PUBLIC

How-To: Extend Master Data Governance Material User Interface

Applicable Releases:

From EHP6 for SAP ERP 6.0 and from S/4HANA 1511

Version 4.0

August 2024

Document History

Document Version	Description
1.0	First official release of this guide
1.1	More UI Extensions
1.2	Hide Buttons, Configuration Search UI
1.3	Example coding
1.4	More UI Extensions
1.5	Small corrections
1.6	Chapter 4.3 Post processing after Model Change
1.7	Small corrections
1.8	Chapter 6.5 Launch Material UI with PFCG Menu Entry
1.9	Update chapter 5.5, Update 5.10.4, Small corrections
2.0	Chapter 5.5.3 Hide Fields
2.1	Corrections in chapter 5.7, chapter 5.10.5, chapter 6.8.1, and new chapter 5.3.1 Make CR Type mandatory
2.2	Update of chapter 5.12 and new chapter 6.11
2.3	Update chapter 5.5
2.4	Update chapter 4.2 and chapter 6.1
2.5	Update chapter 5.5, small updates
2.6	Small updates 6.8.1; Chapter 5.12; update 6.8.3
2.7	Chapter 6.1 Performance for UIBB; additional SAP Notes
2.8	Chapter 6.10 Tabbed UI
3.0	General update (September 2021)
3.1	Chapter 4.8 Selection Mode (September 2023)
4.0	SAP Note for Copy Material

1 Business Scenario

SAP Master Data Governance (MDG) provides business processes to find, create, change, and mark master data for deletion. It supports the governance of master data in 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.

This guide describes how to extend the UI of Master Data Governance for Material to display additional data.

Be aware that Pre-, Post-, and Override-Exists must be verified after upgrading to another support package.

2 Additional Data: Stored in Data Model MM

Additional data can be displayed or maintained with an extended UI. The data can be stored in Data Model MM. See also How-to Information for SAP Master Data Governance in chapter Central Governance of Material Data. (https://community.sap.com/topics/master-data-governance/how-to#central-governance-of-material-data).

2.1 Additional Data: Stored in Data Model MM

2.1.1 Fields from Backend (Mapped; Reuse)

Examples:

- Additional Fields (including z-fields) from MARA, MARC etc.

Necessary Steps:

- Extend Model (Reuse entity) including generated structures for ‘Field Properties’ and ‘Reuse Active Area’.
- SMT Mapping (mandatory to PP, SA)
- UI Configuration

2.2 Additional Data: Not Stored in Data Model MM

There are two ways of implementing additional data, which is not stored in the data model MM:

- Feeder Enhancement (UI-related)
- SPI Enhancement (Backend-related)

2.2.1 Feeder Enhancement

Example:

You want to **display** field MCOD1 from table LFA1 in the UIBB of entity MARAPURCH without enhancing the data model MM.

Necessary Steps:

- Create a new feeder. The feeder should inherit from the original feeder, which is used for entity type MARAPURCH. From MDG 7.0 this is class CL_MDG_BS_MAT_FEEDER_FORM_MAT. For earlier MDG releases, the class depends on the notes you have implemented (relevant notes: 1943327, 1940444). Without these notes, class CL_MDG_BS_MAT_FEEDER_FORM_MARA is used. Create a subclass in customer namespace.

- In this class, redefine the method /PLMU/IF_FRW_G_FIELD_DEF~CHANGE_FIELD_DEFINITION. Insert the additional fields in the catalog (internal table CT_DEFINITION).
- Redefine the method /PLMU/IF_FRW_G_AFTER_GET_DATA~AFTER_GET_DATA.
  Check your node name MARA and your entity type MARAPURCH for which the enhancement is valid.
  With MO_CONTEXT->GET_ATTRIBUTE you get any value from the UIBB, then you must read the field MCOD1 from the data base table LFA1 and with MO_CONTEXT->SET_CONTEXT you can write the value to the UIBB.
- Reset the shared memory
  Transaction: MDGIMG
  Path: Master Data Governance for Material→ Clear Shared Memory Buffer
- Change your UIBB “Manufacture Parts” with CBA. Replace the new feeder class in your configuration. Add the new field to the UIBB.
- The extension is only effective in the UIBBs in which the feeder is used.

2.2.2 Enhance SPI Model MDG_MAT

Examples:

- Read data from MARA and show it on the UI
- Input-enabled new field for administrative purposes (completely independent from MDG)

Steps for SPI Enhancement

Definition

- Enhance SPI Metadata with fields that are not stored in data model MM. Use SPI-BAdI (enhancement spot /PLMB/ES_SPI: BADI /PLMB/EX_SPI_METADATA method ENRICH_NODE_DEFINITION) to enhance the list of nodes for getting the additional fields.
- To check if the node is added to the meta data of the SPI as expected, use transaction MDB (Metadata Browser) with the ABB-ID ‘MDG_MAT’.
- To display the new fields, enhance the UI configuration with a new UIBB. Depending on the requested features, an existing PLM UI feeder (e.g., /PLMU/CL_FRW_G_FEEDER_FORM) or a MDG Material feeder (For example: CL_MDG_BS_MAT_FEEDER_FORM) can be reused. If their features are not sufficient, you must implement your own feeder (if possible, it should inherit from one of the existing ones).
- Include the new UIBB in the used UI configuration(s) and adapt the wiring accordingly. In case the wiring doesn’t exist or is incorrect, the new UIBB will remain empty and all its fields will be read-only.

Runtime

- Use SPI enhancement spot /PLMB/ES_SPI with BADI /PLMB/EX_SPI_APPL_ACCESS and implement method BEFORE_RETRIEVE ‘Before Service Provider RETRIEVE call’ to skip the standard retrieval for the added node.
- Use SPI enhancement spot /PLMB/ES_SPI with BADI /PLMB/EX_SPI_APPL_ACCESS and implement method AFTER_RETRIEVE ‘After Service Provider RETRIEVE call’ to determine the data to be displayed. If the method returns at least one empty entry, then the field is input-enabled. If the method returns nothing, then the fields are empty and read-only.
- Optional: Use SPI enhancement spot /PLMB/ES_SPI with BADI /PLMB/EX_SPI_PROPERTIES_ACCESS and implement method GET_PROPERTIES ‘Get Properties call’ to change the field property, for example to read-only.
- Optional: Use SPI enhancement spot /PLMB/ES_SPI with BADI /PLMB/EX_SPI_APPL_ACCESS and implement method BEFORE_UPDATE ‘Before Service Provider UPDATE call’ to disable standard coding and use SPI enhancement spot /PLMB/ES_SPI with BADI /PLMB/EX_SPI_PROPERTIES_ACCESS and implement method AFTER_UPDATE ‘After Service Provider UPDATE call’ to save the field value .
- UI Configuration Usage: Custom UIBB which could be used together with MDG_M UIBBs on OVP, detail screen or pop-up.

Storage:

- Not in staging
- Independent of activation
- Customer or backend tables

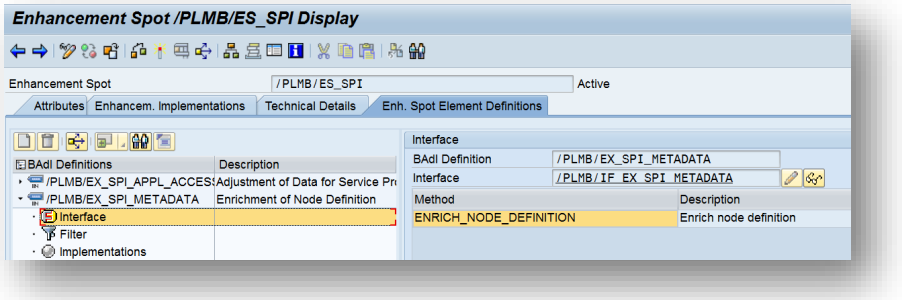
Example of a New Node

To extend the UI with fields that are not in the data model, you have to introduce new nodes in the context at design time and runtime. Usually, the nodes are derived from the entities of the data model; here, we define them in SPI.

This is done in a BAdI extension of the PLM UI framework.

Enhancement spot: /PLMB/ES_SPI

BAdI definition: /PLMB/EX_SPI_METADATA



In the implementing class, add the following example coding:

Method /PLMB/IF_EX_SPI_METADATA~ENRICH_NODE_DEFINITION

```
  DATA: ls_node LIKE LINE OF ct_metadata_node.

  READ TABLE ct_metadata_node
    INTO ls_node
    WITH KEY name = 'MARA'.
  IF sy-subrc IS INITIAL.

    ls_node-name                       = 'YYCUSTOM'.
    ls_node-name_parent                = 'MARA'.
    ls_node-transient                  = abap_true.
    ls_node-data_struc                 = 'YCUSTOM_STR'.
    ls_node-update_relevant            = abap_false.
    ls_node-supported_operation_group = 06.
    CLEAR: ls_node-data_description,
           ls_node-actions[],
           ls_node-update_sideeffect.
    ls_node-supported_properties-fields-insert                = abap_false.
    ls_node-supported_properties-fields-transient_data_record = abap_true.

    INSERT ls_node INTO TABLE ct_metadata_node.

  ENDIF.
```

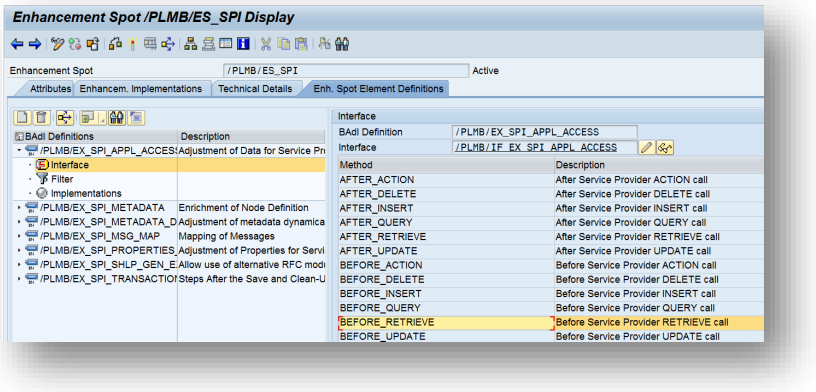
In this example, YYCUSTOM is the new node (sub-node below MARA), and YCUSTOM_STR is the structure containing the fields to be displayed. To set the parameters in a simple way, we copied the entry from node MARA and modified it accordingly.

SAP Notes 1899613 and 1973708 may be necessary to avoid the exception /PLMB/CX_SPI_ERROR because of the missing action ‘SET_ENTITY’.

After this step, the new node YYCUSTOM is visible in the Web Dynpro Floor Plan Manager (FPM), and the fields of the structure are available for designing the UI (UIBB).

The second enhancement is for the provision of data at runtime:

Enhancement spot: /PLMB/ES_SPI, BAdI definition: /PLMB/EX_SPI_APPL_ACCESS



Method: /PLMB/IF_EX_SPI_APPL_ACCESS~BEFORE_RETRIEVE

```
IF iv_node_name = 'YYCUSTOM'.
  cv_skip_standard = abap_true.
ENDIF.
```

This disables the execution of the standard retrieval for the added node.

Method: /PLMB/IF_EX_SPI_APPL_ACCESS~AFTER_RETRIEVE

```
DATA: ls_values  TYPE YCUSTOM_STR,
      ls_node_id TYPE matnr,
      lt_values  TYPE ycustom_delta_tt,
      lr_values  TYPE REF TO YCUSTOM_STR.

IF iv_node_name = 'YYCUSTOM'.
READ TABLE it_node_id INDEX 1 INTO ls_node_id.
  ls_values-matnr = '1'.
  ls_values-name  = 'NAME1'.
  ls_values-value = 'VALUE1'.
  APPEND ls_values TO lt_values.
  INSERT LINES OF lt_values INTO TABLE ct_node_data.

ENDIF.
```

The key (MATNR) is provided in importing table IT_NODE_ID. This example contains the structure of the node with the fields NAME and VALUE.

You can now display the new fields in a new UIBB or in a popup (see chapter 5 UI Extension).

Hint: The transaction MDB provides a good overview of MDG_MAT.

The customer node added in this example is intended to display additional data to an already existing node or entity. Here additional data on the material level shall be displayed.
In case you need the customer node to create new entries in the governed data (for example new plant assignments: the customer node would list all plants currently not assigned to the material), the implementation and check for the correct node will be different (e.g. via IV_TARGET_NODE_NAME).

2.2.3 Additional Fields (Not Mapped)

Additional fields could be helper fields, info fields, and calculated fields.

Temporary Fields (Transient)

Enhancement of single fields.

Examples:

- Calculate volume of dimensions of alternative UoM and show result on UI. Calculation of volume is done in BRF+ or BAdI Derivation (BAdI USMD_RULE_SERVICE, Method DERIVE_ENTITY Derivation of Data for a Master Record)
- Net weight for alternative UoM. Calculation of net weight is done in BRF+ or BADI Derivation

Storage:

- Only in staging
- Value lost after activation of material

Necessary Steps:

- Extend data model (Reuse Area blank)
- UI configuration in an existing UIBB or a new UIBB
- Field property. The standard SAP behavior of field property is input-enabled. To change the field property, use enhancement spot usmd_acc_fld_prop_cust_dep_set (Method MODIFY_FLD_PROP_ATTR Change Field Attributes). The prerequisite SAP Note is 1706227.

SAP does not recommend this type of data modeling because it contradicts the concept of governance (value loss, mapping problems). You should use the Flex Entity or see chapter 4.2.

Permanent Fields (Flex Entity)

Enhancement of a set of fields and/or a complete entity.

Examples:

- Custom Organization Hierarchy
- Custom Material Hierarchy
- Fields for calculating material description

Necessary Steps:

- Extend data model (new flex entity; Reuse Area = MDG) including generated structures for ‘Field Properties’ and ‘Reuse Active Area’
- UI Configuration using a new UIBB
- Add wiring details (prerequisite SAP Note is 1749938)
- For flex entities with a 1:1 relationship to a MARA entity (for example, the entity MATERIAL), to use the mapping functions you have to add the mapping class to the wiring definition in your component configuration:
    - Open the OVP component configuration
    - Mark the wiring entry of the configuration that contains the flex entity
    - Enter "CL_MDG_BS_MAT_MATNR2MATERIAL" (new mapping class) in the field Mapping Class at the wiring attributes and choose 'Save'
- Field property. The standard SAP behavior of field property is input-enabled. To change field property, implement enhancement spot usmd_acc_fld_prop_cust_dep_set (Method MODIFY_FLD_PROP_ATTR Change Field Attributes or MODIFY_ENTITY_PROPERTIES Change Entity Properties). The prerequisite SAP Note is 1706227.

Storage:

- In staging

- Kept after activation

2.3 Mandatory Postprocessing after Model Change

After you have changed the Data Model or the SPI model you must run these two reports after extending the model to avoid errors (like the language fields in the UI do not show the corresponding descriptions):

- Report MDG_BS_MAT_METADATA_INVALIDATE
- Report MDG_BS_MAT_TEXT_DELETE

You can find the reports in the Customizing settings under *Cross-Application Components →Processes and Tools for Enterprise Applications →Master Data Governance→Master Data Governance for Material→ Clear Shared Memory Buffer and Delete Text Mapping Information*.

You must execute these IMG activities not only in the development system, but also in all system that is supplied with these changes (for example the test and production systems) after the transport is imported.

Relevant SAP Notes:

1669963 No description for language field (Report MDG_BS_MAT_TEXT_DELETE)

1793913 Performance: Buffer for UI Metadata (MDG_BS_MAT_METADATA_INVALIDATE)

1821810 Adding Shared Memory report to IMG

1824036 Inserting Delete Text Mapping Information report

3 UI Extension

For the extension of the model and SMT, see the How To Guides https://community.sap.com/topics/master-data-governance/how-to#central-governance-of-material-data .

See more information about Floorplan Manager for Web Dynpro ABAP | How to Adapt FPM.

There are different ways to influence the UI of MDG-M:

- **Copy** delivered application configuration and dependent configuration (deep-copy)
  - CBA possible (also custom CBA schema)
  - Updates from SAP do not affect the UI.
  - More information can be found in chapter 3.2 Custom UI Configuration

- **Copy** delivered application configuration and CBA container, but use delivered UI configuration layout.
  - CBA possible (also custom CBA schema)
  - Updates from SAP affect the reused UIBBs

- **Enhance** application configuration and UI configuration.
  - CBA possible (but no custom CBA schema)
  - Updates from SAP affect the reused UIBBs.

- Use delivered application configuration and make only **Customizing** settings.
  - Context-based adaptation (CBA) is not possible.
  - Updates from SAP affect the reused UIBBs

- Use delivered application configuration and make only **Configuration/Modification** settings.
  - CBA possible (but no custom CBA schema)
  - Updates from SAP affect the reused UIBB's.

- **Context-Based Adaptations**
  - This is an adaptation concept that allows you to have adaptations based on the runtime environment.
  - More information can be found in chapter 3.1.1Context-Based Adaptation (CBA)

	Copy Configuration	Enhancement	Customizing	CBA
Description	Just copy (reuse of UIBBs possible); Co-exists with CBA	In Configurator (upper right corner, menu 'additional functions'); Like Enhancement in SE80 --> "Unconditional CBA"	SAP-CONFIG-MODE=X; Add new fields to UI from the entity assigned to UIBB, new UIBBs (asked for structure and entity) Co-exists with everything	Rather simple; Co-exists with "copied configuration"
Criteria	MDG customizing: UI per CR type and WF step	n/a	n/a	CBA criteria (e.g., CR type, material type); can be enhanced, see SAP Note 1606341
Dynamic	n/a	n/a	n/a	For fields
Downside	Redundancy – manually align	Performance	Not recommended for major changes	Transparency of decision tree why which CBA is used
Transport	Cross-client --> workbench request	Cross-client --> workbench request	Client specific --> customizing request	Cross-client --> workbench request

3.1 Delivered Application Configuration

Package: MDG_BS_MAT_UI

Web Dynpro Application: MDG_BS_MAT_OVP

Application Configurations: BS_MAT_OVP_xx and BS_MAT_OVP_xxH

3.1.1 Context-Based Adaptation (CBA)

The option to adapt applications without the need for modification is one of the main benefits offered by FPM. The possibilities of Web Dynpro Customizing in combination with the Enhancement Framework already provide far more options than the classic Dynpro in this respect. With CBA, the UI can be adapted depending on runtime parameters.

Note: CBA cannot be used for the UIBB of Classification because it is implemented with Web Dynpro bypassing FPM.

Adaptation Schema

The adaptation schema is a list of dimensions (or characteristics) that can be used for adaptations. The adaptation schema used for material is BS_MAT. The delivered dimensions are: Action, Change Request Type, Material Type, and Workflow Step.

The adaptation schema and its dimensions can be maintained using transaction SM34 with view cluster FPM_VC_ADAPT_SCHEMA.

- If you need more or other dimensions to influence your governance UI create a new adaptation schema (probably as copy of BS_MAT)

- To use the new schema, you need to copy the original application configuration (for example BS_MAT_OVP_03). Create a new configuration (in standard BS_MAT_OVP_CBA) of the component FPM_ADAPTABLE_OVP and assign in this configuration your own adaption schema to it

Dimensions with values already specified at the call (in the URL) can be used to influence the layout on floorplan level (for example configuration BS_MAT_OVP_LAYOUT_03). Therefore, they have to be defined as application parameters (here WD application MDG_BS_MAT_OVP). On this configuration level (floorplan) they are only evaluated once at the start of the application.

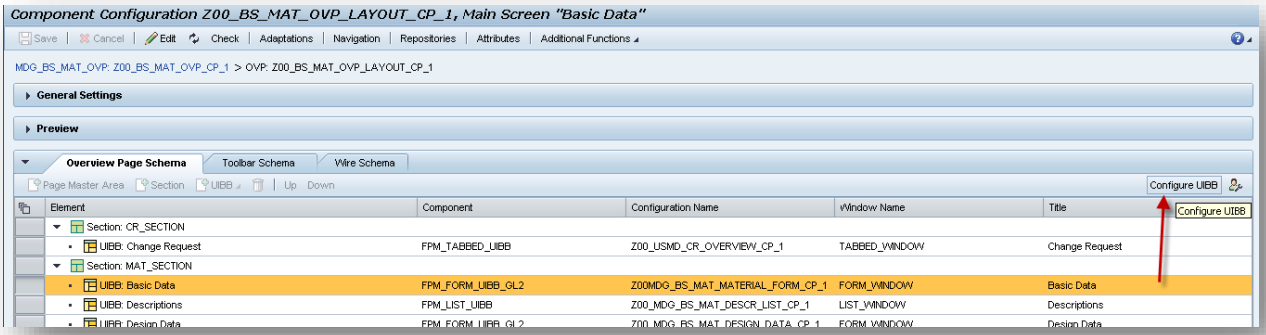
If the already started OVP should be adapted dynamically to values entered by the user, the configurations have to be defined on UIBB level (as adaptation of the component configurations).

Action and mostly CR-Type are determined by URL, therefore these parameters can be used for adding/hiding UIBBs and adding/hiding fields in UIBBs. Material Type and Workflow Step are in standard not in the URL, therefore only adding/hiding fields in UIBB is possible.

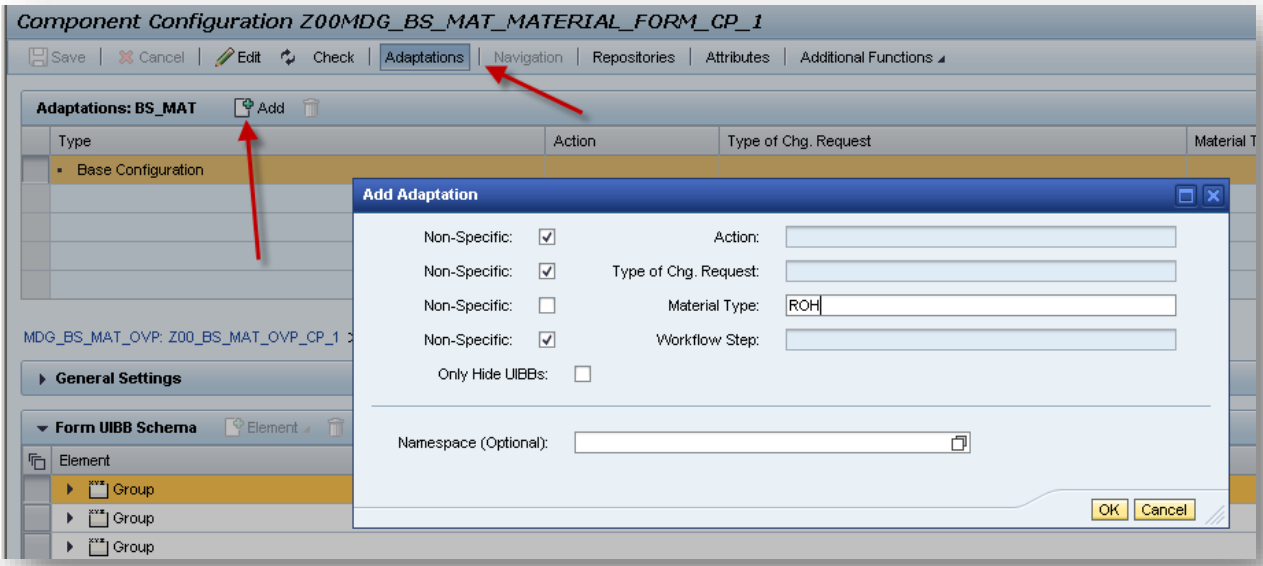
If you want to use your own dimensions, it is necessary to publish their values at the start of the application and at every change of the values. You can use the method VALUE_WAS_CHANGED of the class CL_MDG_BS_MAT_ASSIST_CBA. Please see method /PLMU/IF_FRW_G_AFTER_GET_DATA~AFTER_GET_DATA of the feeder class CL_MDG_BS_MAT_FEEDER_FORM as an example.

In case the origin of the value is a new UIBB, you can publish the value in the corresponding feeder in a comparable way. An alternative is an enhancement of the previously mentioned method of the feeder class.

Example of Context-Based Adaptation (CBA)
You want to adapt the UI based on data you entered. You want to hide the fields for group Configuration on Basic Data UIBB for material type **ROH**. Go to your UIBB for Basic Data.
Important: Start from the Application Configuration, otherwise the button for ADAPTATION will not be available.



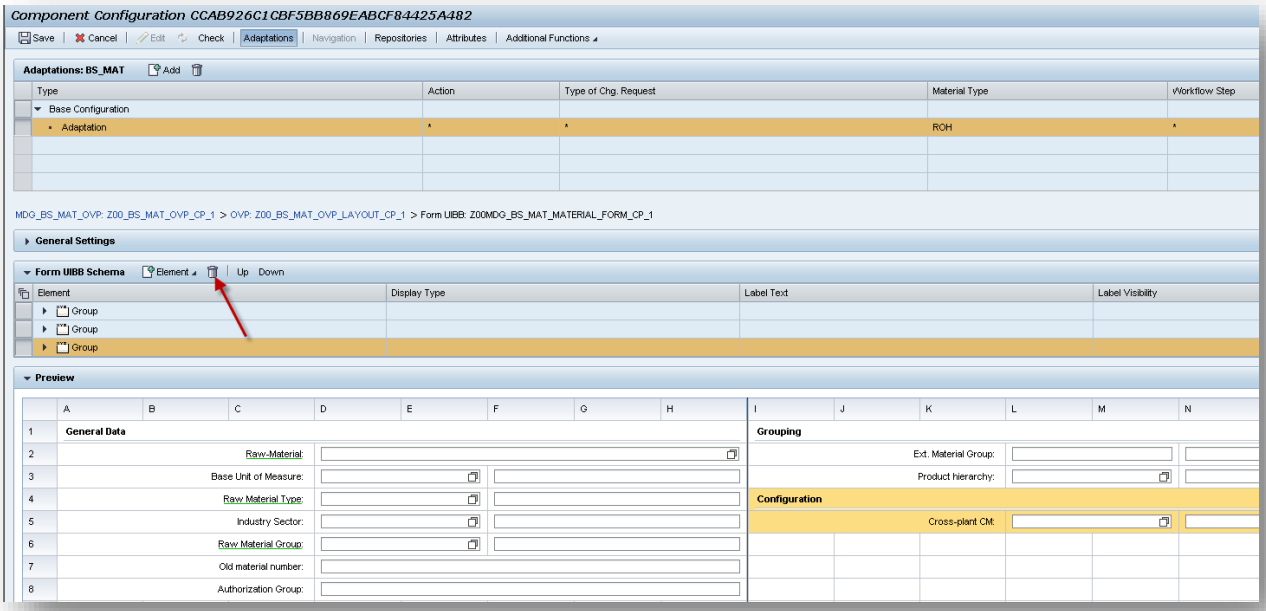
Create an adaptation for material type **ROH**. Click on **Adaptations**. Click on the Add button. Deselect Non-Specific for Material Type and enter ROH.



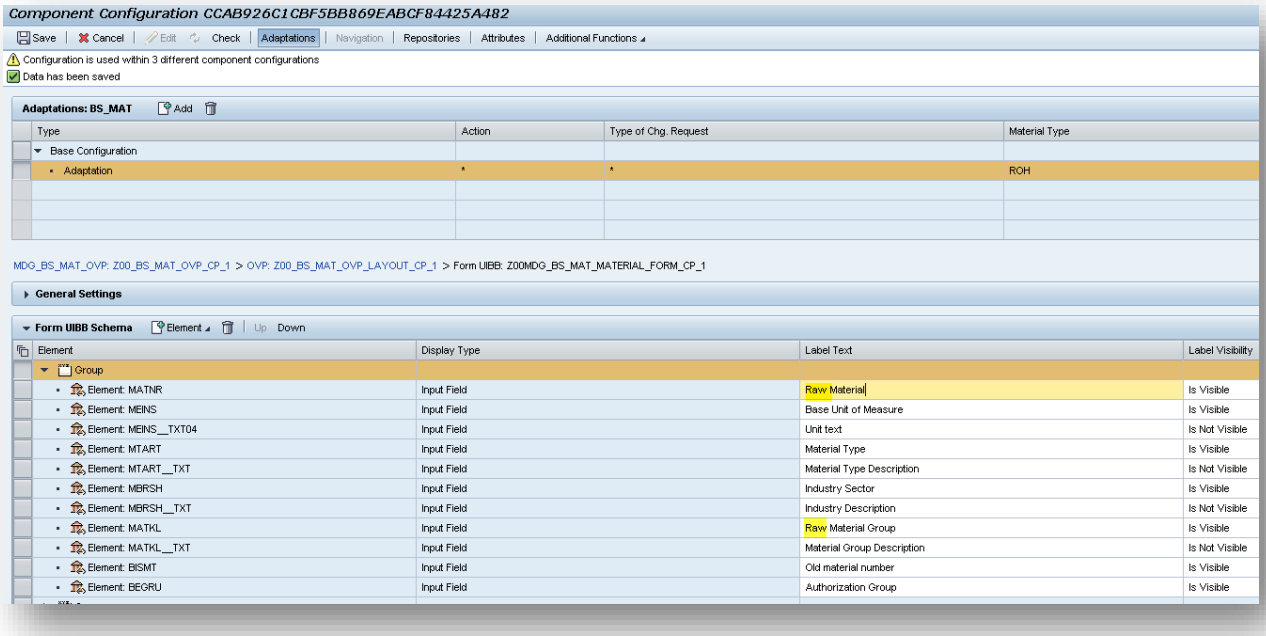
A popup with details shows the possible “dimensions” for the SAP-provided CBA schema. There are four possible parameters: Action, Type of CR, Material Type, and Workflow Step. You can extend this set of “CBA dimensions” with more fields.

Important: fill in a customer namespace to avoid SAP object registration. After entering the namespace, you have to assign the CBA to a package. Then enter a transport request.

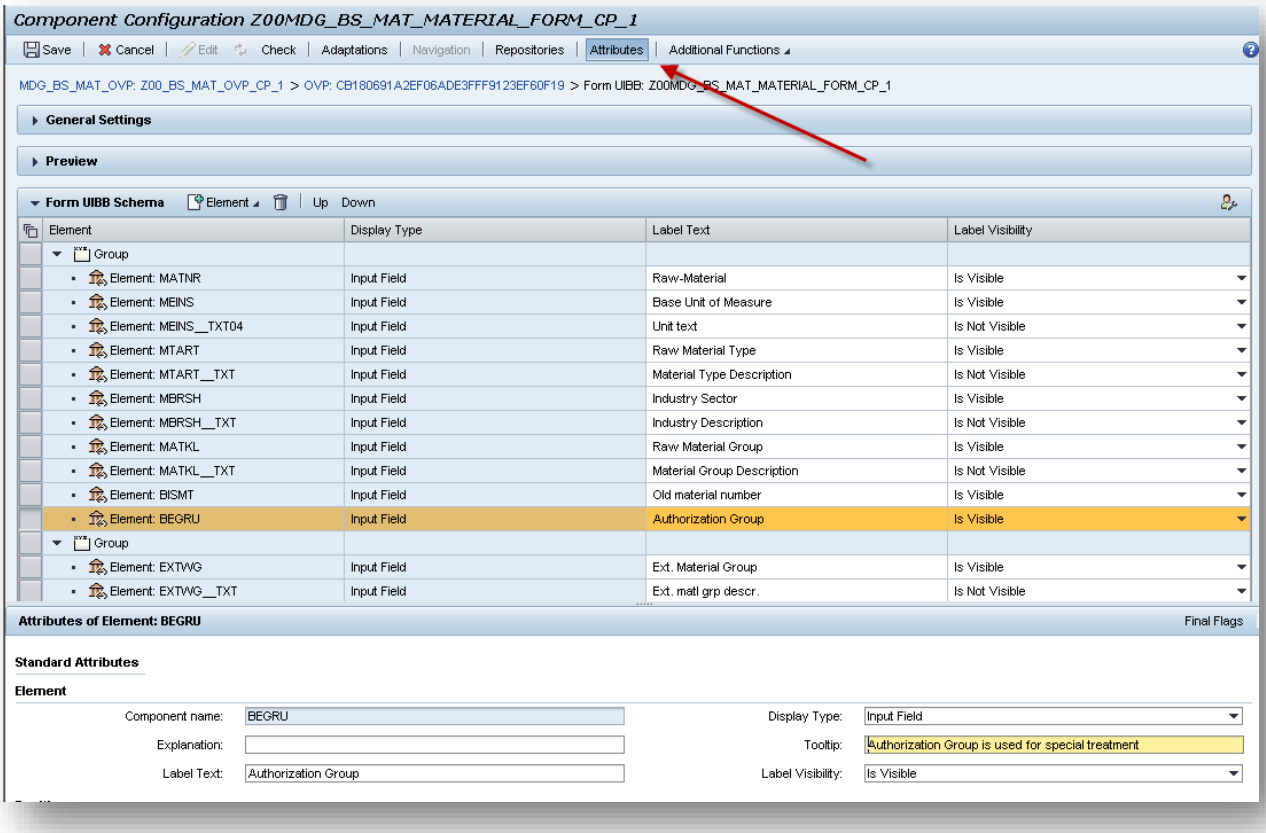
Mark sure that your adaptation is marked. Now you can adapt the UI. Choose Edit.
Delete the group for Configuration.



Change Label Text.

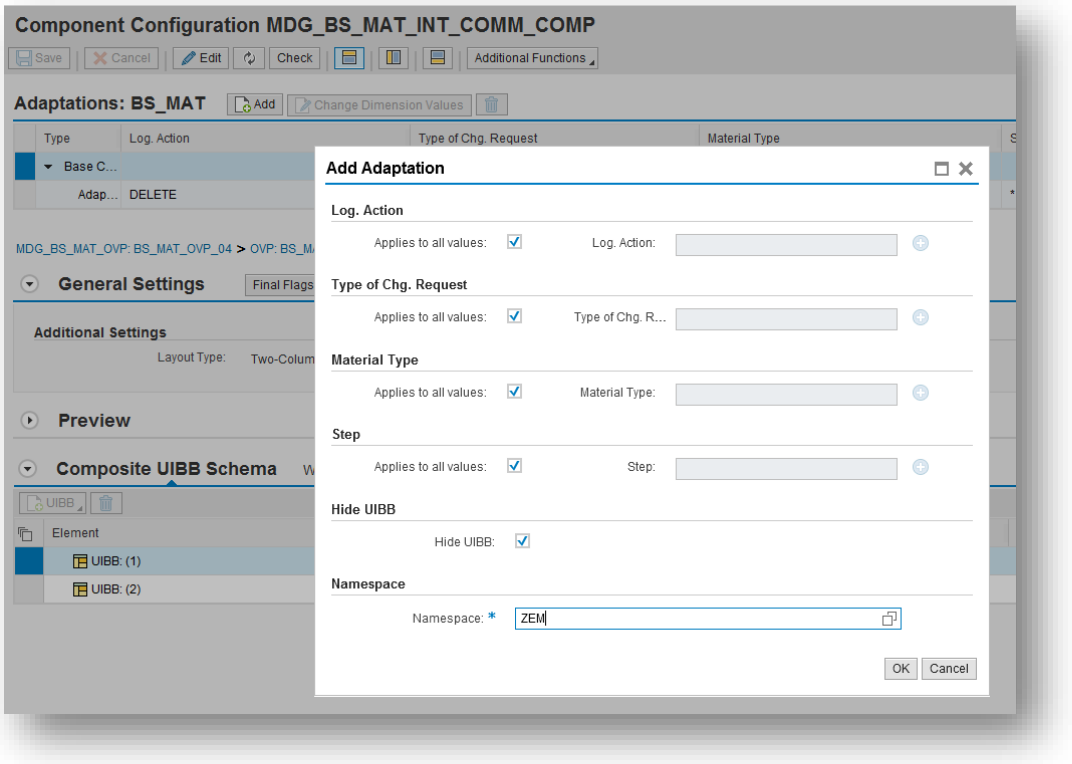


Change **Tool Tip**. Mark the Element BEGRU and click on Attributes. Change the tool tip.



Save your UI CBA.

If you want to hide a complete UIBB, then you have to create a CBA on the level of the relevant UIBB. Flag ‘Hide UIBB’.

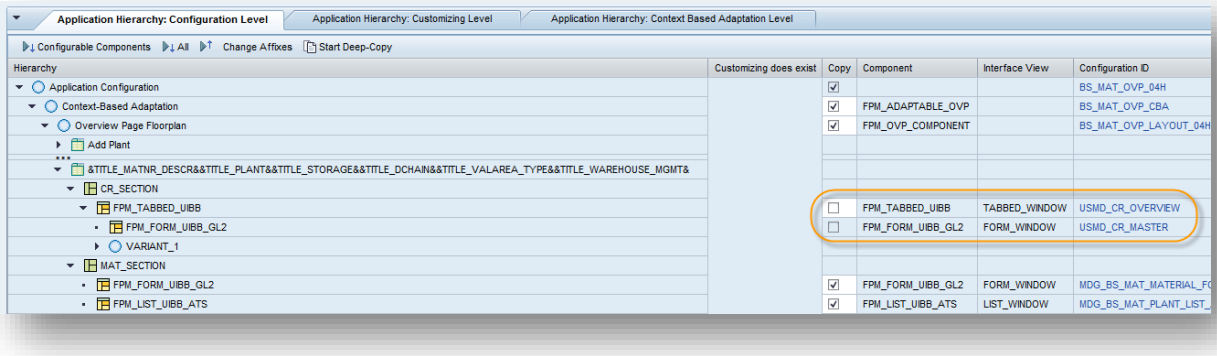


3.2 Custom UI Configuration

If customizing is not sufficient, use the application configuration BS_MAT_OVP of the Web Dynpro Application MDG_BS_MAT_OVP as a template for the creation of your UI configuration:

1. Select this in SE80 in package MDG_BS_MAT_UI in folder Web Dynpro/Web Dynpro Application/MDG_BS_MAT_OVP/Applic. Configurations.
2. Start the configurator for BS_MAT_OVP (EhP6) or BS_MAT_OVP_03 (MDG6.1) and click on Continue in Display Mode.
3. Click on the configuration BS_MAT_OVP_LAYOUT/ BS_MAT_OVP_LAYOUT_03.
4. There, go to Additional Functions/Deep Copy. Click on Expand Configurable Components.
    a. You should name all your target UI configurations with prefix Z##_. Use Button ‘Change Affixes’
    b. Uncheck all elements you want to reuse (not copy).
    Usually you can uncheck all the components belonging to the popups like ‘Add Plant’, ‘Copy Plant’ etc.

5. **Uncheck USMD_CR_OVERVIEW.** The Change Request Header UIBB must not be copied due to technical reasons.



6. Click on Start Deep-Copy.
7. Navigate to your application configuration Z##_ BS_MAT_OVP*.
8. Then click on your OVP configuration Z##_ BS_MAT_OVP_LAYOUT*.

Now you can change your UI. Use breadcrumbs to go back to your OVP_LAYOUT.

3.2.1 Customizing for Custom UI Configuration

Assignment of UI to the Complete Process
If you want to use the new custom UI configuration for the complete process, then go to the customizing (MDGIMG → Master Data Governance → General Settings → Process Modeling → Business Activities → Link Actions with UI Application and Business Activity: Custom Definition).
Select your entry: BO type 194, Action CREATE, Current Application MDG_BS_MAT, Current UI Configuration BS_MAT_INIT. Change the target UI configuration from BS_MAT_OVP to your newly created UI configuration.

For the authorization, go to the customizing (MDGIMG → Master Data Governance → General Settings → Process Modeling → Business Activities → Link Actions with Business Activity: Custom Definition).
Create a new entry for UI application name MDG_BS_MAT_OVP and your new UI configuration with action CREATE and business activity MAT1.

If you want to use the new custom UI for Material Change, you have to maintain these two customizing tables for the Actions CHANGE (MAT2) and DISPLAY (MAT3). Please make sure that you maintain both because the UI always goes first to the display mode and then by user action into change mode.

Assignment of UI to Different Steps
If you want to use the new custom UI configuration for one step you can assign your configuration ID to a step of your change request type in the customizing. (MDGIMG → Master Data Governance → General Settings → Process Modeling → Change Requests → Configure Properties of Change Request Step). Don't forget also to maintain the authorization link. (MDGIMG → Master Data Governance → General Settings → Process Modeling → Business Activities → Link Actions with Business Activity: Custom Definition).

This allows you to deviate from settings defined in the Customizing activities explained in above chapter (Link Actions to a UI Application and Business Activity: Standard Definition/Link Actions to a UI Application and Business Activity: Customer Definition).

Note:
Instead of creating many custom UI configurations and assigning it with this customizing table to the steps, you can use one custom UI configuration and use CBA for the steps.
If you have only small differences per step and one global UI for all processes use CBA. If you have different UI's for the different processes use Custom UI Configurations.

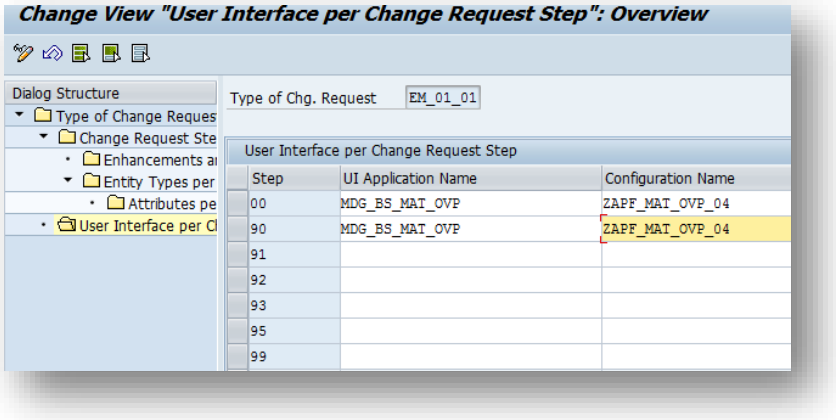
Step-dependent UI are used only if you are the actual processor of the Change Request. CBA is always used.

Navigation Logic to Target UI

Web Dynpro Application	*Menu Entry*
USMD_EDITION_CREQUEST	My Change Request
USMD_EDITION_CREQUEST	Display Change Request
USMD_WF_NAVIGATION	Worklist/Inbox
MDG_BS_MAT_SEARCH	Search Material

Navigation logic to Target UI from Web Dynpro Applications:

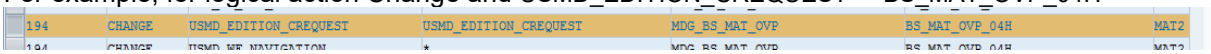
1. Step-dependent UI configuration maintained:



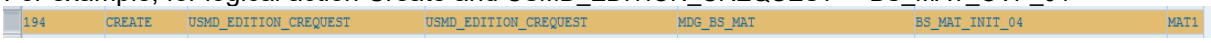
   The step-dependent UI Configuration is used if you are the processor, or logical action MULTI is used. Also, it is used from the initial screen for material if you specify already the Change Request type.
   If you display a Change Request from Display Change Request/My Change Request with more than one open parallel work item and you are the processor of more than one work item, the system is not able to determine one UI configuration. Therefore, the default UI configuration is chosen as explained in the next step.

2. Otherwise (CR is finished/no step-dependent UI configuration maintained/not processor) the default UI Configuration is used from the navigation customizing for:

   For example, for logical action Change and USMD_EDITION_CREQUEST -> BS_MAT_OVP_04H



   For example, for logical action Create and USMD_EDITION_CREQUEST -> BS_MAT_OVP_04



3. CBA for the target UI configuration maintained: UI is adapted accordingly.

Note: Switching from display mode to change mode is not navigation. If you display a material (and no change request type was previous entered) there is no change of UI possible during switch to EDIT mode. If you want to adapt the UI then you have to use CBA.

3.3 Enhance Initial Screen

If you want to enhance the initial screen with more fields then

- Append fields to DDIC structure MDG_BS_MAT_S_MP_SETTINGS_DATA
- Enhance the UI configuration BS_MAT_INIT_xx or create a new initial screen.
  The feeder CL_MDG_BS_MAT_FEEDER_FORM_SET of the initial screen transfers automatically (method GET_CUSTOM_FIELDS) all fields of the structure MDG_BS_MAT_S_MP_SETTINGS_DATA to the following overview page (OVP).
  If you create a new initial screen, don't forget the customizing 5.2.1 Customizing for Custom UI Configuration.
- Enhance feeder for the corresponding receiving UIBB to get the values from table CL_MDG_BS_MAT_ASSIST_UI=>GT_CUSTOM_URL_PARAM.

3.3.1 Make Change Request Type Mandatory

If you want to create a material or mark it for deletion the change request type is mandatory. If you want to change a material, the change request type is not mandatory, to enable the user to display the material independent from any governance perspective.

If you want to force the user to also specify a change request type in this case, you can set this field to mandatory.

Two steps are necessary:

1. Set the field to mandatory:
   a. Enhance the method /PLMB/IF_SPI_PROPERTIES_ACCESS~GET_PROPERTIES of class CL_MDG_BS_MAT_SP_SETTINGS accordingly **or**
   b. Implement BAdI /PLMB/EX_SPI_PROPERTIES_ACCESS SET (recommended) of enhancement spot /PLMB/ES_SPI

   This marks the field with the asterisk '*' at the screen only. It does not imply an automatic check, that the value is specified.
2. Implement the value check 'is value specified':
   a. Enhance the method ON_CONTINUE_EVENT of feeder class CL_MDG_BS_MAT_FEEDER_FORM_SET **or**
   b. Set the indicator 'Check Mandatory' in the general settings of the UIBB containing the field.

   The version 2a requires more effort but provides more flexibility for the user. Dependent on the specified values the code can, for example, generate default values. Version 2b doesn't leave the UI processing if the change request type is not specified. This cannot be influenced by any implementation.
   Remark: The indicator 'Check Mandatory' is not supported in the OVP, only in the initial screen.

3.4 New Form UIBBs

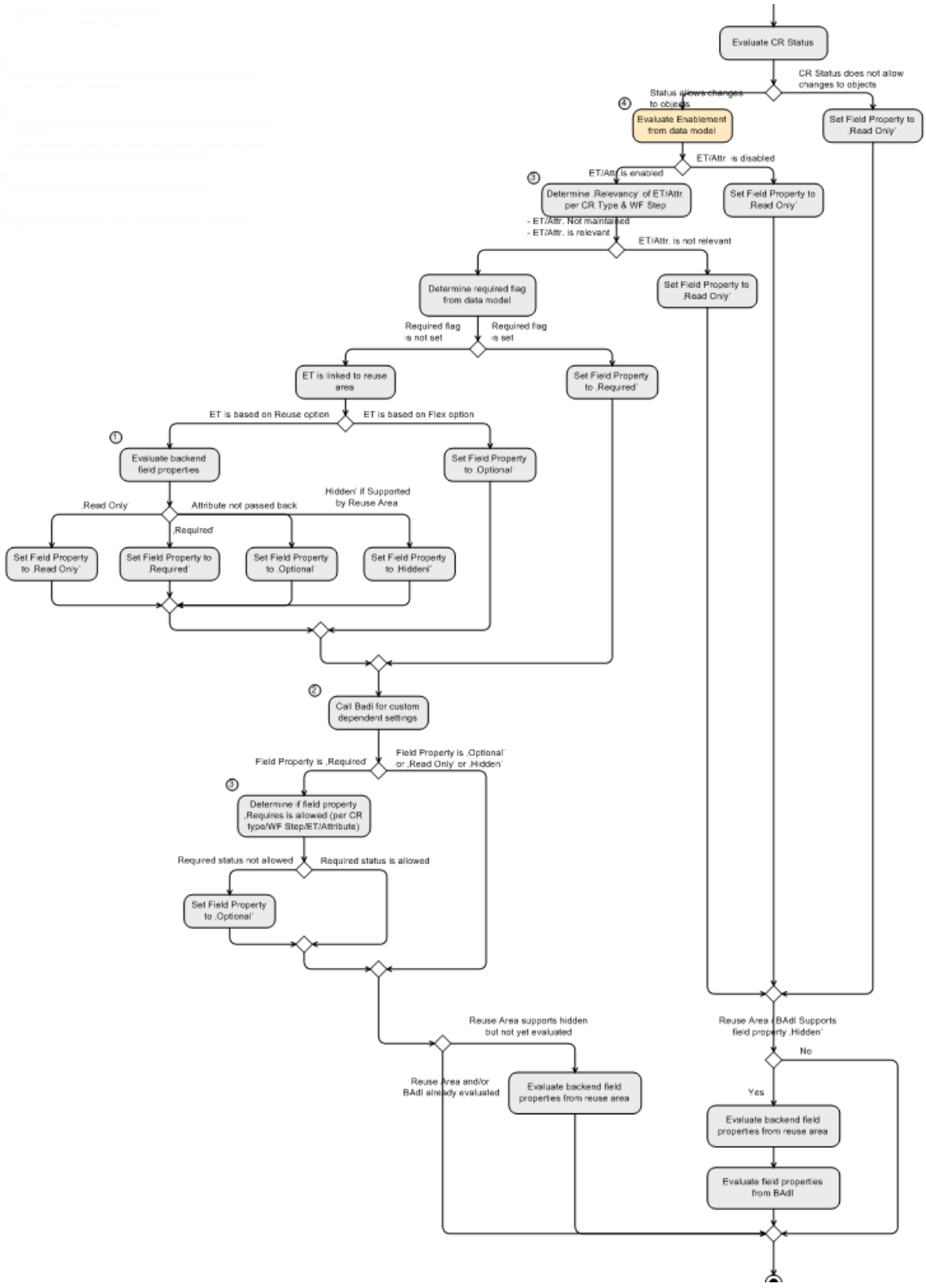
To create a new form UIBB and related feeder class (implementing IF IF_FPM_GUIBB_FORM) see FPM Floorplan Manager for Web Dynpro ABAP.

Additional information for text fields:

- If you run into the problem that fields have no related _TXT-descriptions fields, reorganize the buffer with report MDG_BS_MAT_TEXT_DELETE (see SAP Note 1669963).
- Texts can be translated in SE63:
  - Start the transaction SE63
  - From the menu select Translation -> ABAP Objects -> Transport Object
  - Enter the transport object of the UIBB to be translated
  - Select the correct translation direction and continue
- If translated texts in copied configurations are missing, see SAP Note 1710822

3.5 Field Properties

The following diagram shows how field properties are determined in MDG. Field Properties are hidden, display only, input enabled or mandatory (*).



Abbreviations:

- ET/Attr.: Entity Type/Attribute
- ET: Entity Type

① Evaluation will be done using a new method in the access interface, implemented by the applications.

② Enhancement Spot USMD_ACC_FLD_PROP_CUST_DEP_SET

③ Data are maintained with the same IMG activity

④ Entity Types/Attributes are always relevant until EhP6. With MDG6.1 the governance scope is evaluated.

Details for the diagram

1. Evaluation of the backend field properties will be done by Access Interface.
   For reuse entities in MDG-M, the properties as defined by:
   1.1. Customizing 'Maintain Field Selection for Data Screens' (Transaction OMSR): Here the settings for Industry sector, Material type, transaction MM01, and KB. Also considered are SAP1 and SAP2 but must not be changed by customer. During the check, also the settings for transaction MM02 are considered, as MDG can't distinguish between the backend material create and change. Therefore, SAP recommends keeping the field control for MM01/MM02 in sync to avoid nontransparent messages.
   1.2. Customizing 'Define Attributes of Material Types' (T134, selected user departments)
   1.3. BAdI BADI_MAT_F_SPEC_SEL
   1.4. Hard coded mandatory field checks in the business logic of the material master, for example
   - Weight unit mandatory if material is sales relevant
   - MRP type mandatory for plant specific MRP data
   - Material group mandatory if material is purchase relevant
2. The BAdI USMD_ACC_FLD_PROP_CUST_DEP_SET can/should be used for migrating code of the UI BAdI of EhP5
3. Both settings are maintained with the same IMG activity "Configure Properties of Change Request Step"

3.5.1 Customizing: Configure Properties of CR Step

In this Customizing activity (Master Data Governance→ General Settings→ Process Modeling→ Change Requests→ Configure Properties of Change Request Step), you determine settings for a change request step of a change request type.

View: Enhancements and Checks for each Change Request Step

In this view, you can complete the following actions for a change request step:

- Specify which checks are relevant
- Control the display of messages by specifying a message output. For example, you can ensure some messages display only as warnings
- Determine if the duplicate check is always executed or only executed when data changes

In this view, you can decide if the 'Reuse Area Check' will be executed. The 'Reuse Area Check' gathers information used to determine the field properties for all fields (regardless of whether they appear on the UI or not). The 'Reuse Area Check' gathers the information from the settings in transaction OMSR as well as from the BAdI BADI_MAT_F_SPEC_SEL and the hard-coded mandatory field checks in the business logic. Therefore, if the 'Reuse Area Check' is marked as relevant for a step, the field properties cannot be changed for entities or attributes in the view Entity Types/Attributes per Change Request Step.

Views: Entity Types per Change Request Step and Attributes per Change Request Step

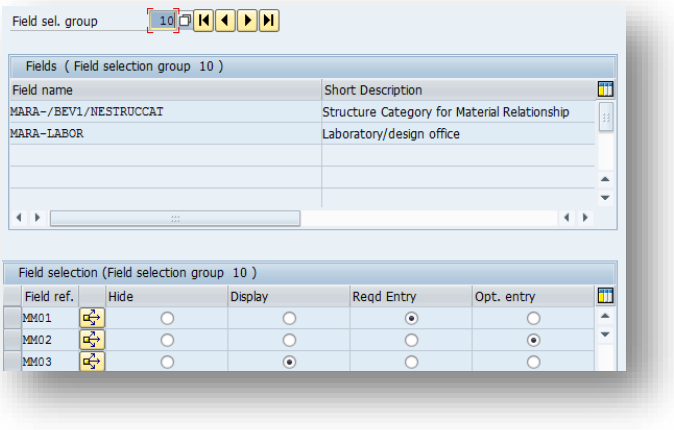
You can complete the following actions for a change request step:

- Set field properties to specify which entity/fields are not relevant (this set the entity or field to read only), and which entity/fields are not mandatory. For example, you can make a mandatory field optional (but only if you deselect the 'Reuse Area Check' before).
- Reduce the number of checks applied to fields by specifying a Check Logic for an entity type

Scenario:

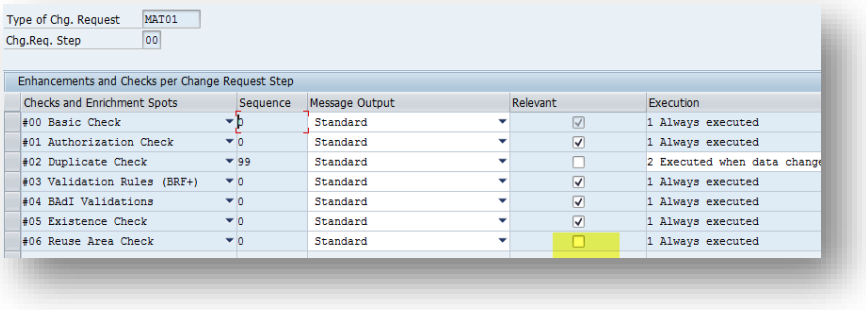
You want to limit the number of mandatory fields in the first steps of your change request process.

Example: LABOR is set to mandatory in T130F (OMSR), but you don’t want the field to be mandatory at step 00.

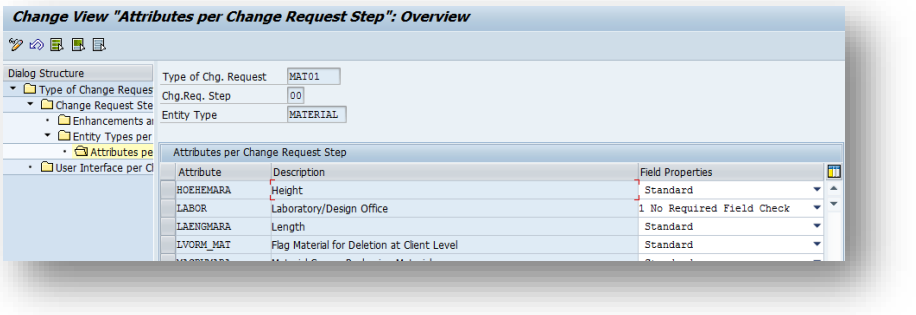


Solution:

If you want to set entities or selected attributes to ‘No required Field Check’ for a step, you have to deselect the 'Reuse Area Check'.



Then the 'Reuse Area Check' is not used, but the field properties from OMSR/OMS9 are still considered and the fields are displayed accordingly on the UI (that means: hidden, read only, optional, and mandatory). Next you can set the system to ignore the field properties by setting complete entities or selected attributes to ‘No required Field Check’.



3.5.2 Enhancement Spot USMD_ACC_FLD_PROP_CUST_DEP_SET

If you would like to change the field properties, in addition to the setting of 'Field Properties', you can use enhancement spot USMD_ACC_FLD_PROP_CUST_DEP_SET.

There you can set the field properties to hidden, read only, optional, or mandatory:

- already hidden fields can't be changed
- for read only fields you can change the field properties only to hidden
- for optional fields, you can set the field properties to hidden, read only, or mandatory
- for mandatory fields, you can set the field properties to hidden, read only, or optional.

You can also make the setting dependent on other values.

More information is available for this BAdI in transaction MDGIMG, under *Master Data Governance → General Settings → Process Modeling → Change Requests → Business Add-Ins*. There you can find the documentation "BAdI: Access to Customer-Dependent Field Property Settings". There is also a small example implementation for this BAdI (Class CL_USMD_ACC_FLD_PROP_EXAMPLE).

Scenario:
When a material is created, both the moving average price and the standard price are changeable, regardless of the settings of the price control indicator.

If the valuation view of an existing material will be changed, only one of the price fields shall be changeable, dependent on the price control indicator. If the price control is S (standard price), the standard price shall be read-only and only the moving average price shall be changeable. If the price control is V (moving average price), the moving average price shall be read-only and only the standard price shall be changeable.

Solution:
In order to distinguish between creation of a new material and changing the valuation view of an existing material, the BAdI USMD_ACC_FLD_PROP_CUST_DEP_SET must be implemented and the field properties must be adapted accordingly in method IF_EX_USMD_ACC_FLD_PROP_CDS~MODIFY_FLD_PROP_ATTR. The example coding in SAP Note 1918422 can be used as a template.

Note: The BAdI is only used for the field properties and is not considered during checks. If you also need messages during the check for new mandatory fields, you have to build the check with a new BRF+ rule or using the BAdI USMD_RULE_SERVICE (see How To Guide http://scn.sap.com/docs/DOC-14915).

3.5.3 Hide Fields

Beside Context-Based Adaptation (CBA) you can use one of the two possibilities described in the following to hide fields.

- Reuse the settings of the backend transaction MM02 for MM model fields
- Use BAdI USMD_ACC_FLD_PROP_CUST_DEP_SET to strengthen the already determined field properties for optional or read only MM model fields

- Use BAdI /PLMB/EX_SPI_PROPERTIES_ACCESS SET to strengthen the already determined field properties for optional or read only MM model fields or change the field properties for SPI fields.

Also, if fields are set to hidden, they can be changed, for example via data import or thru code (feeder, derive).

Setup of T130F

If a field is set to 'Hide' in the setup of the field groups for transaction MM02 (maintenance view V_130F), this setup is not considered in the single object maintenance for material (Web Dynpro Application MDG_BS_MAT_OVP).

To consider this setup in the same way as it is done in transaction MM02, following steps are necessary.

- If you didn't install the newest support pack (corresponding to the notes listed in the following), you need to implement the notes 1921684, 1925163, 1928042, and 1928760.
- In releases before MDG 7.0 SP02 is out of performance reasons the above-mentioned setup **not considered** by default.
  To enable this feature for example in MDG6.1 or MDG7.0, you need an enhancement of the method SET_HIDDEN_IND_IF_POSSIBLE in class CL_MDG_BS_MAT_ASSIST with the following code:

```
Method: CL_MDG_BS_MAT_ASSIST->SET_HIDDEN_IND_IF_POSSIBLE
…
  CV_SET_HIDD_IND_IF_POSSIBL = ABAP_TRUE.
…
```

BAdI USMD_ACC_FLD_PROP_CUST_DEP_SET

Starting with MDG 7.0 you can use the Enhancement Spot USMD_ACC_FLD_PROP_CUST_DEP_SET with interface IF_EX_USMD_ACC_FLD_PROP_CDS (MODIFY methods) to hide the field.

You can also use these methods to overwrite these settings from T130F (described before). Use the method IS_FIELD_PROP_HIDDEN_SUPPORTED to enable the consideration of 'hide'.

Enhancement Spot /PLMB/ES_SPI (SPI BAdI)

To hide fields (besides setting other UI field properties) you can use also the BAdI /PLMB/EX_SPI_PROPERTIES_ACCESS with interface/method /PLMB/IF_EX_SPI_PRPTY_ACCESS~>GET_PROPERTIES. The BAdI is part of the enhancement spot /PLMB/ES_SPI.

If the BAdI /PLMB/EX_SPI_METADATA (Interface /PLMB/IF_EX_SPI_METADATA, method ENRICH_NODE_DEFINITION) was used to add fields (transient fields) beside the governance model (here 'MM') to the UI, the method GET_PROPERTIES can be used to influence the UI field properties.

The implementation of the method GET_PROPERTIES can change the UI field properties with one of the parameters CT_PROPERTIES_SINGLE_IDX, CT_PROPERTIES_MULTI_IDX, or CT_PROPERTIES – they are filled depending on the type of used UIBB.

The attribute OPTION in these parameters defines the property of the field to display. With the value '5' (please see definition on the domain of the used data element) you can hide fields.

Starting with NW 7.31 SP4, FPM can close the gaps, i.e., in case a field is hidden, and the visible fields in the same UIBB in the same column are moved upward.
The fields in the left and right column are not synchronized in this step. If because of this upward movement an initial line would be displayed in the bottom of the UIBB, the height of the UIBB is reduced accordingly.

Please be aware, that changes of the UI field properties in this BAdI are not reflected in the MDG framework, i.e.

- If you set a field to mandatory, the Governance Framework ‘didn’t know this’, i.e., there is no automatic check, if the value was entered or not.
  You have to check this in an enhanced feeder or in the Check methods with BRF+ or a BAdI
- If you set a mandatory field to invisible, it is still requested by the Governance Framework, i.e., if hiding of the field is correct, you have to ensure, that there is a default value, satisfying the framework (e.g., via derivation or thru the feeder).

3.6 Leading Zero’s for NUMC

If the domain of an MDG-M attribute has the type NUMC, the value of this attribute will usually be displayed with leading zeroes.

If these leading zeroes are not required, you have two options:

1. Feeder class redefinition
   The Customer can inherit from the MDG-M feeders (for example CL_MDG_BS_MAT_FEEDER_FORM) and add code for fields as desired in method
   /PLMU/IF_FRW_G_FIELD_DEF~CHANGE_FIELD_DEFINITION
   - Loop over parameter CT_DEFINITION
   - Check NAME for the attribute name where the leading zeroes should be hidden
   - Set NULL_AS_BLANK = abap_true in the line CT_DEFINITION
2. Conversion exit
   In case a customer (Z) domain is used, it’s possible to assign a conversion exit to it. To do this:
   - Copy the conversion exit AXTAL (function group and two function modules CONVERSION_EXIT_AXTAL_INPUT/OUTPUT)
   - Assign the conversion exit to the domain

Note: Do not use the conversion exit ALPHA, as its input conversion exit does not convert blanks into zeroes.

3.7 Additional Buttons

How the additional buttons are added depends on the intended functions. Buttons to switch between different workflow branches and to influence the status of the change request can be added through customizing. Buttons to influence the presentation of the data on the screen or to add additional functionality have to be added through configuration and code.

3.7.1 Buttons to Influence the Workflow

Buttons covering actions to influence the workflow, for example to switch between different branches, can be defined in the customizing using transaction MDGIMG *Master Data Governance → General Settings → Process Modeling → Workflow → Define Change Request Actions* and *Define Change Request Step Types and Assign Actions*

The buttons corresponding to the actions are generated automatically depending on the current workflow step of the change request in the global toolbar of the UI. The generated buttons are independent from the UI configuration used.

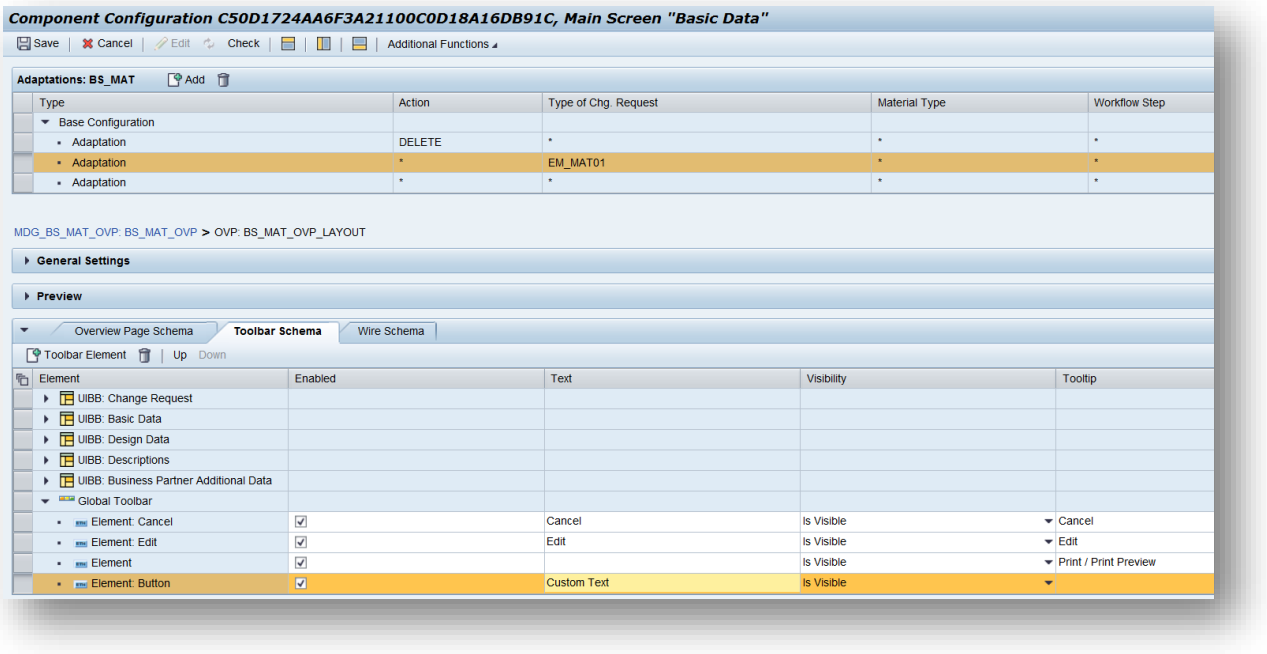
See also the documentation of the configuration steps in the implementation guide.

3.7.2 Other Additional Buttons

All other additional buttons have to be added to the UI configurations they are needed in. In addition, the actions assigned to them have to be implemented.

To enhance the configuration start transaction SE80, select your configuration, and start the Configurator. Go to the Layout.

Scroll down and open the tab strip Toolbar Schema. There you will find the global and the local toolbar. Buttons of the global toolbar are displayed in the header area; local buttons are assigned to the corresponding assignment block.



Add a button where needed. Display the attributes and maintain the values for attributes FPM Event ID, Text, Icon, and separation bar.

Important:

- The automatically generated Element ID should be adapted to the customer namespace to avoid conflicts in later releases
- Do not use an existing FPM event from the list but create a new one instead. See the next chapter regarding the implementation of the logic of the actions

Implementation of the Logic

Depending on the task of the action the additional button is to execute, it can be implemented in the feeder or as part of the SPI:

- **Feeder Enhancement**

  UI related tasks, for example open a popup with additional information, are implemented in the feeder.

  Depending on the intended enhancement you can inherit from a specific feeder (For example, CL_MDG_BS_MAT_FEEDER_FORM_MARC), replace it in the UI configuration and implement the action there. Alternatively, you can implement a post exit in a more generic feeder such as CL_MDG_BS_MAT_FEEDER_FORM or CL_MDG_BS_MAT_FEEDER_LIST.

  The first solution restricts the use to the UIBBs with the new feeder; the second allows the use of the action in several UIBBs.

  The source code for handling your FPM events has to be defined in method /PLMU/IF_FRW_G_ACTIONS~PROCESS_ACTION_EVENT or /PLMU/IF_FRW_G_GLOBAL_EVENTS~PROCESS_GLOBAL_EVENT.

  You can add the action definition with its Event ID in method /PLMU/IF_FRW_G_ACTIONS~GET_ACTION_DEFINITION of the feeder. Defined this way, your FPM events will be attached to the list of already existing FPM events for the configuration described before.
- **SPI Enhancement**

Business Object or governance related tasks shall be implemented in the Service Provider Interface (SPI).

They are embedded in the UI in the same way as the UI related tasks described before (definition in GET_ACTION_DEFINITION and implementation in for example, PROCESS_ACTION_EVENT).

The implementation has to hand over the processing to the SPI, like here for cancel:

```
  CASE io_event->mv_event_id.
    …
    WHEN cl_fpm_event=>gc_event_cancel.
      TRY.
          mo_application_model->action(
           EXPORTING
             iv_node_name   = cl_mdg_bs_mat_c=>gc_node_name_mat
             iv_action_name = cl_mdg_bs_mat_c=>gc_action_rollback
             is_param       = abap_true ).
        CATCH /plmb/cx_spi_error INTO lo_exc.
          BREAK-POINT ID mdg_bs_mat_ui.
      ENDTRY.
…
```

IV_NODE_NAME represents the node the SPI action is called for and IV_ACTION_NAME identifies the task to do. The type of IS_PARAM and EG_PARAM is defined at the action definition in the metadata provider.

Implement the corresponding BAdI interfaces of the enhancement spot /PLMB/ES_SPI to add the actions of the nodes. The method ENRICH_NODE_DEFINITION of the interface /PLMB/IF_EX_SPI_METADATA is needed to define the actions and e.g., the methods BEFORE_ACTION (to skip the standard) and AFTER_ACTION (doing) or the interface /PLMB/IF_EX_SPI_APPL_ACCESS contain the implementation of the action itself.

The class CL_MDG_BS_MAT_BO contains the access to the currently processed material and its change request.

- **Predefined Actions (INSERT, DELETE)**

  To configure Insert and Delete buttons in lists you can use the FPM Event IDs FRW_INSERT and FRW_DELETE. Their actions are predefined.

  Lists using the feeder CL_MDG_BS_MAT_FEEDER_LIST or derivations of it can use it. Both buttons can be configured in the toolbar schema of the UIBB itself as described before. Depending on the type of data of entity they are sometimes already configured in standard.

  If the actions are currently allowed is determined in method /PLMB/IF_SPI_PROPERTIES_ACCESS~GET_OPERATION_PROPERTIES of class CL_MDG_BS_MAT_SP_GENERIC. Reasons, that entities can't be created or delete depend on the used change request type, the data of the entity type, the authorization of the current user etc.

  The result of GET_OPERATION_PROPERTIES can partly be overwritten in method ENRICH_NODE_DEFINITION of the BAdI Interface /PLMB/IF_EX_SPI_METADATA (Enhancement spot /PLMB/ES_SPI) with attribute SUPPORTED_OPERATION_GROUP. This can mainly be used to disable the buttons. Enabling is only supported if the MDG governance framework accepts the corresponding data.

3.8 Hide Buttons

3.8.1 Change Request Buttons

If you want to hide buttons dependent on specific data you can use the following methods:

- To hide buttons (For example: Check, Submit, Buttons for Actions, Save), implement an enhancement of type PostExit for the methode CL_USMD_CR_MASTER→GET_ACTIONS or CL_USMD_CR_MASTER->CHECK_ACTION_USAGE_SINGLE.

- If you want to hide the Edit Button (global and local) you must implement an enhancement of type PostExit for the method CL_USMD_CR_MASTER->CHG_FLD_PROP_EDIT_BUTTON_CR.

The data of CR can be found at the attribute MS_CREQUEST_INFO (in class CL_USMD_CR_MASTER) or already buffered in attributes of CL_MDG_BS_MAT_ASSIST_UI in the context of the material.

Post exits are modification-free enhancements. But you have to verify all these exits after upgrading. (https://help.sap.com/docs/ABAP_PLATFORM_NEW/fc79a39b30fe4d9aa983bad6787ab9ad/49cc8b49506573ffe10000000a421937.html)

3.8.2 Material UIBB Buttons

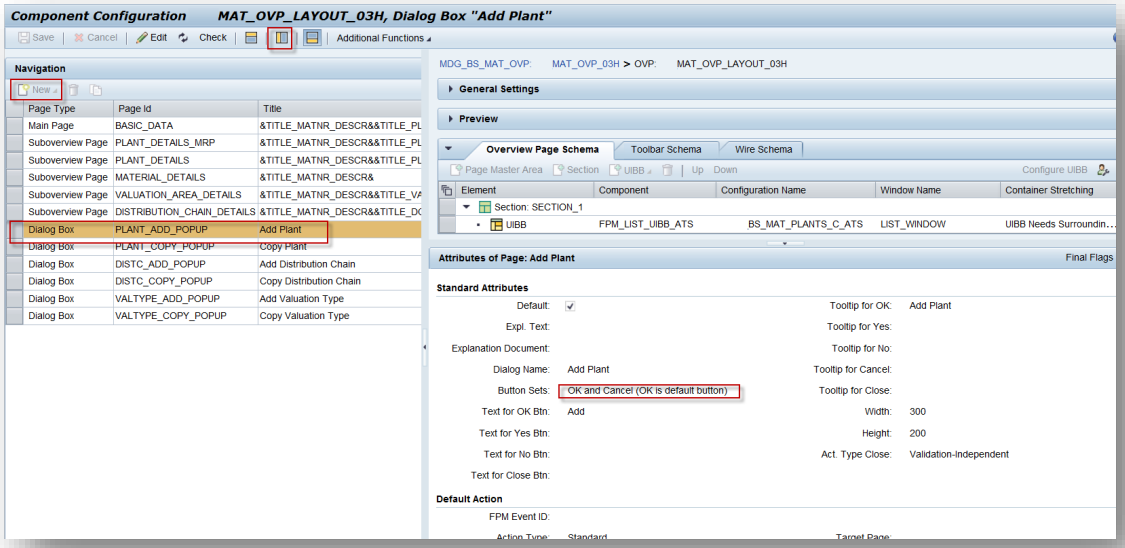
If you want to hide buttons you can remove them from the web dynpro configuration.

If you want to hide buttons dependent on specific data you can use the following methods for the two types of buttons:

- Common Buttons (For example: create, change, delete) can be dynamically hidden by enhancing CL_MDG_BS_MAT_SP_GENERIC-> /PLMB/IF_SPI_PROPERTIES_ACCESS~GET_OPERATION_PROPERTIES. Data of material can be found at the attribute MO_BO.
- Material specific buttons (For example: add and copy) are defined in CL_MDG_BS_MAT_FEEDER_LIST_BASE. They can be dynamically hidden by enhancing /PLMU/IF_FRW_G_AFTER_GET_DATA~AFTER_GET_DATA or CHANGE_LOCAL_ACTION_USAGE.

3.9 Page of Type DIALOG BOX

Click on Navigation and add a new page of type “Dialog Box”, enter a page ID and a title. For the new page, select the button set.



Assign UIBBs to the dialog screen page and assign your custom feeder class (see chapter 5.10 New Feeder). Feeder class parameters: you have to enter the YYCUSTOM node as Node Name and use MDG_MAT as Application Building Block ID.

If necessary, include the UIBBs displayed in the dialog box in the wiring schema.

3.10 New Feeder

Examples:

- Check for material name
- Derive material number
- Restrict values displayed in dropdown list
- Restrict values displayed in input help

Different approaches are possible:

- Define an enhancement on existing classes (button  in transaction SE24). These enhancements can be of type Pre-/Post-/Overwrite Exit
    - In case of complex enhancements (with a combination of Pre-/Post-/Overwrite Exits), this can be confusing, as it is difficult to keep the overview over standard code and exits. Post exits are modification-free enhancements. But you have to verify all these exits after upgrading. (http://help.sap.com/saphelp_smehp1/helpdata/en/c5/f4b9422e0fb911e10000000a1550b0/content.htm)

- Redefine the whole feeder class and use it in your UIBB component configuration
    - We recommend not use inheritance on a class that is used as a superclass anywhere, as this would have impact on the whole inheritance hierarchy. Use the enhancement technology for these scenarios.

For example:

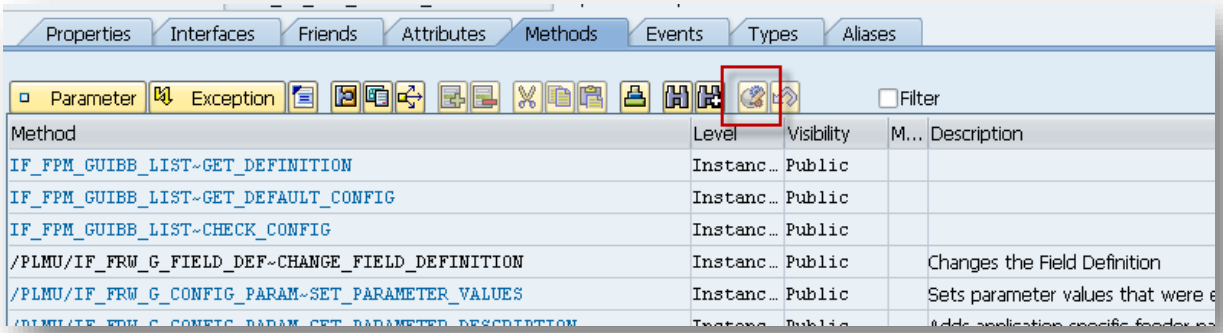
- Enhancement of class CL_MDG_BS_MAT_FEEDER_FORM_MAT (or CL_MDG_BS_MAT_FEEDER_FORM_MARA, respectively, depending on implementing notes – see 4.2.1) for simple issues
- Inheriting from CL_MDG_BS_MAT_FEEDER_FORM_MAT (or CL_MDG_BS_MAT_FEEDER_FORM_MARA, respectively) and redefinition of necessary methods for complex issues (like material number derivation)
- Enhancement of superclass CL_MDG_BS_MAT_FEEDER_FORM (not redefinition, as this class is used as superclass for CL_MDG_BS_MAT_FEEDER_FORM_MAT or CL_MDG_BS_MAT_FEEDER_FORM_MARA, respectively)

3.10.1 Example: Restrict Values Displayed in Dropdown List

Add interface /PLMU/IF_FRW_G_FIELD_DEF to your list feeder class and implement method CHANGE_FIELD_DEFINITION.

This example works only for dropdowns. The UIBB is connected to an MDG feeder class (in this case CL_MDG_BS_MAT_FEEDER_LIST) because it is a list UIBB. Create a custom feeder class with the superclass CL_MDG_BS_MAT_FEEDER_LIST/FORM.

Then re-implement the method CHANGE_FIELD_DEFINITION.



Provide two important things in the method:

- Call the method from the superclass
- Custom coding

Method: /PLMU/IF_FRW_G_FIELD_DEF~CHANGE_FIELD_DEFINITION

```
METHOD /PLMU/IF_FRW_G_FIELD_DEF~CHANGE_FIELD_DEFINITION.

  CALL METHOD SUPER->/PLMU/IF_FRW_G_FIELD_DEF~CHANGE_FIELD_DEFINITION
    CHANGING
      CO_CATALOGUE  = co_catalogue
      CT_DEFINITION = ct_definition.

  DATA:
    ls_nvp          TYPE WDR_CONTEXT_ATTR_VALUE,
    lv_name         TYPE name_komp,
    lv_is_segm_ent  TYPE boole_d.

  FIELD-SYMBOLS:
    <ls_definition> TYPE /plmu/s_frw_g_field_descr_appl.

  LOOP AT ct_definition ASSIGNING <ls_definition>.

    lv_name = <ls_definition>-name.
    IF lv_name EQ 'WERKS'.
* See example coding in chapter
* 5.10.3 Example Coding to build up Value List for Dropdown List and Value Help
      lv_is_segm_ent = 'X'.
      ls_nvp-value   = 'custom'.
      ls_nvp-text    = 'custom'.
      APPEND ls_nvp TO <ls_definition>-fixed_values.
    ENDIF.


  ENDLOOP.

ENDMETHOD.
```

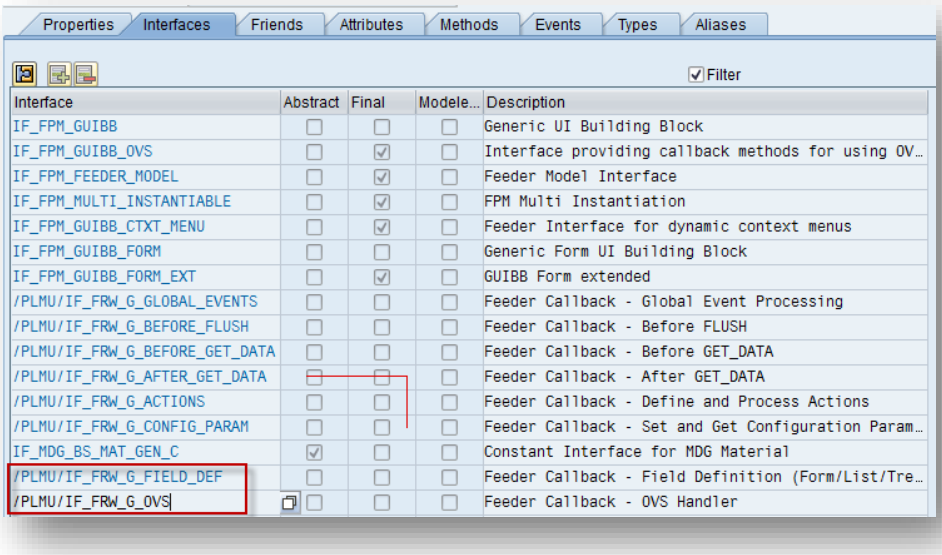
In the UIBB, replace the original feeder with your own class and keep all parameters for the feeder as before.

3.10.2 Example: Restrict Values Displayed in the Input Help

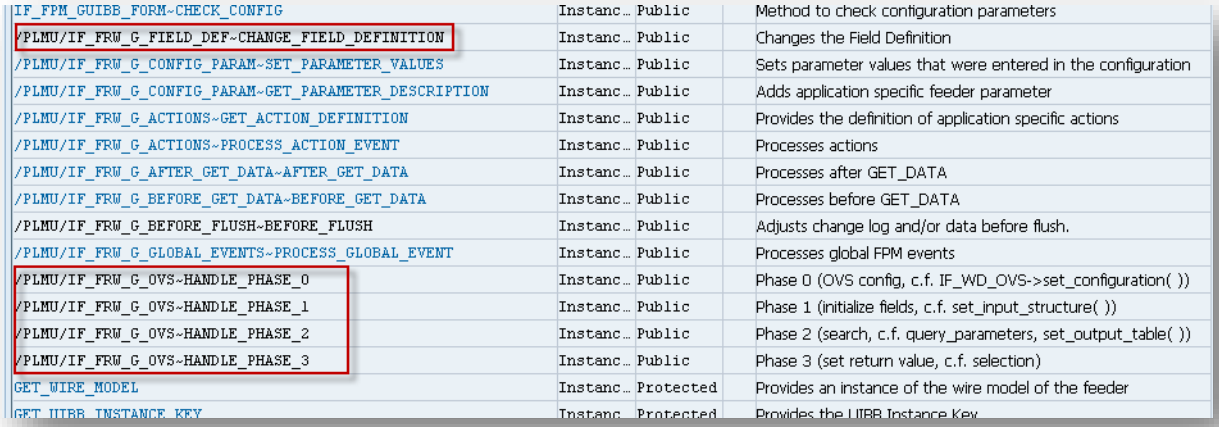
The new feeder class should implement the interface /PLMU/IF_FRW_G_OVS of the PLM UI framework to provide alternative (or restricted) value lists.

If the customer wants to provide a completely new value help for users, you can use the OVS mechanism.

We use the following interfaces for OVS: /PLMU/IF_FRW_G_OVS. Example of custom Feeder Class:



Example of /PLMU/IF_FRW_G_FIELD_DEF



Now you have the four methods for preparing but before you start you have to tell the field attributes that you want to do some OVS on the field.

Step one is to define that there is an OVS help for a certain field. This is done in method CHANGE_FIELD_DEFINITION.

The parameter ct_definition contains an attribute OVS_NAME which is set to the implementing class, in this case the feeder itself because it also contains the methods HANDLE_PHASE_0…3

Method: /PLMU/IF_FRW_G_FIELD_DEF~CHANGE_FIELD_DEFINITION

```
*Display the value in Search Help Screen.
  CALL METHOD super->/plmu/if_frw_g_field_def~change_field_definition
    CHANGING
      co_catalogue  = co_catalogue
      ct_definition = ct_definition.

  LOOP AT ct_definition ASSIGNING <ls_definition>.
```

```
    lv_name = <ls_definition>-name.
    CASE lv_name .
      WHEN 'MSTAE'.
         <ls_definition>-OVS_NAME = 'custom FeederClass'.

    ENDCASE.
  ENDLOOP.
ENDMETHOD.
```

Start at phase 0 (optional) for the selection screen

Method: /PLMU/IF_FRW_G_OVS~HANDLE_PHASE_0

```
method /PLMU/IF_FRW_G_OVS~HANDLE_PHASE_0.
 * initialize search help with single selection
  io_ovs_callback->set_configuration( window_title       = 'OVS Selection
Dialog'
                                      table_multi_select = abap_false ).

endmethod.
```

Phase 1 (optional) for the selection screen

Method: /PLMU/IF_FRW_G_OVS~HANDLE_PHASE_1

```
method /PLMU/IF_FRW_G_OVS~HANDLE_PHASE_1.


  DATA lr_data TYPE REF TO data.
  DATA ls_data TYPE ty_s_sel_screen.


  FIELD-SYMBOLS <ls_settings> TYPE mdg_bs_mat_s_mp_settings_data.

* get field values displayed on FPM page
  io_ovs_callback->get_row_data( IMPORTING er_data = lr_data ).
  ASSIGN lr_data->* TO <ls_settings>.

* fill selection fields with values from FPM page
  ls_data-... = <ls_settings>-...


* set fields and field values displayed on OVS selection screen
  io_ovs_callback-
>set_input_structure( EXPORTING input                     = ls_data
                                                display_values_immediately = a
bap_false ).
endmethod.
```

Phase 2 for the result list: A select can take place here, or use the GET_ROW_DATA method to read all fields from the UI. The displayed table is generated from the given data type.

Method: /PLMU/IF_FRW_G_OVS~HANDLE_PHASE_2

```
METHOD /plmu/if_frw_g_ovs~handle_phase_2.
  DATA lr_data     TYPE REF TO data.
  DATA lt_message  TYPE /plmu/if_frw_g_ovs_callback=>ty_t_messages.
  DATA lt_result   TYPE TABLE OF ty_s_result.
```

```
  FIELD-SYMBOLS <ls_selection> TYPE mdg_bs_mat_s_mp_settings_data.

  CLEAR lt_message.

* get OVS selection screen field values
  io_ovs_callback->get_query_parameters( IMPORTING er_data = lr_data ).
  ASSIGN lr_data->* TO <ls_selection>.

* generate data to be displayed in OVS result list (considering <ls_selection>)
  lt_result = ...

* set messages to be displayed with OVS result list
  IF lt_message IS NOT INITIAL.
    io_ovs_callback->set_messages( messages = lt_message ).
  ENDIF.

* show OVS result list
  io_ovs_callback->set_output_table( lt_result ).

 ENDMETHOD.
```

Phase 3 for the result list: Transfer selected value to field on UI

Method: /PLMU/IF_FRW_G_OVS~HANDLE_PHASE_3

```
METHOD /plmu/if_frw_g_ovs~handle_phase_3.

DATA lr_data      TYPE REF TO data.
  DATA lr_selected  TYPE REF TO data.


  FIELD-SYMBOLS <ls_data>     TYPE mdg_bs_mat_s_mp_settings_data.
  FIELD-SYMBOLS <ls_selected> TYPE ty_s_result.

* get data of selected row in OVS result list
  io_ovs_callback->get_selection( IMPORTING er_data = lr_selected ).
  ASSIGN lr_selected->* TO <ls_selected>.

* get reference to fields on FPM page
  io_ovs_callback->get_row_data( IMPORTING er_data = lr_data ).
  ASSIGN lr_data->* TO <ls_data>.

* copy data of selected row in OVS result list to data structure <ls_data>;
* See example coding in chapter
* 5.10.3 Example Coding to build up Value List for Dropdown List and Value Help
  <ls_data>-... = <ls_selected>-...
* set field values on FPM page
  io_ovs_callback->set_row_data( <ls_data> ).
ENDMETHOD.
```

3.10.3 Example Coding to build up Value List for Dropdown List and Value Help

```
  DATA: lo_model     TYPE REF TO if_usmd_model_ext.
  DATA: lt_sel       TYPE usmd_ts_sel,
        ls_sel       LIKE LINE OF lt_sel,
        lt_message   TYPE usmd_t_message.
  DATA: lr_data      TYPE REF TO data.
  FIELD-SYMBOLS:
        <lt_data>    TYPE ANY TABLE,
        <ls_data>    TYPE any.
* ---------------------------------------------------------------
* get model access
  CALL METHOD cl_usmd_model_ext=>get_instance(
    EXPORTING
      i_usmd_model = 'MM'
    IMPORTING
      eo_instance  = lo_model ).
* ---------------------------------------------------------------
* create container to retriece data
  lo_model->create_data_reference(
    EXPORTING
      i_fieldname  = 'XXXX'    " Field Name
      i_struct     = if_usmd_model_ext=>gc_struct_key_txt
    IMPORTING
      er_data      = lr_data
      et_message   = lt_message ).
* ---------------------------------------------------------------
* selection
  ls_sel-sign      = 'I'.
  ls_sel-option    = 'EQ'.

  ls_sel-fieldname = 'XXXX'.   " Field Name
  ls_sel-low       = 'YY'.     " Field Value
  INSERT ls_sel INTO TABLE lt_sel.
* ...
  * ---------------------------------------------------------------
* read data
  ASSIGN lr_data->* TO <lt_data>.
  lo_model->read_char_value(
    EXPORTING
      i_fieldname = 'XXXX'   " Field Name
      it_sel      = lt_sel
    IMPORTING
      et_data     = <lt_data>
      et_message  = lt_message ).
```

3.10.4 Example: Derive Material Number

The material number of a new material is

- Either specified by the end user (on the initial screen or at any time during processing the change request)
- Or derived internally using a number range during activation of the change request

Scenario:

You want to derive the material number based on the material type and the temporary key provided by the MDGAF.

Solution:

Implement an enhancement of type Post Exit for method BEFORE_FLUSH of class CL_MDG_BS_MAT_FEEDER_FORM_MAT (or CL_MDG_BS_MAT_FEEDER_FORM_MARA, respectively, depending on implementing notes – see 4.2.1). Or, if this feeder class has been reimplemented, also reimplement the method BEFORE_FLUSH by calling the SUPER->BEFORE_FLUSH and then add the coding for material number creation.
Post exits are modification-free enhancements. But you have to verify all these exits after upgrading. (http://help.sap.com/saphelp_smehp1/helpdata/en/c5/f4b9422e0fb911e10000000a1550b0/content.htm)

```
  METHOD ipo_zibtest_mara_matnr~before_flush.
*"----------------------------------------------------------------------*
*" Declaration of POST-method, do not insert any comments here please!
*"
*"methods BEFORE_FLUSH
*"  changing
*"    !CT_CHANGE_LOG type FPMGB_T_CHANGELOG .
*"----------------------------------------------------------------------*
* Replace a temporary MATNR with a concatenated MATNR
* as soon as a material type (MTART) is specified
    DATA: lv_matnr_old TYPE matnr.
    DATA: lv_matnr_new TYPE matnr.
    DATA: BEGIN OF ls_matnr_old,
            material TYPE matnr,
          END OF ls_matnr_old.
    DATA: lv_mtart     TYPE mtart.
    DATA: lv_is_tmp    TYPE xfeld.
    DATA mo_model    TYPE REF TO if_usmd_model_ext.
    DATA: ls_change_log TYPE fpmgb_s_changelog.

    FIELD-SYMBOLS <lv_matnr_new> TYPE matnr.
    FIELD-SYMBOLS <lv_matnr_old> TYPE matnr.


    " Change of MATNR only possible during CREATE
    CHECK cl_mdg_bs_mat_assist_ui=>gv_action_mode =
          cl_mdg_bs_mat_c=>gc_usmd_action_create.
    " Get current MATNR
    core_object->mo_context->get_attribute(
       EXPORTING
         iv_name = cl_mdg_bs_mat_c=>gc_matnr
       IMPORTING
         ev_value = lv_matnr_old ).
    " Check whether current MATNR is temporary
    CALL METHOD cl_usmd_model_ext=>get_instance
      EXPORTING
        i_usmd_model = if_mdg_bs_mat_gen_c=>gc_model_mm
      IMPORTING
        eo_instance  = mo_model.
    ls_matnr_old-material = lv_matnr_old.
    lv_is_tmp = mo_model->is_temporary_key(
        i_fieldname = if_mdg_bs_mat_gen_c=>gc_fieldname_material
        is_key      = ls_matnr_old ).
    IF lv_is_tmp = abap_true.
      " Replace the following line with your MATNR logic
      " Derive new MATNR from MTART and temporary key
      core_object->mo_context->get_attribute(
```

```
        EXPORTING
          iv_name = cl_mdg_bs_mat_c=>gc_field_mtart
        IMPORTING
          ev_value = lv_mtart ).
      IF NOT lv_mtart IS INITIAL.
        CONCATENATE 'MDGM' lv_matnr_old lv_mtart INTO lv_matnr_new.
        " Note: UI later on checks (during a FLUSH) that:
        " - Old and new MATNR are different
        " - New MATNR doesn't yet exist
        " => No call needed to cl_mdg_bs_mat_assist_ui=>exist_anywhere
        core_object->mo_context->set_attribute(
           EXPORTING
             iv_name = cl_mdg_bs_mat_c=>gc_matnr
             iv_value = lv_matnr_new ).

        " Set gv_current_material is important for saving
        cl_mdg_bs_mat_assist_ui=>gv_current_material = lv_matnr_new.
        " Add this MATNR change also to change log
        ls_change_log-name = cl_mdg_bs_mat_c=>gc_field_matnr.
        CREATE DATA ls_change_log-new_value TYPE matnr.
        ASSIGN ls_change_log-new_value->* TO <lv_matnr_new>.
        <lv_matnr_new> = lv_matnr_new.
        CREATE DATA ls_change_log-old_value TYPE matnr.
        ASSIGN ls_change_log-old_value->* TO <lv_matnr_old>.
        <lv_matnr_old> = lv_matnr_old.


        ls_change_log-line_index = 1.
        INSERT ls_change_log INTO TABLE ct_change_log.
      ENDIF.
    ENDIF.
  ENDMETHOD.                    "IPO_ZIBTEST_MARA_MATNR~BEFORE_FLUSH
```

3.10.5 Example: OVS

SAP Note 1976017 provides a toolset to support OVS, for flex entities. The generic OVS help is delivered with this SAP Note for EhP6, MDG6.1, and MDG7.0. The note also contains a code example of how to enable it.
For releases before MDG 7.0 SP02 this note also contains corrections of the code to ensure that the mentioned method is properly called. With MDG7.0 SP02 it is only necessary to implement the enhancement.

3.11 Adjustment of Header Line

Adjusting the header line can be done by enhancement of method SET_TITLE of class CL_MDG_BS_MAT_ASSIST_UI.

Starting with MDG 6.1 SET_TITLE is only used for the Initial Page. At the Single Object Maintenance (SOM) OVP for Material the methods ADJUST_PAGE_TITLE and ADJUST_UIBB_TITLES of class CL_MDG_BS_MAT_APPCC are used instead.

ADJUST_PAGE_TITLE can be enhanced to adapt the header line of the complete page while ADJUST_UIBB_TITLES can be enhanced to adapt the header lines of the assignment blocks.

Page Title

The page title of the WD OVP component configurations can be adjusted in a way that the current key information is included in these texts.

The following placeholders can be used as part of the page title:

- &TITLE_MATNR& -> Material number

- &TITLE_MAT_DESCR& -> Material description
- &TITLE_MATNR_DESCR& -> Material number or material description if available
- &TITLE_PLANT& -> Plant ID
- &TITLE_VALAREA& -> Valuation area ID
- &TITLE_VALAREA_TYPE& -> Valuation area ID and Valuation type is not initial
- &TITLE_DCHAIN& -> Distribution Chain

At runtime, the title is constructed from these placeholders in a two-step approach:
Step 1:
Placeholders are replaced by text symbols (which itself contain one or multiple placeholders). Depending on the placeholder, this replacement may depend on the action mode (create / delete / other mode) and on the value of business data (e.g. if the material description is maintained in the logon language). A translatable separator is attached automatically (';' in language EN).

Example 1:
Placeholder: &TITLE_MAT_DESCR&
Action mode: CREATE
Material number: 12345678
Result of replacement: "Material: New &MAT_DESCR&; "

Example 2:
Placeholder: &TITLE_MAT_DESCR&
Action mode: CREATE
Material number: *initial*
Result of replacement: "Material: New; "

Step 2:
The placeholders of the text symbols are replaced by current values of related key fields (e.g. material number, plant ID). For "&MAT_DESCR&" this replacement depends on the existence of the material description in the logon language. For "&VALUATION&" this replacement depends on the valuation area kind.

Example1:
Text: "Material: New &MAT_DESCR&; "
Material number: 12345678
Material description: *not maintained in logon language*
Result of replacement: "Material: New 12345678"

Example2:
Text: "Material: New &MAT_DESCR&; "
Material number: 12345678
Material description: "Mountain bike"
Result of replacement: "Material: New Mountain bike"

Application Example:
Page title as configured:
&TITLE_MAT_DESCR&&TITLE_PLANT&&TITLE_VALAREA_TYPE&&TITLE_DCHAIN&

Action mode: CHANGE
Material number: 12345678
Material description: "Mountain bike"
Plant ID: ZAPF
Distribution Channel: 01
Sales Organization: 0001
Valuation area kind: Plant
Valuation Area ID: ZAPF
Valuation Type: *initial (header data set)*

Page title at runtime:
“**Material: Mountain bike; Plant: ZAPF; Valuation Area (Plant): ZAPF; Distribution Chain: 0001/01**”

3.12 Adjustment of UIBB Titles

Panel titles of the WD OVP component configurations can be adjusted in a way that the current key information is included in these texts.

The following placeholders can be used as part of the page title:

- &MATNR& -> Material number
- &MAT_DESCR& -> Material description
- &MATNR_DESCR& -> Material number or material description if available
- &WERKS& -> Plant ID
- &VKORK& -> Sales organization
- &VTWEG& -> Distribution channel
- &BWKEY& -> Valuation area ID
- &BWTAR& -> Valuation type
- &VALUATION& -> Valuation area kind (“Company Code” or “Plant”)
- &LGNUM& -> Warehouse Number
- &LGORT& -> Storage Location
- &LGTYP& -> Warehouse Storage Type
- &MDMABERID& -> MRP Area
- &MKALVERID& ->Production Version
- &QPART& -> Quality Inspection Setup Type

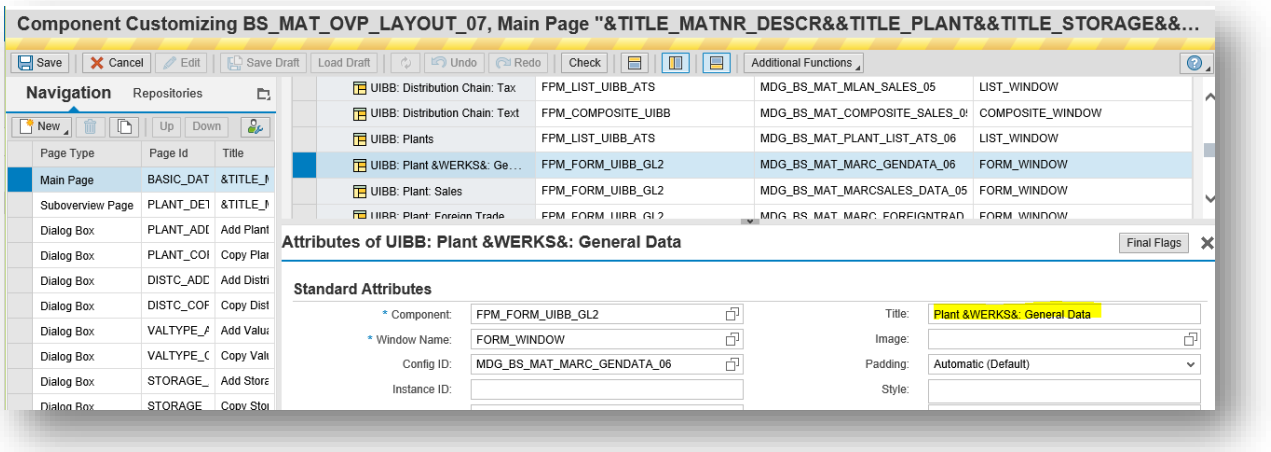
At runtime, the panel titles are constructed from these placeholders as follows.

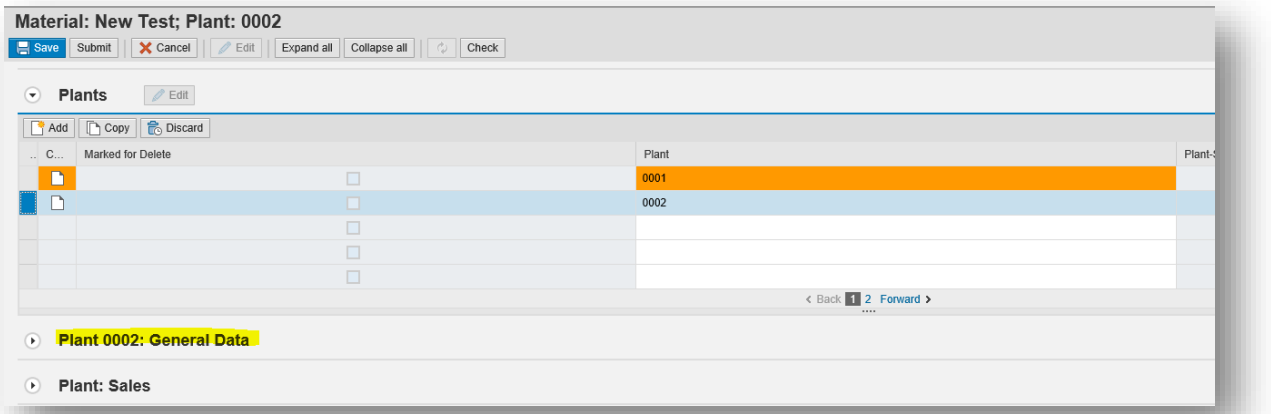
Placeholders are replaced by current values of related key fields (e.g., material number, plant ID). For “&MAT_DESCR&” this replacement depends on the existence of the material description in the logon language. For “&VALUATION&” this replacement depends on the valuation area kind.

Application Example:

Panel title as configured: “Valuation Area (&VALUATION&) &BWKEY&”
Valuation area kind: Plant
Valuation area ID: ZAPF
Result of replacement: “Valuation Area (Plant) ZAPF”

Example Plant:





3.13 Search Help Sequence (in OVP and Search UI)

For the different types of fields, the value help is determined with a different sequence. If one value help is determined the following steps are not executed.

Sequence of consideration for input fields in OVP and Search UI:

1. Customer defined OVS according to chapter 5.10.5 Example: OVS.
2. Search help assignment in data model definition
3. Only OVP: 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
4. Search help assignment on data element (for flex entities)
5. Fixed values on domain
6. Value table on domain (implemented as OVS with SAP note 2003660)

Sequence of consideration for Dropdown List in OVP and Search UI:

1. Fixed values on domain
2. Value table on domain

Sequence of values for characteristics in the Search UI (prerequisite SAP Note 2026600):

1. If check table is assigned to characteristic:

a. Search help assigned to key field of check table
b. Value help derived from assigned check table
c. Search help assignment on data element

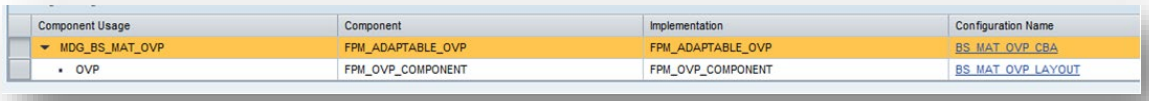
2. Fixed values in characteristics and no additional values allowed → Dropdown list for the values
3. Fixed values in characteristics and additional values allowed → Fixed values as OVS search help

Sequence of values for characteristics in the OVP:

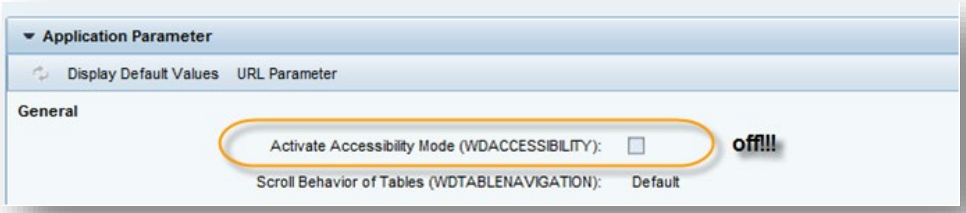
1. If check table is assigned to characteristic:
   a. Search help assigned to key field of check table
   b. Value help derived from assigned check table
   c. Search help assignment on data element
2. Fixed values in characteristics → Pop Up with radio buttons/check boxes to select the values

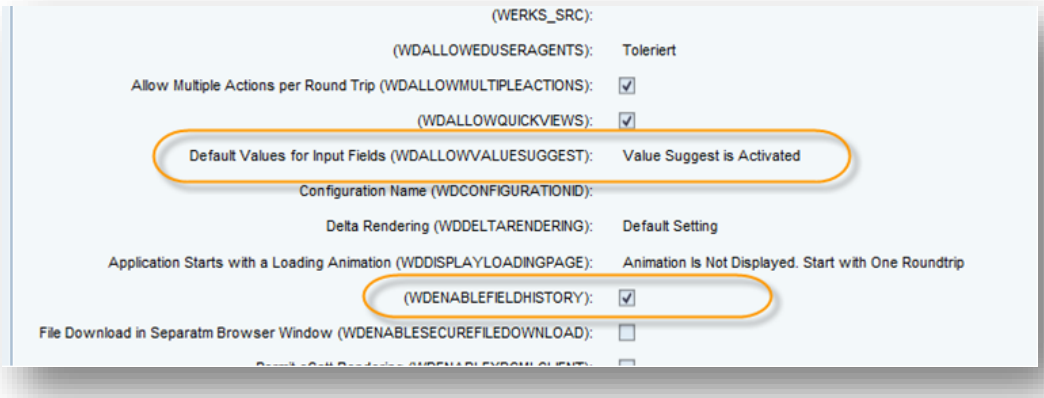
3.14 Type Ahead and Field History

Type ahead and field history can be configured in the application configuration (SE80 → Web Dynpro → Web Dynpro Application → Application Configuration)

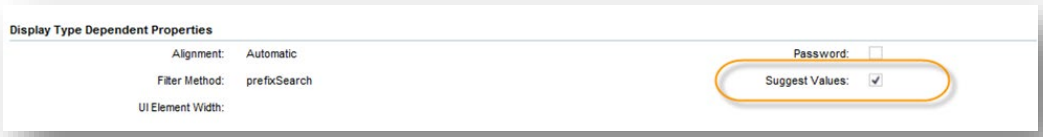


Accessibility Mode must be off.





If you activate ‘type ahead’, then you have to enable this for each field in their attributes.



Type ahead works only if there is DDIC value help available (Type ahead and OVS are not supported

4 Other UI Hints

4.1 Performance for UIBB

Scenario:
If no entity is maintained in the feeder class parameters of an UIBB, the performance of single processing is low. A reason of the low performance is that in every retrieve, insert, update, delete, and in every calculation of the field properties of an UIBB all entities of the corresponding node are considered.

Solution:
In the feeder class parameters of every component configuration of an UIBB a table “Entity Types” is provided. In this table, you maintain the entities whose fields should be shown on the UI. With this the performance of retrieve, insert, update, delete, and of the calculation of the field properties will be improved, since only the maintained entities will be considered.

Implementation:

This refers to all UIBBs (form, list, and CBA UIBBs), especially to the ones corresponding to the nodes “MARA” and “MARC”. The implementation should be realized in 4 steps and in the following order:

Step 1: Open the component configuration of an UIBB and remove all fields from its UIBB schema, which are superfluous, unnecessary, not useful, or not important to be shown on the UI.

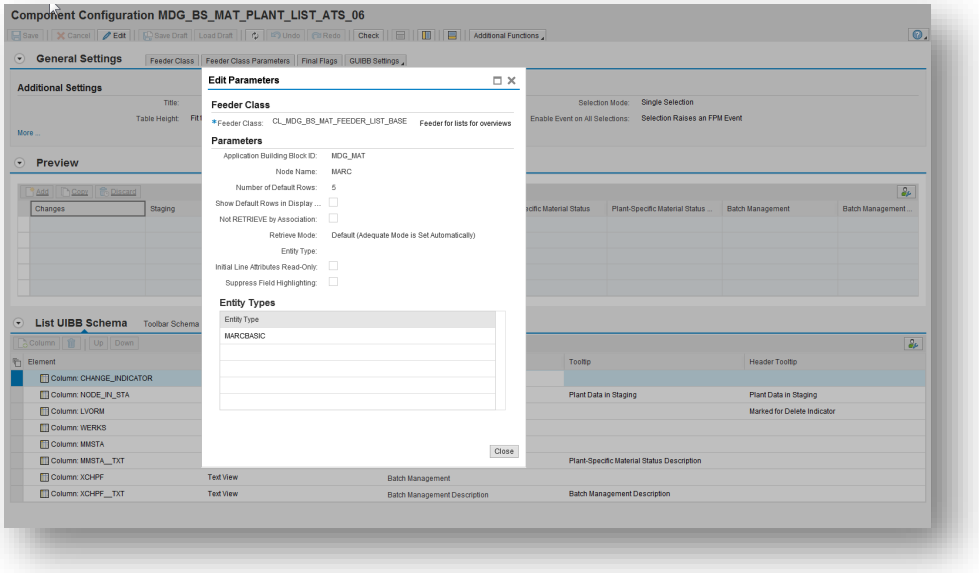
Step 2: By using the report “MDG_BS_MAT_FIND_MODEL_DETAILS” find out the entities which cover the remaining fields.

Step 3: In the feeder parameters maintain the found entities in the table “Entity Types”.

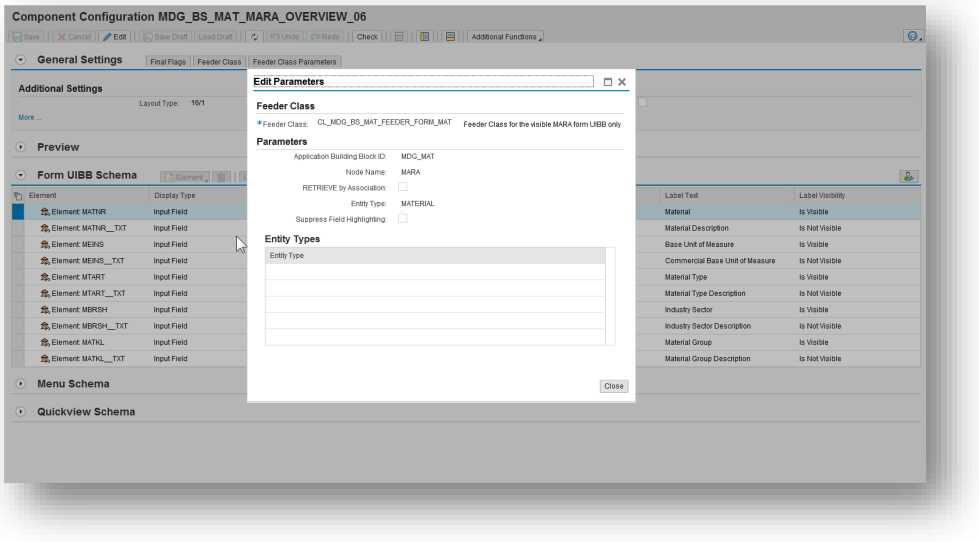
Step 4: Chose “Check”. If no errors are shown, then chose “Save”.

Note, that you must configure the additional feeder parameter also per CBA configuration you are using (except the configurations to hide an UIBB) if you want to get the complete performance improvement.

Example 1: List-UIBB



Example 2: Form-UIBB



4.2 Set OVP UI Immediately in EDIT

The standard behavior of the UI is first to display the material even if you want to change a material. You have to click on the Edit Button to switch to the Edit mode. In the standard UI, no difference is made between the action ‘DISPLAY’ and ‘CHANGE’.

Scenario:
You want to have the UI immediately in EDIT mode.

Solution:
It is recommended only to switch directly to edit mode if the material is chosen with a change request.

Because, if the material is displayed without a change request and the UI would be switched to edit mode automatically. This would always happen, independent from the navigation path (respectively calling application):

1. The system always tries to create a change request, independent of the calling application.
   a) If the user doesn’t have the necessary authorizations, an error message is displayed. The UI remains in display mode.
   b) If the material is already contained in an exclusive change request (e.g., CR type MAT02), a corresponding error message is displayed. The UI remains in display mode.
   c) If the material is already displayed by another user, the current users gets an error message, that the material can’t be locked. The UI remains in display mode.
   d) To create a change request a change request type is necessary. If more than one possible change request type is available, the user is asked with a popup to select the change request type. If the user cancels the selection, the material is displayed only.
2. If it was possible to create a change request, but the user had the intension to only display the material, he has to cancel the UI. Otherwise, a change request is created accidentally.
   a) If he cancels the UI, he gets a loss of data confirmation popup.

For the recommended solution, you have to enhance the code in class CL_MDG_BS_MAT_ASSIST_UI method SET_UI_READ_ONLY_CHECK.
Necessary SAP Note: 1719737. There you will also find the example code.

For releases before MDG7.0 this note also contains corrections of the code to insure, that the mentioned method is properly called. With MDG7.0 it is only necessary to implement the enhancement given in the SAP Note.

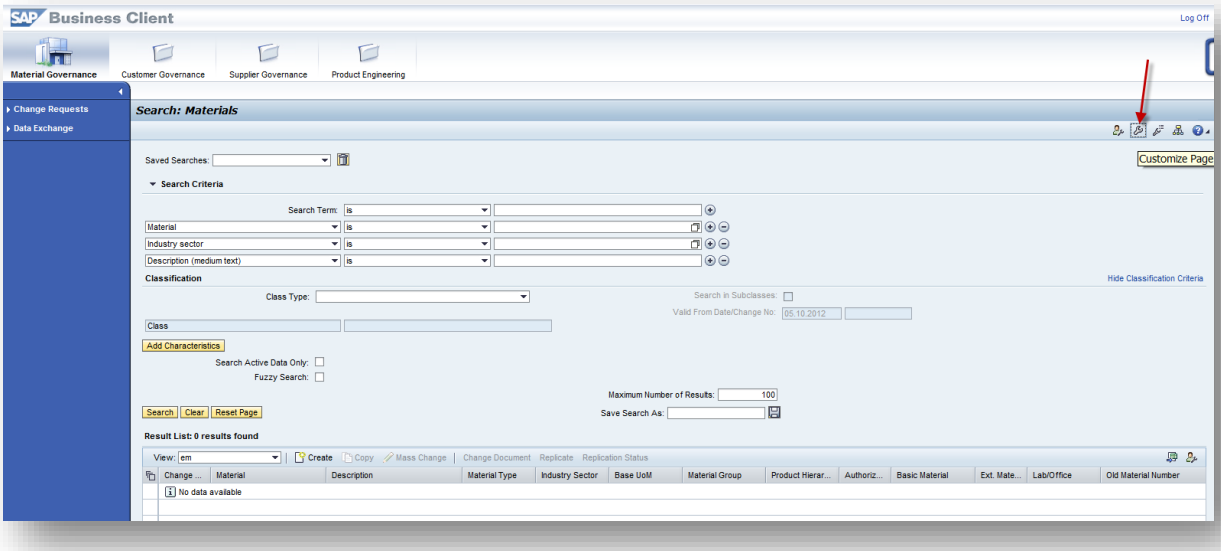
If the example code of the SAP Note is implemented in your system, the UI switches directly to edit mode in case the material is displayed within a change request. If the current user is not a valid processor of the work item, an error message is displayed, that changes are not allowed. The UI remains in display mode.

To avoid this error message or to restrict the automatic edit mode switch you have to adapt the IF condition in your enhancement according to your requirements.
To switch dependent on the navigation path to edit mode, you can use URL parameters. Add for example a parameter ZMDG_EDITMODE = 'X' in your role to the menu entry 'Change Material'. This parameter is automatically passed to the OVP and stored in the internal table CL_MDG_BS_MAT_ASSIST_UI=>GT_CUSTOM_URL_PARAM. You can retrieve and evaluate it according to your requirements.

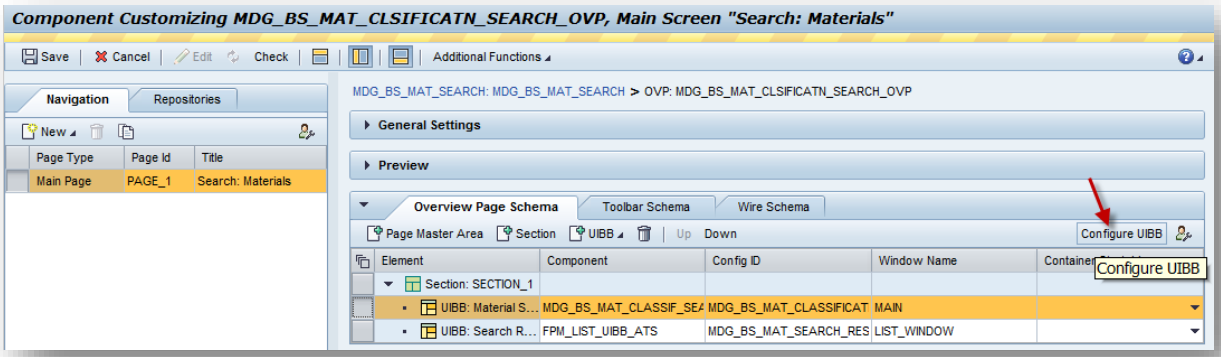
4.3 Search UI: Hide Classification

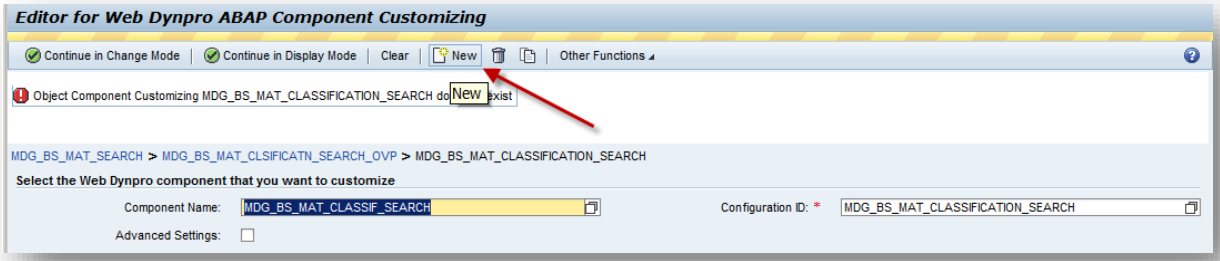
Scenario:
You want to hide the classification on the Search UI.

Solution:
Start Search UI in Customizing mode (in NWBC with &sap-config-mode=A or user parameter FPM_CONFIG_EXPERT = A).

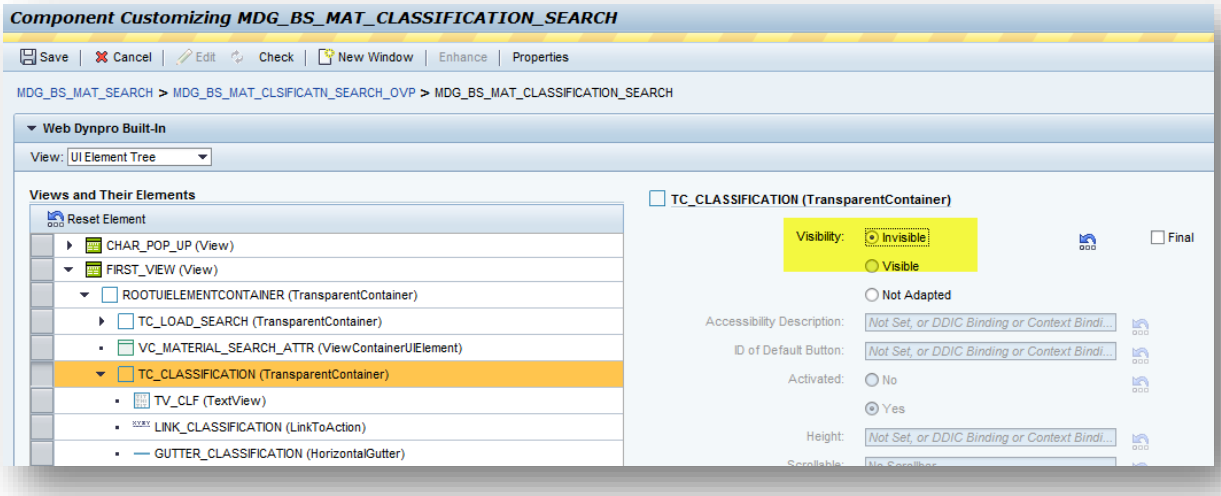


Configure UIBB: Material Search.

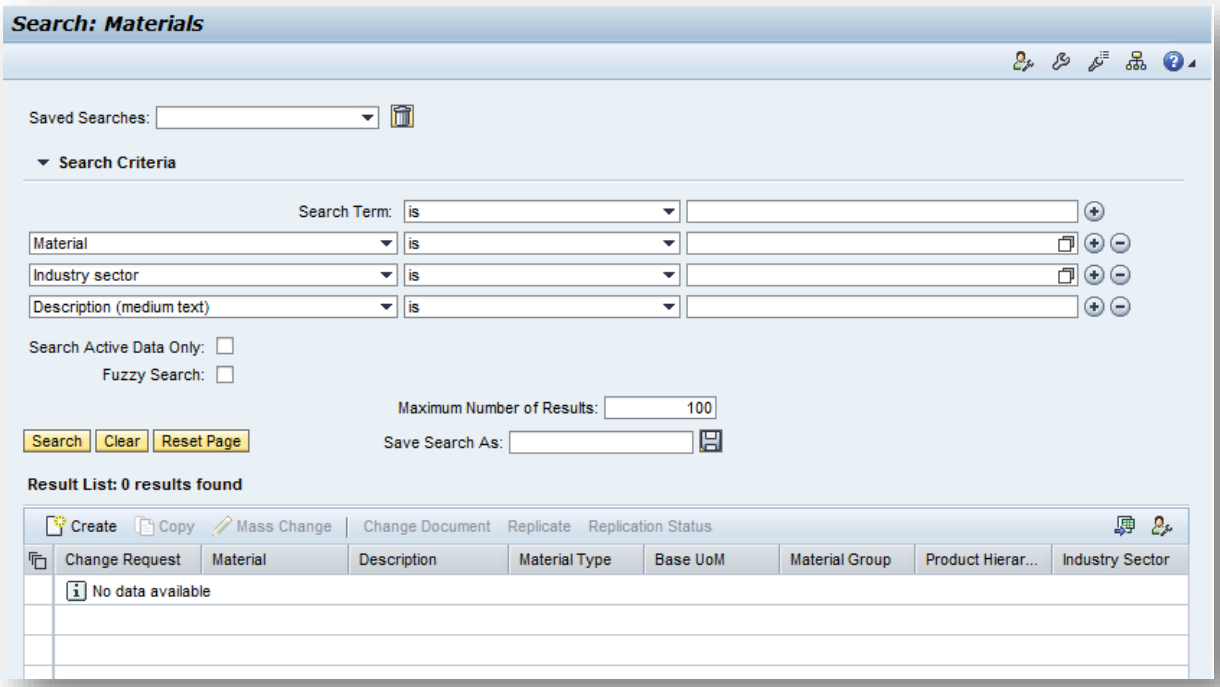




Set TC_CLASSIFICATION to Invisible



Result:



4.4 Governance and Convenience API and Model EXT

Scenario:
You want to implement some enhancements into the MDG functionality.

Restrictions:

- The APIs IF_USMD_CONV_SOM_GOV_API, IF_USMD_GOV_API and the old API IF_USMD_CREQUEST_API may not be used together in one session
- IF_USMD_MODEL_EXT, on the other hand, may be used in combination with any of the above APIs

Therefore, the selection which API should be used must be done on a case-by-case basis.

- Use **IF_USMD_MODEL_EXT** for read-only access, for example implementation for BAdIs of enhancement spots or for implementing a new access class.
- Use **IF_USMD_CONV_SOM_GOV_API** if the code changes material data and runs in the same session as a UI implemented with this API:
  for example, re-implementing the feeder classes CL_MDG_BS_MAT_FEEDER_* or any other feeder class.
- Use **IF_USMD_GOV_API** if the code handles more than one change request in one logical unit of work (LUW); for example, creating test data.

For all other scenarios, we recommend you use the IF_USMD_GOV_API and avoid IF_USMD_CREQUEST_API for new coding. See also How To Guide Master Data Governance Application Programming Interface Guide (http://scn.sap.com/docs/DOC-45127).

4.5 Launch Material UI with Custom Coding

Scenario:
You want to call the Material UI from your custom application.

Example Coding:

```
REPORT zibtest_launch.

DATA: lo_ui_services          TYPE REF TO if_usmd_ui_services_ext,
      lt_application_value    TYPE usmd_t_value,
      lv_application          TYPE string.
FIELD-SYMBOLS:
      <ls_application_value> LIKE LINE OF lt_application_value.

* Provided URL parameters for UI start
* (copied from CL_MDG_BS_MAT_ASSIST_UI=>NAVIGATE_INPLACE)

* OTC material (194)
APPEND INITIAL LINE TO lt_application_value ASSIGNING <ls_application_value>.
<ls_application_value>-fieldname = if_usmd_ui_services_ext=>gc_fld_otc.
<ls_application_value>-value     = cl_mdg_bs_mat_c=>gc_otc_material.

* Business Activity is MAT1
APPEND INITIAL LINE TO lt_application_value ASSIGNING <ls_application_value>.
<ls_application_value>-fieldname = cl_mdg_bs_mat_c=>gc_field_process.
<ls_application_value>-value     = cl_mdg_bs_mat_c=>gc_process_create.

* Change request type is MAT01 (needed in two URL parameters)
APPEND INITIAL LINE TO lt_application_value ASSIGNING <ls_application_value>.
<ls_application_value>-fieldname = cl_mdg_bs_mat_c=>gc_url_crequest_type.
<ls_application_value>-value     = 'MAT01'.
APPEND INITIAL LINE TO lt_application_value ASSIGNING <ls_application_value>.
<ls_application_value>-fieldname = cl_mdg_bs_mat_c=>gc_url_crequest_type_alt.
<ls_application_value>-value     = 'MAT01'.

* Work with change request
APPEND INITIAL LINE TO lt_application_value ASSIGNING <ls_application_value>.
<ls_application_value>-fieldname = cl_mdg_bs_mat_c=>gc_field_staging_mode.
<ls_application_value>-value = abap_true.

* trigger start of new dialog
lv_application = 'MDG_BS_MAT_OVP'.
lo_ui_services = cl_usmd_factory_ext=>get_ui_services_instance( ).

CALL METHOD lo_ui_services->navigate
  EXPORTING
    i_application        = lv_application
    i_target             = if_usmd_ui_services_ext=>gc_navigation_tgt_external
    i_inplace_navigation = abap_true
    i_usmd_action        = cl_mdg_bs_mat_c=>gc_usmd_action_create
    it_data_1            = lt_application_value.
```

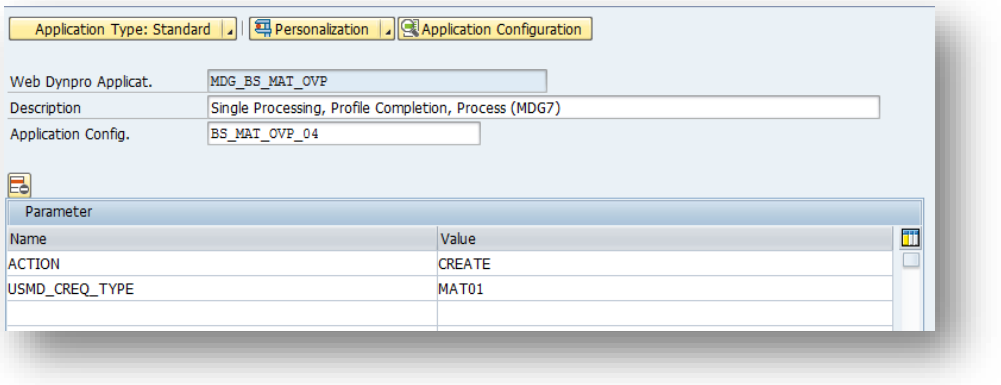
To execute the navigation within a customer owned WebDynpro application:

- Pass the parameter IO_MAIN_COMPONENT in method NAVIGATE of the UI service class; the main component is the WebDynpro component calling the MDG-M UI
- Set the parameter I_INPLACE_NAVIGATION to ABAP_FALSE to avoid side effects to the customer owned WebDynpro (recommendation).

4.6 Launch Material UI with PFCG Menu Entry (Without Initial Screen)

Scenario:
You want to call the Material Overview Page (OVP) from your custom role.

Solution:
Create in the PFCG role an entry like the following to call the application configuration of the OVP directly:



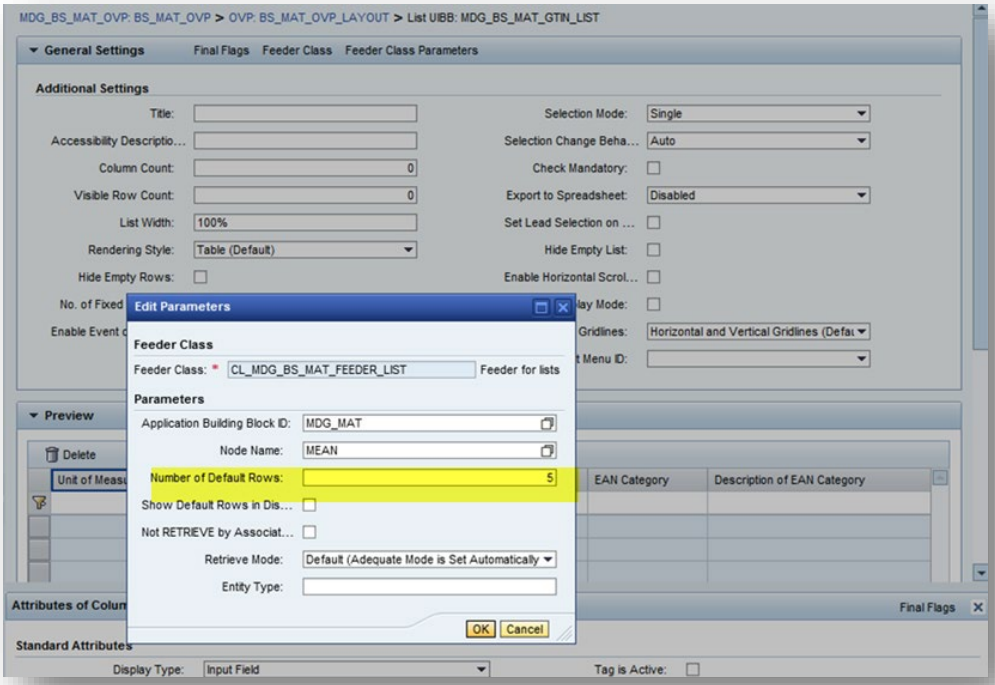
If ACTION=CREATE is used the parameter USMD_CREQ_TYPE=<CR Type> is mandatory. For all other supported actions (DISPLAY, CHANGE, and DELETE) the parameter MATERIAL=<material ID> is necessary.

If you are using CBA depending on the CR type, you also shall set the parameter CRTYPE with the same value as USMD_CREQ_TYPE.

4.7 Scrolling Behaviour of ATS-List (Component FPM_LIST_UIBB_ATS)

The observed scrolling behavior is related to the 5 additional lines which are automatically added to the table content.
If this is not required, there is another option which is anticipated by the application:

The 5 additional lines can be dropped by changing each table configuration (Feeder Class Parameters)



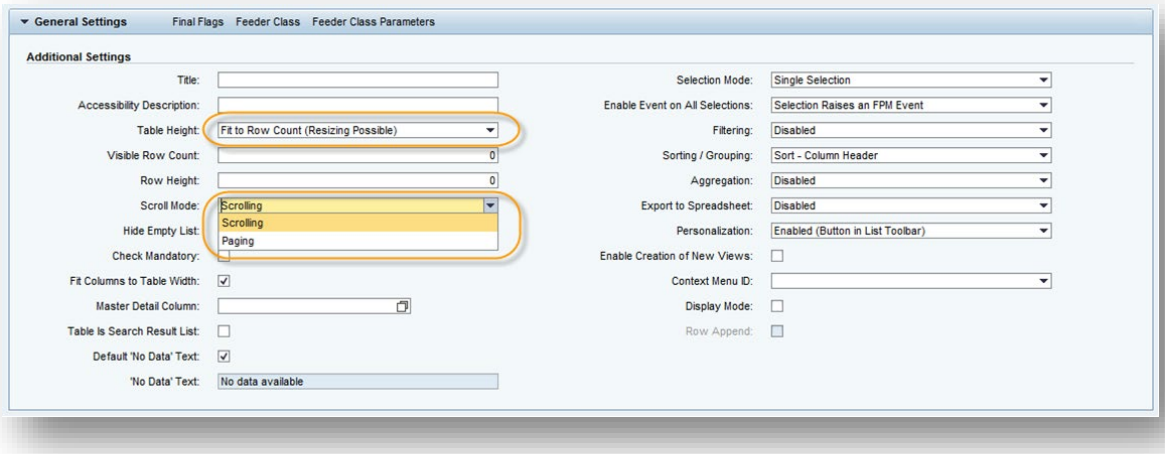
Then, Delete and Insert buttons can be added to the toolbar of each table

- The Delete button has to be assigned the FPM event FRM_DELETE
- The Insert button has to be assigned the FPM event FRM_INSERT

See also chapter 5.7 Additional Buttons above. For EhP6 you need SAP Note 1783881.

Having done this, the two new buttons are visible in change mode and invisible in display mode. In addition, the Delete button is only enabled, if a table row is selected. Inserting a new row and deleting an existing row has already been implemented.

Another option is to change the Scroll Mode from Paging to Scrolling:



4.8 Selection Mode Behaviour of ATS-List (Component FPM_LIST_UIBB_ATS)

Scenario:
You want to select multiple entries but not the first entry in a ATS List. You want to remove the first row selection and then to select the others.

Solution:

Instead of *Multiple selection (plus Lead Selection)* use the *Multiple Selection* mode.

In general, there are three predefined selection modes available, which are:

- Multiple selection (plus Lead Selection)

The first row will be selected. When a second row is selected, you will need to manually remove the lead selection. And if you remove all the selections, the first row will be re-selected by default.

- Multiple Selection

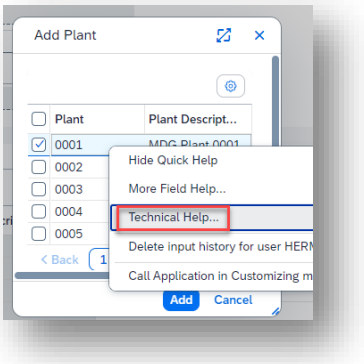
The first row will be selected. When a second row is selected, you will need to manually remove the lead selection. And if you remove all the selections, the first row will not be re-selected by default.

- Single Selection

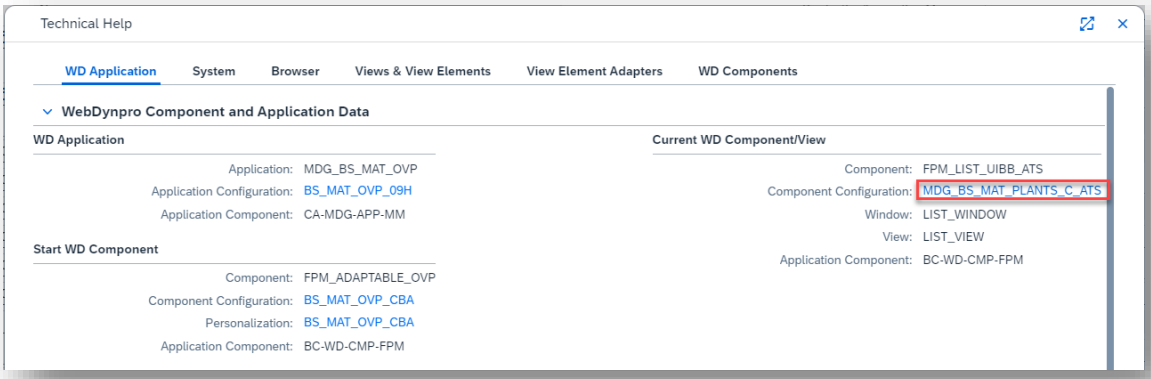
The first row will be selected. When a second row is selected, the check of the lead selection (first row) will be removed. However, it will only be possible to select a single row with this mode.

By default, the initially active selection mode is “Multiple selection (plus Lead Selection)”. However, depending on different scenarios and needs this can be changed via customizing. The steps for the customizing are:

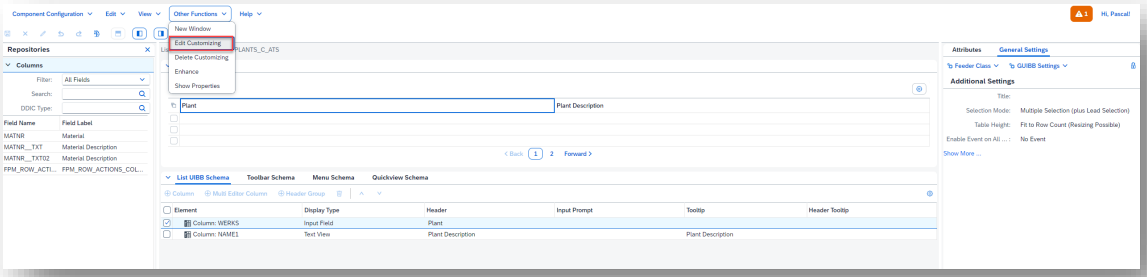
1.) Within the pop-up, in this example the “Add Plant” of the UIBB “Plants”, right-click into the result list and select “Technical Help…”:



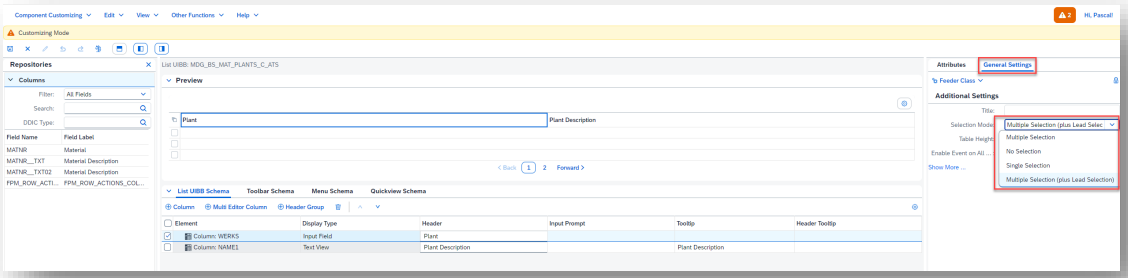
2.) In the appearing “Technical Help” window select the component configuration hyperlink, in this example “MDG_BS_MAT_PLANTS_C_ATS”:



3.) A new tab for the selected component configuration will appear, select “Other Functions” → “Edit Customizing”:



4.) Under “General Settings” you can now select the desired selection mode from the dropdown menu for “Selection Mode”:



Annotation: By design, it is not possible to have no entry selected when opening a pop-up. It is expected and correct that the first entry will always be selected when opening such a pop-up.

4.9 Provide Description in Logon Language

In the UI configurations shipped with the standard MDG-M, the description is provided as a table. If you want to display/maintain a single line with the description in the logon language, the following options are available:

- Component configuration MDG_BS_MAT_DESCR_FORM provides you with one line to maintain the material description in logon language

- Component configuration MDG_BS_MAT_MATERIAL_COMP provides you with the full ‘Basic Data’ UIBB, including one line to maintain the material description in logon language

As a precondition, you must ensure that SAP Note 1750750 is implemented.

4.10 Enhance Copy functionality

4.10.1 Copy all Plant Assignments

Standard behavior from MDG 6.1 is that only up to one plant assignment is copied if a template material is specified on the initial screen during creation.

Scenario:

You want to copy all plant assignments of the template material to the target material.

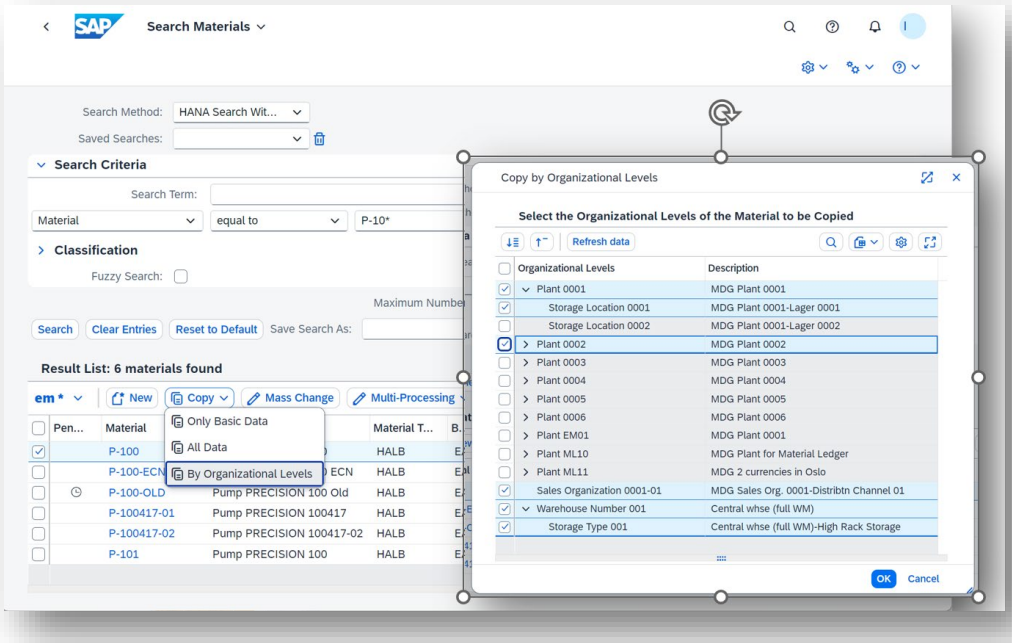
Solution for S/4HANA 2023 and Later:

Starting from SAP S/4HANA 2023, you can implement SAP Notes 3428179 and 3423383 (part of FPS02) to utilize enhanced features for the Copy Button. These features allow you to copy materials using different options tailored to your requirements.

You can copy a selected material from the Search Material Result List and choosing between one of three copying options.

- Only Basic Data: Basic data, long texts, and classifications are copied.
- All Data: Complete data from the template material are copied.
- By Organizational Level: From the dialog box, chose which organizational data you want to copy.

If a pending change request exists for the material, the system copies the material from the pending change request.



Solution for Earlier Versions of S/4HANA:

Technical Background:

If a template material with organizational data is specified, there are two possibilities to implement so that only the specified organizational data is copied:

1. The system reads the complete template material and writes only the specified data to the target
2. The system reads only the specified data from the source and writes this data to the target

For technical reasons the implementation of the copy functionality was enhanced, therefore a solution proposal is described, depending on the implemented notes and support package.

Before the implementation of the SAP Notes 1917085 (MDG 6.1 SP06, MDG 7.0 SP02) and 1965130 (MDG 7.0SP02) the data is filtered only during writing the target material (solution part 1).

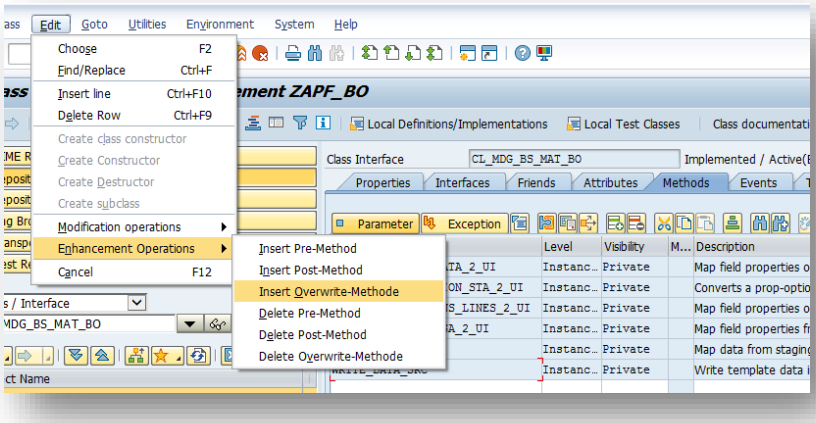
After the implementation of these notes the filtering is done at reading the material data (solution part 1 and 2).

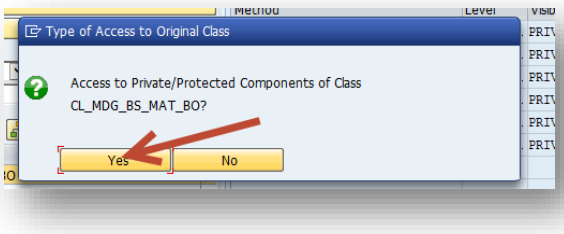
A second kind of filtering is done during writing the data, as the template functionality in standard supports the renaming of organizational data (for example plant 1000 of material A is copied to plant 2000 of a new material). Depending on your implementation of the enhancement the feature of renaming the organizational data may not be possible anymore.

Solution Part 1:

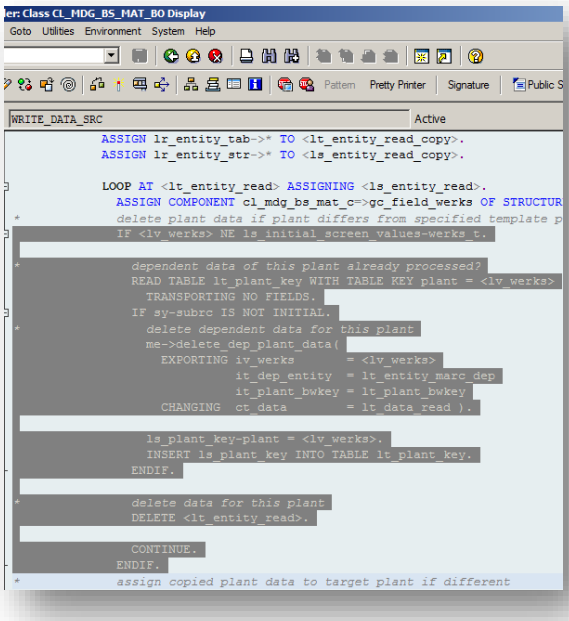
Before implementation of SAP Notes 1917085 (MDG 6.1 SP06, MDG 7.0 SP02) and 1965130 (MDG 7.0 SP02), the data is filtered only during writing the data to the target material.

Implement an enhancement of type Overwrite for method WRITE_DATA_SRC of class CL_MDG_BS_MAT_BO and allow access to private/protected components.





This method is used to transmit the data from the copy template into the buffer for the copy target. Reduce the central “DO 2 TIMES” loop by removing the parts that call the ME->DELETE_DEP_PLANT_DATA. Check the screen shot below for the coding parts to remove.



Add core_object to all private methods.

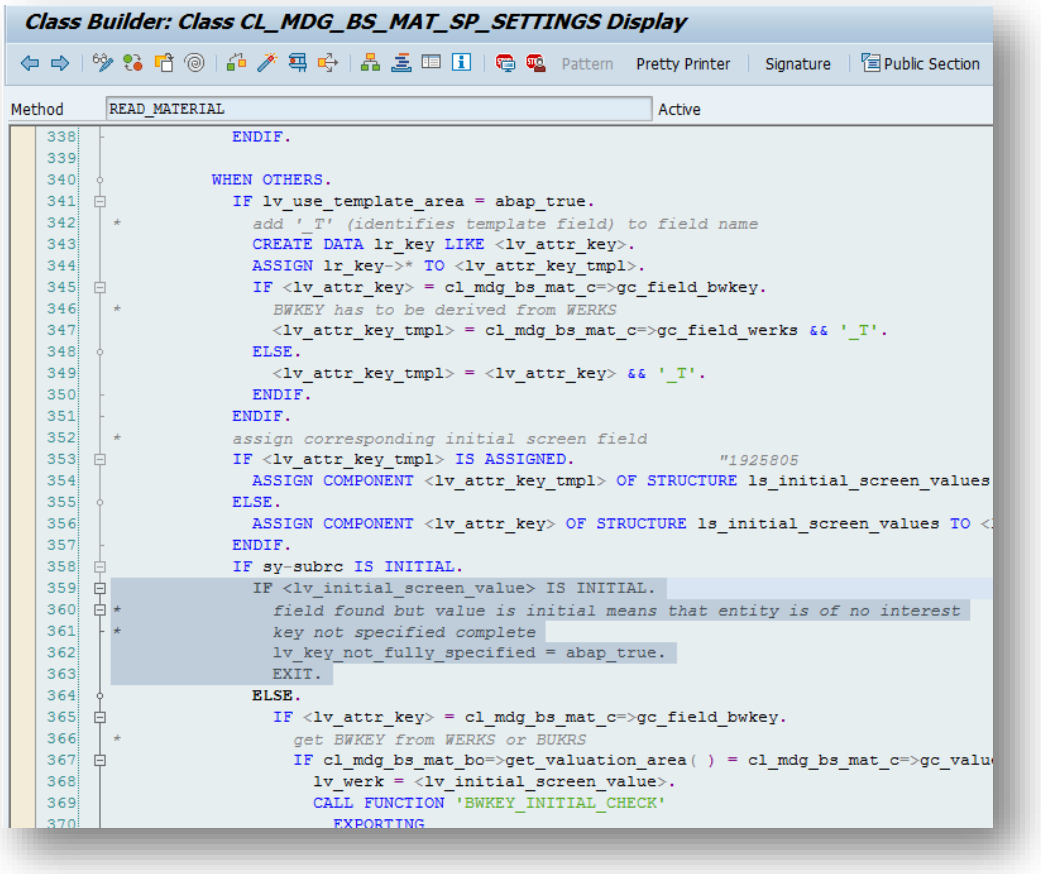
```
*   delete plant and its dependent data from template mater
    core_object->delete_plant_data(
      EXPORTING
        it_entity                = lt_entity_marc
        it_entity_dep            = lt_entity_marc_dep
        it_plant_bwkey           = lt_plant_bwkey
        is_initial_screen_values = ls_initial_screen_values
      CHANGING
        ct_data_read             = lt_data_read ).
*   copy template plant and its dependent data to target pl
    core_object->copy_plant_data(
      EXPORTING
        it_entity                = lt_entity_marc
        it_entity_dep            = lt_entity_marc_dep
        it_plant_bwkey           = lt_plant_bwkey
        is_initial_screen_values = ls_initial_screen_values
      CHANGING
        ct_data_read             = lt_data_read ).
*   delete warehouse number and its dependent data from tem
    core_object->delete_lgnum_data(
      EXPORTING
        it_entity                = lt_entity_mlgn
        it_entity_dep            = lt_entity_mlgn_dep
        is_initial_screen_values = ls_initial_screen_values
```

Solution Part 2:

After implementing SAP Notes 1917085 (MDG 6.1 SP06, MDG 7.0 SP02) and 1965130 (MDG 7.0 SP02), the data is filtered by reading the data from the source material.

In addition to part 1 an enhancement of type Overwrite for method READ_MATERIAL of class CL_MDG_BS_MAT_SP_SETTINGS is necessary. This method is used to read the data from the copy template into the buffer for the copy target.

Check the screen shot below for the coding parts to be enhanced:



With a code like the following all plant data is copied for any entity type with plant as a part of the key:

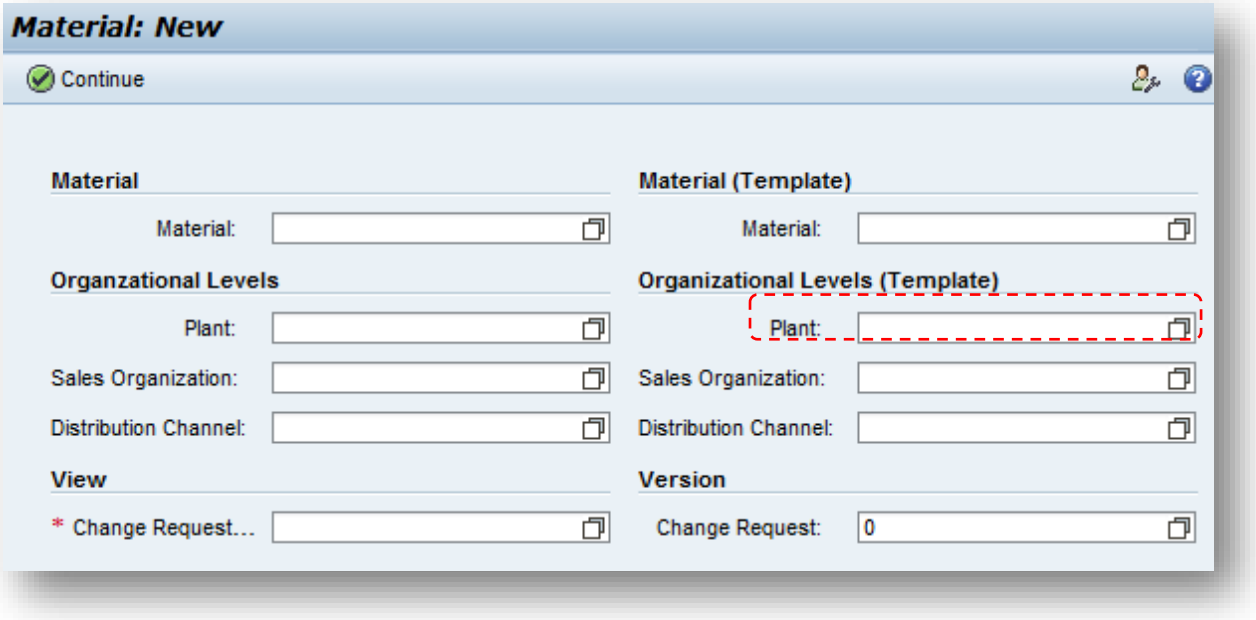
```
...
    IF <lv_initial_screen_value> IS INITIAL.
*     field found but value is initial means that entity is of no interest
*     key not specified complete
      IF <lv_attr_key> = 'WERKS'.
        CLEAR <lv_key_field>.
      ELSE.
        lv_key_not_fully_specified = abap_true.
        EXIT.
      ENDIF.

    ELSE.

...
```

Optional Configuration of the Initial Screen:

You might want to remove the organization level assignment for the organizational data in the template section of the initial screen (see below). This can be done by UI configuration.



4.10.2 Copying Template

It is possible to copy a material into a new Change Request. Usually, the system creates a new change request during copy. A new CR is only created (automatically) if there exists no active one. If the target material (left) is already contained in a CR, this CR is reused. This function can only be used for creating new organizational assignments via template. The copy template must be specified on the initial screen.

Scenario:

You want to create a Change Request with multiple materials using the Template functionality.

Solution:

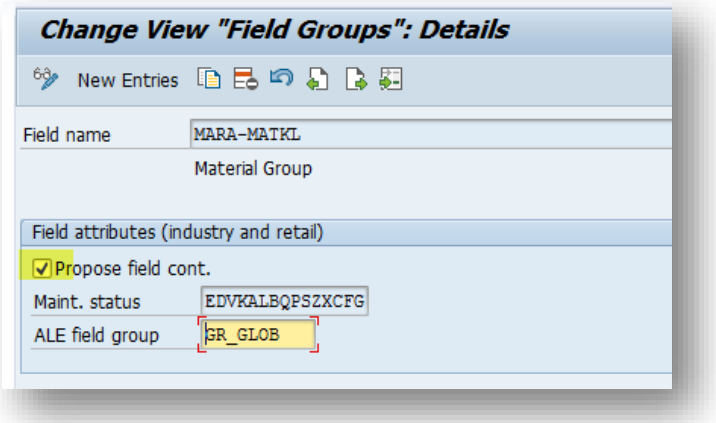
To copy multiple materials into an existing CR, you have to specify the CR ID on the initial screen:

- Enhance the initial screen with an additional input field that provides this parameter. Also, ensure that the CR type is optional or hidden. This can be done by a UI configuration especially for this use case or controlled by the feeder
- A feeder has to be implemented for the initial screen
  - To enhance the value check. Check if the CR is open and the CR type of it is mass enabled
  - The feeder has to transfer the value of the specified target Change Request to the called OVP (via navigation method). The hidden parameter USMD_CREQUEST for target CR is already available

4.10.3 Filter Template Data During Copying

Standard behavior while creating a material with template (copy functionality) is that the complete specified material with the specified organizational levels is used. Also, the settings ‘Propose field cont’ of T130F (Tx

OMSR) in the customizing is considered. MDG also considers this customizing for the long texts (LTXTE_GRUN, LTEXT_IVER, LTEXT_NOTE), which is not possible with the backend transactions (see also SAP Note 662337 - MM01: Unable to deactivate reference handling for long texts).



In addition, you can code your own logic.

Scenario:

You have set up MDG-M so that GTINs/EANs must be unique (can't be used in more than one material). In this case, copying GTINs doesn't make sense, as the new material would be inconsistent. Therefore, you want to filter GTIN information:

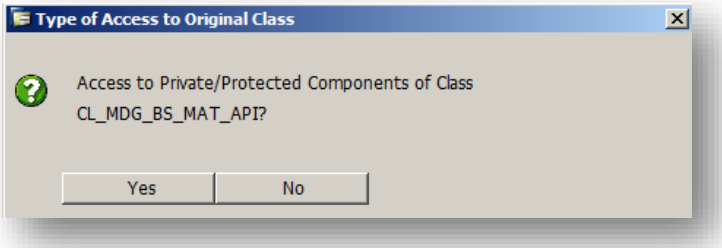
- Remove the entries for entity MEAN_GTIN
- For all entity instances of UNITOFMSR, clear the EAN_MARM, GTIN_VAR2 and NUMTP2
- In entity MATERIAL, clear the EAN_MARA, GTIN_VAR1 and NUMTP1

Solution:

Implement an enhancement of type Post Exit for method SET_DATA_SRC of class CL_MDG_BS_MAT_ASSIST. If the material description is to be filtered, consider method SET_DATA_TXT_SRC of the same class. This method is used to transmit the data from the copy template into the buffer for the copy target. Remove the unwanted records/clear the unwanted fields of table GT_DATA in this Post Exit.

Post exits are modification-free enhancements. But you have to verify all these exits after upgrading. (https://help.sap.com/docs/ABAP_PLATFORM_NEW/fc79a39b30fe4d9aa983bad6787ab9ad/49cc8b49506573ffe10000000a421937.html)

Note: The enhancement must have access to the protected and private attributes of class CL_MDG_BS_MAT_ASSIST. Allow this by clicking 'Yes' on the 'Type of Access popup' when creating the Post Exit:



```
  METHOD ipo_zibtest_gtin_remove~set_data_src.
*"----------------------------------------------------------------------*
*" Declaration of POST-method, do not insert any comments here please!
*"
*"class-methods SET_DATA_SRC
*"  importing
*"    !IT_DATA type USMD_GOV_API_TS_ENT_TABL .
*"----------------------------------------------------------------------*
    DATA: ls_data TYPE usmd_gov_api_s_ent_tabl.
    FIELD-SYMBOLS: <lt_entity> TYPE ANY TABLE.
    FIELD-SYMBOLS: <ls_entity> TYPE any.
    FIELD-SYMBOLS: <lv_ean>    TYPE ean11.
    FIELD-SYMBOLS: <lv_gvar>   TYPE gtin_variant.
    FIELD-SYMBOLS: <lv_numtp>  TYPE numtp.
*   Clear table MEAN_GTIN
    READ TABLE cl_mdg_bs_mat_assist=>gt_data_src INTO ls_data
         WITH KEY entity = if_mdg_bs_mat_gen_c=>gc_entity_mean_gtin.
    IF sy-subrc = 0.
      ASSIGN ls_data-tabl->* TO <lt_entity>.
      IF sy-subrc = 0 AND <lt_entity> IS ASSIGNED.
        CLEAR <lt_entity>.
      ENDIF.
    ENDIF.
*   Clear attribtues EAN_MARM, GTIN_VAR2 and NUMTP2 from entity UNITOFMSR
    READ TABLE cl_mdg_bs_mat_assist=>gt_data_src INTO ls_data
         WITH KEY entity = if_mdg_bs_mat_gen_c=>gc_entity_unitofmsr.
    IF sy-subrc = 0.
      ASSIGN ls_data-tabl->* TO <lt_entity>.
      IF sy-subrc = 0 AND <lt_entity> IS ASSIGNED.
        LOOP AT <lt_entity> ASSIGNING <ls_entity>.
          ASSIGN COMPONENT 'EAN_MARM' OF STRUCTURE <ls_entity>
                 TO <lv_ean>.
          IF sy-subrc = 0 AND <lv_ean> IS ASSIGNED.
            CLEAR <lv_ean>.
          ENDIF.
          ASSIGN COMPONENT 'GTIN_VAR2' OF STRUCTURE <ls_entity>
                 TO <lv_gvar>.
          IF sy-subrc = 0 AND <lv_gvar> IS ASSIGNED.
            CLEAR <lv_gvar>.
          ENDIF.
          ASSIGN COMPONENT 'NUMTP2' OF STRUCTURE <ls_entity>
                 TO <lv_numtp>.
          IF sy-subrc = 0 AND <lv_numtp> IS ASSIGNED.
            CLEAR <lv_numtp>.
          ENDIF.
        ENDLOOP.
      ENDIF.
    ENDIF.
*   Clear attribtues EAN_MARA, GTIN_VAR1 and NUMTP1 from entity MATERIAL
    READ TABLE cl_mdg_bs_mat_assist=>gt_data_src INTO ls_data
         WITH KEY entity = if_mdg_bs_mat_gen_c=>gc_entity_material.
    IF sy-subrc = 0.
      ASSIGN ls_data-tabl->* TO <lt_entity>.
      IF sy-subrc = 0 AND <lt_entity> IS ASSIGNED.
        LOOP AT <lt_entity> ASSIGNING <ls_entity>.
          ASSIGN COMPONENT 'EAN_MARA' OF STRUCTURE <ls_entity>
                 TO <lv_ean>.
          IF sy-subrc = 0 AND <lv_ean> IS ASSIGNED.
            CLEAR <lv_ean>.
          ENDIF.
```

```
          ASSIGN COMPONENT 'GTIN_VAR1' OF STRUCTURE <ls_entity>
                 TO <lv_gvar>.
          IF sy-subrc = 0 AND <lv_gvar> IS ASSIGNED.
            CLEAR <lv_gvar>.
          ENDIF.
          ASSIGN COMPONENT 'NUMTP1' OF STRUCTURE <ls_entity>
                 TO <lv_numtp>.
          IF sy-subrc = 0 AND <lv_numtp> IS ASSIGNED.
            CLEAR <lv_numtp>.
          ENDIF.
        ENDLOOP.
      ENDIF.
    ENDIF.
  ENDMETHOD.                    "IPO_ZIBTEST_GTIN_REMOVE~SET_DATA_SRC
```

Other options for a Pre/Post Exit are

- Post Exit in CL_MDG_BS_MAT_ASSIST=>GET_DATA to filter returning parameter RT_DATA
- Pre Exit in CL_MDG_BS_MAT_BO=>WRITE_DATA_SRC to filter the importing parameter IT_DATA (This is where the key change for the material number is implemented, so this is a complex area)

4.11 “Tabbed” UI

In EhP6 the UI have a different layout than EhP5.

Scenario:

You want to get closer to the EhP5 look-and-feel with the tabbed UI.

Solutions:

1. Stacking:
    a. By Configuration, you can stack the UIBBs so that the org selection table is on the same tab as the detail forms
    b. No coding necessary
    c. Own section for technical UIBB is mandatory (SAP Note 2709506)
    d. Lazy load possible
2. Tabbed UIBB
    a. Create a new configuration for Web Dynpro Component FPM_TABBED_UIBB and embed there the material UIBBs. The UIBBs used in the material OVP can be reused here
    b. Own section for technical UIBB is mandatory (SAP Note 2709506)
    c. Lazy load not possible

4.12 Change Label of Field

Dynamically

Use case: Change label of a field depending on the organizational assignment of the material.

A dynamic exchange of the labels of the fields is not offered by FPM. To change the label dynamically, you have to adapt the UI configuration to use a text field instead of a label.

1. You enhance in feeder method CHANGE_FIELD_DEFINITION the parameter CO_CATALOGUE. Add the new field containing the label here
2. Now you can change the configuration of the UIBB. Disable the label of the field and place the label field at this place instead

3. In the method/PLMU/IF_FRW_G_AFTER_GET_DATA~AFTER_GET_DATA you can set the label resp. text field according to your needs with MO_CONTEXT->SET_ATTRIBUTE

Statically

Replace the field label statically (during the complete execution of the single maintenance UI) by adapting the method /PLMU/IF_FRW_G_FIELD_DEF~CHANGE_FIELD_DEFINITION of class CL_MDG_BS_MAT_FEEDER_FORM (or a class inheriting from it). In parameter CT_DEFINITION you can exchange the description of the field (either by redefinition or a Post Exit enhancement).

Or you can modify the corresponding data element using SE11.

4.13 New Layout for the Classification UIBB with Highlighting Changes

With MDG 7.0 SP04 and SAP_BS_FND 731 SP14 (or SAP_BS_FND 747 SP07), a new layout for the classification UIBB with support of highlighting changes is provided. By default, colors and tooltips are used in the classification UIBB as in all other UIBBs.

However, you can adapt the setting of colors and tooltips, value field length, and page width by modifying the CHANGING parameter RS_UI_SETTINGS with a POST-Exit in the COMPONENTCONTROLLER of WD Component WDC_MDG_BS_CLF_GEN.

These are the standard settings:

```
rs_ui_settings-use_colors_tooltips  = abap_true.
rs_ui_settings-long_value_fields    = abap_true.
rs_ui_settings-full_page_width      = abap_false.
```

In addition, editable fields can be set to ready-only. Change the corresponding values of the parameters CS_CLASSES_DISP_OPT, CT_CLASS, and CT_VALUE with a PRE-Exit in method SET_DISPLAY_OPTIONS of the COMPONENTCONTROLLER in WD Component WDC_MDG_BS_CLF_GEN.

Note:

If you have not installed the mentioned SP for SAP_BS_FND, you can downport the functionality through SAP Note 2036900.

If you have installed the mentioned SPs but you still want to use the old layout you have to implement a POST-Exit in the CLASSCONSTRUCTOR of CL_MDG_BS_CLF_LEAD_OBJECT:

```
CL_MDG_BS_CLF_LEAD_OBJECT=>SET_PLM_NEW_UI( abap_false ).
```

Note:

If you want to hide the engineering change fields, you need to create a post-exit for method WDDOMODIFYVIEW of view V_ROOT in webdynpro component WDC_MDG_BS_CLF_GEN.

4.14 Omit checks

Scenario:

Certain checks can be omitted per CR step via customizing as follows:

Customizing via TA MDGIMG: General Settings -> Process Modelling -> Change Requests -> Configure Properties of CR steps

- Edit "Change Request Step" for selected CR type
- Edit "Enhancements and Checks per Change Request Step" for selected CR step

- => switch of checks that should not be carried out when checks are executed (triggered by user action or by framework)

However, this customizing is respected for all checks, either checks triggered by user using CHECK button or by the framework.

You want to speed-up navigation, and therefore you want to deactivate checks triggered by framework (for example returning from sub overview page to main page), so that the full check is only executed when user press check button, and also when he chooses any action button.

Solutions:

In order to omit all checks at certain points of time, you could enhance method CHECK_DATA (CL_MDG_BS_BOL_TRANSACTION):

- Create pre-exit

- Return to caller of method for FPM events for which CHECKS are not desired

- For example, no checks for FPM event if_fpm_constants=>gc_event-done_and_back_to_main (return from sub overview page to main page)

5 Additional Information

5.1 Further Reading

5.1.1 Information on SAP MDG on SAP S/4HANA

- Exchange knowledge: SAP Community | Q&A | Blog
- Try SAP Master Data Governance on S/4HANA for free: Trial Version
- Try SAP Master Data Governance on S/4HANA on the SAP Cloud Appliance Library: S/4HANA 2022 FPS1
- Learn more: Latest Release | Help Portal | How-to Information | Key Presentations

5.1.2 SAP Roadmap Explorer

- Please see the roadmap for SAP Master Data Governance

5.1.3 Related Information

- Learn more: Floorplan Manager for Web Dynpro ABAP | How to Adapt FPM | FPM Blog | How-to Information | Service Mapping Tool | SAP S/4HANA Cookbook CVI

5.2 SAP Notes

In addition to the detailed explanations written in this document, please see the following SAP Notes for further important information.

Note	Description
3311039	MDG Customer Connection 2023
3043582	MDG Customer Connection 2020
3194967	MDG Customer Connection 2021 for S/4HANA 2022
3311039	MDG Customer Connection 2023
3428179	Master Data Governance: Continuous Influence
3134600	MDG-M: Supported fields in Data Model MM
1806108	Functional restrictions in MDG-M in MDG7 (incl. SP02)
2129261	Functional restrictions in MDG-M in MDG8
2284745	Functional Restrictions in MDG for Material with SAP Master Data Governance 9.0
2461516	Functional Restrictions in MDG for Material with SAP Master Data Governance 9.1
2656693	Functional Restrictions in MDG for Material in SAP Master Data Governance 9.2 and on SAP S/4HANA 1809
2816571	Functional Restrictions in MDG for Material on SAP S/4HANA 1909

2948873	Functional Restrictions in MDG for Material on SAP S/4HANA 2020
3070012	Functional Restrictions in MDG for Material on SAP S/4HANA 2021
3219945	Functional Restrictions in MDG for Material on SAP S/4HANA 2022
3374998	Functional Restrictions in MDG for Material on SAP S/4HANA 2023
2479869	Usage of Lean Classification with SAP Master Data Governance
1619534	How to Create, Enhance and Adapt FPM Applications
1637249	MDG: Information for efficient message processing
2105467	MDG Performance
2561461	Scope of support for SAP Master Data Governance (MDG)
1637249	MDG: Information for efficient message processing