PUBLIC

How To: Maintain Material-Related Objects in a Follow-up Work Item

Applicable Releases:

all

Version 3.0

March 2023

Document History

Document Version	Description
1.0	First official release of this guide (Feb 2014)
2.0	Version update (Jun 2019)
3.0	Layout update (Mar 2023)

Contents

1 BUSINESS SCENARIO 4

1.1 Other Solutions 5

1.2 Change Request is not activated 5

1.3 Additional Steps after Activation (from S/4HANA 1909) 5

2 STEP BY STEP EXPLANATION 6

2.1 Create a customer-specific Workflow and integrate it into the Rule-Based Workflow 6

2.2 Create Web Dynpro for ABAP application 8

2.3 Workflow task and SAP NetWeaver Business Client (POWL) integration 9

2.3.1 Extend the PFCG Role 9

2.3.2 Adapt POWL Feeder Class 10

3 ADDITIONAL INFORMATION 14

3.1 Further Reading 14

3.1.1 Information on SAP MDG on SAP S/4HANA 14

3.1.2 SAP Roadmap Explorer 14

3.1.3 Related Information 14

3.2 SAP Notes 14

1 Business Scenario

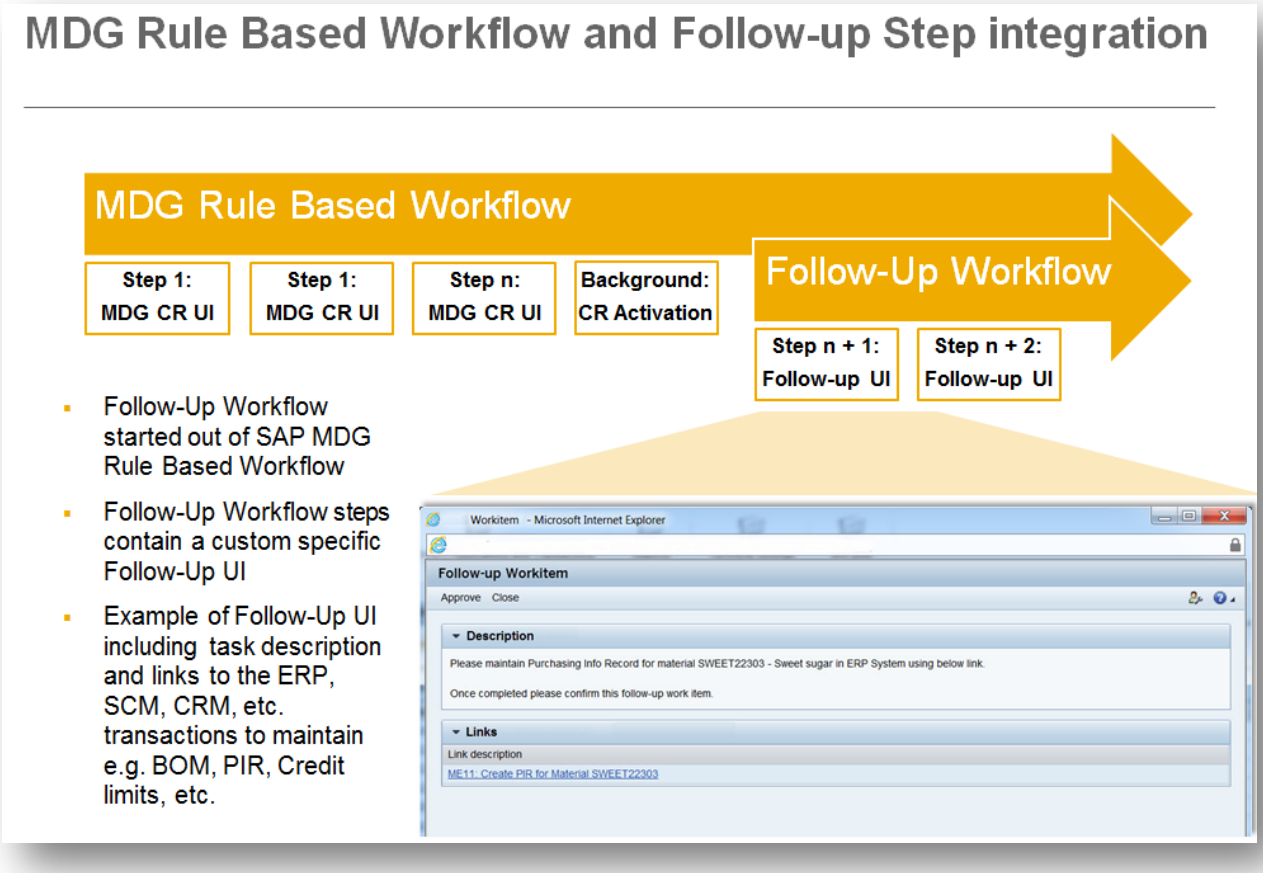
SAP Master Data Governance (MDG) provides business processes to find, create, change, and mark master data for deletion. It supports the governance of master data in a central hub and the distribution to connected operational and business intelligence systems.

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.

This scenario addresses the customer requirement for the maintenance of material related objects that are not in the scope of the SAP standard delivery, for example, Bill of Material, Purchasing Info Records, and Production Versions, but are part of the holistic master data maintenance process.

This How-To Guide describes how to integrate follow-up work items for Bill of Material, Purchasing Info Records, and Production Versions maintenance into the SAP MDG Change Request processing.



In general, the SAP standard rule-based workflow triggers a customer-specific workflow as background follow-up activity after the change request has been successfully activated. The customer-specific workflow creates a new work item with a customer-specific UI (example depicted above) that can be assigned to the relevant user group, role, or department. As with all change request work items, this one is populated in the MDG Inbox (For example, the POWL of SAP NetWeaver Business Client) and can be executed out of this Inbox.

The UI example above is a customer-specific UI and it is independent from SAP MDG Change Request. It consists of 3 sections:

- **Actions:** Buttons like “Confirm”, “Close” or “Send back”
- **Description:** Information about the activity to be done
- **Links**: Links to several transactions in different systems (ERP, SCM, CRM, and so on) to maintain material-related objects

1.1 Other Solutions

1.2 Change Request is Not Activated

When your change request is still in process and not activated, you can use the step relevant UI configuration to assign a reduced UI (with links into the required transaction in different systems). This assignment can be maintained in *MDGIMG -> General Setting -> Process Modelling -> Change Requests -> Configure Properties of Change Request Step*. However, if your change request is already activated (Material, Customer, Supplier, etc. already created in SAP ECC) you are not allowed per standard definition to process the already closed change request.

1.3 Additional Steps After Activation (from S/4HANA 1909)

Master Data Governance, Generic Functions 1909

You can now define additional workflow steps after the activation step, if necessary. In other words, the workflow no longer has to end after a change request has been activated successfully.

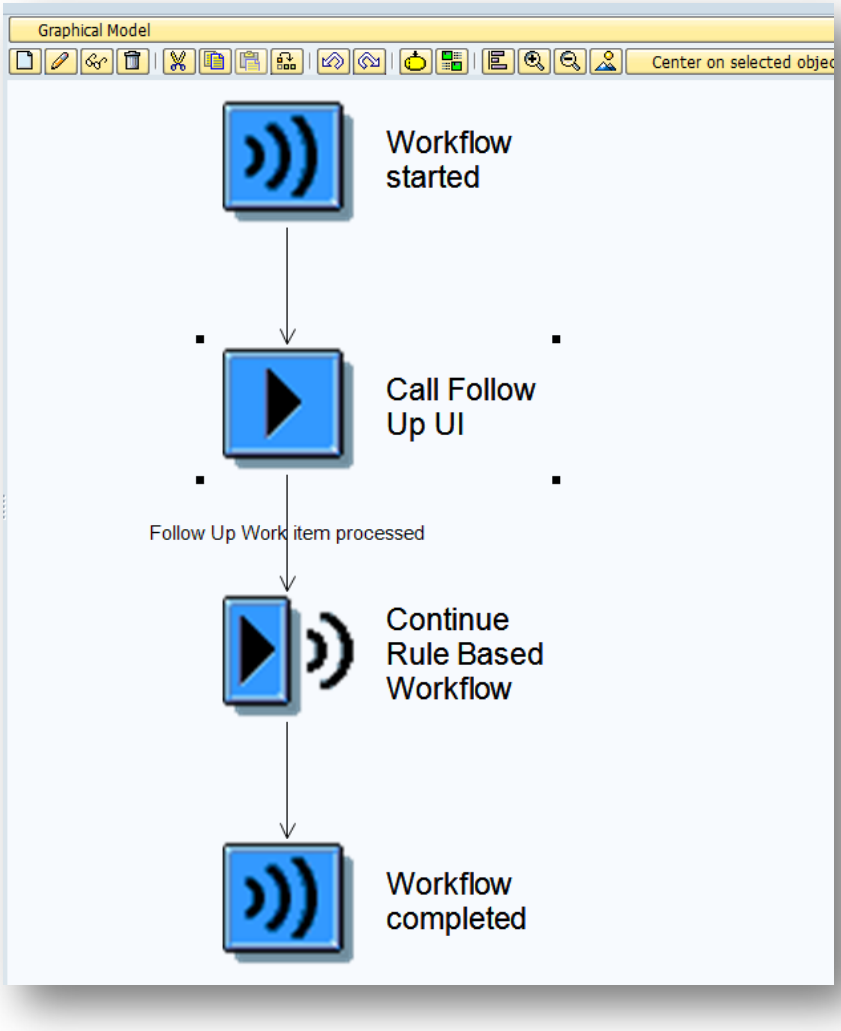
Instead, after a change request has been activated, further actions can be performed to trigger additional dialog workflow steps. These may include additional steps related to the object of the change request itself, and could also result in a follow-up change request as well (as after change request activation, new change requests can be created for the object with new data since staging content no longer exists). A workflow item for these additional steps can then be found in the workflow inbox of the user who has been assigned accordingly.

2 Step by Step Explanation

This document is an extension of the already existing guides. Check existing How-To Guides here: https://community.sap.com/topics/master-data-governance/how-to#central-governance---workflow

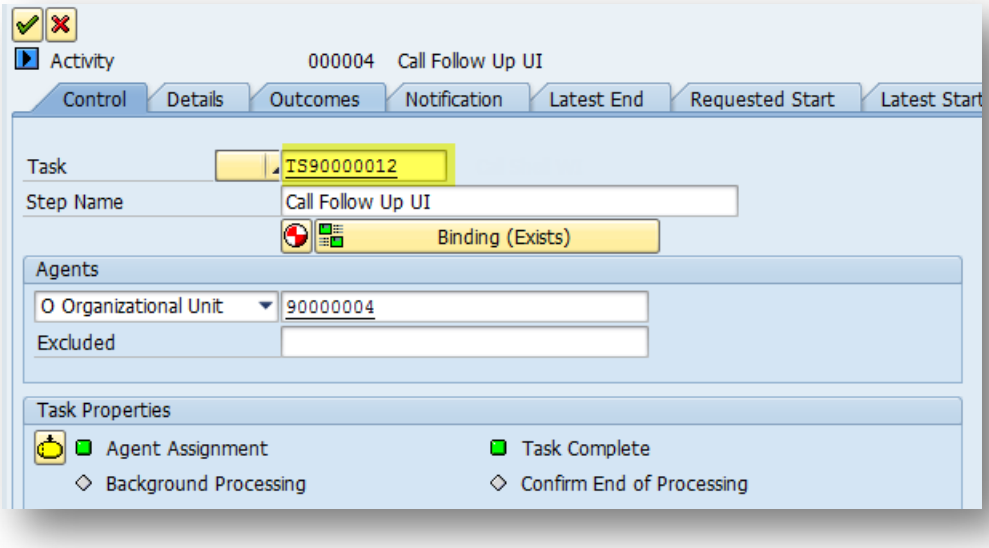
2.1 Create a Customer-Specific Workflow and Integrate It Into the Rule-Based Workflow

Create a new customer-specific workflow including one dialog step for the follow-up task and event trigger for the rule-based workflow. This is based on the How-To Guide: Send a Mail Notification During the Governance Process.



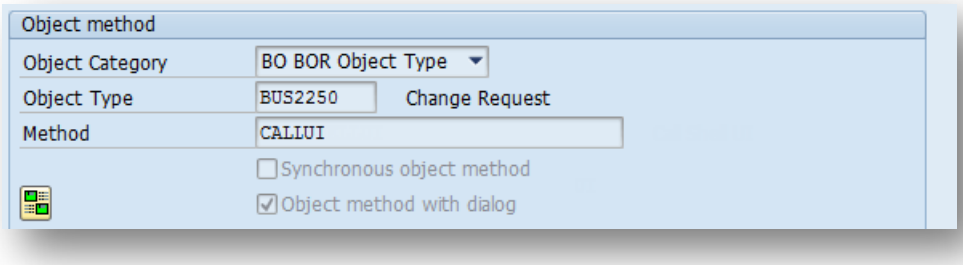
Once implemented, call this workflow from the rule-based workflow BRFplus configuration for the corresponding change request type. Remember your Task ID (For example, TS90000012) of the “Call Follow Up UI” step as it

will be necessary in the next steps:



Maintain the agent assignment: it can be a rule, role, or organization unit. Do not forget to maintain the general task assignment if necessary!

Within the dialog task, call a method (for example, CALLUI) that you created before in the delegated sub-type of SAP standard Business Object Type BUS2250. Refer to SAP Help (Extending Object Types: Inheritance and Delegation) on how to create a sub-type for a business object and a customer-specific method.



If your users use SAP Business Workplace in addition to the POWL in SAP NetWeaver Business Client, add the following code into your BO method. By executing a work item from SAP NetWeaver Business Client, the code of this method won't be executed and you can leave the implementation of this method empty.

The parameter `lv_application_id = 'Z_MD_MDGM_WF_FOLLOW_UP'` is important and needs to be replaced by your Web Dynpro application ID.

```
begin_method callui changing container.

DATA: lo_ui_service               TYPE REF TO if_usmd_ui_services,
      lv_application_id           TYPE string,
      lv_action                   TYPE usmd_action,
      ls_parameter                TYPE usmd_s_value,
      lt_parameters               TYPE usmd_t_value,
      lv_wiid                     TYPE sww_wiid,
```

```
      lv_object_key               LIKE swotobjid-objkey,
      lv_workitem_obj             TYPE swc_object.

swc_get_element    container  '_WORKITEM'  lv_workitem_obj.
swc_get_object_key lv_workitem_obj  lv_object_key.

lv_wiid = lv_object_key.

lv_application_id = 'Z_MD_MDGM_WF_FOLLOW_UP'.
lv_action         = ''.

ls_parameter-fieldname = 'IV_CREQUEST'.
ls_parameter-value = object-key-crequest.
INSERT ls_parameter INTO TABLE lt_parameters.
ls_parameter-fieldname = 'IV_WIID'.
ls_parameter-value = lv_wiid.
INSERT ls_parameter INTO TABLE lt_parameters.

lo_ui_service = cl_usmd_ui_services=>get_instance( ).
lo_ui_service->navigate(
  i_application = lv_application_id
  i_usmd_action = lv_action
  i_inplace_navigation = abap_true
  it_data       = lt_parameters ).

end_method.
```

2.2 Create Web Dynpro for ABAP Application

Refer to the documentation on SAP Community on how to develop a customer-specific user interface using Web Dynpro for ABAP and Floor Plan Manager.

From our point of view, it is important to create two application parameters (IV_CREQUEST and IV_WIID) that will be passed from the calling routine (code above) to the Web Dynpro application. The IV_CREQUEST (Change Request ID) parameter can be used to determine texts or display some change request specific data like material number, description and so on. IV_WIID (Work item ID) is optional and can be used to reserve, put back work item, or to read relevant values from the work item container.

You can store these parameters in the Component Controller or Assistance Class to be accessible in the whole context of the Web Dynpro application.

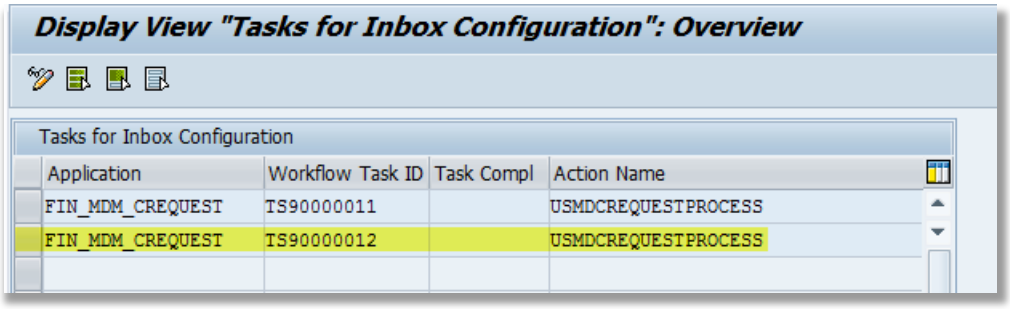
You can develop your Web Dynpro application based on your business requirements. It can include some texts describing the activity to be done and links to the SAP transactions (see example depicted in chapter “Business Scenario”).

Links should include the required transaction parameters (e.g., material, plant number, etc.) to make the maintenance of the material relevant objects (BOM, PIR, PV, or others) more comfortable. If single sign-on is available, the additional logon to the corresponding system is not required. While implementing the solution for the link creation, align with the Basis, Security, and Portal administrators as there are different policies at the customer’s side that cannot be discussed here.

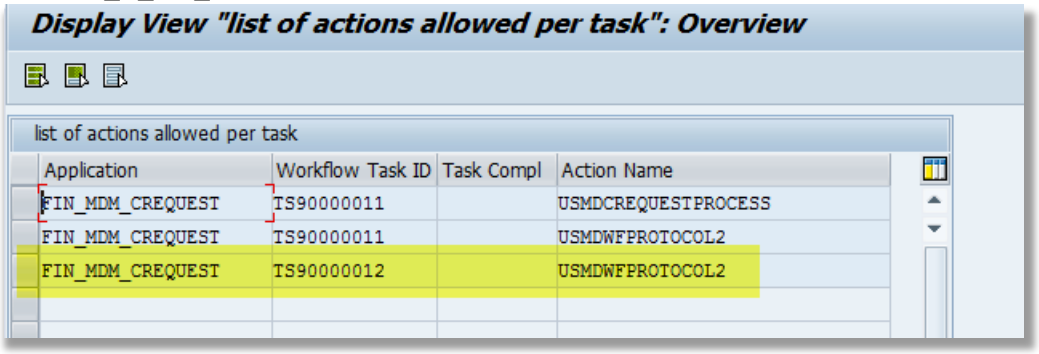
2.3 Workflow Task and SAP NetWeaver Business Client (POWL) Integration

As you have created your own workflow definition with new workflow tasks, the corresponding work items need to be populated in the MDG Inbox. The document SAP Master Data Governance: How to Add an Additional Task to the Inbox (Additional task in the Inbox) describes how you can maintain the task to be populated in MDG Inbox (POWL).

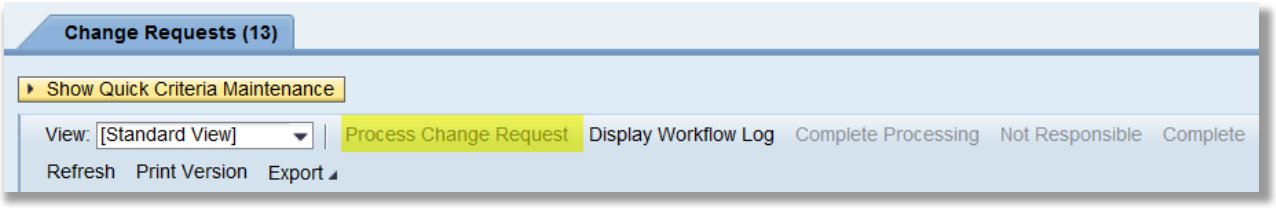
View IBO_C_WF_TAC:



View IBO_C_WF_TTAC:



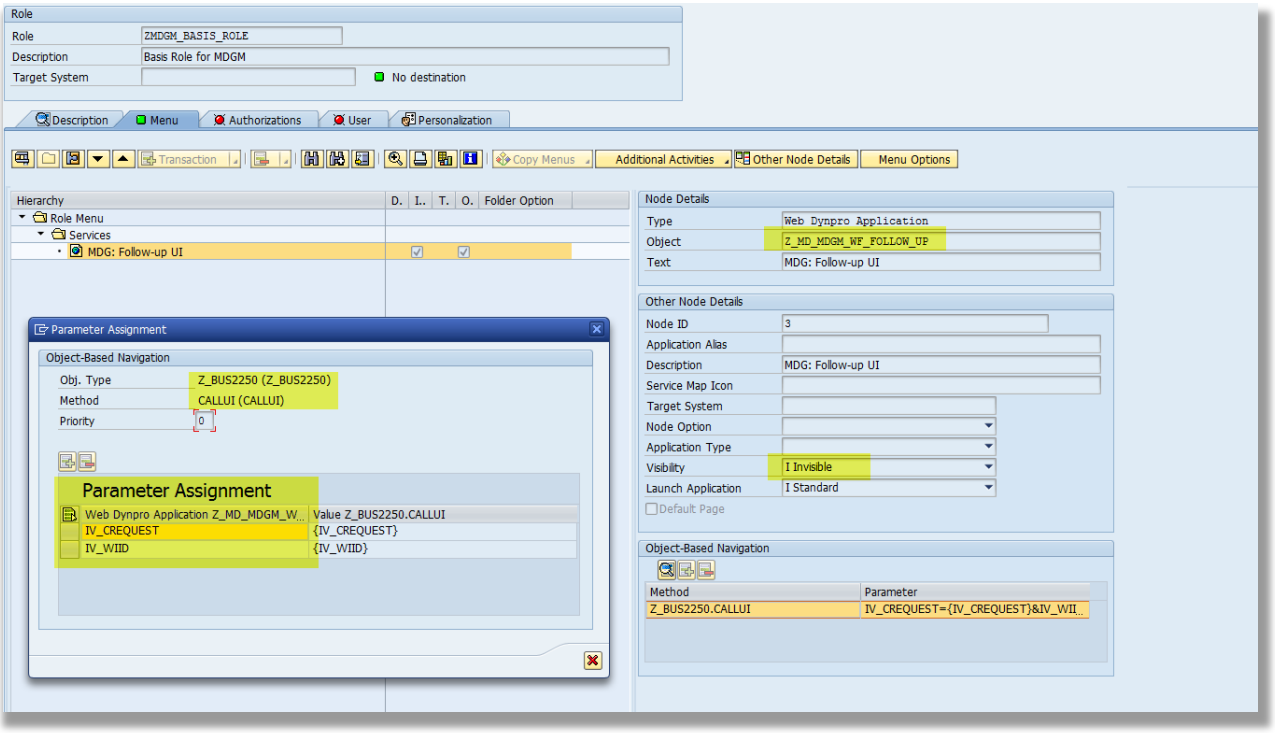
Do not maintain Action Name “USMDCREQUESTPROCESS” for your task in the maintenance view IBO_C_WF_TTAC. As the follow-up work item does not execute the change request maintenance, we need the button “Process Change Request” in the POWL actions to be disabled.



To open the Web Dynpro application, we need additional configuration steps described below.

2.3.1 Extend the PFCG Role

Extend one of the existing PFCG roles or create a new one with a new menu node and OBN parameters pointing to the newly created method of BUS2250 subtype. Choose the Type Web Dynpro Application and maintain in the Object field the name of your Web Dynpro application created in the step above:



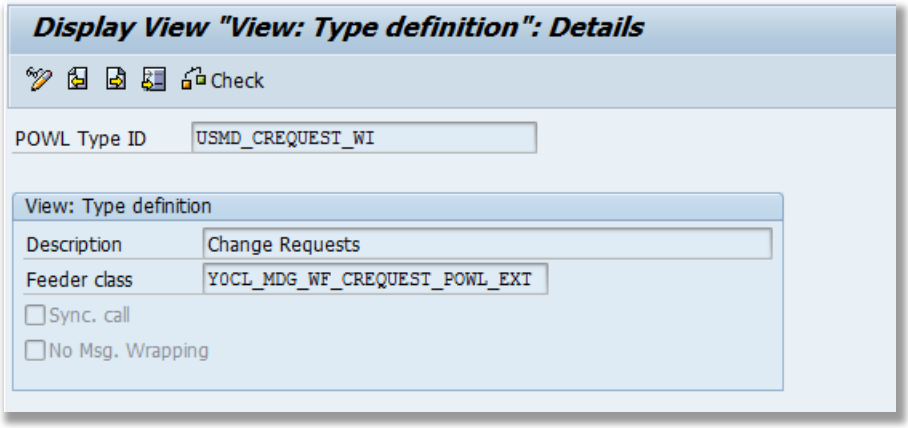
For the object-based navigation parameters maintain the parameters created as application parameters of the Web Dynpro application (IV_CREQUEST and IV_WIID) **AND** please pay attention while creating the CALLUI method that you also maintain these two parameters as importing parameters of the sub-type method. Otherwise, you will not be able to maintain the parameter assignment. Set the parameter “Visibility” to “Invisible” if the link for this application should not be displayed in the navigation tree on the left of SAP NetWeaver Business Client (NWBC) screen.

Assign this role to the users responsible for the follow-up work item execution and maintenance of the material master related objects.

2.3.2 Adapt POWL Feeder Class

Go to transaction POWL_TYPE and select the POWL Type ID “USMD_CREQUEST_WI”. Create a new ABAP/OO sub-class inheriting from the ABAP/OO standard class CL_USMD_CREQUEST_POWL using

transaction SE24. Then maintain your sub-class for the POWL Type:



Redefine the IF_POWL_FEEDER~HANDLE_ACTION method of your sub-class implementing following code:

```
METHOD if_powl_feeder~handle_action.

  CALL METHOD super->if_powl_feeder~handle_action
    EXPORTING
      i_username                = i_username
      i_applid                  = i_applid
      i_type                    = i_type
      i_actionid                = i_actionid
      i_changed                 = i_changed
      i_action_index            = i_action_index
      i_action_conf             = i_action_conf
      i_langu                   = i_langu
      i_additional_data         = i_additional_data
*     i_visible_fields          = i_visible_fields
    IMPORTING
      e_portal_actions          = e_portal_actions
      e_messages                = e_messages
      e_do_refresh              = e_do_refresh
      e_result_lines_changed    = e_result_lines_changed
      e_changes_processed       = e_changes_processed
      e_selected_changed        = e_selected_changed
      e_actions_changed         = e_actions_changed
    CHANGING
      c_selected                = c_selected
      c_result_tab              = c_result_tab
*     c_workflow_result_count   = c_workflow_result_count
      c_action_defs             = c_action_defs
      c_first_visible_row       = c_first_visible_row
      c_first_visible_scroll_col = c_first_visible_scroll_col.


  DATA: ls_workitem             TYPE ibo_s_inbox_workitem,
        lt_navigation_params    TYPE powl_namevalue_tty,
```

```
        ls_navigation_param    TYPE powl_namevalue_sty,
        lv_crequest            TYPE usmd_crequest,
        lv_rcode               TYPE sy-subrc,
        lt_message             TYPE STANDARD TABLE OF swr_mstruc.

  FIELD-SYMBOLS: <ls_message>  TYPE swr_mstruc,
                 <ls_powl_msg> TYPE powl_msg_sty.

  IF i_actionid = 'WI_TEXT'.
    READ TABLE c_result_tab INTO ls_workitem INDEX i_action_index.

*   Check follow-up work item to be called. If so switch portal actions
*   to Follow Up UI and pass relevant parameters
    CHECK ls_workitem-wi_rh_task = 'TS90000012'.

    "Reserve WI before using OBN
    CALL FUNCTION 'SAP_WAPI_RESERVE_WORKITEM'
      EXPORTING
        workitem_id    = ls_workitem-wi_id
        actual_agent   = sy-uname
*       DO_COMMIT      = 'X'
      IMPORTING
        return_code    = lv_rcode
      TABLES
        message_struct = lt_message.


    IF lv_rcode NE 0.
      LOOP AT lt_message ASSIGNING <ls_message>.
        APPEND INITIAL LINE TO e_messages ASSIGNING <ls_powl_msg>.
        <ls_powl_msg>-msgid       = <ls_message>-msgid.
        <ls_powl_msg>-msgnumber   = <ls_message>-msgno.
        <ls_powl_msg>-msgtype     = <ls_message>-msgty.
        <ls_powl_msg>-message_v1  = <ls_message>-msgv1.
        <ls_powl_msg>-message_v2  = <ls_message>-msgv2.
        <ls_powl_msg>-message_v3  = <ls_message>-msgv3.
        <ls_powl_msg>-message_v4  = <ls_message>-msgv4.
      ENDLOOP.
      CLEAR: e_portal_actions. "no navigation
      RETURN.
    ENDIF.

    e_portal_actions-bo_name         = 'Z_BUS2250'.
    e_portal_actions-bo_op_name      = 'CALLUI'.


*   Receive and set Change Request ID into navigation parameters
    ls_navigation_param-key = 'IV_CREQUEST'.

    cl_usmd_wf_service=>get_wi_crequest(
      EXPORTING
```

```
          id_wi_id          = ls_workitem-wi_id
      IMPORTING
          ed_crequest       = lv_crequest ).

    ls_navigation_param-value = lv_crequest.
    APPEND ls_navigation_param TO lt_navigation_params.


*   Set work item ID into navigation parameters
    ls_navigation_param-key = 'IV_WIID'.
    ls_navigation_param-value = ls_workitem-wi_id.
    APPEND ls_navigation_param TO lt_navigation_params.

    e_portal_actions-parameters      = lt_navigation_params.

  ENDIF.
ENDMETHOD.
```

Instead of hard-coding the task in line `CHECK ls_workitem-wi_rh_task = 'TS90000012'` please implement a customizing solution to assign the task as the follow-up one.

Once all the above steps are completed you should receive follow-up work items as part of your process and you can execute them out of the MDG Inbox.

3 Additional Information

3.1 Further Reading

3.1.1 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
- Learn more: Latest Release | Webinars | Help Portal | How-to Information | Key Presentations

3.1.2 SAP Roadmap Explorer

- Please see the roadmap for SAP Master Data Governance

3.1.3 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

3.2 SAP Notes

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

Note	Description
2221398	MDG-BP/C/S/CA: (Un-)Supported Fields in Data Model BP
2847807	MDG-BP/C/S/CA: Usage of MDG Tools and Processes
2313368	Functional restrictions in MDG for Business Partner / Customer / Supplier with SAP Master Data Governance 9.0
2472845	Functional restrictions in MDG for Business Partner / Customer / Supplier with SAP Master Data Governance 9.1
2656712	Functional restrictions in MDG for Business Partner / Customer / Supplier in SAP Master Data Governance 9.2 and on SAP S/4HANA 1809
2816557	Functional restrictions in MDG for Business Partner / Customer / Supplier on SAP S/4HANA 1909
2925030	Functional restrictions in MDG for Business Partner / Customer / Supplier on SAP S/4HANA 2020
3070003	Functional restrictions in MDG for Business Partner / Customer / Supplier on SAP S/4HANA 2021
3220117	Functional restrictions in MDG for Business Partner / Customer / Supplier on SAP S/4HANA 2022
3194967	MDG Customer Connection 2021 for S/4HANA 2022
3043582	MDG Customer Connection 2020
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
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)