PUBLIC

How-To Enhance the UI Building Block (UIBB) for Change Requests by the example of a Requesting Business Area

Applicable releases:
All

Version: 3.0
Date: 02/2025
Owner: PCP Master Data Governance

Document History

Document Version	Description
1.0	First official release of this guide (2013)
2.0	Formatting, Code adjustements
3.0	Formatting, Code adjustements

Table of Contents

1. Business Scenario

SAP Master Data Governance 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.

In this example, you require an extra parameter to control the process and the workflow for change requests - Requesting Business Area.

You do not model this parameter is as part of the MDG data model because it is not part of the business context. Instead, you store the parameter together with the change request number in a Z-table.

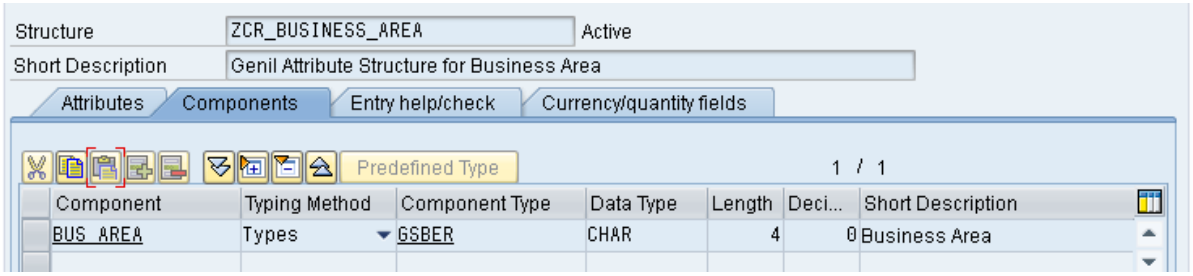
In addition, you place the parameter on the change request UIBB on the tab for the general data. The user can select from business areas defined in Customizing. (The relevant data element is GSBER and the relevant table is TGSB). When a user opens the change request for display, the Requesting Business Area parameter is displayed and cannot be changed.

2. Step by Step Explanation

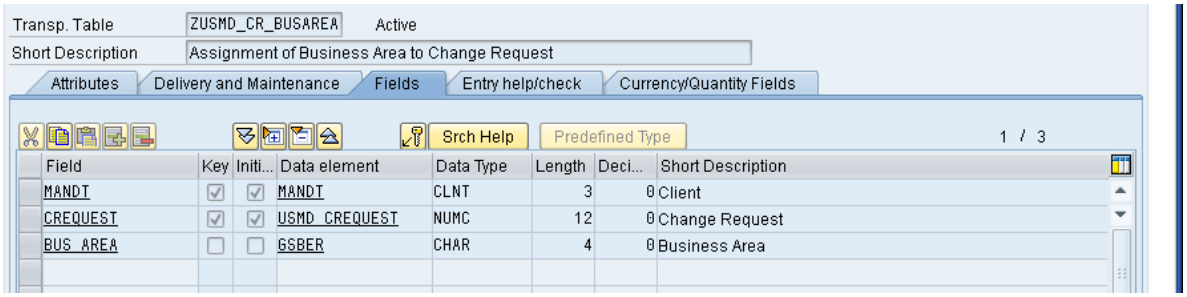
2.1. Create Data Dictionary (DDIC) Objects

The following data dictionary objects will be needed in the configuration steps later on.

2.1.1. Create Structure Z_CR_BUSINESS_AREA



2.1.2. The Genil Attribute Structure for Business Area (ZCR_BUSINESS_AREA) structure contains a component for the additional parameter Requesting Business Area and uses the existing data element GSBER. Later, you use this attribute structure to enhance the Genil data model of the change request.



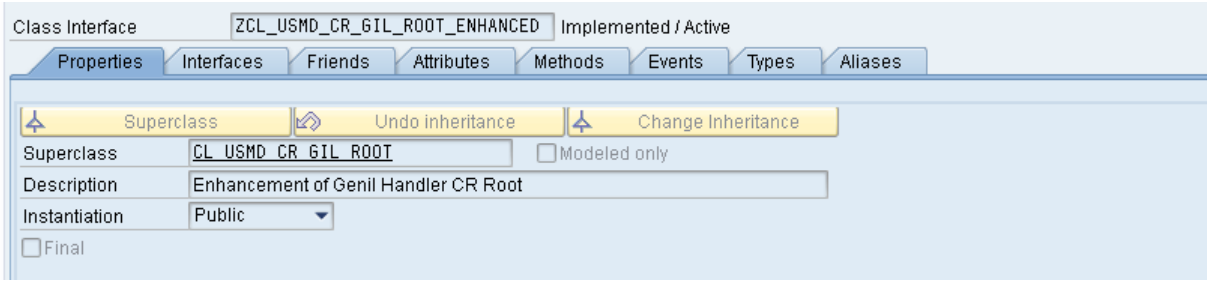
2.1.3. Create Database Table ZUSMD_CR_BUSAREA

Later, you use the *Assignment of Business Area to Change Request* (ZUSMD_CR_BUSAREA) transparent table to store the additional parameter '*Requesting Business Area*'.

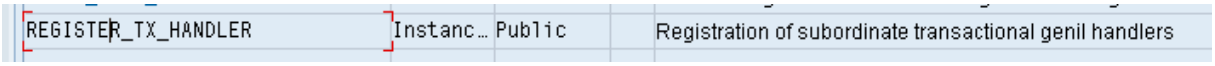
2.2. Create and Implement Classes

2.2.1. Genil Handler Class for Change Request Root: ZCL_USMD_CR_GIL_ROOT_ENHANCED

This class is a redefinition of the *Enhancement of Genil Handler CR Root* (CL_USMD_CR_GIL_ROOT) class.



In our example, we add the Registration of subordinate GenIL handlers (REGISTER_TX_HANDLER) public instance method.



In addition, we define attribute MT_TX_HANDLERS to keep the registered subordinate handler instances and a corresponding table type TT_TX_HANDLER.

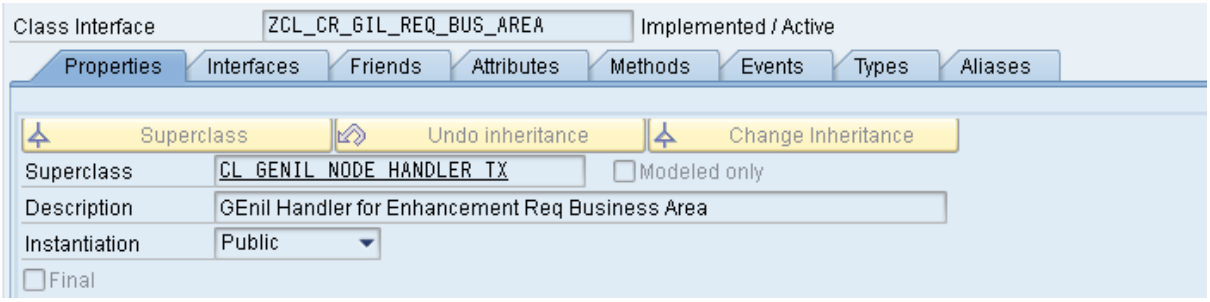
The source code for the ZCL_USMD_CR_GIL_ROOT_ENHANCED class is shown below.

Source code for class ZCL_USMD_CR_GIL_ROOT_ENHANCED

```
CLASS zcl_usmd_cr_gil_root_enhanced DEFINITION
  PUBLIC
  INHERITING FROM cl_usmd_cr_gil_root
  FINAL
  CREATE PUBLIC .

  PUBLIC SECTION.
    TYPES tt_tx_handler TYPE STANDARD TABLE OF REF TO if_genil_node_handler_tx.
    METHODS register_tx_handler
      IMPORTING !io_tx_handler TYPE REF TO if_genil_node_handler_tx.

  PROTECTED SECTION.
    DATA mt_tx_handlers TYPE tt_tx_handler.

  PRIVATE SECTION.
ENDCLASS.

CLASS zcl_usmd_cr_gil_root_enhanced IMPLEMENTATION.
  METHOD register_tx_handler.
    APPEND io_tx_handler TO me->mt_tx_handlers.
  ENDMETHOD.
ENDCLASS.
```

2.2.2. GenIL Handler Class ZCL_CR_GIL_REQ_BUS_AREA

The ZCL_CR_GIL_REQ_BUS_AREA class is a redefinition of the *GenIL Handler for Enhancement Req Business Area* (CL_GENIL_NODE_HANDLER_TX) class.



The handler class of the GenIL object takes care of the data operations (for example Create, Read, and Delete) of the corresponding object. In our example, you must redefine the following methods:

- GET_KEYS_BY_PARENT
  Typically used to get the keys for 1..n relations (for example, retrieving all attachments of a change request).
- CHANGE_OBJECT
  Called if a dependent object is going to be created.
- GET_ATTRIBUTES
  Used to retrieve the attribute data of the dependent object.
- GET_ATTRIBUTE_PROPERTIES
  Used to set the attribute properties (e.g. read only, changeable) of the dependent object.
- CONSTRUCTOR
  The source code for the ZCL_CR_GIL_REQ_BUS_AREA class is shown below.

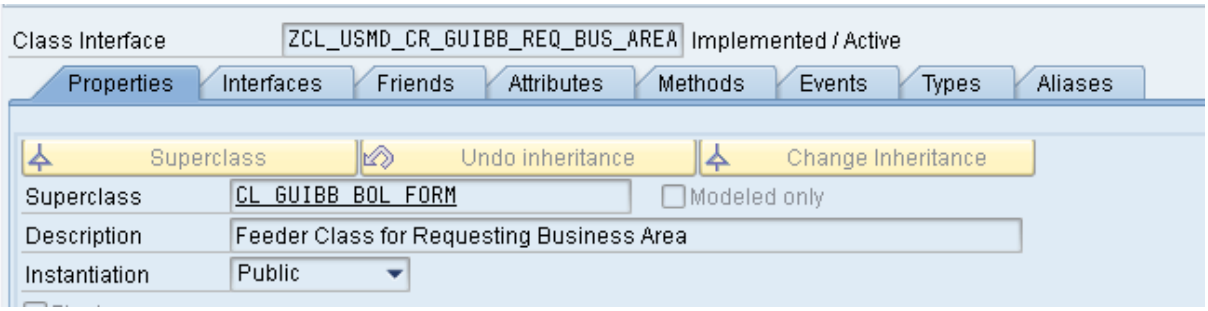
Source code of class ZCL_CR_GIL_REQ_BUS_AREA

```
CLASS zcl_cr_gil_req_bus_area DEFINITION
  PUBLIC
  INHERITING FROM cl_genil_node_handler_tx
  FINAL
  CREATE PUBLIC .

  PUBLIC SECTION.
    METHODS constructor
      IMPORTING   !iv_handler_factory TYPE REF TO cl_genil_node_handler_factory
            !iv_object_name TYPE crmt_ext_obj_name
            !iv_parent_component TYPE REF TO if_genil_appl_intlay.

  PROTECTED SECTION.
    METHODS change_object REDEFINITION.
    METHODS get_attributes REDEFINITION.
    METHODS get_attribute_properties REDEFINITION.
    METHODS get_keys_by_parent REDEFINITION.

  PRIVATE SECTION.
ENDCLASS.

CLASS zcl_cr_gil_req_bus_area IMPLEMENTATION.

  METHOD constructor.
    CALL METHOD super->constructor

      EXPORTING
        iv_handler_factory = iv_handler_factory
        iv_object_name = iv_object_name
        iv_parent_component = iv_parent_component.

    "if this instance of node handler class does implement the [.underline]#tx# interface of
    "GenIL objects (IF_GENIL_NODE_HANDLER_TX) register it to its root as a [.underline]#tx#
    "handler that needs to be called whenever [.underline]#tx# methods are called for the
    "root object itself
```

```
    TRY.
        DATA(tx_handler) = CAST if_genil_node_handler_tx( me ).
      CATCH cx_sy_move_cast_error.
        RETURN.
    ENDTRY.

    DATA(root_object_name) = me->object_model->get_root_object( iv_object_name = me->my_object_name ).
    DATA(root_handler) = CAST zcl_usmd_cr_gil_root_enhanced(
      me->handler_factory->get_base_obj_handler( root_object_name ) ).
    root_handler->register_tx_handler( tx_handler ).
  ENDMETHOD.

  METHOD change_object.
    DATA(attributes) = VALUE zcr_business_area( ).
    iv_cont_obj->get_attributes( IMPORTING es_attributes = attributes ).

    DATA(crequest_id) = CONV usmd_crequest( is_key ).
    IF crequest_id IS NOT INITIAL.

      SELECT SINGLE * FROM zusmd_cr_busarea INTO @DATA(business_area) WHERE crequest = @crequest_id.
      IF sy-subrc <> 0.
        business_area-crequest = crequest_id.
      ENDIF.

      DATA(data_changed) = abap_false.
      LOOP AT it_changed_attributes ASSIGNING FIELD-SYMBOL(<changed_attribute>).
        ASSIGN COMPONENT <changed_attribute> OF STRUCTURE attributes TO FIELD-SYMBOL(<value>).
        IF sy-subrc = 0 AND <value> IS ASSIGNED.
          ASSIGN COMPONENT <changed_attribute> OF STRUCTURE business_area TO FIELD-SYMBOL(<value2>).
          IF sy-subrc = 0 AND <value2> IS ASSIGNED.
            <value2> = <value>.
            data_changed = abap_true.
          ENDIF.
        ENDIF.
      ENDLOOP.

      IF data_changed = abap_true.
        MODIFY zusmd_cr_busarea FROM business_area.
      ENDIF.
    ENDIF.

    rv_success = abap_true.
  ENDMETHOD.

  METHOD get_attributes.
    DATA(key) = CONV bss_cril_root_key( is_key ).

    SELECT SINGLE * FROM zusmd_cr_busarea INTO @DATA(business_area_db) WHERE crequest = @key-cr_id.
    DATA(business_area_ui) = VALUE zcr_business_area( ).
    MOVE-CORRESPONDING business_area_db TO business_area_ui.
    iv_cont_obj->set_attributes( is_attributes = business_area_ui ).
  ENDMETHOD.

  METHOD get_attribute_properties.
    iv_property_object->set_property_by_name(
    iv_name = 'BUS_AREA'
    iv_value = if_genil_obj_attr_properties=>changeable ).
  ENDMETHOD.

  METHOD get_keys_by_parent.
    CASE iv_parent_rel.
      WHEN 'ZCR_ReqBusAreaRel'. " Get CR IS
        DATA(cr_id) = VALUE bss_cril_root_key( ).
        iv_parent->get_key( IMPORTING es_key = cr_id ).
        APPEND cr_id TO ct_child_keys.
      WHEN OTHERS.
    ENDCASE.
  ENDMETHOD.
ENDCLASS.
```

2.2.3. Feeder Class for Additional UIBB ZCL_USMD_CR_GUIBB_REQ_BUS_AREA

The Feeder Class for Requesting Business Area (ZCL_USMD_CR_GUIBB_REQ_BUS_AREA) class is a redefinition of class CL_GUIBB_BOL_FORM.



You must redefine methods IF_FPM_GUIBB_FORM~GET_DATA, IF_FPM_GUIBB_FORM~GET_DEFINITION and GET_ATTR_VALUE_SET (for F4 help).

Source code of class ZCL_USMD_CR_GUIBB_REQ_BUS_AREA

```
CLASS zcl_usmd_cr_guibb_req_bus_area DEFINITION
  PUBLIC
  INHERITING FROM cl_guibb_bol_form
  FINAL
  CREATE PUBLIC .

  PUBLIC SECTION.
    METHODS if_fpm_guibb_form~get_data REDEFINITION.
    METHODS if_fpm_guibb_form~get_definition REDEFINITION.
  PROTECTED SECTION.
    METHODS get_attr_value_set REDEFINITION.
  PRIVATE SECTION.
ENDCLASS.

CLASS zcl_usmd_cr_guibb_req_bus_area IMPLEMENTATION.
  METHOD if_fpm_guibb_form~get_data.
    super->if_fpm_guibb_form~get_data(
      EXPORTING
        io_event = io_event "ID of the FPM Event
        iv_raised_by_own_ui = iv_raised_by_own_ui "Event was triggered by own UI
        it_selected_fields = it_selected_fields "Selected(Used) Fields
        iv_edit_mode = iv_edit_mode "FPM: Edit Mode
        io_extended_ctrl = io_extended_ctrl "Extended PBO control
      IMPORTING
        et_messages = et_messages "FPMGB Messages (T100 & [.underline]#Plaintext#)
        ev_data_changed = ev_data_changed "Boolean Variable (X=True, -=false, space=unknown)
        ev_field_usage_changed = ev_field_usage_changed "Boolean Variable (X=True, -=false, space=unknown)
        ev_action_usage_changed = ev_action_usage_changed "Boolean Variable (X=True, -=False, Space=Unknown)
      CHANGING
        cs_data = cs_data
        ct_field_usage = ct_field_usage "Field Usage
        ct_action_usage = ct_action_usage "Action Definition
    ).
  ENDMETHOD.

  METHOD if_fpm_guibb_form~get_definition.
    super->if_fpm_guibb_form~get_definition(
      IMPORTING
        es_message = es_message
        eo_field_catalog = eo_field_catalog
        et_field_description = et_field_description
        et_action_definition = et_action_definition
        et_special_groups = et_special_groups
        ev_additional_error_info = ev_additional_error_info
        et_dnd_definition = et_dnd_definition ).

    " Enter empty field for [.underline]#dropdown# list box for "Business Area"
    READ TABLE et_field_description ASSIGNING FIELD-SYMBOL(<field_description>) WITH KEY name = 'BUS_AREA'.
```

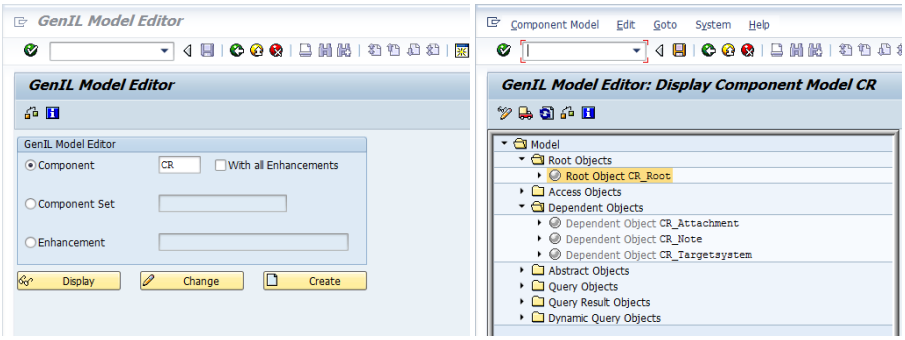
```
    IF sy-subrc = 0 AND <field_description> IS ASSIGNED.
      <field_description>-is_nullable = abap_true.
    ENDIF.
  ENDMETHOD.

  METHOD get_attr_value_set.
    SELECT gsber gsber FROM tgsb INTO TABLE et_value_set. "#EC CI_GENBUFF
    LOOP AT et_value_set ASSIGNING FIELD-SYMBOL(<value_set>).
      SELECT SINGLE gtext FROM tgsbt INTO <value_set>-text
      WHERE spras = sy-langu AND gsber = <value_set>-value.
    ENDLOOP.
  ENDMETHOD.

ENDCLASS.
```

2.3. Create Enhancement of GenIL Data Model of Change Request (GenIL Model CR)

The Change Request is represented by the GenIL component CR in the SAP system. You can have a look at this component in the transaction GENIL_MODEL_BROWSER by entering the component CR and choosing *Display*.

The Change Request itself is represented by a *Root Object* (CR_Root) and several *Dependent Objects*:

- CR_Root:
  Contains the elementary attributes of a change request like ID, Description, Status, CR Type.
- CR_Attachment:
  Represents the attachments (of type file as well as of type link) of a change request
- CR_Note:
  Represents the notes that an end user can attach to a change request.
- CR_Targetsystem:
  Represents the target systems that can be maintained for a change request.

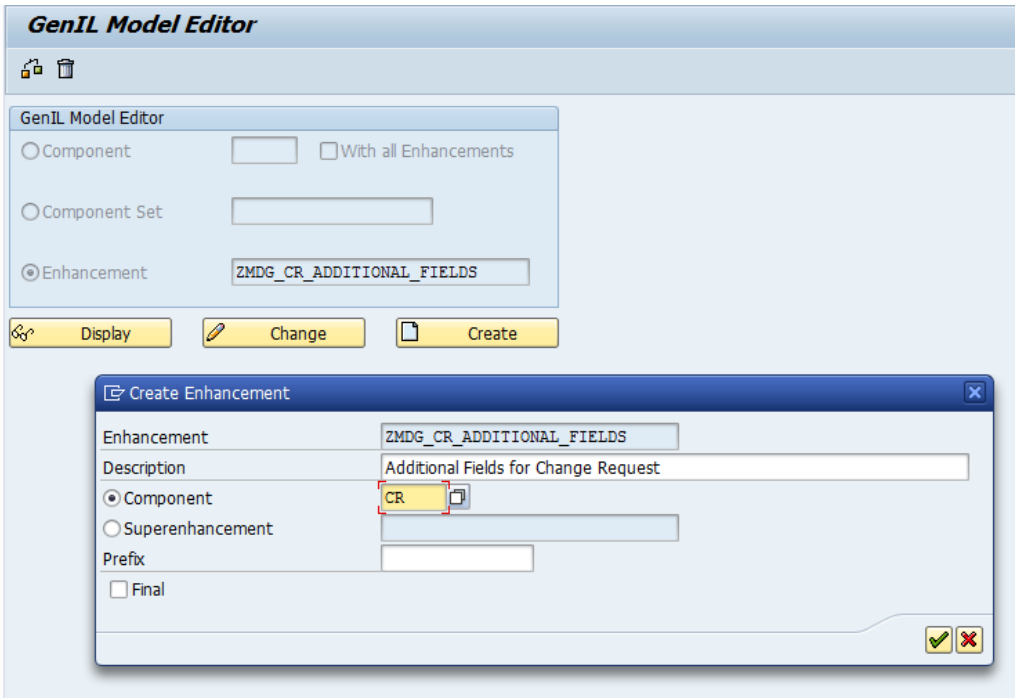


The GenIL modeling approach ensures that an enhancement of the CR component with data that is customer specific, such as the addition of a few additional attributes to the change request, results in the creation of an additional *Dependent Object* in the change request component.

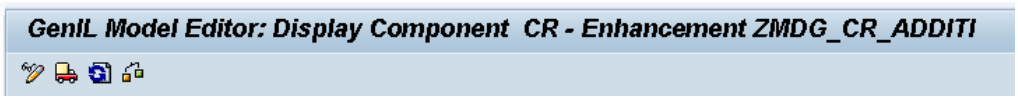
The following steps explain how to create as well as integrate a new *Dependent Object* for the CR component using an enhancement. This enhancement will add the additional process control parameter *Requesting Business Unit* to the change request.

2.3.1. Create Enhancement ZMDG_CR_ADDITIONAL_FIELDS

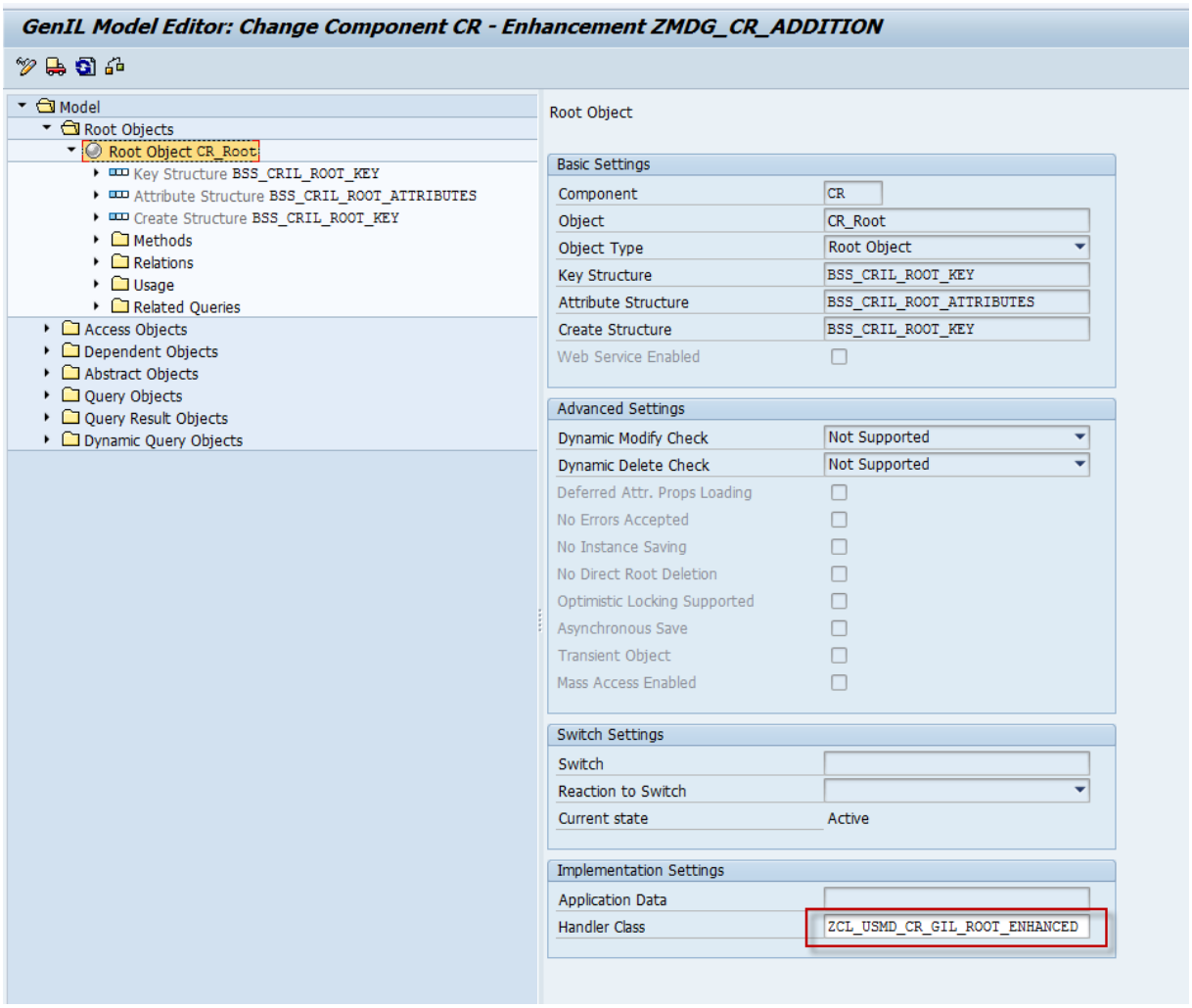
Call transaction GENIL_MODEL_BROWSER, choose the radio button *Enhancement*, enter a corresponding name and choose *Create*. In the dialog box, enter a suitable description, as well as the CR component for the component that needs to be enhanced. Finally, enter a suitable package for the enhancement.



After completing these steps, the CR component is displayed again, this time with a new header that reflects the context of the newly-created enhancement.

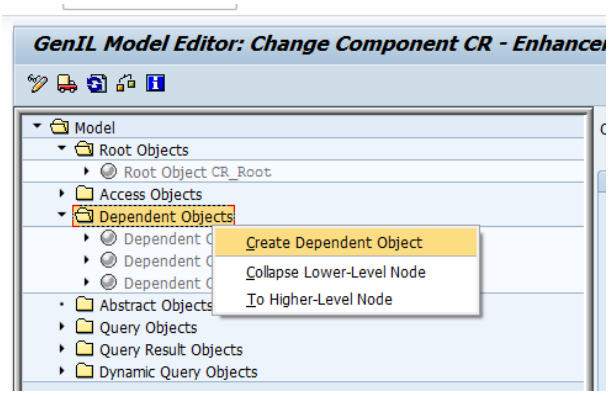


In Root Object CR_ROOT, specify a *Handler Class* of ZCL_USMD_CR_GIL_ROOT_ENHANCED.



2.3.2. Create Dependent Object ZCR_REQ_BUSINESS_AREA

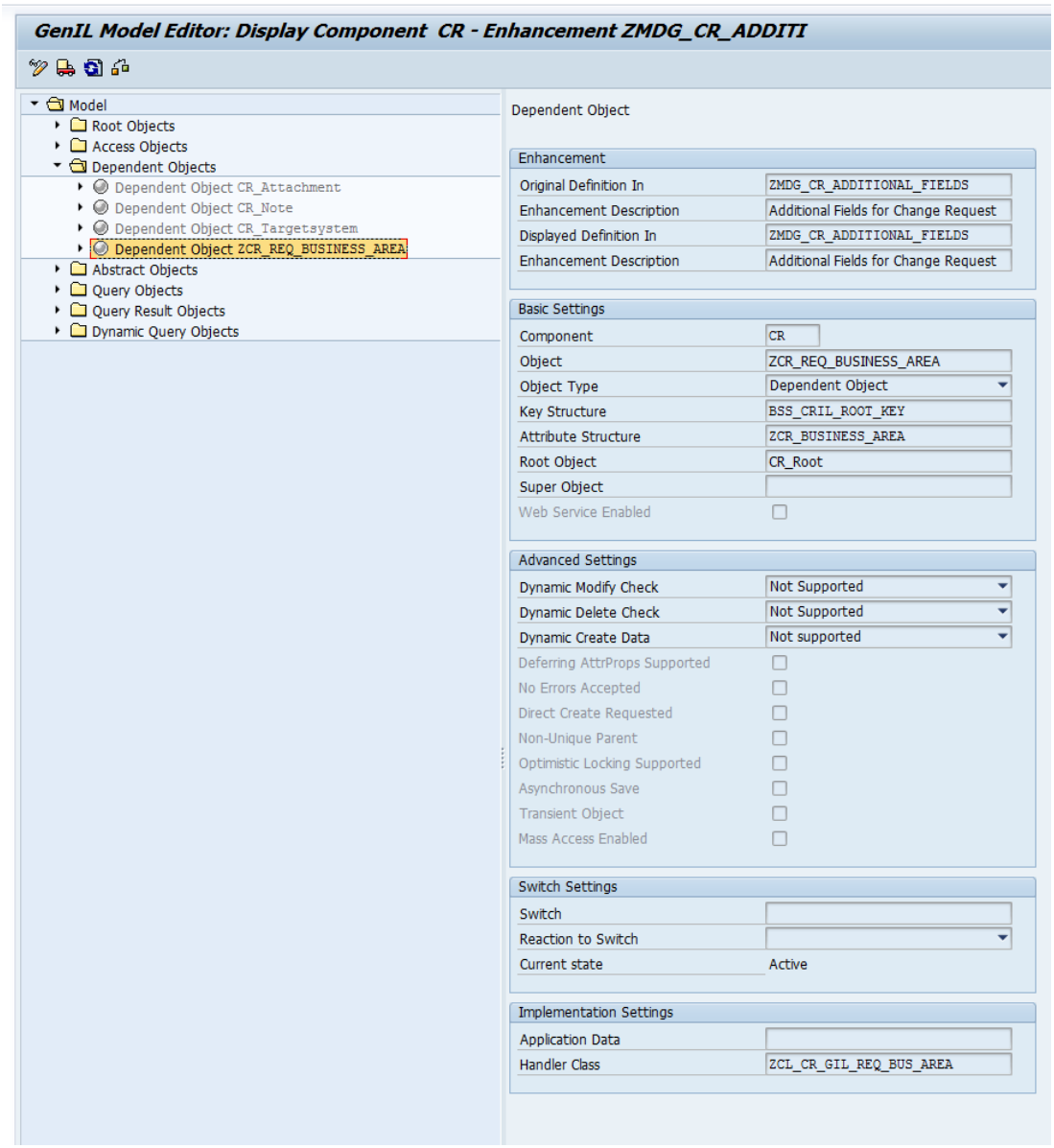
In order to add the additional attribute to the CR component, select the dependent objects, and choose *Create Dependent Object* from the shortcut menu. Enter ZCR_REQ_BUSINESS_AREA as name for the new dependent object.



Maintain the following fields:

- Key Structure: BSS_CRIL_ROOT_KEY
  The key structure represents the key information of the additional dataset. In this example, at least the CR ID must be part of the key in order to create the relationship to the change request itself. You can include the CR ID in the key by including structure BSS_CRIL_ROOT_KEY into the key structure. Additional key fields are only required if the new dependent object can have multiple occurrences.

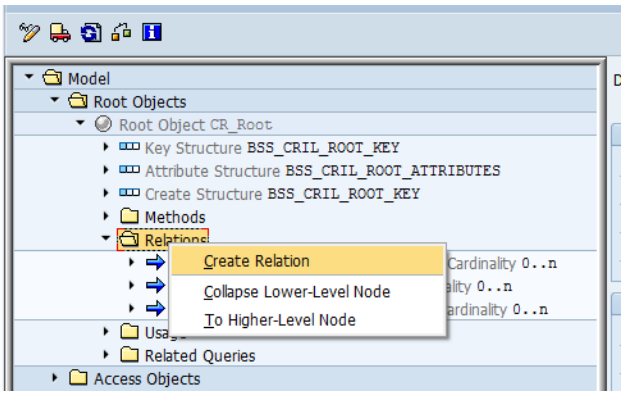
- Attribute structure: ZCR_BUSINESS_AREA
  The attribute structure represents the real attributes that you add to the change request.
  IMPORTANT: To be able to display the key elements of the new dependent object on the user interface, you must add the key elements redundantly to the attribute structure.
- Root Object: CR_Root
  This field indicates to which root object the new dependent object belongs. Note that this field is case sensitive.
- Handler Class: ZCL_CR_GIL_REQ_BUS_AREA
  The handler class takes care of all data operations on the new attributes such as Create, Read, Change, Delete as well as additional information (for example, field properties.)



2.3.3. Create Relation ZCR_ReqBusAreaRel

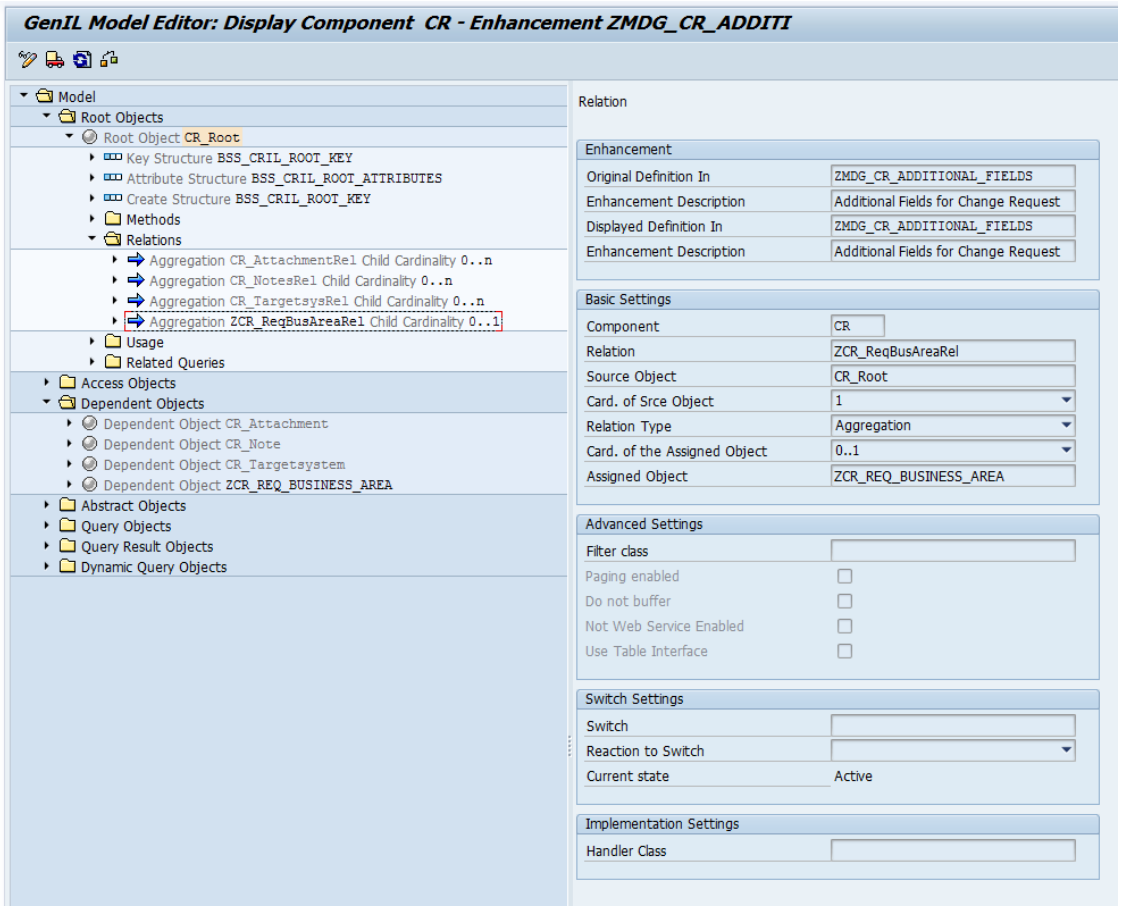
To establish the connection between the root object CR_Root object and the dependent object

ZCR_REQ_BUSINESS_AREA, open the sub-tree of CR_Root and create a new relation:



The relation defines the type of relation that is built up between the two objects as well as the corresponding cardinality. In our example, we create an aggregation with cardinality 0..1.

Additionally, you must maintain the assigned object CR_Root. (Later, we need the name of the relation in the UI configuration).

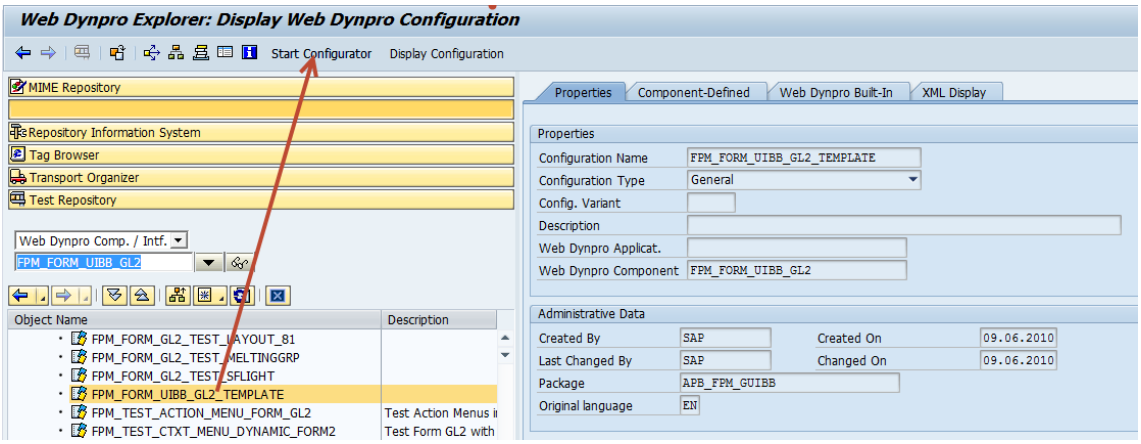


When you have implemented changes, perform a check and make sure that no errors or warnings occur.

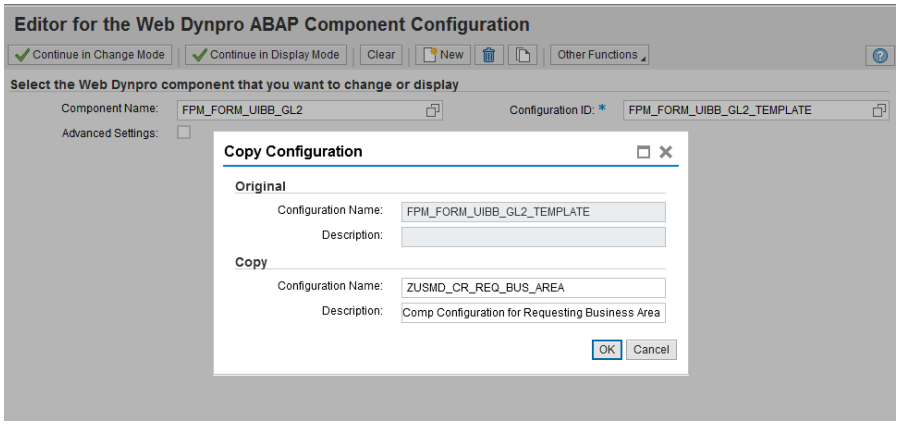
2.4. Creation of New Form UIBB for Additional Data

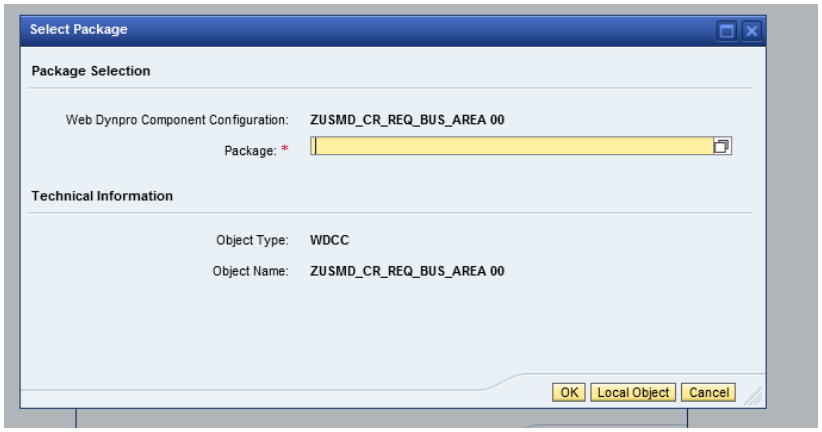
2.4.1. Create Workbench Component Configuration

Call the Web Dynpro Application FPM_FORM_UIBB_GL2 using the Object Navigator (transaction code SE80) and start the component configuration FPM_FORM_UIBB_GL2_TEMPLATE as shown below:



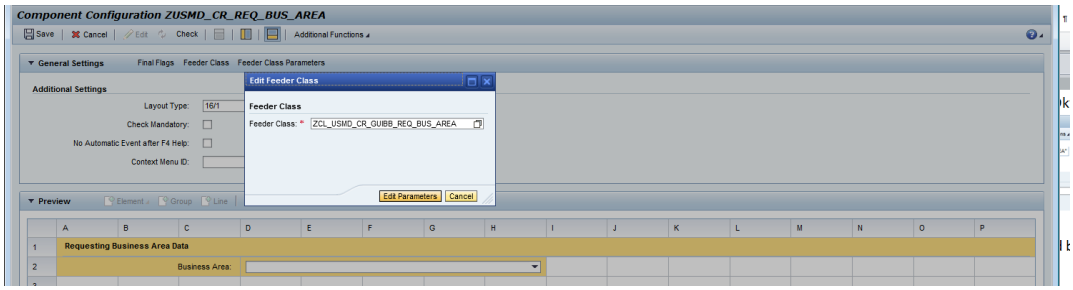
The configuration starts in a web browser. Copy the template FPM_FORM_UIBB_GL2_TEMPLATE to a new configuration (ZUSMD_CR_REQ_BUS_AREA) and assign it to an appropriate package.

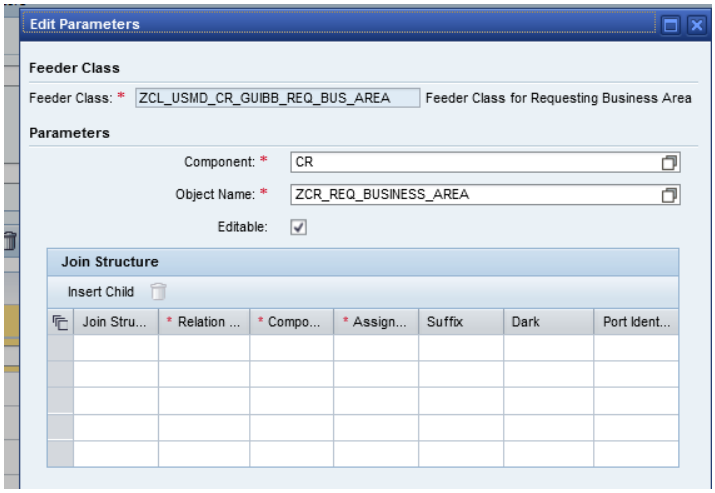




Confirm the copy task by choosing *OK*.

Enable editing of the new configuration by clicking the *Continue in Change Mode* button. In the *General Settings* area, specify a *Feeder Class* and choose the *Edit Parameters* button.

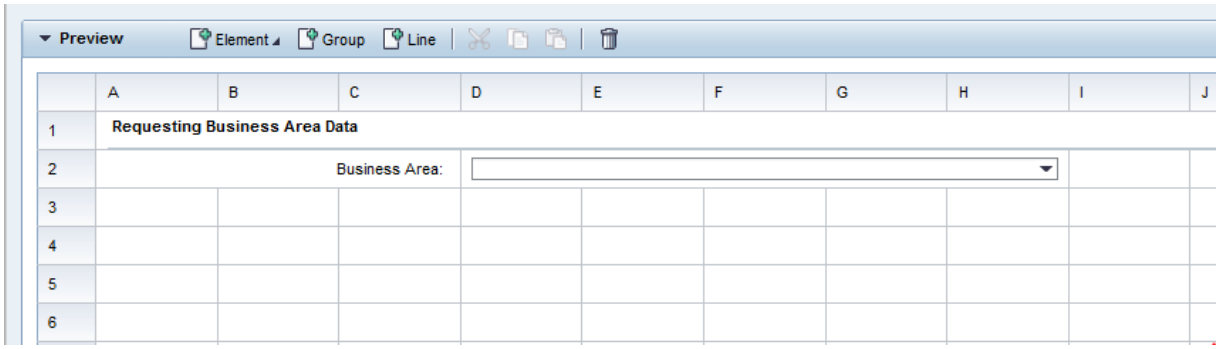




For Feeder Class ZCL_USMD_CR_GUIBB_REQ_BUS_AREA, enter the following parameters

- Component: CR
- Object Name: ZCR_REQ_BUSINESS_AREA (Dependent Object in GenIL Model)

If required, you can adjust the layout can be adjusted as shown in the screenshot.



Save the configuration.

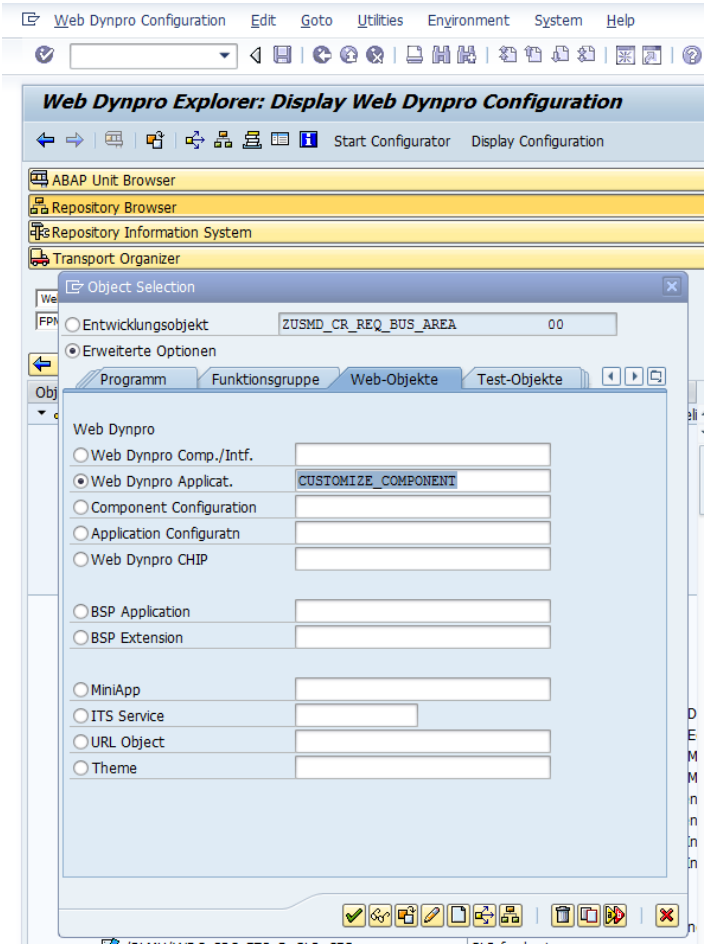
2.5. Integration of new UIBB in CR Tabbed UIBB

In this scenario, you place the additional field *Requesting Business Area* in the *General Data* tab page of the Change Request UIBB.

The following activities are client-dependent and must be performed in the MDG-client.

Add new UIBB

To access the UIBB Configuration Mode, start the Object Navigator and press *Shift-F5*. In the Object Selection popup, select the Web-Objects tab and enter the Web Dynpro Application: CUSTOMIZE_COMPONENT.



Start the configurator in 'Test/Execute' mode. If the configuration is new, continue with NEW, otherwise continue with *Continue in Change Mode*. For new configurations, you must first create the customizing. Continue in Change Mode with the above shown entries. Select Tab: *General* and add a Form Component UIBB. Enter ZUSMD_CR_REQ_BUS_AREA as the Configuration Name.

Add Wire Schema

Select the new UIBB and switch to tab 'Wire Schema'. Click the "Wire" button and maintain the following attributes:

- Component: FPM_FORM_UIBB_GL2
- Configuration Name: ZUSMD_CR_REQ_BUS_AREA
- Source Component: FPM_FORM_UIBB_GL2
- Source Config Name: USMD_CR_MASTER
- Port Type: Lead Selection
- Port Identitifier: STANDARD
- Connector Class: CL_FPM_CONNECTOR_BOL_RELATION
- Relation Name: ZCR_ReqBusAreaRel (Attention: Relation Name is case sensitive)

Now the CR UIBB contains the new customer specific field *Requesting Business Area*. When the change request is saved the entered value for the *Requesting Business Area* together with the Change Request ID is stored to data base table ZUSMD_CR_BUSAREA.

2.6. Extended Business Scenario

The additional parameter *Requesting Business Area* is not relevant for all users. Its visibility shall depend on the change request type. In our example the parameter is visible if change request types T1C01 or T1C02 are used.

Extension of Feeder Class for Additional UIBB: ZCL_USMD_CR_GUIBB_REQ_BUS_AREA

Implement a redefinition of method CHECK_FIELD_USAGE_SINGLE.

```
[...]
  PROTECTED SECTION.
    METHODS get_attr_value_set REDEFINITION.
    METHODS check_action_usage_single REDEFINITION. "<<<<<<<<<<<<<
  PRIVATE SECTION.
ENDCLASS.

[...]
  METHOD check_action_usage_single.
    DATA(context) = cl_usmd_app_context=>get_context( ).
    context->get_attributes( IMPORTING ev_crequest_type = DATA(crequest_type) ).
    IF request_type = 'T1C01' OR request_type = 'T1C02'.
      cs_action_usage-visible = '02'. "01=none, 02=visible
    ELSE.
      cs_action_usage-visible = '01'. "01=none, 02=visible
    ENDIF.
  ENDMETHOD.
[...]
```

Source code for class ZCL_USMD_CR_GUIBB_REQ_BUS_AREA

Additional remarks:

- The above shown pattern can be used to evaluate other parameters returned by the method IF_USMD_APP_CONTEXT~GET_ATTRIBUTES (e.g. EV_PROCESS).
- Alternatively the change request type can be evaluated in a redefinition of method GET_ATTRIBUTE_PROPERTIES of class ZCL_USMD_CR_GIL_ROOT_ENHANCED. This approach means that the evaluation is valid for all UIBBs using this GENIL MODEL extension.
- If you use one GenIL Model Extension together with different UIBBs and corresponding feeder classes, the evaluation of parameters can be different for different feeder classes.