651 contains an example implementation (MDG_BS_MAT_ES_CP_FLEX_ENTITY) to create ES change pointers on activation of a flex entity. This note is delivered.

Create a new BAdI Implementation and adapt the example implementation.

9.5 Staging Area: Connector for MDG

Refer chapters 5.5.1 Template Modeler to 5.5.3 Test Connector: ZMDG_MATERIAL and follow the procedure. Ensure that the request attribute names in staging template and corresponding active area template are the same.

9.5.1 Create new search connector template for MDG

Refer to chapter 5.5.1 Template Modeler.

Create a new Enterprise Search connector template by using transaction MDG_ES_TEMPL or in the implementation guide under General Settings > Data Quality and Search > Create Search Object Connector Templates. Once the Enterprise Search template is generated, a relationship has to be maintained between the Object Type Code (OTC) and the Enterprise Search Template in the
implementation guide under General Settings > Data Quality and Search > Assign Search Object Connector Templates to Object Types.

Create new search connector template for MDG under the new software component and assign the new connector to object type 194.

9.5.2 Add the request fields to Z1MDG_MATERIAL template

Refer to chapter 5.5.2.2 Add the new attributes to ZMDG_MATERIAL template request fields
9.5.3 Schedule Indexing
Refer to chapter 5.4.2 Administration Cockpit

9.5.4 Test Connector: Z1MDG_MATERIAL
Refer to chapter 5.5.3 Test Connector: ZMDG_MATERIAL

9.6 Customizing
Refer to chapter 5.6 Customizing.
10. Step-by-Step Procedure: Extend Search UI

10.1 Enhance UIBB

**MDG_BS_MAT_SEARCH_RESULT_LIST**

**Scenario:**

You want to see the added fields in the search results list. Until MDG6.1 the list only supports fields that are contained in the response structure of the MDG template. In MDG7 you can choose all attributes from entities with relation 1:1 to entity material.
10.2 Change Search Criteria Values

Scenario:
You want to change the defaults of the following search criteria’s in the Search UI: Material, Industry sector, and Description.

Solution:
This is not possible with configuration but it can be done with an overwrite-exit.

WD Component: MDG_BS_MAT_CLASSIF_SEARCH

Controller/Method:
Material_Search /Method WDDOINIT
->wd_comp_controller->mo_mat_sel_options->add_attributes

Componentcontroller / Method PROCESS_EVENT
->wd_this->mo_mat_sel_options->reset_attribute_definition

First_View / Method ONACTIONLOAD_SAVED_SEARCH
->wd_this->mo_mat_sel_options->reset_attribute_definition

```
* set initial order of attributes
IF ls_search_crit_det-name =
  if_mdg_bs_mat_assist_ui_search=>gc_so_attr_material OR
  ls_search_crit_det-name =
  if_mdg_bs_mat_assist_ui_search=>gc_so_attr_txtmi OR
  ls_search_crit_det-name =
  if_mdg_bs_mat_assist_ui_search=>gc_so_attr_mbrsh. <= here enhance

  ls_sel_attr_value-attribute = ls_search_crit_det-name.

  "Example Coding
  ls_sel_attr_value-sign = 'I'.
  ls_sel_attr_value-option = 'EQ'.
  ls_sel_attr_value-low = <custom value>.
  "Example Coding end

  ls_sel_attr_value-unvaluated_row = abap_true.
  APPEND ls_sel_attr_value TO lt_sel_attr_value.
ENDIF.
```
For additional hints see the How to Guide for the UI:
http://scn.sap.com/docs/DOC-30192 Chapter 6.2 Search UI: Hide Classification

10.2.1 Change Search Criteria from MDG7.0 on

With MDG7.0 there are 2 new methods which allow you to change the UI easier with an exit.

METHOD DEFINE_GLOBAL_OPTIONS allows numerous adaptations of the search UI because of DDIC structure WDR_SO_S_GLOBAL_OPTIONS. With an exit you can change for example:

- Number of defaulted rows (maximum number = 4)
- Execution of search after F4 help
- Number of search results

METHOD DEFINE INITIAL_ATTRIBUTES

With an exit you can change the 3 defaulted search criteria (material, description and industry sector).

10.3 Default for class type

Example Coding:

View FIRST_VIEW-> Methode WDDOINIT, Postexit

```lisp
DATA lo_nd_classification TYPE REF TO if_wd_context_node.
DATA lo_el_classification TYPE REF TO if_wd_context_element.
DATA ls_classification TYPE wd_this->element_search_clf.

lo_nd_classification = wd_context->get_child_node( name = wd_this->wdctx_search_clf).
lo_el_classification = lo_nd_classification->get_element( ).
lo_el_classification->set_attribute( EXPORTING
   name = 'KLART'
   value = '<WERT>' ) .
```
10.4 Add field descriptions to the result list

Scenario:
You want to see the field descriptions for the displayed data in the result list.

Solution:
1. Create a post exit for the method CL_MDG_BS_MAT_BO_SEARCH -> GET_SEARCHABLE_ATTRIBUTES. Paste the code from chapter 10.4.1 into the post exit method and activate the code.
2. Create a post exit for the method CL_MDG_BS_MAT_SP_SEARCH -> COMPLETE_DATA. Paste the code from chapter 10.4.2 into the post exit method and activate the code.
3. Invalidate the SPI metadata (Use transaction MDGIMG -> Master Data Governance for Material -> Clear UI Metadata Buffers or execute report MDG_BS_MAT_METADATA_INVALIDATE).
4. Customize the component configuration for your search result list; the field catalog now contains the new text fields.

When you restart the search, the search result list displays texts with the entity fields.

10.4.1 POST-EXIT for method CL_MDG_BS_MAT_BO_SEARCH->GET_SEARCHABLE_ATTRIBUTES

```
DATA ls_component TYPE abap_componentdescr.
DATA ls_entity_attr TYPE usmd_s_entity_attr_prop_ext.
DATA ls_entity_prop TYPE usmd_s_entity_prop_ext.
DATA lv_entity TYPE usmd_entity.
DATA ls_entity_cont TYPE usmd_s_entity_cont.
DATA lv_entity_key TYPE usmd_entity.
DATA lv_add_entity TYPE boole_d.
DATA lt_result_comp TYPE abap_component_tab.
DATA lt_attr_fprop TYPE usmd_ts_fprop_ext.
DATA lt_txt_fprop TYPE usmd_ts_fprop_ext.
DATA lo_text TYPE REF TO cl_mdg_bs_mat_text.
DATA ls_fprop TYPE usmd_s_fprop_ext.
DATA lv_field TYPE typename.
DATA lv_field_id TYPE typename.
DATA lt_text_field_id TYPE mdg_bs_mat_t_typename.
DATA lv_text_field_id TYPE typename.
DATA lv_text_field TYPE typename.

FIELD-SYMBOLS <fs_fdep> TYPE usmd_s_fdep.
FIELD-SYMBOLS <fs_fprop> TYPE usmd_s_fprop_ext.

lo_text = cl_mdg_bs_mat_text=>get_instance(
    iv_appl = cl_mdg_bs_mat_c=>gc_buffer_appl).
```

* Get properties of main entity (MATERIAL)
READ TABLE cl_mdg_bs_mat_bo_search=>so_model=>dt_entity_prop
INTO ls_entity_prop
WITH KEY usmd_entity = if_mdg_bs_mat_gen_c=>gc_entity_material.

IF sy-subrc = 0.
* Get properties of main entity (DDIC info)
LOOP AT cl_mdg_bs_mat_bo_search=>so_model->dt_fdep ASSIGNING <fs_fdep>
  WHERE fieldname = ls_entity_prop-r_fprop=>fieldname.
  READ TABLE cl_mdg_bs_mat_bo_search=>so_model->dt_fprop
    ASSIGNING <fs_fprop>
    WITH KEY fieldname = <fs_fdep>-masterfield.
  lv_entity = <fs_fprop>-usmd_entity.
  ls_component-name = <fs_fprop>-fieldname.
  ls_component-type ?= cl_abap_typedescr=>describe_by_name({
    <fs_fprop>-rollname }).
  INSERT ls_component INTO TABLE lt_result_comp.
ENDLOOP.

cl_mdg_bs_mat_smt=>find_prop_entity{
  EXPORTING
    iv_respect_switch = abap_false
    iv_entity = if_mdg_bs_mat_gen_c=>gc_entity_material
  IMPORTING
    et_attr_fprop = lt_attr_fprop
    et_txt_fprop = lt_txt_fprop ).

* RESPONSE attributes *******************************
* Get all attributes of main entity
LOOP AT cl_mdg_bs_mat_bo_search=>so_model->dt_entity_attr_prop
  INTO ls_entity_attr
  WHERE usmd_entity = ls_entity_prop-usmd_entity AND
    f_read_only IS INITIAL AND
    f_direct_attr = abap_true.
* Attribute customized as response field?
  READ TABLE cl_mdg_bs_mat_bo_search=>so_model->dt_fld_uiprop
    TRANSPORTING NO FIELDS "VC_USMD006"
    WITH KEY usmd_entity = ls_entity_attr-usmd_entity
      fieldname = ls_entity_attr-r_fprop=>fieldname
      no_result_list = abap_false.
  IF sy-subrc = 0.
    CLEAR ls_component.
    ls_component-name = ls_entity_attr-r_fprop=>fieldname.
    ls_component-type ?= cl_abap_typedescr=>describe_by_name({
      ls_entity_attr-r_fprop=>rollname }).
    INSERT ls_component INTO TABLE lt_result_comp.
  *
  *++++++ get assigned text fields ++++++++++++++++++++++++++++++
  READ TABLE lt_attr_fprop INTO ls_fprop
    WITH KEY fieldname = ls_component-name.
  IF sy-subrc = 0.
    lv_field = ls_fprop-rolname.
    lv_field_id = ls_fprop-fieldname.
lt_text_field_id = lo_text->get_text_field_id(
    lv_field = lv_field
    lv_field_id = lv_field_id).

LOOP AT lt_text_field_id INTO lv_text_field_id.
  *
  Get data element (rollname) for given text field ID
  lv_text_field = lo_text->get_text_field( lv_text_field_id ).
  IF lv_text_field IS INITIAL.
    CONTINUE.
  ENDIF.
  READ TABLE lt_result_comp TRANSPORTING NO FIELDS
    WITH KEY name = lv_text_field_id.
  IF sy-subrc <> 0.
    ls_component-name = lv_text_field_id.
    ls_component-type ?= cl_abap_typedescr->describe_by_name(  
      lv_text_field ).
    INSERT ls_component INTO TABLE lt_result_comp.
  ENDIF.
ENDLOOP.
ENDIF.

* Get extended response attributes
  LOOP AT cl_mdg_bs_mat_bo_search->so_model->dt_entity_cont
    INTO ls_entity_cont
    WHERE usmd_entity = ls_entity_prop-usmd_entity.
  lv_add_entity = abap_false.
  *
  Check key field of dependent entities -> only 1:1 relation relevant
  IF ls_entity_cont-t_entity_key IS INITIAL.
    lv_add_entity = abap_true.
  ELSEIF lines( ls_entity_cont-t_entity_key ) = 1.
    READ TABLE ls_entity_cont-t_entity_key INTO lv_entity_key INDEX 1.
    IF lv_entity_key = if_mdg_bs_mat_gen_c->gc_entity_material.
      lv_add_entity = abap_true.
  ENDIF.
ENDIF.
IF lv_add_entity = abap_true.
  cl_mdg_bs_mat_smt->find_prop_entity(
    EXPORTING
      iv_respect_switch = abap_false
      iv_entity = ls_entity_cont-usmd_entity_cont
    IMPORTING
      et_attr_fprop = lt_attr_fprop
      et_txt_fprop = lt_txt_fprop ).
  *
  Add all entity attributes to response attribute list
  LOOP AT cl_mdg_bs_mat_bo_search->so_model->dt_entity_attr_prop
    INTO ls_entity_attr
    WHERE usmd_entity = ls_entity_cont-usmd_entity_cont AND
      f_read_only IS INITIAL AND
      f_direct_attr = abap_true.
  *
  Attribute already contained in response attribute list?
READ TABLE lt_result_comp TRANSPORTING NO FIELDS 
   WITH KEY name = ls_entity_attr-r_fprop->fieldname.
IF sy-subrc = 0.
   CONTINUE.
ENDIF.
* Attribute customized as response field?
READ TABLE cl_mdg_bs_mat_bo_search->so_model->dt_fld_uiprop 
   TRANSPORTING NO FIELDS "VC_USMD006"
   WITH KEY usmd_entity = ls_entity_attr-usmd_entity
   fieldname = ls_entity_attr-r_fprop->fieldname
   no_result_list = abap_false.
IF sy-subrc = 0.
   CLEAR ls_component.
   ls_component-name = ls_entity_attr-r_fprop->fieldname.
   ls_component-type != cl_abap_typedescr->describe_by_name( ls_entity_attr-r_fprop->rollname ).
   INSERT ls_component INTO TABLE lt_result_comp.
* 
* +++++++++++++++++++ get assigned text fields ++++++++++++++++++++++++ 
READ TABLE lt_attr_fprop INTO ls_fprop
   WITH KEY fieldname = ls_component-name.
IF sy-subrc = 0.
   lv_field = ls_fprop-rollname.
   lv_field_id = ls_fprop-fieldname.
   lt_text_field_id = lo_text->get_text_field_id( 
      iv_field = lv_field
      iv_field_id = lv_field_id ) .
   LOOP AT lt_text_field_id INTO lv_text_field_id.
   * Get data element (rollname) for given text field ID
   lv_text_field = lo_text->get_text_field( lv_text_field_id ).
   IF lv_text_field IS INITIAL.
      CONTINUE.
   ENDIF.
   READ TABLE lt_result_comp TRANSPORTING NO FIELDS 
      WITH KEY name = lv_text_field_id.
   IF sy-subrc <> 0.
      ls_component-name = lv_text_field_id.
      ls_component-type != cl_abap_typedescr->describe_by_name( lv_text_field ).
      INSERT ls_component INTO TABLE lt_result_comp.
   ENDIF.
   ENDLOOP.
   ENDIF.
*
SORT lt_result_comp BY name.
et_result_list_attributes = lt_result_comp.

10.4.2 POST-EXIT for method CL_MDG_BS_MAT_SP_SEARCH-
>COMPLETE_DATA

DATA lo_struc TYPE REF TO cl_abap_structdescr.
DATA lt_comp TYPE abap_compdescr_tab.
DATA lo_text TYPE REF TO cl_mdg_bs_mat_text.

FIELD-SYMBOLS <ls_data> TYPE mdg_bs_mat_s_sp_search_data.
FIELD-SYMBOLS <ls_data_mat> TYPE any.
FIELD-SYMBOLS <ls_comp> TYPE abap_compdescr.
FIELD-SYMBOLS <lv_txt_field> TYPE any.

lo_text = cl_mdg_bs_mat_text=>get_instance{
    iv_appl = cl_mdg_bs_mat_c=>gc_buffer_appl }.

* add text field values
LOOP AT ct_data ASSIGNING <ls_data>.
    ASSIGN <ls_data>-s_data>* TO <ls_data_mat>.
    IF sy-tabix = 1.
        lo_struc ?= cl_abap_structdescr=>describe_by_data( <ls_data_mat> ).
        LOOP AT lo_struc->components ASSIGNING <ls_comp>
            WHERE name CS '__TXT'.
            INSERT <ls_comp> INTO TABLE lt_comp.
        ENDLOOP.
    ENDIF.
    ENDLOOP.
LOOP AT lt_comp ASSIGNING <ls_comp>.
    ASSIGN COMPONENT <ls_comp>-name
        OF STRUCTURE <ls_data_mat> TO <lv_txt_field>.
    lo_text=>get_text_field_value( EXPORTING
        is_field_value = <ls_data_mat>
        iv_text_field_id = <ls_comp>-name
    IMPORTING
        ev_text_field_value = <lv_txt_field> ).
ENDLOOP.
ENDLOOP.
11. Multiple Languages

Search is performed always on all languages. The response returns texts in the log-on language in the first place. If no description is available in that language, the effective language of the text is determined by a language vector which is defined in customizing:

SAP Customizing Implementation Guide > SAP NetWeaver > Search and Operational Analytics > SAP NetWeaver Enterprise Search/Presentation of Objects > Define Preferred Language Sequence for the Presentation of Objects

Example:

With Customizing: DE Language 1, EN Language 2
Material text maintained in DE original and EN (translation)
Search:
- In EN you get the English text
- In DE you get the German text
- In FR you get the German text

Without Customizing:
Material text maintained in DE original and EN (translation)
- In EN you get the English text
- In DE you get the German text
- In FR you get the English text

Material text maintained in EN original and DE (translation)
- In EN you get the English text
- In DE you get the German text
- In FR you get the English text

Without customizing the strategy is that the system uses the system language not the original language.