



PUBLIC

How-To: Create a User Interface for Multiple-Record Processing for Material

Applicable Releases:

From MDG 7.0 and from SAP S/4HANA 1511

Version 6.0

April 2024

Document History

Document Version	Description
1.0	First official release of this guide
1.1	Update of chapter 3.8 with color table
1.2	New chapter 3.6.8 and 3.6.9
1.3	Exchange generic genIL model MDGMC to MDGMCM
1.4	New screenshots in chapter 3.2.1
2.0	MDG8 and SAP S/4HANA 1610 new features: Add, Copy, and Discard, <i>CL_USMD_MC_FEEDER_FORM</i>
3.0	Small updates
4.0	Lean Classification for Material, Customer, Supplier
4.1	Additional information in chapter 4.4.5. Configure the Add and Copy Popup
4.2	New screenshot in chapter 4.4.2.3
5.0	New chapter 5.3 Create Feeder Class with Performance Hints (April 2020)
6.0	Update Layout (April 2024)

1	BUSINESS SCENARIO.....	5
2	BACKGROUND INFORMATION	5
3	FUNCTIONAL RESTRICTIONS.....	7
4	STEP BY STEP EXPLANATION	8
4.1	Create and Configure WD Application Configuration.....	8
4.2	Create a Model-Specific Assistance Class	9
4.3	Create Feeder Classes for List and Form UIBBs	10
4.3.1	Transient Fields	10
4.3.2	Performance	11
4.4	Create a Configuration for the Application Controller.....	12
4.5	Create and Configure WD Component Configurations	12
4.5.1	Create and Configure OVP Component Configuration	12
4.5.2	Create and Configure ATS List Configuration	14
4.5.3	Create and Configure Forms.....	18
4.5.4	Configure the Find and Replace Popup	19
4.5.5	Configure the Add and Copy Popup.....	19
4.5.6	Define the Wire Schema.....	20
4.6	Assign Search Help to Fields	23
4.7	Integrate Change Request UIBB	23
4.8	Define Button in Global Toolbar to Navigate to WD Application USMD_CREQUEST_PROCESS	24
4.9	Adjust Customizing	25
4.9.1	Define Change Request Types Z*	25
4.9.2	Define Logical Actions.....	26
4.9.3	Define Available UI Applications	26
4.9.4	Define Business Activities	27
4.9.5	Link Logical Actions with Business Activities	27
4.9.6	Define Change Request Steps for Rule-Based Workflow	28
4.9.7	Configure Properties of Change Request Step (UI and Highlight Changes)	28
4.9.8	Adapt Limit of Multiple-Record Processing	29
4.10	Activate Service	30
4.11	Highlighting Changed Table Content	30
4.11.1	Color Schema.....	30
4.11.2	Support of Color Schema	31
4.11.3	Selecting the Color Schema	31
4.11.4	Setting Colors	32
4.11.5	Colors	32

4.12 Transfer Object Keys from Source Application to Multiple-Record Processing WD Application 35

4.12.1 Transfer by Change Request 35

4.12.2 Transfer by Global Context 35

4.12.3 Transfer by URL 37

4.13 Work with Copies of Supplied Data (for example from material search UI MDG_BS_MAT_SEARCH_06) 37

5 ADDITIONAL INFORMATION38

5.1 Further Reading 38

5.1.1 Information on SAP MDG on SAP S/4HANA 38

5.1.2 SAP Roadmap Explorer 38

5.1.3 Related Information 38

5.2 SAP Notes..... 38

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.

MDG offers change request (CR)-based processing of master data with integrated workflow, staging, approval, activation, and distribution.

From MDG 7.0 and SAP S/4HANA 1511 SAP also supports the processing of multiple materials in a tabular user interface. Multiple-Record Processing offers a streamlined process, with a UI that enables you to create change requests for multiple records with greater efficiency.

SAP offers a range of sample user interface (UI) configurations to cater to various scenarios. We provide a total of nine sample UI configurations to support five specific scenarios. For four of these scenarios, we offer two different UI configurations. The scenarios include Planning Data, Purchasing Data, Sales Data, Dimension Date, and Classification Data. You can utilize these pre-designed UI configurations as templates to create new tabular user interfaces based on your needs.

Please note that the provided UI configurations and change request (CR) types are just examples and can be used as a starting point. It is important to take into account your own business requirements while developing your own UI configurations and CR types.

This How-To Guide describes the solution to create new UI configurations for material.

2 Background Information

You can use the new FPM (Floor Plan Manager) list feeder class, `CL_USMD_MC_FEEDER_LIST` and `CL_USMD_MC_FEEDER_FORM`, to implement a list displaying data sets for a wide range of data models.

In addition to the standard functionality of an ATS list (ABAP Table Services), the following functionality is offered by the list feeder class feeder `CL_USMD_MC_FEEDER_LIST`:

1. You can mark all data sets having a predefined value for a given attribute (*Select Rows with Identical Values*)
2. You can copy the value of a given attribute to all marked data sets (*Copy Value to Selected Rows*)
3. You can replace any number of attribute values for all marked data sets (*Find and Replace*)
4. You can remove Type-1 entities (with all dependent entities)– even if the type-1 entity has already been assigned to the object list of a change request
5. You can copy a selected data set to a new data set. Dependent on the list configuration, related entities are also copied. If a value will be copied depends also on customizing of the backend material master (transaction OMSR).
6. You can add a new data set to the list. Dependent on the list configuration, related entities are also added
7. All changes in respect to the active data can be discarded for all selected data sets. Changes of the dependent data sets are also discarded
8. All data related to the supplied type-1 entity keys can be copied at startup (replication). In this case, the application works on the copied data. If a value will be copied depends also on customizing of the backend material master (transaction OMSR).
9. For all new data sets, the type-1 entity key can be altered

10. Active data for type-4 entities and text entities can be deleted
11. The deletion of entities can be discarded
12. For USMD models MM and BP, classification data can be maintained in the context of lean classification

The first and second function can only be reached using context menu, while the third function can either be triggered using the context menu or a related toolbar button. To remove (4), copy (5), add (6), or discard (7) entities, the related toolbar button must be pressed. To create new entities for all supplied type-1 entity keys, the application parameter *USMD_NEW_MASS_COPY* must set to X. Replication takes place at startup of the application.

The FPM form feeder class *CL_USMD_MC_FEEDER_LIST* does not provide special functionality to manipulate the data.

Table cells can be highlighted using different colors for the following cases:

- The saved cell value is different from the active value
- The current cell value is different from the saved value

Table cells and form fields can be highlighted using different colors. Highlighting indicates that the current value is different from the saved or active value, respectively. Table cells that are highlighted offer a tooltip describing the changes. In addition, a change indicator column can be configured for each list. Here an icon is indicating any change in respect to the active data. If required, you can switch on and off the table cell highlighting at runtime.

Changes	Material	Material type	Industry sector	Base Unit of Measure	Material Group	X-plant mat status	Valid from
49	HALB	M	EA	01	01	16.12.2014	
7454	HALB	M	EA	02		16.12.2014	
52	KMAT	C	KG	01	01		
60	FERT	C	EA	01		11.12.2014	
69	FERT	M	KG	02			
70	HALB	M	EA	01			
74	FGTR	C	EA				

Material list based on the generic mass change list feeder

The functions to save, check, and activate the changed data are provided by the MDG change request UIBB (UI Building Block) that must be embedded in the same page as the multiple-record processing list.

3 Functional Restrictions

The following restrictions apply to a WD (Web Dynpro) application based on the new feeder class:

- To maintain type-4 entity data, you need to configure UIBBs for all entities being part of the relationship chain between the root type-1 entity and the type-4 entity. All UIBBs need to be embedded into the main page of the application and they need to be connected using wires.
Example: you want to maintain plants. This requires that you also configure a UIBB displaying materials, since the type-4 entity related to plants depends on the root type-1 entity (MATERIAL).
- Data displayed by forms and lists can be constructed by joining entities. Joined entities are only displayed if the root entity exists.
Example: A material has assigned one plant. Only the MARCBASIC entity exists for this plant. A table displaying plant data, but having not MARCBASIC as its root entity will not display the material's plant.
- Objects for dependent entities are retrieved generically. The retrieval only depends on the objects displayed for the parent entity. Filter information supplied by the calling application is not considered.
Example: The calling application sends a filter value for the plant specific material status. However, for the retrieval of the type-1 main entity – which is MATERIAL – this information is not relevant. On the other side, this information is also not considered when plants are retrieved, since this process only depends on the selected material.
- To connect two UIBBs using wires, the root entities displayed by the two UIBBs must have a 1:1 or 1:cn relationship. This means: the key of the parent UIBB must be part of the key of the child UIBB.
- Only the main type-1 entity type can be handled by the multiple-record processing. For example, Production Versions cannot be maintained together with materials.
- Non-lean classification is not supported.
- “Lean classification” is not supported for USMD models different from MM and BP.
- If you copy materials from search, non-lean classification data will not be copied. Lean classification data will be copied for USMD models MM and BP.

4 Step by Step Explanation

The following explanation shows you how to configure the UI for multiple-record processing. You need to define a new WD application configuration, which is assigned to the change request type and step representing the maintenance scenario. The WD application configuration can be assigned to the WD application MDG_BS_MAT_MC.

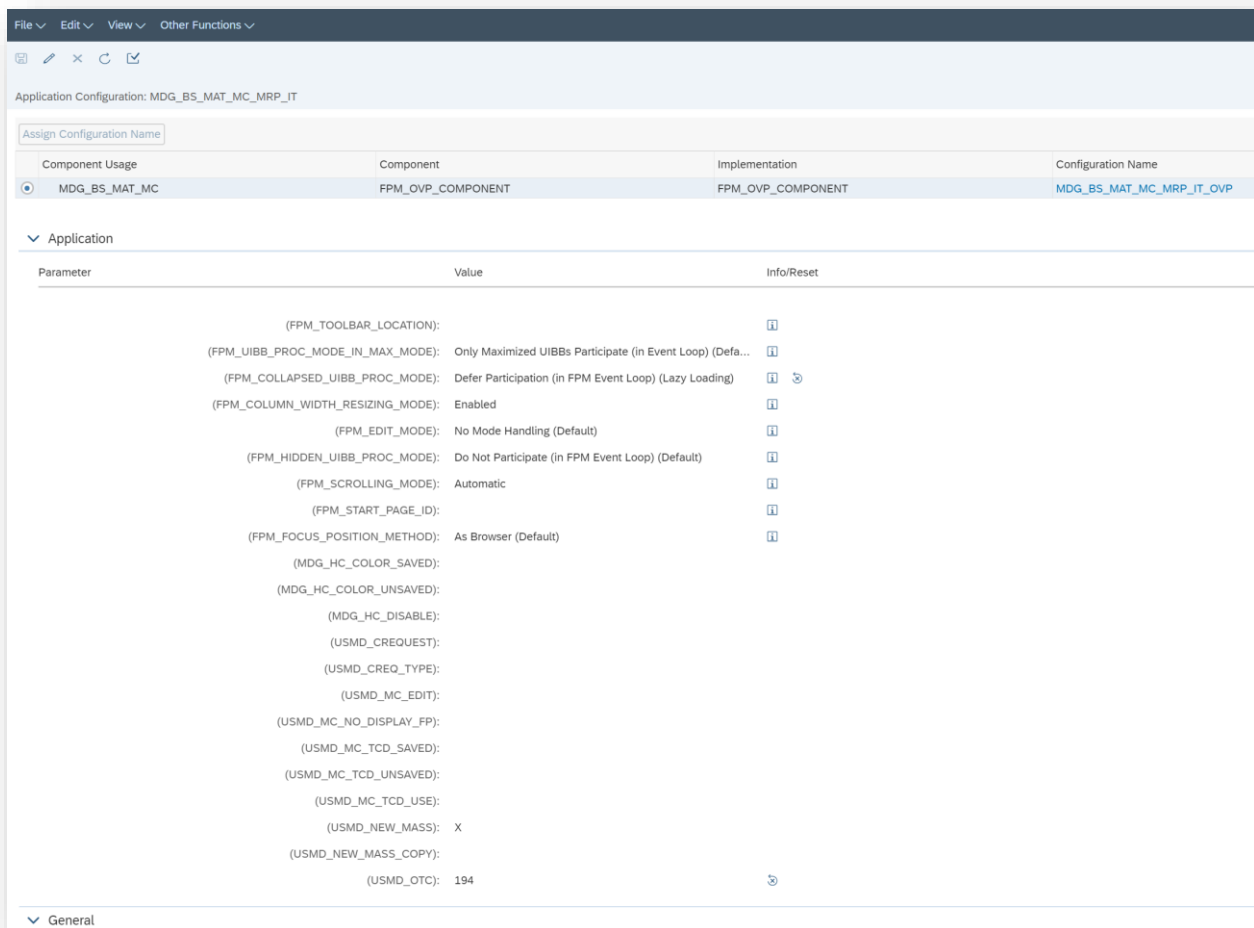
The implementation of a multiple record processing application based on the list feeder class *CL_USMD_MC_FEEDER_LIST* and *CL_USMD_MC_FEEDER_FORM* consists of the following steps:

- Create and configure a WD application configuration for WD application MDG_BS_MAT_MC
- Create a model-specific assistance class inheriting from *CL_USMD_MC_GEN_ASSIST*
- Create a configuration for the application controller
- Create and configure the floorplan configuration
- Create and configure the related WD list and form component configurations
- Integrate the change request UIBB
- Implement the methods of your model specific assistance class
- Enhance the source application (For example, search) to transfer the entity keys to the WD multiple-record processing application
- Define customizing (CR types, navigation)
- Adjust the Global Context definition (optional)

4.1 Create and Configure WD Application Configuration

Create a WD application configuration for WD application MDG_BS_MAT_MC. Configure the WD application configuration as follows:

- Assign the name of the OVP component configuration to the WD component *FPM_OVP_COMPONENT*
- Configure the application parameter *USMD_OTC*: Assign the business object type 194 of your data model MM to this parameter
- Do not configure the application parameter *USMD_CREQ_TYPE*: This parameter is used to transfer the type of an existing change request to the multiple-record processing application
- Do not configure the application parameter *USMD_CREQUEST*: This parameter can be used to transfer the number of an existing change request to the multiple-record processing application
- Do not configure the application parameter *USMD_NEW_MASS_COPY*: This parameter can be used to force the copying of the type-1 entities defined by the supplied object keys (replication). The application is then working with the copied entities.
- You can configure the parameters for highlighting changes here. If you have added the application parameters *MDG_HC_COLOR_SAVED* (to highlight saved inactive data different from active data), *MDG_HC_COLOR_UNSAVED* (to highlight current data different from saved data), or *MDG_HC_DISABLE* in the last step, you can configure the parameter values here. If you have added the application parameters *USMD_MC_TCD_SAVED* (to highlight saved inactive data different from active data), *USMD_MC_TCD_UNSAVED* (to highlight current data different from saved data), or *USMD_MC_TCD_USE* (highlighting changes initially switched on/off) in the last step, you can configure the parameter values here. See chapter 4.11 Highlighting Changed Table Content for more details.
- You can configure the parameters for Lazy Loading here. Set the parameter "FPM_COLLAPSED_UIBB_PROC_MODE" to "Defer Participation (in FPM Event Loop) (Lazy Loading)". See also SAP Note [2885149](#). Lazy Loading brings a significant runtime improvement for UI Building Blocks which are collapsed when you access the UI.



Create / maintain WD application configuration (example for Material Planning Data (indep. Table)).

4.2 Create a Model-Specific Assistance Class

Create a new class inheriting from super class `CL_USMD_MC_GEN_ASSIST`. This class allows you to define model-specific code for the following situation:

- add new entities
- copy existing entities
- discard entity changes
- delete existing entities
- copy existing entities before entering the multiple-record application
- retrieve descriptions for entity attributes
- implement OVS help for add/copy dialog box
- adapt highlight changes information
- clear /reset data of root entity (implemented for characteristic values)
- add / remove row (implemented for multi-value characteristics)

For data model MM, an implementation already exists (`CL_MDG_BS_MAT_MC_ASSIST`). For data model BP, a corresponding template class is `CL_BS_CUSP_MRP_GEN_ASSIST`.

4.3 Create Feeder Classes for List and Form UIBBs

Although this step is optional, it is essential if you want or need to perform customer-specific enhancements, especially if you want to perform the tasks mentioned in 5.3.1 and 5.3.2 below.

Create a new class inheriting from super class `CL_USMD_MC_FEEDER_LIST` for List UIBBs and a new class inheriting from super class `CL_USMD_MC_FEEDER_FORM` for Form UIBBs.

For more hints and tricks related to this, please refer to the following two documents:

- <https://archive.sap.com/documents/docs/DOC-28799> - A cookbook explaining the basics of adapting Floorplan Manager applications
- <https://www.sap.com/documents/2015/07/960eb94c-5b7c-0010-82c7-eda71af511fa.html> - A guide explaining how to extend the MDG-M user interface, especially sections 4.2.1 on feeder enhancement and 5.4. on creating new Form UIBBs.

4.3.1 Transient Fields

You can also create additional fields that are not in the data model. These Transient Fields can then be used to show data that is derived from other fields. For example, the data model contains a field for a timestamp but you want to display the date and time. Also, you can use these fields to display the descriptions of entity attributes.

Example of redefined method `CREATE_STRUCT_RTTI` in your feeder class:

```
METHOD create_struct_rtti.  
  
DATA lt_component TYPE cl_abap_structdescr=>component_table.  
FIELD-SYMBOLS <ls_component> LIKE LINE OF lt_component.  
  
super->create_struct_rtti( ).  
    if ms_object_key-object_name = 'MARCBASIC'.  
  
        APPEND INITIAL LINE TO lt_component ASSIGNING <ls_component>.  
        <ls_component>-type = me->mo_struct_rtti.  
        <ls_component>-as_include = abap_true.  
  
        APPEND INITIAL LINE TO lt_component ASSIGNING <ls_component>.  
        <ls_component>-name = 'PROFITCENTERDESC'.  
        <ls_component>-type = cl_abap_elemdescr=>get_string( ).  
  
        me->mo_struct_rtti = cl_abap_structdescr=>create( lt_component ).  
  
    endif.  
  
ENDMETHOD.
```

To provide the data, you have to redefine method `GET_ENTITY_DATA` as follows:

```
method GET_ENTITY_DATA.  
  
FIELD-SYMBOLS:
```

```

<lv_string> TYPE any,
<PRCTR>      type PRCTR.

data wa_prctr_text_buffer type ty_prctr_text_buffer.

super->get_entity_data(
  EXPORTING io_access = io_access
  CHANGING cs_data = cs_data ).

CHECK io_access IS BOUND.

* If you use the feeder class for multiple UIBBs, make sure your code
runs only * when needed. Otherwise delete the if / endif
if ME->MS_INSTANCE_KEY-CONFIG_ID = '<name of the UIBB
configuration>'.

  ASSIGN COMPONENT 'TEST_SAP' OF STRUCTURE cs_data TO <lv_string>.
  ASSIGN COMPONENT 'PRCMARCBA' OF STRUCTURE cs_data TO <PRCTR>.

  read table mt_prctr_text_buffer INTO wa_prctr_text_buffer WITH KEY
    prctr = <PRCTR>.

  if sy-subrc = 0.
    <lv_string> = wa_prctr_text_buffer-ltext.
  else.
    clear wa_prctr_text_buffer.
    select single ltext from CEPCT into wa_prctr_text_buffer-
ltext WHERE
      spras = sy-langu and
      kokrs = 'DE01' and
      PRCTR = <PRCTR> .

    if sy-subrc = 0.
      <lv_string> = wa_prctr_text_buffer-ltext.
      wa_prctr_text_buffer-PRCTR = <PRCTR> .
      insert wa_prctr_text_buffer into TABLE mt_prctr_text_buffer.
    endif.
  endif.
endif.

endmethod.

```

In this example we use type `TY_PRCTR_TEXT_BUFFER` which is a private type of the feeder class. Buffer `MT_PRCTR_TEXT_BUFFER` is a private instance attribute of the feeder class.

4.3.2 Performance

To improve the performance of your application, we recommend you only execute the `GET_DATA` method of your feeder class if needed. If you know that you do not change data in the back end while processing a given event, you can skip `GET_DATA`. This might especially be the case if the roundtrip was triggered by an event of another UIBB.

Example Code:

```
METHOD if_fpm_guiabb_list~get_data.

CASE iv_eventid->mv_event_id.
...
    when 'FPM_TRAY_TOGGLE'.
        if iv_raised_by_own_ui = abap_false.
            return.
        endif.
    ENDCASE.

super->if_fpm_guiabb_list~get_data(
    EXPORTING
        iv_eventid                = iv_eventid
        it_selected_fields        = it_selected_fields
    ...
)
```

4.4 Create a Configuration for the Application Controller

Create a component configuration for the WD component *USMD_MC_GEN_APP_CTRL*, which will serve as the application controller of your application. Set the component defined attributes as follows:

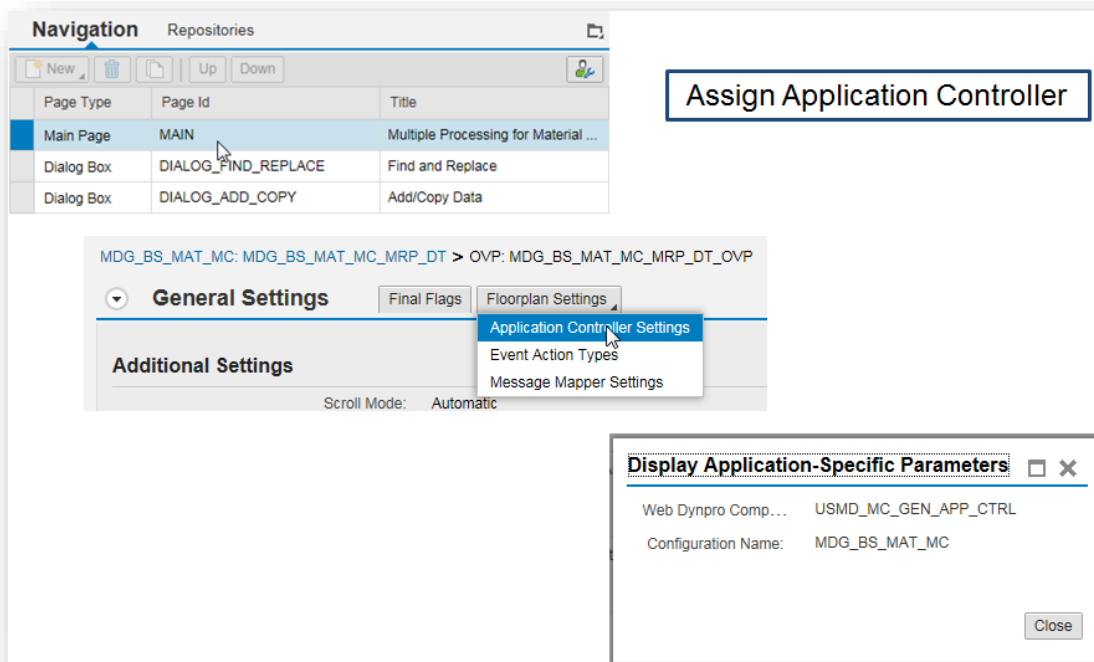
Name = <Name of your model-specific assistance class>

4.5 Create and Configure WD Component Configurations

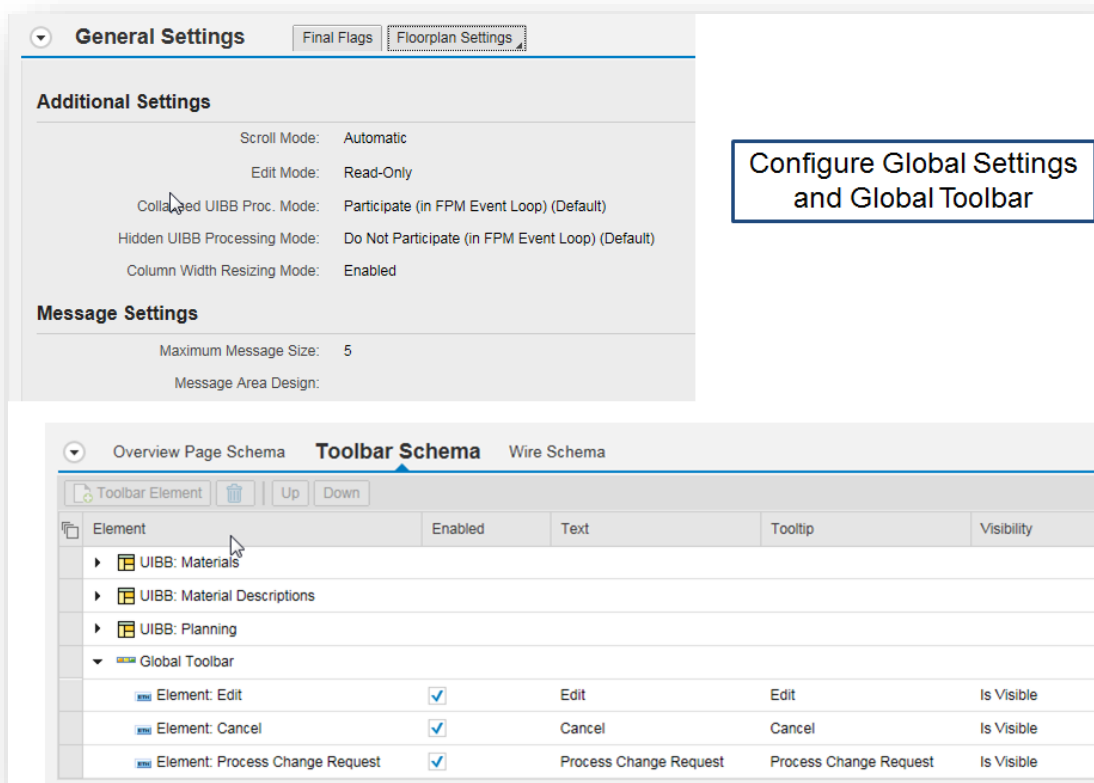
4.5.1 Create and Configure OVP Component Configuration

For each scenario you need to configure a new WD component configuration for the WD component *FPM_OVP_COMPONENT*.

- Make sure that the page ID is *MAIN* and use the WD component *USMD_MC_GEN_APP_CTRL* as Application Configuration Controller for your WD component. From the General Setting panel select *Floorplan Settings* → *Application Controller Settings*.



- Define the global toolbar schema. Define a button for each of the following FPM events: *FPM_EDIT* and *FPM_CANCEL*. You also have to maintain the properties: text, tooltip, and image.



- Optionally: Define a toggle button in the global toolbar for the FPM event *MDG_HIGHLIGHT_TC*. This allows you to switch on/off table cell highlighting at runtime

- In *General Settings* section set *Edit Mode* to *Read-Only* and *Maximum Message Size* to 5.

Main Entity (Type-1 Entity)

Embed the ATS list UIBB (FPM_LIST_UIBB_ATS) and assign a component configuration (to be created) to this UIBB.

Element	Component	Config ID	Window Name
Section: SECTION_1			
UIBB: Materials	FPM_LIST_UIBB_ATS	MDG_BS_MAT_MC_MATERIAL_LIST	LIST_WINDOW

Type-4 Entities

For each dependent entity type in the data model hierarchy, embed another ATS list UIBB (FPM_LIST_UIBB_ATS) or a form UIBB (FPM_FORM_UIBB_GL2) to the floorplan component. To each UIBB assign a component configuration.

Element	Component	Config ID	Window Name
Section: MAIN			
UIBB: Materials	FPM_LIST_UIBB_ATS	MDG_BS_MAT_MC_MRP_MATERIAL	LIST_WINDOW
UIBB: Material Descriptions	FPM_LIST_UIBB_ATS	MDG_BS_MAT_MC_MATERIAL_TXT	LIST_WINDOW
UIBB: Planning	FPM_LIST_UIBB_ATS	MDG_BS_MAT_MC_MRP_MARC	LIST_WINDOW

Create / maintain configuration for OVP component (example for changing material data)

4.5.2 Create and Configure ATS List Configuration

Define Feeder Class Parameters

The following procedure has to be conducted one or more times: Once for the header UIBB, displaying the root type-1 entity data, and once for each UIBB displaying a type-4 entity

- Assign the feeder class *CL_USMD_MC_FEEDER_LIST* to the ATS list component configuration
- Set the feeder class parameters:
 - Component* = MDGMCM (Name of GenIL component)
 - Object Name* = <Entity of Type-1 / Type-4>
 - Editable* = checked

The genIL model name depends on the application you use. For MDG model MM, the model MDGMCM is to be used.
- Optional: Add entities in the “Join Structure” table. This allows you to display not only the root entity, but additional entities having a 1:1 relation to the root entity
- When you close the dialog box choosing the ok button the system automatically configures ATS list

Edit Parameters

Feeder Class

Feeder Class: CL_USMD_MC_FEEDER_LIST

Generic List for Mass Changes

Parameters

* Component: MDGMCM

* Object Name: MATERIAL

Editable: ☒

Join Structure

Join Structure	*Relation Name	*Component	*Assigned Obj.	Suffix	Dark	Port Ide...
MATERIAL2MARASALESRel	MATERIAL2MARASALESRel	MDGMCM	MARASALES	_SA	<input type="checkbox"/>	

Close

Set feeder class parameters for ATS list UIBB configuration (example for changing material data)

Note:

Each entity type you add will have a direct influence on the application performance. If the root entity exists, the joined entity and its properties also have to be retrieved.

On the other side, if you want to offer the possibility to add new entities (Add/Copy), you may need to join entities you have not in mind. Customizing may set information as mandatory that is defined in entities other than the root entity.

Example: You add a new plant to a material. The derived maintenance status requires that some fields in other entities get mandatory fields. These entities need to be configured, too.

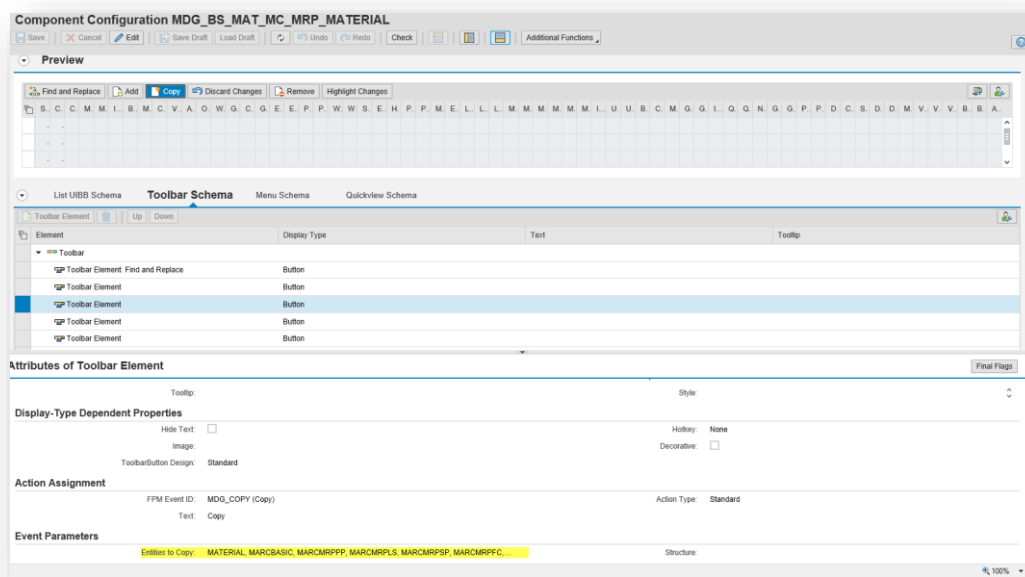
Define List UIBB Schema

Configure the columns you want to offer to the user of the application. Make sure that the entity key fields (common to root and joined entities) are only configured once.

- Delete the table column FRW_ROW_ACTIONS_COLUMN.
For attributes containing translatable texts choose a non-editable field type (For example: Text View). For all other attributes, choose editable field types.
- For each column you would like to offer for administration or personalization, but which should not be displayed at startup, set the visibility to not visible.
- If you want to indicate via an icon that data has changed (meaningful, if the changed attributes are not displayed), you can add a related column to your list configuration. Three attributes may be configured:
 - Attribute *STAGING* is displayed if the old color scheme is selected
 - Attribute *CHANGES* is displayed if the new color scheme is selected, but the business function *MDG_APPLICATION_FRAMEWORK_5* is switched off
 - Attribute *USMD_CHANGE_INDICATOR* is displayed if the new color scheme is selected and if the business function *MDG_APPLICATION_FRAMEWORK_5* is switched on
 - You may configure columns for all of these attributes. At runtime only one of these columns will be displayed

Define the Toolbar Schema

- Remove all buttons that have been created automatically
- Configure a button for each of the FPM events *MDG_FIND_REPLACE*, *MDG_ADD*, *MDG_COPY*, and *MDG_DISCARD*, or *MDG_DELETE* (if you want to offer the related functionality). For the ATS list displaying the type-1 main entity, you can also configure a button for the FPM event *MDG_REMOVE* (button is programmatically hidden for all other lists)
- Define the attributes of the toolbar elements:
 - **ADD button:**
Set the event parameters *Entities to Add* to define, which entities are created when a new row is added to the list. If you leave this field empty, only the root entity is created (default). If you want to create additional entity types, you can list these entity types separated by commas. However, only entities having the same key as the root entity will be created. For lists displaying a text entity, only the text entity will be created. This behavior may be adjusted by the model-specific application class you created before.
Set the event parameter *Structure* to define the structure of the list displayed on the dialog box for adding new rows to the list. If you leave this field empty, the structure is defined by the key fields of the root entity (default). If you want to override the default, you can enter the name of a DDIC structure. This allows you to offer sophisticated DDIC search helps without having to code a line.
 - **COPY button:**
Set the event parameters *Entities to Copy* to define, which entities are created when an existing row is copied. If you leave this field empty, all entities having the same key as the root entity of the selected row will be copied (default). If you want to copy only specified entity types, you can list these entity types separated by commas.



However, only entities including the key of the selected root entity will be copied. For lists displaying a text entity, only the text entity of the selected row will be copied. This behavior may be adjusted by the model-specific application class you created before.

The usage of the event parameter *Structure* is described above.

If a value will be copied depends also on customizing of the backend material master (transaction OMSR).

- **DISCARD button:**
Set the event parameters *Entities to Discard* to define, which entity changes are discarded

for the selected rows. If you leave this field empty, the changes of all entities having the same key as the root entity are discarded (default). If you want to override the default, you can list the entities to be discarded separated by commas. However, only entities including the keys of the selected root entities will be discarded. For lists displaying a text entity, only the text entities for the selected rows will be discarded. This behavior may be adjusted by the model-specific application class you created before.

Set the event parameter *Excl. From Discard* to define which entities are to be excluded from the discard procedure. If you leave this field empty, all entities determined as described above will be discarded.

- **DELETE button:**

Set the event parameters *Entities to Delete* to define, which entities are deleted for the selected rows. You need to list the entity types to be deleted separated by commas (no default). However, only entities including the keys of the selected root entities will be deleted. The model-specific application class you created before allows you to exclude single entities from deletion.

If you configure *Delete* and *Discard* button, both buttons should work on the same entities.

- **CLEAR button:**

All non-key attributes of the selected root entities are initialized. The model-specific application class allows you to define which of the selected entities should be excluded from initializing.

- **RESET button:**

All non-key attributes of the selected root entities are initialized. The model-specific application class allows you to define which of the selected entities should be excluded from initializing. In addition, after initializing, the model-specific application class allows you to manipulate single attributes again before writing them to the staging area.

The template implementations for MM and BP handle CLEAR and RESET for characteristic values as follows:

- CLEAR initializes all non-key fields. The resulting valuation will be dropped when activating the change request
- RESET initializes all non-key fields but ATCOD. This means that the initial value is stored as a characteristic value. Conversion from external to internal value takes place. This value will not be dropped when activating the change request

- **ADD ROW button:**

There is no default handling for this button. However, the template implementations for MM and BP handle *ADD ROW* for characteristic values as follows:

- For each selected multi-value characteristic, a new row is added. This allows you to enter additional characteristic values.

- **REMOVE ROW button:**

There is no default handling for this button. However, the template implementations for MM and BP handle *ADD ROW* for characteristic values as follows:

- Each selected row of a multi-value characteristic is removed. This allows you to remove multiple values of multi-value characteristics at once. However, the last value can only be removed by clicking the *CLEAR* button

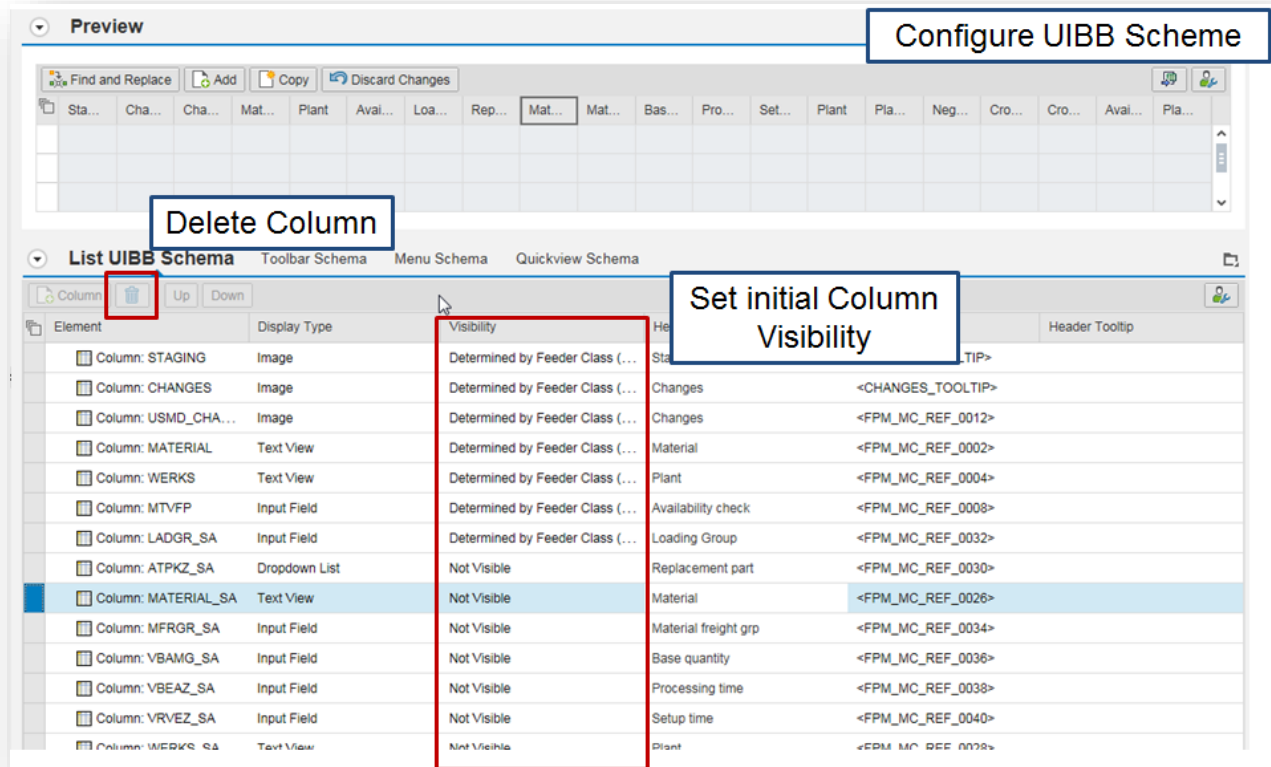
Define General Settings

In General Settings section, set the following attributes:

Selection Mode = Multiple Selection

Enable Event on All Selections = Selection Raises an FPM Event

Set other General Settings attributes as desired.



Configure ATS list (example for changing material data)

Note: To reduce the number of roundtrips, the property *Enable Event on All Selections* can be set to *Selection Raises No FPM Event*. If the UIBB is a wire source and the wire port type is set to *Lead Selection*, then the selection of additional rows will not trigger an FPM event.

4.5.3 Create and Configure Forms

Define Feeder Class Parameters

The following procedure must be conducted for each form UIBB.

- Assign the feeder class *CL_USMD_MC_FEEDER_FORM* to the form component configuration
- Set the feeder class parameters:
Component = <genIL> model
Object Name = <Entity of Type-1 / Type-4>
Editable = checked
 The genIL model name depends on the application you use. For MDG model MM, the model MDGMCM is to be used.
- Optional: Add entities in "Join Structure" table. This allows you to display not only the root entity, but additional entities having a 1:1 relation to the root entity.
- Finish dialog → form is configured automatically

Define Form UIBB Schema

Configure the fields you want to offer to the user of the application. Make sure that the entity key fields (common to root and joined entities) are only configured once.

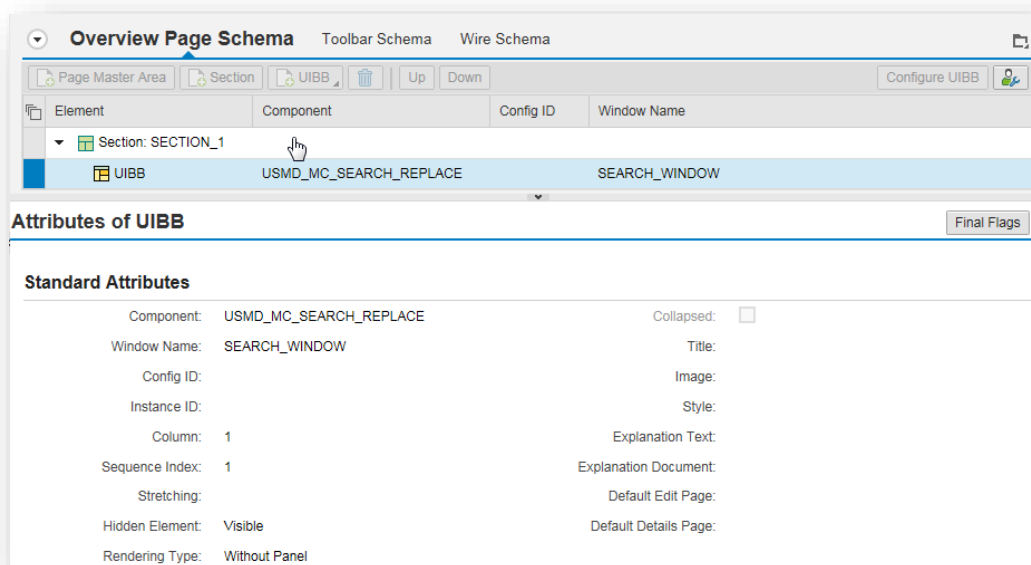
- For attributes containing translatable texts choose a non-editable field type (e.g., *Text View*). For all other attributes, choose editable field types.

For each column you would like to offer for administration or personalization, but which should not be displayed at startup, set the visibility to *not visible*.

4.5.4 Configure the Find and Replace Popup

Navigate back to the OVP component configuration.

- On the *Navigation* tab, create a new page of type *Dialog Box*
- Give it the page ID *DIALOG_FIND_REPLACE*. It's important to choose exactly this name because it is hard coded.
- Set the page attributes for the popup:
Choose OK and *CANCEL* buttons.
Choose *Find and Replace* as dialog name.
Use the text *Find and Replace* for the OK button as text and tooltip.
- Embed the following UIBB on the existing section:
Component: *USMD_MC_SEARCH_REPLACE*
Window Name: *SEARCH_WINDOW*



Configure *Find and Replace* dialog box

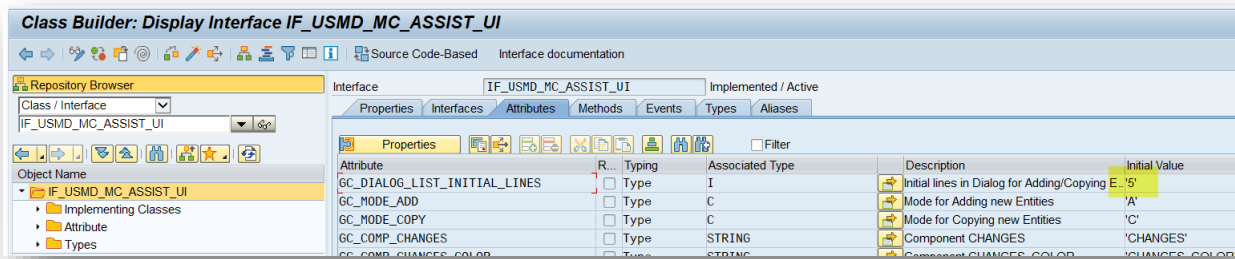
4.5.5 Configure the Add and Copy Popup

Edit the OVP component configuration.

- On the *Navigation* tab, create a new page of type *Dialog Box*
- Give it the page ID *DIALOG_ADD_COPY*. It's important to choose exactly this name

- Display *OK* and *CANCEL* buttons
- Embed the following UIBB on the already existing section:
Component: *USMD_MC_ADD_COPY*
Window Name: *MAIN_WINDOW*

Additional information for customer adaptation: Overwrite initial value for GC_DIALOG_LIST_INITIAL_LINES.



In the Add and Copy Popup you always get 5 empty lines for insert material numbers. After ENTER the system fill additional lines that you have again 5 empty lines. If you want to have a different number, you can change this with a post exit.

Solution:

Create Post-Exit for Component Controller method WDDOINIT of WD Component USMD_MC_ADD_COPY and insert following line of code, where <n> is the number of empty lines:

```
wd_this->mv_no_initial_lines = <n>
```

4.5.6 Define the Wire Schema

List Displaying Main Entity

- Add a wire.
- Use the ATS list component displaying type-1 entity as the target.
- Set the connector class to *CL_USMD_MC_CONNECTOR_QUERY_URL*.
- Use the value help to set the name of the dynamic query.
- Use the value help to set the GenIL component name to *MDGMCM* for MDG model *MM*.
- Set the transaction handler class to *CL_MDG_BS_BOL_TRANSACTION_MM*.

Transaction: Transaction handler (CL_USMD_MC_BOL_TRANSACTION_MM)

Element	Component	Config ID	Port Type	Source Component	Source Config Name
Wire: List MDG_BS_MAT_...	FPM_LIST_UIBB_ATS	MDG_BS_MAT_MC_MRP_MARC	Selection	FPM_LIST_UIBB_ATS	MDG_BS_MAT_MC_MRP_MATERIAL
Wire: List MDG_BS_MAT_...	FPM_LIST_UIBB_ATS	MDG_BS_MAT_MC_MRP_MATERIAL			
Wire: List MDG_BS_MAT_...	FPM_LIST_UIBB_ATS	MDG_BS_MAT_MC_MATERIAL_TXT	Selection	FPM_LIST_UIBB_ATS	MDG_BS_MAT_MC_MRP_MATERIAL

Attributes of Wire: List MDG_BS_MAT_MC_MRP_MATERIAL

Standard Attributes

* Component:	FPM_LIST_UIBB_ATS	Src Inst. ID:	
Config ID:	MDG_BS_MAT_MC_MRP_MATERIAL	Port Type:	
Instance ID:		Port Identifier:	
Source Component:		* Connector Class:	CL_USMD_MC_CONNECTOR_QUERY_URL
Source Config Name:			

Connector Parameters

Component Name:	MDGMCM	(Dynamic) Query Name:	DynamicQueryMATERIAL
Skip initial screen if result is unique:	Initial Screen will be skipped		

Define wire schema (example for changing material data)

UIBBs Displaying Other Entities

- Add a new wire for each UIBB
- Select source and target UIBB, respectively. The target UIBB needs to display entities having a key that incorporates the key of the parent UIBB.
- Set connector class to
 - *CL_MDG_BS_CONNECTOR_BOL_REL*, if the root entities of source and target UIBBs are not identical
 - *CL_FPM_CONNECTOR_BOL_IDENTITY*, if the root entities of source and target UIBBs are identical
- Set port identifier to *STANDARD*
- Set port type to
 - *Collection* to display entity data for all data sets displayed in the parent list UIBB
 - *Selection* to display entity data for multiple data sets selected in the parent list UIBB
 - *Lead Selection* to display entity data only for the parent UIBB data set at lead selection
- If you selected the connector *CL_MDG_BS_CONNECTOR_BOL_REL*, you may need to select the *Relation Name* (use drop down menu) to relate parent entity (name <TP>) and target entity (name <TT>)
 - non-text target entity: select the item <TP>2<TT>Rel
 - text target entity: select the item <TP>2DTxt<TT>Rel
 - For parent entity name <TP> and target entity name <TT>, select the item <TP>2<TT>Rel
 - **Note:** If there is no relation name provided in the drop-down list, there is no direct relation between the target entity and the parent entity. In this case, it is necessary to add further UIBB's to complete the dependency queue

Overview Page Schema Toolbar Schema **Wire Schema**

Transaction: Transaction handler (CL_USMD_MC_BOL_TRANSACTION_MM)

Wire Graphical Wire Editor

Element	Component	Config ID	Port Type	Source Component	Source Config Name
Wire: List MDG_BS_MAT_...	FPM_LIST_UIBB_ATS	MDG_BS_MAT_MC_MRP_MARC	Selection	FPM_LIST_UIBB_ATS	MDG_BS_MAT_MC_MRP_MATERIAL
Wire: List MDG_BS_MAT_...	FPM_LIST_UIBB_ATS	MDG_BS_MAT_MC_MRP_MATERIAL			
Wire: List MDG_BS_MAT_...	FPM_LIST_UIBB_ATS	MDG_BS_MAT_MC_MATERIAL_TXT	Selection	FPM_LIST_UIBB_ATS	MDG_BS_MAT_MC_MRP_MATERIAL

Attributes of Wire: List MDG_BS_MAT_MC_MRP_MARC Final Flags

Standard Attributes

* Component:	FPM_LIST_UIBB_ATS	Src Inst. ID:	
Config ID:	MDG_BS_MAT_MC_MRP_MARC	Port Type:	Selection
Instance ID:		Port Identifier:	STANDARD
Source Component:	FPM_LIST_UIBB_ATS	* Connector Class:	CL_MDG_BS_CONNECTOR_BOL_REL
Source Config Name:	MDG_BS_MAT_MC_MRP_MATERIAL		

Connector Parameters

Relation Name:	MATERIAL2MARCBasicRel	Creation Mode:	Creation with Default Values (Using Template if Available)
----------------	-----------------------	----------------	--

Define wire schema for non-text entities (example for changing plant data)

Overview Page Schema Toolbar Schema **Wire Schema**

Transaction: Transaction handler (CL_USMD_MC_BOL_TRANSACTION_MM)

Wire Graphical Wire Editor

Element	Component	Config ID	Port Type	Source Component	Source Config Name
Wire: List MDG_BS_MAT_...	FPM_LIST_UIBB_ATS	MDG_BS_MAT_MC_MRP_MARC	Selection	FPM_LIST_UIBB_ATS	MDG_BS_MAT_MC_MRP_MATERIAL
Wire: List MDG_BS_MAT_...	FPM_LIST_UIBB_ATS	MDG_BS_MAT_MC_MRP_MATERIAL			
Wire: List MDG_BS_MAT_...	FPM_LIST_UIBB_ATS	MDG_BS_MAT_MC_MATERIAL_TXT	Selection	FPM_LIST_UIBB_ATS	MDG_BS_MAT_MC_MRP_MATERIAL

Attributes of Wire: List MDG_BS_MAT_MC_MATERIAL_TXT Final Flags

Standard Attributes

* Component:	FPM_LIST_UIBB_ATS	Src Inst. ID:	
Config ID:	MDG_BS_MAT_MC_MATERIAL_TXT	Port Type:	Selection
Instance ID:		Port Identifier:	STANDARD
Source Component:	FPM_LIST_UIBB_ATS	* Connector Class:	CL_MDG_BS_CONNECTOR_BOL_REL
Source Config Name:	MDG_BS_MAT_MC_MRP_MATERIAL		

Connector Parameters

Relation Name:	MATERIAL2DTxtMATERIALRel	Creation Mode:	Creation with Default Values (Using Template if Available)
----------------	--------------------------	----------------	--

Define wire schema for text entities (example for changing plant data)

Optional: Not all UIBBs need to be visible at runtime. A UIBB may be hidden, if no user interaction is necessary to define which data is to be displayed by the related target UIBB. To hide a parent UIBB, select port type *Collection*. In addition, set *Hidden Element* to *Hidden but processed in Event Loop (Technical)* for the parent UIBB in the *Overview Page Schema* panel

Attributes of UIBB: Materials

Standard Attributes

* Component:	FPM_LIST_UIBB_ATS	
* Window Name:	LIST_WINDOW	
Config ID:	MDG_BS_MAT_MC_MRP_MATERIAL	
Instance ID:		
Column:		1
Sequence Index:		1
Hidden Element:	Visible	▼
Rendering Type:	Visible Hidden and Excluded from Event Loop Hidden but Processed in Event Loop (Technical)	
Collapsed:		

Hide UIBB (example for changing material data)

4.6 Assign Search Help to Fields

A value help is available for a field if:

1. A DDIC search help is assigned to the field in the MDG data model
2. A DDIC search help is assigned to the underlying data element
3. The related domain contains fixed values

Enhancement options:

1. You can enhance this by using a post exit in method CL_USMD_MC_FEEDER_LIST or CL_USMD_MC_FEEDER_FORM → ASSIGN_VALUE_HELP_EXPL.
2. Value help is displayed if a DDIC search help is assigned to the field in an MDG-structure. You may assign a DDIC search help to the generated entity structures using transaction SE11 to assign a complex value help to a field.

Note: you must reassign the DDIC search help each time after you regenerate the data model MM.

Sequence of consideration:

1. DDIC search help is assigned to the field in the MDG data model
2. Enhancement 2
3. Enhancement 1
4. DDIC search help is assigned to the underlying data element
5. The related domain contains fixed values

4.7 Integrate Change Request UIBB

Following component configuration is necessary to enable the 'communicator' to place the CR UIBB automatically in the multiple-record processing UI:

- Create a new component configuration for the WD component *MDG_BS_GOV_COMMUNICATOR*. The name (=ID) of your component configuration of MDG_BS_GOV_COMMUNICATOR must be identical to the name (=ID) of your multiple-record processing application configuration from above.
- Maintain the component-defined properties of the component configuration. In the *Configuration Context* group, choose the *New* button and select the menu item *crWires* to create a sub element of the *Settings* node.

Select the element *crWires (MAIN)* in the *Configuration Context*. Set the attributes of this element as follows:

<i>Page ID:</i>	<i>MAIN</i>
<i>Connector:</i>	<i>CL_MDG_BS_CONNECTOR_BOL_CR_REL</i>
<i>Port Type:</i>	<i>Collection</i>
<i>Port Identifier:</i>	<i>Standard</i>
<i>Source Component:</i>	<i>FPM_LIST_UIBB_ATS</i>
<i>Source Config Name:</i>	<i><Comp Config. of ATS List UIBB displaying Root Type-1 Entity></i>
<i>Src Config Type:</i>	<i>General</i>

Component-Defined

Configuration Context

Elements

- context
 - settings
 - crWires (MAIN)**

Attributes of the Element crWires (MAIN)

* Page Id:	MAIN	<input checked="" type="checkbox"/> Final
connector:	CL_MDG_BS_CONNECTOR_BOL_CR_REL	<input type="checkbox"/> Final
Port Type:	Collection	<input type="checkbox"/> Final
Port Identifier:	STANDARD	<input type="checkbox"/> Final
Source Component:	FPM_LIST_UIBB_ATS	<input type="checkbox"/> Final
Source Config Name:	MDG_BS_MAT_MC_MRP_MATERIAL	<input type="checkbox"/> Final
Src Config Type:	General	<input type="checkbox"/> Final
Src Config Var:		<input type="checkbox"/> Final
Src Inst. ID:		<input type="checkbox"/> Final

Maintain configuration for communicator component

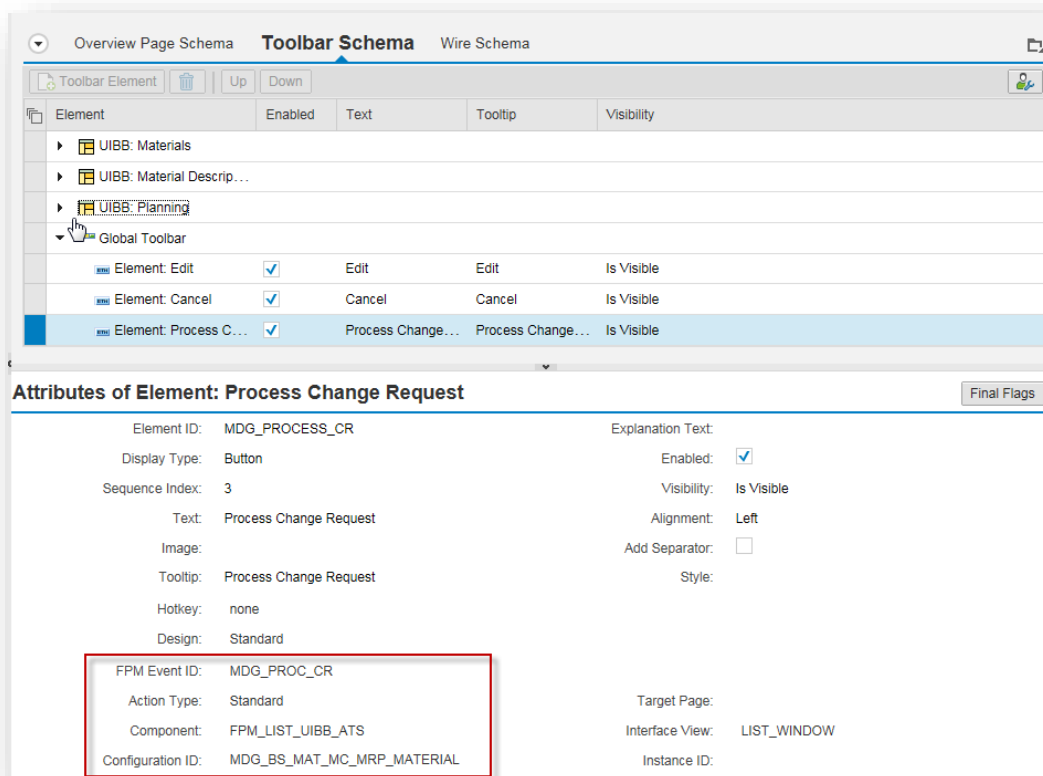
4.8 Define Button in Global Toolbar to Navigate to WD Application USMD_CREQUEST_PROCESS

After having embedded at least one list UIBB to the OVP, you can define the button to navigate to the WD application *USMD_CREQUEST_PROCESS*. This application allows you to analyze change requests generically.

- Edit the component configuration of your OVP component
Open the *Toolbar Scheme* section and add a new button. Set text and tooltip to *Process Change Request*
- Use the value help to set the *FPM Event ID*.

Note:

For each ATS list you have embedded, the FPM event *MDG_PROC_CR* is displayed once in the value help. It is not important which one of these FPM events you select.



Define button to navigate to WD application USMD_CREQUEST_PROCESS

4.9 Adjust Customizing

Start transaction *MDGIMG*.

4.9.1 Define Change Request Types Z*

Define the required change request types for data model MM and business activity MATM. You can use existing change request types for MATM as templates.

Navigate to

General Settings → *Process Modeling* → *Change Requests* → *Create Change Request Type*

These Change Request Types are delivered by SAP using the BC-Sets

MDGM_MDG_MATERIAL_CR_MC_04 - *MDGM_MDG_MATERIAL_CR_MC_08*.

Maintaining classification data is restricted to objects / class types that adhere the “lean classification” standard. For MDG for Material, non-lean classification was supported before MDG 9.1. To distinguish between change request types that support all classification features, and change request types that support the restricted features of lean classification only, the “Switch Classification” check box was introduced. For Multi Record change request types for MDG for Material, this check box needs to be checked.

Change View "Type of Change Request": Overview

New Entries | BC Set: Change Field Values

Type of Chg. Request	Type of Chg. Request	Data Model	Description (medium text)	Obj...	Single Object	Paral...	Main Entity Type	Workflow	Targ...	Default UI Application Na...	UI Configurati...	Switch Clas...
MATLCLF	MM	Multi Pro. for Classification (select)					MATERIAL	WS60800086				<input checked="" type="checkbox"/>
MATLMRP	MM	Multi Pro. for Planning Data (select)					MATERIAL	WS60800086				<input type="checkbox"/>
MATLPUR	MM	Multi Pro. for Purchasing Data (select)					MATERIAL	WS60800086				<input type="checkbox"/>
MATLSAL	MM	Multi Pro. for Sales Data (select)					MATERIAL	WS60800086				<input type="checkbox"/>
MATLUOM	MM	Multi Pro. for Dimensions Data (select)					MATERIAL	WS60800086				<input type="checkbox"/>
MATMRP	MM	Multi Pro. for Planning Data (full)					MATERIAL	WS60800086				<input type="checkbox"/>
MATPUR	MM	Multi Pro. for Purchasing Data (full)					MATERIAL	WS60800086				<input type="checkbox"/>
MATSAL	MM	Multi Pro. for Sales Data (full)					MATERIAL	WS60800086				<input type="checkbox"/>
MATUOM	MM	Multi Pro. for Dimensions Data (full)					MATERIAL	WS60800086				<input type="checkbox"/>

Example for customer CR types ZMM_MAT and ZMM_MARC:

Type of Chg. Request	Type of Chg. Request	Edition Type	Data Model	Description (medium text)	Objects Required	Single Object	Parallel	Main Entity Type	Workflow	Target Sys
ZMM_MARC		MM	Process Multiple Materials (MARC)		<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	MATERIAL	WS60800086	<input type="checkbox"/>
ZMM_MAT		MM	Process Multiple Materials (MATERIAL)		<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	MATERIAL	WS60800086	<input type="checkbox"/>

Note: If you want to use MURP change request types and the parallel change request function (flag for parallel) than make sure, that no Add and Copy Buttons are on the UI. Parallel change requests are only possible for material change.

4.9.2 Define Logical Actions

Check if logical action *MULTI* is maintained.

Navigate to

General Settings → Process Modeling → Business Activities → Define Logical Actions

Change View "Action: Definition": Overview

New Entries

Log. Action	Description
BLOCK	Lock
CHANGE	Change
CHANGE_DQR	Request Change for Data Quality Reasons
CHANGE_DX	Change Using Data Exchange
CLEANSE	Clean Up
CREATE	Create
CREATE_DX	Create Using Data Exchange
DELETE	Delete
DISPLAY	Display
ERASE	Delete (Awesome Vanishing Trick)
HIERARCHY	Hierarchy Processing
LOAD	Data Exchange
MASS	Mass Processing
MULTI	Multi-Object Processing

4.9.3 Define Available UI Applications

Check if WD application MDG_BS_MAT_MC is available.

Navigate to

General Settings → Process Modeling → Business Activities → Define Available UI Applications

MDG UI Applications	
UI Application Name	
MDG_BS_GEN_MC_OVP	
MDG_BS_MAT	
MDG_BS_MAT_MC	
MDG_BS_MAT_OVP	
MDG_BS_MAT_OVP_CORE	
MDG_BS_MAT_SEARCH	

4.9.4 Define Business Activities

Check if the business activity for the respective data model, BO type, and logical action is maintained.

Navigate to

General Settings → Process Modeling → Business Activities → Create Business Activity

Business Activity: Definition							
Bus.Acty	Description (medium text)	D.	Description (medium text)	BO Type	Description	Log. Action	Description
MAT1	Create Material	MM	Material Maintenance MM01/02/MM03	194	Material	CREATE	Create
MAT2	Process Material	MM	Material Maintenance MM01/02/MM03	194	Material	CHANGE	Change
MAT3	Display Material	MM	Material Maintenance MM01/02/MM03	194	Material	DISPLAY	Display
MAT6	Mark Material for Deletion	MM	Material Maintenance MM01/02/MM03	194	Material	DELETE	Delete
MATA	Process Multiple Materials	MM	Material Maintenance MM01/02/MM03	194	Material	MASS	Mass Processing
MATB	Import Material	MM	Material Maintenance MM01/02/MM03	194	Material	LOAD	Data Exchange
MATM	Multi-Records Processing	MM	Material Maintenance MM01/02/MM03	194	Material	MULTI	Multi-Object Processing
PCG1	Create Profit Center Group	PC	Chart of Accounts and Org. Units	806	Profit Center Group	CREATE	Create

4.9.5 Link Logical Actions with Business Activities

Check, if WD application configuration MDG_BS_MAT_MC as potential navigation target exists.

Navigate to

*General Settings → Process Modeling → Business Activities → Link Log. Actions with Bus. Activity:
Standard Definition (client dependent)*

Change View "Business Activity: Determination": Overview						
New Entries						
Business Activity: Determination						
UI Application Name	UI Configuration	Log. Action	Description	Bu...	Description (medium text)	
MDG_BS_MAT_MC	*	MULTI	Multi-Object Processing	MATM	Multi-Records Processing	
MDG_BS_MAT_OVP	BS_MAT_OVP	CHANGE	Change	MAT2	Process Material	

4.9.6 Define Change Request Steps for Rule-Based Workflow

Define your steps for each change request type ZMM_MAT.

Navigate to

General Settings → Process Modeling → Workflow → Rule-Based Workflow → Define Change Request Steps for Rule-Based Workflow

Workflow Step Numbers					
Type of Chg. Request	CR Step	Keys	Validation	Description (medium text)	
ZMM_MAT	0	<input type="checkbox"/>	<input type="checkbox"/>	Processing	
ZMM_MAT	90	<input type="checkbox"/>	<input checked="" type="checkbox"/>	Final Check	
ZMM_MAT	91	<input type="checkbox"/>	<input type="checkbox"/>	Activation	
ZMM_MAT	92	<input type="checkbox"/>	<input type="checkbox"/>	Revision	
ZMM_MAT	93	<input type="checkbox"/>	<input type="checkbox"/>	Validation	
ZMM_MAT	95	<input type="checkbox"/>	<input type="checkbox"/>	Revision Processing	
ZMM_MAT	99	<input type="checkbox"/>	<input type="checkbox"/>	Complete	

4.9.7 Configure Properties of Change Request Step (UI and Highlight Changes)






Assign the UI configurations to the change request steps.

Navigate to

General Settings → Process Modeling → Change Requests → Configure Properties of Change Request Step

Dialog Structure		Type of Change Request	
▼ Type of Change Request		Type of Chg. Request	Description (medium text)
▼ Change Request Step		ZMAT12	ZCreate Material (classic partial act)
• Enhancements and Checks per Change Request Step		ZMATJK	Test Jochen MM
▼ Entity Types per Change Request Step		ZMM_MARC	Process Multiple Materials (MARC)
• Attributes per Change Request Step		ZMM_MAT	Process Multiple Materials (MATERIAL)
• User Interface per Change Request Step		ZPCTRH	profit center group hierarchy

Change View "User Interface per Change Request Step": Overview



Dialog Structure

Type of Chg. Request ZMM_MAT

Type of Change Request

Change Request Step

Enhancements at

Entity Types per

Attributes per

User Interface per C

User Interface per Change Request Step

Step	UI Application Name	Configuration ID	Highlight Changes
00	MDG_BS_MAT_MC	ZMDG_BS_MAT_MC_MATFORM_EM	<input checked="" type="checkbox"/>
90	MDG_BS_MAT_MC	ZMDG_BS_MAT_MC_MATFORM_EM	<input checked="" type="checkbox"/>
91			<input type="checkbox"/>
92			<input type="checkbox"/>
93			<input type="checkbox"/>
95	MDG_BS_MAT_MC	ZMDG_BS_MAT_MC_MATFORM_EM	<input checked="" type="checkbox"/>
99			<input type="checkbox"/>

The *Highlight Changes* checkbox allows you to define for which step highlighting changes should be supported. This setting is only relevant, if you use the new highlighting schema (which means that both parameters *USMD_MC_TCD_SAVED* and *USMD_MC_TCD_UNSAVED* are initial). See also chapter 4.11 Highlighting Changed Table Content. Mark *Highlight Changes* checkbox for CR types that should support highlighting changes.

4.9.8 Adapt Limit of Multiple-Record Processing

You can adapt the limits of multiple-record processing. The limits restrict the number of objects which can be processed by multiple-record processing. You can specify the limits per MDG data model. There is a warning message for data model MM for more than 80 entries.

As objects the leading type-1 entities are counted, for example the material or the business partner.

Following limits are possible:

- **Warning Limit**
If this limit is reached a warning is given to the user that he had to expect long response times of the application. The user can abort the start of multiple-record processing and adapt his selection in the search.
- **Error Limit**
The multiple-record processing doesn't except more objects. The application doesn't continue as an exception as of extreme long times is expected.

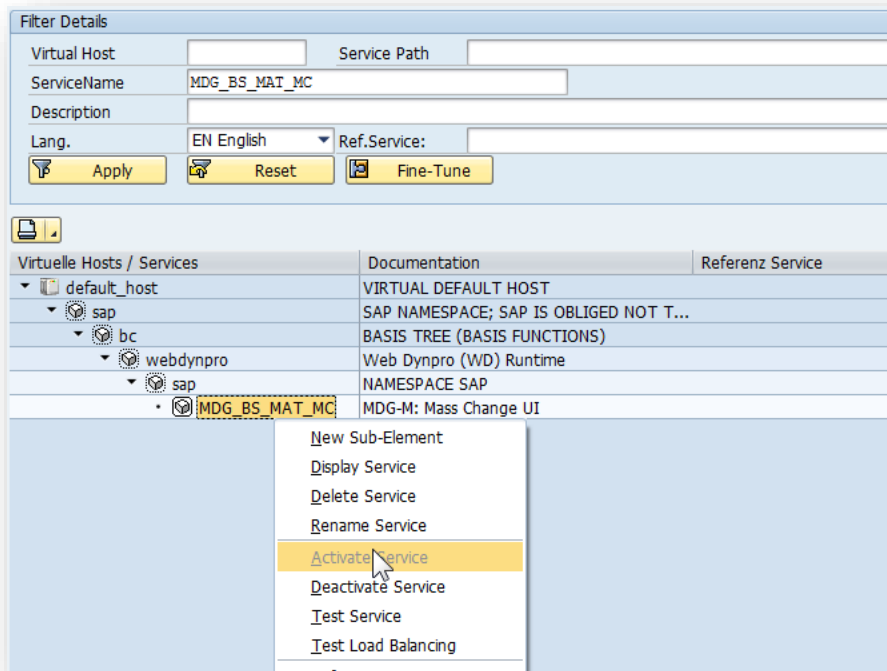
The limit is ignored if it is set to zero. If you set the corresponding warning level to zero, no warning message is given. If you set the corresponding error level to zero, no error message is given.

Call transaction SM30 and enter as Table/View 'V_USMD_MC_LIMIT'. In the following maintenance view, you can specify the limits per governance model. The limits depend mainly on the size of the governance data model, the governance scope, the complexity of the UIs and the performance of the system.

You can also introduce limits for your own governance data models. If not, the generic limit entry (with data model '*') is used instead.

4.10 Activate Service

Activate the HTTP service for your multiple-record processing WD application using transaction *SICF*. The service name equals the name of the WD application. Right-click on the service name and select the menu item *Activate Service*.



4.11 Highlighting Changed Table Content

You can highlight all table cells and form fields if the field contains a value different from the active value. For highlighted fields, tooltip indicate the changes of the field values. In addition, you can indicate changes of your data using an icon in an extra column of your lists.

4.11.1 Color Schema

There are two schemas you can use to highlight fields:

- Schema 1: This schema is provided with all MDG releases supporting the creation of multiple-record processing applications. Colors and icons are used as follows:
 - *unsaved color* is used for changed fields having a value either not saved yet or being different from the saved value – independent of the related active value
 - *saved color* is used for changed fields having a value that is saved but that is also different from the active value
 - *staging icon* is displayed if any entity displayed by the list is changed or created
- Schema 2: This schema supports a unique color schema across single object maintenance UI and multiple-records processing UI. It is available as of MDG 7.0 SP02
 - *unsaved color* is used for changed fields having an unsaved value different from the related active value
 - *saved color* is used for changed fields having a saved value different from the active value
 - *new icon* is displayed, if the related root entity does not exist in the active area
 - *changed icon* is displayed if the related root entity already exists in the active area and at least one of the following statements is true:
 - any field of the root entity changed

- any field of a dependent entity changed
- any dependent entity was created

Schema 1 must be used, if you use MDG7.0 prior to SP02 or if you have not activated the business function set MDG_FOUNDATION_5. It may also be used to keep the UI behavior stable after having installed MDG 7.0 SP2. The *Delete* button is hidden for this color scheme and highlighting of changes is not supported for classification data. Therefore, we strongly recommend using the color scheme 2.

Schema 2: Highlighting changes is only carried out, if allowed for the current CR step (see chapter 4.9.7 Configure Properties of Change Request Step (UI and Highlight Changes)). At application startup, highlighting changes is always switched off for performance reasons. At runtime, highlighting can be toggled. To offer this feature, a toggle button needs to be defined in the floorplan configuration (global toolbar or toolbar of main entity UIBB).

4.11.2 Support of Color Schema

We recommend using the color scheme 2.

Color scheme 1 does not support displaying deleted entities (delete icon, strikethrough text). If you maintain classification data (class assignment and characteristic values), color scheme 2 is obligatory. Change icons, changed fields and field tooltips are not displayed correctly in color scheme 1.

4.11.3 Selecting the Color Schema

The color scheme is influenced by the following parameters:

- Schema 1:
application parameters *USMD_MC_TCD_SAVED* (saved color), *USMD_MC_TCD_UNSAVED* (unsaved color)
SET/GET parameters *USMD_MC_TCD_SAVED* (saved color), *USMD_MC_TCD_UNSAVED* (unsaved color)
- Schema 2:
application parameters *MDG_HC_COLOR_SAVED* (saved color), *MDG_HC_COLOR_UNSAVED* (unsaved color)
SET/GET parameters *MDG_HC_COLOR_SAVED* (saved color), *MDG_HC_COLOR_UNSAVED* (unsaved color), *MDG_HC_DISABLE* (switch off highlighting)

Schema 1 is implicitly selected, if at least one of the parameters *USMD_MC_TCD_SAVED* or *USMD_MC_TCD_UNSAVED* is non-initial. In this case, all parameters related to schema 2 are ignored.

Schema 2 is implicitly selected for all other cases. If the parameter *MDG_HC_COLOR_SAVED* is initial, the default color light orange (65) is used. If the parameter *MDG_HC_COLOR_UNSAVED* is initial, the default color light yellow (82) is used. To switch of the usage of colors (and the calculation of list icons), the SET/GET parameter *MDG_HC_DISABLE* must set to X.

For schema 1, colorizing fields can be suppressed at startup via the SET/GET parameter *USMD_MC_TCD_USE*. Allowed parameter values are Y (Yes) and N (No). All other values will be interpreted as undefined.

For schema 2, this parameter *USMD_MC_TCD_USE* is ignored and the highlighting of changed is always switched off at startup. For both color schema, you can configure a button to toggle the highlighting state at runtime (see chapter 4.4 Create a Configuration for the Application Controller).

4.11.4 Setting Colors

You can set the colors via SET/GET parameters in the user profile. Allowed colors are given by the fixed values of domain `WDUI_TABLE_CELL_DESIGN`.

You can also pass the colors via application parameters. This requires that you have added these parameters to the list of application parameters. Then, you can set the colors via application configuration or customizing. You can also send the required information via URL, (e.g., by attaching name/value pairs to the query string).

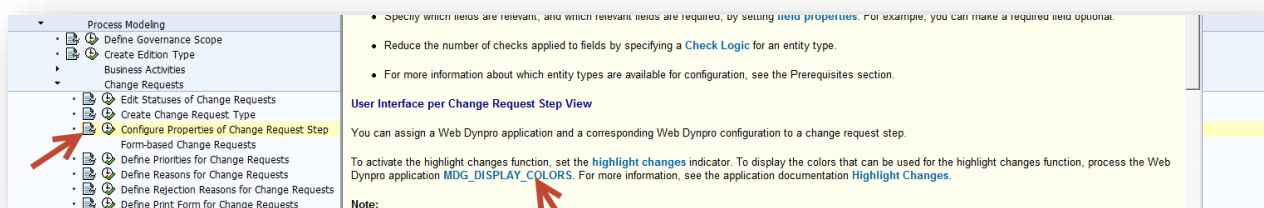
If more than one possibility to set the colors is used, the application checks these possibilities in the following sequence:

- SET/GET parameter
- Application parameters (configured value) if SET/GET parameter is initial
- URL parameter (incl. Shared Objects memory set by navigation) if application parameter is initial

4.11.5 Colors

Start Web Dynpro application `WDR_TEST_TABLE` and click the `CELL DESIGN` link within this application. You get an overview about the available cell designs. In addition, check the fixed values of domain `WDUI_TABLE_CELL_DESIGN` to determine the color number related to a given cell design.

There is a link to the Web Dynpro application [MDG_DISPLAY_COLORS](#) in the documentation of the customizing node *General Settings → Process Modeling → Change Requests → Configure Properties of Change Request Step*.



Starting MDG with Corbu theme the following colors are available:

Colors for Highlighting Table Cells

Color Constant Name: CL_WD_TABLE_COLUMN=>E_CELL_DESIGN-	Color	Color Value
STANDARD		00
NEGATIVE		01
POSITIVE		02
BADVALUE_DARK		03
BADVALUE_MEDIUM		04
BADVALUE_LIGHT		05
CRITICALVALUE_DARK		06
CRITICALVALUE_MEDIUM		07
CRITICALVALUE_LIGHT		08
GOODVALUE_DARK		09
GOODVALUE_MEDIUM		10
GOODVALUE_LIGHT		11
KEY_MEDIUM		12
GROUP_LEVEL1		13
GROUP_LEVEL2		14
GROUP_LEVEL3		15
ONE		16
TWO		17
THREE		18
FOUR		19
CALENDARMETAL		20
CALENDARPEACH		21
CALENDARBLUE		22
CALENDARROSE		23
CALENDARPURPLE		24
CALENDARAQUA		25
CALENDARTEAL		26
CALENDARYELLOW		27
CALENDARBROWN		28
CALENDARSTANDARD		29
CALENDARGREEN		30
FIVE		31
SELECTION		32
TOTAL		33
SUBTOTAL		34
SUBTOTAL_LIGHT		35
ALTERNATING		36
CRITICAL		37

GROUP_HIGHLIGHTED		38
GROUP_HIGHLIGHTED_LIGHT		39
MARKED		40
FILTER		41
FILTERICON		42
POPIN		43
NONE		44
DISABLED		45
TODAY		46
INTENSIFIED		47
COLOR_BLACK		48
COLOR_BROWN		49
COLOR_OLIVE_GREEN		50
COLOR_DARK_GREEN		51
COLOR_DARK_TEAL		52
COLOR_DARK_BLUE		53
COLOR_INDIGO		54
COLOR_GRAY_80		55
COLOR_DARK_RED		56
COLOR_ORANGE		57
COLOR_DARK_YELLOW		58
COLOR_GREEN		59
COLOR_TEAL		60
COLOR_BLUE		61
COLOR_BLUE_GRAY		62
COLOR_GRAY_50		63
COLOR_RED		64
COLOR_LIGHT_ORANGE		65
COLOR_LIME		66
COLOR_SEA_GREEN		67
COLOR_AQUA		68
COLOR_LIGHT_BLUE		69
COLOR_VIOLET		70
COLOR_GRAY_40		71
COLOR_PINK		72
COLOR_GOLD		73
COLOR_YELLOW		74
COLOR_BRIGHT_GREEN		75

COLOR_TURQOISE		76
COLOR_SKY_BLUE		77
COLOR_PLUM		78
COLOR_GRAY_25		79
COLOR_ROSE		80
COLOR_TAN		81
COLOR_LIGHT_YELLOW		82
COLOR_LIGHT_GREEN		83
COLOR_LIGHT_TURQUOISE		84
COLOR_PALE_BLUE		85
COLOR_LAVENDER		86
COLOR_WHITE		87
TRANSPARENT		88

4.12 Transfer Object Keys from Source Application to Multiple-Record Processing WD Application

Currently, the multiple-record processing WD application can only retrieve data qualified by a simple key (For example, material number). The key data can be transferred by URL (resp. Shared Objects memory) or by Global Context. In addition, the object list can be retrieved from a change request by transferring the change request number to the multiple-record processing application. If more than one transfer channel is used, the priority is as follows:

1. Change Request Object List
2. Global Context if change request number is not supplied
3. URL (resp. Shared Objects memory) if Global Context does not contain data

4.12.1 Transfer by Change Request

You can start the multiple-record processing application passing the change request number by the application parameter *USMD_CREQUEST*. In this case, the root type-1 entity object keys are read from the change request.

4.12.2 Transfer by Global Context

Check if the Global Context is already configured as described below. If not, the following steps are required:

Configure Source Application

- Start transaction *BS_CONTEXT*.
- Double-click dialog structure *Define Accords*. Define an accord for your data model (For example, *BS_MAT*) and save.
- Double-click dialog structure *Define Accords*. Select your accord.
- Double-click *According Context IDs*. Choose the *New Entries* button.
- Use value help to select context ID *MC_DATA*. Check the *Mutating* flag. Save.
- Restart transaction *BS_CONTEXT*.

- Now add your source application to the composing instruments. Double-click dialog structure *Define Instrument Types*. Select *WD_APPL*.
- Double-click dialog structure *Composing Instruments*. Click *New Entries* toolbar button.
- Enter source application name (e.g., name of Search WD application) in *Instrument* column and the name of your accord in *Accord* column. Choose *Save*.
- Restart transaction *BS_CONTEXT*.
- Check if your application is displayed in dialog structure *Referring Instruments* for context ID *MC_DATA* and for your accord.

The screenshot shows three SAP configuration screens. The top screen, 'Context IDs', is a table with columns: Context ID, Description, Type Name, Origin, Mutable, and Local. The row for 'MC_DATA' is highlighted in yellow. A blue arrow points from this row to the middle screen, 'Referring Accords', which has a 'Context ID' field set to 'MC_DATA' and a table of 'Referring Accords' with columns 'Accord' and 'Mutating'. The 'MDG_BS_MC' row is highlighted. Another blue arrow points from this row to the right screen, 'Referring Instruments', which has an 'Accord' field set to 'MDG_BS_MC' and a table of 'Referring Instruments' with columns 'InstrType' and 'Instrument'. The rows for 'WD_APPL' with instruments 'MDG_BS_GEN_MC_OVP' and 'MDG_BS_MAT_SEARCH' are highlighted.

Context ID	Description	Type Name	Origin	Mutable	Local
BP	Current Business Partner ID	BU_PARTNER	BS_BP	<input type="checkbox"/>	<input type="checkbox"/>
BPCATEGORY	Current Business Partner's Cate...	BU_TYPE	BS_BP	<input type="checkbox"/>	<input type="checkbox"/>
BP_IDS	List of Business Partner IDs	BST_BUSINESS_PARTNERS	BS_BP	<input checked="" type="checkbox"/>	<input type="checkbox"/>
BP_SELPARS	Selection parameters of the m...	FPMGB_T_SEARCH_CRITERIA	BS_BP	<input type="checkbox"/>	<input checked="" type="checkbox"/>
MC_DATA	Data in Mass Change Scenario	MDG_BS_MC_S_CONTEXT	MDG_BS_MC	<input checked="" type="checkbox"/>	<input type="checkbox"/>
MM	Context for material	TABLE_MAINR	BS_MAT	<input checked="" type="checkbox"/>	<input type="checkbox"/>
REPL_PAR	Replication Parameters	MDG_T_REPLICATION_PARAM	BS_REPL	<input checked="" type="checkbox"/>	<input type="checkbox"/>

Accord	Mutating
BS_BP	<input checked="" type="checkbox"/>
BS_MAT	<input checked="" type="checkbox"/>
MDG_BS_MC	<input checked="" type="checkbox"/>

InstrType	Instrument
WD_APPL	MDG_BS_GEN_MC_OVP
WD_APPL	MDG_BS_MAT_SEARCH

Configure Global Context (example for changing material data)

Write Object Keys to Global Context from Source Application

The list of object keys must be transformed to XML to obtain a type independent representation (*STRING*).

- The XML string needs to be derived from the list of object keys using the identity transformation *ID*.
- The list of object keys needs to have a line type given by the root type-1 entity key structure (For example *MDG_BS_MAT_S_MATERIAL* for entity *MATERIAL*).

The following displays a source code extract for transferring material numbers. The bold code sections must be adapted to fit your scenario. The method *GET_SELECTED_OBJECT_KEYS* must be developed by you.

```
DATA ls_data      TYPE mdg_bs_mc_s_context.
DATA lt_material TYPE mdg_bs_mat_t_material.

* Fill internal table (object keys are contained in column MATERIAL)
get_selected_object_keys (
  IMPORTING
    et_material_id = lt_material ).

* Fill context structure
ls_data-model = 'MM'.
ls_data-entity = 'MATERIAL'.
```

```

CALL TRANSFORMATION id
      SOURCE tab = lt_material
      RESULT XML  ls_data-data.

* Write data to Global Context
TRY.
    lo_global_context = cl_bs_context_factory=>get_global_context( ).
    lo_global_context->set_value(
        EXPORTING
            iv_id      = 'MC_DATA'
            iv_data    = ls_data ).

    CATCH cx_bs_ct_core.
*      Do message handling here
ENDTRY.

```

Note: If you use the edition concept, you can pass the edition value to the to the multiple-record processing application using the *EDITION* component.

4.12.3 Transfer by URL

Use the standard method for navigation purposes *IF_USMD_UI_SERVICES_EXT~NAVIGATE*. Attach corresponding name/value pairs to the importing parameter *IT_DATA*.

If you cannot use this navigation method, transfer the object keys using the Shared Objects memory. Use the area *CL_USMD_WD_PARAMETER_SHM*. Create a new area instance and pass the GUID of this area instance via the URL parameter *USMD_SHM_INSTANCE* to the multiple-record processing application.

Note: The name used in the name/value pairs must be identical to the name of the root type-1 entity for your data model.

4.13 Work with Copies of Supplied Data (for example from material search UI MDG_BS_MAT_SEARCH_06)

You can copy the entities qualified by the supplied object keys when starting your multiple-record processing application. Your application then works on the copies of the supplied objects. The type-1 entity keys are derived via internal number assignment and can be changed at runtime.

To trigger the copy process, you need to send the name/value pair *USMD_NEW_MASS_COPY=X* via URL parameters.

All data is copied. But there is a restriction for entities with leading entity according to table *MDG_BS_MAT_LENTY*. These entities are only copied if there is also a value for an attribute in addition to the key.

If a field value will be copied depends also on customizing of the backend material master (transaction *OMSR*).

Note: Do not send this information if you supply the object keys implicitly via an already existing change request.

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
3043582	MDG Customer Connection 2020
3194967	MDG Customer Connection 2021 for S/4HANA 2022
3311039	MDG Customer Connection 2023
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
2950412	Functional restrictions in MDG Process Analytics on SAP S/4HANA 2020
3066855	Functional restrictions in MDG Process Analytics on SAP S/4HANA 2021
3225098	Functional restrictions in MDG Process Analytics on SAP S/4HANA 2022
3381795	Functional restrictions in MDG Process Analytics on SAP S/4HANA 2022
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
2883107	MDG Customer Connection: Multi-Record Processing Performance Improvement (Composite SAP Note for performance improvements)
2479869	Usage of Lean Classification with SAP Master Data Governance
2846478	MURP: Unexpected field property errors after copy of material
2803671	Multi Record Processing: F4 help has no values in 'Find and Replace' pop-up
2757392	MDG-M: Poor performance when reading valuation area data for plants
2724258	Multiple Record Processing: Switch to "Edit" mode causes extended runtime
2755519	Multiple-Record processing: Entities based on Custom UI Nodes not copied