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 by using BAdI MDG_BS_MAT_API_SEGMENTS_EXT.

Created on: April 2015

Version: 6.0

© 2012 SAP AG

Table of Contents

Applies to: 1
Summary 1
Table of Contents 2
1. Introduction 3
2. Prerequisites 3
3. Scenario 3
3.1. High Level Requirements 3
3.2. Governance Process 4
4. Introduction 4
4.1. Data Modeling 4
4.1.1. Basic Data Model for Material 4
4.1.2. Data Modeling Concepts in MDG 5
4.2. User Interface Configuration 13
4.2.1. Adaptation Options in Floorplan Manager 14
4.2.2. Handling Previous Enhancements Using the “UI BAdI” 14
4.2.3. Removing Customizing or Personalization 15
5. Implementation 16
5.1. Prepare MDG-M API 16
5.1.1. Create DDIC Objects 16
5.1.2. Extend MDG BS MAT (Material Master Data Structure) 22
5.1.3. BAdI: Extension of the API with Customer-Specific Segments 24
5.2. MDG Data Model Extension 29
5.2.1. Extend MDG Data Model 30
5.2.2. Generate Model-Specific Structures 37
5.2.1. Clear UI Metadata Buffers 39
5.3. Create SMT-Mapping 39
5.3.1. Create Mapping Entries in Customizing 40
5.3.2. Map the Active Area to the Staging Area 41
5.3.3. Map the Staging Area to the Active Area 44
5.4. Adjust Staging Area of Linked Change Requests 48
5.5. Extend User Interface 49
5.5.1. Create FPM List UIBB 49
5.5.2. Add List UIBB to Material UI 57
5.5.3. Clear UI Metadata Buffers 61
6. Testing Your Data Model Extension 61
7. Related Content 62
8. Copyright 63

© 2012 SAP AG

1. 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 by using BAdI MDG_BS_MAT_API_SEGMENTS_EXT. The attribute values of the new entity type will be copied to the corresponding customer tables (reuse option) after activation of the change request.

2. 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)

3. Scenario

3.1. 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.

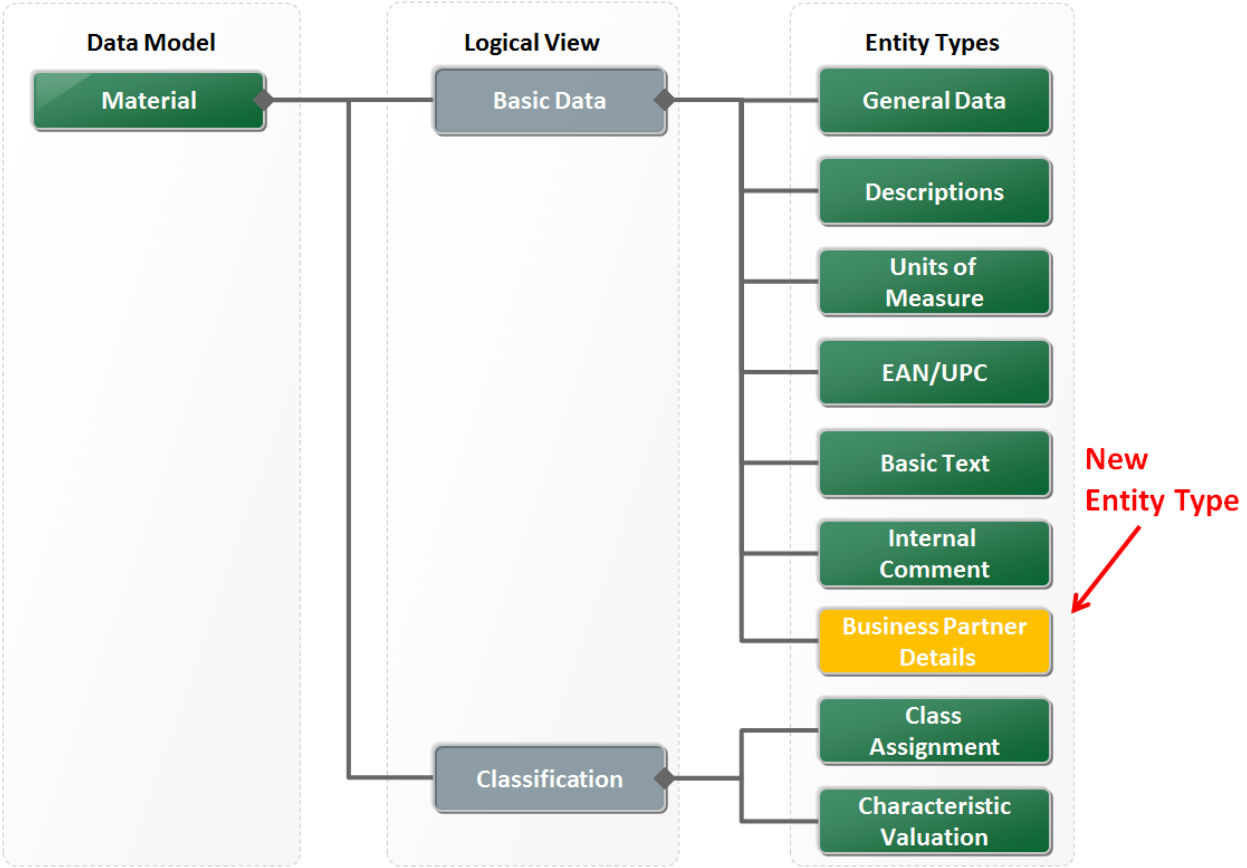


Figure: Data Model – Material (Scope of 2011 Delivery) with custom entity type "Business Partner Details"

© 2012 SAP AG

3.2. 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.

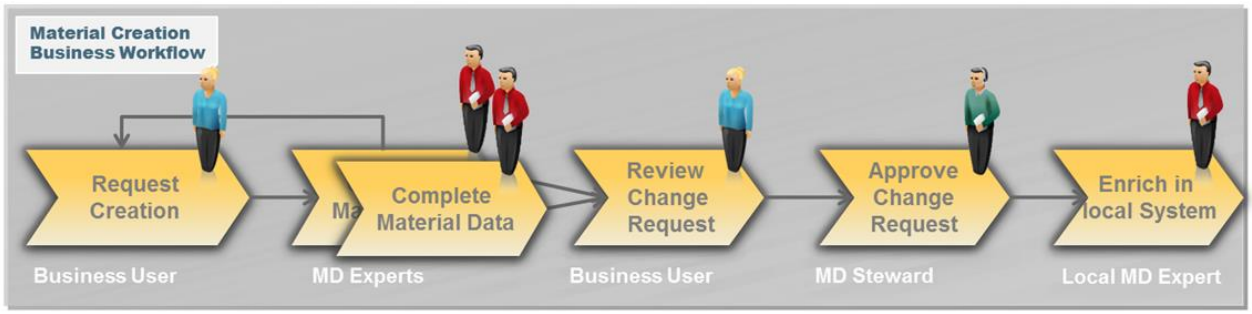


Figure: Material Processing in Master Data Governance

4. 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.

4.1. 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.

4.1.1. 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

© 2012 SAP AG

Additional Information:

- A BAdI is available for data enhancement during change request activation (MDG_BS_MAT_API_ENRICH)
- Authorization Concept: Depend on the reuse of backend logic and pre-delivered roles defined in PFCG
- Field control: visibility and mandatory fields are controlled with the field control feature that re-uses the backend logic and existing settings (T130F).

4.1.2. 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.

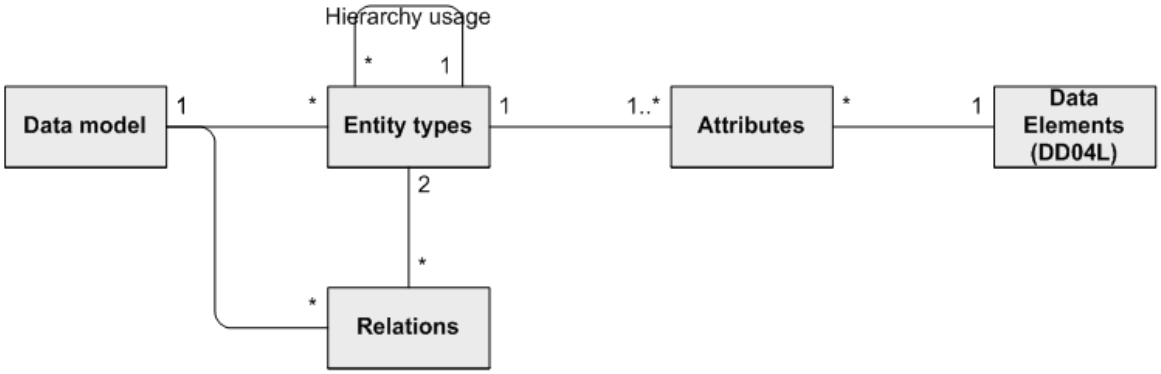
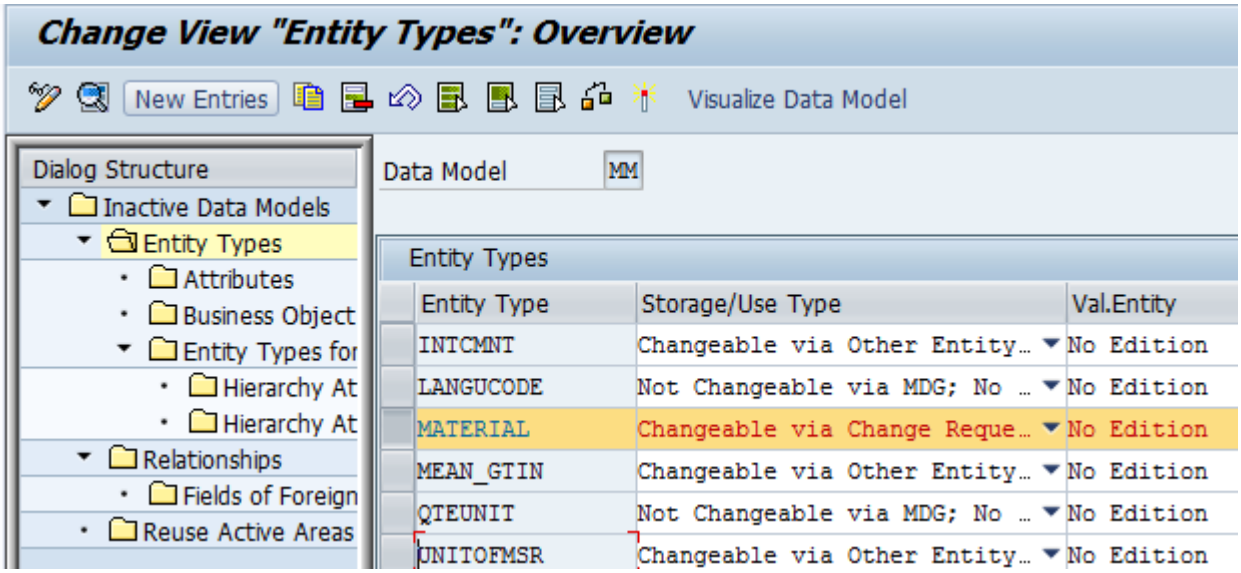


Figure: The meta model is an entity-relationship-model (ERM)

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`**.



© 2012 SAP AG

4.1.2.1. Storage and Use Types

You assign 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.

© 2012 SAP AG

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**.

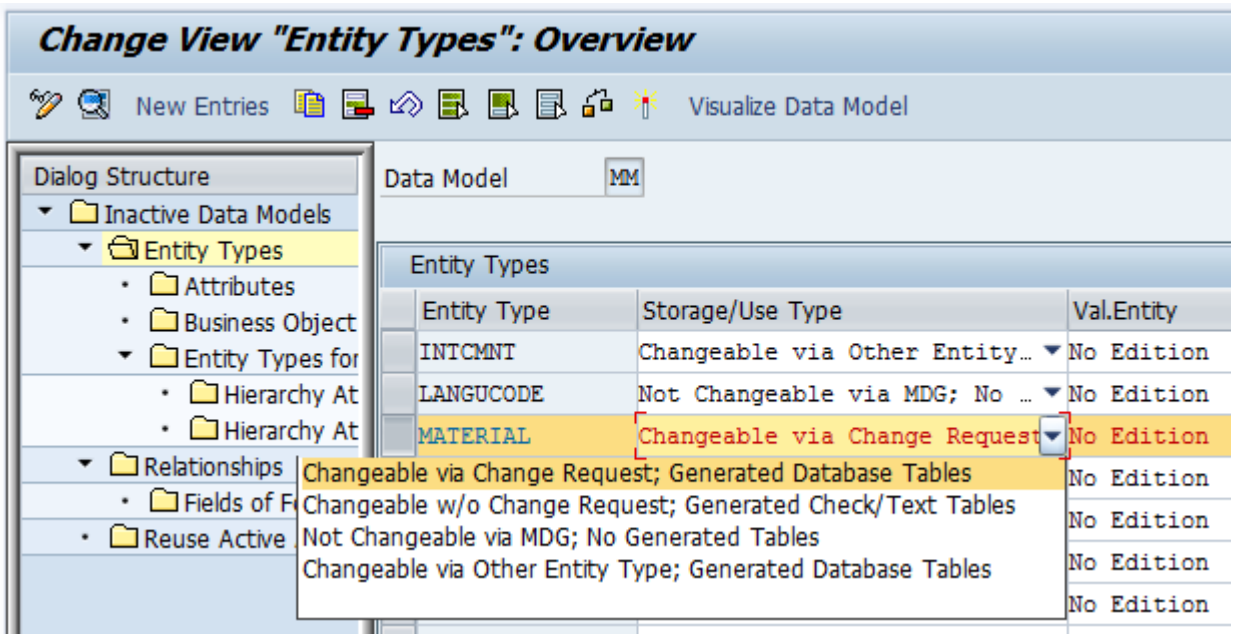


Figure: Selection box showing the different storage/use types in MDG

The table below gives you some criterions for the decision between storage and use type 1 and 4.

Criterion	Type1	Type4
Cross-dependencies	Need to handle dependencies between Type1 entities (include into CR, blocking, …)	Easy to implement
Parallel CRs	Not needed	Necessary if two Type4 entities for same Type1 entity shall be maintained independently
Snapshot	Independent	Together with corresponding Type1 entity and all other Type4 entities (→ Performance)
Enqueue („Sperrschatten“)	This entity plus corresponding Type4 entities	This entity, corresponding parent entity and all its child entities
Authorization	Own authorization objects	Depends on authorization object of Type1 entity (plus maybe additional own ones)
Implementation effort	Own access class	MDG_BS_MAT_API_SEGMENTS_EXT BAdI (lean, no CD mapping, enqueue, Query, …) or own access class
Mass Maintenance	OK	Not possible (exception: 1:1 Relationship)

© 2012 SAP AG

Multi-Record Processing	No difference	
Use case	• Production Version (MKAL) • Document Assignment (DRAD)	The rest of the material entities (for example MARC or QMAT)

Note

It's usually not meaningful to define several Type1 entities with the same key (example: MARASALES/-PURCH).

Important

For more complex requirements like search, mapping, change documents, field properties, authorization, locks or derives, you need to provide an own access class by implementing interface if_usmd_pp_access instead of using MDG_BS_MAT_API_SEGMENTS_EXT BAdI. This guide only describes the usage of the BAdI MDG_BS_MAT_API_SEGMENTS_EXT.

4.1.2.2. 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.

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

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

Cardinality	Definition
1:N	This cardinality represents a mandatory relationship in which one or more *To-Entity Types* can be assigned to a *From-Entity Type*. This cardinality is valid for relationships with the relationship types *Leading*, *Qualifying*, and *Referencing*. In relationships with the relationship type *referencing*, the *From-Entity Type* is a required attribute of the *To-Entity Type*.
0:N	This cardinality represents an optional relationship in which any number *To-Entity Types* can be assigned to a *From-Entity Type*.

© 2012 SAP AG

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.

4.1.2.3. 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.

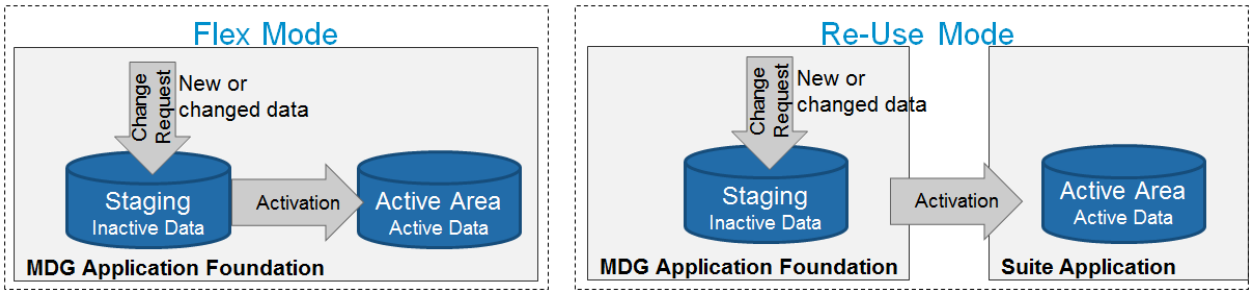


Figure: Flexibility Option (left) versus Reuse Option (right)

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.

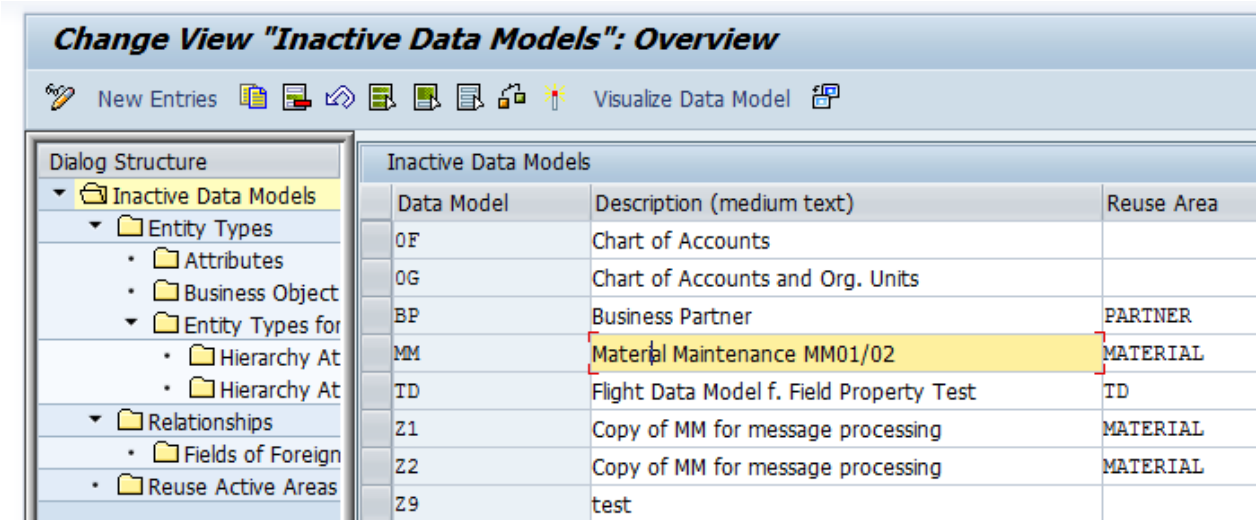


*Figure: Assignment of Reuse Area for the Data Model **MM***

© 2012 SAP AG

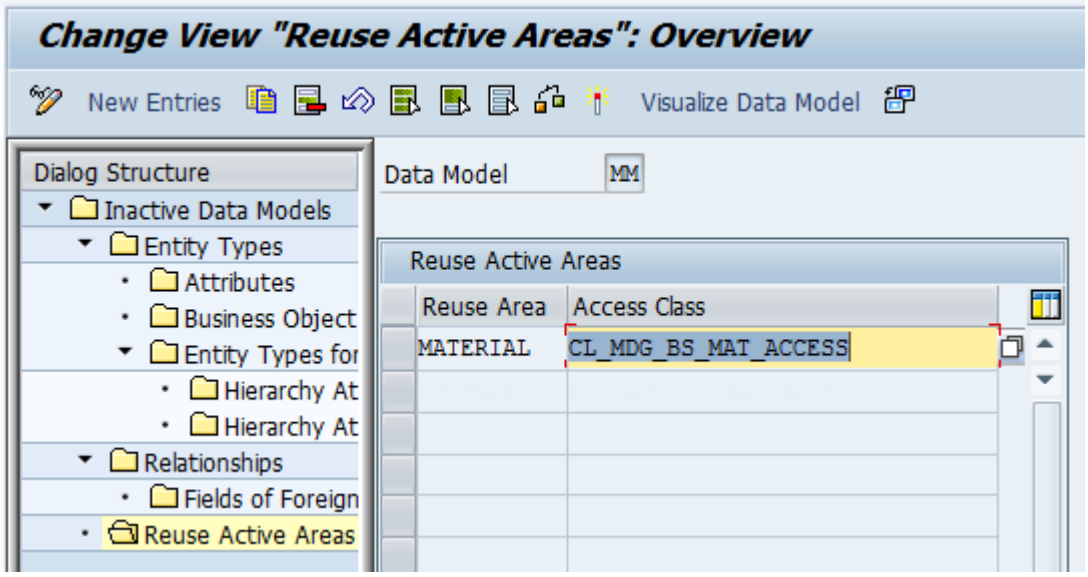


Figure: Assignment of Access Class for Reuse Area ***MATERIAL***

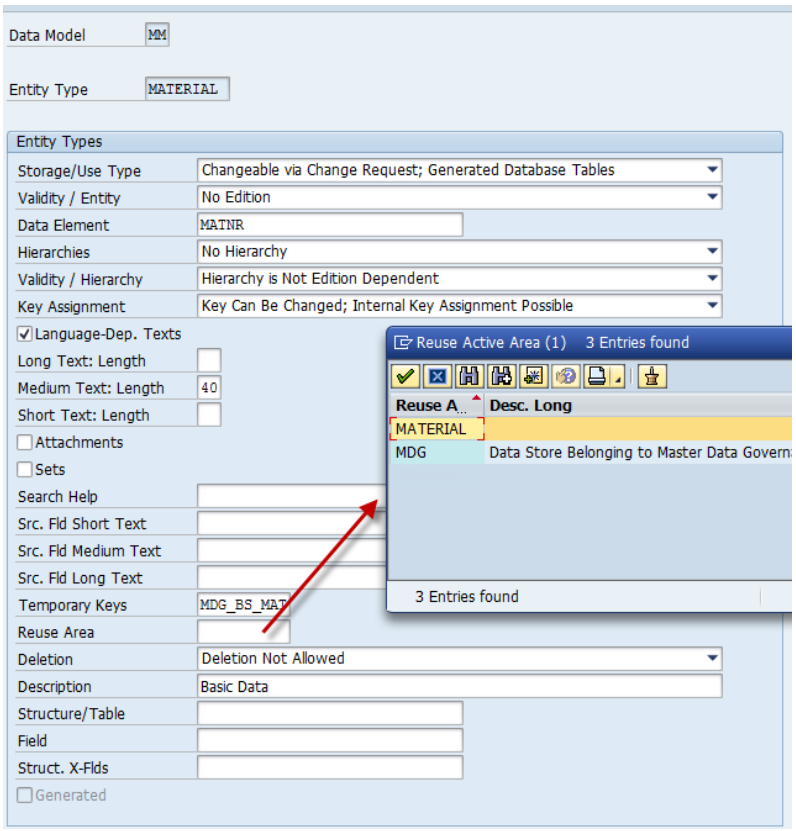


Figure: Alternative assignment of Reuse Area on Entity Type level

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.

The table below gives you some criterions for the decision between flex or reuse mode.

© 2012 SAP AG

Criterion	Flex	Reuse (Z table)
Direct Update of Active Area (for migration, …)	Not possible	Possible and used in IDoc/DIF/LSMW/API/…
DIF usage	Not recommended for MDG-M (can't write to active area)	OK
SOA	Not recommended for MDG-M (can't write to active area)	OK
File Upload	No difference (updates only staging)	
Access/usage in productive applications (in co-deployment) like • Search • Typed read access • …	Dynamic tables, access only via IF_USMD_MODEL_EXT-> READ_CHAR_VALUE (or complex select), no indices on DB level	Can be tuned via indices, view definition possible and transportable, …
Mass Maintenance/Multi-Record Processing	No difference	
Performance in maintenance	• Flex entities need no snapshot (and snapshot comparison) • Number of records in active area has impact on performance	• Reuse entities need snapshot • Number of records in reuse active area has no impact on performance
DRF usage	No difference	
Project effort	No additional coding required (plus optional additional checks, feeder class, …)	Coding needed – either implementation of BAdI MDG_BS_MAT_API_SEGMENTS_EXT or specific access class
Use case	Co-Deployment: Stays in MDG, no need to access data operatively	Co-Deployment: Needs to be accessed by operative processes. Both deployments: Take existing data/extensions under governance Both deployments: Data import must be done without CR for performance reasons (or no governance for migration/import needed)

© 2012 SAP AG

4.1.2.4. 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

4.1.2.5. 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.

4.1.2.6. 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.

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

Example Implementation of Independent and Interdependent UIBBs

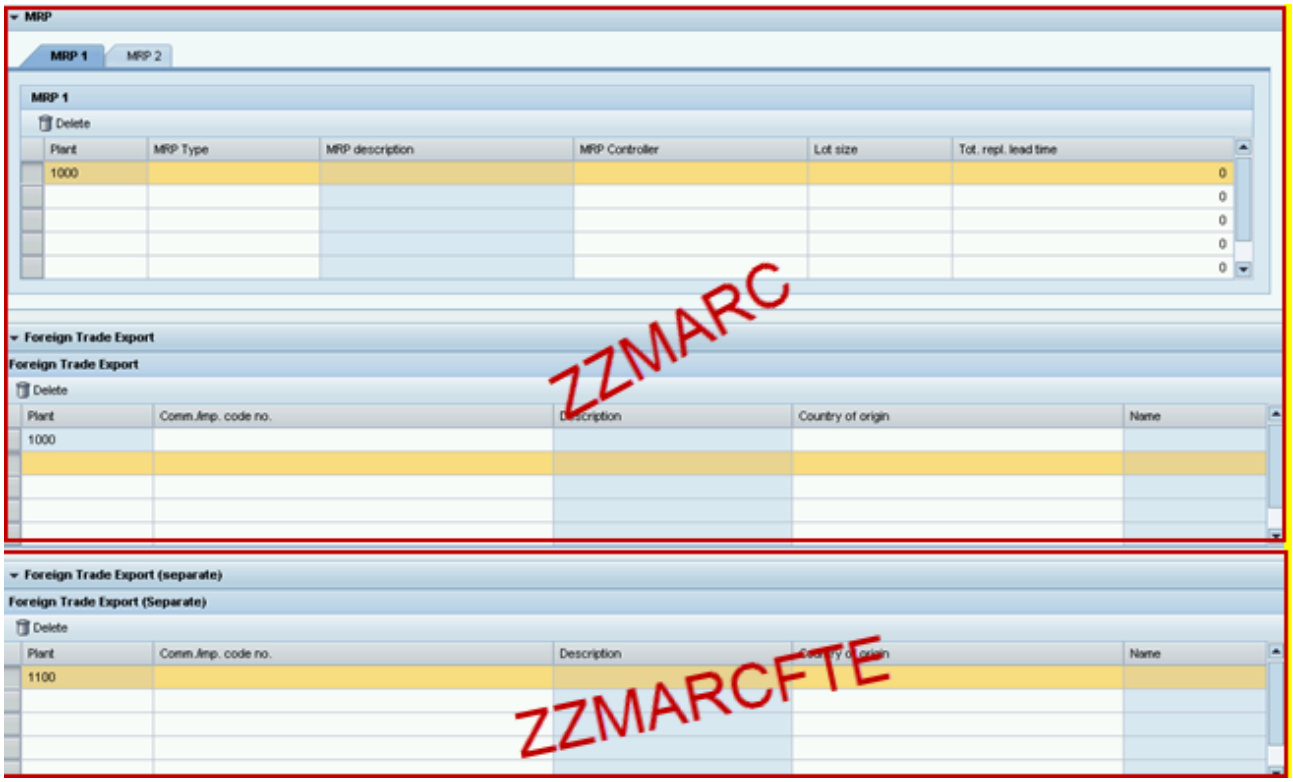
You have created ***interdependent*** list UIBBs for the following user interfaces using the **`ZZMARC`** entity type:

- **`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.

© 2012 SAP AG



4.2. 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:

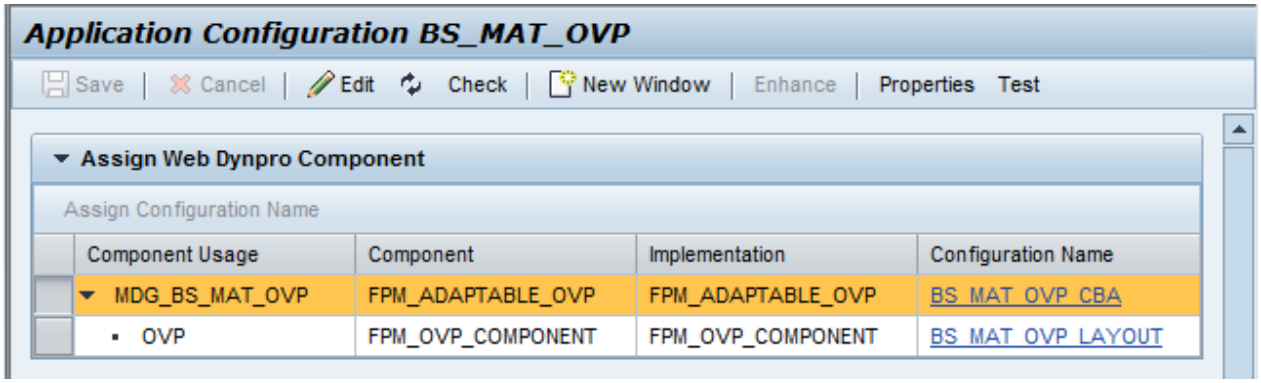
- **Web Dynpro ABAP Application:** Contains the information about the main component and window of the application
- **Web Dynpro ABAP Application Configuration:** Contains the information about the configuration used for starting the main component

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`**

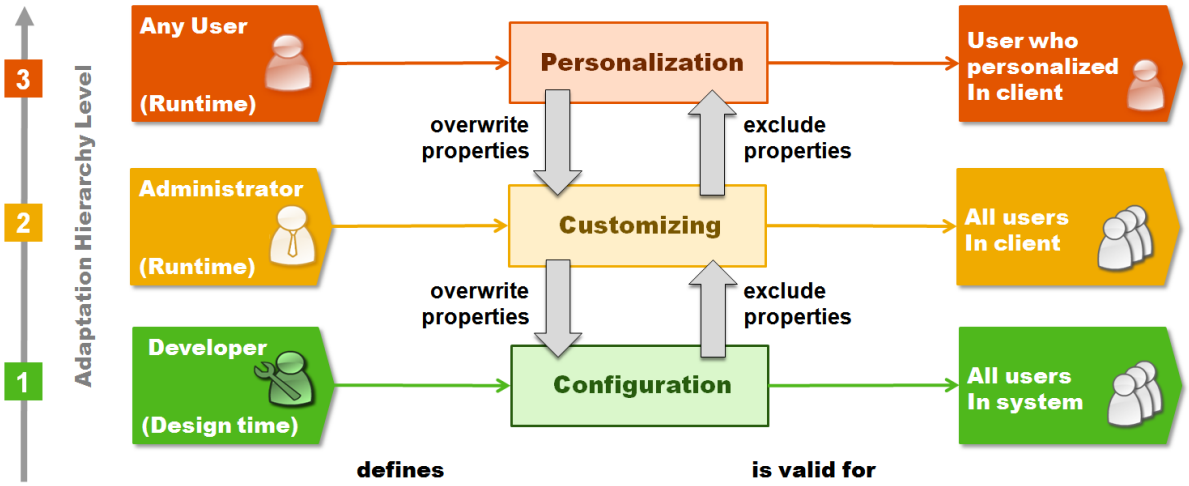
In the screenshot below application **`MDG_BS_MAT_OVP`** is using component **`FPM_ADAPTABLE_OVP`** with configuration **`BS_MAT_OVP_CBA`** to provide the context-based adaptation (CBA) functionality. The layout is provided by **`FPM_OVP_COMPONENT`** in combination with configuration **`BS_MAT_OVP_LAYOUT`**.

© 2012 SAP AG



4.2.1. 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

4.2.2. 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:

© 2012 SAP AG

- 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.

4.2.3.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`**

© 2012 SAP AG

5. 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.

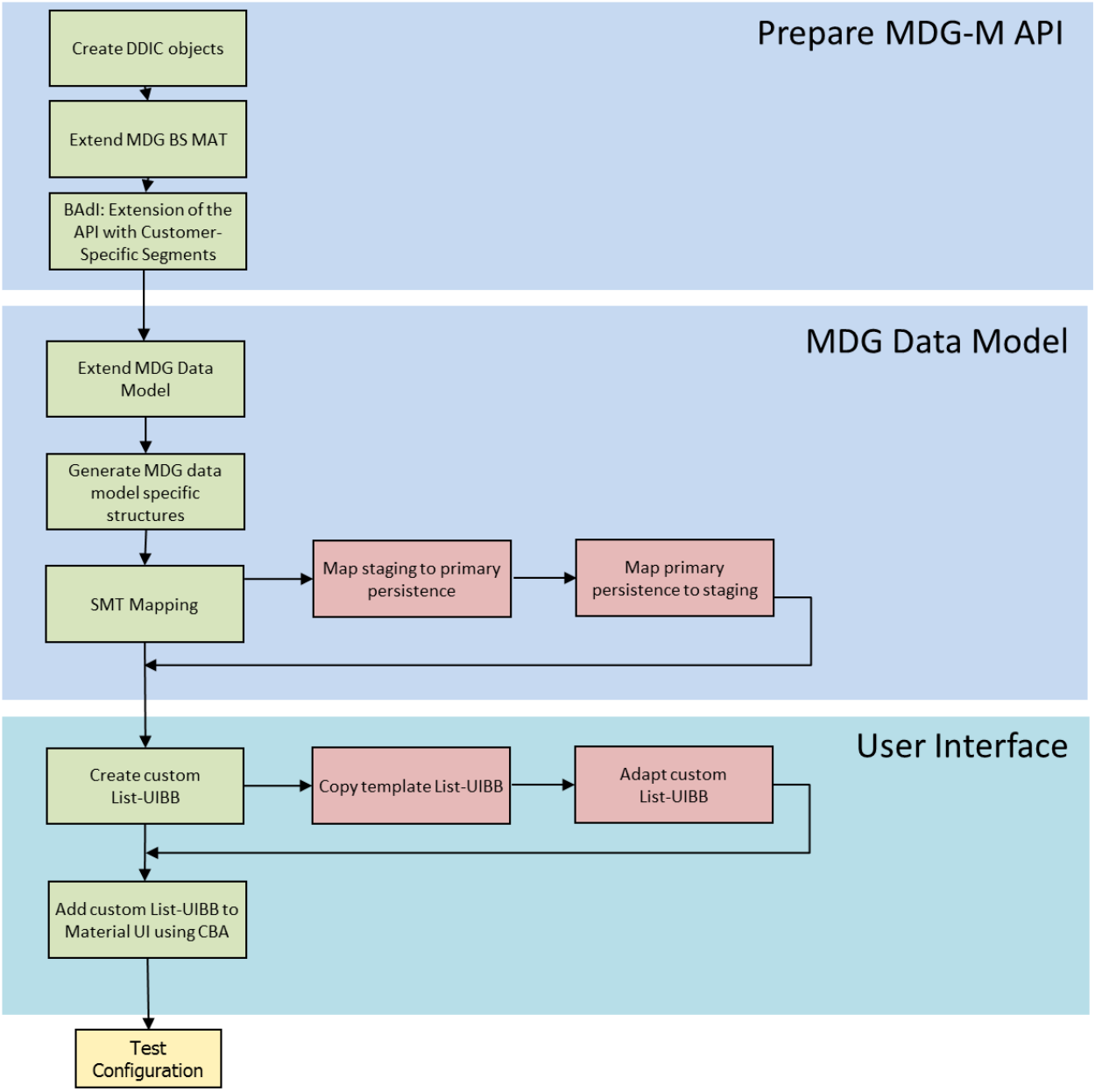


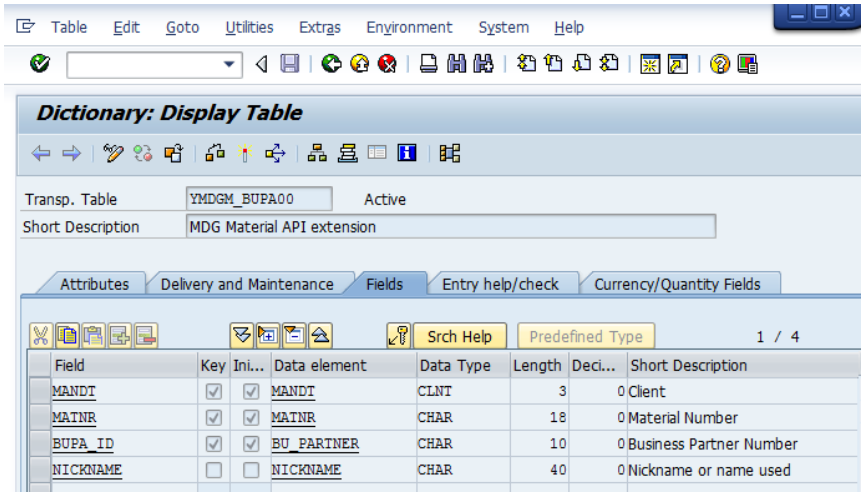
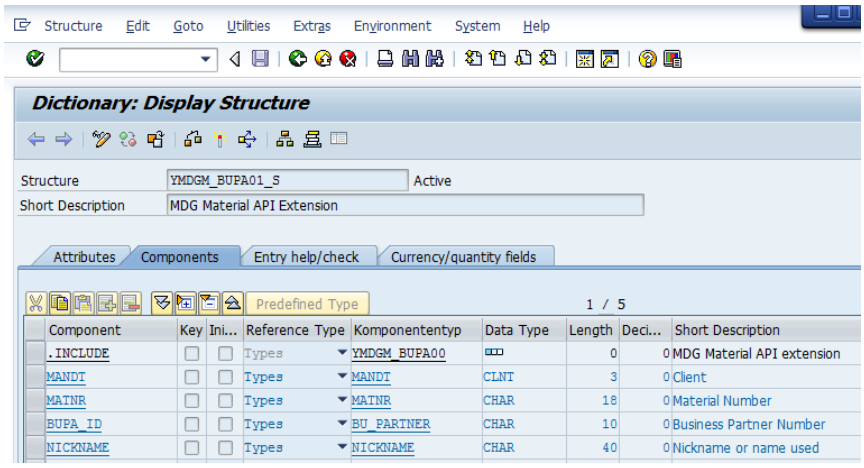
Figure: Implementation steps for re-use Entity-Type extension

5.1. Prepare MDG-M API

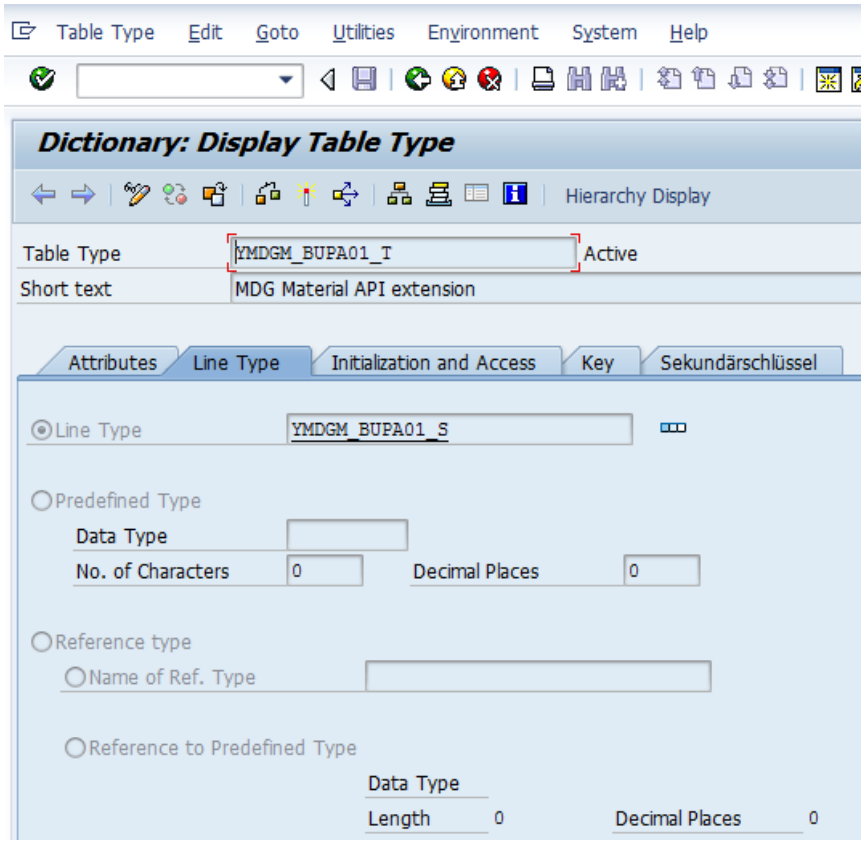
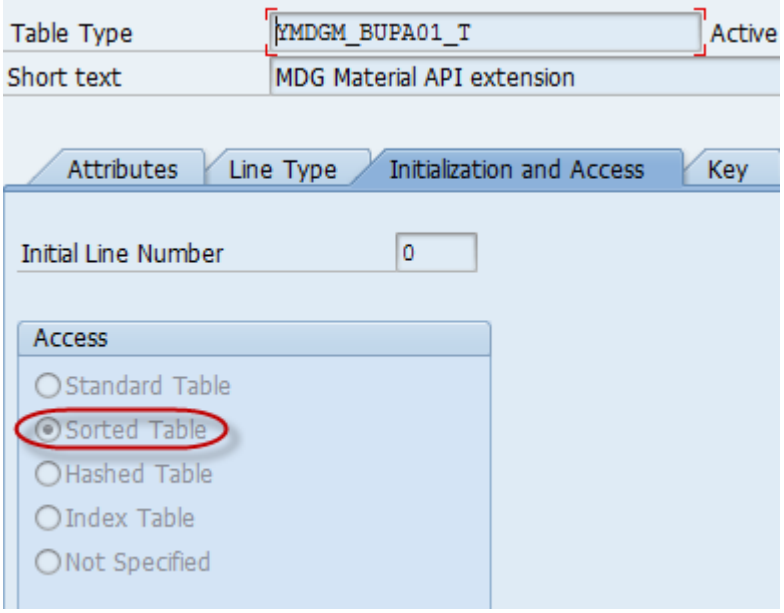
5.1.1. Create DDIC Objects

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.

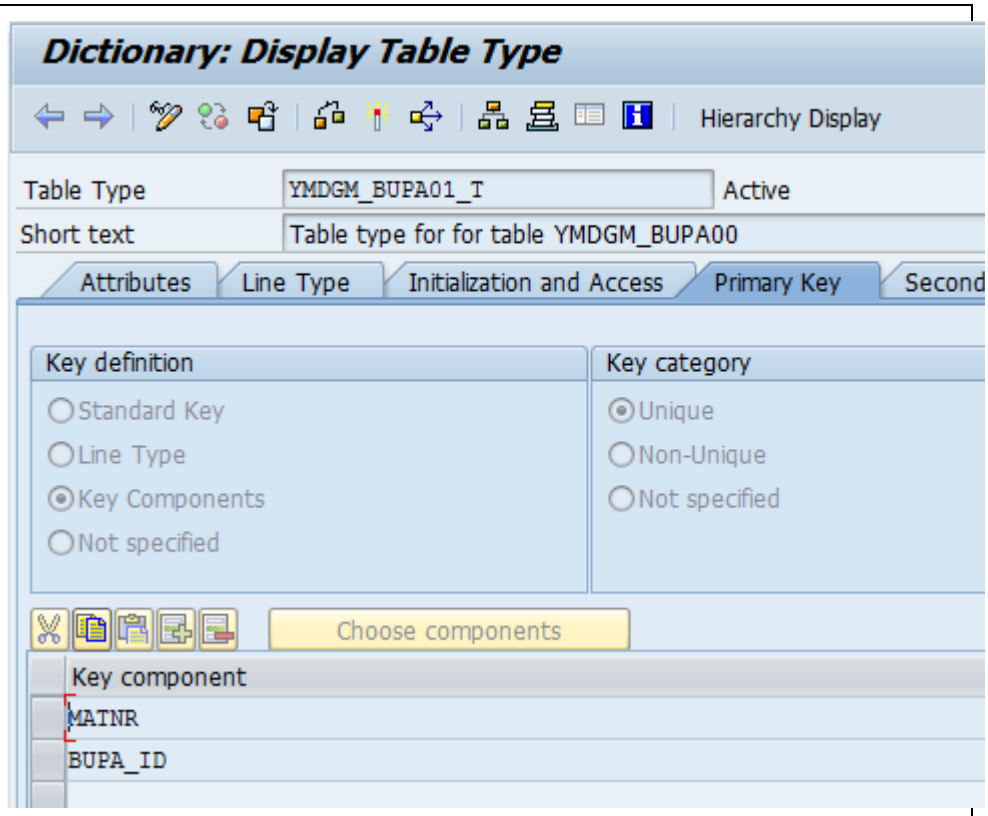
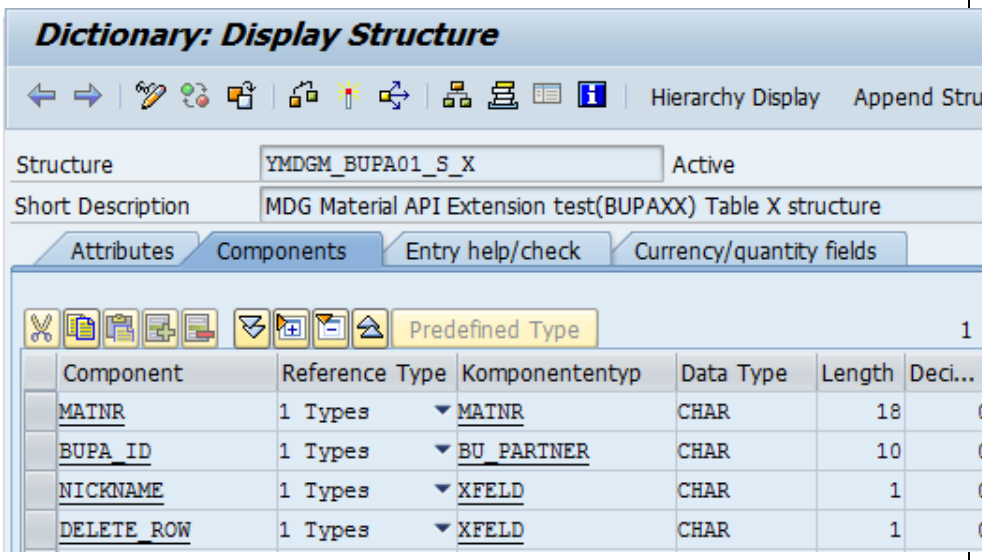
© 2012 SAP AG

1.	In transaction **SE11** create table **YMDGM_BUPA00** as shown.	
2.	In transaction **SE11** create structure **YMDGM_BUPA01_S** as shown.	

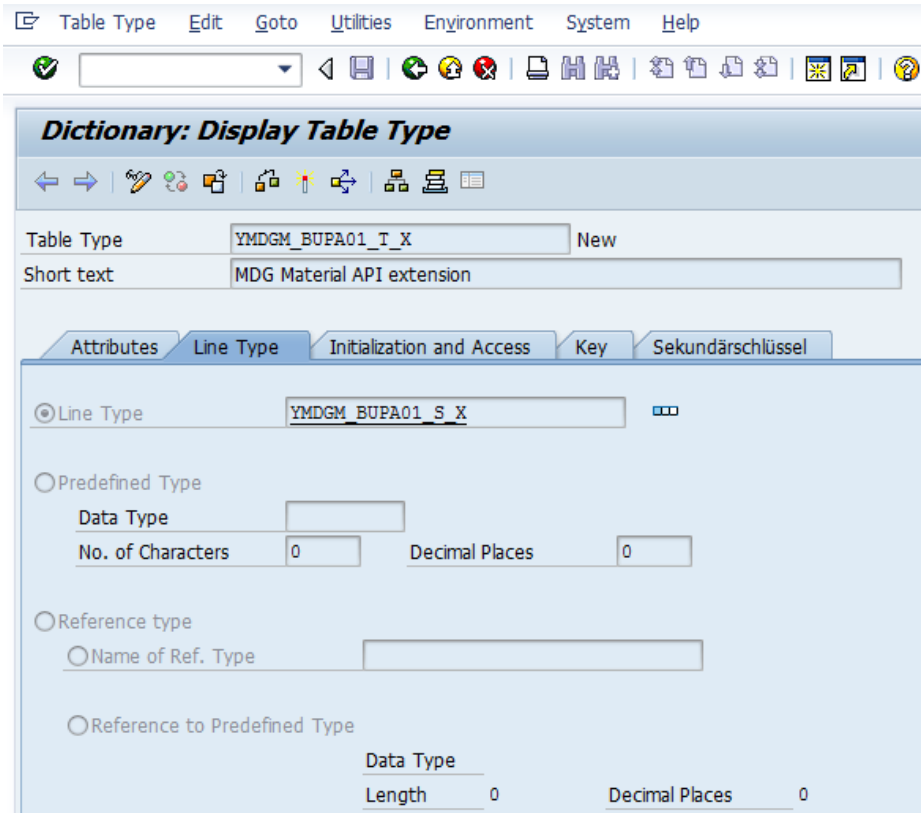
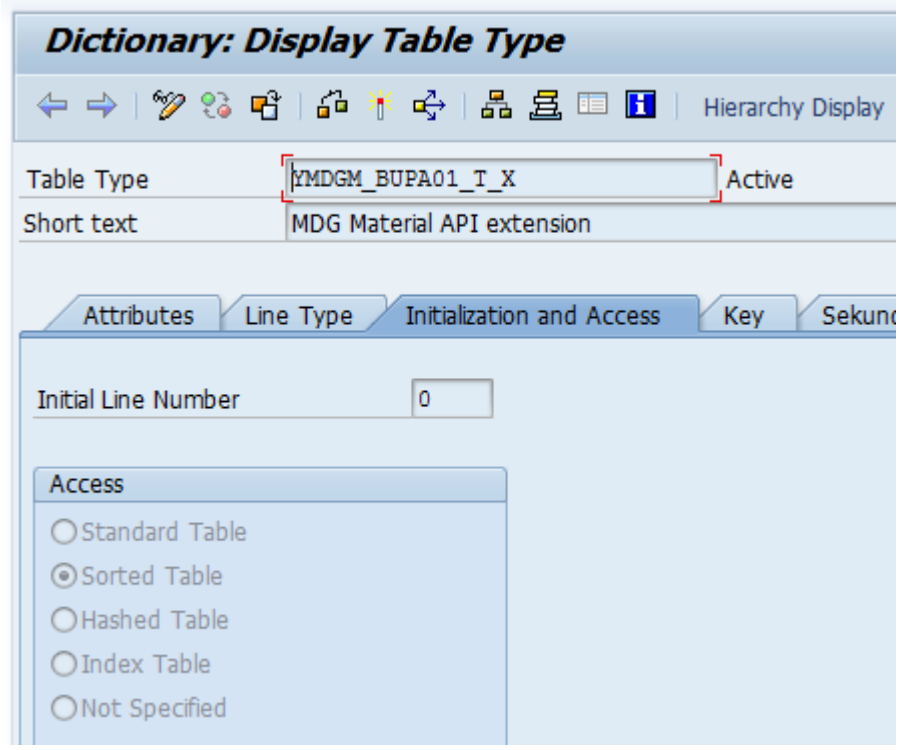
© 2012 SAP AG

3.	In transaction **SE11** create table type **YMDGM_BUPA01_T** as shown.	
4.	Make sure table type **YMDGM_BUPA01_T** is a ***Sorted Table***.	

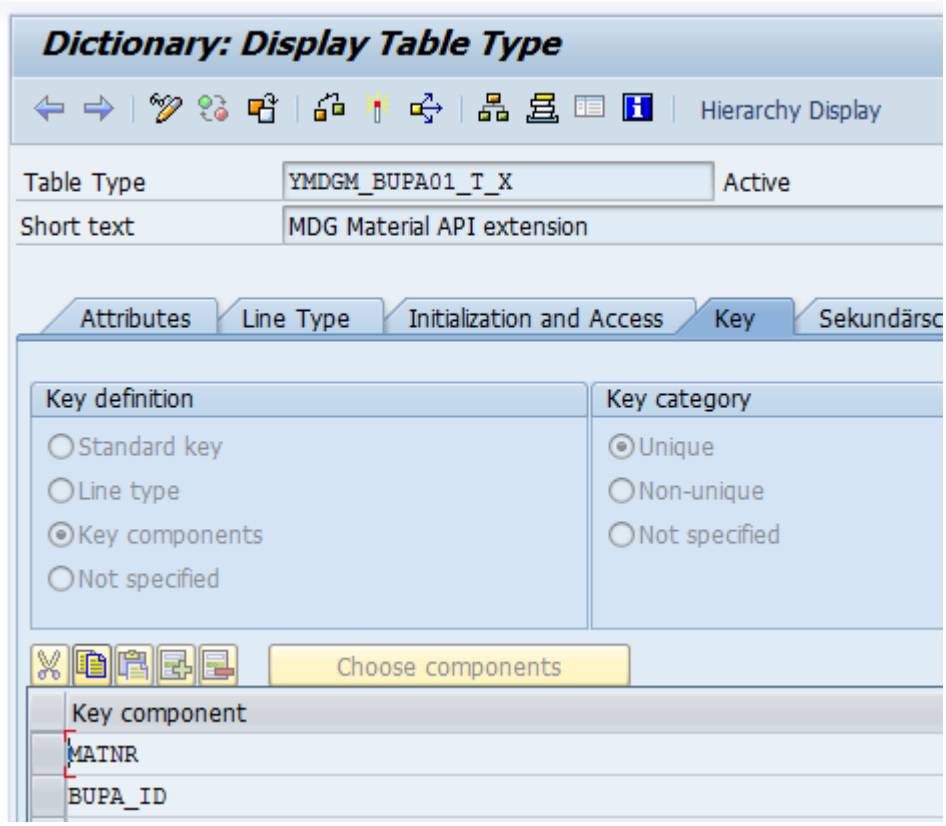
© 2012 SAP AG

5.	Make sure table type **YMDGM_BUPA01_T** uses the keys as shown.	
6.	In transaction **SE11** create structure **YMDGM_BUPA01_S_X** as shown. Note: DELETE_ROW indicates that this row shall be deleted	

© 2012 SAP AG

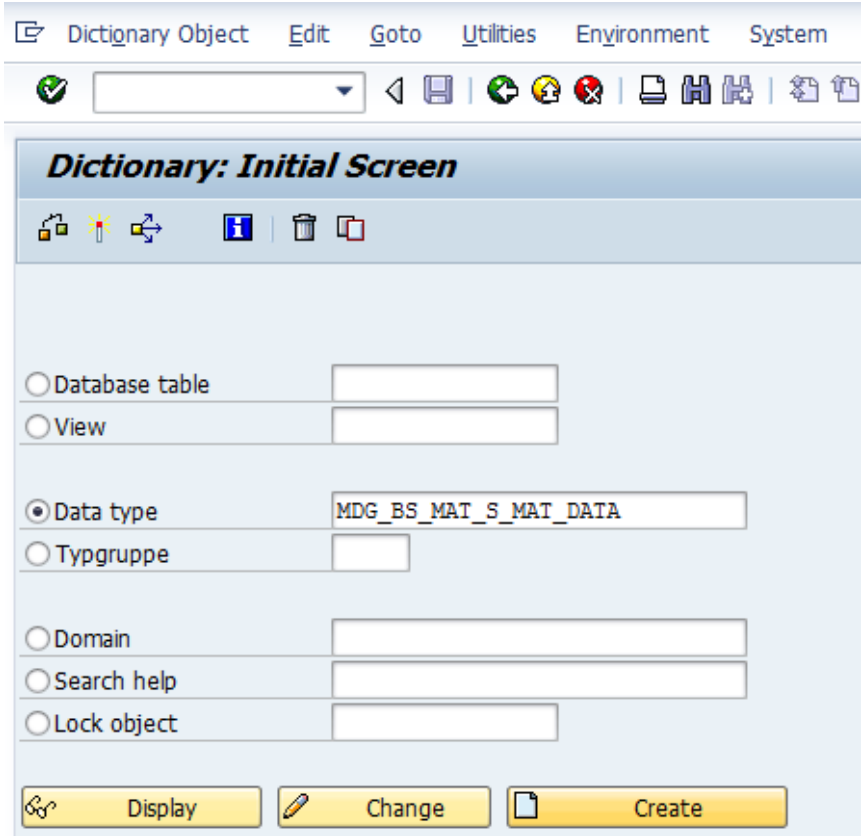
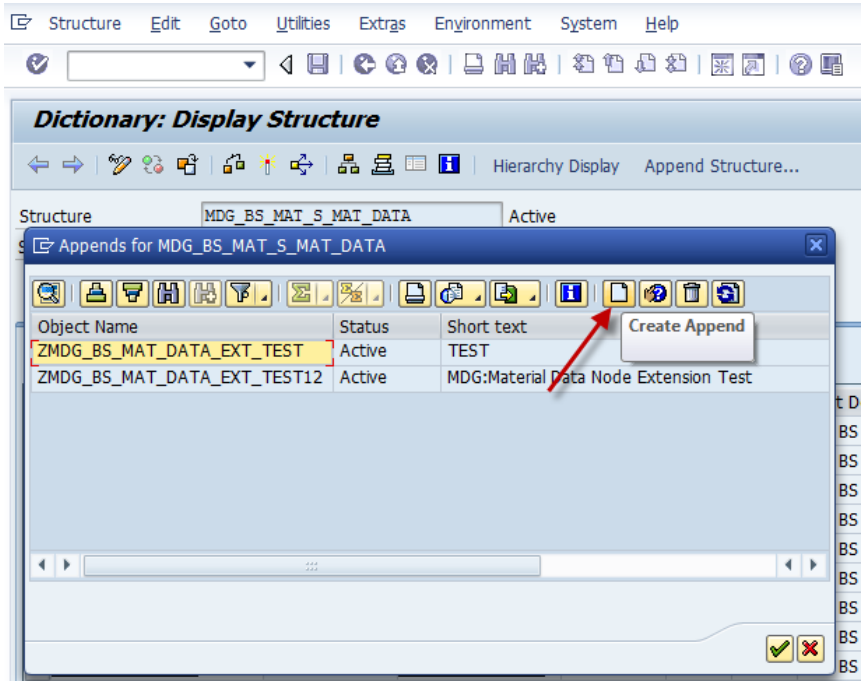
7.	In transaction **SE11** create table type **YMDGM_BUPA01_T_X** as shown.	
8.	Make sure table type **YMDGM_BUPA01_T_X** is a sorted table.	

© 2012 SAP AG

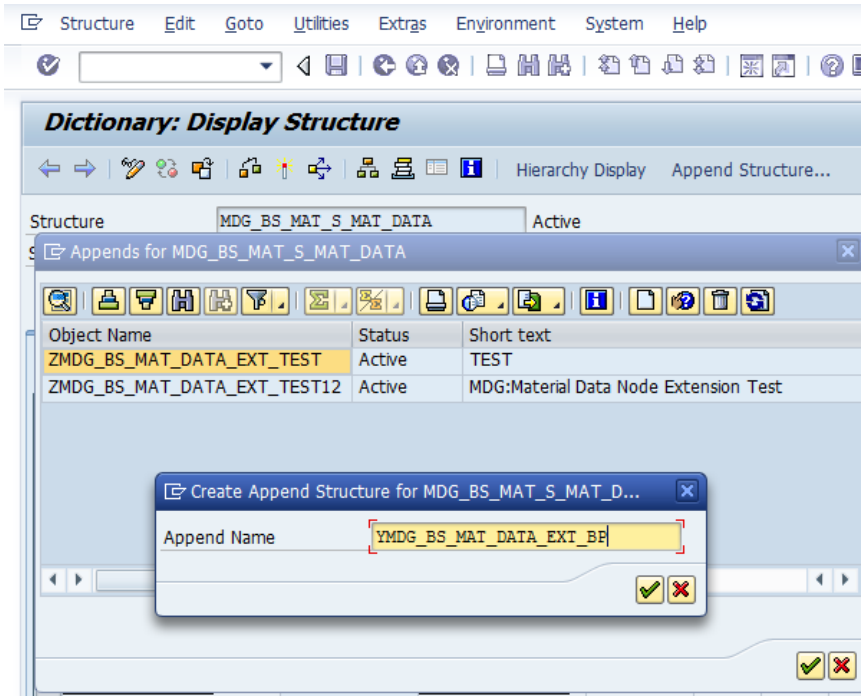
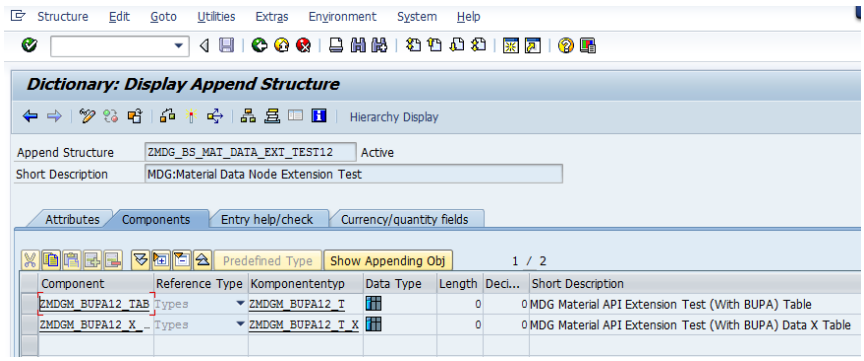
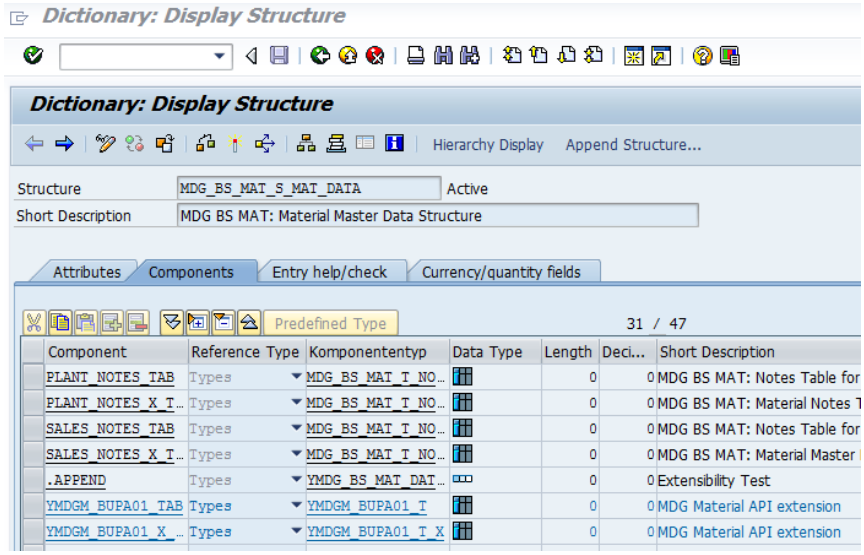
9.	Make sure table type **YMDGM_BUPA01_T_X** has the key fields as shown.	

© 2012 SAP AG

5.1.2.Extend MDG BS MAT (Material Master Data Structure)

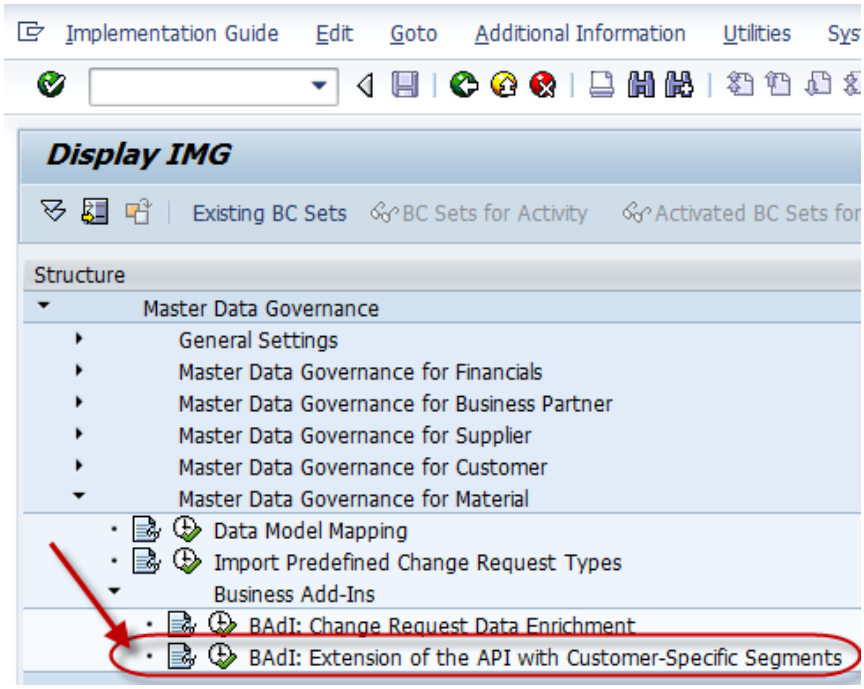
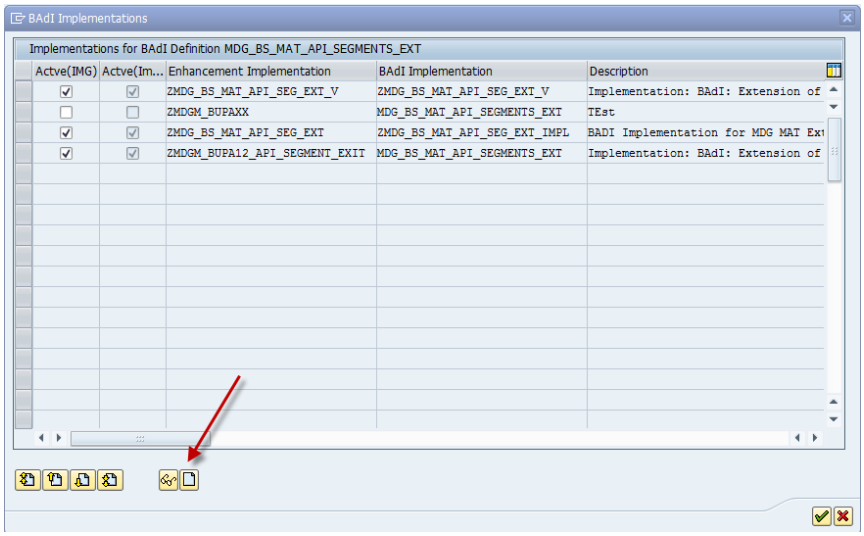
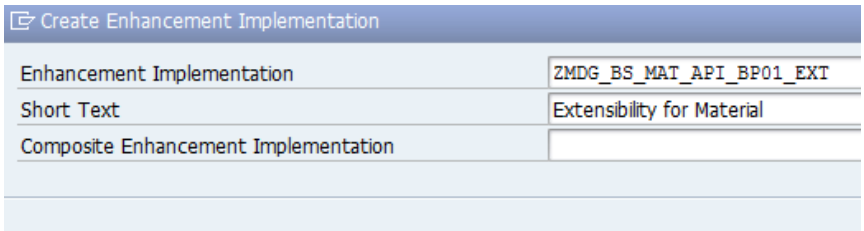
1.	Start transaction **SE11**. Enter **MDG_BS_MAT_S_MAT_DATA** as shown. Choose ***Display***.	
2.	Create a new ***Append***.	

© 2012 SAP AG

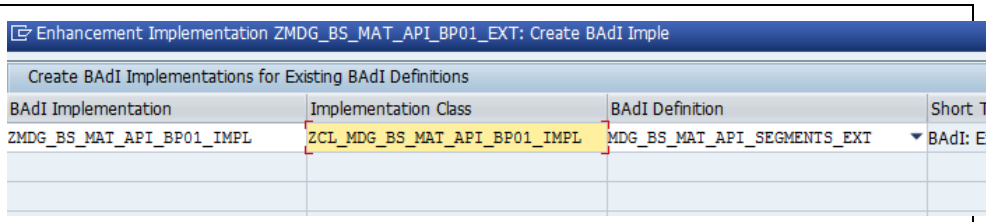
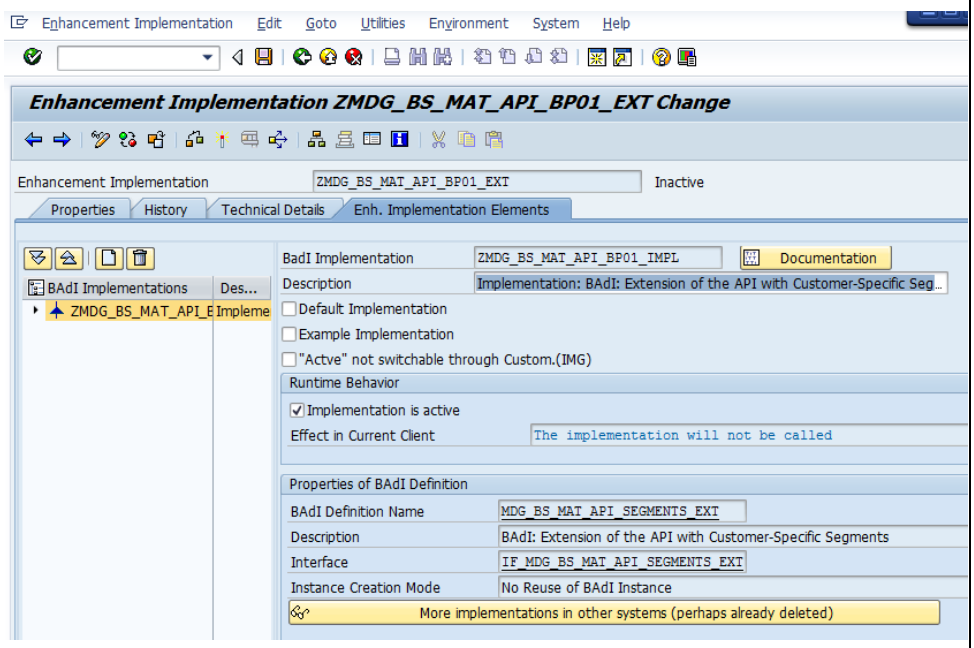
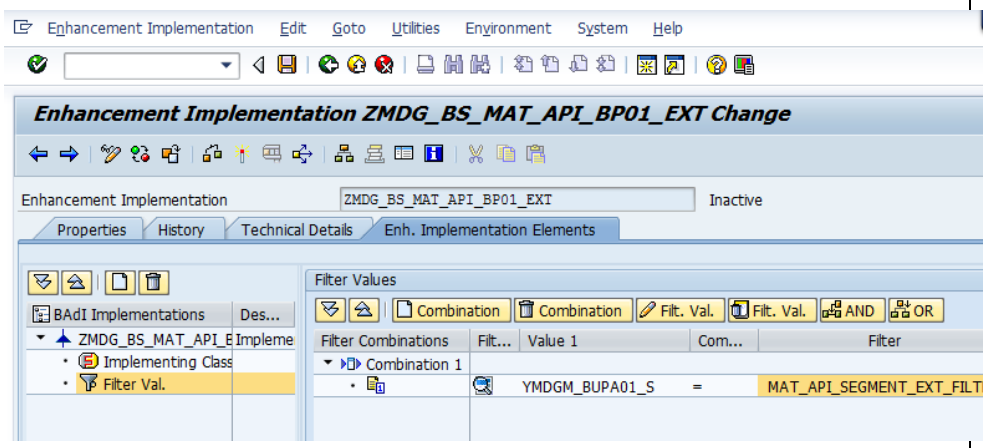
3.	Enter the ***Append Name*** as shown.	
4.	Enter the append component details as shown.	
5.	Save your changes and make sure that the append appears in structure **MDG_BS_MAT_S_MAT_DATA** as shown. Important naming convention: <Tablename>_TAB for the data part and <Tablename>_X_TAB for the change structure with <Tablename> being the name of the database table you want to put under governance.	

© 2012 SAP AG

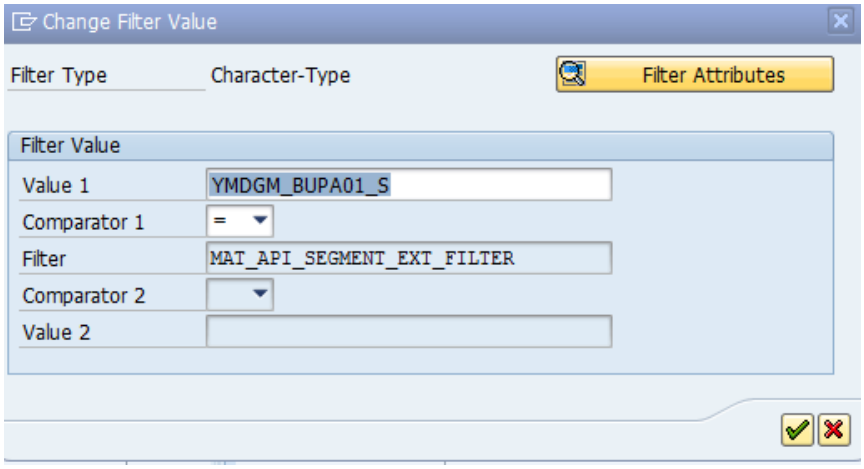
5.1.3.BAdI: Extension of the API with Customer-Specific Segments

1.	In MDG customizing start the selected entry as shown.	
2.	Create a new BAdI implementation.	
3.	Enter the name of the ***Enhancement Implementation*** as shown.	

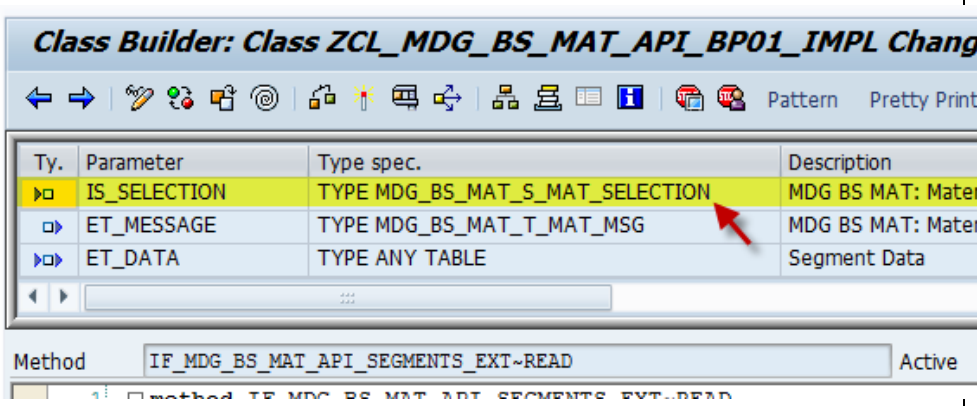
© 2012 SAP AG

4.	Enter further details for the BAdI as shown in the screenshot.	
5.	Check your ***Enhancement Implementation*** details look similar to the screenshot.	
6.	Create a new filter entry.	

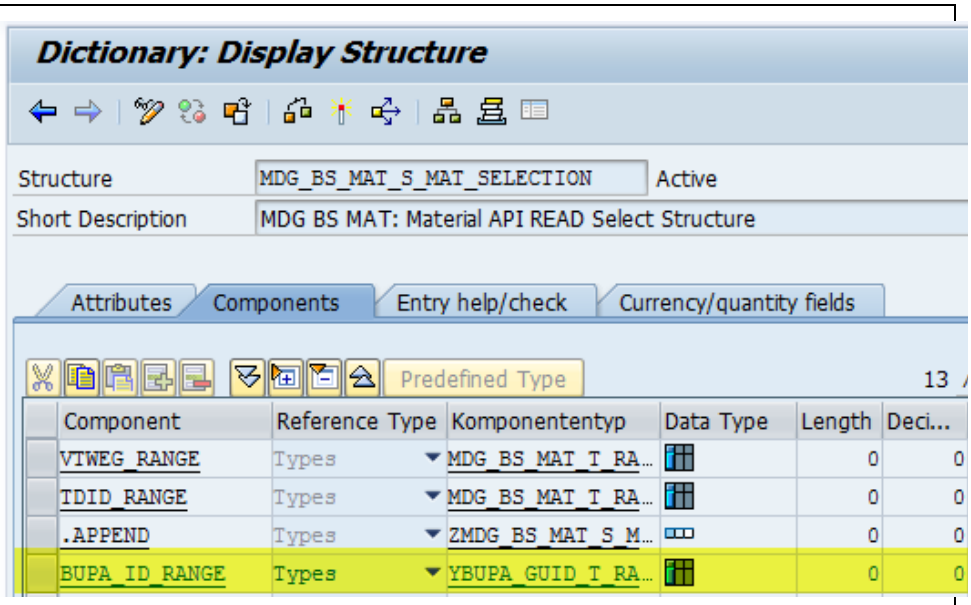
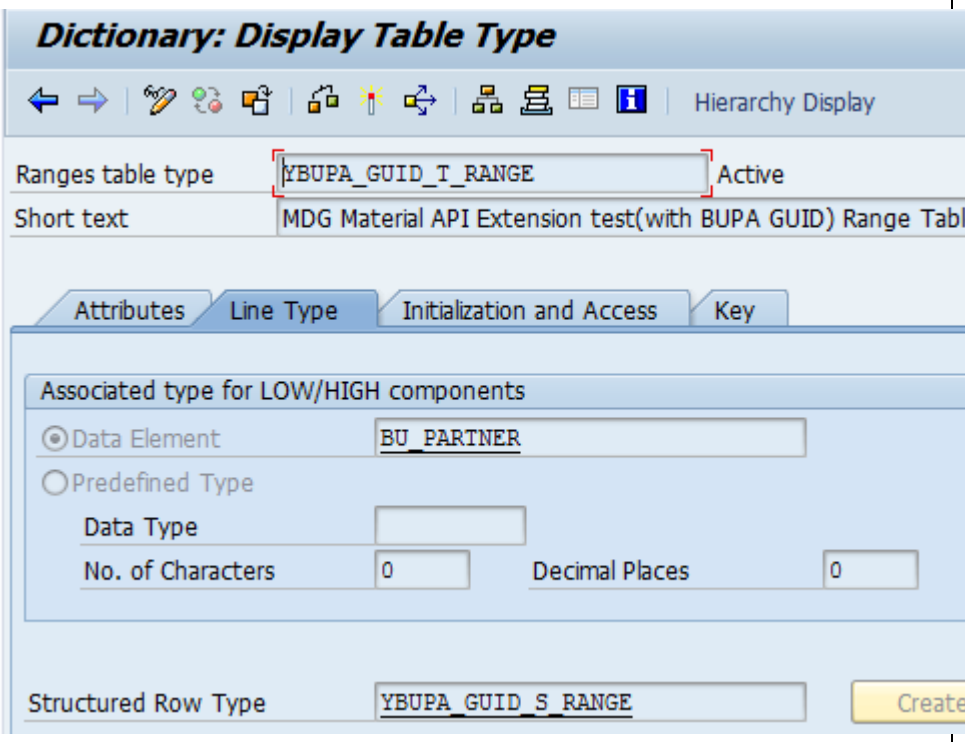
© 2012 SAP AG

7.	Enter the filter value **YMDGM_BUPA01_S**	

5.1.3.1. 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.	

© 2012 SAP AG

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.	

5.1.3.2. BAdI Implementation: READ Method

1.	`METHOD if_mdg_bs_mat_api_segments_ext~read.` `  DATA: lt_mdgm_bupa TYPE ymdgm_bupa01_t.`

© 2012 SAP AG

```
  CLEAR: et_data, et_message.
  SELECT * FROM ymdgm_bupa00 INTO CORRESPONDING FIELDS OF TABLE lt_mdgm_bupa
    WHERE matnr   IN is_selection-matnr_range
      AND bupa_id IN is_selection-bupa_id_range.
  INSERT LINES OF lt_mdgm_bupa INTO TABLE et_data.
ENDMETHOD.
```

5.1.3.3. BAdI Implementation: CHECK_AND_SAVE Method

2.

```
METHOD if_mdg_bs_mat_api_segments_ext~check_and_save.

  DATA: ls_data    TYPE ymdgm_bupa01_s.
  DATA: ls_data_x  TYPE ymdgm_bupa01_s_x.
  DATA: ls_message TYPE mdg_bs_mat_s_mat_msg.
  DATA: lt_modify TYPE STANDARD TABLE OF ymdgm_bupa00.
  DATA: ls_modify TYPE ymdgm_bupa00.
  DATA: lt_delete  TYPE STANDARD TABLE OF ymdgm_bupa00.
  DATA: ls_delete  TYPE ymdgm_bupa00.

  CLEAR et_message.
* First some checks
  LOOP AT is_data-ymdgm_bupa01_tab INTO ls_data.
    " All fields must be filled, otherwise MESSAGE e000(00) WITH ls_data-
matnr ls_data-bupa_id ls_data-nickname.
    IF ls_data-matnr IS INITIAL OR ls_data-bupa_id IS INITIAL OR ls_data-
nickname IS INITIAL.
      ls_message-msgid = '00'.
      ls_message-msgno = '000'.
      ls_message-msgty = 'E'.
      ls_message-msgv1 = 'Empty value detected in: '.
      ls_message-msgv2 = ls_data-matnr.
      ls_message-msgv3 = ls_data-bupa_id.
      ls_message-msgv4 = ls_data-nickname.
      INSERT ls_message INTO TABLE et_message.
    ENDIF.
    " BuPa must exist
    " ...
  ENDLOOP.
* In test mode, we're finished now
  CHECK iv_test_mode = abap_false.
* Determine which records to delete or to insert/update
  LOOP AT is_data-ymdgm_bupa01_x_tab INTO ls_data_x.
    IF ls_data_x-delete_row = abap_true. " Keep key fields for deletion
      ls_delete-matnr = ls_data_x-matnr.
      ls_delete-bupa_id = ls_data_x-bupa_id.
      INSERT ls_delete INTO TABLE lt_delete.
    ELSE.
      READ TABLE is_data-ymdgm_bupa01_tab INTO ls_data
           WITH TABLE KEY matnr   = ls_data_x-matnr
                          bupa_id = ls_data_x-bupa_id.
      ASSERT sy-subrc = 0. " No _x record without data record
      MOVE-CORRESPONDING ls_data TO ls_modify.
      INSERT ls_modify INTO TABLE lt_modify.
    ENDIF.
  ENDLOOP.
* Now change the DB
* Note: In a productive implementation, these DB changes must be done ON COMM
```


© 2012 SAP AG

```
IT (PERFORM xxx ON COMMIT or CALL FUNCTION xxx IN UPDATE TASK)
* Note: We are basically ignoring the _X structure here -
 if data gets changed, all fields get updated
  IF NOT lt_delete IS INITIAL.
    DELETE  ymdgm_bupa00 FROM TABLE lt_delete.
  ENDIF.
  IF NOT lt_modify IS INITIAL.
    MODIFY ymdgm_bupa00 FROM TABLE lt_modify.
  ENDIF.
ENDMETHOD.
```

5.1.3.4. BAdI Implementation: GET_ES_NODEINFO Method

Implement GET_ES_NODEINFO Method if you have extended the enterprise search model MATERIAL. See also extensibility guide for the Search: http://scn.sap.com/docs/DOC-29002.

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.

5.2. 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.

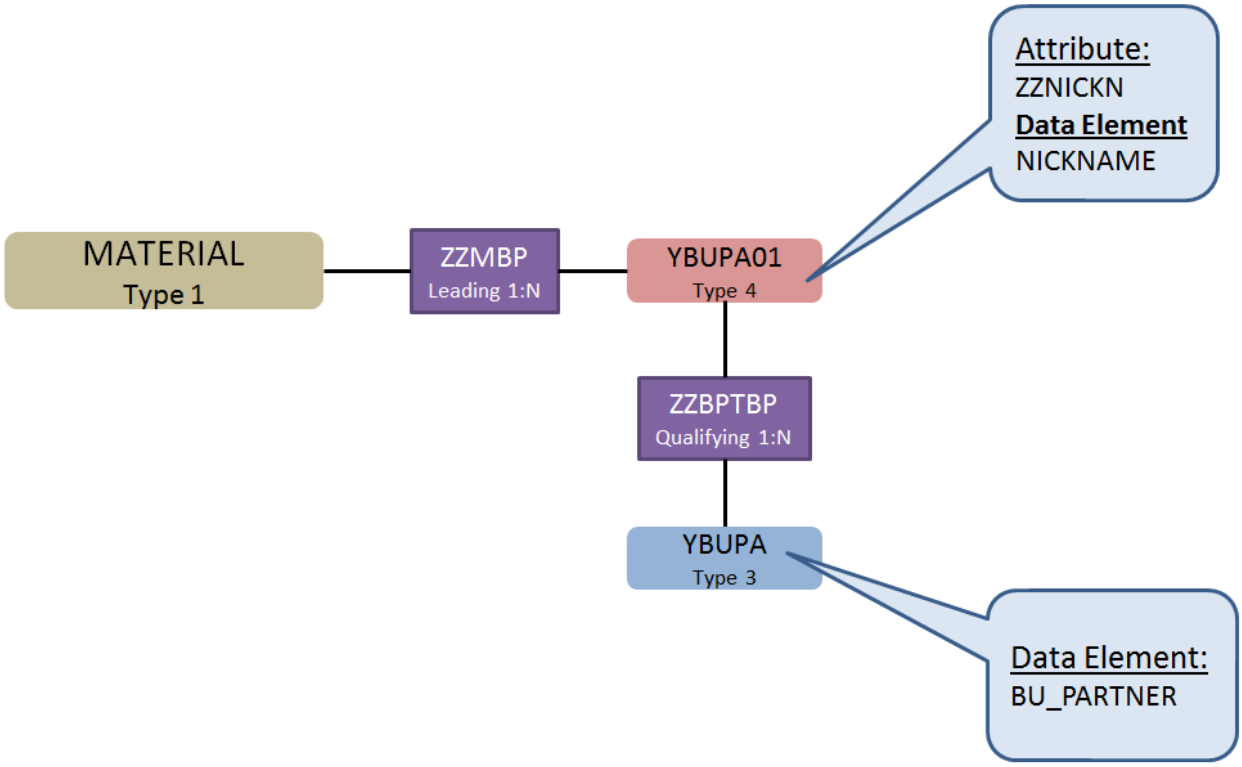
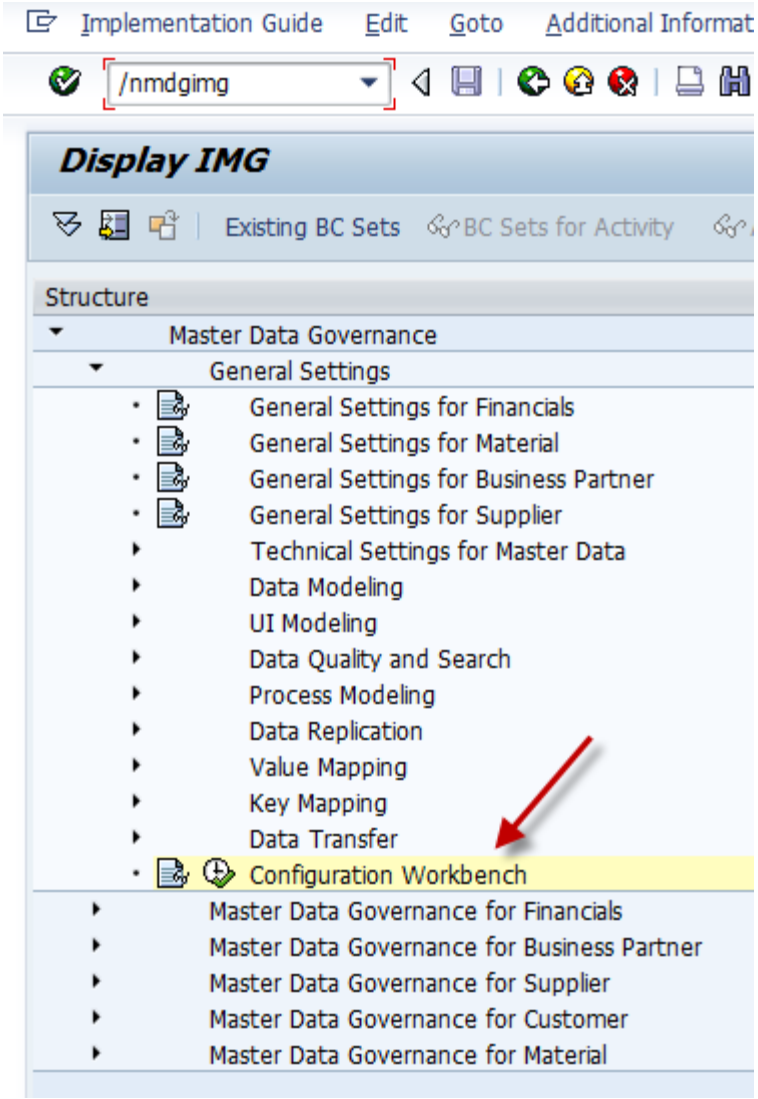


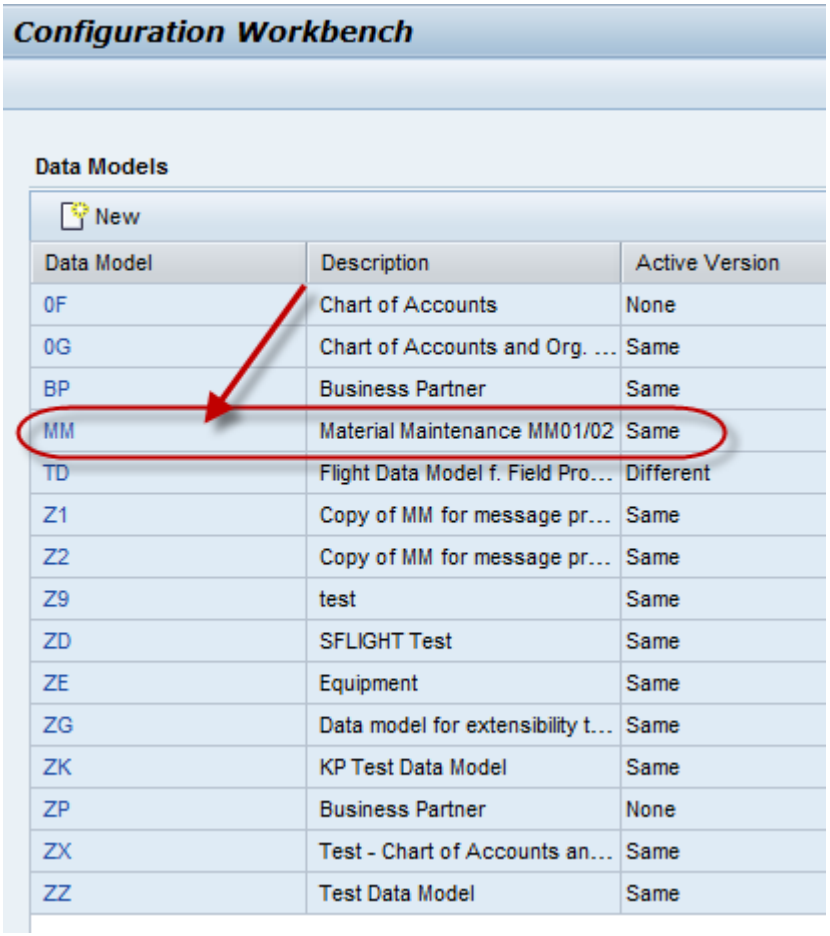
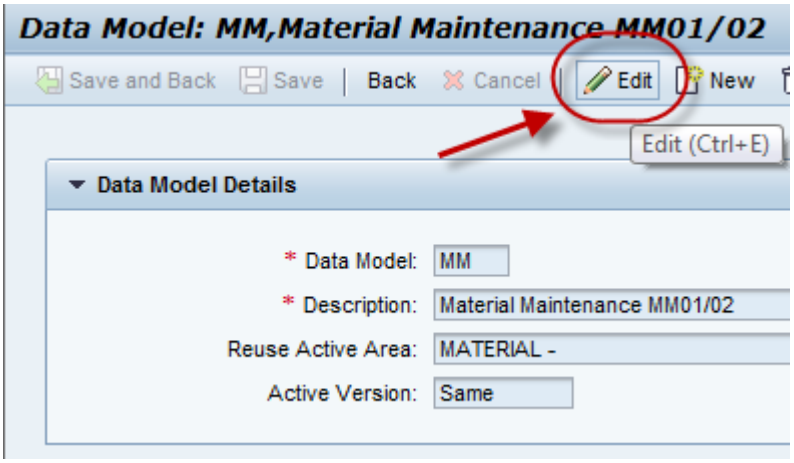
Figure: Data Model details for extension

© 2012 SAP AG

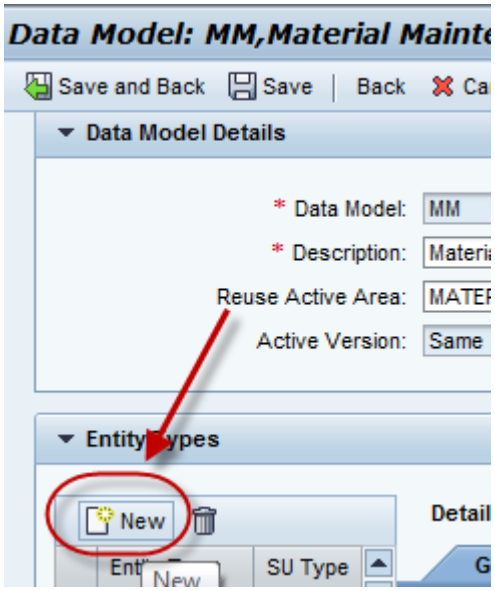
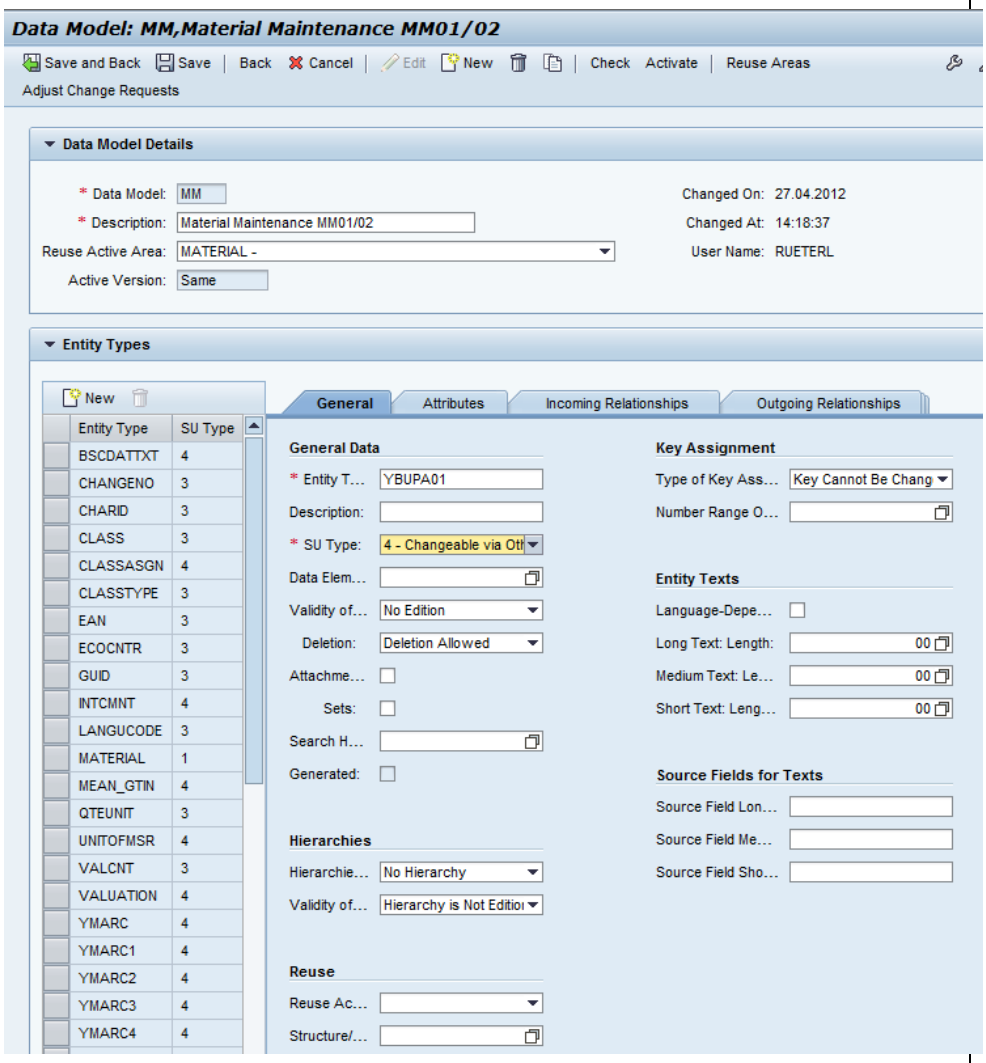
5.2.1.Extend MDG Data Model

1.	Start the MDG customizing with transaction **`MDGIMG`** and start the ***Configuration Workbench*** as shown.	

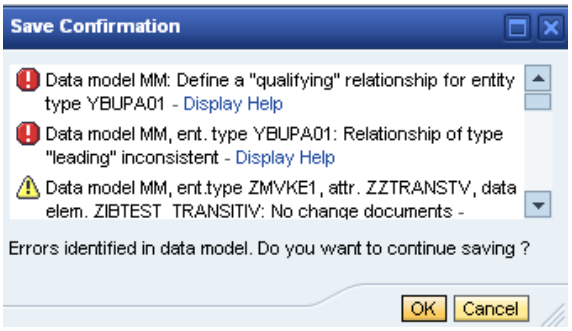
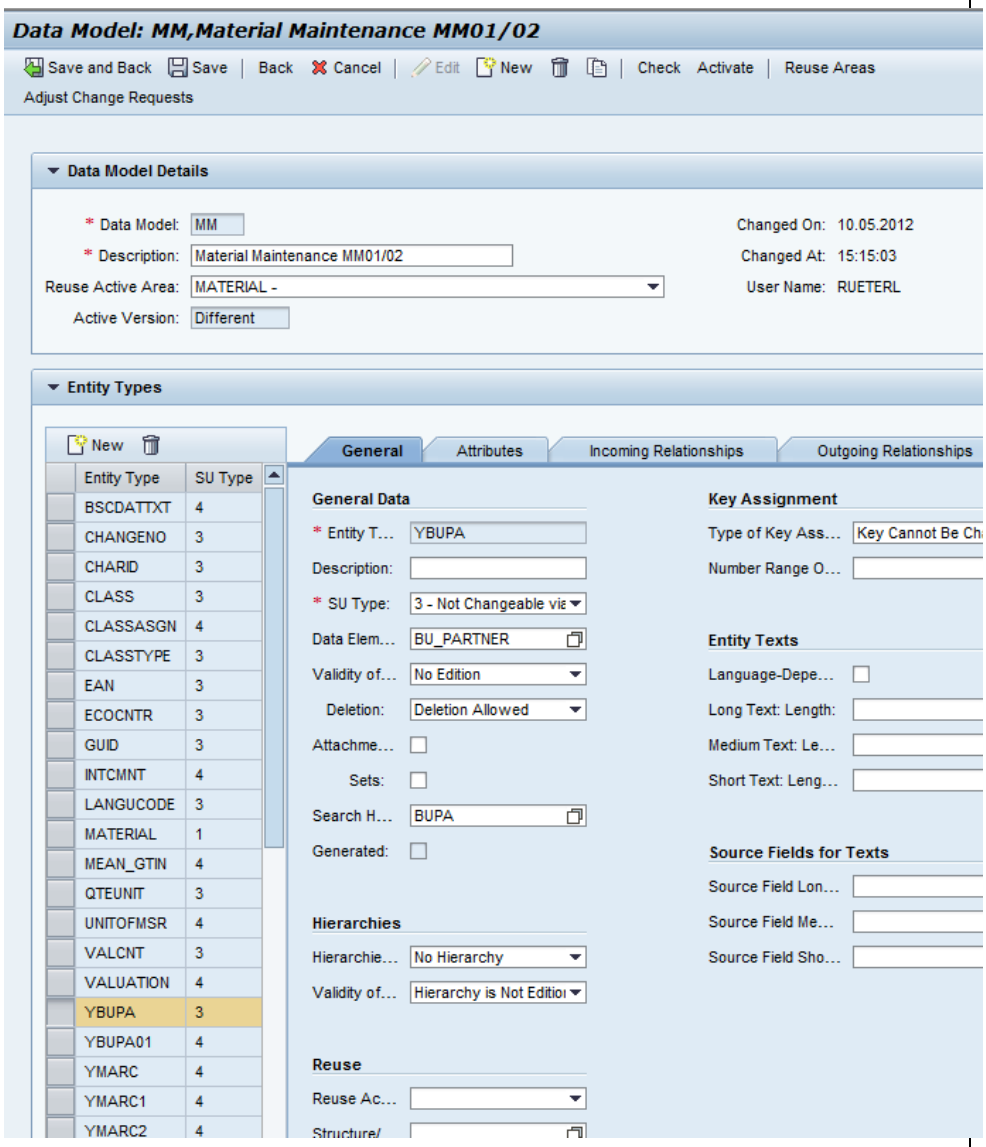
© 2012 SAP AG

2.	In the Configuration Workbench open the **MM *Data Model*.**	
3.	Choose the ***Edit*** pushbutton.	

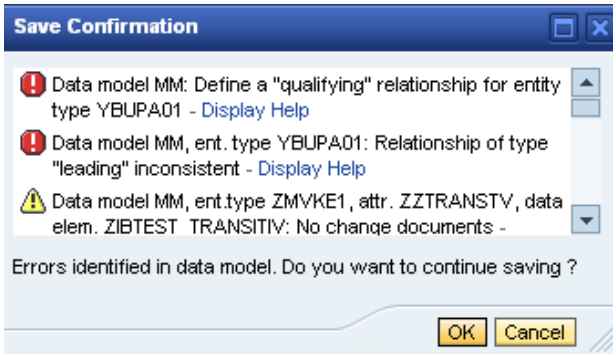
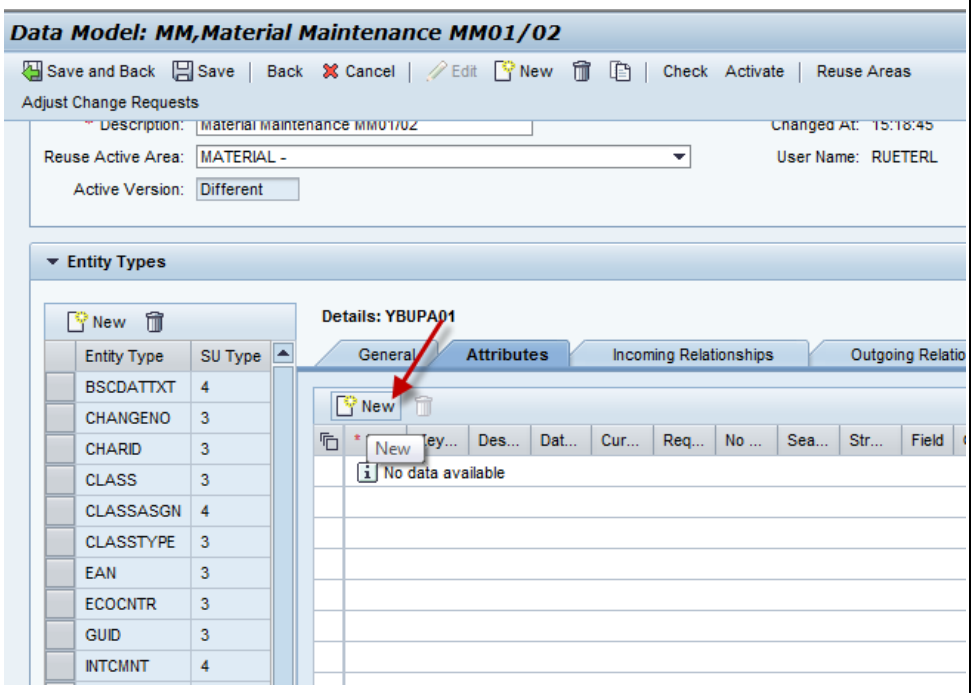
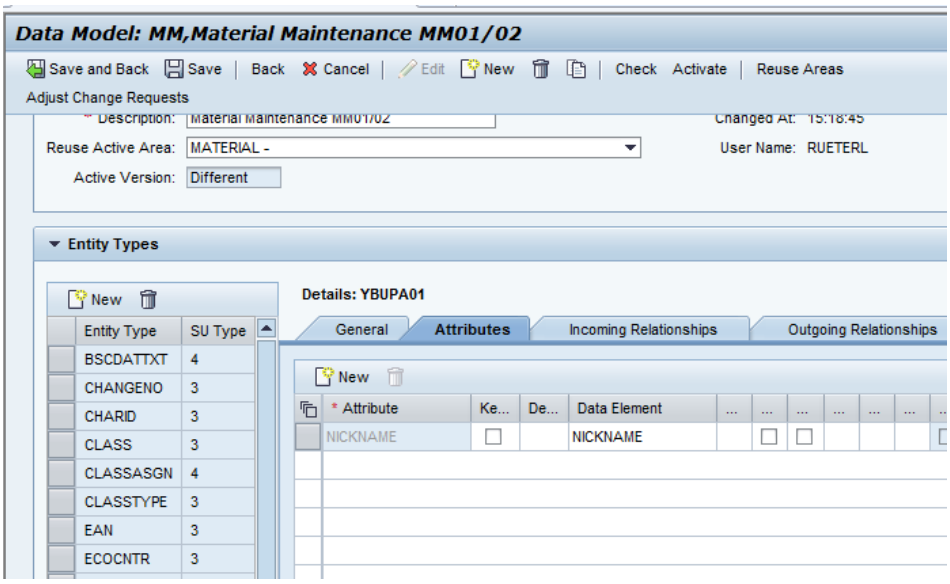
© 2012 SAP AG

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.	

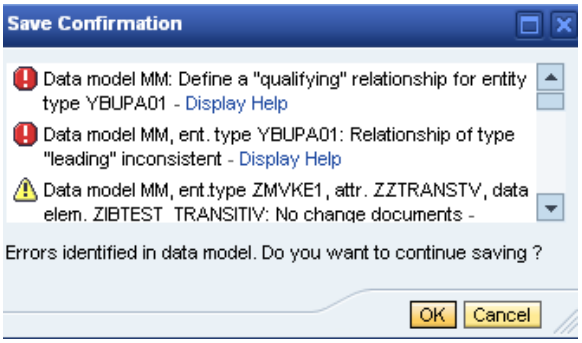
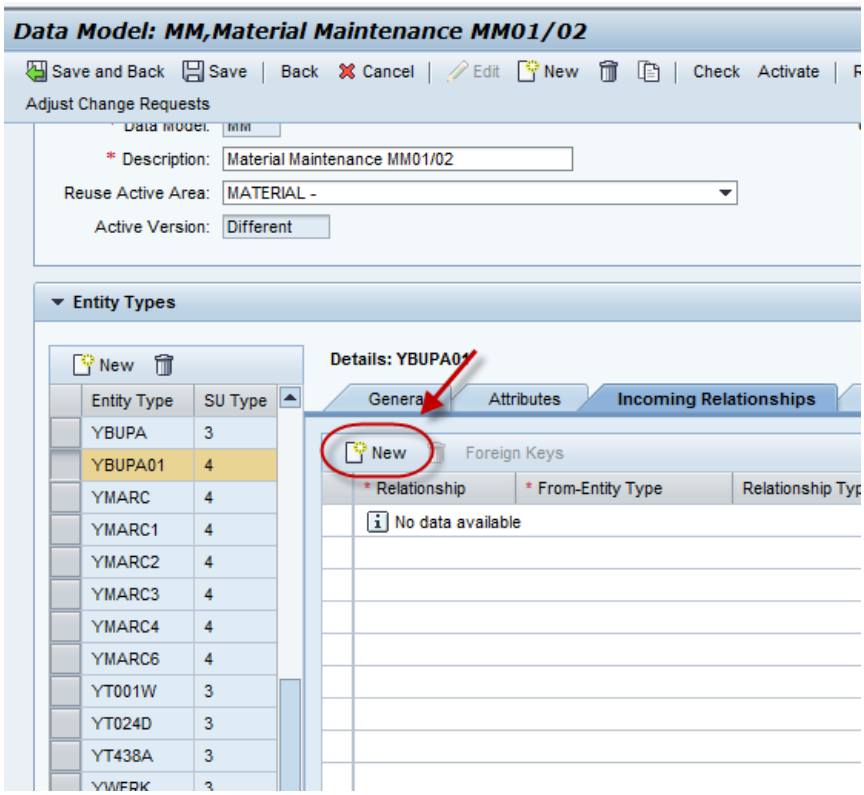
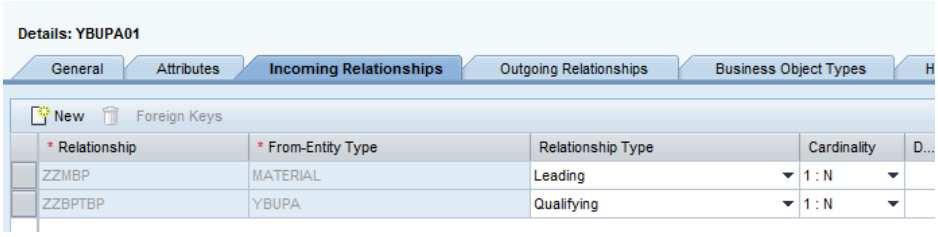
© 2012 SAP AG

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.	

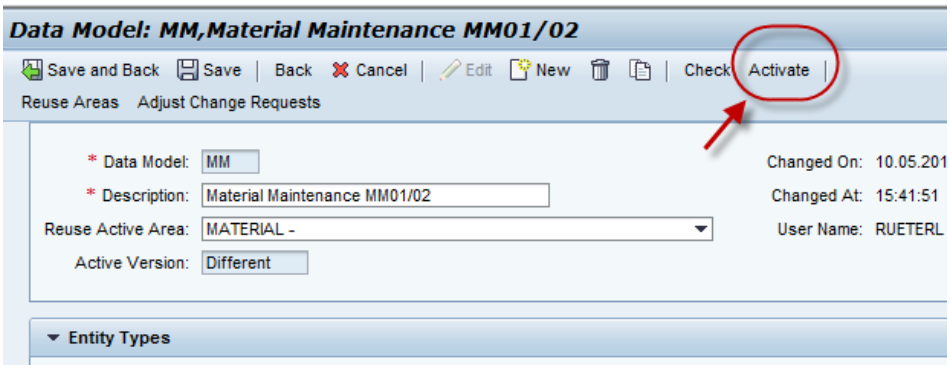
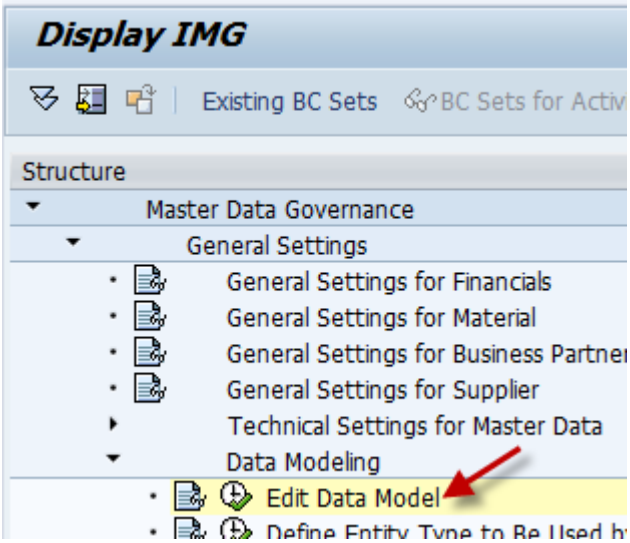
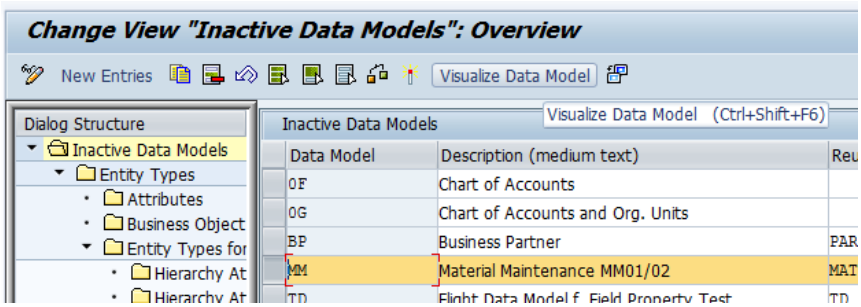
© 2012 SAP AG

8.	Choose ***Save***. Choose ***OK*** in the confirmation dialog.	
9.	Go back to Entity Type **YBUPA**01. Switch the ***Attributes*** tab. Choose the ***New*** pushbutton.	
10.	Create the new attribute **NICKNAME** as shown.	

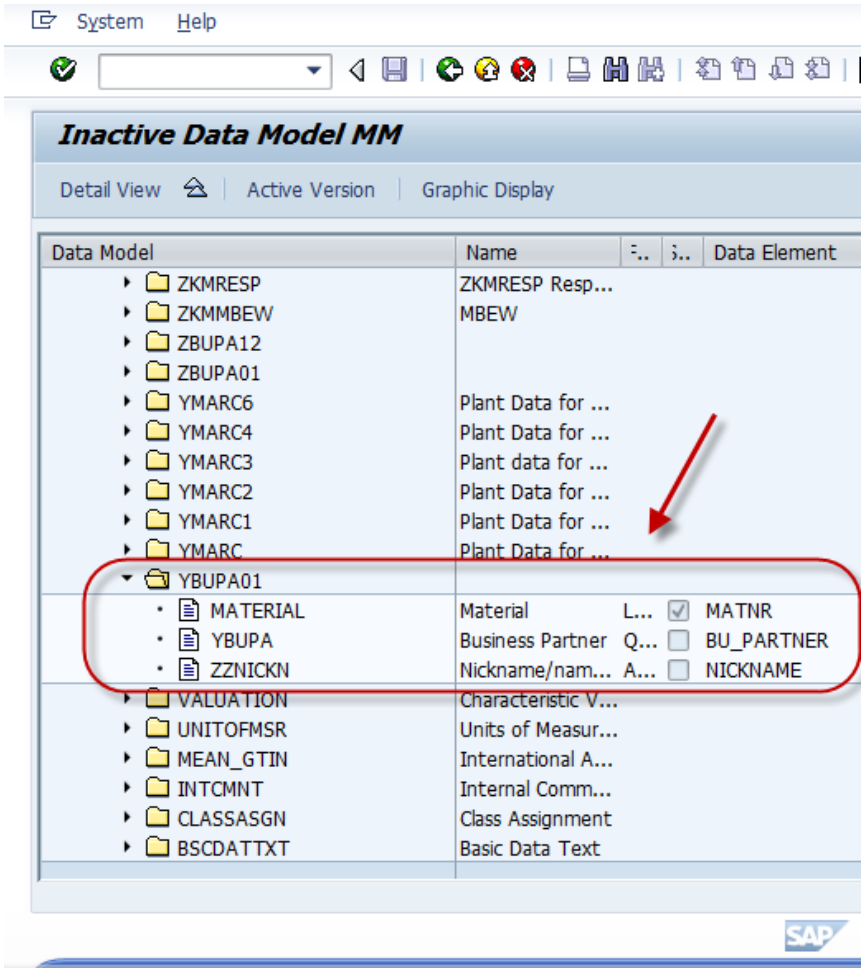
© 2012 SAP AG

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.	

© 2012 SAP AG

14.	Choose the ***Activate*** pushbutton to activate your data model.	
15.	To check the activated data model open the ***Edit Data Model*** Customizing activity as shown.	
16.	Choose the ***Visualize Data Model*** pushbutton .	

© 2012 SAP AG

17.	Make sure that your extension of the MDG Data Model looks similar to the details shown in the screenshot.	

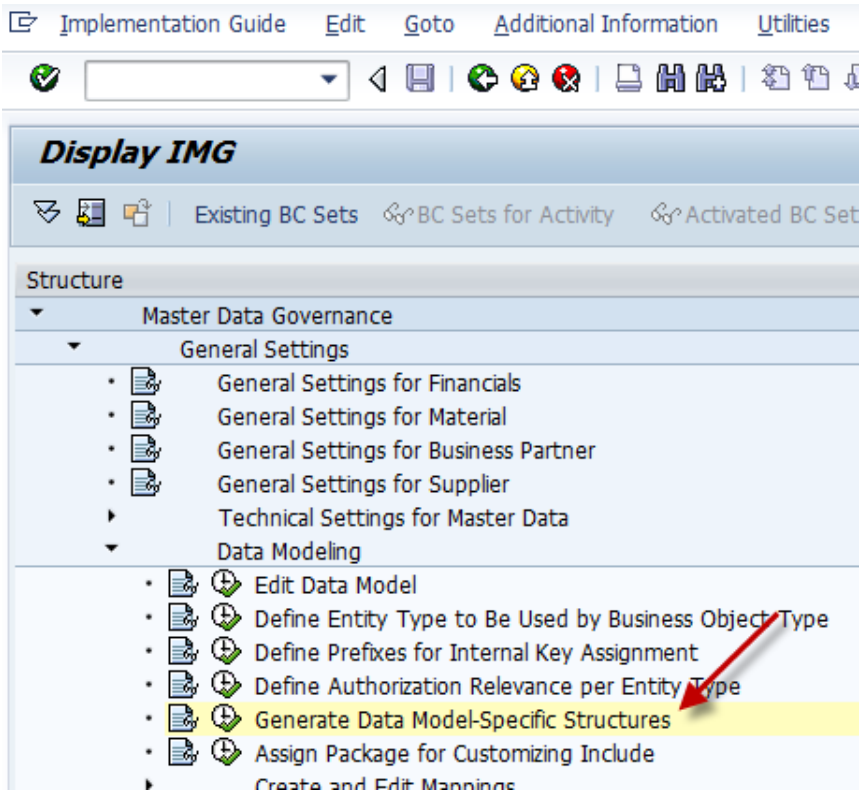
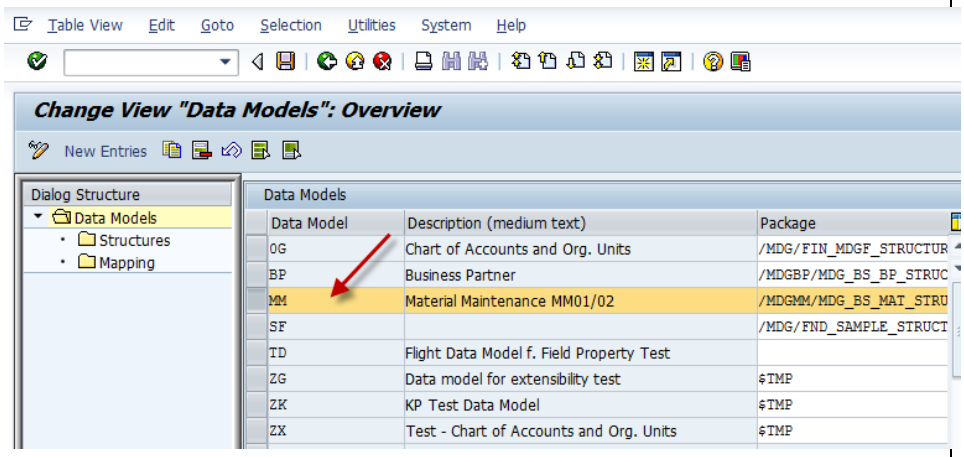
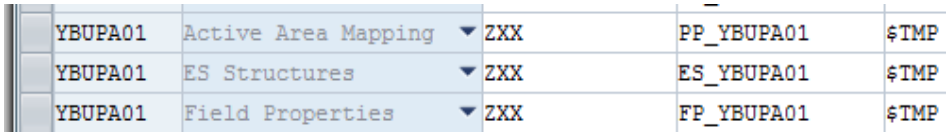
5.2.2.Generate Model-Specific Structures

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.

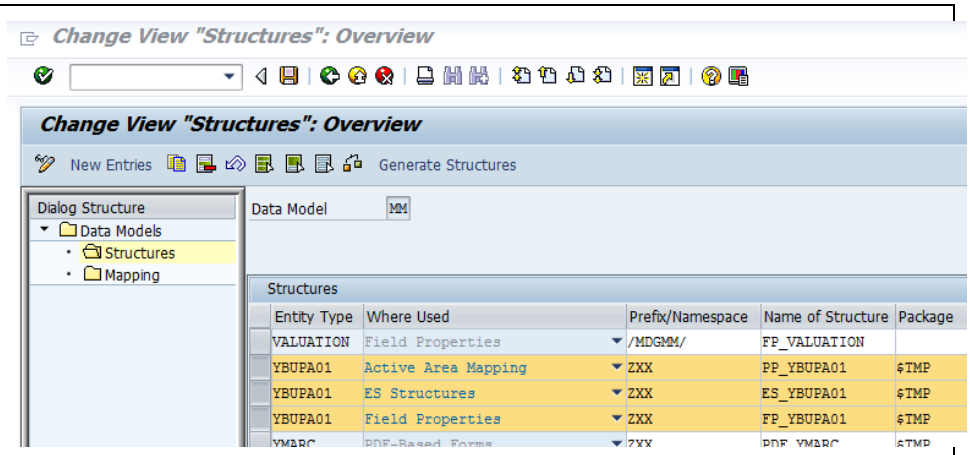
Note

In general if you change a data model (for example, if you change attributes of entity types or relationships); you must regenerate the structures.

© 2012 SAP AG

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`**.	

© 2012 SAP AG

4.	Save your changes.	

5.2.1. Clear UI Metadata Buffers

After finishing the UI, clear the metadata buffers. You can find the report in customizing Master Data Governance, Central Governance-> Master Data Governance for Material-> Clear UI Metadata Buffers.

This report clears the following buffers in this sequence:

- Text Buffer
- Search Help Buffer
- SMT-Mapping Data Buffer
- SPI Metadata Buffer

You should use this report after extending the data model to make sure that the metadata is consistent with the MDG customizing and UI configuration.

5.3. Create SMT-Mapping

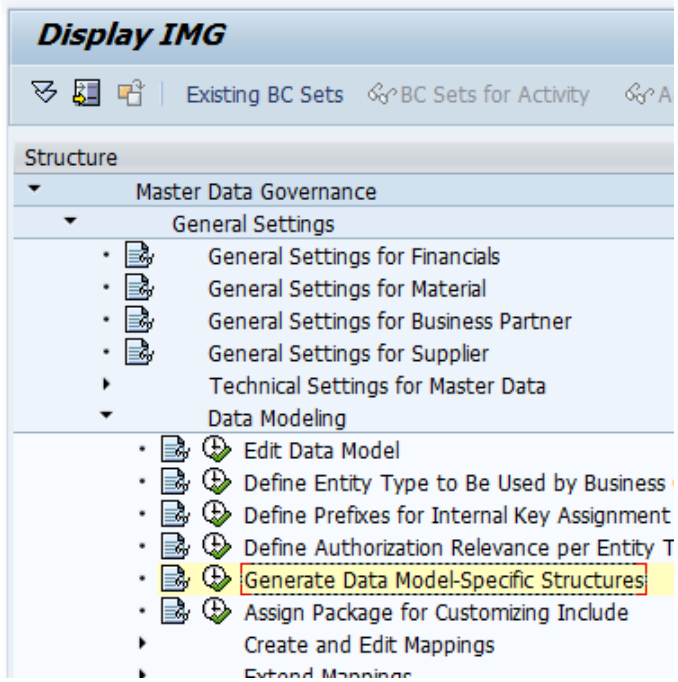
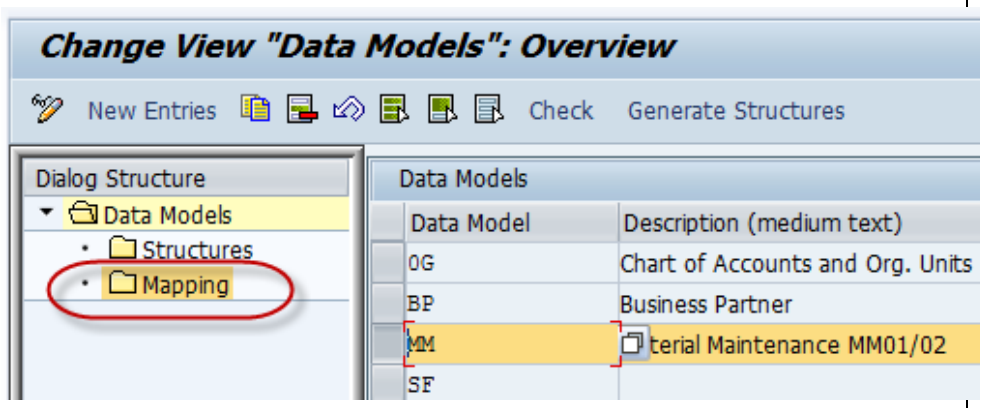
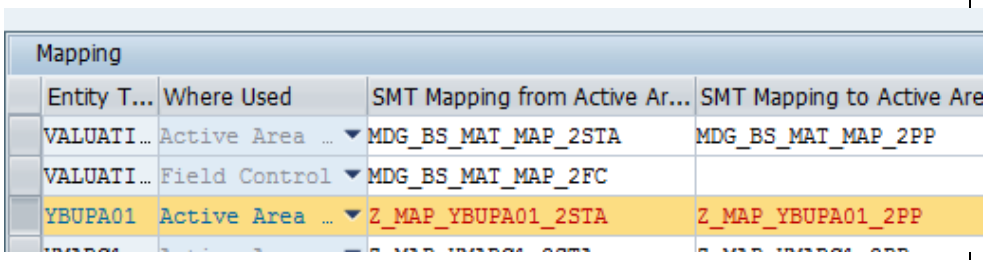
You extend mappings by creating new transformations (complex transformations, field mappings) and field checks for them or by editing them.

Important

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

© 2012 SAP AG

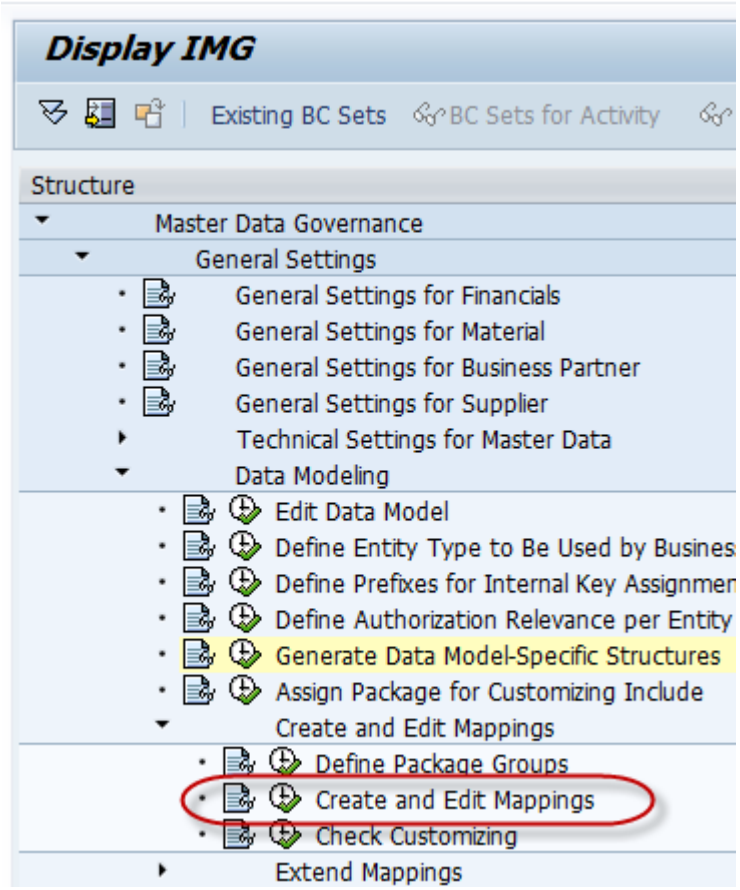
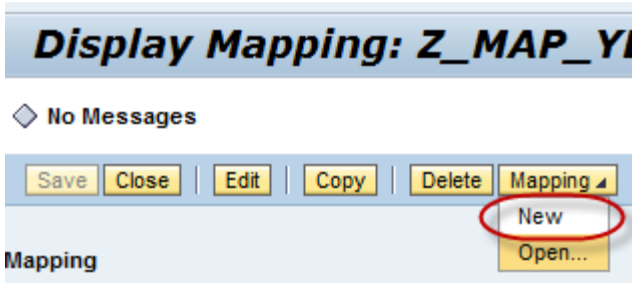
5.3.1.Create Mapping Entries in Customizing

1.	In MDG customizing start ***Generate Data Model-Specific Structures***.	
2.	Select the **MM** data model. Open the ***Mapping*** view.	
3.	Create a new entry for Entity Type **YBUPA01** / Active Area Mapping. Mapping from the active area to the staging area: **Z_MAP_YBUPA01_2STA** Mapping from the staging area to the active area: **Z_MAP_YBUPA01_2**	

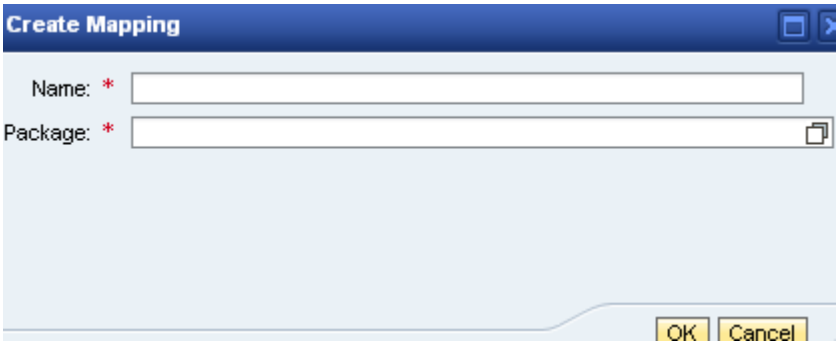
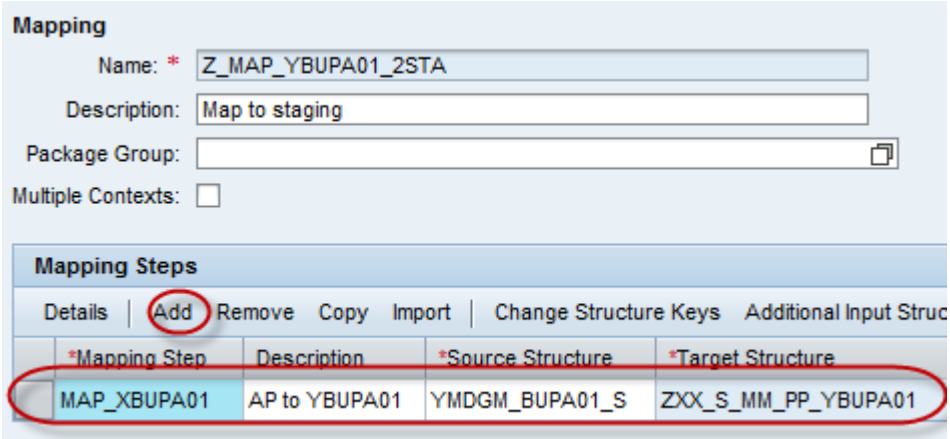
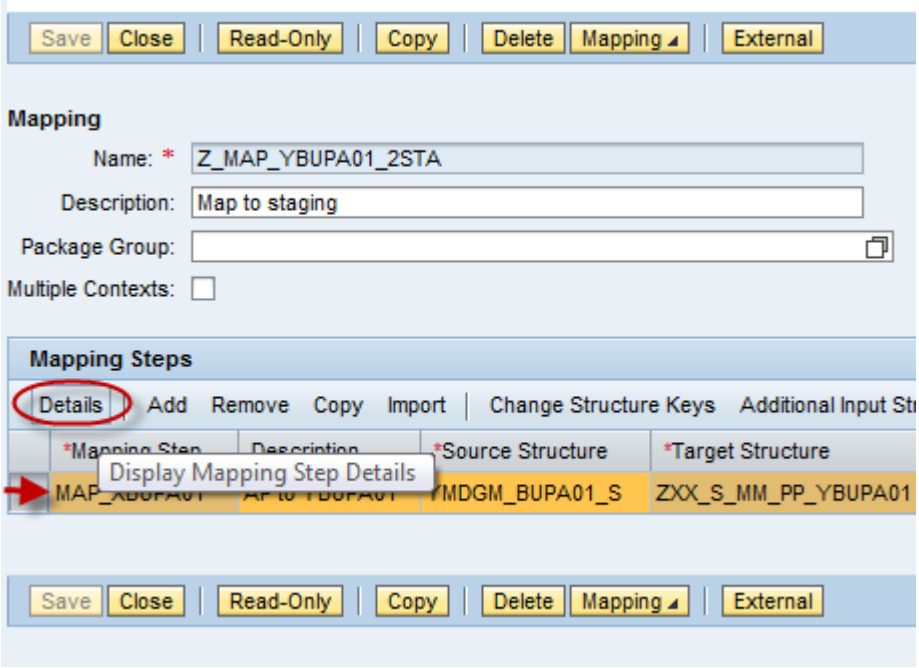
© 2012 SAP AG

	PP	

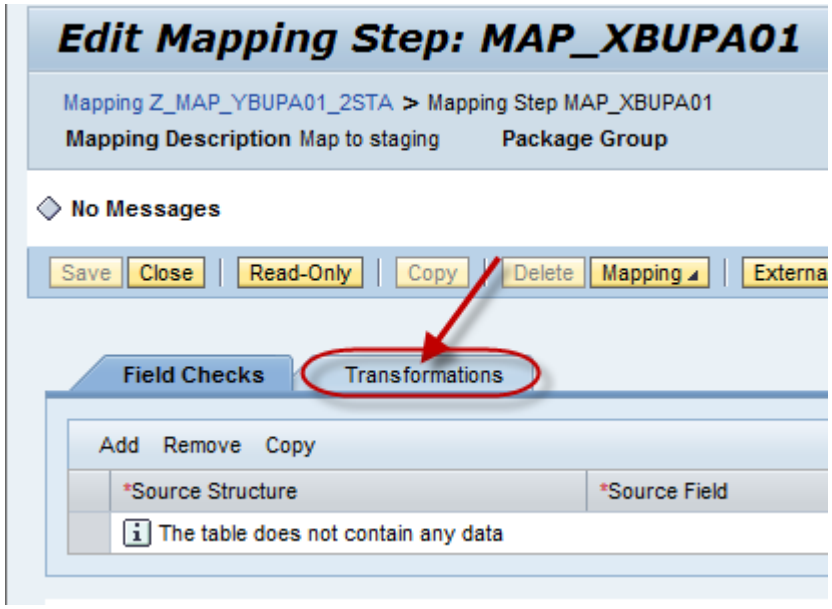
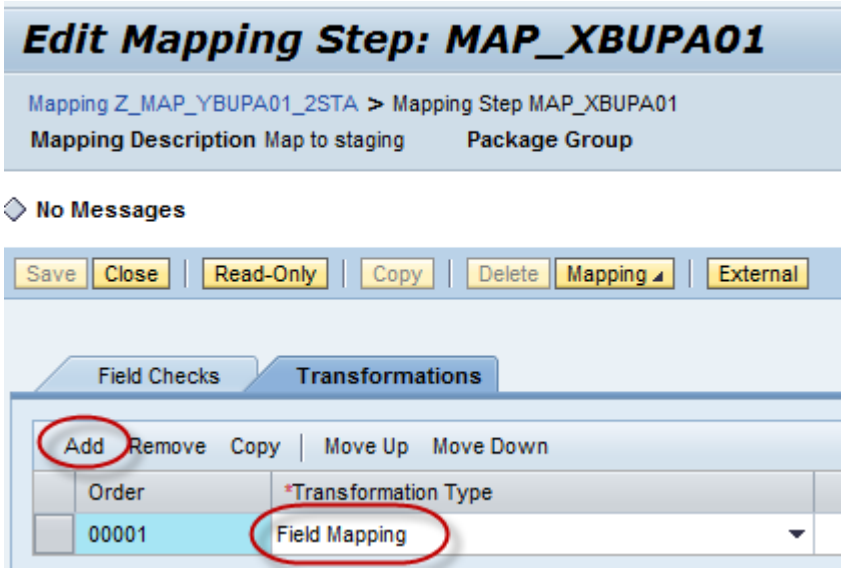
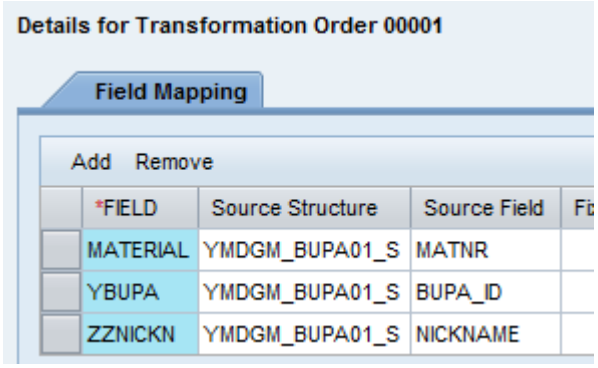
5.3.2.Map the Active Area to the Staging Area

4.	In MDG customizing start ***Create and Edit Mappings.***	
5.	Select ***Mapping -> New***.	

© 2012 SAP AG

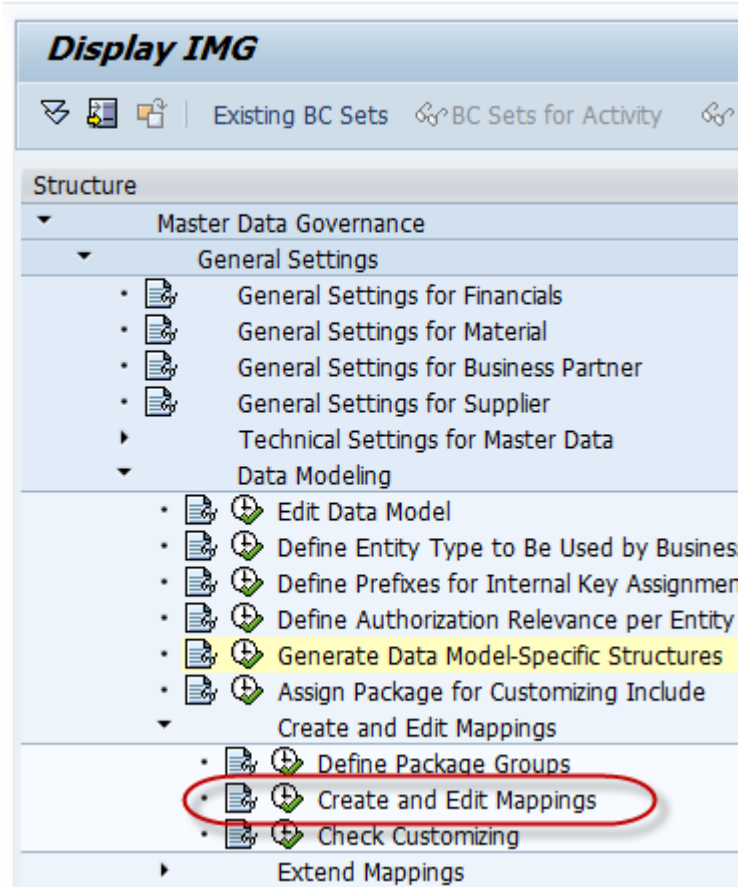
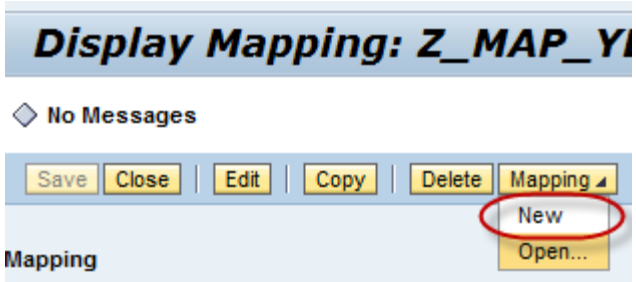
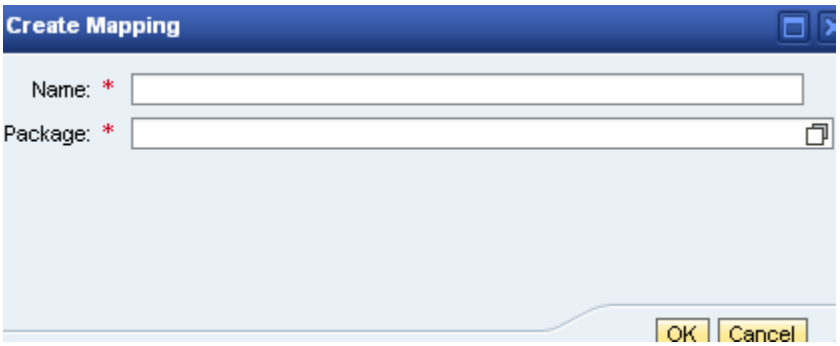
6.	Enter the name of the mapping: **`Z_MAP_YBUPA01_2STA`**	
7.	Create a new mapping step as shown.	
8.	Select you mapping step and choose the ***Details*** pushbutton.	

© 2012 SAP AG

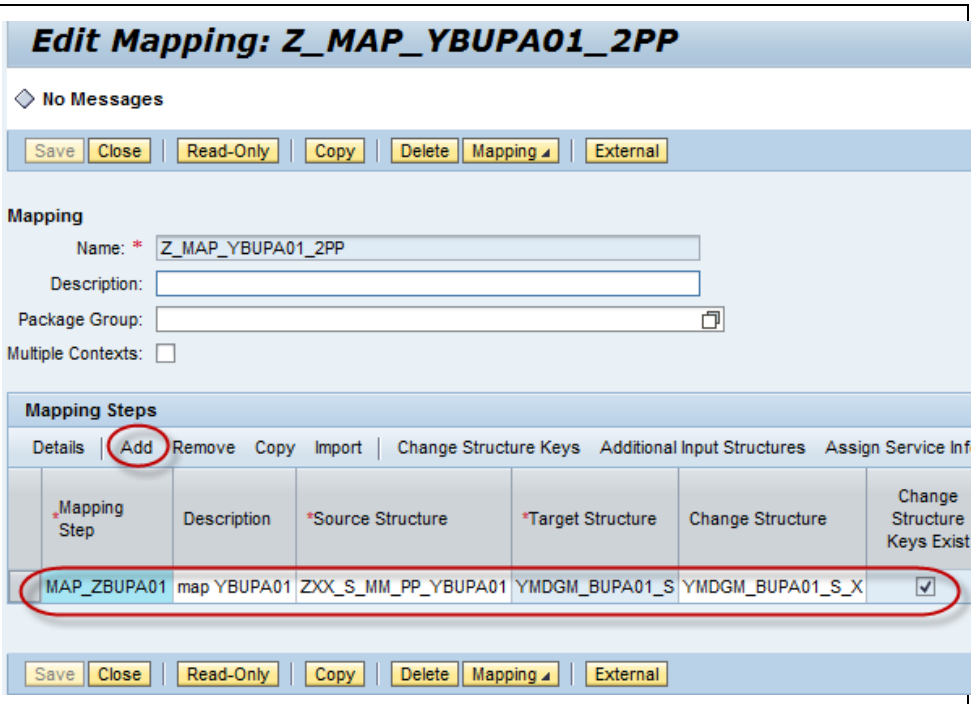
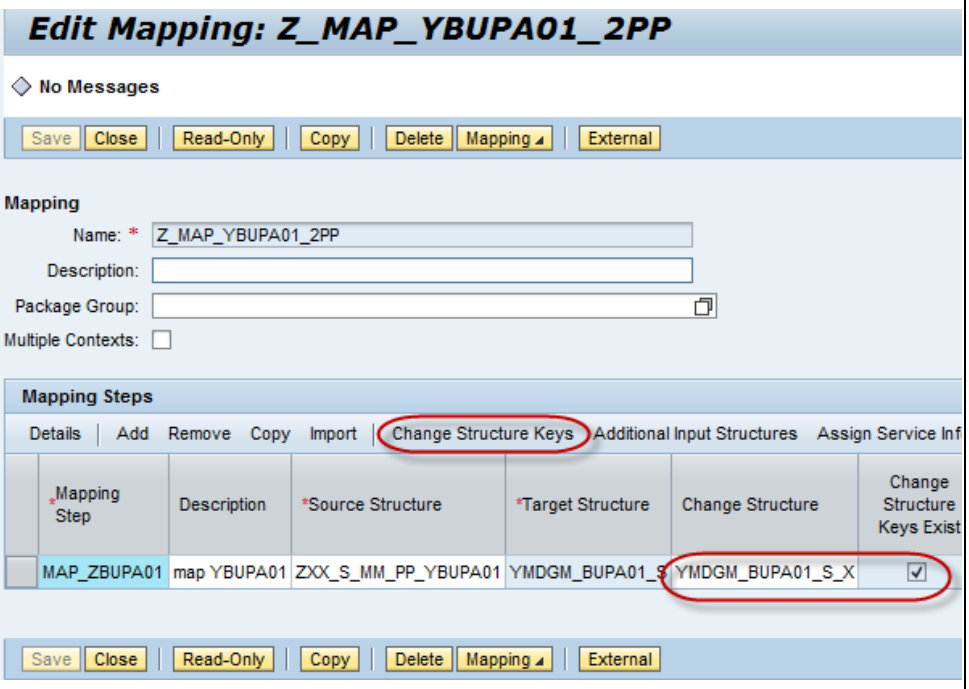
9.	Open the ***Transformations*** tab.	
10.	Add a new ***Field Mapping***.	
11.	Enter the field mappings as shown. Save your changes.	

© 2012 SAP AG

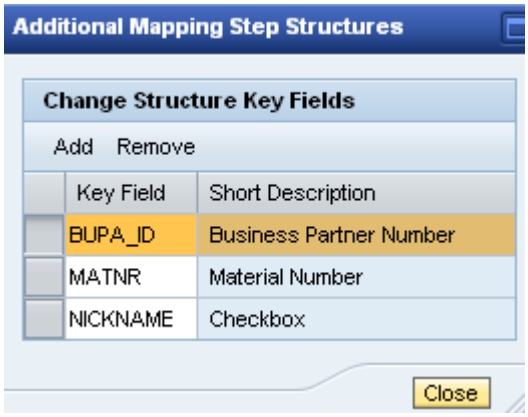
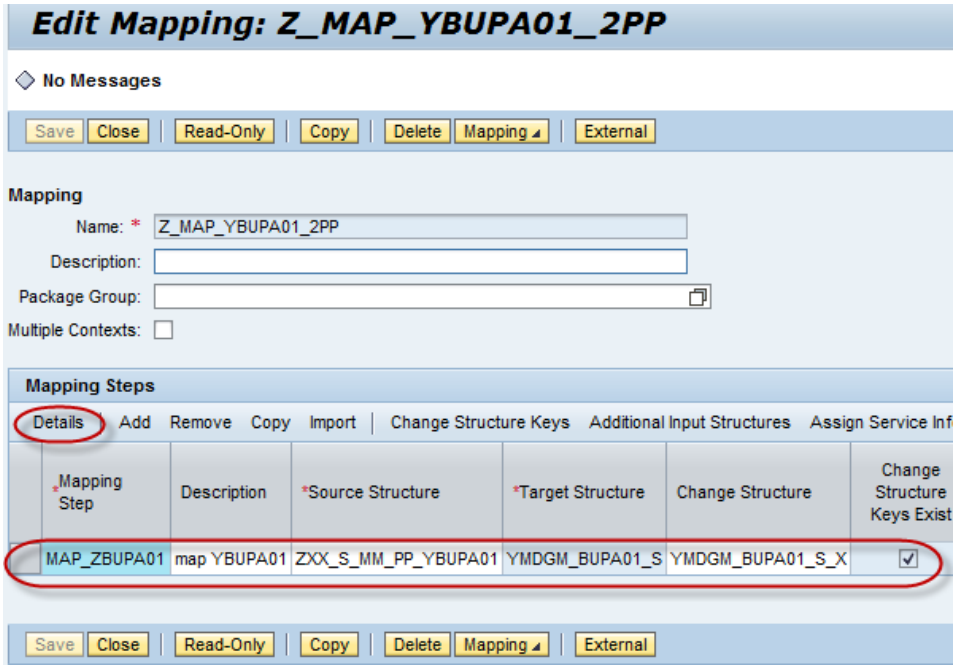
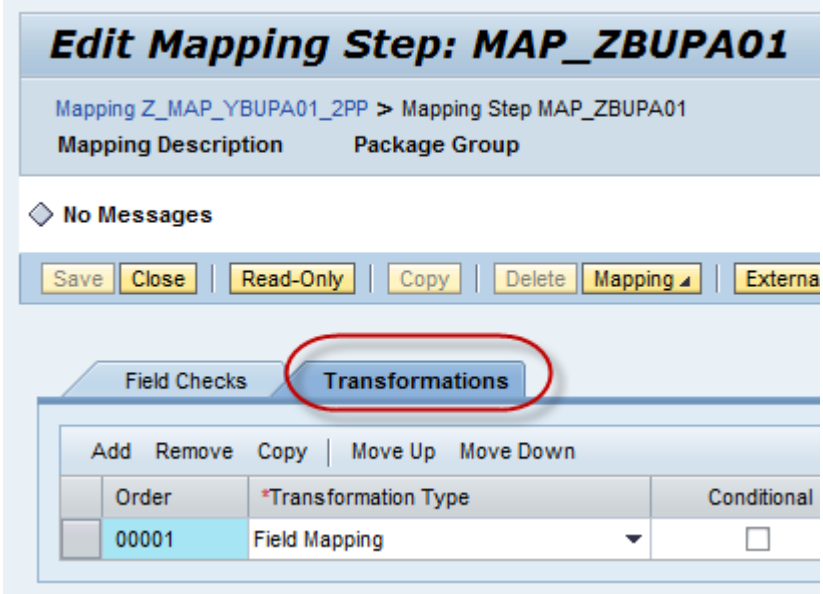
5.3.3.Map the Staging Area to the Active Area

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_2PP`**	

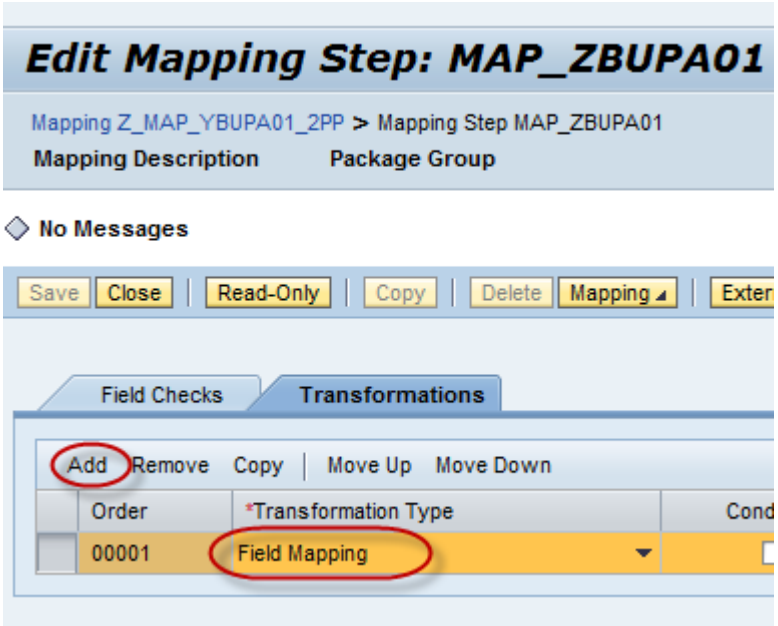
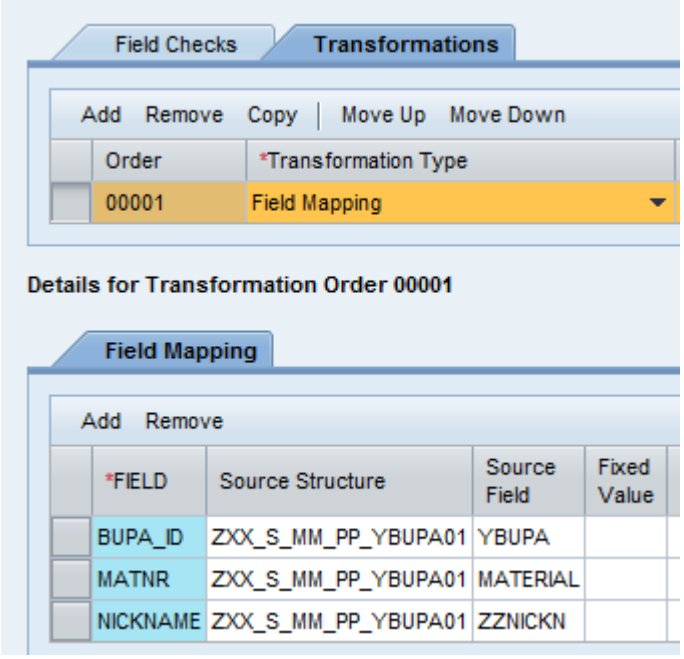
© 2012 SAP AG

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.	

© 2012 SAP AG

6.	Add the key field BUPA_ID as shown.	
7.	Select your mapping step and choose the ***Details*** pushbutton.	
8.	Open the ***Transformations*** tab.	

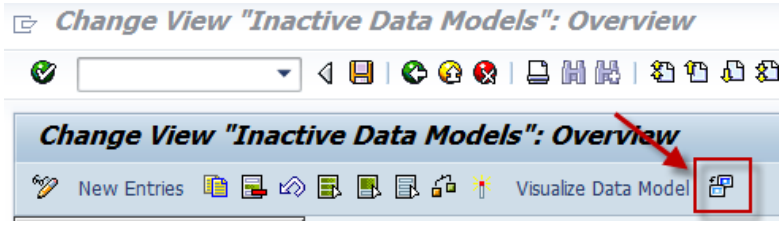
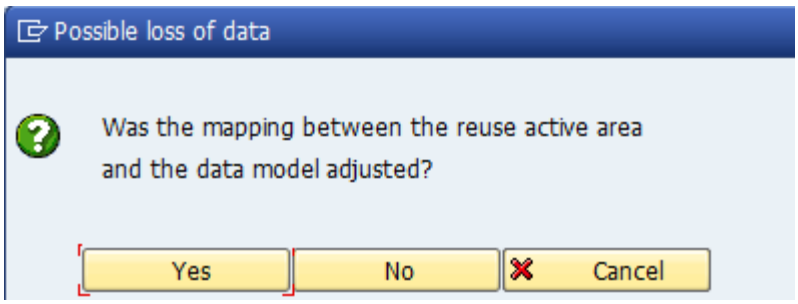
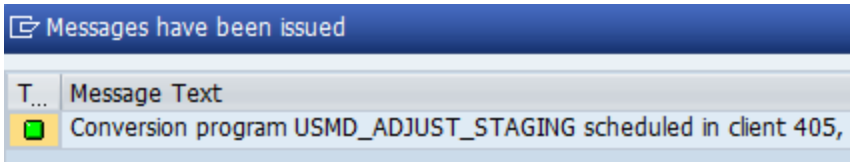
© 2012 SAP AG

9.	Choose the ***Add*** pushbutton to create a new **`Field Mapping`** entry.	
10.	Create the field mapping as shown. Save your changes.	

© 2012 SAP AG

5.4. 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*** (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.	

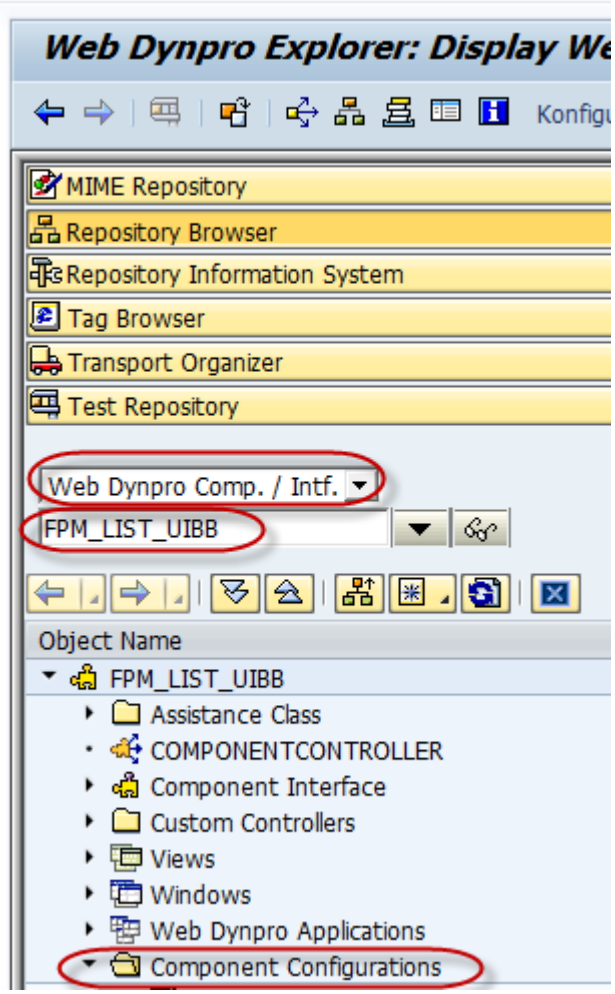
© 2012 SAP AG

5.5. Extend User Interface

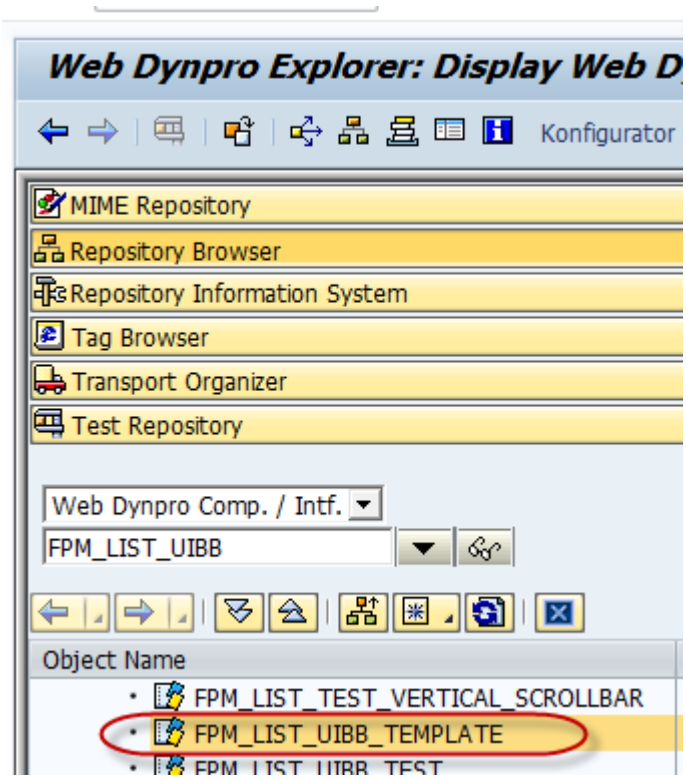
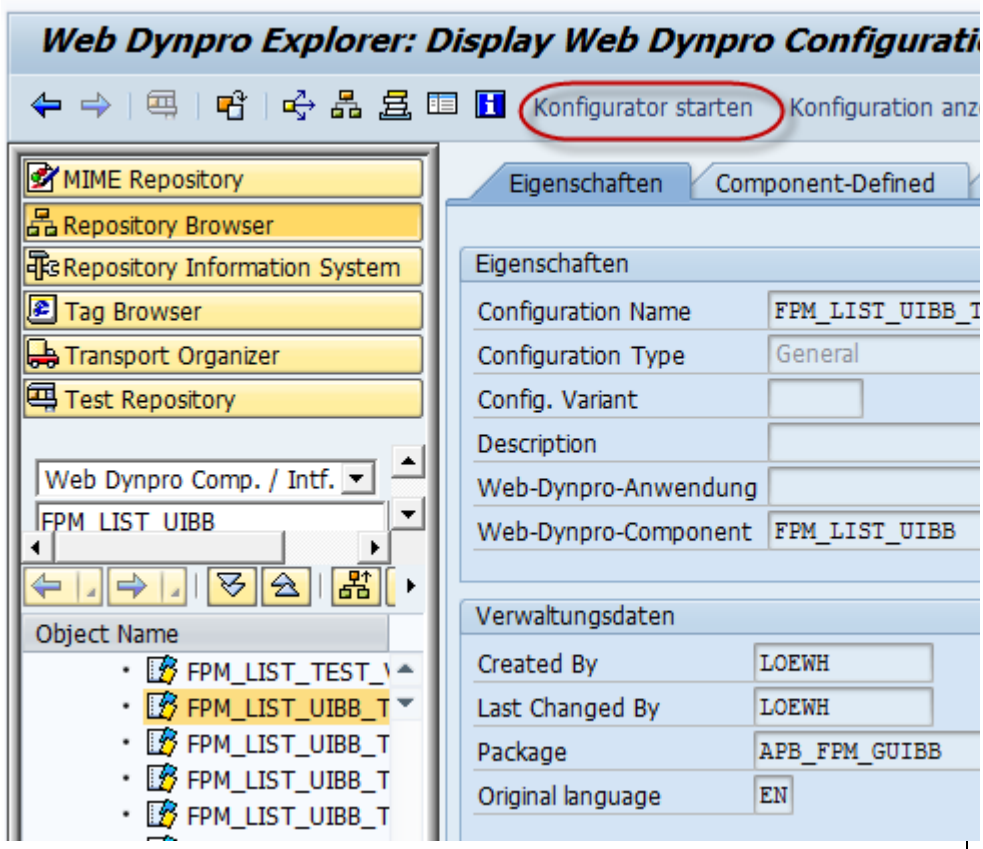
You can follow the described example or you can create an ATS list UIBB instead without a template.

See also extensibility guide for UI: http://scn.sap.com/docs/DOC-30192.

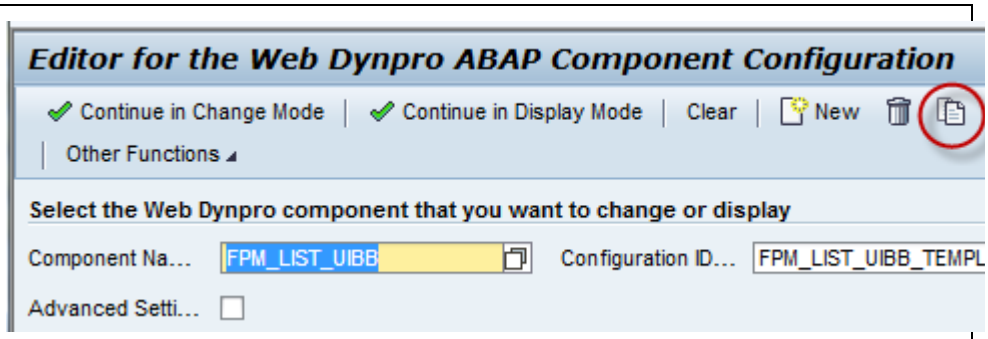
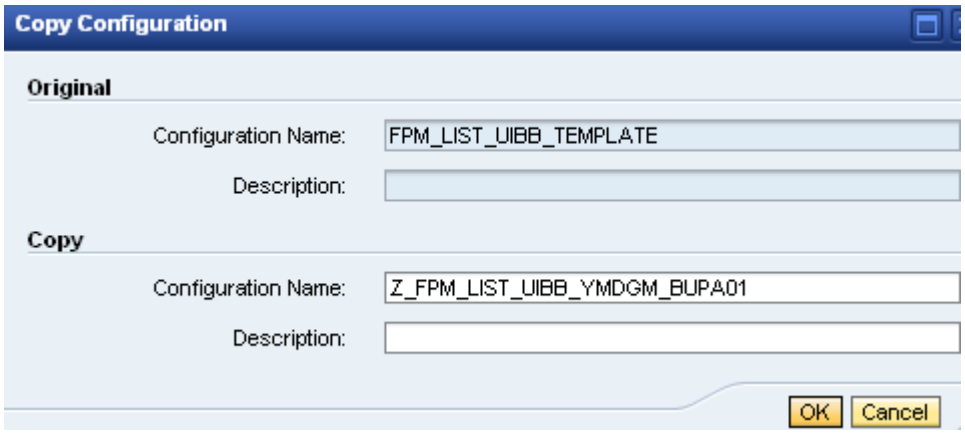
5.5.1. Create FPM List UIBB

1.	Navigate to the Component Configuration as shown in the screenshot.	

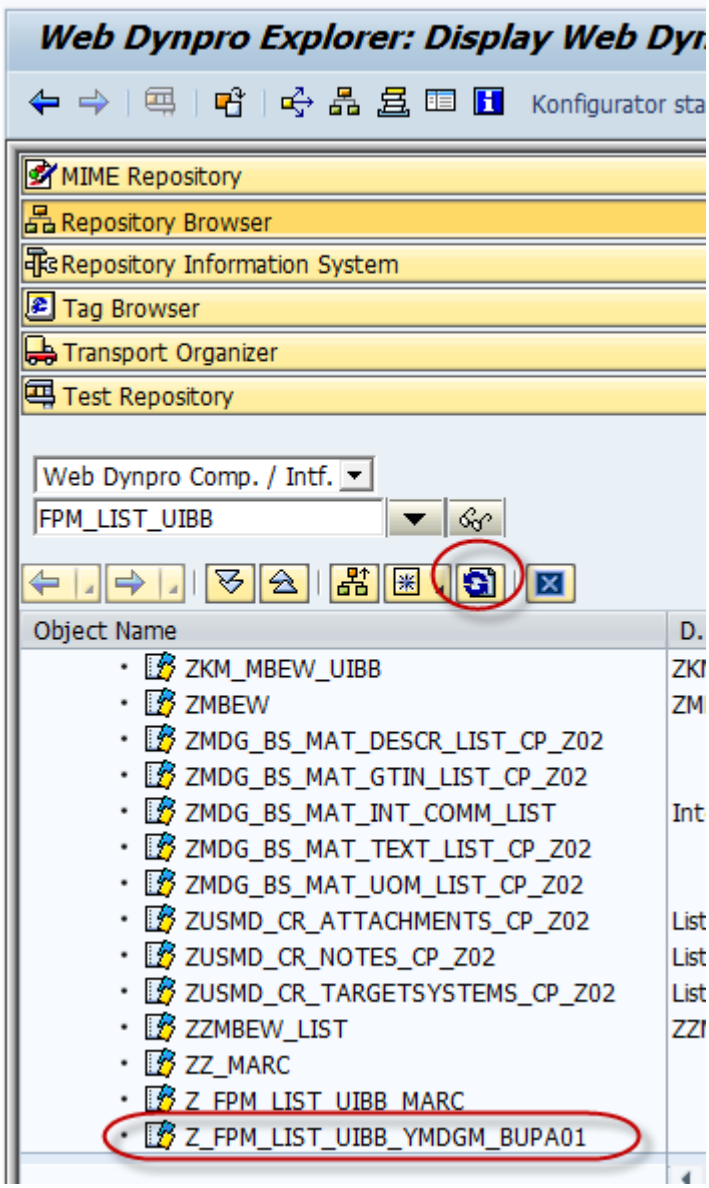
© 2012 SAP AG

2.	Open configuration **FPM_LIST_UIBB_TEMPLATE**.	
3.	Choose the ***Start Configuration*** pushbutton.	

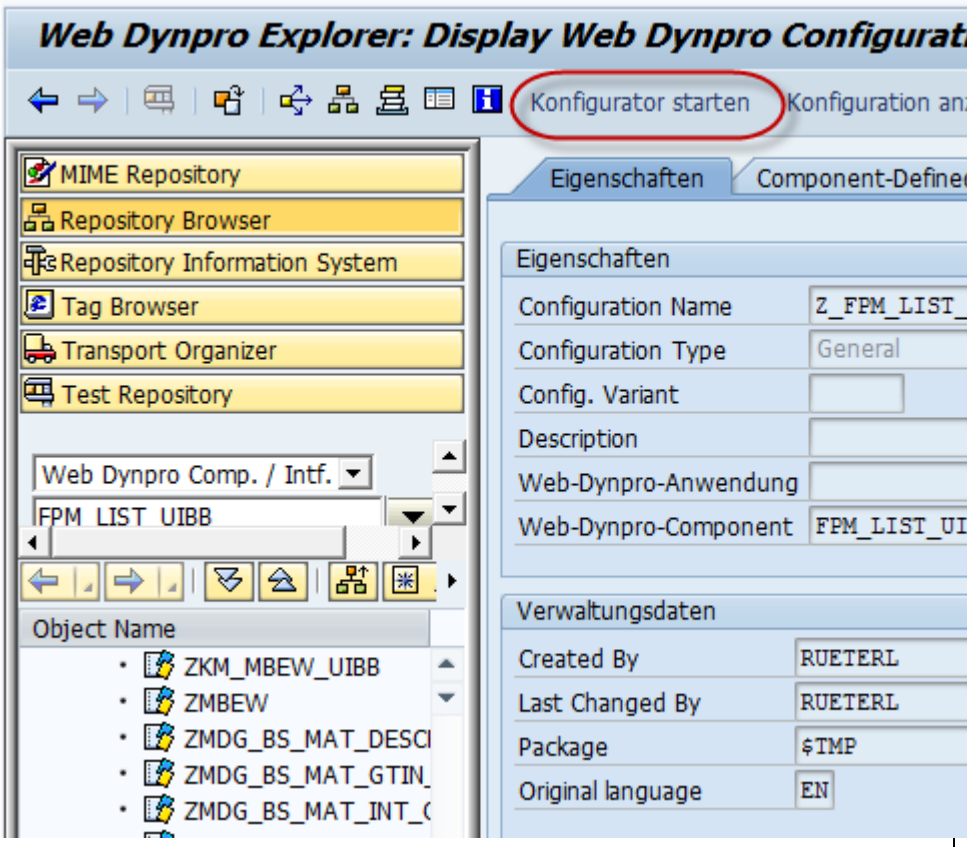
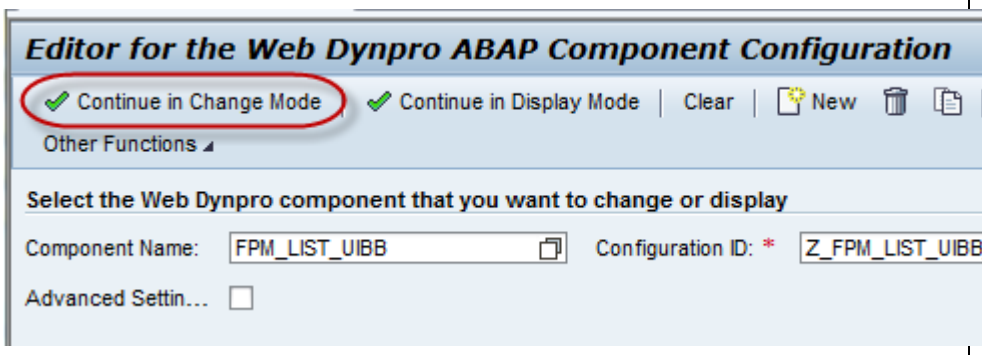
© 2012 SAP AG

4.	Choose the ***Copy*** icon.	
5.	Enter a name for the copy as shown.	

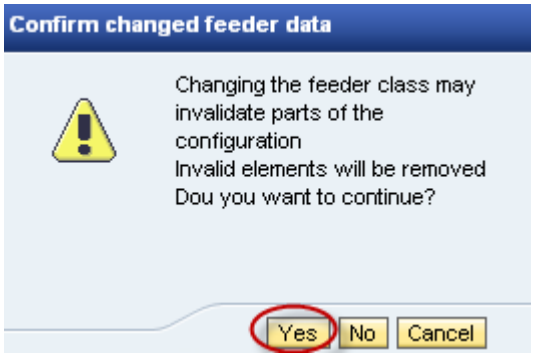
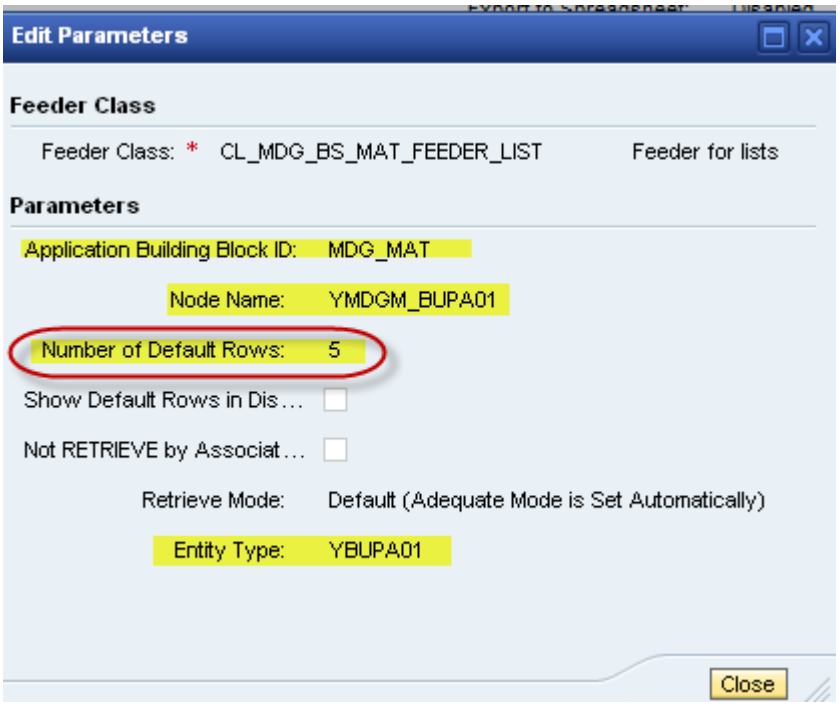
© 2012 SAP AG

6.	Refresh the navigation tree and open the new configuration.	

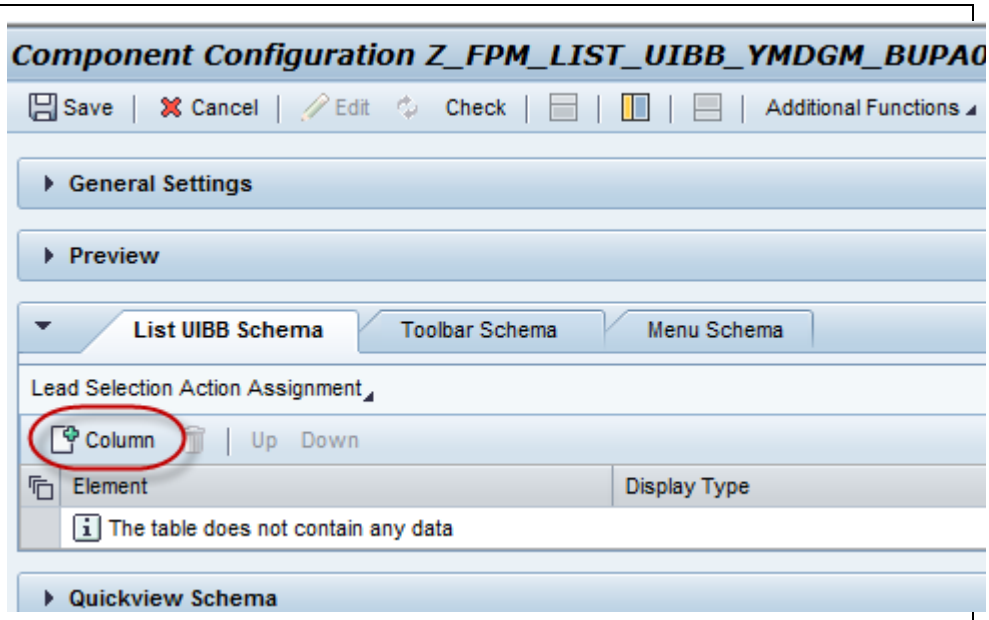
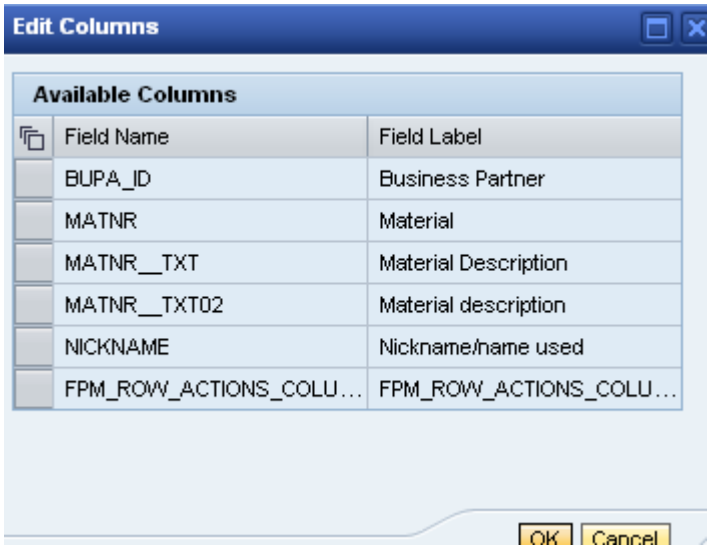
© 2012 SAP AG

7.	Choose the ***Start Configuration*** pushbutton.	
8.	Choose the pushbutton Continue in Change Mode	
9.	Under ***General Settings*** choose the ***Feeder Class*** pushbutton.	

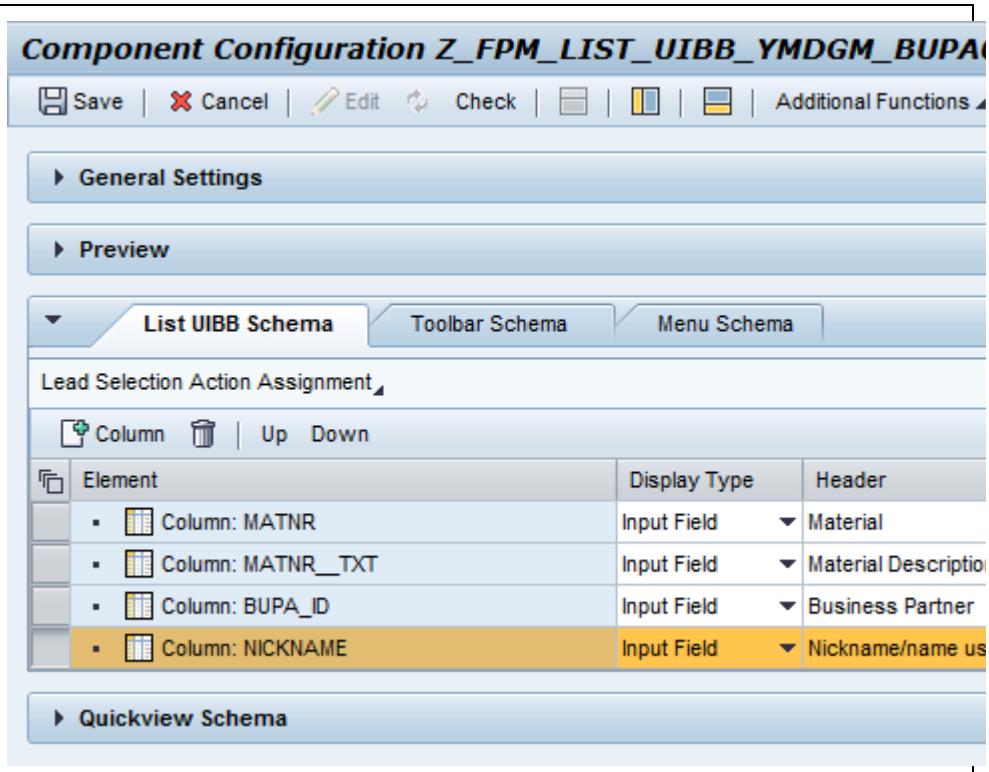
© 2012 SAP AG

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.	

© 2012 SAP AG

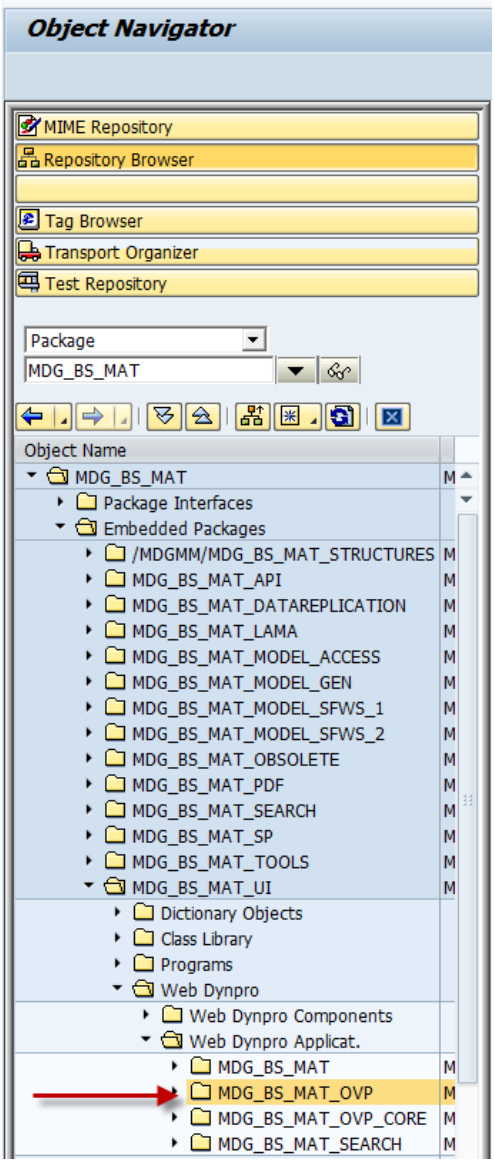
13.	Choose the ***Column*** pushbutton.	
14.	View the list of available columns.	

© 2012 SAP AG

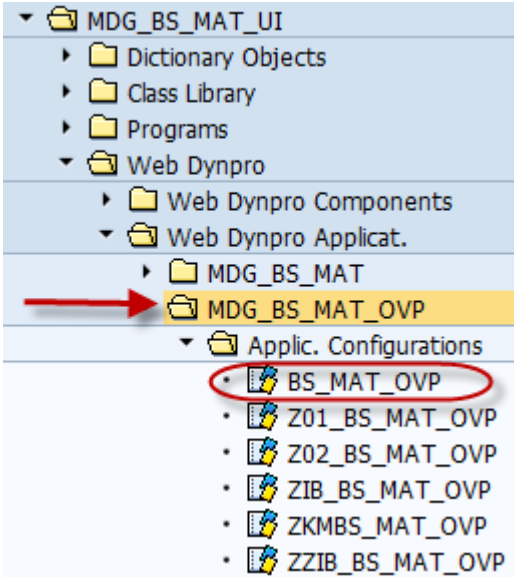
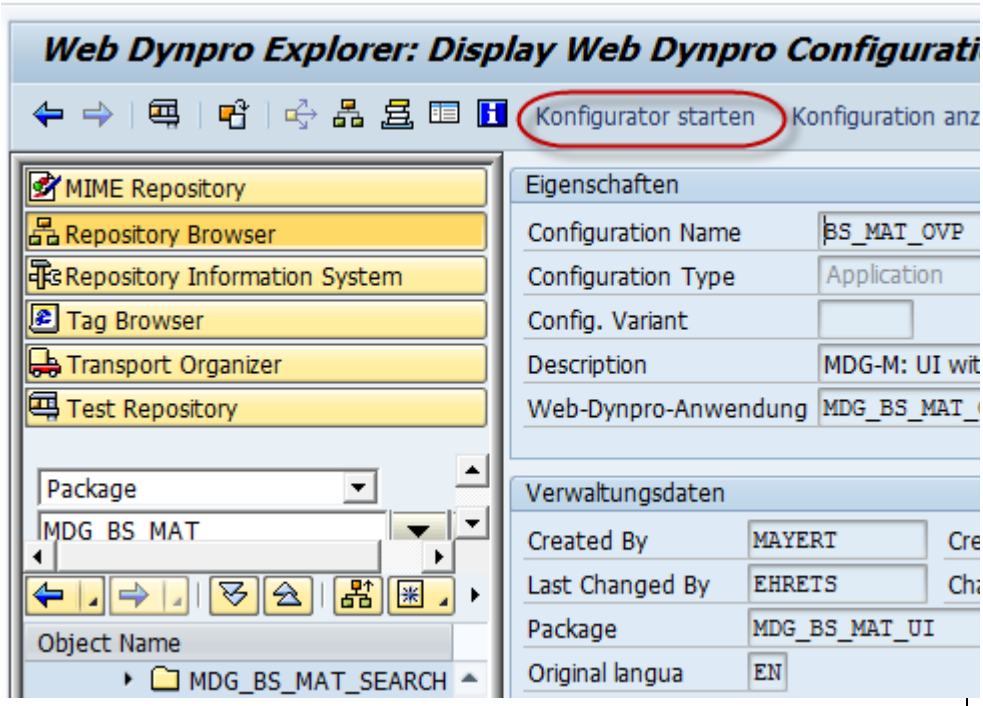
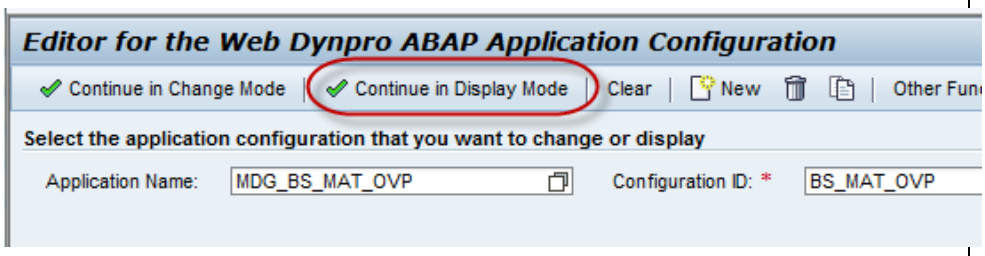
15.	Select the columns from the list of available columns as shown.	

© 2012 SAP AG

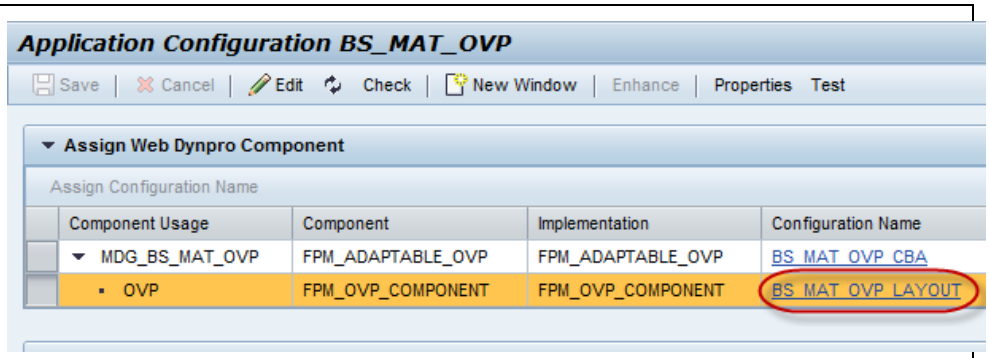
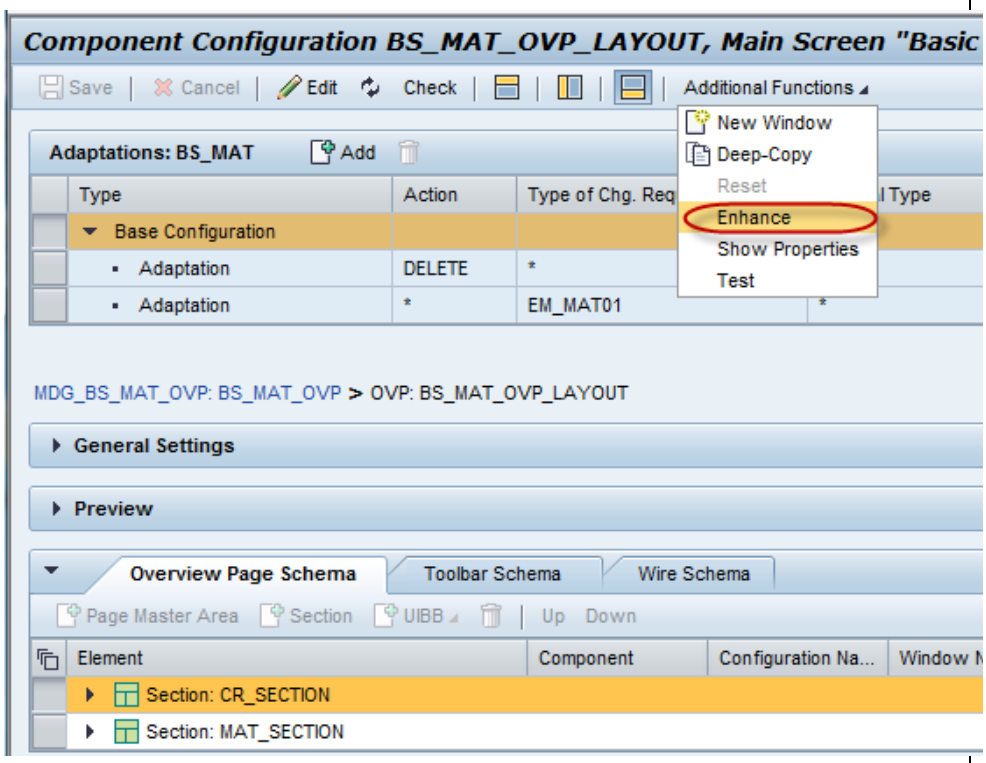
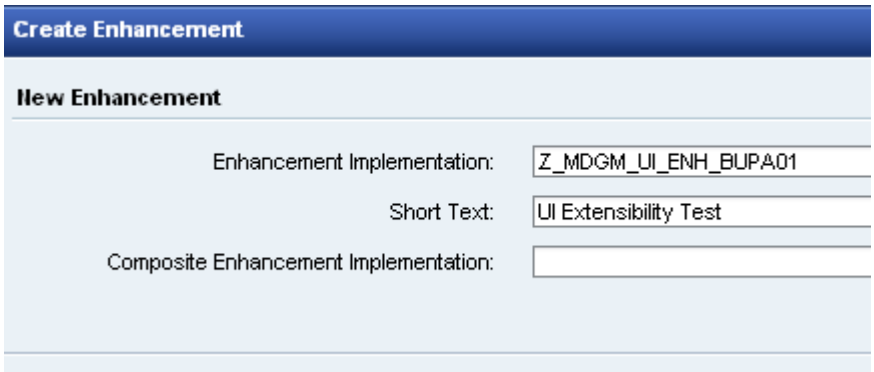
5.5.2. Add List UIBB to Material UI

1.	Navigate to the ***Application Configuration*** as shown.	

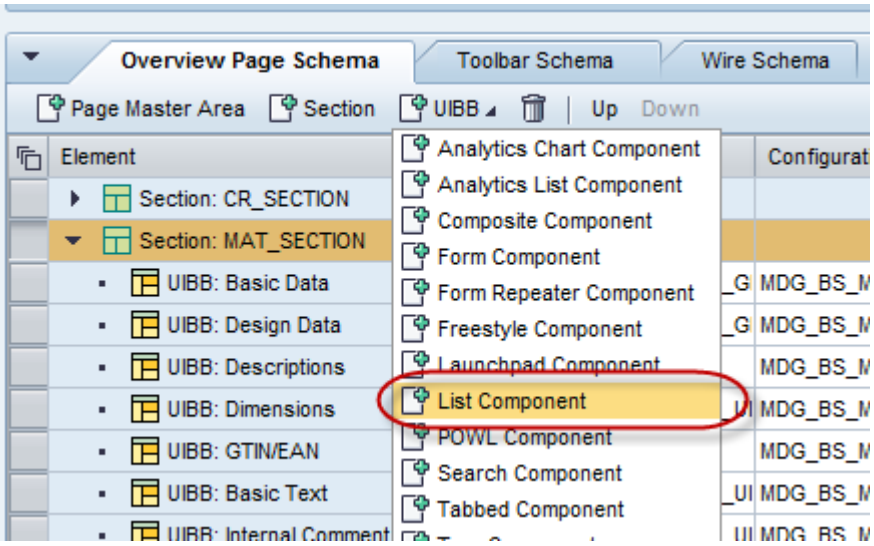
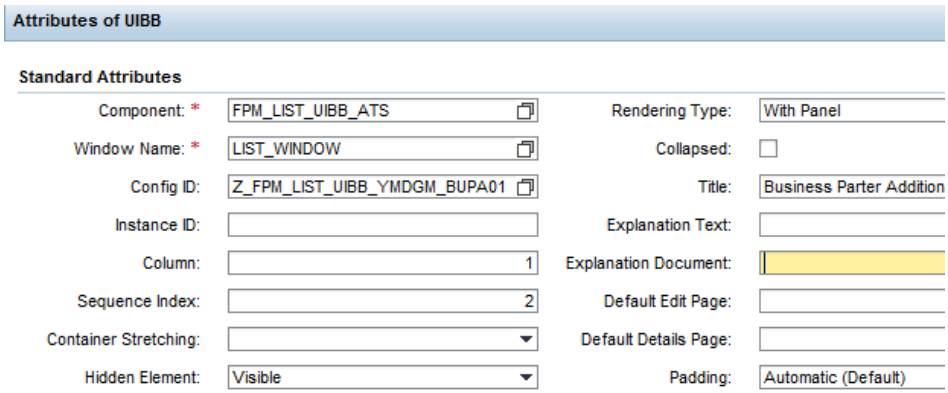
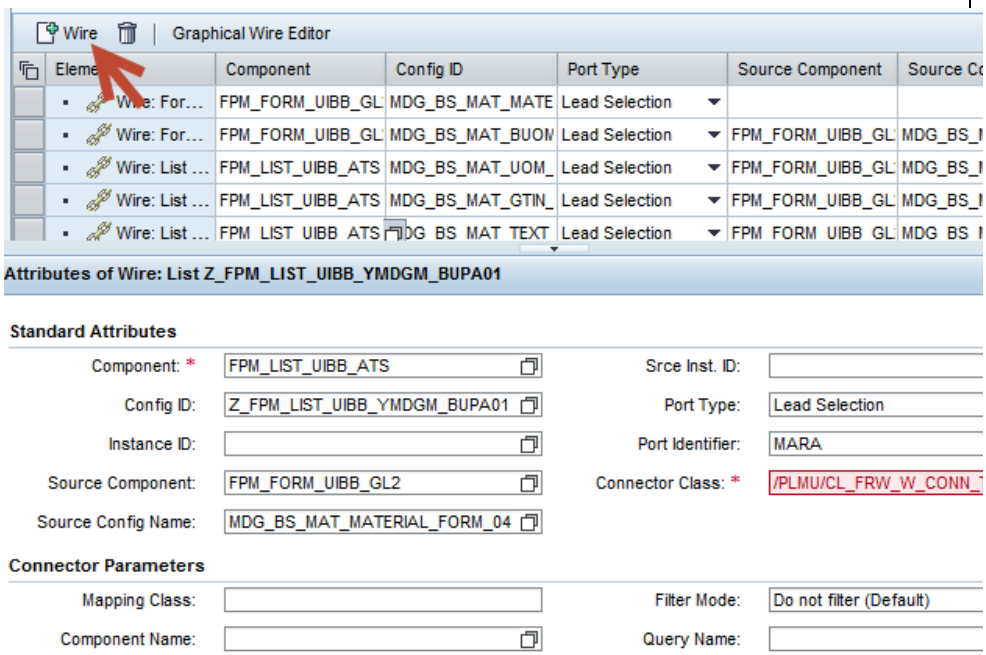
© 2012 SAP AG

2.	Open the configuration **BS_MAT_OVP**.	
3.	Choose the ***Start Configuration*** pushbutton.	
4.	Choose the ***Continue in Display Mode*** pushbutton.	

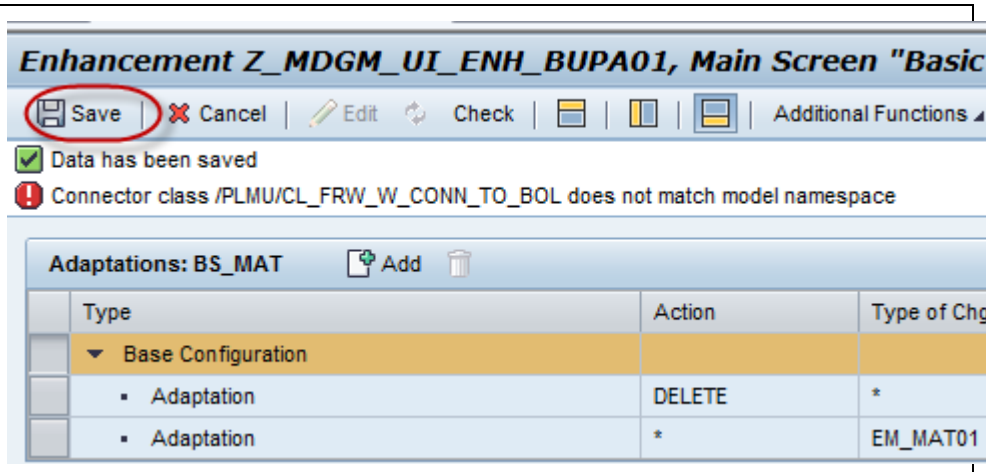
© 2012 SAP AG

5.	Navigate to the configuration **BS_MAT_OVP_LAYOUT**.	
6.	Select ***Additional Functions -> Enhance***. Or you can create a new CBA or you can use customizing mode.	
7.	Provide a name for your enhancement.	

© 2012 SAP AG

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. Use connector class PLMU/CL_FRW_W_CONN_DEFAULT	

© 2012 SAP AG

11.	Save your changes.	

5.5.3. Clear UI Metadata Buffers

After finishing the UI, clear the metadata buffers. You can find the report in customizing Master Data Governance, Central Governance-> Master Data Governance for Material-> Clear UI Metadata Buffers.

This report clears the following buffers in this sequence:

- Text Buffer
- Search Help Buffer
- SMT-Mapping Data Buffer
- SPI Metadata Buffer

You should use this report after extending the data model to make sure that the metadata is consistent with the MDG customizing and UI configuration.

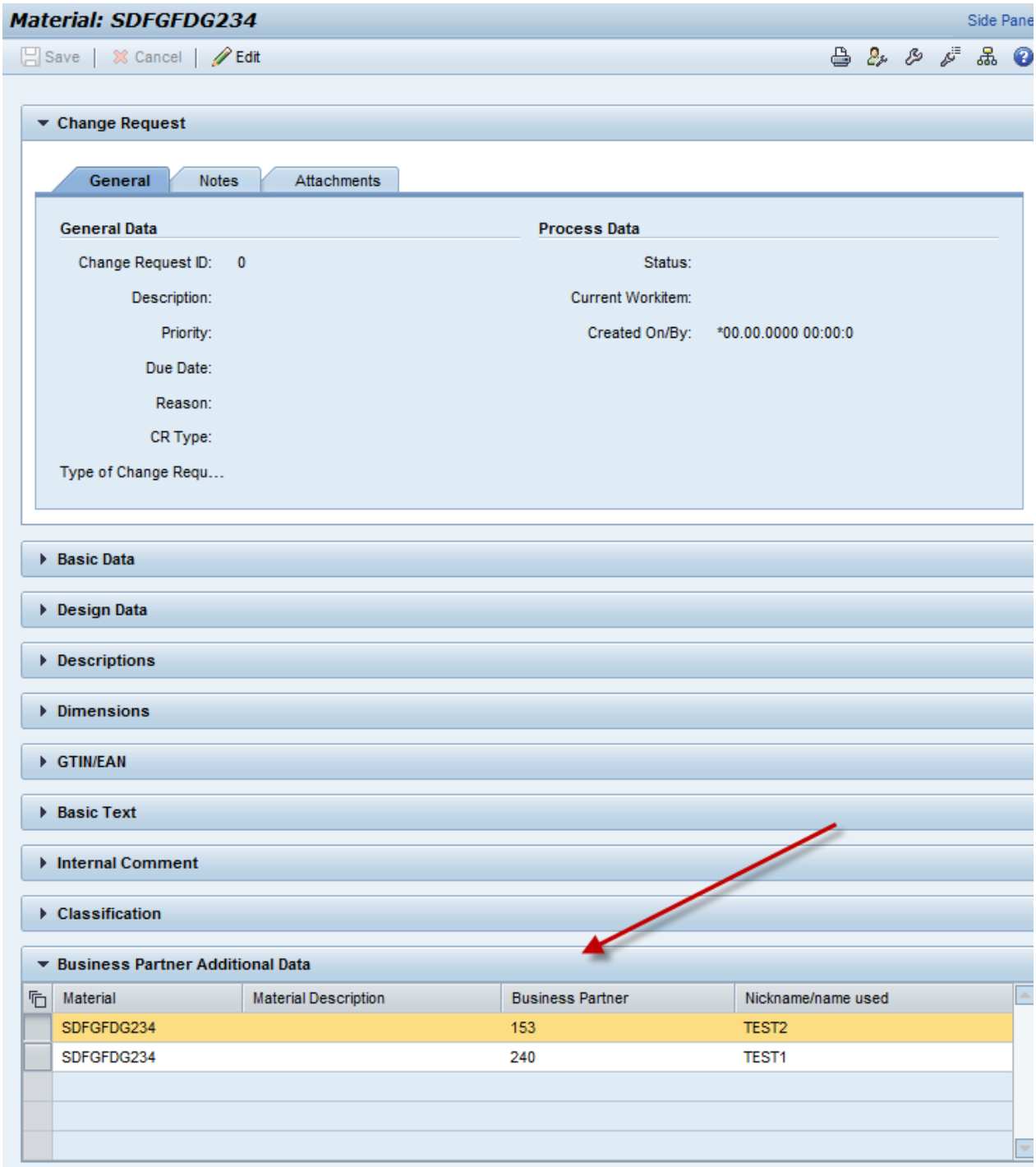
6. Testing Your Data Model Extension

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.

© 2012 SAP AG



7. Related Content

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

© 2012 SAP AG

Copyright

© 2015 SAP SE SE or an SAP SE affiliate company. 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 SE. The information contained herein may be changed without prior notice.

Some software products marketed by SAP SE and its distributors contain proprietary software components of other software vendors. National product specifications may vary.

These materials are provided by SAP SE and its affiliated companies ("SAP SE Group") for informational purposes only, without representation or warranty of any kind, and SAP SE Group shall not be liable for errors or omissions with respect to the materials. The only warranties for SAP SE 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.

SAP SE and other SAP SE products and services mentioned herein as well as their respective logos are trademarks or registered trademarks of SAP SE in Germany and other countries.

Please see
http://www.sap.com/corporate-en/legal/copyright/index.epx#trademark
for additional trademark information and notices.