PUBLIC

How-To Guide
SAP Marketing Cloud
Document Version: 2.0 - 2019-12-18

Initial Load of Marketing Permissions from SAP Customer Relationship Management to SAP Marketing Cloud

Document History

Document Version	Description
2.0	Second official release of this guide

PUBLIC
© 2019 SAP SE or an SAP affiliate company. All rights reserved.

Table of Contents

1 Purpose of this Document 4

2 Standard Integration Scenario 4
- 2.1 Overview 4
- 2.2 Common Objects 5
- 2.3 BAdI CRMXIF_PARTNER_MAP (Marketing Permissions for Accounts) 11
- 2.4 BAdI CRMXIF_PARTNER_R_MAP (Marketing Permissions for Contacts) 21
- 2.5 Activate BAdI Implementations 31

3 Extended Integration Scenario for Marketing Areas Determination 35
- 3.1 Overview 35
- 3.2 Prerequisite 36
- 3.3 Maintain Marketing Organization 36
- 3.4 Maintain Employee Responsible in Master Data 40
- 3.5 Dictionary Objects 43
- 3.6 Maintain the Mapping Table Z or Y<MKTAREA_MAP> 46
- 3.7 IDoc Generation 48
- 3.8 Common Objects 52
- 3.9 BAdI CRMXIF_PARTNER_MAP (Marketing Permissions for Accounts) 55
- 3.10 BAdI CRMXIF_PARTNER_R_MAP (Marketing Permissions for Contacts) 59
- 3.11 Maintain Mapping Table 64
- 3.12 Create Logical System, RFC Destination, Port, Partner Profile, Site 66
- 3.13 Extending the Permission iFlow in SAP Cloud Platform Integration (CPI) 67

© 2019 SAP SE or an SAP affiliate company. All rights reserved.

1 Purpose of this Document

This document is intended as a how-to-guide for customers and partners who are facing the requirement to load marketing permissions from SAP Customer Relationship Management (SAP CRM) to SAP Marketing Cloud by using a standard integration scenario delivered by SAP or by extending an integration scenario using marketing area determination in SAP CRM.

In the standard integration scenario, you need to implement the Business Add-Ins (BAdI) CRMXIF_PARTNER_MAP and CRMXIF_PARTNER_R_MAP.

In the extended integration scenario using marketing area determination, you have to extend the IDoc's structure, IFlows, and BAdI implementation for the BAdIs CRMXIF_PARTNER_MAP and CRMXIF_PARTNER_R_MAP.

As a sample use case, this guide shows how to implement the two integration scenarios, standard integration scenario (without marketing area determination in SAP CRM) and extended integration scenario (with marketing area determination in SAP CRM).

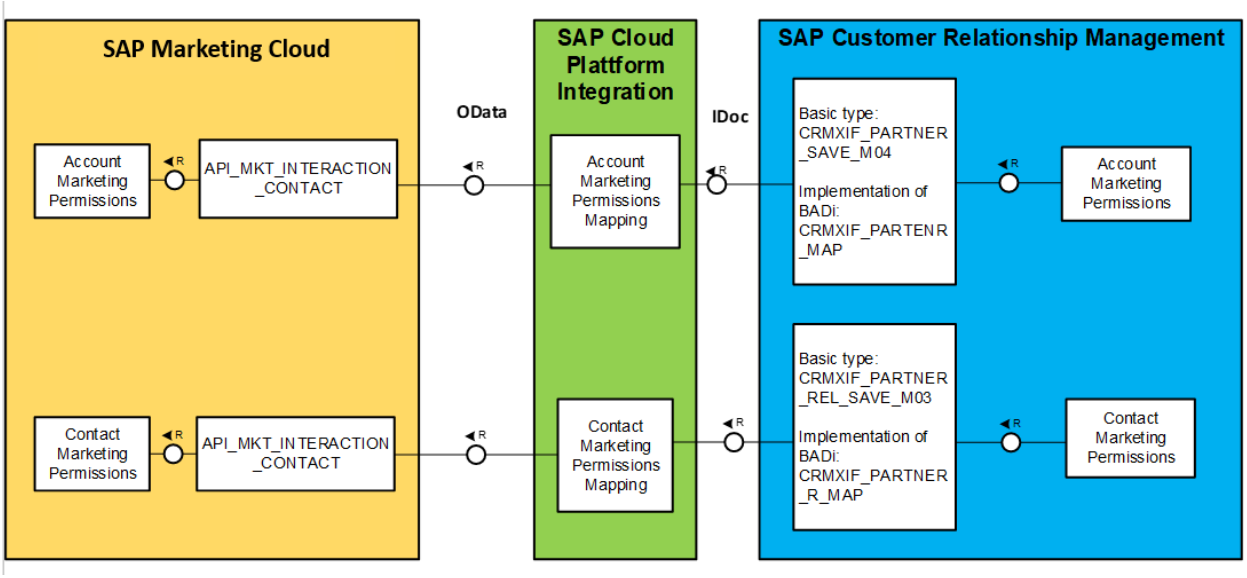
All required functionality is available with release 1805.

2 Standard Integration Scenario

2.1 Overview

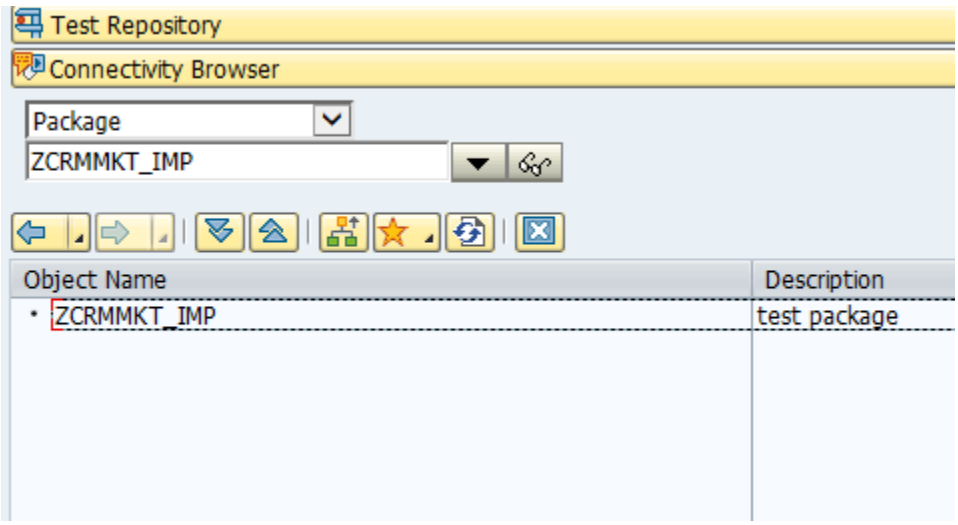
- In the standard integration scenario, marketing permissions from SAP CRM will be initially loaded to SAP Marketing Cloud without marketing area information.
- Therefore, in SAP Marketing Cloud the use of marketing areas in the execution of campaigns may not be activated.
- Only explicit marketing permissions are supported and considered during initial load of permissions from SAP CRM to SAP Marketing Cloud.
- BAdI implementation must be done manually.
- Only the following channels are supported and allowed to be loaded into Marketing Cloud solution from SAP CRM:
  Phone (TEL), Fax (FAX), Email (INT), SMS (PAG)
- If communication detail is a mobile phone number a new communication channel (TE2) will be generated in Outbound IDoc.
- Only marketing permissions of contact person are loaded, while marketing permissions of individual account are not considered and filtered out, if a business partner is both individual account and contact person
- Communication details will be determined from Business Partner Address, if they are not maintained and empty
- If communication details are empty and cannot be determined from Business Partner Address because communications are not maintained, relevant permissions will not be transferred.
- The role *Account* (CRMACC) must be maintained in master data of accounts (corporate, individual) for initial load.
- The flag *Activate Marketing Area for Campaign Execution in SAP Marketing* may not be activated, otherwise you have to apply the extended integration scenario with marketing areas determination in SAP Customer Relationship Management.

© 2019 SAP SE or an SAP affiliate company. All rights reserved.



2.2 Common Objects

1. Create a new development package Z or Y<CRMMKT_IMP> in customer namespace.

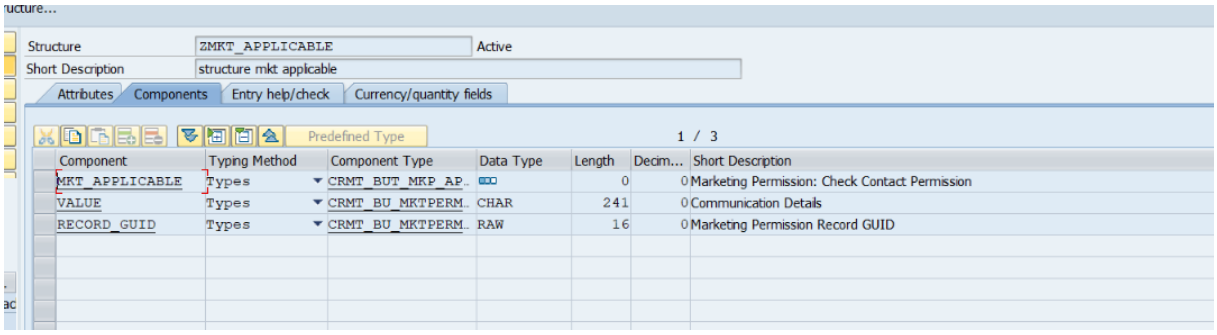


2. Create dictionary objects.

   Structure: Z_or Y<MKT_APPLICABLE>

Component	Component Type
MKT_APPLICABLE	CRMT_BUT_MKP_APPLICABLE
VALUE	CRMT_BU_MKTPERM_COM
RECORD_GUID	CRMT_BU_MKTPERM_GUID

© 2019 SAP SE or an SAP affiliate company. All rights reserved.



Structure: Z or Y<MKT_VALUES>

Component	Component Type
VALUE	CRMT_BU_MKTPERM_COM
STD	CRMT_BOOLEAN
R_3_USER	AD_FLGMOB

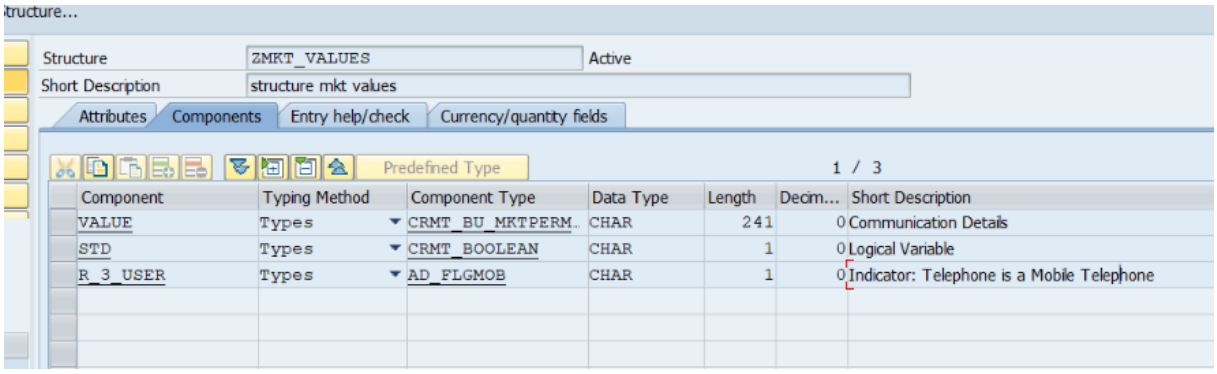


Table Type: Z or Y <MKT_APPLICABLE_T>
Line type: Z or Y<MKT_APPLICABLE>

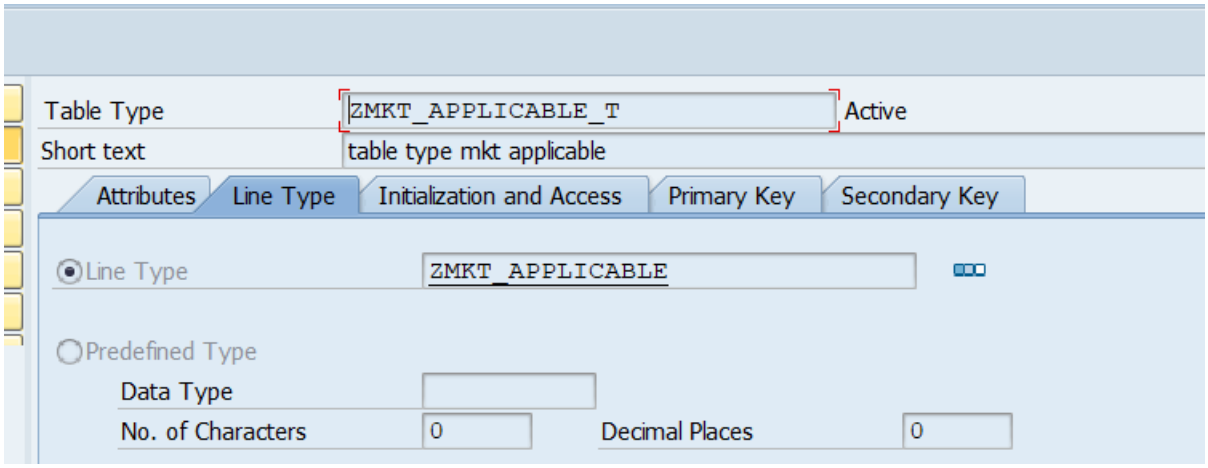
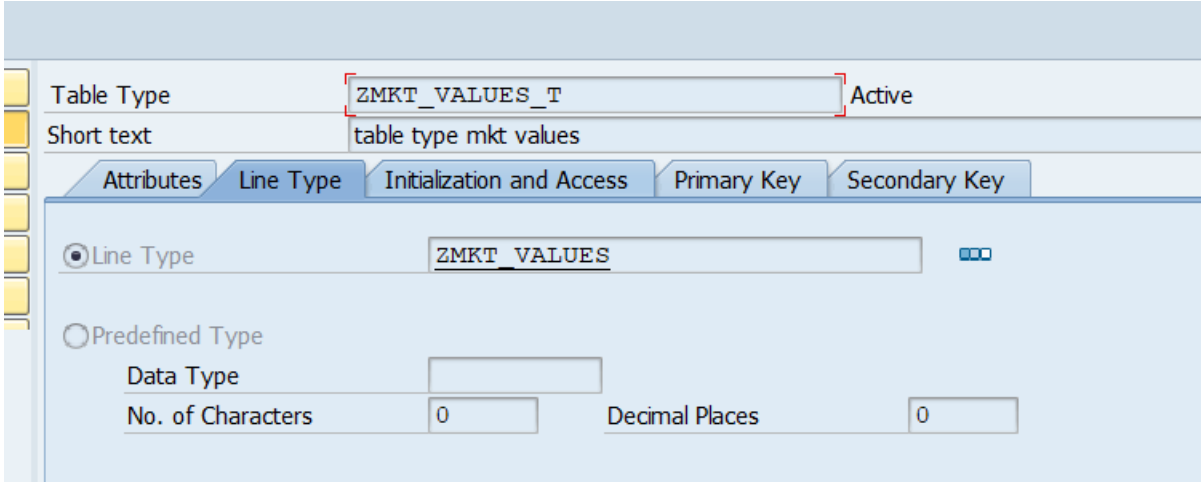
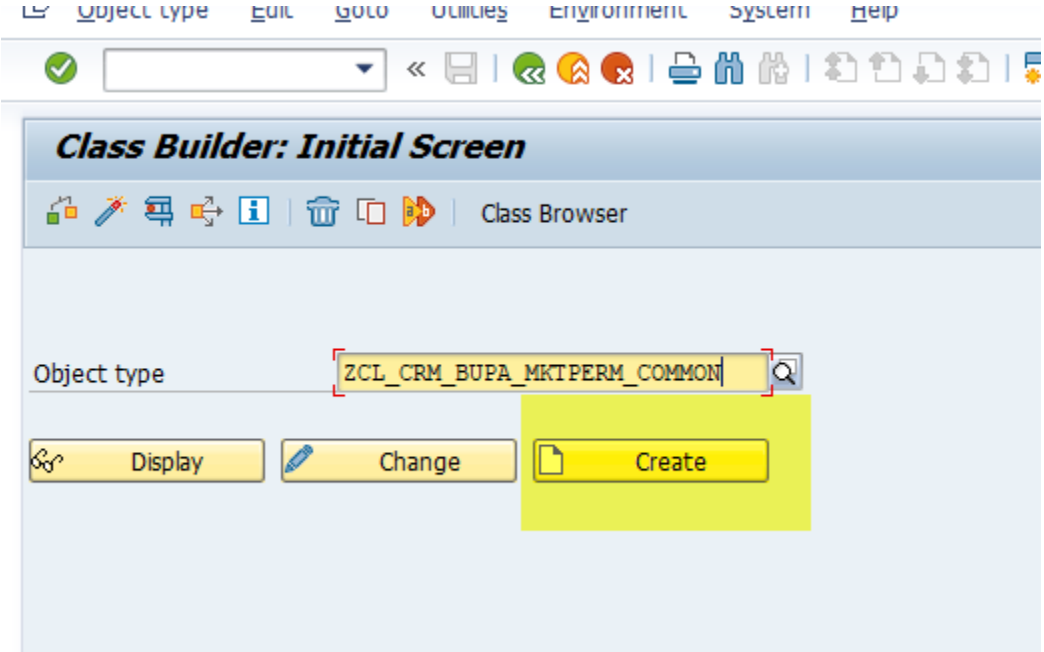


Table type: Z or Y<MKT_VALUES_T>
Line type: Z or Y<MKT_VALUES>

© 2019 SAP SE or an SAP affiliate company. All rights reserved.

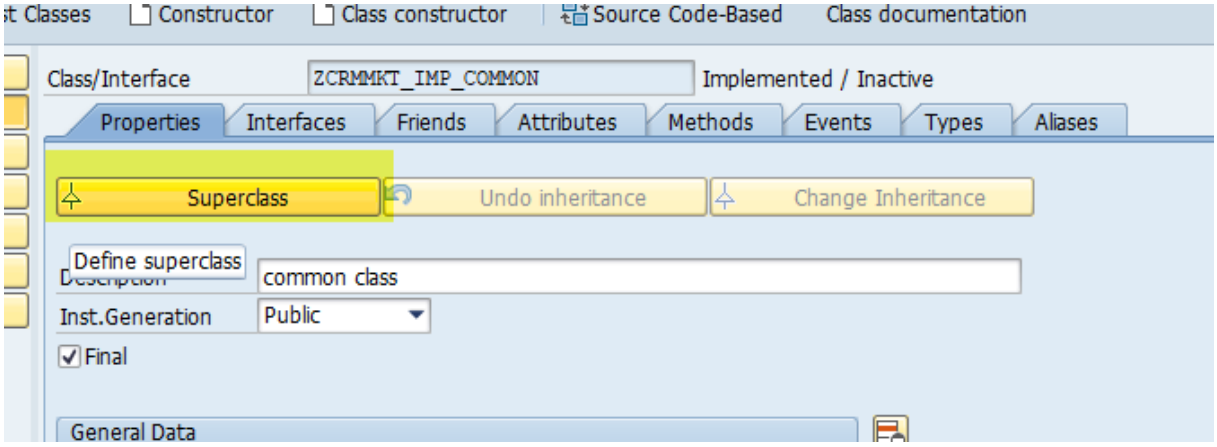


3. Create class object.
   Start transaction SE24.
   Enter a class name Z or Y<CL_CRM_BUPA_MKTPERM_COMMON> and click on create.

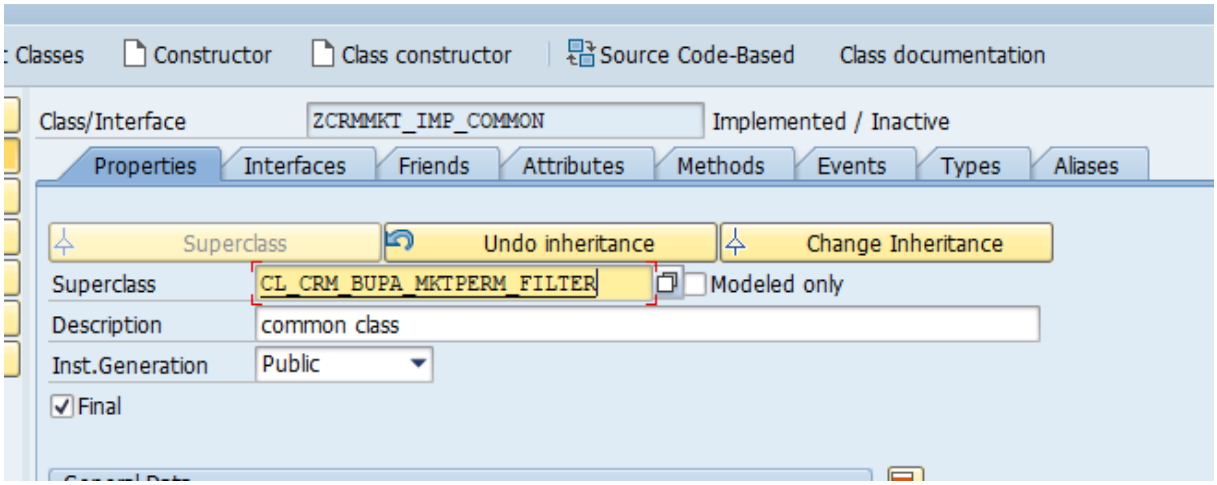


© 2019 SAP SE or an SAP affiliate company. All rights reserved.

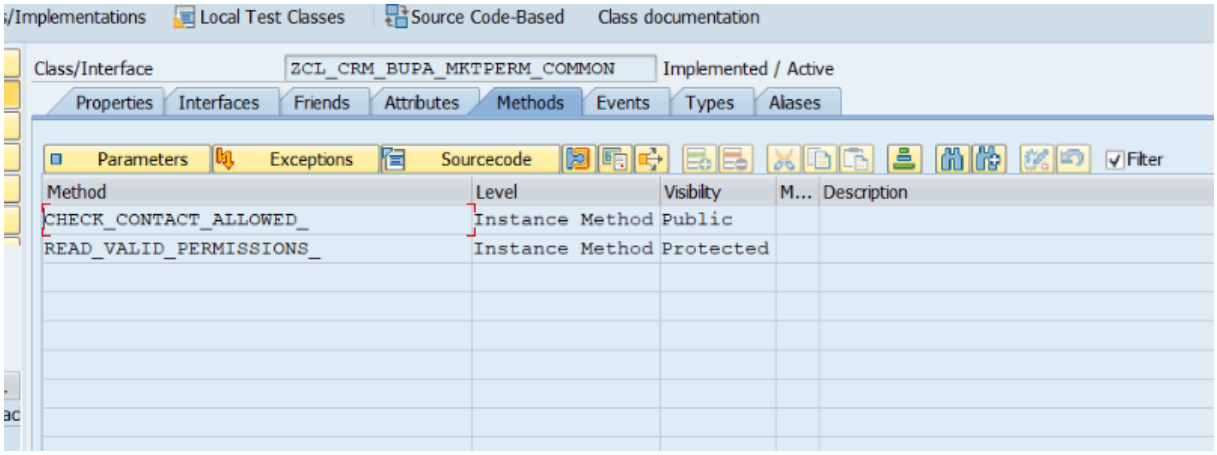
4. Go to the tab *Properties* and click on *Superclass*.



5. Enter some details on the tab.
   Superclass = CL_CRM_BUPA_MKTPERM_FILTER
   Description = common class
   Inst. Generation = Public

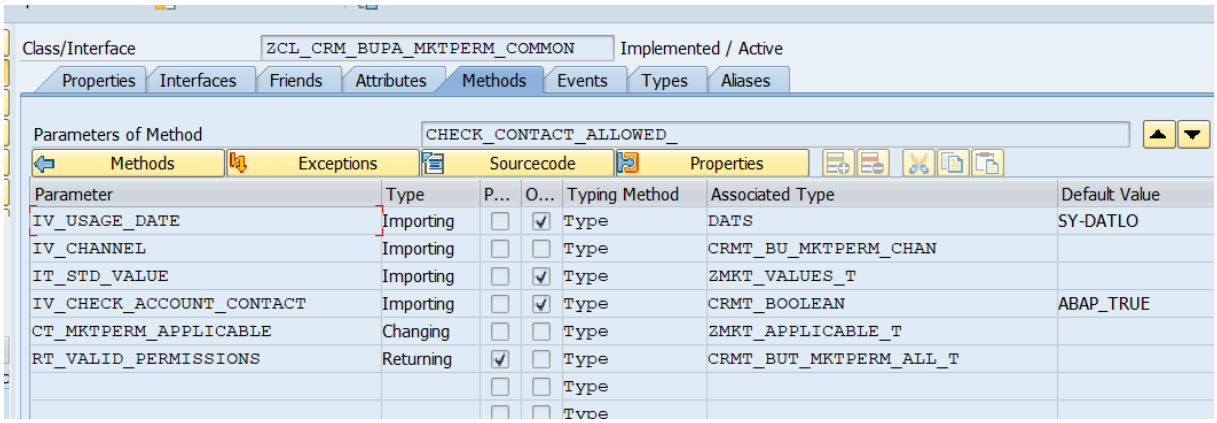


6. Save your entries.
   Go to the tab *Methods* and add two new methods as follows:
   CHECK_CONTACT_ALLOWED: Instance method, public.
   READ_VALID_PERMISSIONS: Instance method, protested.

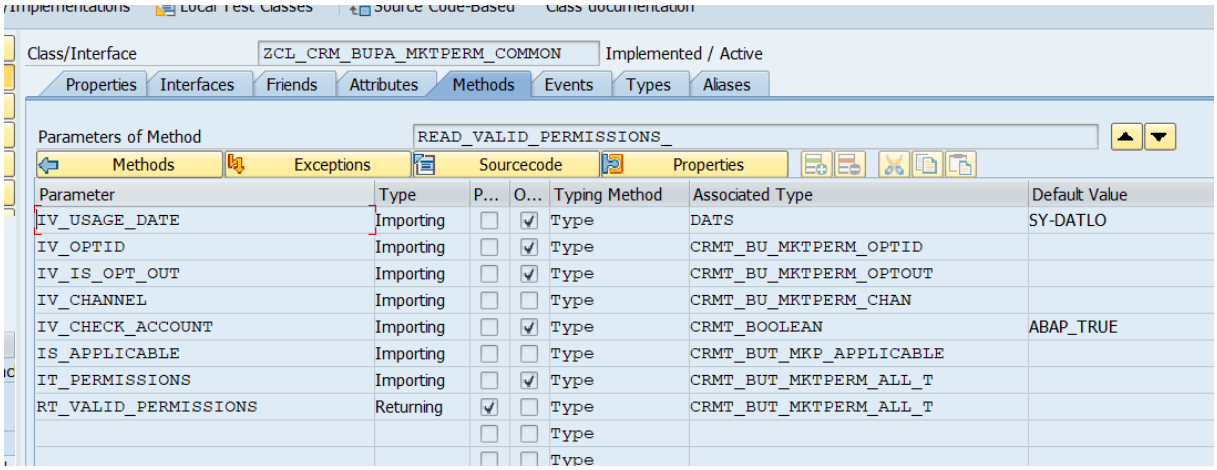


© 2019 SAP SE or an SAP affiliate company. All rights reserved.

7. Select the method CHECK_CONTACT_ALLOWED_, click on *Parameters* , and enter the parameters as follows:



8. Select the method READ_VALID_PERMISSIONS_, click on *Parameters,* and enter the parameters as follows:



9. Implement the method CHECK_CONTACT_ALLOWED_.

```
method CHECK_CONTACT_ALLOWED_.
CONSTANTS lc_initial_guid   TYPE bu_partner_guid VALUE '0'.
DATA: ls_contact_allowed            TYPE CRMT_BUT_MKP_CONTACT_ALLOWED,
    ls_mktperm_applicable TYPE CRMT_BUT_MKP_APPLICABLE,
    ls_mkt_applicable             TYPE ZMKT_APPLICABLE,
    lt_valid_permissions     TYPE CRMT_BUT_MKTPERM_ALL_T,
    ls_valid_permissions     TYPE CRMT_BUT_MKTPERM_ALL,
    ls_compared_permissions TYPE CRMT_BUT_MKTPERM_ALL,
    lv_return                                TYPE CRMT_BOOLEAN,
    lt_permissions                  TYPE CRMT_BUT_MKTPERM_ALL_T,
    ls_permission                   like LINE OF lt_permissions,
    ls_std_value                    TYPE CRMT_BU_MKTPERM_COM.

DATA:lo_mkp_filter TYPE REF TO cl_crm_bupa_mktperm_filter,
   lt_valid_cont_perm TYPE crmt_but_mktperm_all_t.

LS_MKTPERM_APPLICABLE-CONTPGUID = GV_CONTPGUID.
LS_MKTPERM_APPLICABLE-PARTNERGUID = GV_PARTNERGUID.

*filter valid permissions
CALL METHOD me->READ_VALID_PERMISSIONS_
  EXPORTING
    IV_USAGE_DATE      = IV_USAGE_DATE
    IV_CHANNEL         = IV_CHANNEL
    IV_CHECK_ACCOUNT   = IV_CHECK_ACCOUNT_CONTACT
    IS_APPLICABLE      = LS_MKTPERM_APPLICABLE
```


PUBLIC
© 2019 SAP SE or an SAP affiliate company. All rights reserved.

```
    RECEIVING
     RT_VALID_PERMISSIONS = LT_VALID_PERMISSIONS.

  *evaluate valid permissions
   LOOP AT LT_VALID_PERMISSIONS INTO LS_VALID_PERMISSIONS.
     ls_contact_allowed-CHANNEL = iv_channel.
     ls_contact_allowed-CONTPGUID = LS_VALID_PERMISSIONS-CONTPGUID.
     ls_contact_allowed-PARTNERGUID = LS_VALID_PERMISSIONS-
                                                                                PARTNERGUID.
            ls_contact_allowed-VALUE = ls_valid_permissions-VALUE.
         * evaluate permissions
          lv_return = me->evaluate_permission(
          iv_usage_date      = iv_usage_date
          IT_VALID_PERMISSIONS = LT_VALID_PERMISSIONS
          is_contact_allowed   = ls_contact_allowed ).
          ls_contact_allowed-CONTACT_ALLOWED = lv_return.
          clear LS_MKT_APPLICABLE.
          MOVE-CORRESPONDING ls_contact_allowed TO LS_MKT_APPLICABLE-
                                                                              MKT_APPLICABLE.
    LS_MKT_APPLICABLE-VALUE = ls_contact_allowed-VALUE.
     LS_MKT_APPLICABLE-RECORD_GUID = LS_VALID_PERMISSIONS-RECORD_GUID.
     append LS_MKT_APPLICABLE to CT_MKTPERM_APPLICABLE.
     ENDLOOP.
     RT_VALID_PERMISSIONS = LT_VALID_PERMISSIONS.
endmethod.
```

10. Implement the method READ_VALID_PERMISSIONS_.

```
  method READ_VALID_PERMISSIONS_.
    CONSTANTS lc_initial_guid  TYPE bu_partner_guid VALUE '0'.
    DATA:
    lt_valid_permissions      TYPE CRMT_BUT_MKTPERM_ALL_T,
    lt_valid_cont_perm            TYPE CRMT_BUT_MKTPERM_ALL_T,
    ls_valid_permissions      TYPE CRMT_BUT_MKTPERM_ALL.

*  evaluate valid marketing permissions
*  valid entries for accounts:
    IF IT_PERMISSIONS IS SUPPLIED.
    me->filter_permission(
          EXPORTING iv_usage_date = iv_usage_date
         IT_PERMISSIONS = it_permissions
          IMPORTING et_valid_permissions = lt_valid_permissions ).
    ELSE.
    me->filter_permission(
          EXPORTING iv_usage_date      = iv_usage_date
          IMPORTING et_valid_permissions = lt_valid_permissions ).
    ENDIF.
    IF NOT IS_APPLICABLE-CONTPGUID IS INITIAL.
*   add valid entries for contacts:
    me->filter_contact_permission(
          EXPORTING iv_usage_date      = iv_usage_date
          IMPORTING et_valid_permissions = lt_valid_cont_perm ).
    APPEND LINES OF lt_valid_cont_perm TO lt_valid_permissions.
          DELETE LT_VALID_PERMISSIONS
                     WHERE PARTNERGUID = GV_PARTNERGUID   AND
                    CONTPGUID   = lc_initial_guid   AND
                    CHANNEL     = IV_CHANNEL       AND
                    PERMISSION  = cl_crm_bupa_mktperm_tools=>GC_GIVEN.
          ENDIF.

    DELETE LT_VALID_PERMISSIONS WHERE CHANNEL <> IV_CHANNEL.
* delete redundant account permission
    LOOP AT LT_VALID_PERMISSIONS INTO LS_VALID_PERMISSIONS WHERE
   PARTNERGUID = GV_PARTNERGUID   AND
   CONTPGUID   = lc_initial_guid   AND
   CHANNEL     = IV_CHANNEL       AND
   PERMISSION  = cl_crm_bupa_mktperm_tools=>GC_GIVEN.
    READ TABLE LT_VALID_PERMISSIONS WITH KEY
```

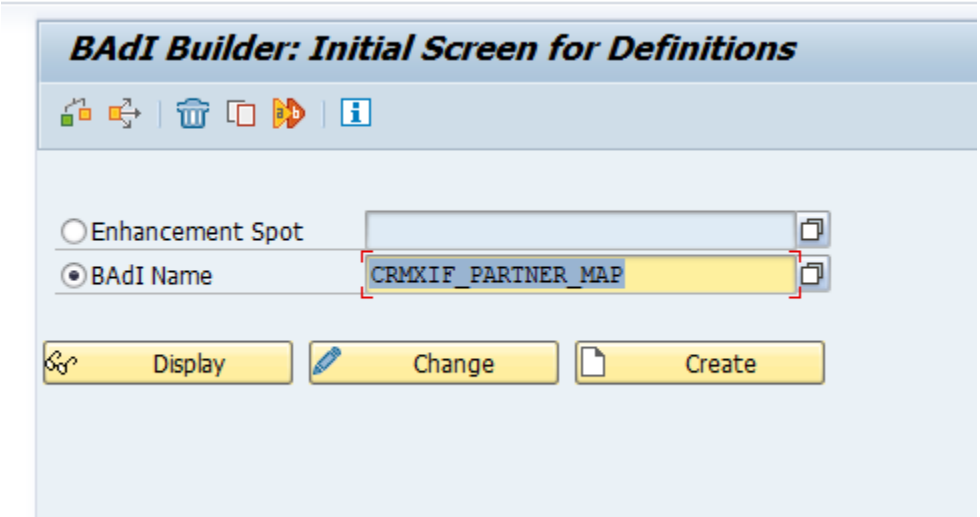

© 2019 SAP SE or an SAP affiliate company. All rights reserved.

```
            PARTNERGUID = GV_PARTNERGUID
        CONTPGUID   = lc_initial_guid
        CHANNEL     = IV_CHANNEL
        PERMISSION  = cl_crm_bupa_mktperm_tools=>GC_REJECTED
        VALID_FROM  = LS_VALID_PERMISSIONS-VALID_FROM
        VALUE       = LS_VALID_PERMISSIONS-VALUE
        TRANSPORTING NO FIELDS.
    IF sy-subrc EQ 0.
        DELETE LT_VALID_PERMISSIONS.
    ENDIF.
    ENDLOOP.

* delete redundant contact permission
    LOOP AT LT_VALID_PERMISSIONS INTO LS_VALID_PERMISSIONS WHERE
    PARTNERGUID = GV_PARTNERGUID   AND
    CONTPGUID   = GV_CONTPGUID     AND
    CHANNEL     = IV_CHANNEL       AND
    PERMISSION  = cl_crm_bupa_mktperm_tools=>GC_GIVEN.
        READ TABLE LT_VALID_PERMISSIONS WITH KEY
            PARTNERGUID = GV_PARTNERGUID
            CONTPGUID   = GV_CONTPGUID
            CHANNEL     = IV_CHANNEL
            PERMISSION  = cl_crm_bupa_mktperm_tools=>GC_REJECTED
            VALID_FROM  = LS_VALID_PERMISSIONS-VALID_FROM
            VALUE       = LS_VALID_PERMISSIONS-VALUE
        TRANSPORTING NO FIELDS.
        IF sy-subrc EQ 0.
            DELETE LT_VALID_PERMISSIONS.
        ENDIF.
    ENDLOOP.
    RT_VALID_PERMISSIONS = LT_VALID_PERMISSIONS.
    endmethod.
```

12. Activate your implementation.

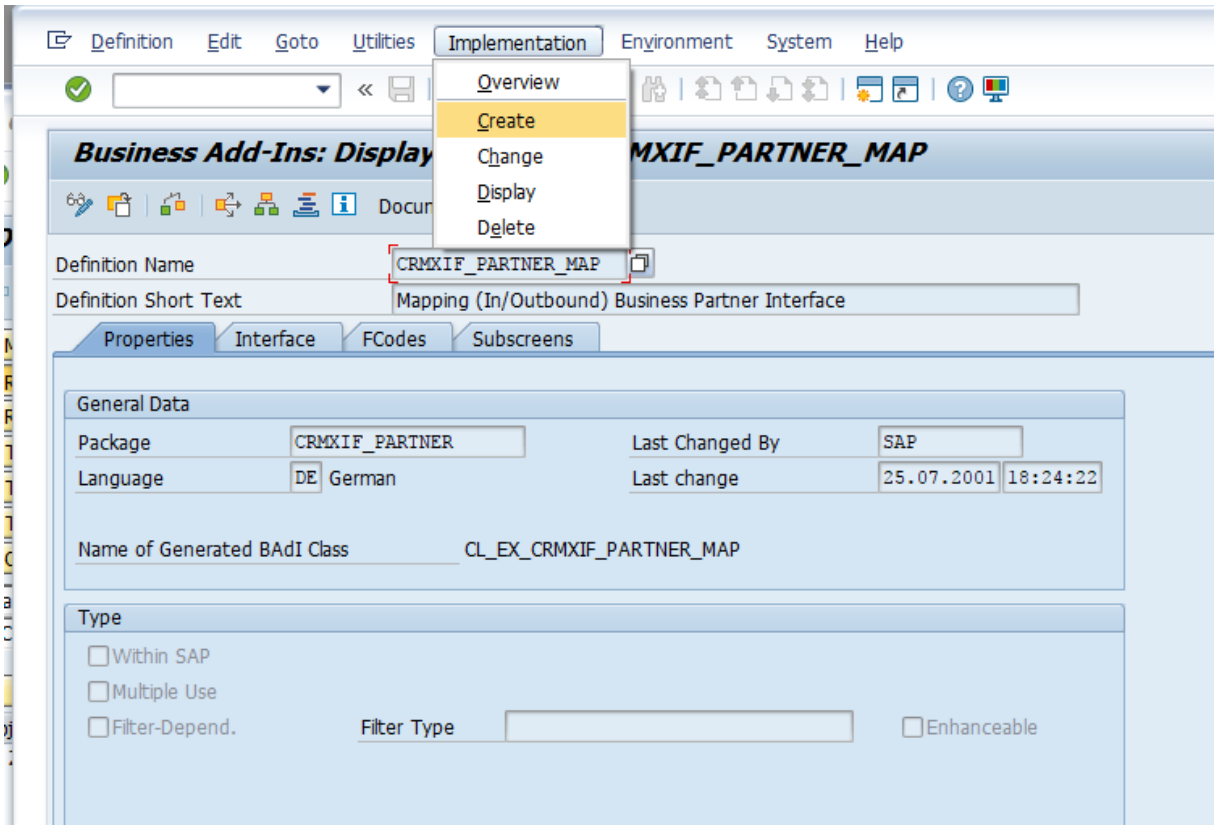
2.3 BAdI CRMXIF_PARTNER_MAP (Marketing Permissions for Accounts)

1. Start transaction SE18, select BAdI *Name CRMXIF_PARTNER_MAP*, and choose *Display.*

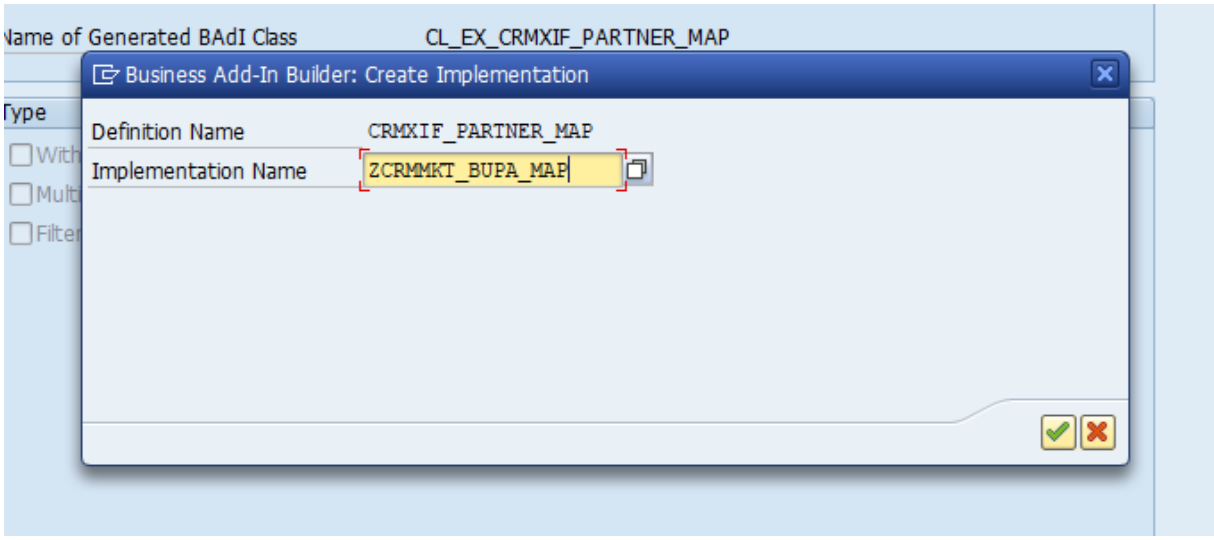


2. Choose *Implementation -> Create*.

© 2019 SAP SE or an SAP affiliate company. All rights reserved.

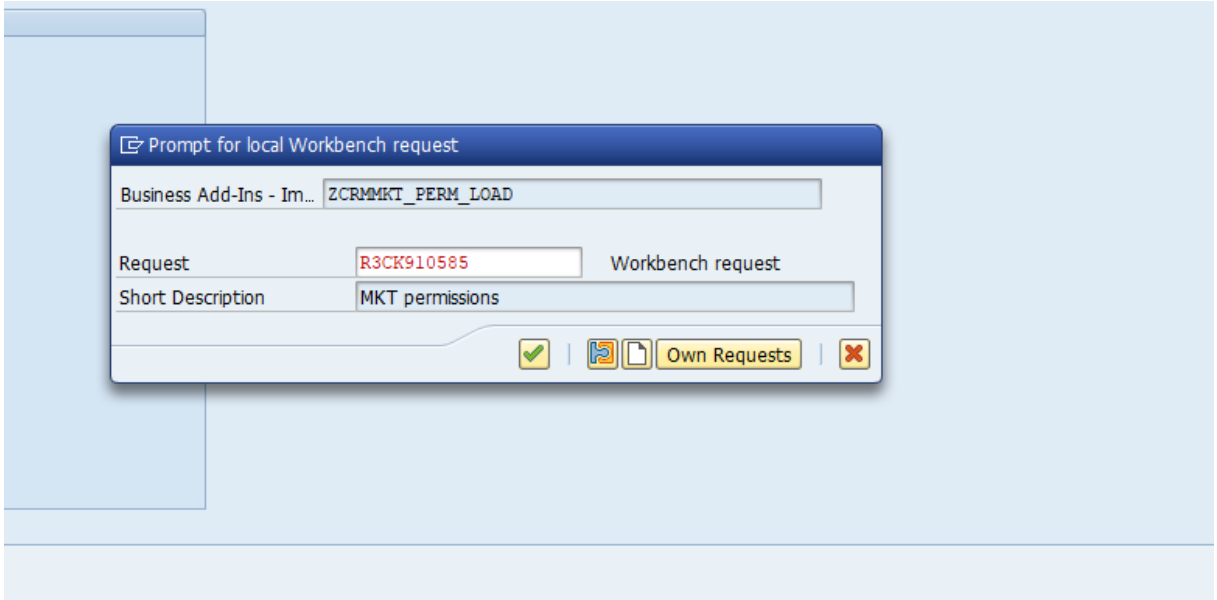


3. Enter an Implementation Name Z or Y<CRMMKT_BUPA_MAP> and click *OK.*

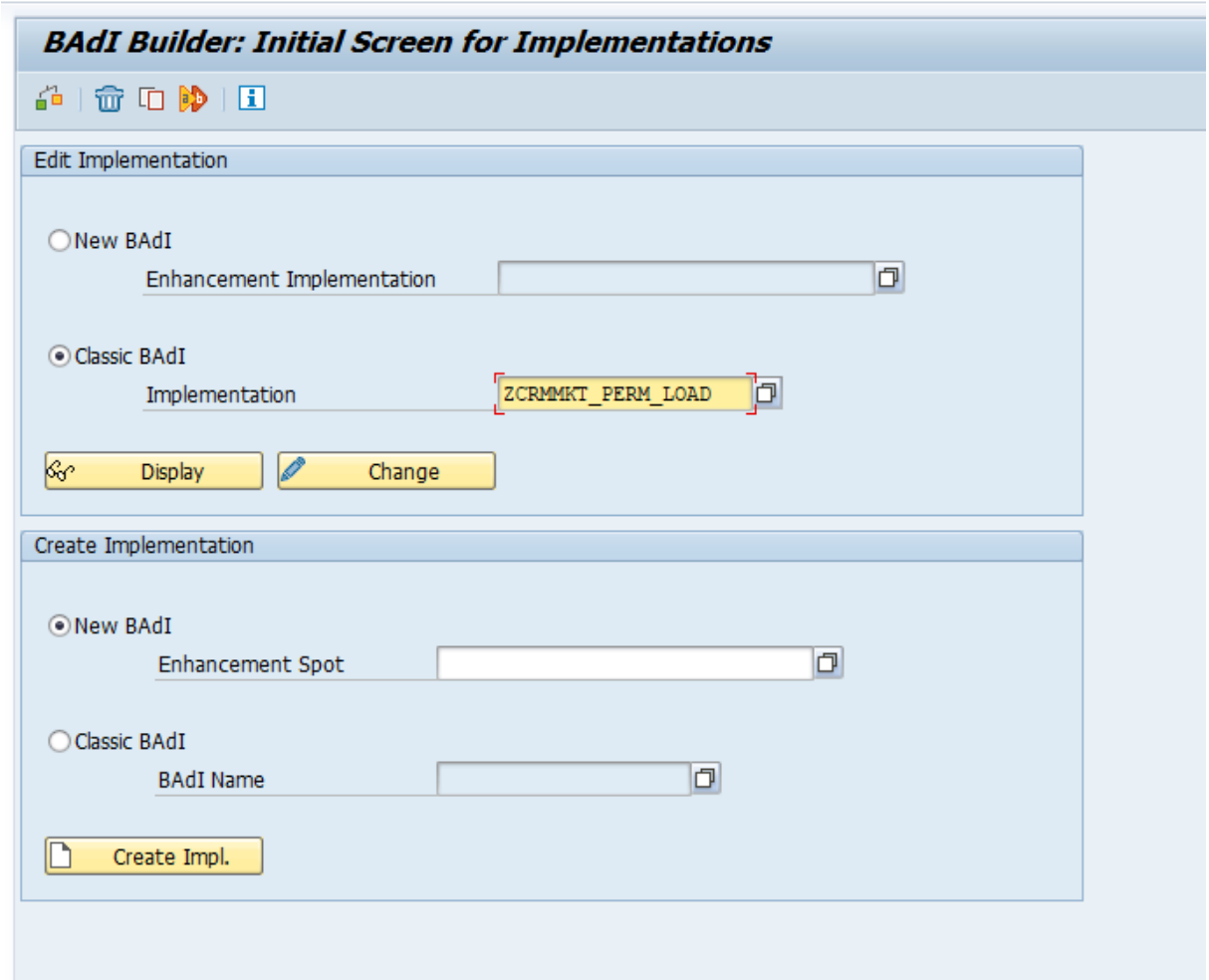


4. Enter Implementation Short Text and click on *Save*.
5. A pop up comes up and you have to enter your development package Z or Y<CRMMKT_IMP> created in 2.2.1, Click on *Save.*
6. A pop up for workbench request comes up and you have to enter your transport request.

© 2019 SAP SE or an SAP affiliate company. All rights reserved.

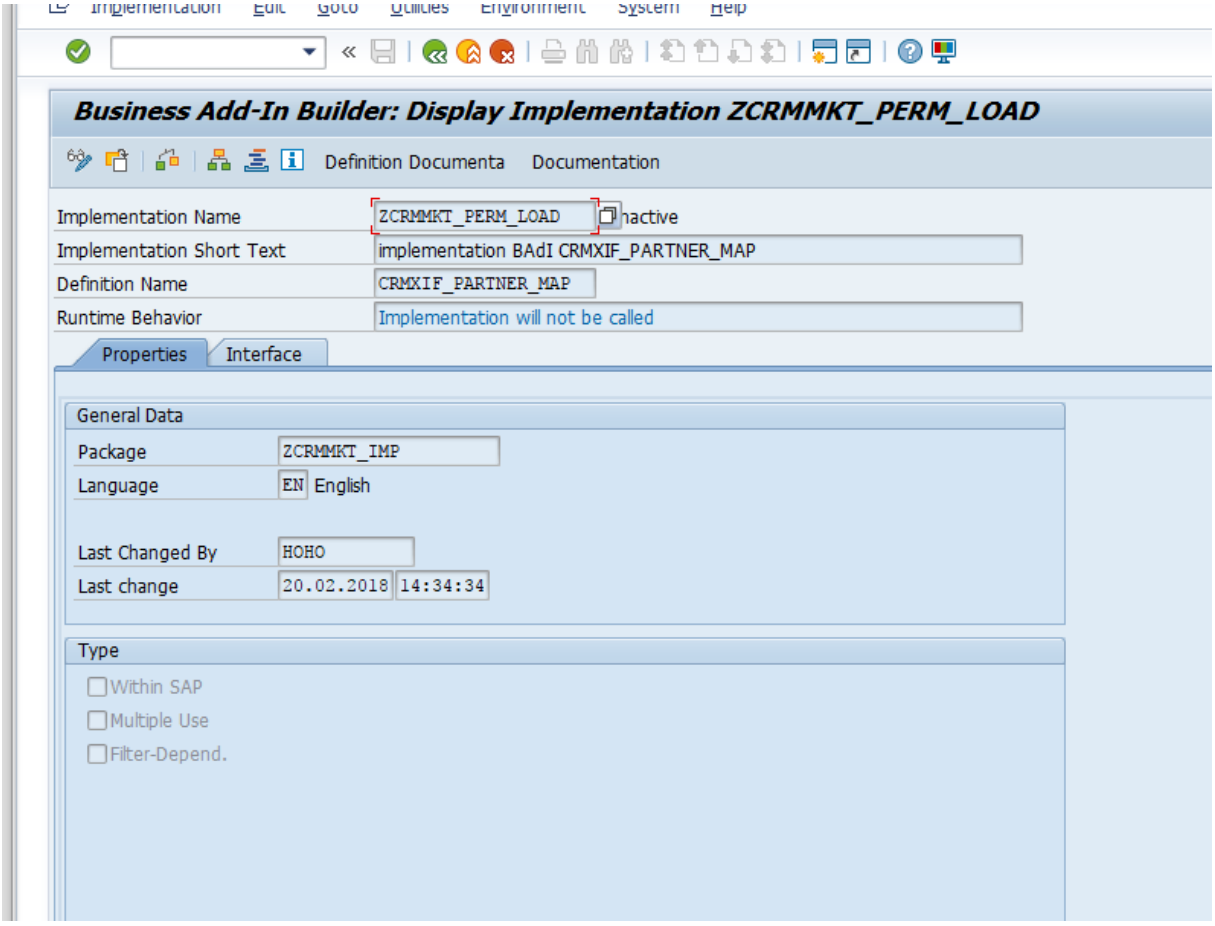


7. Start transaction SE19.
8. Search for the implementation, for example Z or Y<CRMMKT_PERM_LOAD>.



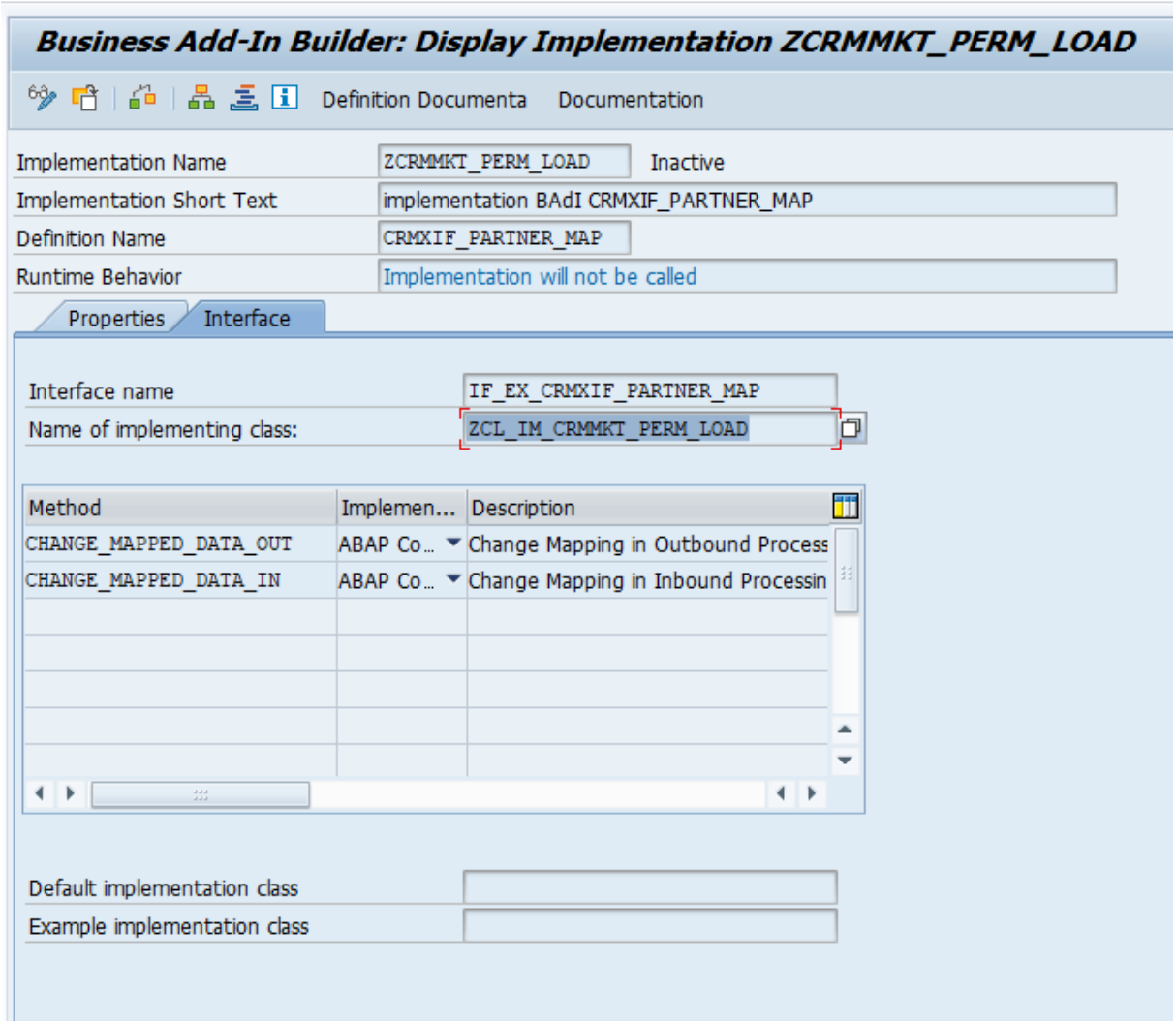
© 2019 SAP SE or an SAP affiliate company. All rights reserved.

9. Click on *Display*.

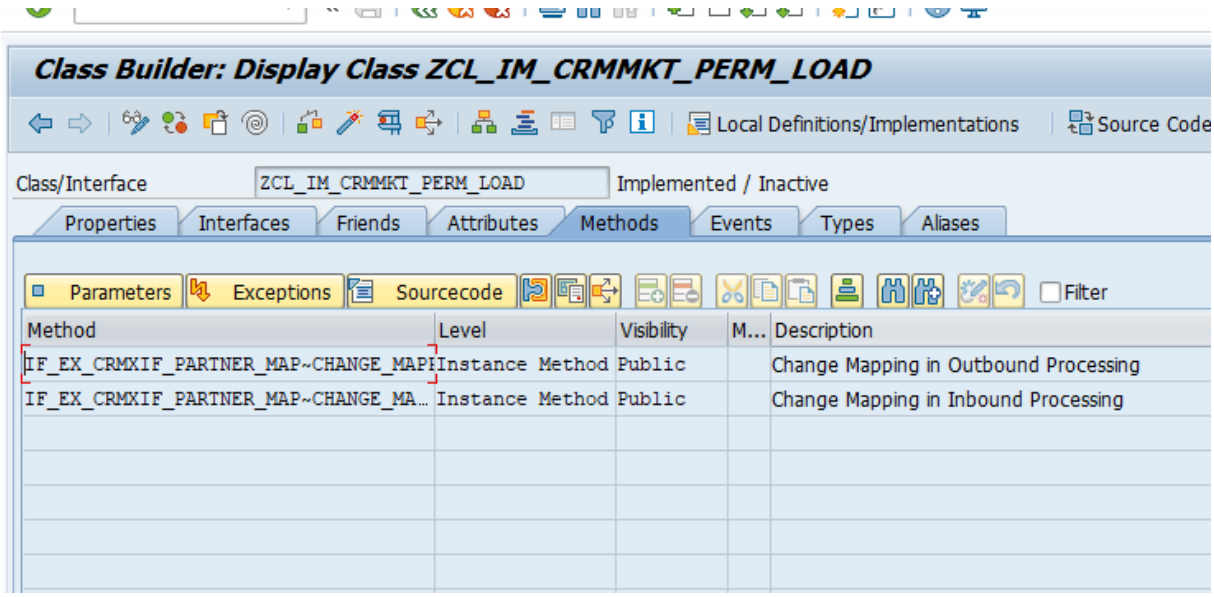


10. Go to the tab *Interface.*

PUBLIC
© 2019 SAP SE or an SAP affiliate company. All rights reserved.

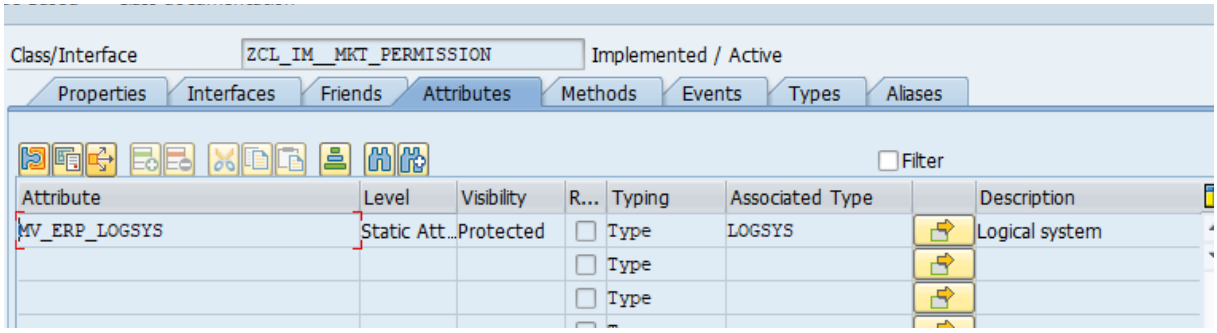


11. Double click on the *Implementing Class.*

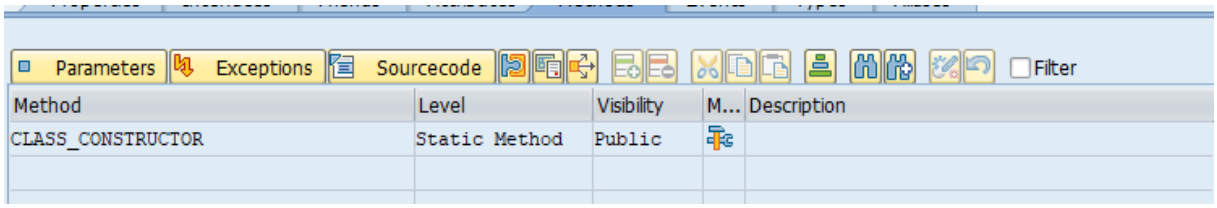


© 2019 SAP SE or an SAP affiliate company. All rights reserved.

12. Go to the tab *Attributes* and create a new attribute.



13. Go to the tab *Methods* and create a new class constructor.



14. Implement the CLASS_CONSTRUCTOR.

```
 method CLASS_CONSTRUCTOR.
DATA lt_erpsites TYPE TABLE OF smof_erpsh.
DATA ls_erpsites TYPE smof_erpsh.

CALL FUNCTION 'SMOF_READ_SMOFERPSH'
 EXPORTING
  i_sitetypeid = 'SMOF_ERPSITE'
 TABLES
  t_erpsites   = lt_erpsites.

      CHECK LINES( lt_erpsites ) = 1.
  READ TABLE lt_erpsites INTO ls_erpsites INDEX 1.
  mv_erp_logsys = ls_erpsites-erplogsys.
endmethod.
```

Implement the interface IF_EX_CRMXIF_PARTNER_MAP~CHANGE_MAPPED_DATA_OUT.

```
method IF_EX_CRMXIF_PARTNER_MAP~CHANGE_MAPPED_DATA_OUT.
    CONSTANTS lc_initial_guid   TYPE bu_partner_guid VALUE '0'.
    FIELD-SYMBOLS:
        <mkt_permission_t>          TYPE CRMT_BUS_EI_MKT_PERMISSIONS_T,
        <mkt_permission>            TYPE CRMT_BUS_EI_MKT_PERMISSIONS,
        <mkt_permission_channel> TYPE CRMT_BUS_EI_MKT_PERMISSIONS,
        <xif_partner_complex_t> type CRMXIF_PARTNER_COMPLEX_T,
        <xif_partner_complex>  type CRMXIF_PARTNER_COMPLEX.

    DATA: ls_partner_complex type CRMXIF_PARTNER_COMPLEX.
    DATA:   lt_mkp_applicable type ZMKT_APPLICABLE_T,
    ls_mkp_applicable like line of lt_mkp_applicable,
    lt_channel                  type CRMT_BUS_EI_MKT_PERMISSIONS_T,
    ls_channel                  type CRMT_BUS_EI_MKT_PERMISSIONS,
    lt_valid_permissions type crmt_but_mktperm_all_t,
    ls_valid_permission       type crmt_but_mktperm_all,
    lt_valid_mktperm          type crmt_but_mktperm_all_t,
    lt_applicable_account type CRMT_BUT_MKP_APPLICABLE_T,
    ls_applicable_account like line of lt_applicable_account,
    ls_value type ZMKT_VALUES,
    lt_value type ZMKT_VALUES_T.

* address dependent communication channel
    DATA:    lt_address type BUS_EI_BUPA_ADDRESS_T,
```


PUBLIC
© 2019 SAP SE or an SAP affiliate company. All rights reserved.

```
                ls_address like line of lt_address,
                lt_phone type BUS_EI_BUPA_TELEPHONE_T,
                ls_phone type BUS_EI_BUPA_TELEPHONE,
            lt_fax      type BUS_EI_BUPA_FAX_T,
            ls_fax      type BUS_EI_BUPA_FAX,
            lt_email    type BUS_EI_BUPA_SMTP_T,
            ls_email    type BUS_EI_BUPA_SMTP,
            lt_pager    type BUS_EI_BUPA_PAG_T,
            ls_pager    type BUS_EI_BUPA_PAG.

    DATA: lv_timestamp_valid_from  TYPE timestamp,
          lv_initial_time        TYPE syst_uzeit,
          lv_timestamp_current_utc TYPE timestamp,
          lv_initial_timezone      TYPE ttzz-tzone,
          lv_date_valid_from       TYPE sydatum,
          lv_date_UTC            TYPE sydatum.

      DATA: lv_index type integer.
* standard scenario
      DATA:     ls_mkt_area                    TYPE zmkt_areas,
                ls_bus_ei_mkt_permissions      TYPE CRMT_BUS_EI_MKT_PERMISSIONS,
                ls_mkt_permissions             TYPE CRMT_BUS_EI_MKT_PERMISSIONS.
* standard scenario
      DATA: lo_zcl_crm_bupa_mktperm type ref to zcl_crm_bupa_mktperm_common.
* CRM -> C4C
      FIELD-SYMBOLS <ls_data_mapped>   LIKE LINE OF cs_data_mapped.
      FIELD-SYMBOLS <ls_role>          LIKE LINE OF <ls_data_mapped>- central_data-role-roles.
      FIELD-SYMBOLS <ls_ident_numbers> TYPE bus_ei_bupa_identification.
      LOOP AT cs_data_mapped ASSIGNING <ls_data_mapped>.
        DELETE <ls_data_mapped>-central_data-role-roles
                    WHERE data- rolecategory = 'BUP003'.  "Employee

          "No role is left -> Ignore the business partner
        IF sy-subrc IS INITIAL AND <ls_data_mapped>-central_data-role- roles IS INITIAL.
          DELETE cs_data_mapped.
          CONTINUE.
        ENDIF.
        IF mv_erp_logsys IS NOT INITIAL.
          READ TABLE <ls_data_mapped>-central_data-ident_number-
                    ident_numbers ASSIGNING <ls_ident_numbers>
                          WITH KEY  data_key-identificationcategory = 'CRM002'.  "ERP Customer ID

          IF sy-subrc IS INITIAL.
              <ls_ident_numbers>-data-idinstitute = mv_erp_logsys.
          ENDIF.
        ENDIF.
      ENDLOOP.
      * CRM -> C4C

      LOOP AT cs_data_mapped ASSIGNING <xif_partner_complex>.
        clear lt_mkp_applicable.
        clear ls_mkp_applicable.
        clear lt_address.
        clear lt_channel.
        UNASSIGN <mkt_permission_t>.
        UNASSIGN <mkt_permission>.
        UNASSIGN <mkt_permission_channel>.

        IF <xif_partner_complex>-HEADER-OBJECT_TASK EQ 'C'.
      * address dependent communication channel
          lt_address = <xif_partner_complex>-CENTRAL_DATA-ADDRESS-ADDRESSES.
          check lt_address is not initial.
          READ TABLE lt_address INTO ls_address WITH KEY DATA-POSTAL-DATA-STANDARDADDRESS = 'X'.
          IF sy-subrc EQ 0.
            LOOP AT ls_address-DATA-COMMUNICATION-PHONE-PHONE INTO ls_phone
              WHERE CURRENTLY_VALID = 'X'.
                APPEND ls_phone TO lt_phone.
            ENDLOOP.

            READ TABLE ls_address-DATA-COMMUNICATION-FAX-FAX INTO ls_fax
```


© 2019 SAP SE or an SAP affiliate company. All rights reserved.

```
        WITH KEY  CONTACT-DATA-STD_NO = 'X'
              CURRENTLY_VALID = 'X'.

       READ TABLE ls_address-DATA-COMMUNICATION-SMTP-SMTP INTO ls_email
        WITH KEY  CONTACT-DATA-STD_NO = 'X'
              CURRENTLY_VALID = 'X'.

       READ TABLE ls_address-DATA-COMMUNICATION-PAGER-PAGER INTO ls_pager
        WITH KEY  CONTACT-DATA-STD_NO = 'X'
              CURRENTLY_VALID = 'X'.

     ENDIF.
     ASSIGN <xif_partner_complex>-CRM_DATA-MKT_PERMISSION-MKT_PERMISSIONS TO <mkt_permission_t>.

     ls_mkp_applicable-MKT_APPLICABLE-PARTNERGUID = <xif_partner_complex>-HEADER-OBJECT_INSTANCE-
PARTNER_GUID.
     ls_mkp_applicable-MKT_APPLICABLE-CONTPGUID = LC_INITIAL_GUID.

* standard scenario
     CREATE OBJECT LO_ZCL_CRM_BUPA_MKTPERM
      exporting
       IV_PARTNERGUID = ls_mkp_applicable-MKT_APPLICABLE-PARTNERGUID
       IV_CONTPGUID   = ls_mkp_applicable-MKT_APPLICABLE-CONTPGUID.
* standard scenario

     LOOP AT <mkt_permission_t> ASSIGNING <mkt_permission_channel>.
      READ TABLE <xif_partner_complex>-central_data-role-roles WITH KEY data-rolecategory = 'BUP001'
         TRANSPORTING NO FIELDS.
      IF sy-subrc NE 0.
       IF  <mkt_permission_channel>-DATA-CHANNEL EQ 'TEL' OR
          <mkt_permission_channel>-DATA-CHANNEL EQ 'FAX' OR
          <mkt_permission_channel>-DATA-CHANNEL EQ 'INT' OR
          <mkt_permission_channel>-DATA-CHANNEL EQ 'PAG'.
        READ TABLE lt_channel WITH KEY
           DATA-CHANNEL = <mkt_permission_channel>-DATA-CHANNEL
           TRANSPORTING NO FIELDS.
       IF sy-subrc NE 0.
          <mkt_permission_channel>-DATA-ORIGIN = ''.
          <mkt_permission_channel>-DATA-PERMISSION = ''.
          <mkt_permission_channel>-DATA-VALID_FROM = ''.
          <mkt_permission_channel>-DATA-VALUE = ''.
          APPEND <mkt_permission_channel> TO lt_channel.
        ENDIF.
       ENDIF.
      ENDIF.
      DELETE <mkt_permission_t>.
     ENDLOOP.
* process pro channel
     LOOP AT lt_channel INTO ls_channel.
       clear lt_mkp_applicable.
       clear lt_valid_permissions.
       clear lt_value.
       CASE ls_channel-DATA-CHANNEL.
*      TEL -> PHONE
        WHEN 'TEL'.
         LOOP AT lt_phone INTO ls_phone.
          ls_value-VALUE = ls_phone-CONTACT-DATA-TEL_NO.
          ls_value-STD = ls_phone-CONTACT-DATA-STD_NO.
          ls_value-R_3_USER = ls_phone-CONTACT-DATA-R_3_USER.
          APPEND ls_value TO lt_value.
         ENDLOOP.
*      FAX -> FAX
        WHEN 'FAX'.
         ls_value-VALUE = ls_fax-CONTACT-DATA-FAX_NO.
         ls_value-STD = ls_fax-CONTACT-DATA-STD_NO.
         APPEND ls_value TO lt_value.
*      INT -> EMAIL
        WHEN 'INT'.
         ls_value-VALUE = ls_email-CONTACT-DATA-E_MAIL.
         ls_value-STD = ls_email-CONTACT-DATA-STD_NO.
         APPEND ls_value TO lt_value.
*      PAG -> SMS
```


© 2019 SAP SE or an SAP affiliate company. All rights reserved.

```
          WHEN 'PAG'.
            ls_value-VALUE = ls_pager-CONTACT-DATA-PAGER_NO.
            ls_value-STD = ls_pager-CONTACT-DATA-STD_NO.
            APPEND ls_value TO lt_value.
          WHEN OTHERS.
        ENDCASE.

      IF lt_value IS INITIAL.
        CONTINUE.
      ENDIF.

        CALL METHOD LO_ZCL_CRM_BUPA_MKTPERM->CHECK_CONTACT_ALLOWED_
          EXPORTING
            IV_CHANNEL           = ls_channel-DATA-CHANNEL
            IT_STD_VALUE          = lt_value
            IV_CHECK_ACCOUNT_CONTACT = abap_false
          CHANGING
            CT_MKTPERM_APPLICABLE    = lt_mkp_applicable
          RECEIVING
            RT_VALID_PERMISSIONS     = lt_valid_permissions.

        IF lt_valid_permissions IS INITIAL OR lt_mkp_applicable IS INITIAL.
          CONTINUE.
        ENDIF.
*************************************
        LOOP AT lt_valid_permissions INTO ls_valid_permission.
          IF ls_valid_permission-VALUE IS INITIAL AND ls_value-VALUE IS INITIAL.
            CONTINUE.
          ENDIF.
          APPEND INITIAL LINE TO <mkt_permission_t>  ASSIGNING <mkt_permission>.
          lv_index = sy-TABIX.

      *check against mobile phone
          IF ls_channel-DATA-CHANNEL EQ 'TEL'.
            IF ls_valid_permission-VALUE IS INITIAL.
              READ TABLE lt_value INTO ls_value WITH KEY
                STD = 'X'.
              IF sy-subrc EQ 0.
                IF ls_value-R_3_USER = '2' OR ls_value-R_3_USER = '3'.
                  <mkt_permission>-DATA-CHANNEL = 'TE2'.
                ENDIF.
              ENDIF.
            ELSE.
              READ TABLE ls_address-DATA-COMMUNICATION-PHONE-PHONE INTO ls_phone
                WITH KEY CONTACT-DATA-TEL_NO = ls_valid_permission-value.
              IF sy-subrc EQ 0.
                IF ls_phone-CONTACT-DATA-R_3_USER = '2' OR ls_phone-CONTACT-DATA-R_3_USER = '3'.
                  <mkt_permission>-DATA-CHANNEL = 'TE2'.
                ENDIF.
              ENDIF.
            ENDIF.
          ENDIF.
*check against mobile phone
          IF <mkt_permission>-DATA-CHANNEL IS INITIAL.
            <mkt_permission>-DATA-CHANNEL = ls_channel-DATA-CHANNEL.
          ENDIF.
      *       simulate CPI conversion of time stamp; add initial time to timestamp valid from date is not in future
              CONVERT DATE ls_valid_permission-valid_from TIME lv_initial_time INTO TIME STAMP
lv_timestamp_valid_from TIME ZONE lv_initial_timezone.
      *       get current timestamp in utc
              GET TIME STAMP FIELD  lv_timestamp_current_utc.

      *       get dates from timestamps initial timezone is UTC
              CONVERT TIME STAMP lv_timestamp_valid_from  TIME ZONE lv_initial_timezone INTO DATE
lv_date_valid_from.
              CONVERT TIME STAMP lv_timestamp_current_utc TIME ZONE lv_initial_timezone INTO DATE lv_date_utc.

      *       if currecnt UTC date is smaller / valid from date is in future => current UTC should be passed
```


© 2019 SAP SE or an SAP affiliate company. All rights reserved.

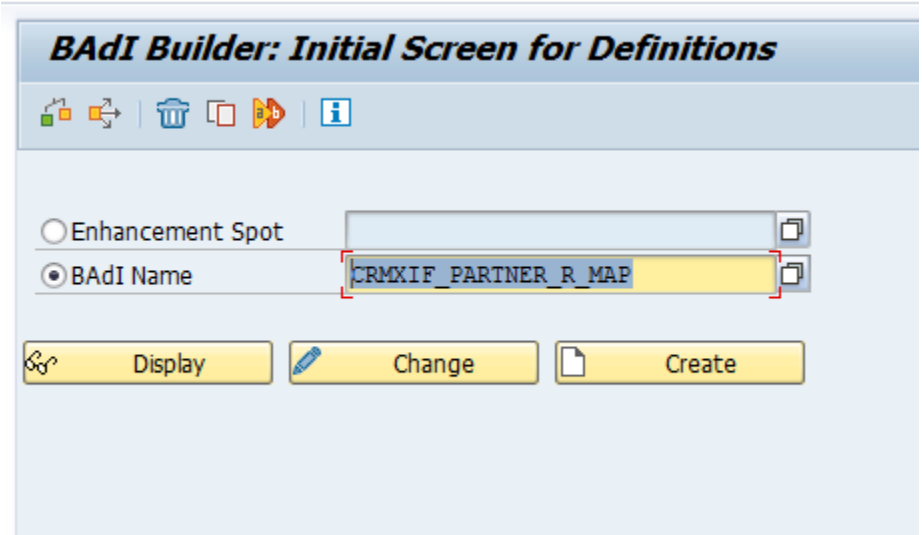
```
            IF lv_date_valid_from > lv_date_utc.
             ls_valid_permission-valid_from = lv_date_utc.
            ENDIF.

        <mkt_permission>-DATA-VALID_FROM = ls_valid_permission-VALID_FROM.
        READ TABLE lt_mkp_applicable INTO ls_mkp_applicable
         WITH KEY MKT_APPLICABLE-PARTNERGUID = ls_mkp_applicable-MKT_APPLICABLE-PARTNERGUID
              MKT_APPLICABLE-CONTPGUID   = ls_mkp_applicable-MKT_APPLICABLE-CONTPGUID
              VALUE = ls_valid_permission-VALUE
              RECORD_GUID = LS_VALID_PERMISSION-RECORD_GUID.
        IF ls_mkp_applicable-MKT_APPLICABLE-CONTACT_ALLOWED EQ 'X'.
         <mkt_permission>-DATA-PERMISSION = '001'.
        ELSE.
         <mkt_permission>-DATA-PERMISSION = '002'.
        ENDIF.
*         <mkt_permission>-DATA-ORIGIN = ls_valid_permission-ORIGIN.
        <mkt_permission>-DATA_KEY-RECORD_GUID = ls_valid_permission-RECORD_GUID.

        IF ls_valid_permission-VALUE IS INITIAL.
          READ TABLE lt_value INTO ls_value WITH KEY STD = 'X'.
          IF sy-subrc EQ 0.
            <mkt_permission>-DATA-VALUE = ls_value-VALUE.
*something wrong, remove the permissions
          ELSE.
            DELETE <MKT_PERMISSION_T> INDEX lv_index.
            CONTINUE.
          ENDIF.
        ELSE.
         READ TABLE lt_value WITH KEY VALUE = LS_VALID_PERMISSION-VALUE TRANSPORTING NO FIELDS.
         IF sy-subrc EQ 0.
          <mkt_permission>-DATA-VALUE = ls_valid_permission-VALUE.
*something wrong, remove the permissions
         ELSE.
          DELETE <MKT_PERMISSION_T> INDEX lv_index.
          CONTINUE.
         ENDIF.
        ENDIF.
       ENDLOOP.
     ENDLOOP.
*move duplicates
* standard scenario
     DELETE ADJACENT DUPLICATES from <MKT_PERMISSION_T> COMPARING DATA-CHANNEL DATA-
PERMISSION DATA-VALID_FROM DATA-VALUE.
* standard scenario

*move all redundant entries
     LOOP AT <mkt_permission_t> ASSIGNING <mkt_permission_channel>.
*move all older entries
      LOOP AT <mkt_permission_t> ASSIGNING <mkt_permission>
       WHERE DATA-CHANNEL    EQ <mkt_permission_channel>-DATA-CHANNEL
*        AND  DATA-VALID_FROM LT <mkt_permission_channel>-DATA-VALID_FROM
       AND  DATA-VALUE     EQ <mkt_permission_channel>-DATA-VALUE.
*         entry classified, remove from working table
        IF <mkt_permission>-DATA-VALID_FROM LT <mkt_permission_channel>-DATA-VALID_FROM.
         DELETE <mkt_permission_t>.
        ELSE.
         IF <mkt_permission>-DATA-VALID_FROM EQ <mkt_permission_channel>-DATA-VALID_FROM.
           IF <MKT_PERMISSION>-DATA-PERMISSION EQ cl_crm_bupa_mktperm_tools=>GC_GIVEN AND
             <MKT_PERMISSION_CHANNEL>-DATA-PERMISSION EQ cl_crm_bupa_mktperm_tools=>GC_REJECTED.
            DELETE <mkt_permission_t>.
           ENDIF.
         ENDIF.
        ENDIF.
      ENDLOOP.
*move all older entries
     ENDLOOP.
*move all redundant entries
   ENDIF.
  ENDLOOP.
 endmethod.
```


© 2019 SAP SE or an SAP affiliate company. All rights reserved.

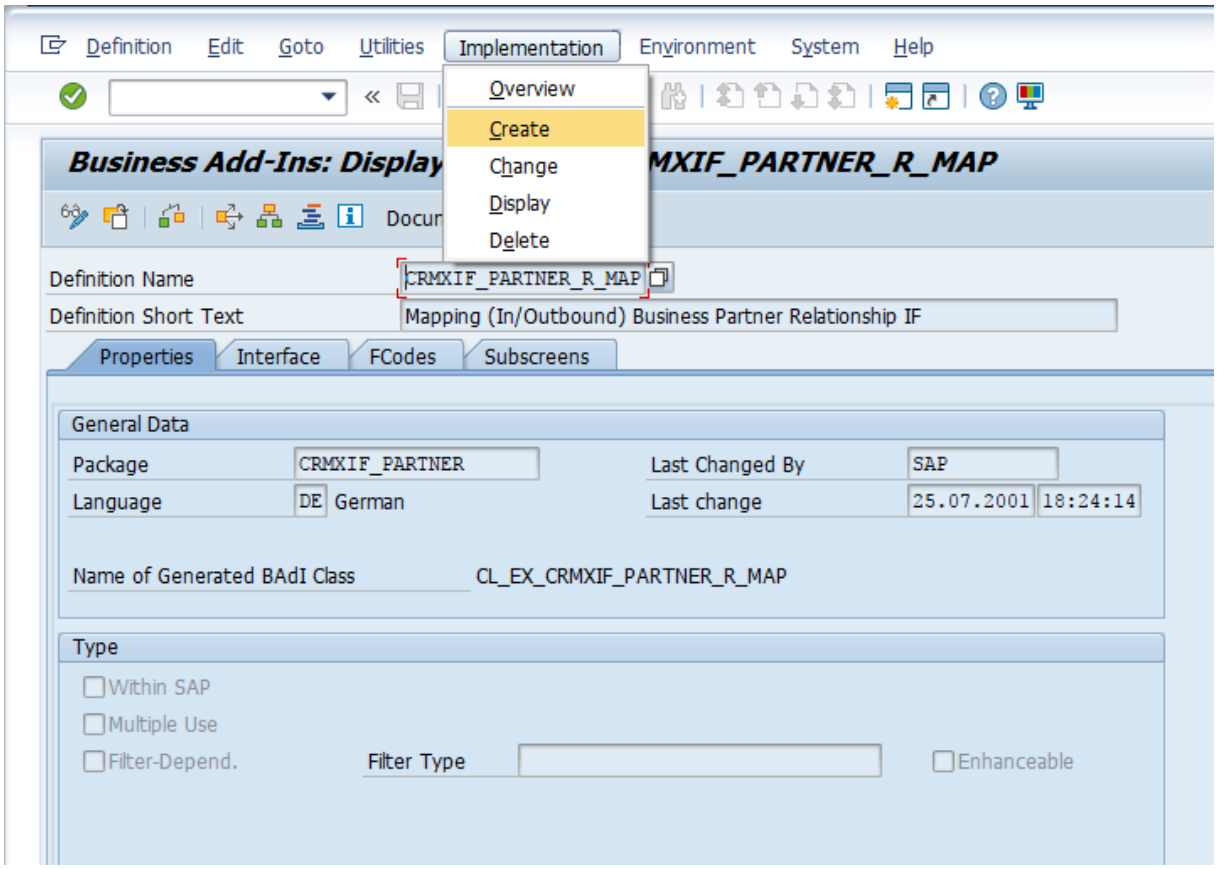
15. Activate your implementation.

2.4 BAdI CRMXIF_PARTNER_R_MAP (Marketing Permissions for Contacts)

1. Start SE18, select BAdI Name *CRMXIF_PARTNER_R_MAP*, and click on *Display.*

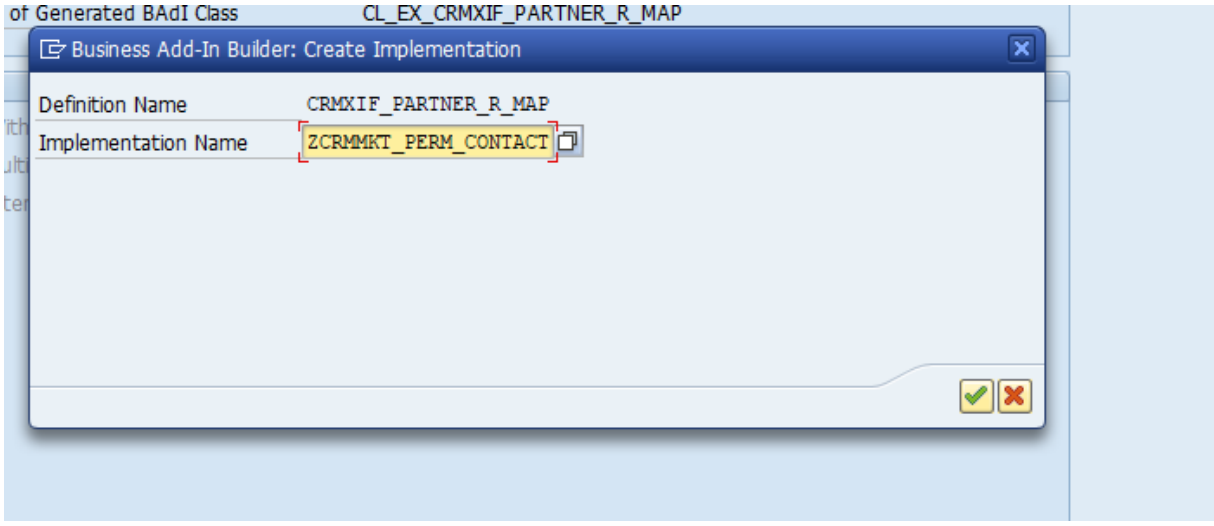


2. Choose *Implementation-> Create.*

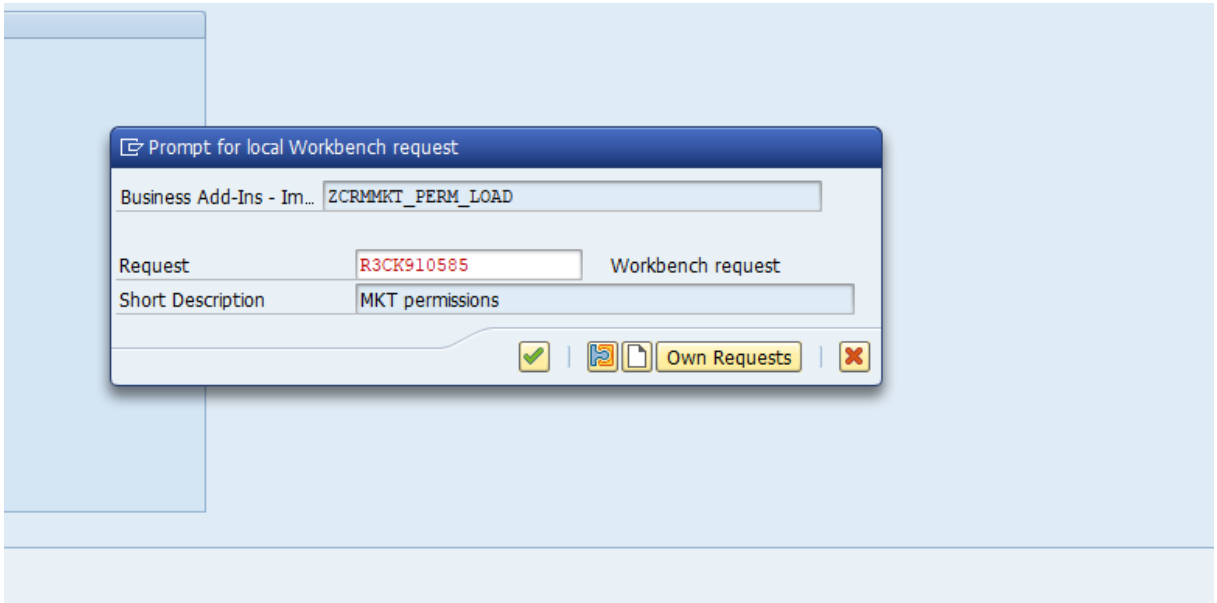


3. Enter an Implementation Name Z or Y<CRMMKT_PERM_CONTACT> and click *OK.*

© 2019 SAP SE or an SAP affiliate company. All rights reserved.

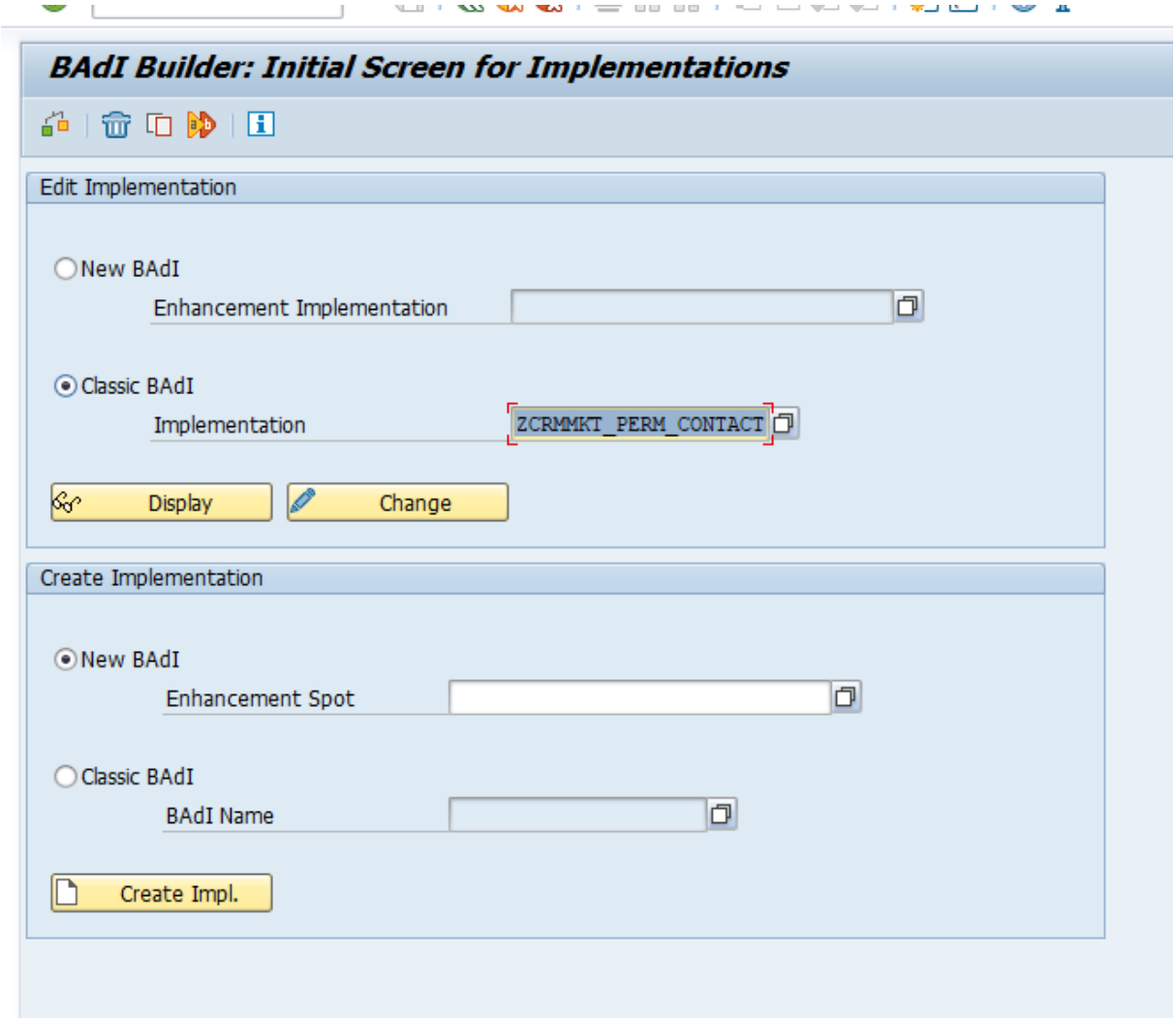


4. Enter Implementation Short Text and click on *Save.*
5. A pop up comes up and you have to enter your development package and click on *Save.*
6. A pop up for workbench request comes up and you have to enter your transport request.



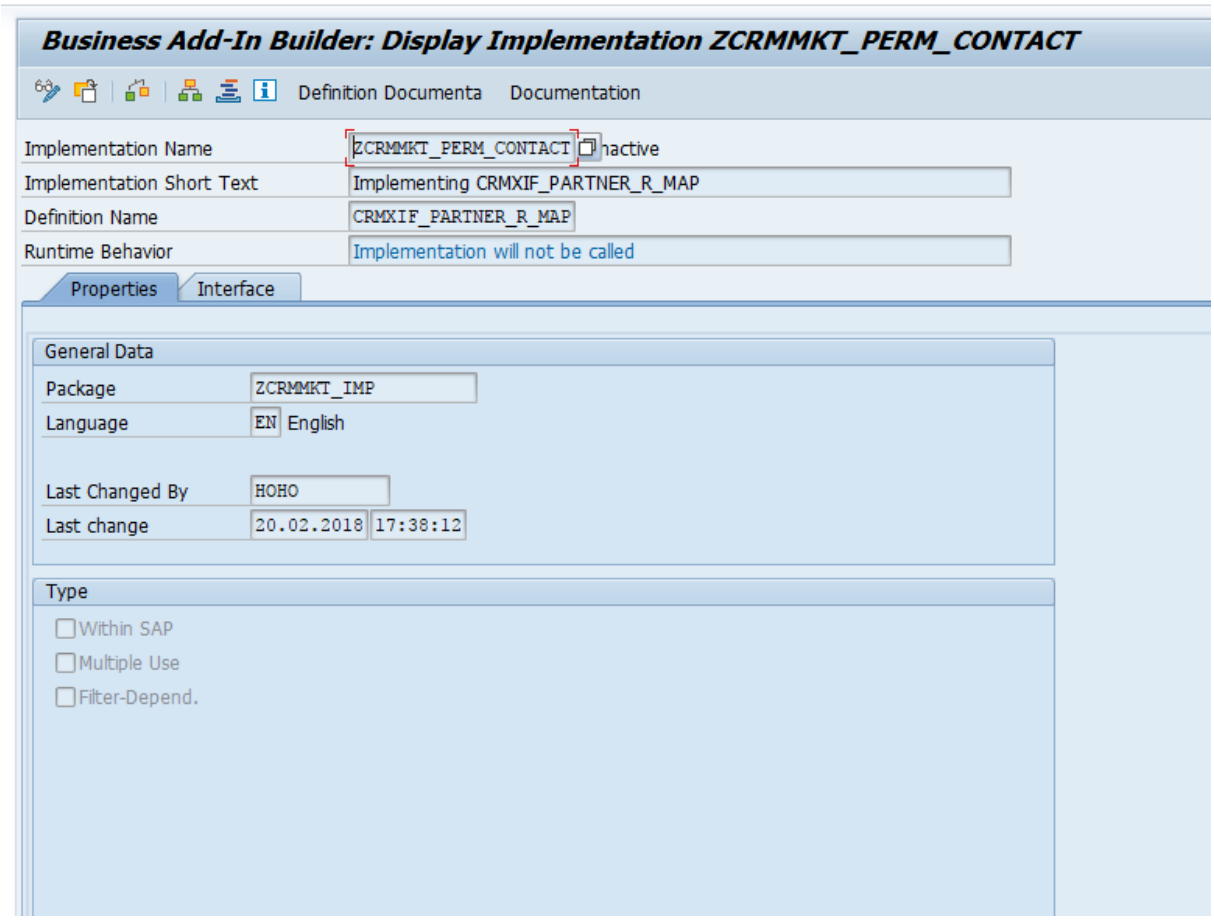
7. Start SE19.
8. Search for the implementation name.

© 2019 SAP SE or an SAP affiliate company. All rights reserved.



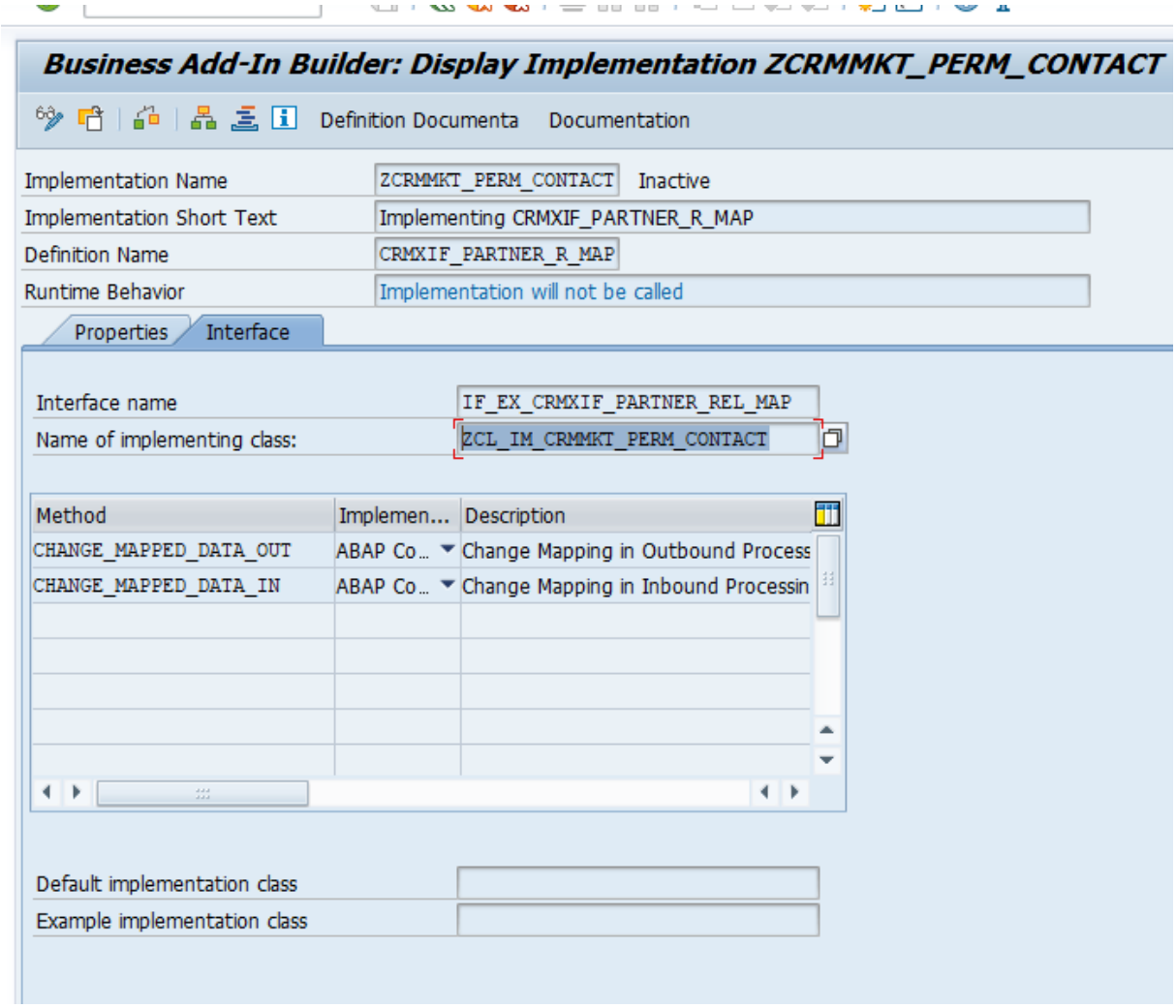
9. Click on *Display.*

PUBLIC
© 2019 SAP SE or an SAP affiliate company. All rights reserved.

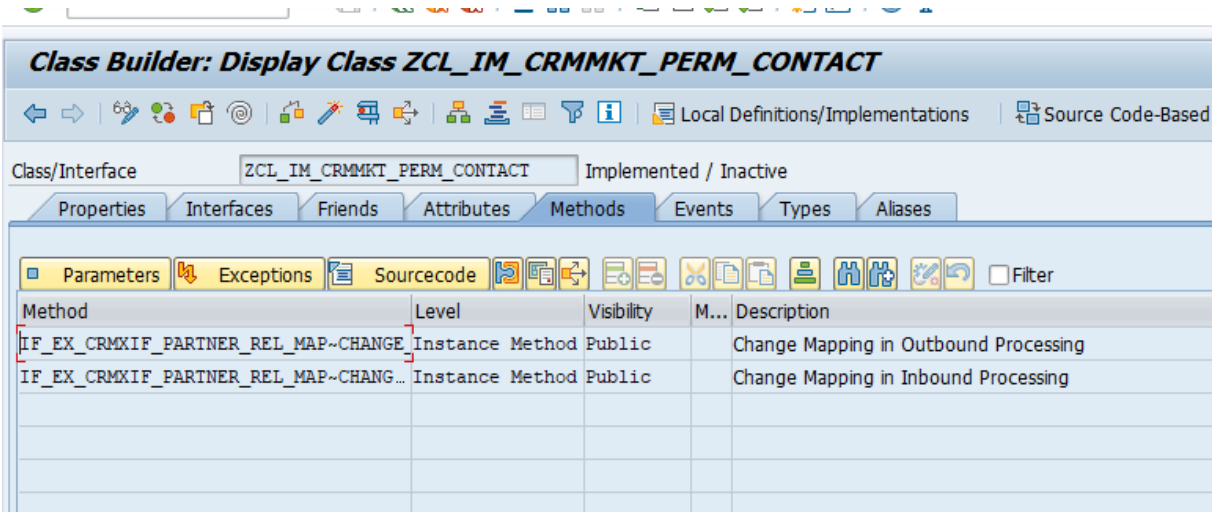


10. Go to the tab *Interface.*

© 2019 SAP SE or an SAP affiliate company. All rights reserved.

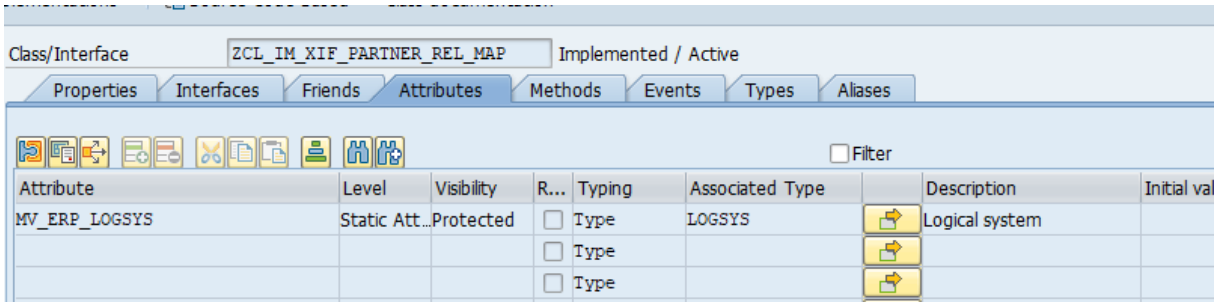


11. Double click on the *Implementing Class*.

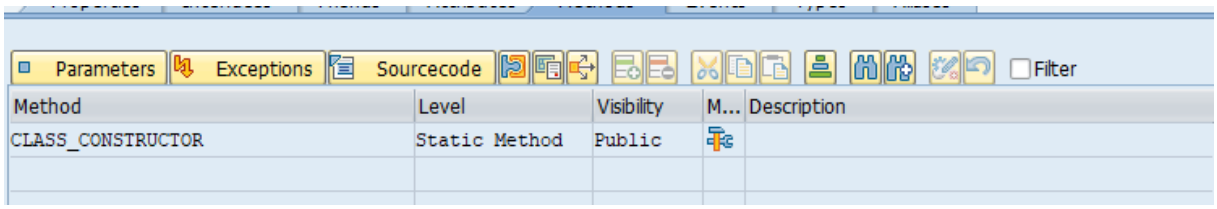


12. Go to the tab *Attributes* and create a new attribute.

© 2019 SAP SE or an SAP affiliate company. All rights reserved.



13. Go to the tab *Methods* and create a new class constructor.



14. Implement the CLASS_CONSTRUCTOR.

```
method CLASS_CONSTRUCTOR.
  DATA lt_erpsites TYPE TABLE OF smof_erpsh.
  DATA ls_erpsites TYPE smof_erpsh.

  CALL FUNCTION 'SMOF_READ_SMOFERPSH'
   EXPORTING
    i_sitetypeid = 'SMOF_ERPSITE'
   TABLES
    t_erpsites   = lt_erpsites.

  CHECK LINES( lt_erpsites ) = 1.

  READ TABLE lt_erpsites INTO ls_erpsites INDEX 1.

  mv_erp_logsys = ls_erpsites-erplogsys.
endmethod.
```

Implement the interface IF_EX_CRMXIF_PARTNER_REL_MAP~CHANGE_MAPPED_DATA_OUT.

```
method IF_EX_CRMXIF_PARTNER_REL_MAP~CHANGE_MAPPED_DATA_OUT.
  CONSTANTS lc_initial_guid   TYPE bu_partner_guid VALUE '0'.
  FIELD-SYMBOLS:
   <mkt_permission_t> TYPE CRMT_BUS_EI_MKT_PERMISSIONS_T,
   <mkt_permission> TYPE CRMT_BUS_EI_MKT_PERMISSIONS,
   <mkt_permission_channel> TYPE CRMT_BUS_EI_MKT_PERMISSIONS,
   <xif_partner_rel_complex_t> type CRMXIF_PARTNER_REL_COMPLEX_T,
   <xif_partner_rel_complex> type CRMXIF_PARTNER_REL_COMPLEX.

  DATA: ls_partner_complex type CRMXIF_PARTNER_REL_COMPLEX,
      lt_bupa_cp type crmt_but_mkp_applicable_t,
      ls_bupa_cp like line of lt_bupa_cp.
  DATA:   lt_mkp_applicable type ZMKT_APPLICABLE_T,
       ls_mkp_applicable like line of lt_mkp_applicable,
       lt_channel type CRMT_BUS_EI_MKT_PERMISSIONS_T,
       ls_channel type CRMT_BUS_EI_MKT_PERMISSIONS,
       lt_valid_permissions type crmt_but_mktperm_all_t,
       ls_valid_permission type crmt_but_mktperm_all,
       lt_valid_mktperm type crmt_but_mktperm_all_t,
       lt_applicable_account type CRMT_BUT_MKP_APPLICABLE_T,
       ls_applicable_account like line of lt_applicable_account,
       ls_value type ZMKT_VALUES,
       lt_value type ZMKT_VALUES_T.
```


© 2019 SAP SE or an SAP affiliate company. All rights reserved.

```
* address dependent communication channel
  DATA: lt_address type BURS_EI_REL_ADDRESS_T,
      ls_address like line of lt_address,
      lt_phone type BUS_EI_BUPA_TELEPHONE_T,
      ls_phone type BUS_EI_BUPA_TELEPHONE,
      lt_fax type BUS_EI_BUPA_FAX_T,
      ls_fax type BUS_EI_BUPA_FAX,
      lt_email type BUS_EI_BUPA_SMTP_T,
      ls_email type BUS_EI_BUPA_SMTP,
      lt_pager type BUS_EI_BUPA_PAG_T,
      ls_pager type BUS_EI_BUPA_PAG,
      lb_contact_valid type crmt_boolean.


  DATA: lt_tel type BAPIADTEL_T,
      ls_tel type BAPIADTEL,
      lt_bapifax type BAPIADFAX_T,
      ls_bapifax type BAPIADFAX,
      lt_smtp type BAPIADSMTP_T,
      ls_smtp type BAPIADSMTP,
      lt_pag type table of bapiadpag,
      ls_pag type bapiadpag,
      lt_return type BAPIRET2_T.

DATA: lv_timestamp_valid_from  TYPE timestamp,
      lv_initial_time        TYPE syst_uzeit,
      lv_timestamp_current_utc TYPE timestamp,
      lv_initial_timezone      TYPE ttzz-tzone,
      lv_date_valid_from       TYPE sydatum,
      lv_date_UTC            TYPE sydatum.

  DATA: lv_index type integer.
* standard scenario
  DATA: ls_mkt_area TYPE ZMKT_AREAS_APPEND,
      ls_bus_ei_mkt_permissions TYPE CRMT_BUS_EI_MKT_PERMISSIONS,
      ls_mkt_permissions TYPE CRMT_BUS_EI_MKT_PERMISSIONS.
* standard scenario

  DATA: lo_zcl_crm_bupa_mktperm type ref to zcl_crm_bupa_mktperm_common.

*------------------ CRMPCD_BUPA_REL_MAP-----------------------------
 DATA ls_map_contact          TYPE crmm_but_contno.
 DATA ls_identification       TYPE bapibus1006_identification_key.
 DATA lv_org_guid             TYPE bu_partner_guid.
 DATA lv_person_guid          TYPE bu_partner_guid.
*------------------ CRMPCD_BUPA_REL_MAP-----------------------------

 FIELD-SYMBOLS <ls_data_mapped> TYPE crmxif_partner_rel_complex.

 LOOP AT cs_data_mapped ASSIGNING <ls_data_mapped>.
*------------------ CRMPCD_BUPA_REL_MAP-----------------------------
  CLEAR ls_map_contact.

  lv_org_guid   = <ls_data_mapped>-header-object_instance-partner1-partner_guid.
  lv_person_guid = <ls_data_mapped>-header-object_instance-partner2-partner_guid.

   "Is there a ERP contact ID available for the CRM contact relationship?
  CALL FUNCTION 'BUPA_MAP_CONTACT_GETDETAIL'
   EXPORTING
    iv_org_guid      = lv_org_guid
    iv_person_guid   = lv_person_guid
   IMPORTING
    es_map_contact   = ls_map_contact
   EXCEPTIONS
    no_entry_found   = 1
    invalid_parameter = 2
    OTHERS           = 3.

  CHECK sy-subrc IS INITIAL AND ls_map_contact IS NOT INITIAL.

  ls_identification-identificationcategory = 'CRM014'.
```


PUBLIC
© 2019 SAP SE or an SAP affiliate company. All rights reserved.

```
  IF mv_erp_logsys IS NOT INITIAL.
   CONCATENATE mv_erp_logsys '#' ls_map_contact-contact_no INTO ls_identification-identificationnumber.
  ELSE.
   ls_identification-identificationnumber   = ls_map_contact-contact_no.
  ENDIF.

  INSERT ls_identification INTO TABLE <ls_data_mapped>-header-object_instance-partner2-ident_numbers.
 ENDLOOP.
*----------------- CRMPCD_BUPA_REL_MAP-----------------------------


*----------------- marketing permissions -------------------------
 LOOP AT cs_data_mapped ASSIGNING <ls_data_mapped>.
* address dependent communication channel
   clear lt_mkp_applicable.
   clear ls_mkp_applicable.
   clear lt_address.
   clear lt_channel.
   UNASSIGN <mkt_permission_t>.
   UNASSIGN <mkt_permission>.
   UNASSIGN <mkt_permission_channel>.
   IF <ls_data_mapped>-HEADER-OBJECT_TASK EQ 'C'.
    lt_address = <ls_data_mapped>-CENTRAL_DATA-ADDRESS-ADDRESSES.
    check lt_address is not initial.
    READ TABLE lt_address INTO ls_address WITH KEY DATA-POSTAL-DATA-STANDARDADDRESS = 'X'.
    IF sy-subrc EQ 0.
     LOOP AT ls_address-DATA-COMMUNICATION-PHONE-PHONE INTO ls_phone
      WHERE CURRENTLY_VALID = 'X'.
       APPEND ls_phone TO lt_phone.
     ENDLOOP.

     READ TABLE ls_address-DATA-COMMUNICATION-FAX-FAX INTO ls_fax
      WITH KEY  CONTACT-DATA-STD_NO = 'X'
            CURRENTLY_VALID = 'X'.

     READ TABLE ls_address-DATA-COMMUNICATION-SMTP-SMTP INTO ls_email
      WITH KEY  CONTACT-DATA-STD_NO = 'X'
            CURRENTLY_VALID = 'X'.

     READ TABLE ls_address-DATA-COMMUNICATION-PAGER-PAGER INTO ls_pager
      WITH KEY  CONTACT-DATA-STD_NO = 'X'
            CURRENTLY_VALID = 'X'.

    ENDIF.
    ASSIGN <ls_data_mapped>-CRM_DATA-MKT_PERMISSION-MKT_PERMISSIONS TO <mkt_permission_t>.
    ls_mkp_applicable-MKT_APPLICABLE-PARTNERGUID = <ls_data_mapped>-HEADER-OBJECT_INSTANCE-
PARTNER1-PARTNER_GUID.
    ls_mkp_applicable-MKT_APPLICABLE-CONTPGUID = <ls_data_mapped>-HEADER-OBJECT_INSTANCE-
PARTNER2-PARTNER_GUID.

* standard scenario
    CREATE OBJECT LO_ZCL_CRM_BUPA_MKTPERM
     exporting
      IV_PARTNERGUID = ls_mkp_applicable-MKT_APPLICABLE-PARTNERGUID
      IV_CONTPGUID   = ls_mkp_applicable-MKT_APPLICABLE-CONTPGUID.
* standard scenario

    LOOP AT <mkt_permission_t> ASSIGNING <mkt_permission_channel>.
      IF  <mkt_permission_channel>-DATA-CHANNEL EQ 'TEL' OR
         <mkt_permission_channel>-DATA-CHANNEL EQ 'FAX' OR
         <mkt_permission_channel>-DATA-CHANNEL EQ 'INT' OR
         <mkt_permission_channel>-DATA-CHANNEL EQ 'PAG' .
         READ TABLE lt_channel WITH KEY DATA-CHANNEL = <mkt_permission_channel>-DATA-
CHANNEL TRANSPORTING NO FIELDS.
         IF sy-subrc NE 0.
          <mkt_permission_channel>-DATA-ORIGIN = ''.
          <mkt_permission_channel>-DATA-PERMISSION = ''.
          <mkt_permission_channel>-DATA-VALID_FROM = ''.
          <mkt_permission_channel>-DATA-VALUE = ''.
          APPEND <mkt_permission_channel> TO lt_channel.
         ENDIF.
      ENDIF.
```


© 2019 SAP SE or an SAP affiliate company. All rights reserved.

```
      DELETE <mkt_permission_t>.
    ENDLOOP.
    LOOP AT lt_channel INTO ls_channel.
      clear lt_mkp_applicable.
      clear lt_valid_permissions.
      clear lt_value.
      CASE ls_channel-DATA-CHANNEL.
*   TEL -> PHONE
        WHEN 'TEL'.
          LOOP AT lt_phone INTO ls_phone.
            ls_value-VALUE = ls_phone-CONTACT-DATA-TEL_NO.
            ls_value-STD = ls_phone-CONTACT-DATA-STD_NO.
            ls_value-R_3_USER = ls_phone-CONTACT-DATA-R_3_USER.
            APPEND ls_value TO lt_value.
          ENDLOOP.
*   FAX -> FAX
        WHEN 'FAX'.
          ls_value-VALUE = ls_fax-CONTACT-DATA-FAX_NO.
          ls_value-STD = ls_fax-CONTACT-DATA-STD_NO.
          APPEND ls_value TO lt_value.
*   INT -> EMAIL
        WHEN 'INT'.
          ls_value-VALUE = ls_email-CONTACT-DATA-E_MAIL.
          ls_value-STD = ls_email-CONTACT-DATA-STD_NO.
          APPEND ls_value TO lt_value.
*   PAG -> SMS
        WHEN 'PAG'.
          ls_value-VALUE = ls_pager-CONTACT-DATA-PAGER_NO.
          ls_value-STD = ls_pager-CONTACT-DATA-STD_NO.
          APPEND ls_value TO lt_value.
        WHEN OTHERS.
      ENDCASE.

      IF lt_value IS INITIAL.
        CONTINUE.
      ENDIF.

      CALL METHOD LO_ZCL_CRM_BUPA_MKTPERM->CHECK_CONTACT_ALLOWED_
        EXPORTING
          IV_CHANNEL            = ls_channel-DATA-CHANNEL
          IT_STD_VALUE          = lt_value
          IV_CHECK_ACCOUNT_CONTACT = abap_true
        CHANGING
          CT_MKTPERM_APPLICABLE    = lt_mkp_applicable
        RECEIVING
          RT_VALID_PERMISSIONS     = lt_valid_permissions.

        IF lt_valid_permissions IS INITIAL OR lt_mkp_applicable IS INITIAL.
          CONTINUE.
        ENDIF.
        LOOP AT lt_valid_permissions INTO ls_valid_permission.
          IF ls_valid_permission-VALUE IS INITIAL AND lt_value IS INITIAL.
            CONTINUE.
          ENDIF.
          APPEND INITIAL LINE TO <mkt_permission_t>  ASSIGNING <mkt_permission>.
          lv_index = sy-TABIX.
*check against mobile phone
          IF ls_channel-DATA-CHANNEL EQ 'TEL'.
            IF ls_valid_permission-VALUE IS INITIAL.
              READ TABLE lt_value INTO ls_value WITH KEY
                STD = 'X'.
              IF sy-subrc EQ 0.
                IF ls_value-R_3_USER = '2' OR ls_value-R_3_USER = '3'.
                  <mkt_permission>-DATA-CHANNEL = 'TE2'.
                ENDIF.
              ENDIF.
            ELSE.
              READ TABLE ls_address-DATA-COMMUNICATION-PHONE-PHONE INTO ls_phone
                WITH KEY CONTACT-DATA-TEL_NO = ls_valid_permission-value.
              IF sy-subrc EQ 0.
                IF ls_phone-CONTACT-DATA-R_3_USER = '2' OR ls_phone-CONTACT-DATA-R_3_USER = '3'.
                  <mkt_permission>-DATA-CHANNEL = 'TE2'.
```


PUBLIC
© 2019 SAP SE or an SAP affiliate company. All rights reserved.

```
                ENDIF.
              ENDIF.
            ENDIF.
          ENDIF.
*check against mobile phone
          IF <mkt_permission>-DATA-CHANNEL IS INITIAL.
            <mkt_permission>-DATA-CHANNEL = ls_channel-DATA-CHANNEL.
          ENDIF.
*     simulate CPI conversion of time stamp; add initial time to timestamp valid from date is not in future
                CONVERT DATE ls_valid_permission-valid_from TIME lv_initial_time INTO TIME STAMP
    lv_timestamp_valid_from TIME ZONE lv_initial_timezone.
*     get current timestamp in utc
                GET TIME STAMP FIELD  lv_timestamp_current_utc.

*     get dates from timestamps initial timezone is UTC
                CONVERT TIME STAMP lv_timestamp_valid_from  TIME ZONE lv_initial_timezone INTO DATE
    lv_date_valid_from.
                CONVERT TIME STAMP lv_timestamp_current_utc TIME ZONE lv_initial_timezone INTO DATE lv_date_utc.

*     if currecnt UTC date is smaller / valid from date is in future => current UTC should be passed
                IF lv_date_valid_from > lv_date_utc.
                  ls_valid_permission-valid_from = lv_date_utc.
                ENDIF.

          <mkt_permission>-DATA-VALID_FROM = ls_valid_permission-VALID_FROM.
          READ TABLE lt_mkp_applicable INTO ls_mkp_applicable
           WITH KEY MKT_APPLICABLE-PARTNERGUID = ls_mkp_applicable-MKT_APPLICABLE-PARTNERGUID
                 MKT_APPLICABLE-CONTPGUID   = ls_mkp_applicable-MKT_APPLICABLE-CONTPGUID
                 VALUE = ls_valid_permission-VALUE
                 RECORD_GUID = LS_VALID_PERMISSION-RECORD_GUID.
          IF ls_mkp_applicable-MKT_APPLICABLE-CONTACT_ALLOWED EQ 'X'.
           <mkt_permission>-DATA-PERMISSION = '001'.
          ELSE.
           <mkt_permission>-DATA-PERMISSION = '002'.
          ENDIF.

          <mkt_permission>-DATA-ORIGIN = ls_valid_permission-ORIGIN.
          <mkt_permission>-DATA_KEY-RECORD_GUID = ls_valid_permission-RECORD_GUID.

          IF ls_valid_permission-VALUE IS INITIAL.
            READ TABLE lt_value INTO ls_value WITH KEY STD = 'X'.
            IF sy-subrc EQ 0.
              <mkt_permission>-DATA-VALUE = ls_value-VALUE.
*something wrong, remove the permissions
            ELSE.
              DELETE <MKT_PERMISSION_T> INDEX lv_index.
              CONTINUE.
            ENDIF.
          ELSE.
           READ TABLE lt_value WITH KEY VALUE = LS_VALID_PERMISSION-VALUE TRANSPORTING NO FIELDS.
           IF sy-subrc EQ 0.
            <mkt_permission>-DATA-VALUE = ls_valid_permission-VALUE.
*something wrong, remove the permissions
           ELSE.
            DELETE <MKT_PERMISSION_T> INDEX lv_index.
            CONTINUE.
           ENDIF.
          ENDIF.

        ENDLOOP.
      ENDLOOP.

*move duplicates
* standard scenario
      DELETE ADJACENT DUPLICATES from <MKT_PERMISSION_T> COMPARING DATA-CHANNEL DATA-
PERMISSION DATA-VALID_FROM DATA-VALUE.
* standard scenario
```


© 2019 SAP SE or an SAP affiliate company. All rights reserved.

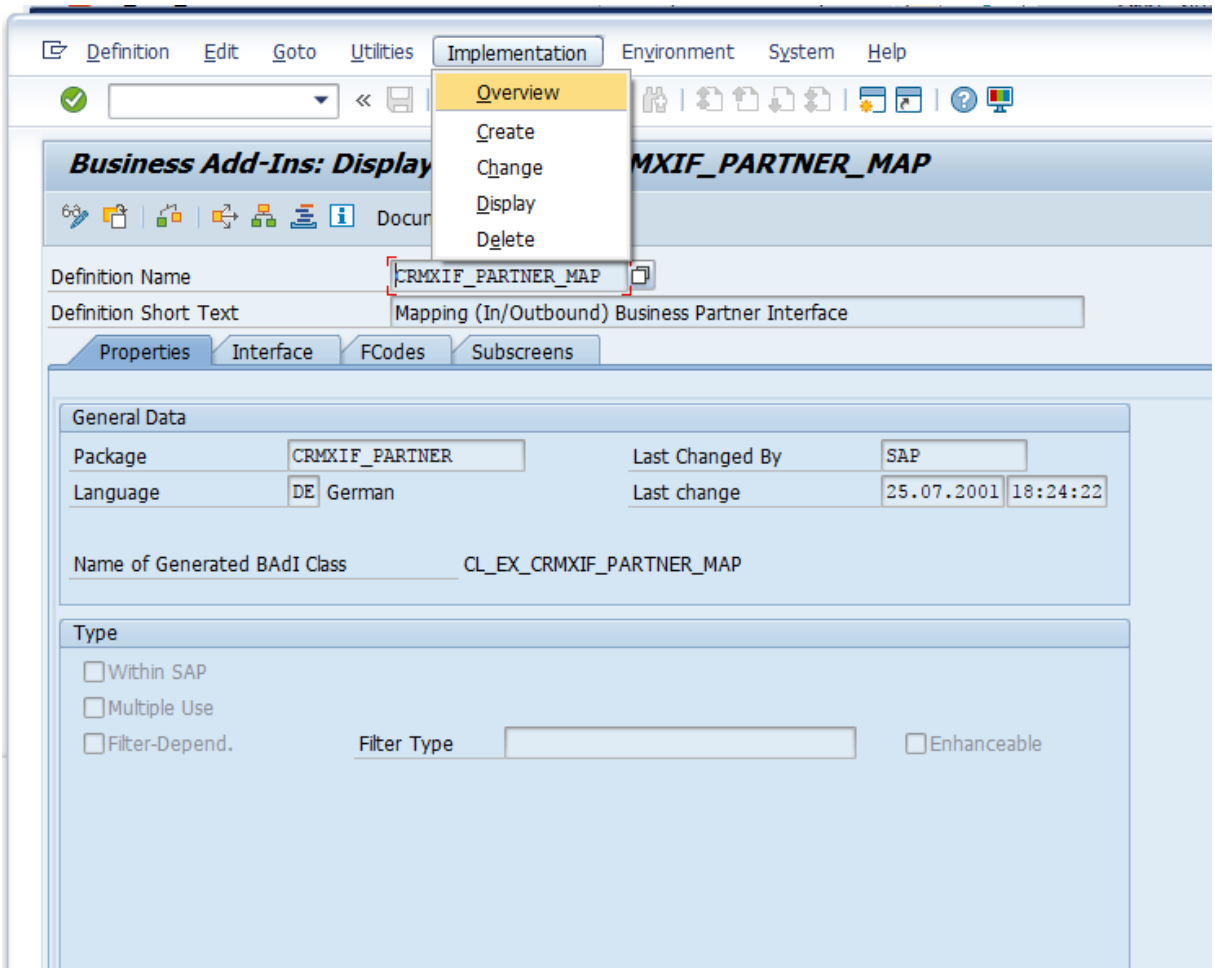
```
*move all redundant entries
    LOOP AT <mkt_permission_t> ASSIGNING <mkt_permission_channel>.
     LOOP AT <mkt_permission_t> ASSIGNING <mkt_permission>
      WHERE DATA-CHANNEL    EQ <mkt_permission_channel>-DATA-CHANNEL
       AND  DATA-VALUE     EQ <mkt_permission_channel>-DATA-VALUE.
*move all older entries
        IF <mkt_permission>-DATA-VALID_FROM LT <mkt_permission_channel>-DATA-VALID_FROM.
         DELETE <mkt_permission_t>.
        ELSE.
         IF <mkt_permission>-DATA-VALID_FROM EQ <mkt_permission_channel>-DATA-VALID_FROM.
           IF <MKT_PERMISSION>-DATA-PERMISSION EQ cl_crm_bupa_mktperm_tools=>GC_GIVEN AND
             <MKT_PERMISSION_CHANNEL>-DATA-PERMISSION EQ cl_crm_bupa_mktperm_tools=>GC_REJECTED.
            DELETE <mkt_permission_t>.
           ENDIF.
         ENDIF.
        ENDIF.
      ENDLOOP.
    ENDLOOP.
*move all redundant entries
*  ------------------ marketing permissions ---------------------------
   ENDIF.
 ENDLOOP.
endmethod.
```

15. Activate your implementation.

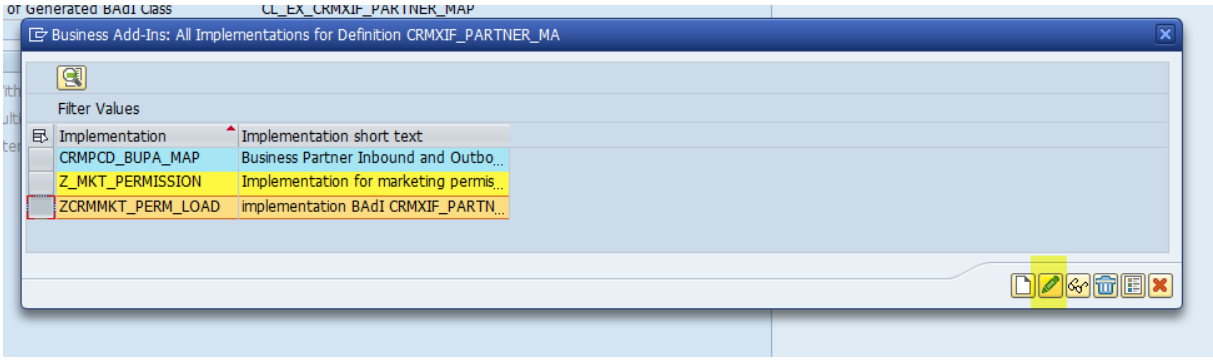
2.5 Activate BAdI Implementations

1. Start transaction SE18.
2. Display BAdI CRMXIF_PARTNER_MAP.
3. Choose *Implementation->Overview*.

PUBLIC
© 2019 SAP SE or an SAP affiliate company. All rights reserved.

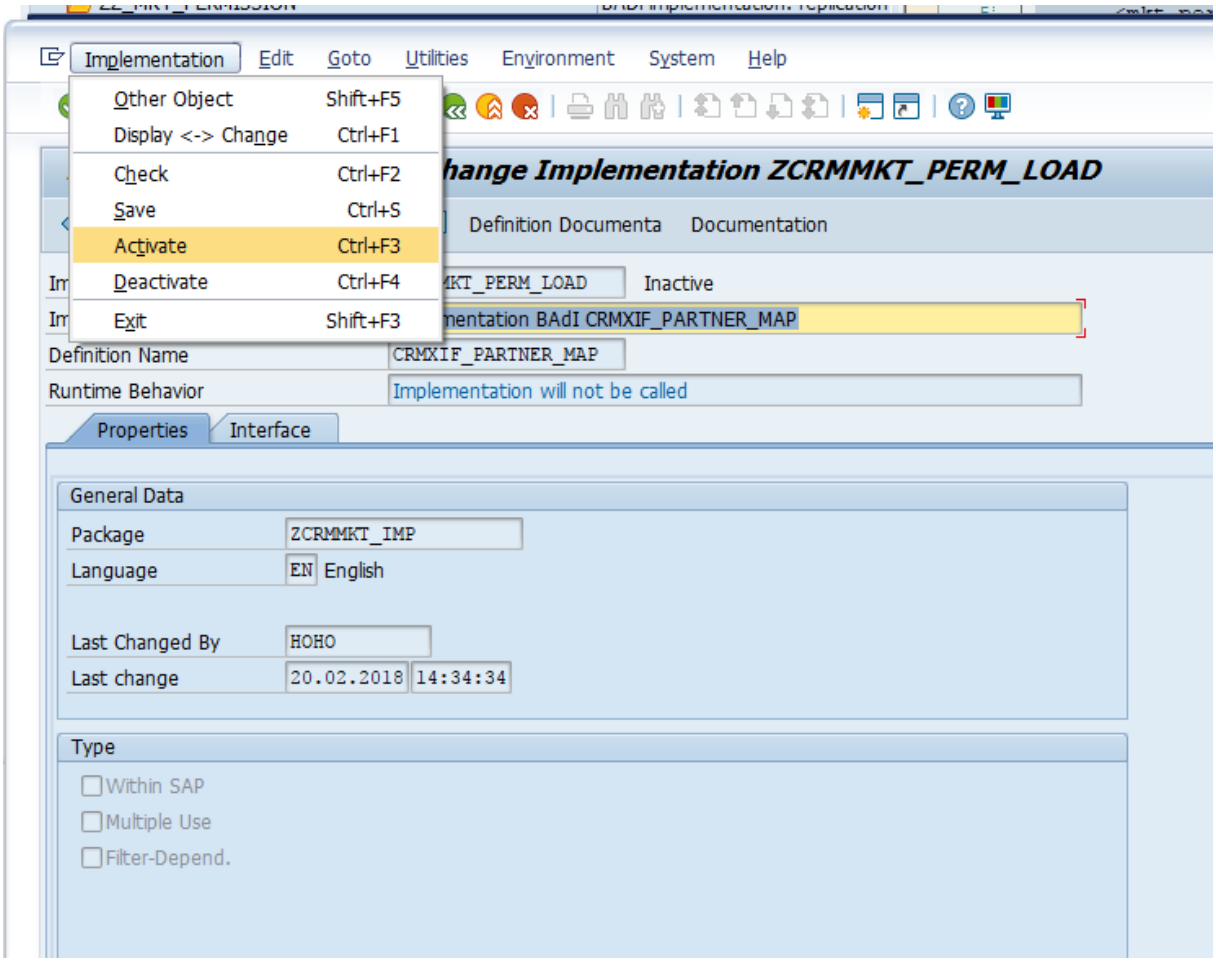


4. Select the implementation and click on *Change* (F6).



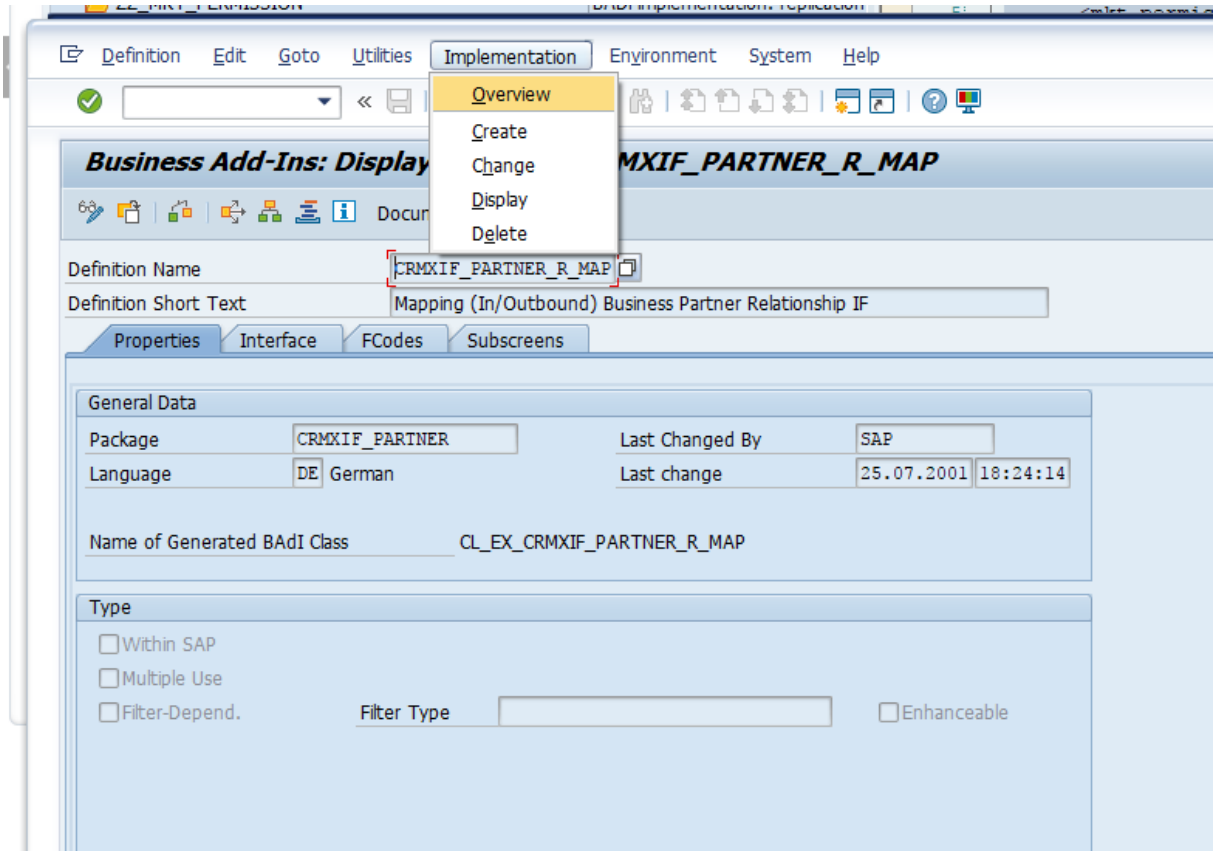
5. Choose *Implementation->Activate* (Ctrl + F3).

© 2019 SAP SE or an SAP affiliate company. All rights reserved.

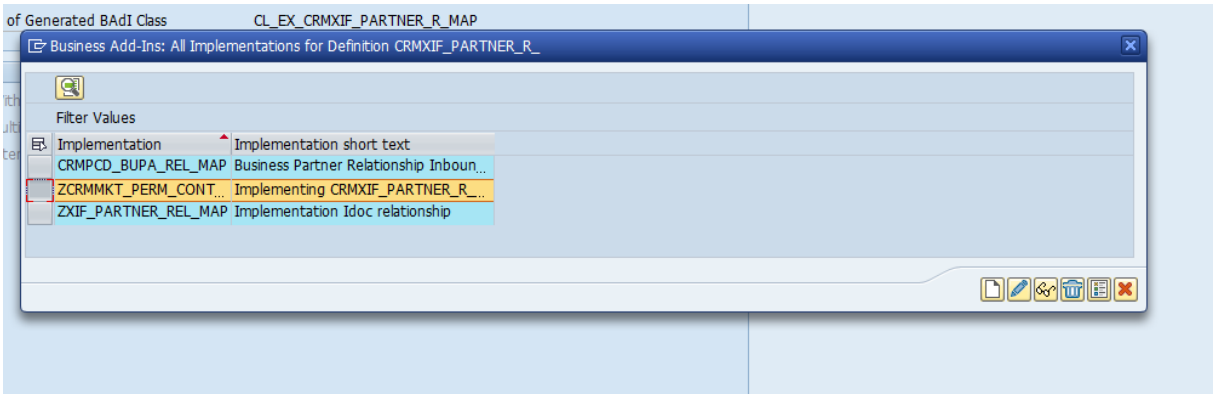


6. Go back to SE18.
7. Display BAdI CRMXIF_PARTNER_R_MAP.
8. Choose *Implementation->Overview.*

© 2019 SAP SE or an SAP affiliate company. All rights reserved.

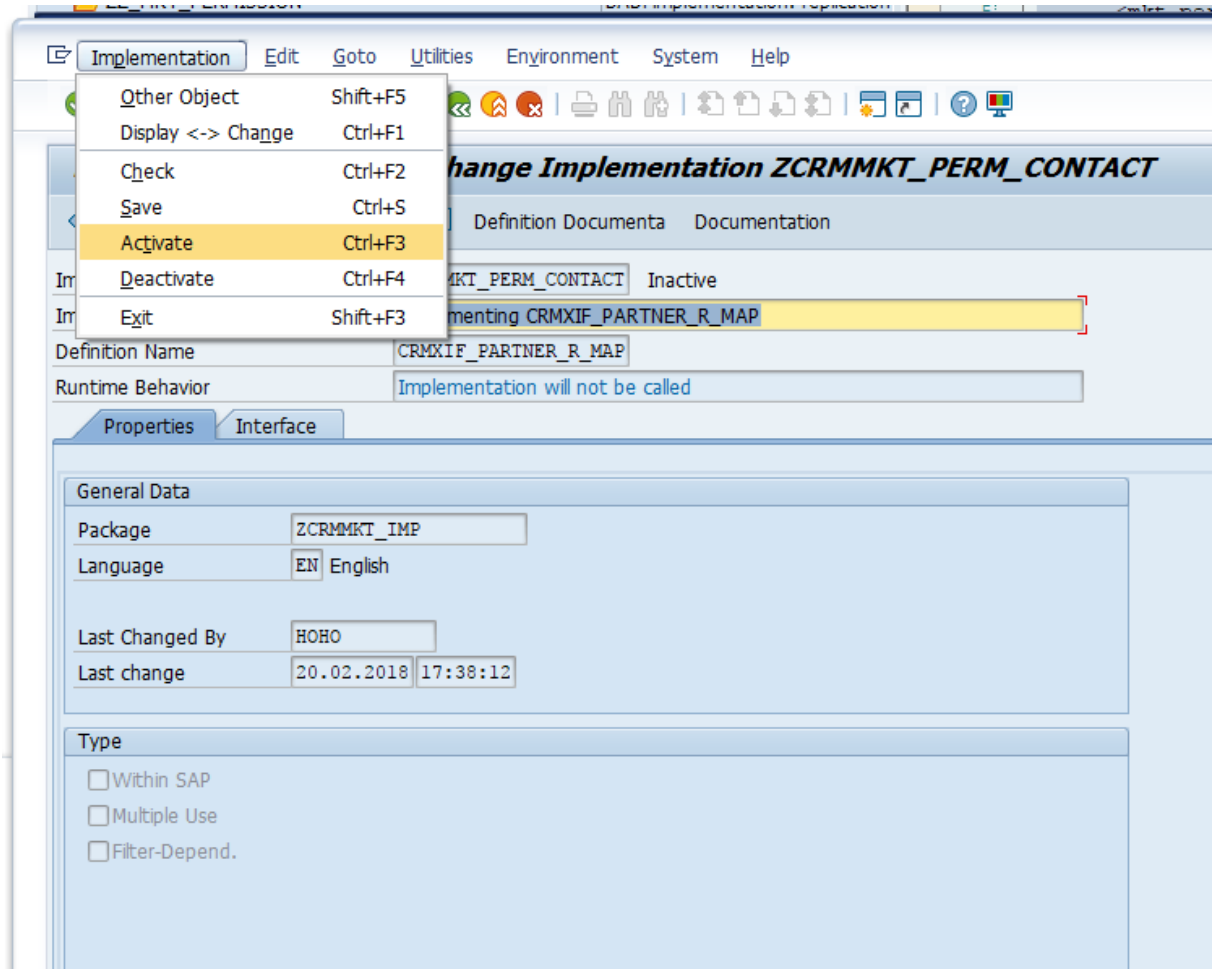


9. Select the implementation and click on *Change* (F6).



10. Choose *Implementation->Activate* (Ctrl + F3).

© 2019 SAP SE or an SAP affiliate company. All rights reserved.



3 Extended Integration Scenario for Marketing Areas Determination

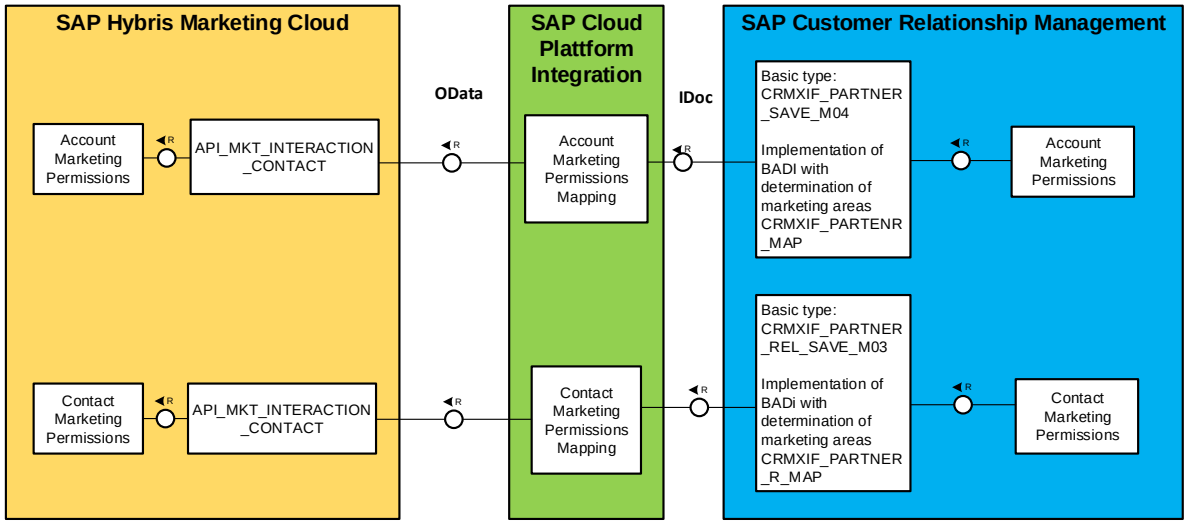
3.1 Overview

In the extended integration scenario, marketing permissions from SAP CRM will be initially loaded to SAP Marketing Cloud with marketing area information. Marketing area determination and mapping will happen on SAP CRM side. The use of marketing areas in the execution of campaigns can be activated in SAP Marketing Cloud.

Supported are:

- Marketing permissions that are assigned to marketing organization in organization unit model.
- The flag "Activate Marketing Area for Campaign Execution in SAP Marketing is activated,

© 2019 SAP SE or an SAP affiliate company. All rights reserved.



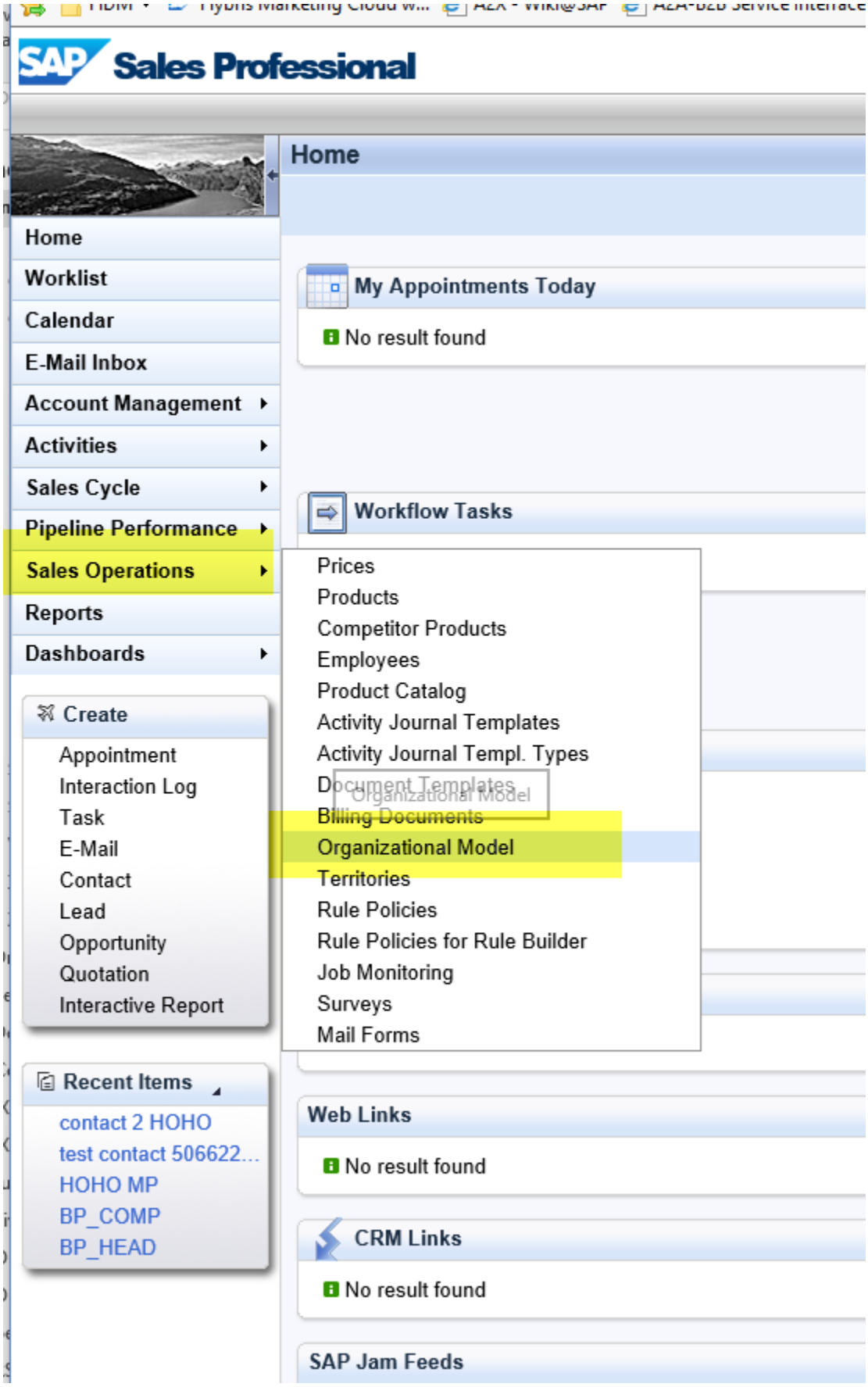
3.2 Prerequisite

- Marketing organization must be maintained in Organization Unit Model.
- Employee responsible for marketing must be assigned to marketing organization and maintained in master data of accounts (corporate, individual) or contacts.
- In master data of accounts (corporate, individual) or contacts there must be at least one standard employee responsible assigned.
- BAdI implementation in standard integration scenario described in chapter 2.2 might have been done before and must be available.

3.3 Maintain Marketing Organization

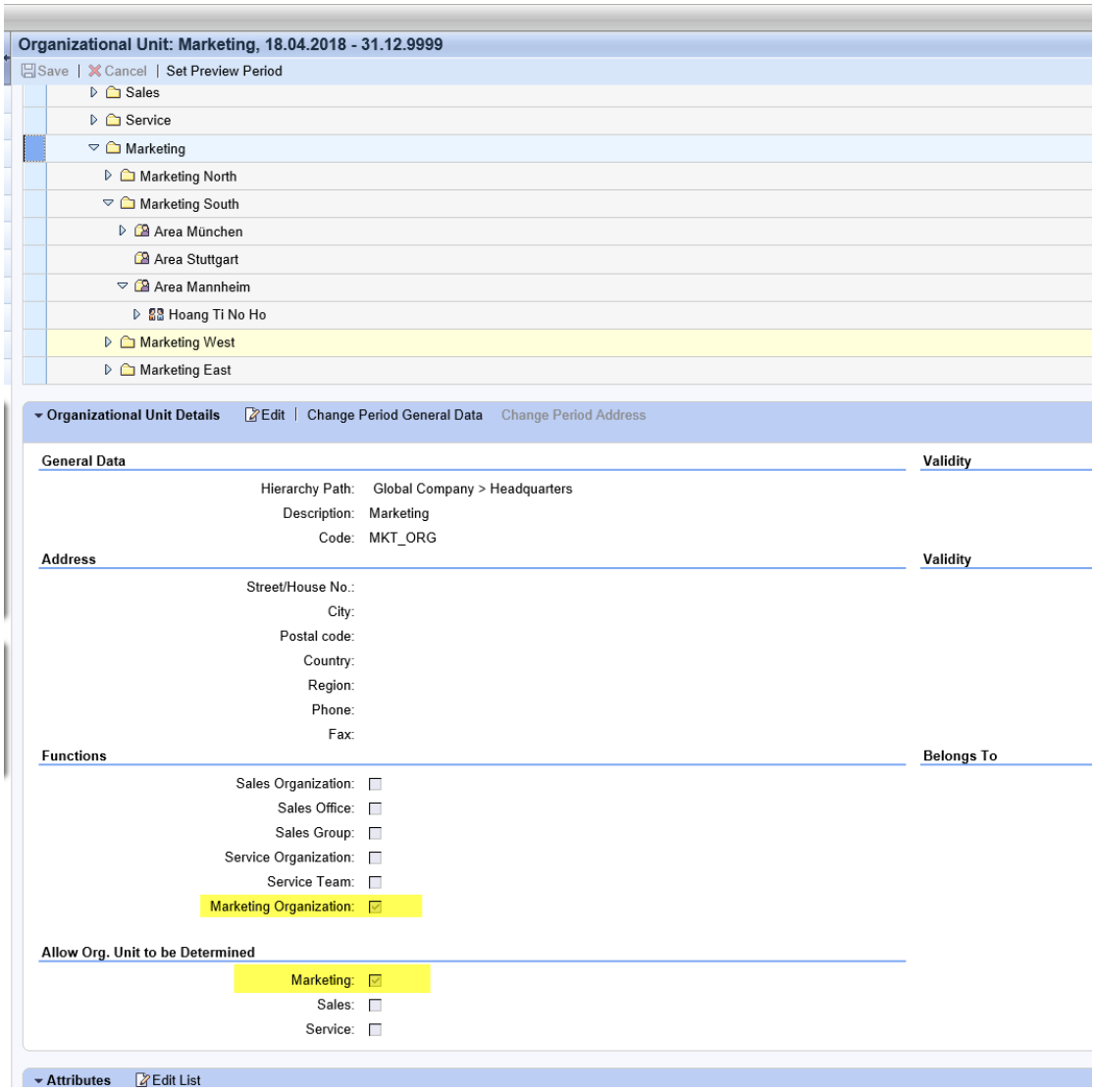
1. Start the WebUI with transaction *WUI.*
2. Choose *Sales Operations -> Organizational Model.*

© 2019 SAP SE or an SAP affiliate company. All rights reserved.



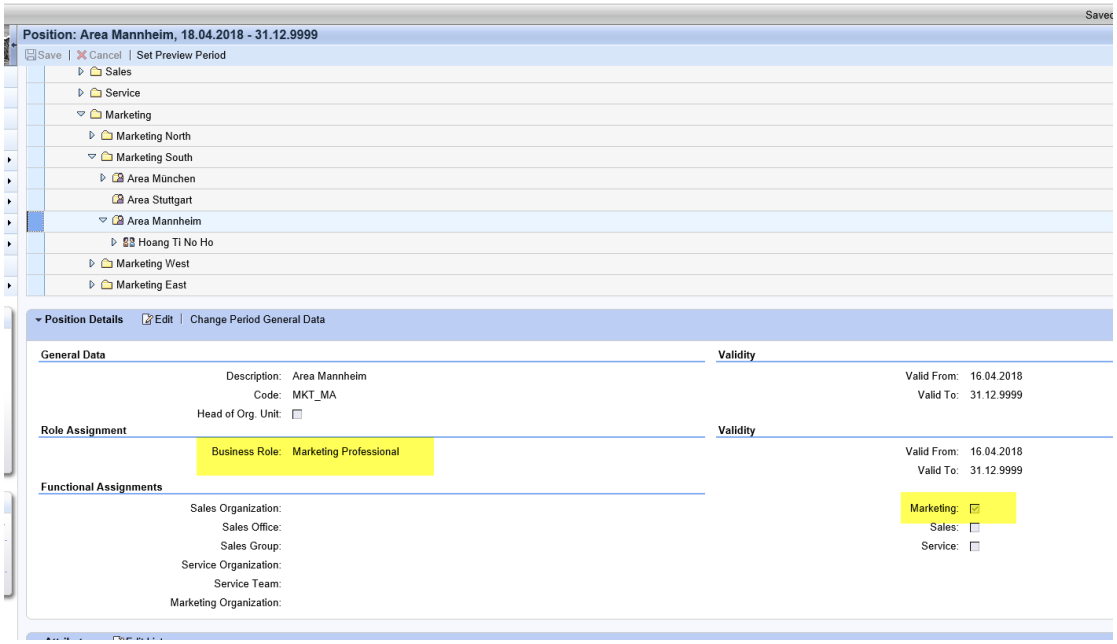
© 2019 SAP SE or an SAP affiliate company. All rights reserved.

3. Search for the existing organizational unit you want to model.
4. Click *Organizational Unit New*.
5. Enter description and code.
6. Activate the flag *Marketing* in the section *Allow Org. Unit to be Determined*-
7. Activate the flag *Marketing Organization* in the section *Functions*.

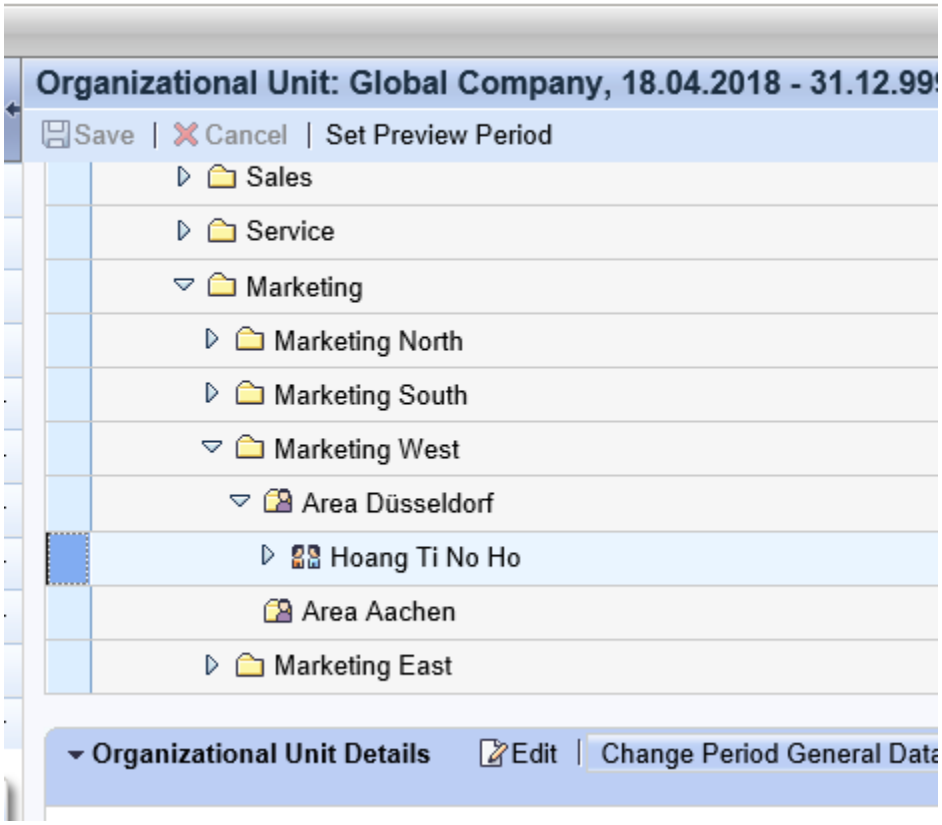


8. Create a subnode organizational unit.
9. Create a position subnode under the node marketing organization.
10. Enter description, *code.*
11. Activate the flag *Marketing* in the section *Functional Assignments*.
12. Select the business role *Marketing Professional*.

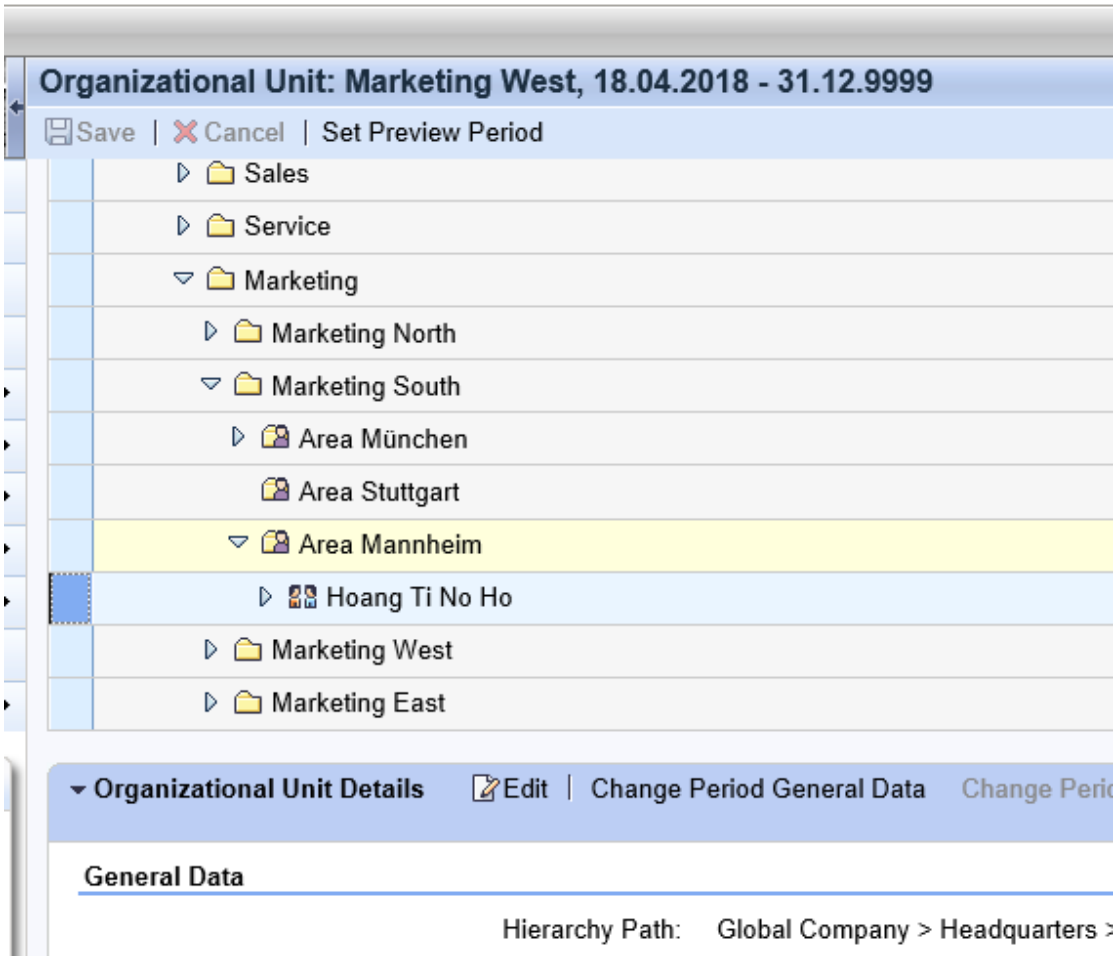
© 2019 SAP SE or an SAP affiliate company. All rights reserved.



13. Create an employee sub node under the position node.
14. Search for the employee responsible.
15. Assign the employee responsible to the position node.



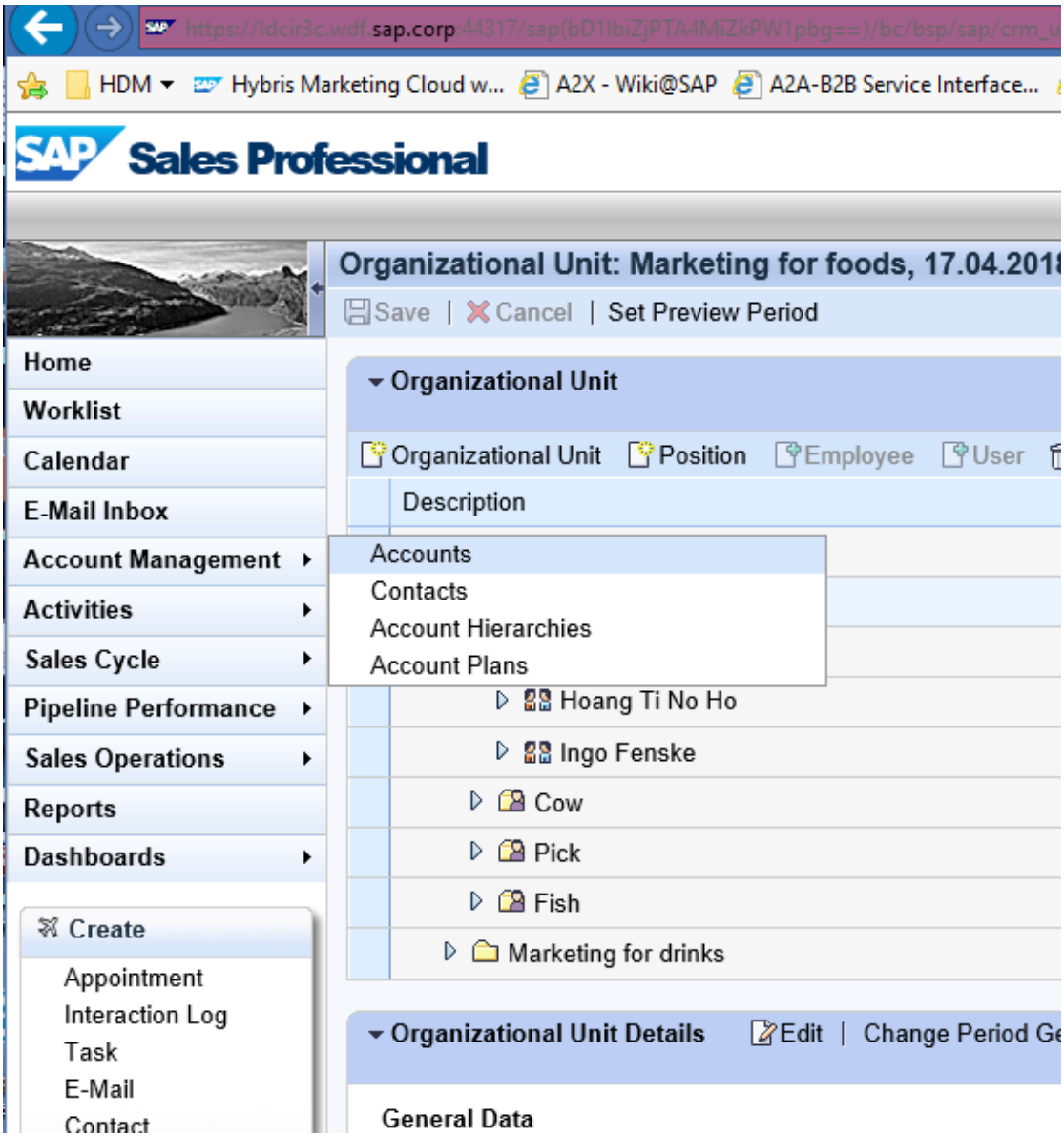
© 2019 SAP SE or an SAP affiliate company. All rights reserved.



3.4 Maintain Employee Responsible in Master Data

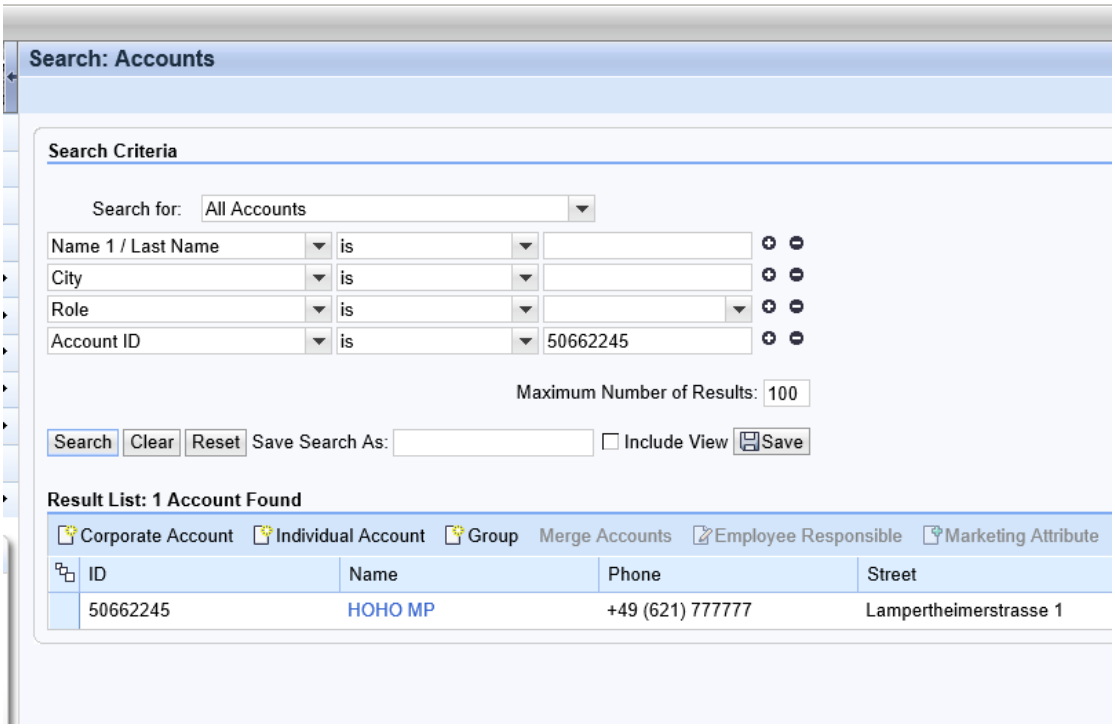
1) Choose *Account Management -> Accounts* or *Account Management -> Accounts Contacts.*

© 2019 SAP SE or an SAP affiliate company. All rights reserved.



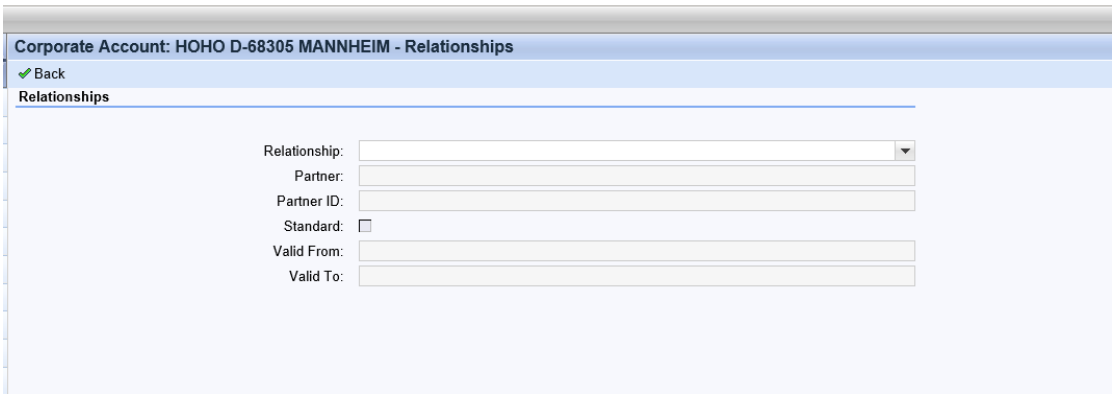
2) Search for your account or contact.

© 2019 SAP SE or an SAP affiliate company. All rights reserved.



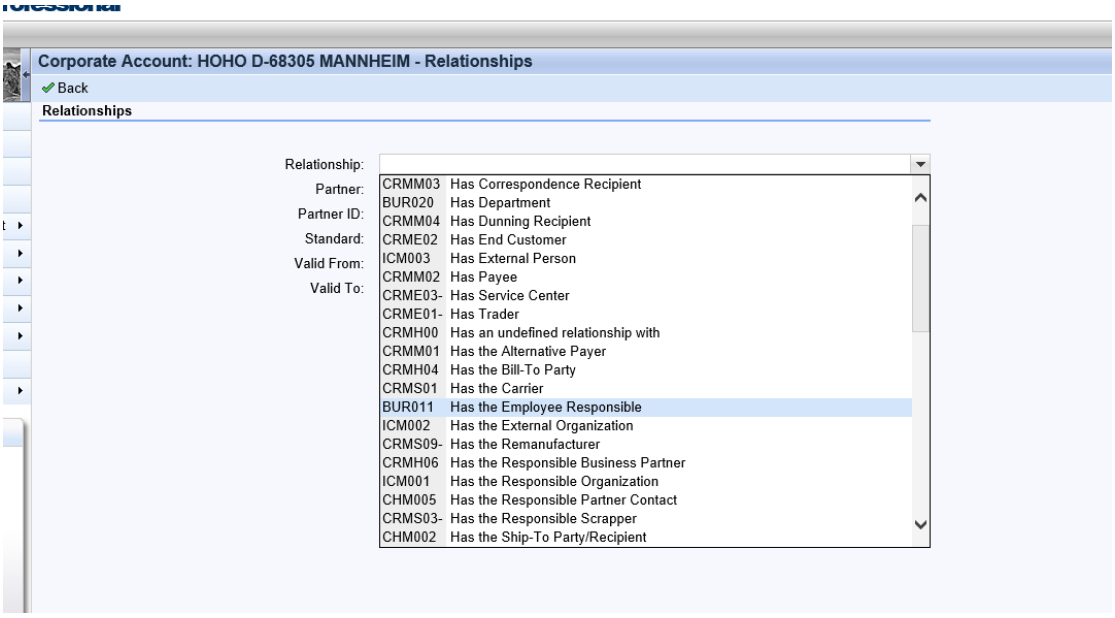
3) Go to details by clicking on the link of account or contact.
4) Go to the section *Relationships* and expand it.

5) Create a new relationship by clicking on the button *New*.

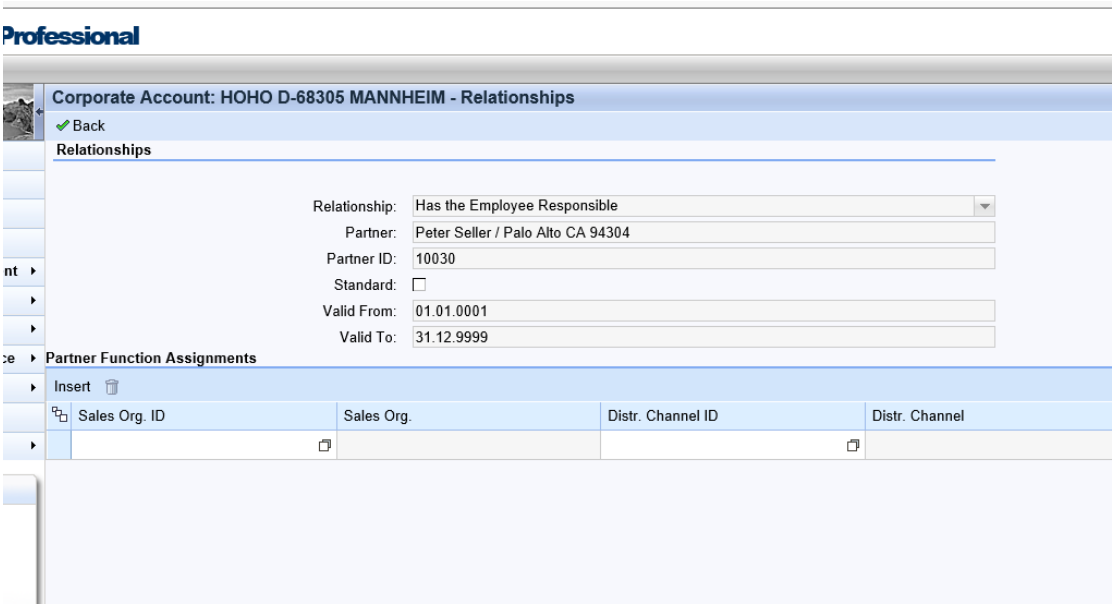


6) Select the relationship *Has the Employee Responsible*.

PUBLIC
© 2019 SAP SE or an SAP affiliate company. All rights reserved.



7) Select the employee responsible.



8) Activate the flag *Standard*, if the employee responsible should be the main responsible (at least there must be a main responsible).
9) Click on the button *Back*.
10) Save your entries.

3.5 Dictionary Objects

1. Database table: create a new mapping table Z or Y<MKTAREA_MAP> for marketing area in SAP Marketing Cloud.

© 2019 SAP SE or an SAP affiliate company. All rights reserved.

Transparent Table: ZZMKTAREA_MAP — Active

Short Description: mapping table crm org unit <-> ymkt area

Attributes | Delivery and Maintenance | Fields | Entry help/check | Currency/Quantity Fields

Srch Help | Predefined Type

Field	Key	Ini...	Data element	Data Type	Length	Deci...	Short Description
CLIENT	☑	☑	MANDT	CLNT	3	0	Client
ZZMKT_ORG	☑	☑	CRM_MKTPL_MORG	CHAR	14	0	Marketing Organization
ZZYMKT_AREA	☐	☐	CHAR40	CHAR	40	0	Character field of length 40

Dictionary: Display Technical Settings

Revised<->Active

Name: ZZMKTAREA_MAP — Transparent Table

Short Descript.: mapping table crm org unit <-> ymkt area

Last Changed: HOHO — 17.04.2018

Status: Actv. — Saved

General Properties | DB-Specific Properties

Logical Storage Parameters

Data Class: APPL2 — Organization and customizing

Size Category: 0 — Exected data records 0 to 5.300

Buffering

- (•) Buffering Not Allowed
- ( ) Buffering allowed but switched off
- ( ) Buffering Activated

Buffering Type

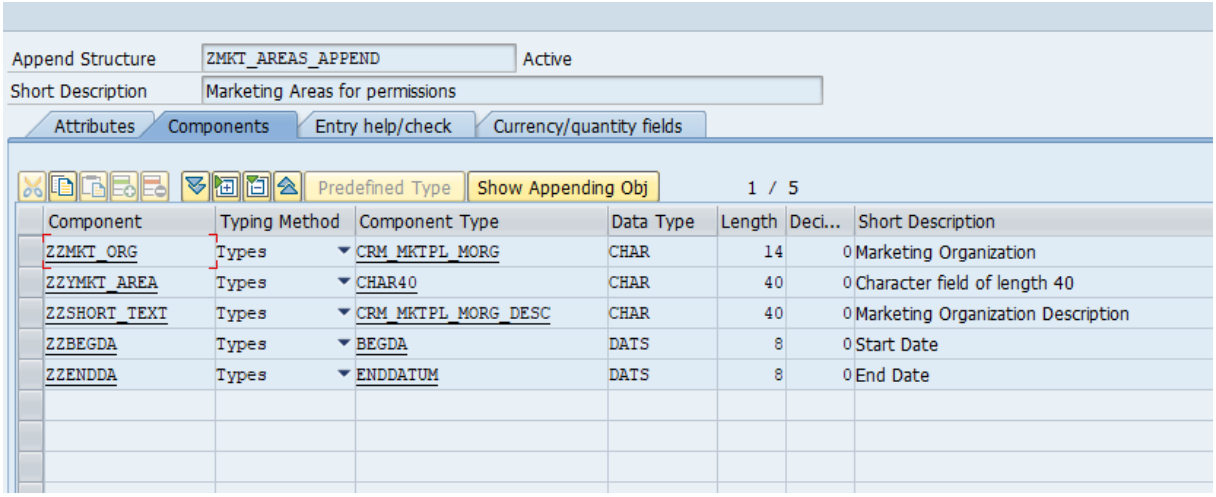
- ☐ Single Records Buff.
- ☐ Generic Area Buffered
- ☐ Fully Buffered

Number of Key Fields: 0

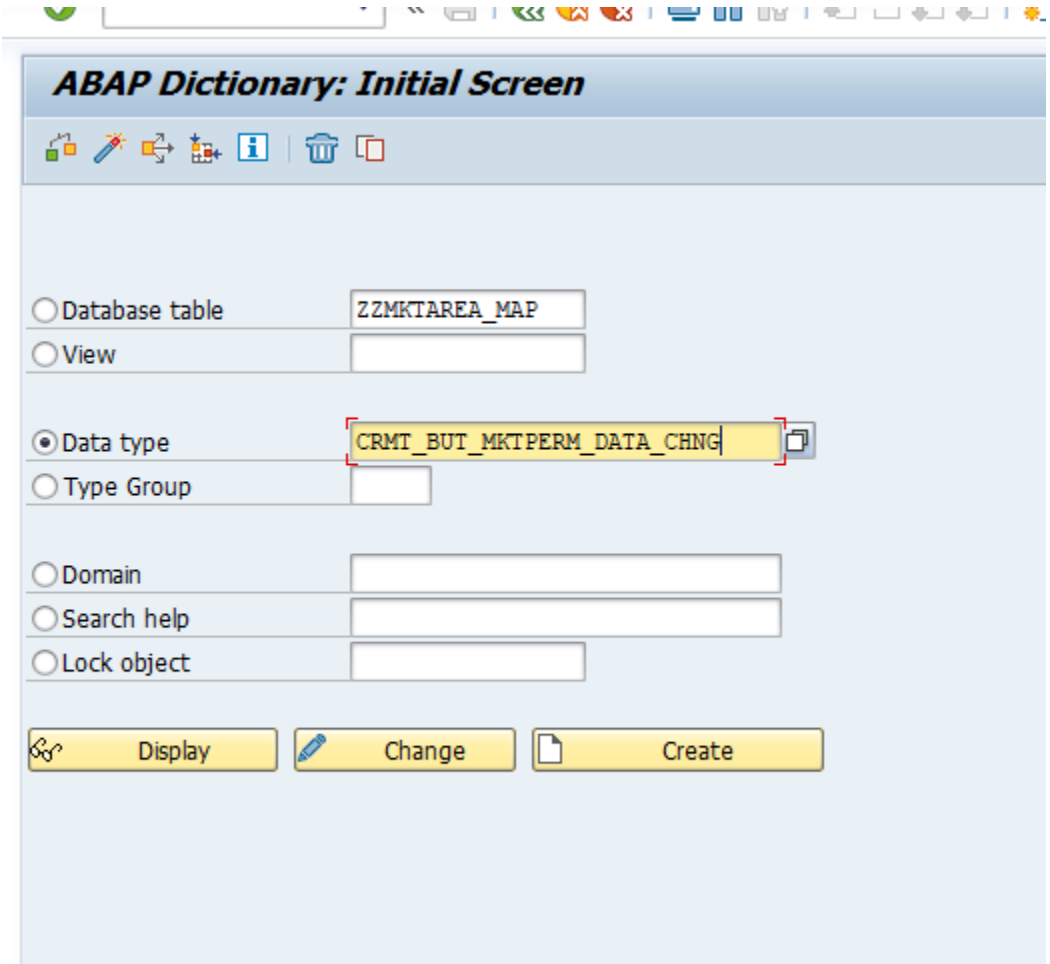
- ☐ Log Data Changes
- ☐ Write access only with JAVA

PUBLIC
© 2019 SAP SE or an SAP affiliate company. All rights reserved.

2. Create a new append structure Z or Y<MKT_AREAS_APPEND>.

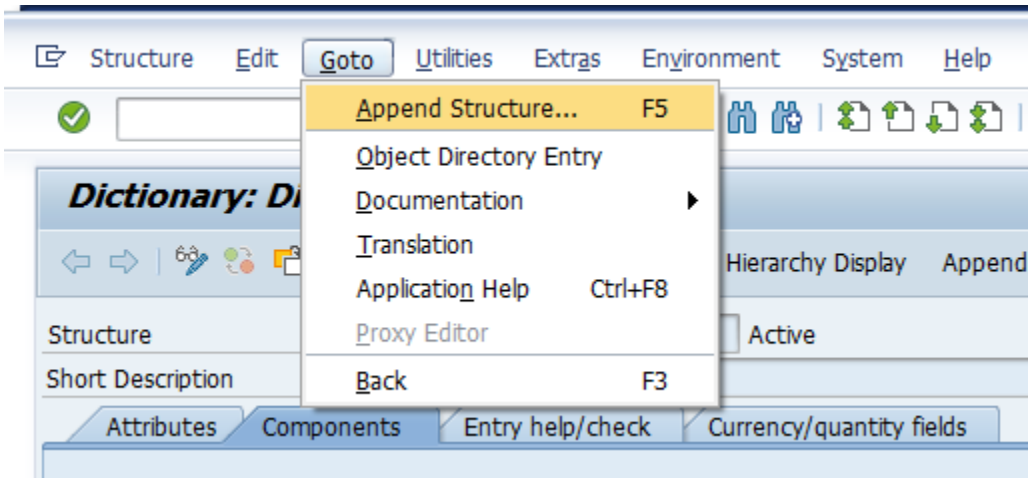


3. Start transaction SE11.
4. Select Data type and enter CRMT_BUT_MKTPERM_DATA_CHNG.
5. Click on *Display.*

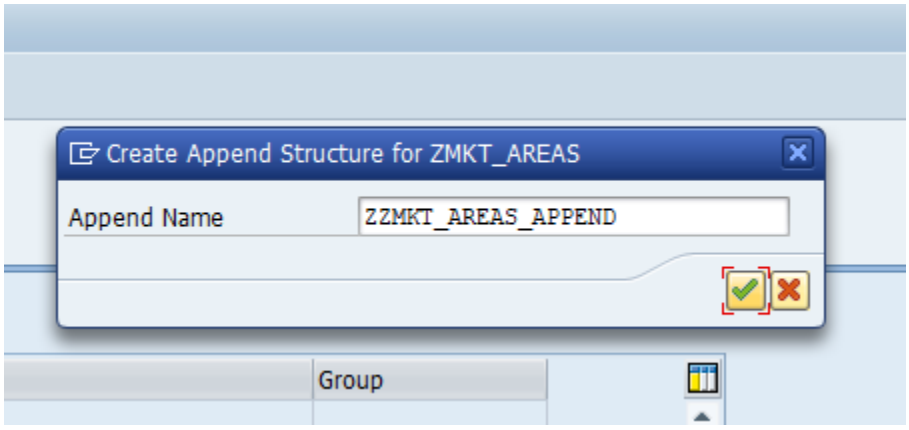


6. Choose *Goto->Append Structure.*

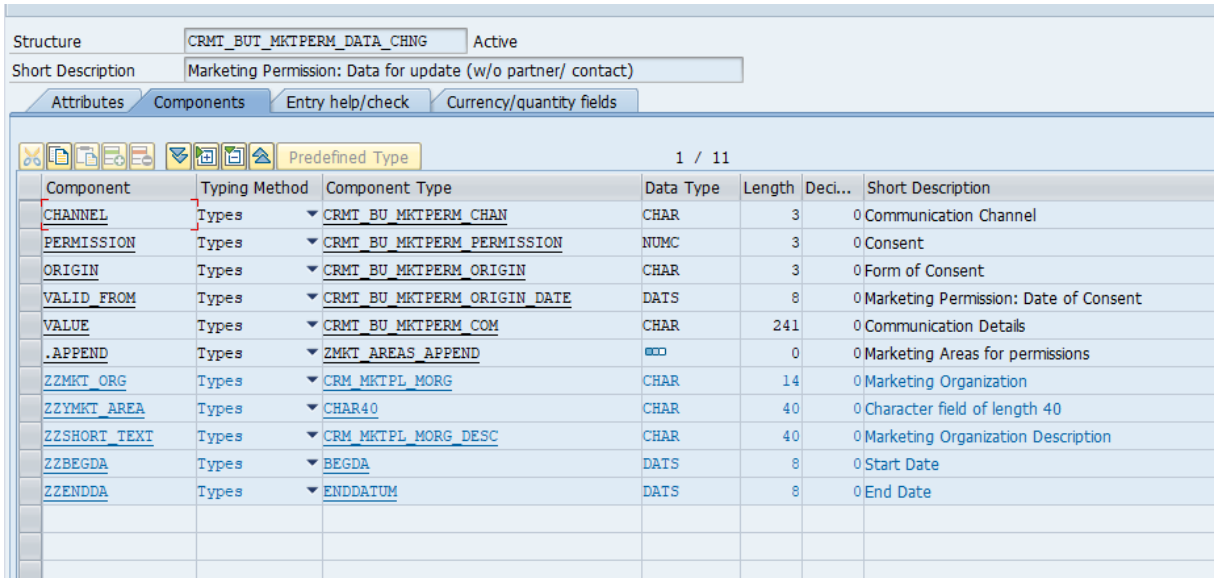
© 2019 SAP SE or an SAP affiliate company. All rights reserved.



7. Enter append structure Z or Y<MKT_AREAS_APPEND> created before in the field *Append Name*.



8. Click on *continue.*
9. *Save* and *activate*.



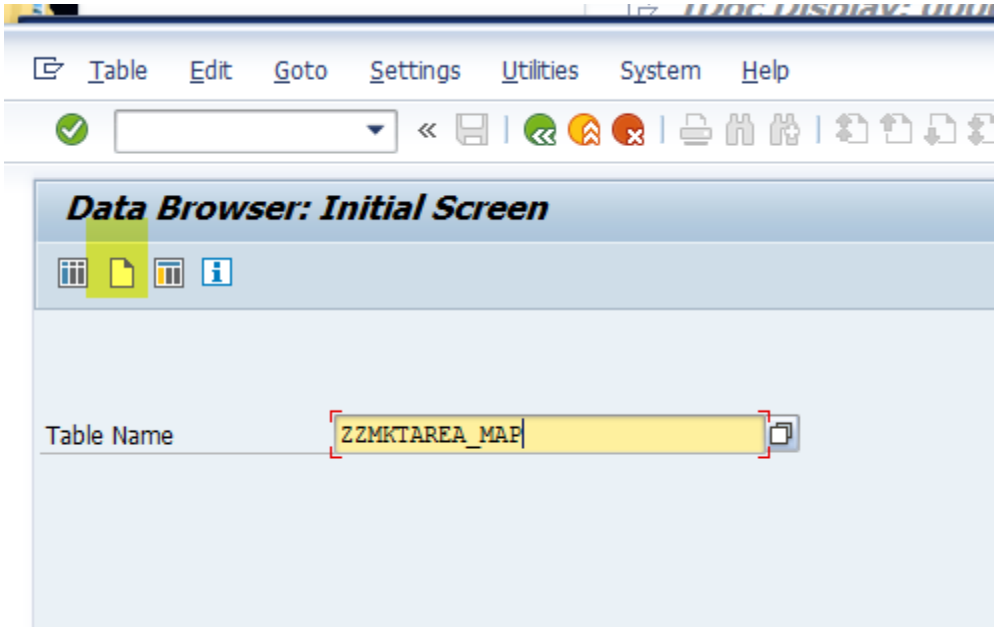
Component	Typing Method	Component Type	Data Type	Length	Deci...	Short Description
CHANNEL	Types	CRMT_BU_MKTPERM_CHAN	CHAR	3	0	Communication Channel
PERMISSION	Types	CRMT_BU_MKTPERM_PERMISSION	NUMC	3	0	Consent
ORIGIN	Types	CRMT_BU_MKTPERM_ORIGIN	CHAR	3	0	Form of Consent
VALID_FROM	Types	CRMT_BU_MKTPERM_ORIGIN_DATE	DATS	8	0	Marketing Permission: Date of Consent
VALUE	Types	CRMT_BU_MKTPERM_COM	CHAR	241	0	Communication Details
.APPEND	Types	ZMKT_AREAS_APPEND		0	0	Marketing Areas for permissions
ZZMKT_ORG	Types	CRM_MKTPL_MORG	CHAR	14	0	Marketing Organization
ZZYMKT_AREA	Types	CHAR40	CHAR	40	0	Character field of length 40
ZZSHORT_TEXT	Types	CRM_MKTPL_MORG_DESC	CHAR	40	0	Marketing Organization Description
ZZBEGDA	Types	BEGDA	DATS	8	0	Start Date
ZZENDDA	Types	ENDDATUM	DATS	8	0	End Date

3.6 Maintain the Mapping Table Z or Y<MKTAREA_MAP>

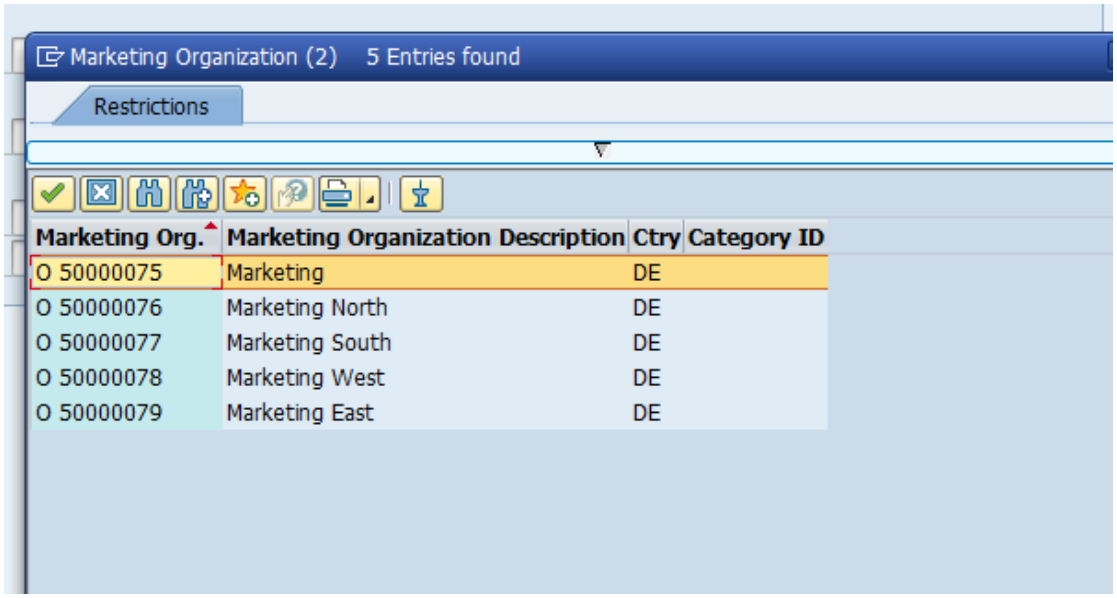
1. Start the transaction SE16.

© 2019 SAP SE or an SAP affiliate company. All rights reserved.

2. Enter the table name Z or Y<MKTAREA_MAP> and press the button *Create Entries*.

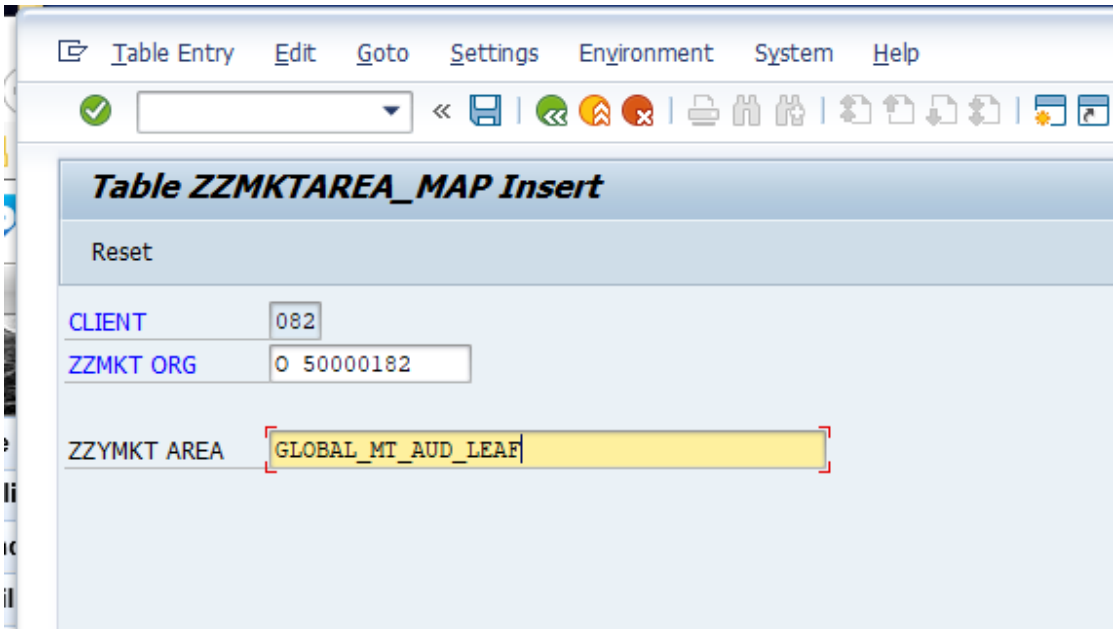


3. Select a marketing organization in the list of F4 value help.

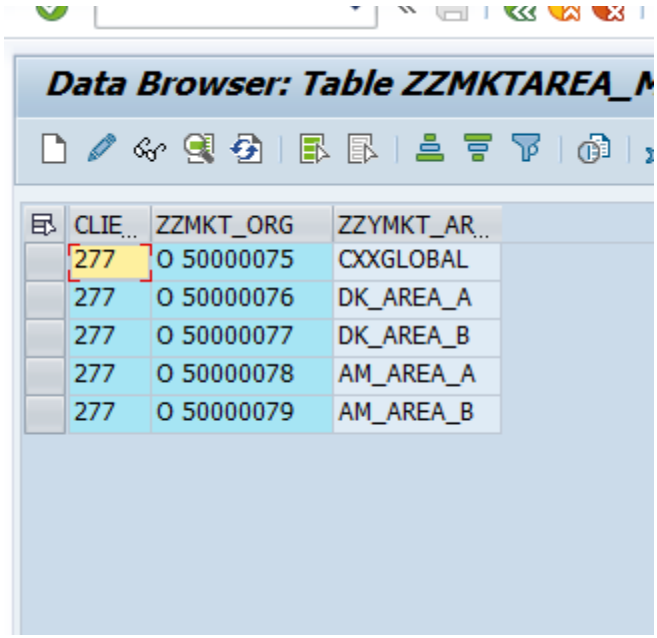


4. Enter a mapping value for marketing area in SAP Hybris Marketing Cloud.

© 2019 SAP SE or an SAP affiliate company. All rights reserved.



5. Save your entries.

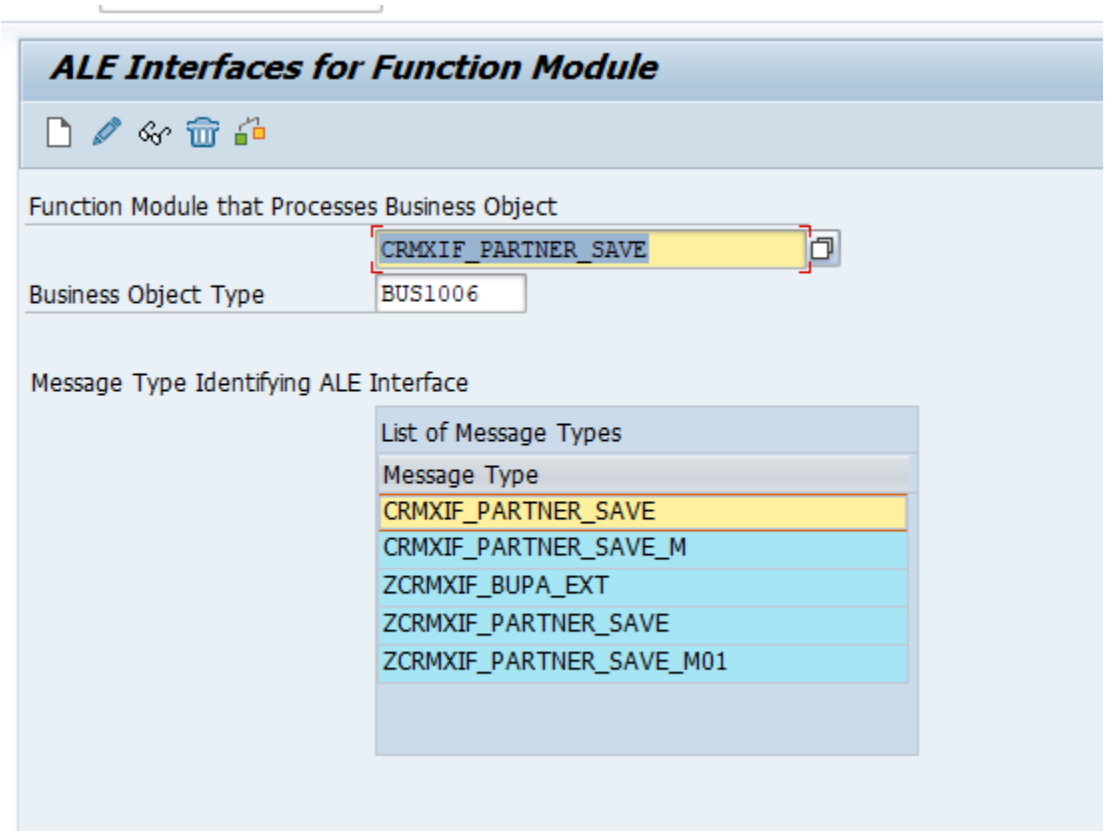


3.7 IDoc Generation

Business partner

1. Create a new function group Z or Y<ALE_BUPA_MKT_EXT> and activate it.
2. Start transaction BDFG.
3. Enter function module as CRMXIF_PARTNER_SAVE and press *Enter*. The corresponding Business Object and relevant Message Type are displayed.

© 2019 SAP SE or an SAP affiliate company. All rights reserved.



4. From the menu, choose *Interface -> Create*.
5. Enter the parameters as shown below for Outbound Processing. Leave the Inbound Processing parameters unpopulated.

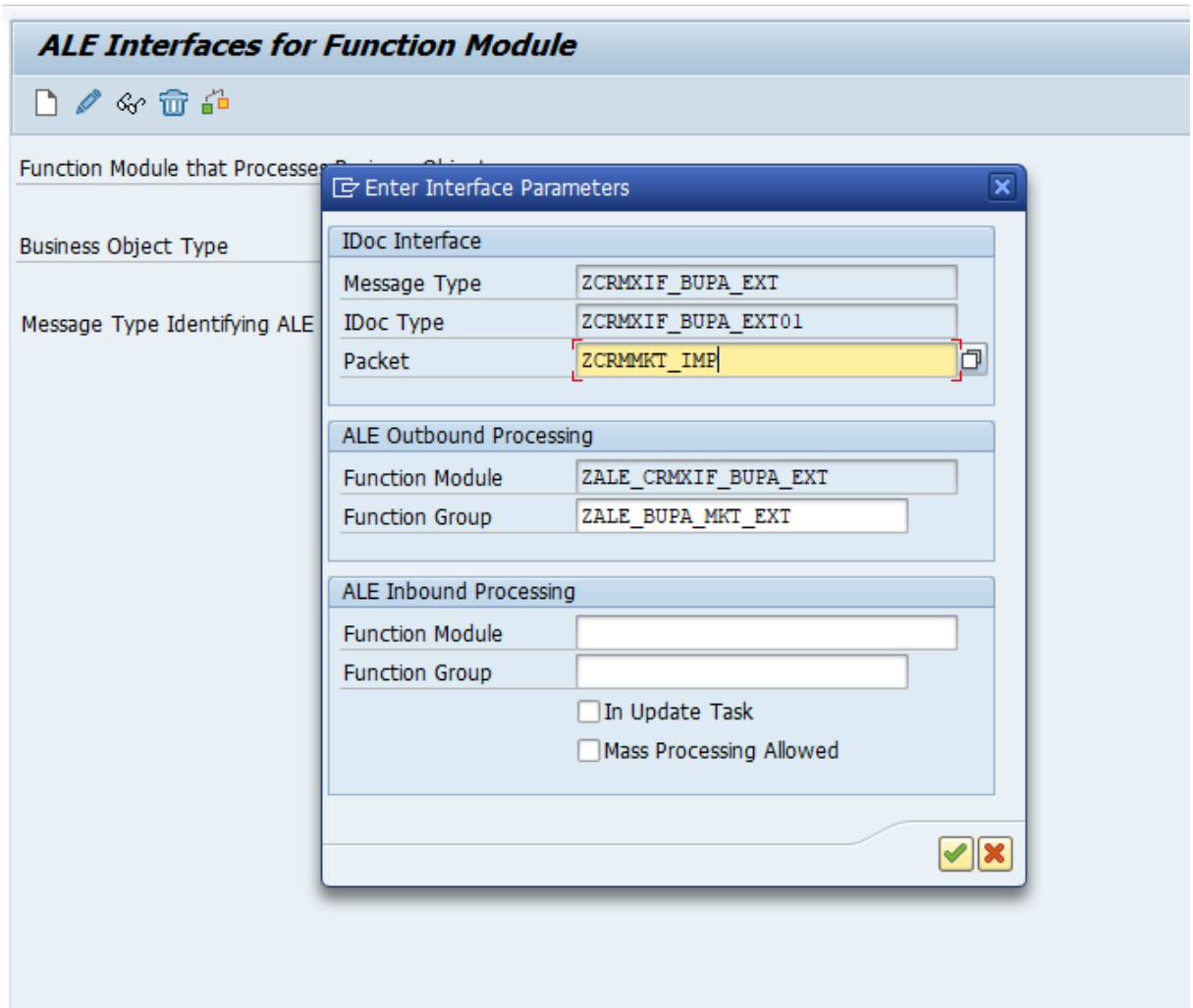
   Message type: Z or Y<CRMXIF_BUPA_EXT>

   IDoc type: Z or Y<CRMXIF_BUPA_EXT01>

   Package: Z or Y<CRMMKT_IMP> (see 2.2.1)

   Function group: Z or Y<ALE_BUPA_MKT_EXT>

© 2019 SAP SE or an SAP affiliate company. All rights reserved.

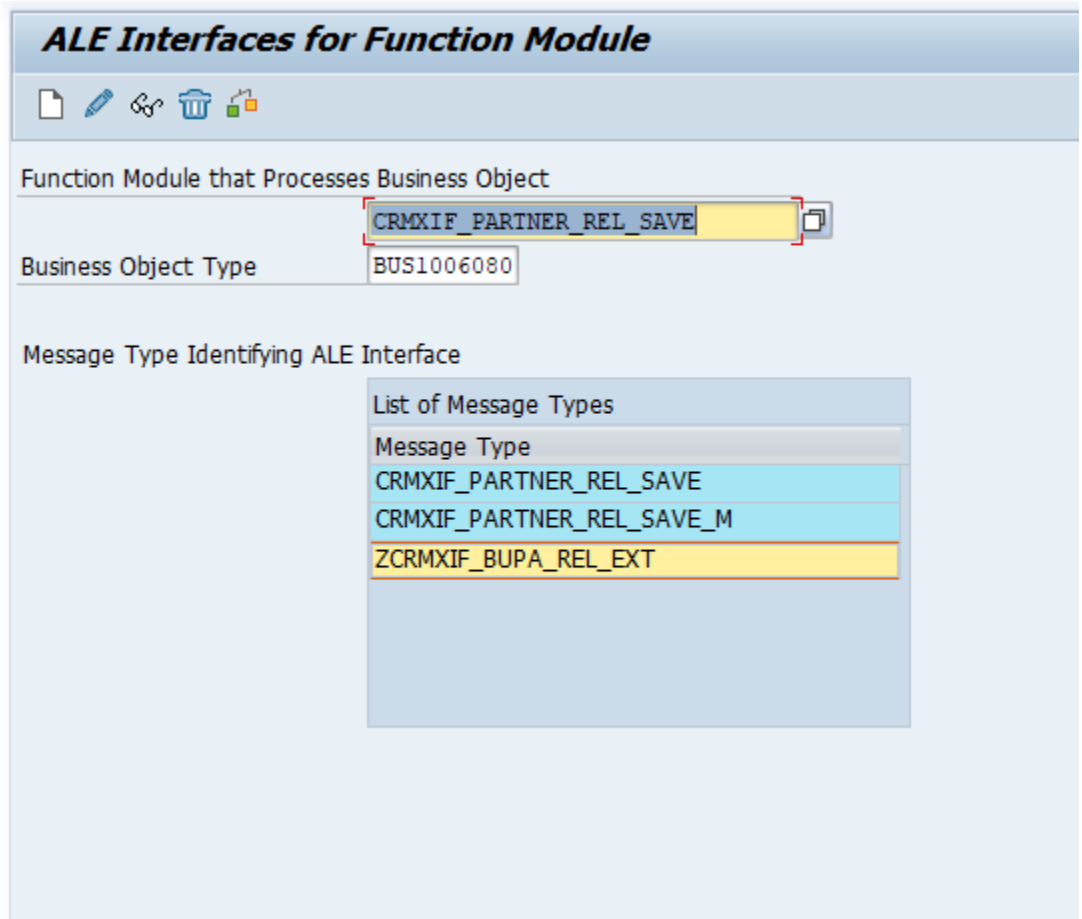


6. Press the *Continue* to proceed.
7. Once the interface is generated, choose the menu option *Interface -> Check* to see if all components have been created and activated successfully.
8. Go to transaction WE30, enter IDoc Type Z or Y<CRMXIF_BUPA_EXT_01> and choose the menu option *Development Object -> Check*. All the non-released segments are highlighted in the check.
9. Set the *Release* for all the non-released segment types through transaction WE31.

- *Business partner relationship*

1. Create a new function group Z or Y<ALE_BUPA_REL_MKT_EXT> and activate it.
2. Start transaction BDFG.
3. Enter function module as CRMXIF_PARTNER_REL_SAVE and press Enter. The corresponding Business Object and relevant Message Type are displayed.

PUBLIC
© 2019 SAP SE or an SAP affiliate company. All rights reserved.



4. From the menu, choose *Interface -> Create.*
5. Enter the parameters as shown below for Outbound Processing. Leave the Inbound Processing parameters unpopulated.

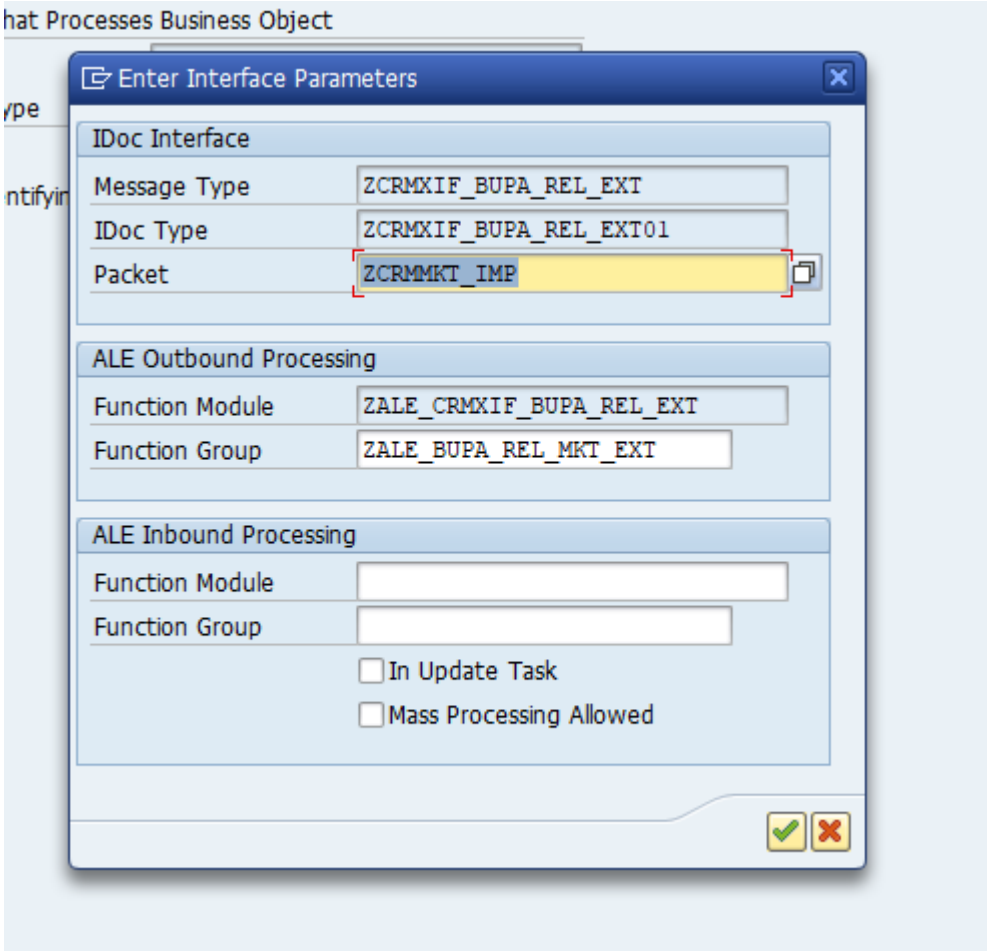
   Message type: Z or Y<CRMXIF_BUPA_REL_EXT>

   IDoc type: Z or Y<CRMXIF_BUPA_REL_EXT01>

   Package: Z or Y<CRMMKT_IMP> (see 2.2.1)

   Function group: Z or Y<ALE_BUPA_REL_MKT_EXT>

PUBLIC
© 2019 SAP SE or an SAP affiliate company. All rights reserved.



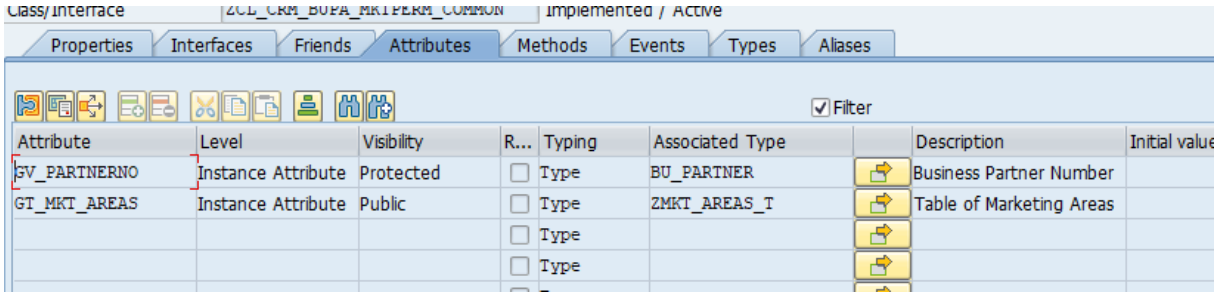
6. Press the *Continue* button to proceed.
7. Once the interface is generated, choose the menu option *Interface -> Check* to see if all components have been created and activated successfully.
8. Go to transaction WE30, enter IDoc Type Z or Y<CRMXIF_BUPA_REL_EXT_01> and choose the menu option *Development Object -> Check*. All the non-released segments are highlighted in the check.
9. Set the *Release* for all the non-released segment types through transaction WE31.

3.8 Common Objects

1. Go to the tab *Attributes*, add two new attributes to the common class Z or Y<CL_CRM_BUPA_MKTPERM_COMMON> (see 2.2.3)

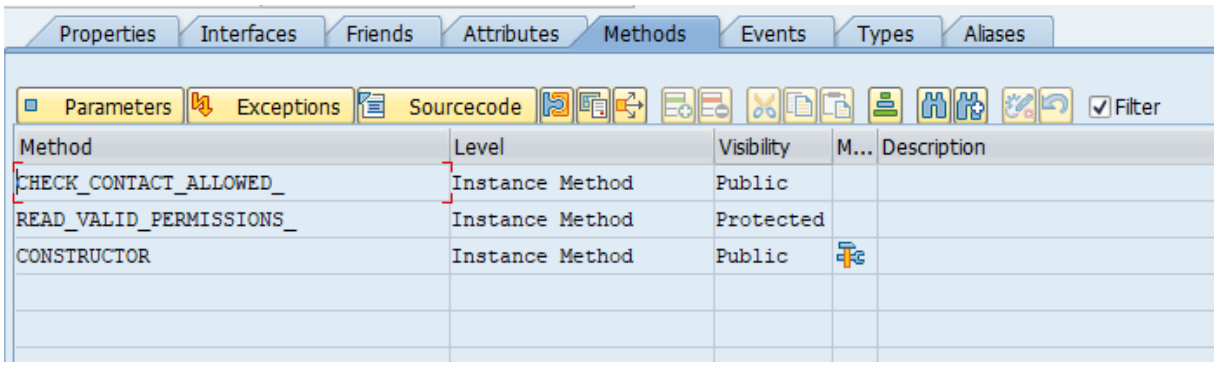
   GV_PARTNERNO: Instance Attribute, Protected, BU_PARTNER

   GT_MKT_AREAS: Instance Attribute, Public, ZMKT_AREAS_T

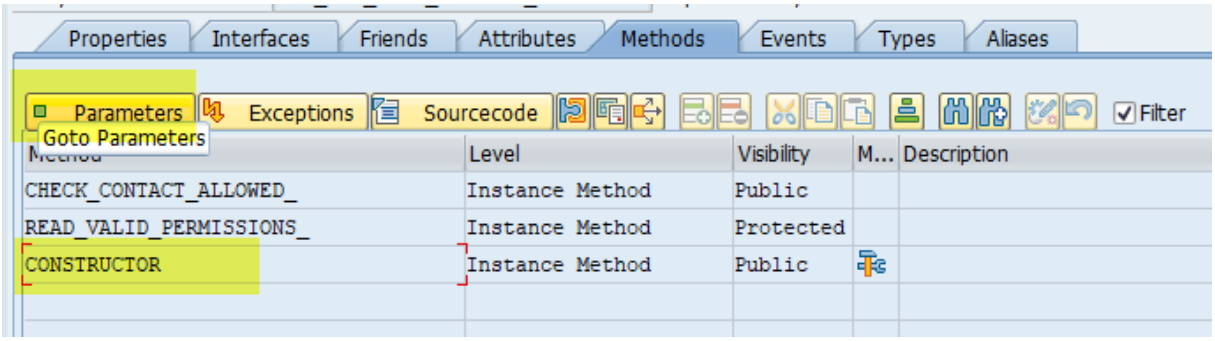


© 2019 SAP SE or an SAP affiliate company. All rights reserved.

2. Go to the tab *Methods* and add a new constructor:
   CONSTRUCTOR: Instance Method, Public

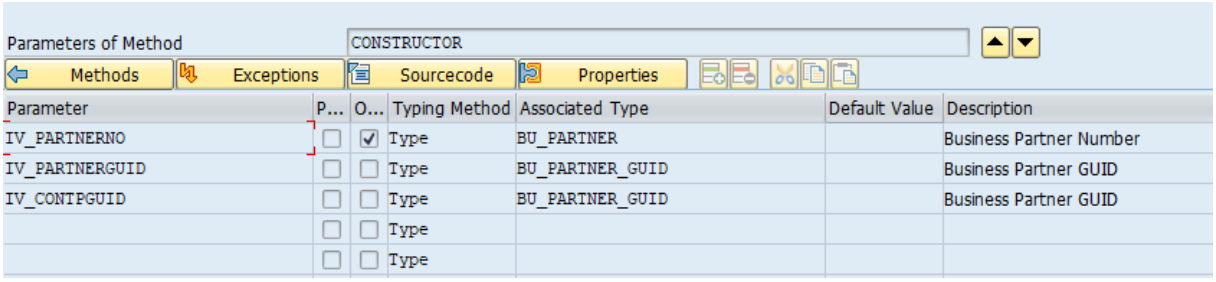


3. Select CONSTRUCTOR and click on *Parameters* button



4. Add three new parameters for the constructor.

   IV_PARTNERNO: Optional, BU_PARTNER
   IV_PARTNERGUID: BU_PARTNER_GUID
   IV_CONTPGUID: BU_PARTNER_GUID



5. Implement the method CONSTRUCTOR.

```
method CONSTRUCTOR.
  super->constructor( IV_PARTNERGUID = IV_PARTNERGUID
                      IV_CONTPGUID   = IV_CONTPGUID ).

  GV_PARTNERNO = IV_PARTNERNO.

*get marketing organization
*marketing organization data
  DATA: lt_relations TYPE TABLE OF BAPIBUS1006_RELATIONS,
        ls_relation type BAPIBUS1006_RELATIONS,
        lt_return TYPE TABLE OF BAPIRET2,
        lt_mkt_org TYPE CRMT_ORGMAN_MKT_ORG_F4_TAB,
        ls_mkt_org LIKE LINE OF lt_mkt_org,
        lt_result_struc TYPE TABLE OF STRUC,
        ls_result_struc LIKE LINE OF lt_result_struc,
        lt_result_object TYPE TABLE OF OBJEC,
```


© 2019 SAP SE or an SAP affiliate company. All rights reserved.

```
      ls_result_object LIKE LINE OF lt_result_object,
      lt_result_tab TYPE TABLE OF SWHACTOR,
      ls_result_tab LIKE LINE OF lt_result_tab,
      lt_mkt_area TYPE zmkt_areas_t,
      ls_mkt_area TYPE ZMKT_AREAS_APPEND,
      ls_mkt_area_map TYPE ZZMKTAREA_MAP,
      ls_bus_ei_mkt_permissions TYPE CRMT_BUS_EI_MKT_PERMISSIONS,
      ls_mkt_permissions TYPE CRMT_BUS_EI_MKT_PERMISSIONS.
*marketing organization data

    CALL FUNCTION 'BUPA_RELATIONSHIPS_GET'
     EXPORTING
      IV_PARTNER              = GV_PARTNERNO
      IV_RELATIONSHIP_CATEGORY     = 'BUR011'
     TABLES
      ET_RELATIONSHIPS          = lt_relations
      ET_RETURN               = lt_return.

    READ TABLE lt_relations INTO ls_relation WITH KEY DEFAULTRELATIONSHIP  = 'X'.

    CLEAR ls_mkt_org.
    CLEAR lt_mkt_org.
    CALL METHOD CL_CRM_MKTGS_MD_SERVICES=>GET_MKT_ORGS
     EXPORTING
      IV_PERS_RESP      = ls_relation-PARTNER2
     IMPORTING
      ET_MKT_ORGS       = lt_mkt_org.

    CALL FUNCTION 'RH_STRUC_GET'
     EXPORTING
      ACT_OTYPE         = 'BP'
      ACT_OBJID         = ls_relation-PARTNER2
      ACT_WEGID         = 'BP-CP-O'
     TABLES
      RESULT_TAB        = lt_result_tab
      RESULT_OBJEC        = LT_RESULT_OBJECT
      RESULT_STRUC        = LT_RESULT_STRUC
     EXCEPTIONS
      NO_PLVAR_FOUND      = 1
      NO_ENTRY_FOUND      = 2
      OTHERS            = 3.

    CLEAR ls_mkt_area.
    CLEAR lt_mkt_area.
    LOOP AT lt_result_object INTO ls_result_object.
     CONCATENATE ls_result_object-OTYPE ls_result_object-OBJID INTO DATA(ls_objid) SEPARATED BY space.
     READ TABLE LT_MKT_ORG INTO ls_mkt_org WITH KEY MKT_ORG = ls_objid.
     IF sy-subrc EQ 0.
       SELECT SINGLE * FROM ZZMKTAREA_MAP INTO ls_mkt_area_map WHERE ZZMKT_ORG EQ ls_mkt_org-
MKT_ORG.
       IF sy-subrc EQ 0.
        ls_mkt_area-ZZMKT_ORG = LS_MKT_AREA_MAP-ZZMKT_ORG.
        ls_mkt_area-ZZYMKT_AREA = LS_MKT_AREA_MAP-ZZYMKT_AREA.
        ls_mkt_area-ZZSHORT_TEXT = ls_mkt_org-SHORT_TEXT.
        ls_mkt_area-ZZBEGDA = ls_mkt_org-BEGDA.
        ls_mkt_area-ZZENDDA = ls_mkt_org-ENDDA.
        APPEND ls_mkt_area TO lt_mkt_area.
       ENDIF.
     ENDIF.
    ENDLOOP.
    GT_MKT_AREAS = lt_mkt_area.
 endmethod.
```


© 2019 SAP SE or an SAP affiliate company. All rights reserved.

3.9 BAdI CRMXIF_PARTNER_MAP (Marketing Permissions for Accounts)

BAdI Implementation: Z or Y<CRMMKT_BUPA_MAP> (see 2.3.3)

1. Implement the method.
   IF_EX_CRMXIF_PARTNER_MAP~CHANGE_MAPPED_DATA_OUT

```
method IF_EX_CRMXIF_PARTNER_MAP~CHANGE_MAPPED_DATA_OUT.
  CONSTANTS lc_initial_guid   TYPE bu_partner_guid VALUE '0'.
  FIELD-SYMBOLS:
   <mkt_permission_t> TYPE CRMT_BUS_EI_MKT_PERMISSIONS_T,
   <mkt_permission> TYPE CRMT_BUS_EI_MKT_PERMISSIONS,
   <mkt_permission_channel> TYPE CRMT_BUS_EI_MKT_PERMISSIONS,
   <xif_partner_complex_t> type CRMXIF_PARTNER_COMPLEX_T,
   <xif_partner_complex> type CRMXIF_PARTNER_COMPLEX.

  DATA: ls_partner_complex type CRMXIF_PARTNER_COMPLEX.

  DATA:  lt_mkp_applicable type ZMKT_APPLICABLE_T,
       ls_mkp_applicable like line of lt_mkp_applicable,
       lt_channel type CRMT_BUS_EI_MKT_PERMISSIONS_T,
       ls_channel type CRMT_BUS_EI_MKT_PERMISSIONS,
       lt_valid_permissions type crmt_but_mktperm_all_t,
       ls_valid_permission type crmt_but_mktperm_all,
       lt_valid_mktperm type crmt_but_mktperm_all_t,
       lt_applicable_account type CRMT_BUT_MKP_APPLICABLE_T,
       ls_applicable_account like line of lt_applicable_account,
       ls_value type ZMKT_VALUES,
       lt_value type ZMKT_VALUES_T.

* address dependent communication channel
  DATA: lt_address type BUS_EI_BUPA_ADDRESS_T,
      ls_address like line of lt_address,
      lt_phone type BUS_EI_BUPA_TELEPHONE_T,
      ls_phone type BUS_EI_BUPA_TELEPHONE,
      lt_fax type BUS_EI_BUPA_FAX_T,
      ls_fax type BUS_EI_BUPA_FAX,
      lt_email type BUS_EI_BUPA_SMTP_T,
      ls_email type BUS_EI_BUPA_SMTP,
      lt_pager type BUS_EI_BUPA_PAG_T,
      ls_pager type BUS_EI_BUPA_PAG.

DATA: lv_timestamp_valid_from  TYPE timestamp,
      lv_initial_time        TYPE syst_uzeit,
      lv_timestamp_current_utc TYPE timestamp,
      lv_initial_timezone      TYPE ttzz-tzone,
      lv_date_valid_from      TYPE sydatum,
      lv_date_UTC           TYPE sydatum.


  DATA: lv_index type integer.
*marketing organization data
  DATA: ls_mkt_area TYPE zmkt_areas_append,
      ls_bus_ei_mkt_permissions TYPE CRMT_BUS_EI_MKT_PERMISSIONS,
      ls_mkt_permissions TYPE CRMT_BUS_EI_MKT_PERMISSIONS.
*marketing organization data

  DATA: lo_zcl_crm_bupa_mktperm type ref to zcl_crm_bupa_mktperm_common.

* CRM -> C4C
 FIELD-SYMBOLS <ls_data_mapped> LIKE LINE OF cs_data_mapped.
 FIELD-SYMBOLS <ls_role>      LIKE LINE OF <ls_data_mapped>-central_data-role-roles.
 FIELD-SYMBOLS <ls_ident_numbers> TYPE bus_ei_bupa_identification.

*Employees should not be replicated via the business partner replication anymore but
*via the dedicated employee replication. Therefore, ignore employees or delete the
*employee role if the employee has also other supported roles like Sold-To or Contact
```


© 2019 SAP SE or an SAP affiliate company. All rights reserved.

```
 LOOP AT cs_data_mapped ASSIGNING <ls_data_mapped>.
  DELETE <ls_data_mapped>-central_data-role-roles WHERE data-rolecategory = 'BUP003'.  "Employee

   "No role is left -> Ignore the business partner
  IF sy-subrc IS INITIAL AND <ls_data_mapped>-central_data-role-roles IS INITIAL.
    DELETE cs_data_mapped.
    CONTINUE.
  ENDIF.

  IF mv_erp_logsys IS NOT INITIAL.
    READ TABLE <ls_data_mapped>-central_data-ident_number-ident_numbers ASSIGNING <ls_ident_numbers>
                                  WITH KEY data_key-identificationcategory = 'CRM002'.  "ERP Customer ID

    IF sy-subrc IS INITIAL.
     <ls_ident_numbers>-data-idinstitute = mv_erp_logsys.
    ENDIF.
  ENDIF.
 ENDLOOP.
* CRM -> C4C

 LOOP AT cs_data_mapped ASSIGNING <xif_partner_complex>.
  clear lt_mkp_applicable.
  clear ls_mkp_applicable.
  clear lt_address.
  clear lt_channel.
  UNASSIGN <mkt_permission_t>.
  UNASSIGN <mkt_permission>.
  UNASSIGN <mkt_permission_channel>.

  IF <xif_partner_complex>-HEADER-OBJECT_TASK EQ 'C'.
* address dependent communication channel
     lt_address = <xif_partner_complex>-CENTRAL_DATA-ADDRESS-ADDRESSES.
     check lt_address is not initial.
     READ TABLE lt_address INTO ls_address WITH KEY DATA-POSTAL-DATA-STANDARDADDRESS = 'X'.
     IF sy-subrc EQ 0.
      LOOP AT ls_address-DATA-COMMUNICATION-PHONE-PHONE INTO ls_phone
       WHERE CURRENTLY_VALID = 'X'.
        APPEND ls_phone TO lt_phone.
      ENDLOOP.

      READ TABLE ls_address-DATA-COMMUNICATION-FAX-FAX INTO ls_fax
       WITH KEY  CONTACT-DATA-STD_NO = 'X'
             CURRENTLY_VALID = 'X'.

      READ TABLE ls_address-DATA-COMMUNICATION-SMTP-SMTP INTO ls_email
       WITH KEY  CONTACT-DATA-STD_NO = 'X'
             CURRENTLY_VALID = 'X'.

      READ TABLE ls_address-DATA-COMMUNICATION-PAGER-PAGER INTO ls_pager
       WITH KEY  CONTACT-DATA-STD_NO = 'X'
             CURRENTLY_VALID = 'X'.

     ENDIF.
     ASSIGN <xif_partner_complex>-CRM_DATA-MKT_PERMISSION-MKT_PERMISSIONS TO <mkt_permission_t>.

     ls_mkp_applicable-MKT_APPLICABLE-PARTNERGUID = <xif_partner_complex>-HEADER-OBJECT_INSTANCE-
PARTNER_GUID.
     ls_mkp_applicable-MKT_APPLICABLE-CONTPGUID = LC_INITIAL_GUID.

* scenario: marketing organization
     CREATE OBJECT LO_ZCL_CRM_BUPA_MKTPERM
      exporting
       IV_PARTNERNO   = <xif_partner_complex>-HEADER-OBJECT_INSTANCE-PARTNER_NO
       IV_PARTNERGUID = ls_mkp_applicable-MKT_APPLICABLE-PARTNERGUID
       IV_CONTPGUID   = ls_mkp_applicable-MKT_APPLICABLE-CONTPGUID.
* scenario: marketing organization

     LOOP AT <mkt_permission_t> ASSIGNING <mkt_permission_channel>.
      READ TABLE <xif_partner_complex>-central_data-role-roles WITH KEY data-rolecategory = 'BUP001'
          TRANSPORTING NO FIELDS.
      IF sy-subrc NE 0.
```


© 2019 SAP SE or an SAP affiliate company. All rights reserved.

```
        IF  <mkt_permission_channel>-DATA-CHANNEL EQ 'TEL' OR
          <mkt_permission_channel>-DATA-CHANNEL EQ 'FAX' OR
          <mkt_permission_channel>-DATA-CHANNEL EQ 'INT' OR
          <mkt_permission_channel>-DATA-CHANNEL EQ 'PAG'.
         READ TABLE lt_channel WITH KEY DATA-CHANNEL = <mkt_permission_channel>-DATA-
CHANNEL TRANSPORTING NO FIELDS.
         IF sy-subrc NE 0.
          <mkt_permission_channel>-DATA-ORIGIN = ''.
          <mkt_permission_channel>-DATA-PERMISSION = ''.
          <mkt_permission_channel>-DATA-VALID_FROM = ''.
          <mkt_permission_channel>-DATA-VALUE = ''.
          APPEND <mkt_permission_channel> TO lt_channel.
         ENDIF.
        ENDIF.
       ENDIF.
       DELETE <mkt_permission_t>.
      ENDLOOP.
*process pro channel
      LOOP AT lt_channel INTO ls_channel.
        clear lt_mkp_applicable.
        clear lt_valid_permissions.
        clear lt_value.
        CASE ls_channel-DATA-CHANNEL.
*       TEL -> PHONE
         WHEN 'TEL'.
          LOOP AT lt_phone INTO ls_phone.
           ls_value-VALUE = ls_phone-CONTACT-DATA-TEL_NO.
           ls_value-STD = ls_phone-CONTACT-DATA-STD_NO.
           ls_value-R_3_USER = ls_phone-CONTACT-DATA-R_3_USER.
           APPEND ls_value TO lt_value.
          ENDLOOP.
*       FAX -> FAX
         WHEN 'FAX'.
          ls_value-VALUE = ls_fax-CONTACT-DATA-FAX_NO.
          ls_value-STD = ls_fax-CONTACT-DATA-STD_NO.
          APPEND ls_value TO lt_value.
*       INT -> EMAIL
         WHEN 'INT'.
          ls_value-VALUE = ls_email-CONTACT-DATA-E_MAIL.
          ls_value-STD = ls_email-CONTACT-DATA-STD_NO.
          APPEND ls_value TO lt_value.
*       PAG -> SMS
         WHEN 'PAG'.
          ls_value-VALUE = ls_pager-CONTACT-DATA-PAGER_NO.
          ls_value-STD = ls_pager-CONTACT-DATA-STD_NO.
          APPEND ls_value TO lt_value.
         WHEN OTHERS.
        ENDCASE.

       IF lt_value IS INITIAL.
        CONTINUE.
       ENDIF.

        CALL METHOD LO_ZCL_CRM_BUPA_MKTPERM->CHECK_CONTACT_ALLOWED_
         EXPORTING
          IV_CHANNEL            = ls_channel-DATA-CHANNEL
          IT_STD_VALUE          = lt_value
          IV_CHECK_ACCOUNT_CONTACT = abap_false
         CHANGING
          CT_MKTPERM_APPLICABLE    = lt_mkp_applicable
         RECEIVING
          RT_VALID_PERMISSIONS     = lt_valid_permissions.

        IF lt_valid_permissions IS INITIAL OR lt_mkp_applicable IS INITIAL.
         CONTINUE.
        ENDIF.
*************************************
        LOOP AT lt_valid_permissions INTO ls_valid_permission.
         IF ls_valid_permission-VALUE IS INITIAL AND ls_value-VALUE IS INITIAL.
          CONTINUE.
         ENDIF.
         APPEND INITIAL LINE TO <mkt_permission_t>  ASSIGNING <mkt_permission>.
```


© 2019 SAP SE or an SAP affiliate company. All rights reserved.

```
            lv_index = sy-TABIX.
    *check against mobile phone
            IF ls_channel-DATA-CHANNEL EQ 'TEL'.
             IF ls_valid_permission-VALUE IS INITIAL.
              READ TABLE lt_value INTO ls_value WITH KEY
               STD = 'X'.
              IF sy-subrc EQ 0.
               IF ls_value-R_3_USER = '2' OR ls_value-R_3_USER = '3'.
                <mkt_permission>-DATA-CHANNEL = 'TE2'.
               ENDIF.
              ENDIF.
             ELSE.
              READ TABLE ls_address-DATA-COMMUNICATION-PHONE-PHONE INTO ls_phone
                WITH KEY CONTACT-DATA-TEL_NO = ls_valid_permission-value.
              IF sy-subrc EQ 0.
               IF ls_phone-CONTACT-DATA-R_3_USER = '2' OR ls_phone-CONTACT-DATA-R_3_USER = '3'.
                <mkt_permission>-DATA-CHANNEL = 'TE2'.
               ENDIF.
              ENDIF.
             ENDIF.
            ENDIF.
    *check against mobile phone
            IF <mkt_permission>-DATA-CHANNEL IS INITIAL.
             <mkt_permission>-DATA-CHANNEL = ls_channel-DATA-CHANNEL.
            ENDIF.
*     simulate CPI conversion of time stamp; add initial time to timestamp valid from date is not in future
    CONVERT DATE ls_valid_permission-valid_from TIME lv_initial_time INTO TIME STAMP lv_timestamp_valid_from TIME
    ZONE lv_initial_timezone.
*     get current timestamp in utc
    GET TIME STAMP FIELD  lv_timestamp_current_utc.

*     get dates from timestamps initial timezone is UTC
                  CONVERT TIME STAMP lv_timestamp_valid_from  TIME ZONE lv_initial_timezone INTO DATE
                  lv_date_valid_from.
                    CONVERT TIME STAMP lv_timestamp_current_utc TIME ZONE lv_initial_timezone INTO DATE lv_date_utc.

*     if currecnt UTC date is smaller / valid from date is in future => current UTC should be passed
                    IF lv_date_valid_from > lv_date_utc.
                     ls_valid_permission-valid_from = lv_date_utc.
                    ENDIF.

            <mkt_permission>-DATA-VALID_FROM = ls_valid_permission-VALID_FROM.
            READ TABLE lt_mkp_applicable INTO ls_mkp_applicable
             WITH KEY MKT_APPLICABLE-PARTNERGUID = ls_mkp_applicable-MKT_APPLICABLE-PARTNERGUID
                  MKT_APPLICABLE-CONTPGUID   = ls_mkp_applicable-MKT_APPLICABLE-CONTPGUID
                  VALUE = ls_valid_permission-VALUE
                  RECORD_GUID = LS_VALID_PERMISSION-RECORD_GUID.
            IF ls_mkp_applicable-MKT_APPLICABLE-CONTACT_ALLOWED EQ 'X'.
             <mkt_permission>-DATA-PERMISSION = '001'.
            ELSE.
             <mkt_permission>-DATA-PERMISSION = '002'.
            ENDIF.
    *       <mkt_permission>-DATA-ORIGIN = ls_valid_permission-ORIGIN.
            <mkt_permission>-DATA_KEY-RECORD_GUID = ls_valid_permission-RECORD_GUID.

            IF ls_valid_permission-VALUE IS INITIAL.
              READ TABLE lt_value INTO ls_value WITH KEY STD = 'X'.
              IF sy-subrc EQ 0.
                <mkt_permission>-DATA-VALUE = ls_value-VALUE.
    *something wrong, remove the permissions
              ELSE.
                DELETE <MKT_PERMISSION_T> INDEX lv_index.
                CONTINUE.
              ENDIF.
            ELSE.
             READ TABLE lt_value WITH KEY VALUE = LS_VALID_PERMISSION-VALUE TRANSPORTING NO FIELDS.
             IF sy-subrc EQ 0.
```


© 2019 SAP SE or an SAP affiliate company. All rights reserved.

```
          <mkt_permission>-DATA-VALUE = ls_valid_permission-VALUE.
*something wrong, remove the permissions
         ELSE.
          DELETE <MKT_PERMISSION_T> INDEX lv_index.
          CONTINUE.
         ENDIF.
        ENDIF.

* scenario: marketing organization
        IF LO_ZCL_CRM_BUPA_MKTPERM->GT_MKT_AREAS IS NOT INITIAL.
         LS_BUS_EI_MKT_PERMISSIONS = <MKT_PERMISSION>.
         DELETE <MKT_PERMISSION_T> INDEX lv_index.
         LOOP AT LO_ZCL_CRM_BUPA_MKTPERM->GT_MKT_AREAS INTO ls_mkt_area.
          ls_mkt_permissions = LS_BUS_EI_MKT_PERMISSIONS.
          ls_mkt_permissions-DATA-ZZBEGDA = ls_mkt_area-ZZBEGDA.
          ls_mkt_permissions-DATA-ZZENDDA = ls_mkt_area-ZZENDDA.
          ls_mkt_permissions-DATA-ZZMKT_ORG = ls_mkt_area-ZZMKT_ORG.
          ls_mkt_permissions-DATA-ZZYMKT_AREA = ls_mkt_area-ZZYMKT_AREA.
          ls_mkt_permissions-DATA-ZZSHORT_TEXT = ls_mkt_area-ZZSHORT_TEXT.
          append ls_mkt_permissions to <mkt_permission_t>.
         ENDLOOP.
        ENDIF.
* scenario: marketing organization

       ENDLOOP.
     ENDLOOP.
*move duplicates

* scenario: marketing organization
     DELETE ADJACENT DUPLICATES from <MKT_PERMISSION_T> COMPARING DATA-CHANNEL DATA-
PERMISSION DATA-VALID_FROM DATA-VALUE DATA-ZZMKT_ORG.
* scenario: marketing organization

*move all redundant entries
     LOOP AT <mkt_permission_t> ASSIGNING <mkt_permission_channel>.
*move all older entries
      LOOP AT <mkt_permission_t> ASSIGNING <mkt_permission>
       WHERE DATA-CHANNEL    EQ <mkt_permission_channel>-DATA-CHANNEL
*        AND  DATA-VALID_FROM LT <mkt_permission_channel>-DATA-VALID_FROM
       AND  DATA-VALUE     EQ <mkt_permission_channel>-DATA-VALUE.
*        entry classified, remove from working table
        IF <mkt_permission>-DATA-VALID_FROM LT <mkt_permission_channel>-DATA-VALID_FROM.
         DELETE <mkt_permission_t>.
        ELSE.
         IF <mkt_permission>-DATA-VALID_FROM EQ <mkt_permission_channel>-DATA-VALID_FROM.
           IF <MKT_PERMISSION>-DATA-PERMISSION EQ cl_crm_bupa_mktperm_tools=>GC_GIVEN AND
             <MKT_PERMISSION_CHANNEL>-DATA-PERMISSION EQ cl_crm_bupa_mktperm_tools=>GC_REJECTED.
            DELETE <mkt_permission_t>.
           ENDIF.
         ENDIF.
        ENDIF.
      ENDLOOP.
*move all older entries
     ENDLOOP.
*move all redundant entries
   ENDIF.
  ENDLOOP.
 endmethod.
```

3.10 BAdI CRMXIF_PARTNER_R_MAP (Marketing Permissions for Contacts)

BAdI Implementation: Z or Y<CRMMKT_BUPA_REL_MAP (see 2.4.3)

Implement the method IF_EX_CRMXIF_PARTNER_REL_MAP~CHANGE_MAPPED_DATA_OUT.

```
method IF_EX_CRMXIF_PARTNER_REL_MAP~CHANGE_MAPPED_DATA_OUT.
 CONSTANTS lc_initial_guid   TYPE bu_partner_guid VALUE '0'.
```

 © 2019 SAP SE or an SAP affiliate company. All rights reserved.

```
  FIELD-SYMBOLS:
   <mkt_permission_t> TYPE CRMT_BUS_EI_MKT_PERMISSIONS_T,
   <mkt_permission> TYPE CRMT_BUS_EI_MKT_PERMISSIONS,
   <mkt_permission_channel> TYPE CRMT_BUS_EI_MKT_PERMISSIONS,
   <xif_partner_rel_complex_t> type CRMXIF_PARTNER_REL_COMPLEX_T,
   <xif_partner_rel_complex> type CRMXIF_PARTNER_REL_COMPLEX.

  DATA: ls_partner_complex type CRMXIF_PARTNER_REL_COMPLEX,
      lt_bupa_cp type crmt_but_mkp_applicable_t,
      ls_bupa_cp like line of lt_bupa_cp.

  DATA:   lt_mkp_applicable type ZMKT_APPLICABLE_T,
       ls_mkp_applicable like line of lt_mkp_applicable,
       lt_channel type CRMT_BUS_EI_MKT_PERMISSIONS_T,
       ls_channel type CRMT_BUS_EI_MKT_PERMISSIONS,
       lt_valid_permissions type crmt_but_mktperm_all_t,
       ls_valid_permission type crmt_but_mktperm_all,
       lt_valid_mktperm type crmt_but_mktperm_all_t,
       lt_applicable_account type CRMT_BUT_MKP_APPLICABLE_T,
       ls_applicable_account like line of lt_applicable_account,
       ls_value type ZMKT_VALUES,
       lt_value type ZMKT_VALUES_T.

* address dependent communication channel
  DATA: lt_address type BURS_EI_REL_ADDRESS_T,
      ls_address like line of lt_address,
      lt_phone type BUS_EI_BUPA_TELEPHONE_T,
      ls_phone type BUS_EI_BUPA_TELEPHONE,
      lt_fax type BUS_EI_BUPA_FAX_T,
      ls_fax type BUS_EI_BUPA_FAX,
      lt_email type BUS_EI_BUPA_SMTP_T,
      ls_email type BUS_EI_BUPA_SMTP,
      lt_pager type BUS_EI_BUPA_PAG_T,
      ls_pager type BUS_EI_BUPA_PAG,
      lb_contact_valid type crmt_boolean.


  DATA: lt_tel type BAPIADTEL_T,
      ls_tel type BAPIADTEL,
      lt_bapifax type BAPIADFAX_T,
      ls_bapifax type BAPIADFAX,
      lt_smtp type BAPIADSMTP_T,
      ls_smtp type BAPIADSMTP,
      lt_pag type table of bapiadpag,
      ls_pag type bapiadpag,
      lt_return type BAPIRET2_T.


DATA: lv_timestamp_valid_from  TYPE timestamp,
      lv_initial_time        TYPE syst_uzeit,
      lv_timestamp_current_utc TYPE timestamp,
      lv_initial_timezone      TYPE ttzz-tzone,
      lv_date_valid_from      TYPE sydatum,
      lv_date_UTC           TYPE sydatum.


  DATA: lv_index type integer.
* scenario: marketing organization
  DATA: ls_mkt_area TYPE ZMKT_AREAS_APPEND,
      ls_bus_ei_mkt_permissions TYPE CRMT_BUS_EI_MKT_PERMISSIONS,
      ls_mkt_permissions TYPE CRMT_BUS_EI_MKT_PERMISSIONS.
* scenario: marketing organization

  DATA: lo_zcl_crm_bupa_mktperm type ref to zcl_crm_bupa_mktperm_common.

*------------------ CRMPCD_BUPA_REL_MAP-----------------------------
 DATA ls_map_contact         TYPE crmm_but_contno.
 DATA ls_identification       TYPE bapibus1006_identification_key.
 DATA lv_org_guid            TYPE bu_partner_guid.
 DATA lv_person_guid          TYPE bu_partner_guid.
*------------------ CRMPCD_BUPA_REL_MAP-----------------------------

 FIELD-SYMBOLS <ls_data_mapped> TYPE crmxif_partner_rel_complex.
```


© 2019 SAP SE or an SAP affiliate company. All rights reserved.

```
 LOOP AT cs_data_mapped ASSIGNING <ls_data_mapped>.
*----------------- CRMPCD_BUPA_REL_MAP-----------------------------
  CLEAR ls_map_contact.

  lv_org_guid   = <ls_data_mapped>-header-object_instance-partner1-partner_guid.
  lv_person_guid = <ls_data_mapped>-header-object_instance-partner2-partner_guid.

   "Is there a ERP contact ID available for the CRM contact relationship?
  CALL FUNCTION 'BUPA_MAP_CONTACT_GETDETAIL'
   EXPORTING
    iv_org_guid      = lv_org_guid
    iv_person_guid   = lv_person_guid
   IMPORTING
    es_map_contact   = ls_map_contact
   EXCEPTIONS
    no_entry_found   = 1
    invalid_parameter = 2
    OTHERS           = 3.

  CHECK sy-subrc IS INITIAL AND ls_map_contact IS NOT INITIAL.

  ls_identification-identificationcategory = 'CRM014'.
  IF mv_erp_logsys IS NOT INITIAL.
   CONCATENATE mv_erp_logsys '#' ls_map_contact-contact_no INTO ls_identification-identificationnumber.
  ELSE.
   ls_identification-identificationnumber   = ls_map_contact-contact_no.
  ENDIF.

  INSERT ls_identification INTO TABLE <ls_data_mapped>-header-object_instance-partner2-ident_numbers.
 ENDLOOP.
*----------------- CRMPCD_BUPA_REL_MAP-----------------------------


*----------------- marketing permissions --------------------------
 LOOP AT cs_data_mapped ASSIGNING <ls_data_mapped>.
* address dependent communication channel
   clear lt_mkp_applicable.
   clear ls_mkp_applicable.
   clear lt_address.
   clear lt_channel.
   UNASSIGN <mkt_permission_t>.
   UNASSIGN <mkt_permission>.
   UNASSIGN <mkt_permission_channel>.
   IF <ls_data_mapped>-HEADER-OBJECT_TASK EQ 'C'.
    lt_address = <ls_data_mapped>-CENTRAL_DATA-ADDRESS-ADDRESSES.
    check lt_address is not initial.
    READ TABLE lt_address INTO ls_address WITH KEY DATA-POSTAL-DATA-STANDARDADDRESS = 'X'.
    IF sy-subrc EQ 0.
     LOOP AT ls_address-DATA-COMMUNICATION-PHONE-PHONE INTO ls_phone
      WHERE CURRENTLY_VALID = 'X'.
       APPEND ls_phone TO lt_phone.
     ENDLOOP.

     READ TABLE ls_address-DATA-COMMUNICATION-FAX-FAX INTO ls_fax
      WITH KEY  CONTACT-DATA-STD_NO = 'X'
            CURRENTLY_VALID = 'X'.

     READ TABLE ls_address-DATA-COMMUNICATION-SMTP-SMTP INTO ls_email
      WITH KEY  CONTACT-DATA-STD_NO = 'X'
            CURRENTLY_VALID = 'X'.

     READ TABLE ls_address-DATA-COMMUNICATION-PAGER-PAGER INTO ls_pager
      WITH KEY  CONTACT-DATA-STD_NO = 'X'
            CURRENTLY_VALID = 'X'.

    ENDIF.
    ASSIGN <ls_data_mapped>-CRM_DATA-MKT_PERMISSION-MKT_PERMISSIONS TO <mkt_permission_t>.
    ls_mkp_applicable-MKT_APPLICABLE-PARTNERGUID = <ls_data_mapped>-HEADER-OBJECT_INSTANCE-
PARTNER1-PARTNER_GUID.
    ls_mkp_applicable-MKT_APPLICABLE-CONTPGUID = <ls_data_mapped>-HEADER-OBJECT_INSTANCE-
PARTNER2-PARTNER_GUID.
```


© 2019 SAP SE or an SAP affiliate company. All rights reserved.

```
* scenario: marketing organization
    CREATE OBJECT LO_ZCL_CRM_BUPA_MKTPERM
     exporting
      IV_PARTNERNO   = <ls_data_mapped>-HEADER-OBJECT_INSTANCE-PARTNER2-PARTNER_NO
      IV_PARTNERGUID = ls_mkp_applicable-MKT_APPLICABLE-PARTNERGUID
      IV_CONTPGUID   = ls_mkp_applicable-MKT_APPLICABLE-CONTPGUID.
* scenario: marketing organization

    LOOP AT <mkt_permission_t> ASSIGNING <mkt_permission_channel>.
      IF  <mkt_permission_channel>-DATA-CHANNEL EQ 'TEL' OR
        <mkt_permission_channel>-DATA-CHANNEL EQ 'FAX' OR
        <mkt_permission_channel>-DATA-CHANNEL EQ 'INT' OR
        <mkt_permission_channel>-DATA-CHANNEL EQ 'PAG' .
        READ TABLE lt_channel WITH KEY DATA-CHANNEL = <mkt_permission_channel>-DATA-
CHANNEL TRANSPORTING NO FIELDS.
        IF sy-subrc NE 0.
         <mkt_permission_channel>-DATA-ORIGIN = ''.
         <mkt_permission_channel>-DATA-PERMISSION = ''.
         <mkt_permission_channel>-DATA-VALID_FROM = ''.
         <mkt_permission_channel>-DATA-VALUE = ''.
         APPEND <mkt_permission_channel> TO lt_channel.
        ENDIF.
      ENDIF.
     DELETE <mkt_permission_t>.
    ENDLOOP.
*     DELETE ADJACENT DUPLICATES FROM lt_channel COMPARING CHANNEL.
    LOOP AT lt_channel INTO ls_channel.
     clear lt_mkp_applicable.
     clear lt_valid_permissions.
     clear lt_value.
     CASE ls_channel-DATA-CHANNEL.
*   TEL -> PHONE
      WHEN 'TEL'.
       LOOP AT lt_phone INTO ls_phone.
        ls_value-VALUE = ls_phone-CONTACT-DATA-TEL_NO.
        ls_value-STD = ls_phone-CONTACT-DATA-STD_NO.
        ls_value-R_3_USER = ls_phone-CONTACT-DATA-R_3_USER.
        APPEND ls_value TO lt_value.
       ENDLOOP.
*   FAX -> FAX
      WHEN 'FAX'.
       ls_value-VALUE = ls_fax-CONTACT-DATA-FAX_NO.
       ls_value-STD = ls_fax-CONTACT-DATA-STD_NO.
       APPEND ls_value TO lt_value.
*   INT -> EMAIL
      WHEN 'INT'.
       ls_value-VALUE = ls_email-CONTACT-DATA-E_MAIL.
       ls_value-STD = ls_email-CONTACT-DATA-STD_NO.
       APPEND ls_value TO lt_value.
*   PAG -> SMS
      WHEN 'PAG'.
       ls_value-VALUE = ls_pager-CONTACT-DATA-PAGER_NO.
       ls_value-STD = ls_pager-CONTACT-DATA-STD_NO.
       APPEND ls_value TO lt_value.
      WHEN OTHERS.
     ENDCASE.

     IF lt_value IS INITIAL.
      CONTINUE.
     ENDIF.

     CALL METHOD LO_ZCL_CRM_BUPA_MKTPERM->CHECK_CONTACT_ALLOWED_
      EXPORTING
       IV_CHANNEL           = ls_channel-DATA-CHANNEL
       IT_STD_VALUE          = lt_value
       IV_CHECK_ACCOUNT_CONTACT = abap_true
      CHANGING
       CT_MKTPERM_APPLICABLE    = lt_mkp_applicable
      RECEIVING
       RT_VALID_PERMISSIONS     = lt_valid_permissions.
```


© 2019 SAP SE or an SAP affiliate company. All rights reserved.

```
          IF lt_valid_permissions IS INITIAL OR lt_mkp_applicable IS INITIAL.
           CONTINUE.
          ENDIF.
          LOOP AT lt_valid_permissions INTO ls_valid_permission.
           IF ls_valid_permission-VALUE IS INITIAL AND lt_value IS INITIAL.
            CONTINUE.
           ENDIF.
           APPEND INITIAL LINE TO <mkt_permission_t>  ASSIGNING <mkt_permission>.
           lv_index = sy-TABIX.
     *check against mobile phone
           IF ls_channel-DATA-CHANNEL EQ 'TEL'.
            IF ls_valid_permission-VALUE IS INITIAL.
             READ TABLE lt_value INTO ls_value WITH KEY
              STD = 'X'.
             IF sy-subrc EQ 0.
              IF ls_value-R_3_USER = '2' OR ls_value-R_3_USER = '3'.
               <mkt_permission>-DATA-CHANNEL = 'TE2'.
              ENDIF.
             ENDIF.
            ELSE.
             READ TABLE ls_address-DATA-COMMUNICATION-PHONE-PHONE INTO ls_phone
               WITH KEY CONTACT-DATA-TEL_NO = ls_valid_permission-value.
             IF sy-subrc EQ 0.
              IF ls_phone-CONTACT-DATA-R_3_USER = '2' OR ls_phone-CONTACT-DATA-R_3_USER = '3'.
               <mkt_permission>-DATA-CHANNEL = 'TE2'.
              ENDIF.
             ENDIF.
            ENDIF.
           ENDIF.
     *check against mobile phone
           IF <mkt_permission>-DATA-CHANNEL IS INITIAL.
            <mkt_permission>-DATA-CHANNEL = ls_channel-DATA-CHANNEL.
           ENDIF.
*    simulate CPI conversion of time stamp; add initial time to timestamp valid from date is not in future
                CONVERT DATE ls_valid_permission-valid_from TIME lv_initial_time INTO TIME STAMP
     lv_timestamp_valid_from TIME ZONE lv_initial_timezone.
*    get current timestamp in utc
                GET TIME STAMP FIELD  lv_timestamp_current_utc.
*    get dates from timestamps initial timezone is UTC
                CONVERT TIME STAMP lv_timestamp_valid_from  TIME ZONE lv_initial_timezone INTO DATE
     lv_date_valid_from.
                CONVERT TIME STAMP lv_timestamp_current_utc TIME ZONE lv_initial_timezone INTO DATE lv_date_utc.
*    if currecnt UTC date is smaller / valid from date is in future => current UTC should be passed
                IF lv_date_valid_from > lv_date_utc.
                 ls_valid_permission-valid_from = lv_date_utc.
                ENDIF.

           <mkt_permission>-DATA-VALID_FROM = ls_valid_permission-VALID_FROM.
           READ TABLE lt_mkp_applicable INTO ls_mkp_applicable
            WITH KEY MKT_APPLICABLE-PARTNERGUID = ls_mkp_applicable-MKT_APPLICABLE-PARTNERGUID
                 MKT_APPLICABLE-CONTPGUID   = ls_mkp_applicable-MKT_APPLICABLE-CONTPGUID
                 VALUE = ls_valid_permission-VALUE
                 RECORD_GUID = LS_VALID_PERMISSION-RECORD_GUID.
           IF ls_mkp_applicable-MKT_APPLICABLE-CONTACT_ALLOWED EQ 'X'.
            <mkt_permission>-DATA-PERMISSION = '001'.
           ELSE.
            <mkt_permission>-DATA-PERMISSION = '002'.
           ENDIF.

           <mkt_permission>-DATA-ORIGIN = ls_valid_permission-ORIGIN.
           <mkt_permission>-DATA_KEY-RECORD_GUID = ls_valid_permission-RECORD_GUID.

           IF ls_valid_permission-VALUE IS INITIAL.
             READ TABLE lt_value INTO ls_value WITH KEY STD = 'X'.
             IF sy-subrc EQ 0.
               <mkt_permission>-DATA-VALUE = ls_value-VALUE.
     *something wrong, remove the permissions
             ELSE.
```

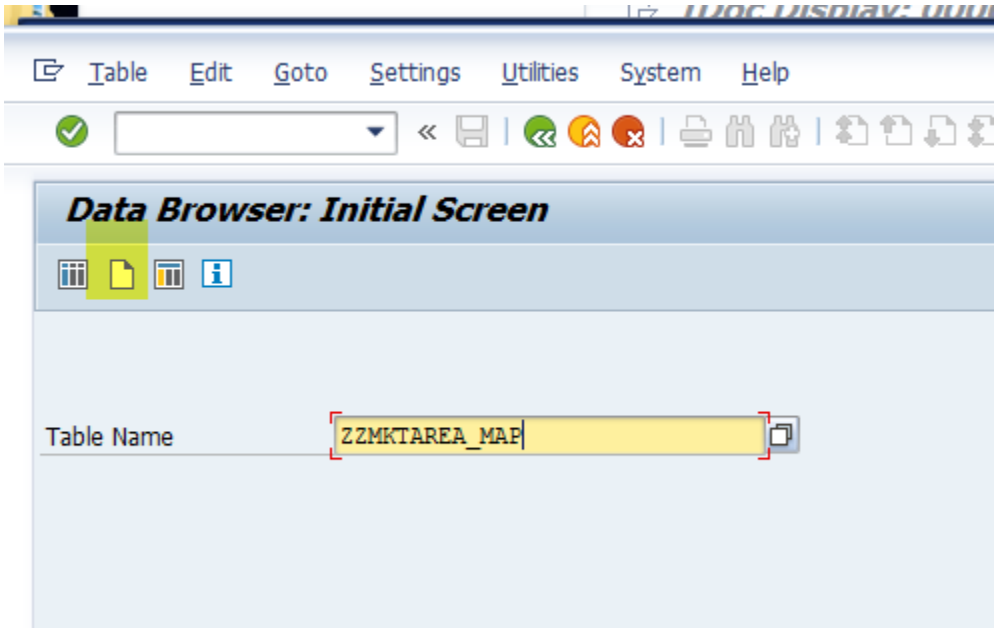

© 2019 SAP SE or an SAP affiliate company. All rights reserved.

```
            DELETE <MKT_PERMISSION_T> INDEX lv_index.
            CONTINUE.
          ENDIF.
        ELSE.
          READ TABLE lt_value WITH KEY VALUE = LS_VALID_PERMISSION-VALUE TRANSPORTING NO FIELDS.
          IF sy-subrc EQ 0.
            <mkt_permission>-DATA-VALUE = ls_valid_permission-VALUE.
*something wrong, remove the permissions
          ELSE.
            DELETE <MKT_PERMISSION_T> INDEX lv_index.
            CONTINUE.
          ENDIF.
        ENDIF.

* scenario: marketing organization
        IF LO_ZCL_CRM_BUPA_MKTPERM->GT_MKT_AREAS IS NOT INITIAL.
          LS_BUS_EI_MKT_PERMISSIONS = <MKT_PERMISSION>.
          DELETE <MKT_PERMISSION_T> INDEX lv_index.
          LOOP AT LO_ZCL_CRM_BUPA_MKTPERM->GT_MKT_AREAS INTO ls_mkt_area.
            ls_mkt_permissions = LS_BUS_EI_MKT_PERMISSIONS.
            ls_mkt_permissions-DATA-ZZBEGDA = ls_mkt_area-ZZBEGDA.
            ls_mkt_permissions-DATA-ZZENDDA = ls_mkt_area-ZZENDDA.
            ls_mkt_permissions-DATA-ZZMKT_ORG = ls_mkt_area-ZZMKT_ORG.
            ls_mkt_permissions-DATA-ZZYMKT_AREA = ls_mkt_area-ZZYMKT_AREA.
            ls_mkt_permissions-DATA-ZZSHORT_TEXT = ls_mkt_area-ZZSHORT_TEXT.
            append ls_mkt_permissions to <mkt_permission_t>.
          ENDLOOP.
        ENDIF.
* scenario: marketing organization
      ENDLOOP.
    ENDLOOP.

*move duplicates
* scenario: marketing organisation
    DELETE ADJACENT DUPLICATES from <MKT_PERMISSION_T> COMPARING DATA-CHANNEL DATA-
PERMISSION DATA-VALID_FROM DATA-VALUE DATA-ZZMKT_ORG.
* scenario: marketing organization

*move all redundant entries
    LOOP AT <mkt_permission_t> ASSIGNING <mkt_permission_channel>.
      LOOP AT <mkt_permission_t> ASSIGNING <mkt_permission>
       WHERE DATA-CHANNEL    EQ <mkt_permission_channel>-DATA-CHANNEL
        AND  DATA-VALUE     EQ <mkt_permission_channel>-DATA-VALUE.
*move all older entries
        IF <mkt_permission>-DATA-VALID_FROM LT <mkt_permission_channel>-DATA-VALID_FROM.
          DELETE <mkt_permission_t>.
        ELSE.
          IF <mkt_permission>-DATA-VALID_FROM EQ <mkt_permission_channel>-DATA-VALID_FROM.
            IF <MKT_PERMISSION>-DATA-PERMISSION EQ cl_crm_bupa_mktperm_tools=>GC_GIVEN AND
              <MKT_PERMISSION_CHANNEL>-DATA-PERMISSION EQ cl_crm_bupa_mktperm_tools=>GC_REJECTED.
              DELETE <mkt_permission_t>.
            ENDIF.
          ENDIF.
        ENDIF.
      ENDLOOP.
    ENDLOOP.
*move all redundant entries
* ----------------- marketing permissions --------------------------
  ENDIF.
 ENDLOOP.
endmethod.
```

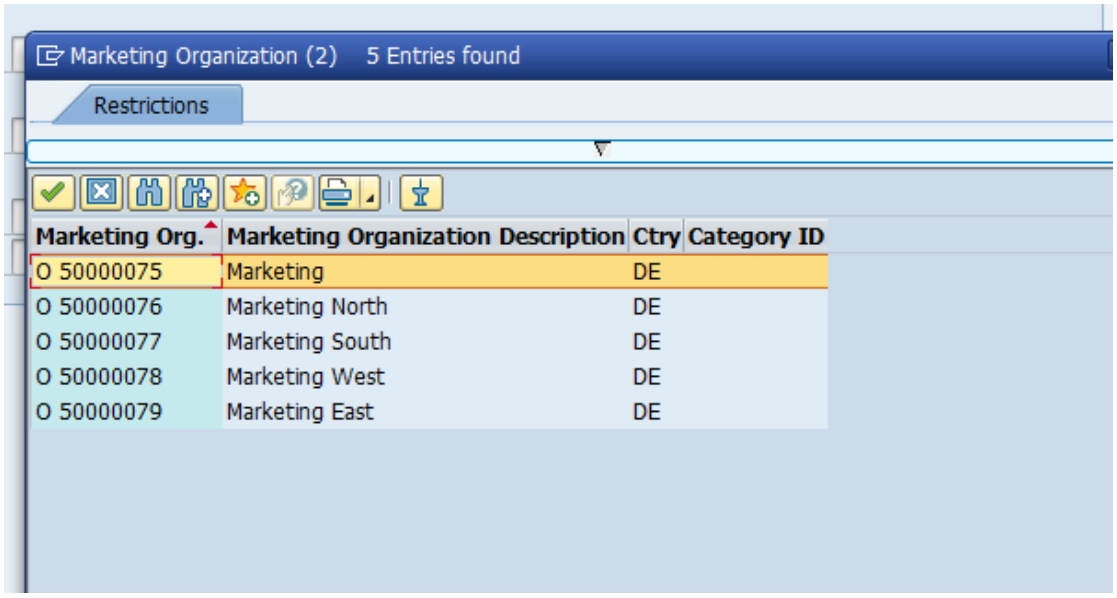
3.11 Maintain Mapping Table

1. Start the transaction SE16.
2. Enter the table name Z or Y<MKTAREA_MAP> (see 3.3) and click *Create Entries*.

© 2019 SAP SE or an SAP affiliate company. All rights reserved.

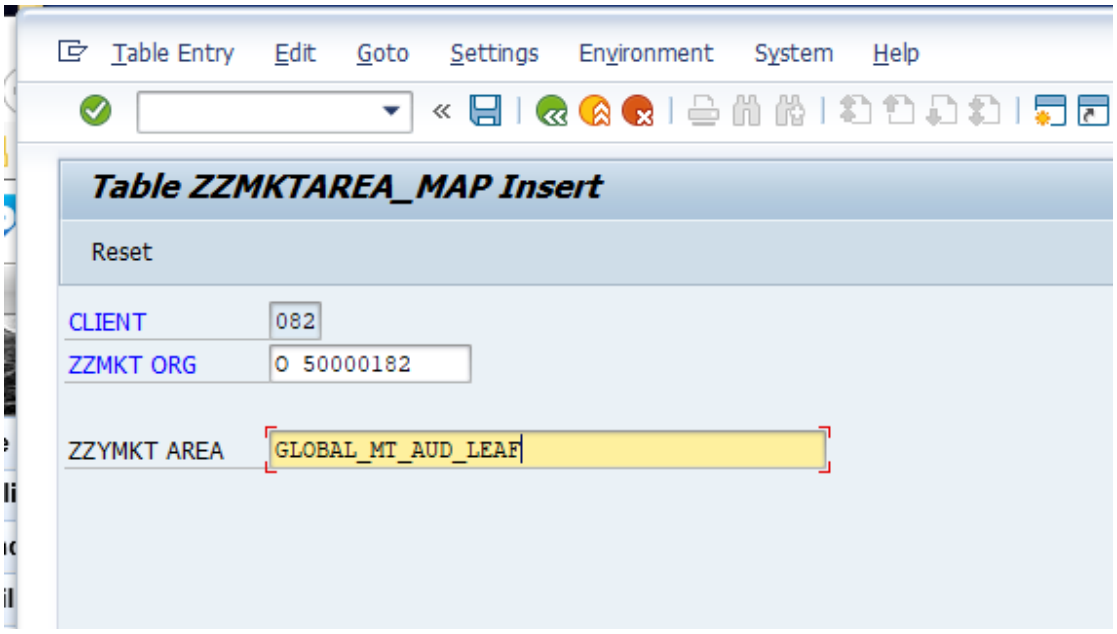


3. Select a marketing organization by clicking F4 value help.

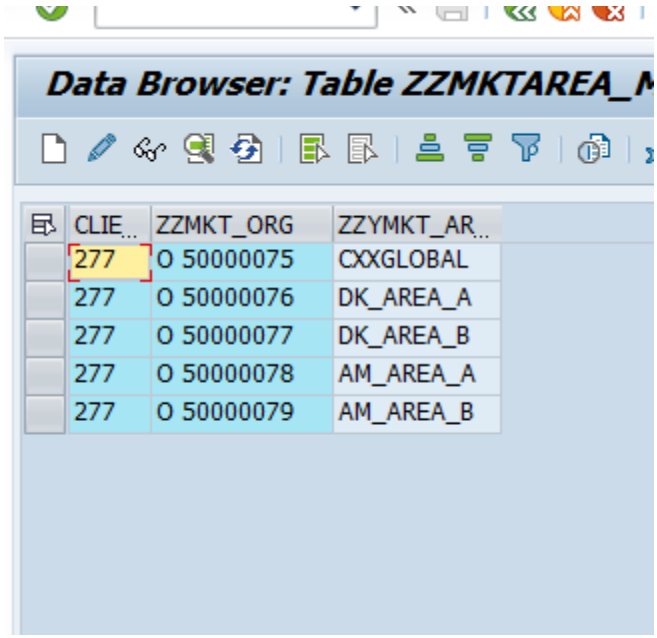


4. Enter a mapping value for marketing area in SAP Hybris Marketing Cloud.

PUBLIC
© 2019 SAP SE or an SAP affiliate company. All rights reserved.



5. Save your entries.



3.12 Create Logical System, RFC Destination, Port, Partner Profile, Site

Logical system:
Create a new logical system according to the setup guide.
RFC destination:
Create two new RFC destinations according to the setup guide.
Port:
Create two new ports according to the setup guide.
Partner profile:
Create a new partner profile according to the setup guide.
Business partner:

© 2019 SAP SE or an SAP affiliate company. All rights reserved.

Message type: Z or Y<CRMXIF_BUPA_EXT>
Basic type: Z or Y<CRMXIF_BUPA_EXT01>
Business partner relationship:
Message type: Z or Y<CRMXIF_BUPA_REL_EXT>
Basic type: Z or Y<CRMXIF_BUPA_REL_EXT01>

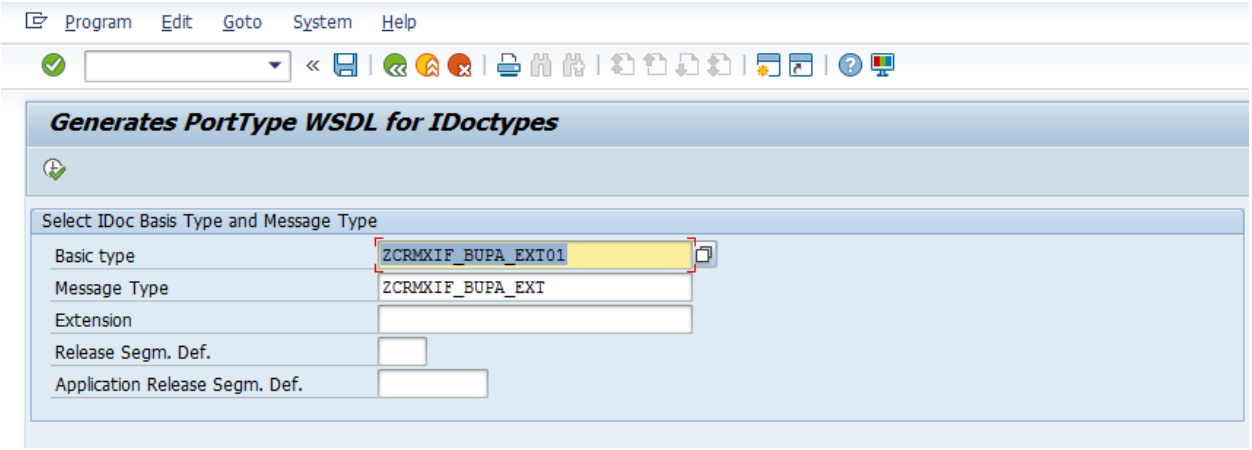
Site:
Create a new site according to the setup guide

3.13 Extending the Permission iFlow in SAP Cloud Platform Integration (CPI)

1. Download WSDL file for new IDoc Type

For the CPI field mapping the metadata of the new IDoc type must be downloaded. You can do this with the report SRT_IDOC_WSDL_NS. Enter the new created IDoc type as message type and its basic type. Press *F8* to download the WSDL file.

You can start a report with the transaction SE38 in the SAP CRM backend system.



2. Adapt WSDL File

To upload the WSDL file to you CPI mapping several changes of the content of the WSDL file are necessary. The changes are necessary so that the existing field mappings do not get lost.

- Open the WSDL file with a text editor and remove the target namespace from the schema element as shown below.

```
<wsdl:types>
 <xsd:schema elementFormDefault="qualified" targetNamespace="urn:sap-com:document:sap:idoc:soap:messages">
  <xsd:complexType name="E101CRMXIF_CUPRT_F.000">
   <xsd:sequence>
```

```
 <xsd:schema elementFormDefault="qualified">
  <xsd:complexType name="E101CRMXIF_CUPRT_F.000">
```

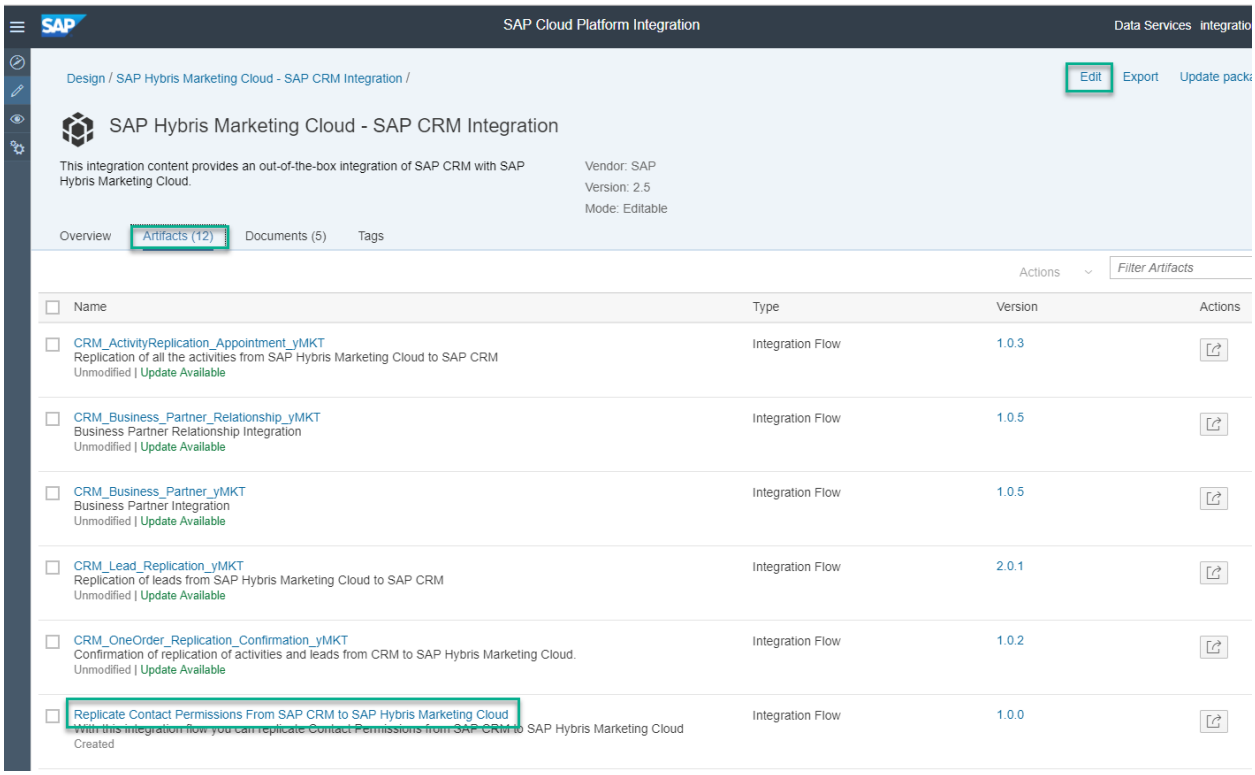

© 2019 SAP SE or an SAP affiliate company. All rights reserved.

- Replace all the occurrence of tns: with empty value.
- Add maxOccurs attribute to the IDoc element to support bulking as in the original mapping.

```
<xsd:element name="ZCRMXIF_ORDER_SAVE_M01">
 <xsd:complexType>
  <xsd:sequence>
   <xsd:element name="IDOC" type=" ZCRMXIF_ORDER_SAVE.ZCRMXIF_ORDER_SAVE_M01" maxOccurs="9999"/>
  </xsd:sequence>
 </xsd:complexType>
</xsd:element>
```

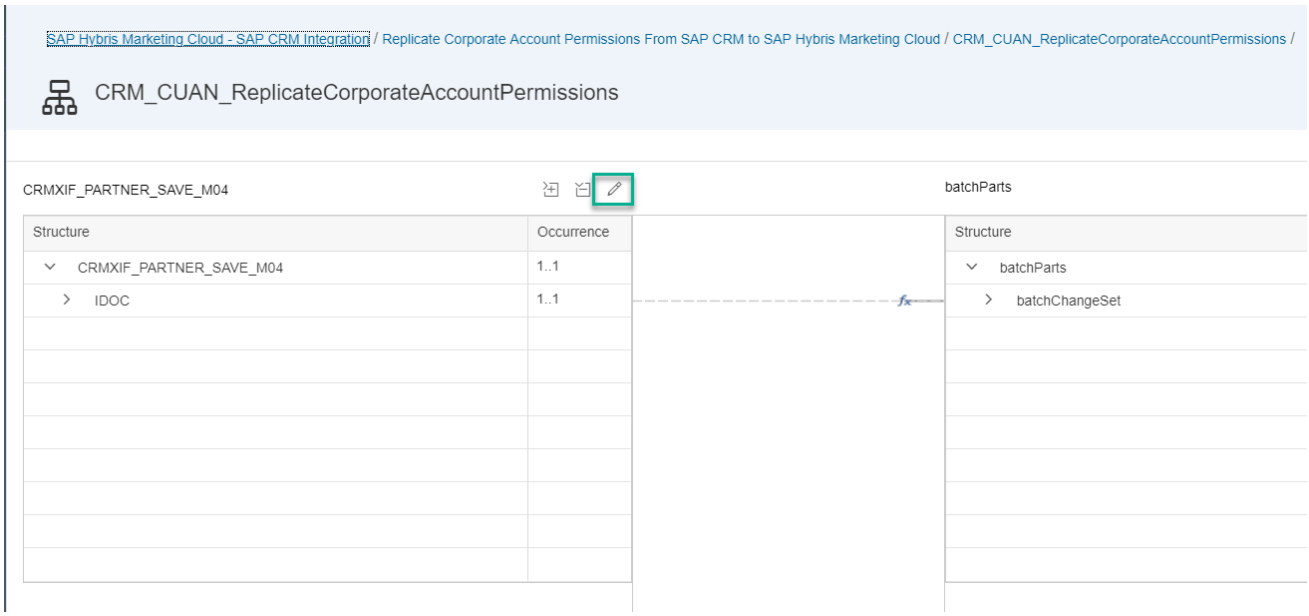
3. Adapt iFlow SAP Cloud Platform Integration (CPI)

Launch SAP Cloud Platform Integration Web UI and navigate to your package that contains the iFlow that connects the SAP CRM system and your SAP Cloud hybris Marketing system and navigate to artifacts. You can see the iFlows *Replicate Contact/ Account Permissions From SAP CRM to SAP Hybris Marketing Cloud*.



In the next step, the metadata of the mapping must be changed. Choose *Edit* to change to change mode and navigate to the mapping /CRMCUAN_ReplicateCOrporateAccountPermissions,mmap of the iFlow.

© 2019 SAP SE or an SAP affiliate company. All rights reserved.

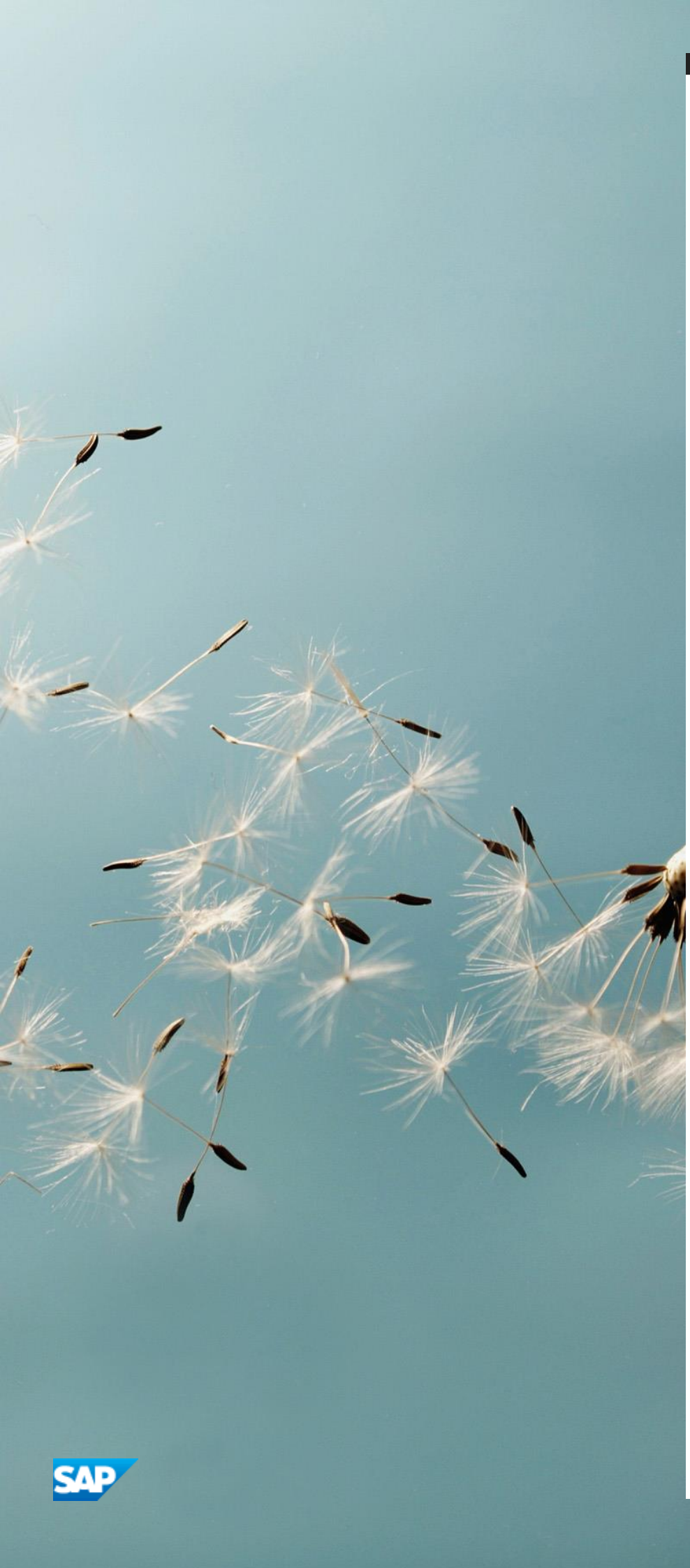


In the mapping, click on the pencil icon on the source message. On the next screen press again on the pencil icon and upload the iDoc Type WSDL file that you downloaded in step 1. Confirm this step with OK in the right upper corner.

After successfully uploading the WSDL file you see in the source the new segment. Press *OK* to accept the change in the structure save and deploy the integration flow.

Now you can map the new field from the IDoc Type to the field *MarketingArea* on the target message side.

© 2019 SAP SE or an SAP affiliate company. All rights reserved.



www.sap.com/contactsap

© 2019 SAP SE or an SAP affiliate company. All rights reserved.

No part of this publication may be reproduced or transmitted in any form or for any purpose without the express permission of SAP SE or an SAP affiliate company.

The information contained herein may be changed without prior notice. Some software products marketed by SAP SE and its distributors contain proprietary software components of other software vendors. National product specifications may vary.

These materials are provided by SAP SE or an SAP affiliate company for informational purposes only, without representation or warranty of any kind, and SAP or its affiliated companies shall not be liable for errors or omissions with respect to the materials. The only warranties for SAP or SAP affiliate company products and services are those that are set forth in the express warranty statements accompanying such products and services, if any. Nothing herein should be construed as constituting an additional warranty.

SAP and other SAP products and services mentioned herein as well as their respective logos are trademarks or registered trademarks of SAP SE (or an SAP affiliate company) in Germany and other countries. All other product and service names mentioned are the trademarks of their respective companies.

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