

**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

---

# 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

---

# 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 (BAI) 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 BAI implementation for the BAIs CRMXIF\_PARTNER\_MAP and CRMXIF\_PARTNER\_R\_MAP.

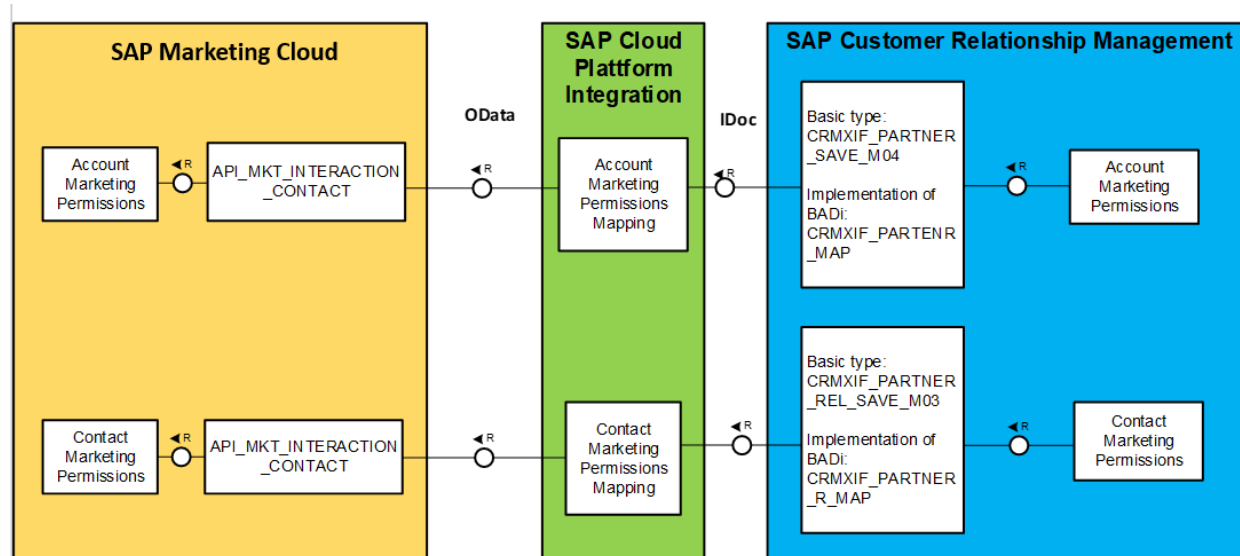
As a sample use case, this guide shows how to implement the two integration scenarios, standard integration scenario (without marketing area determination in SAP CRM) and extended integration scenario (with marketing area determination in SAP CRM).

All required functionality is available with release 1805.

## 2 Standard Integration Scenario

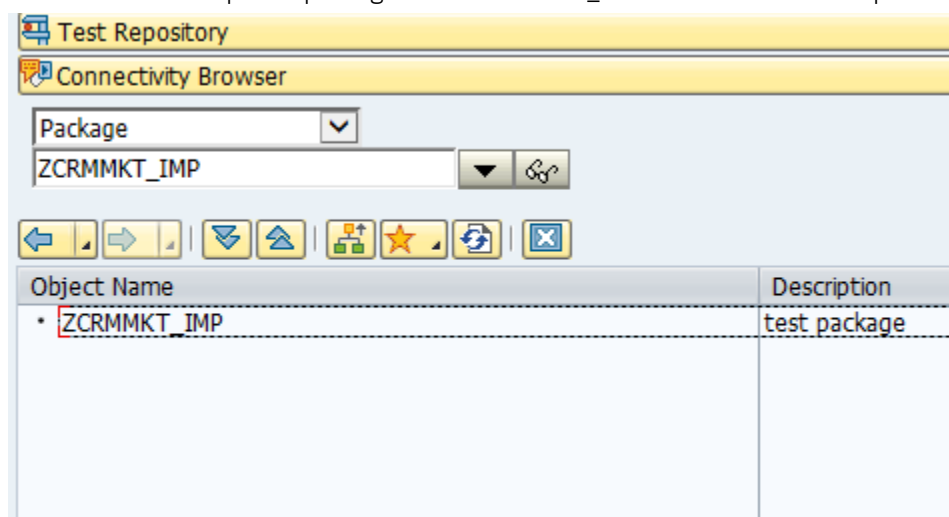
### 2.1 Overview

- 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.
- BAI 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.



## 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

Structure...

Structure: ZMKT\_APPLICABLE Active

Short Description: structure mkt applicable

Attributes Components Entry help/check Currency/quantity fields

Predefined Type 1 / 3

Component	Typing Method	Component Type	Data Type	Length	Decim...	Short Description
MKT_APPLICABLE	Types	CRMT_BUT_MKP_AP..	CHAR	0		0 Marketing Permission: Check Contact Permission
VALUE	Types	CRMT_BU_MKTPERM..	CHAR	241		0 Communication Details
RECORD_GUID	Types	CRMT_BU_MKTPERM..	RAW	16		0 Marketing Permission Record GUID

Structure: Z or Y<MKT\_VALUES>

Component	Component Type
VALUE	CRMT_BU_MKTPERM_COM
STD	CRMT_BOOLEAN
R_3_USER	AD_FLGMOB

Structure...

Structure: ZMKT\_VALUES Active

Short Description: structure mkt values

Attributes Components Entry help/check Currency/quantity fields

Predefined Type 1 / 3

Component	Typing Method	Component Type	Data Type	Length	Decim...	Short Description
VALUE	Types	CRMT_BU_MKTPERM..	CHAR	241		0 Communication Details
STD	Types	CRMT_BOOLEAN	CHAR	1		0 Logical Variable
R_3_USER	Types	AD_FLGMOB	CHAR	1		0 Indicator: Telephone is a Mobile Telephone

Table Type: Z or Y <MKT\_APPLICABLE\_T>

Line type: Z or Y<MKT\_APPLICABLE>

Table Type: ZMKT\_APPLICABLE\_T Active

Short text: table type mkt applicable

Attributes Line Type Initialization and Access Primary Key Secondary Key

Line Type: ZMKT\_APPLICABLE

Predefined Type

Data Type:

No. of Characters: 0 Decimal Places: 0

Table type: Z or Y<MKT\_VALUES\_T>

Line type: Z or Y<MKT\_VALUES>

Table Type	ZMKT_VALUES_T	Active
Short text	table type mkt values	
<div> <div>Attributes</div> <div>Line Type</div> <div>Initialization and Access</div> <div>Primary Key</div> <div>Secondary Key</div> </div>		
<input checked="" type="radio"/> Line Type <div>ZMKT_VALUES</div>		
<input type="radio"/> Predefined Type <div> <div>Data Type</div> <div>No. of Characters</div> <div>0</div> <div>Decimal Places</div> <div>0</div> </div>		

3. Create class object.  
Start transaction SE24.  
Enter a class name Z or Y<CL\_CRM\_BUPA\_MKTPERM\_COMMON> and click on create.

Object type

Edit

Goto

Utilities

Environment

System

Help

☒

<<

Save

Undo

Redo

Print

Home

Back

Forward

Find

Help

**Class Builder: Initial Screen**

Class Browser

Object type

ZCL\_CRM\_BUPA\_MKTPERM\_COMMON

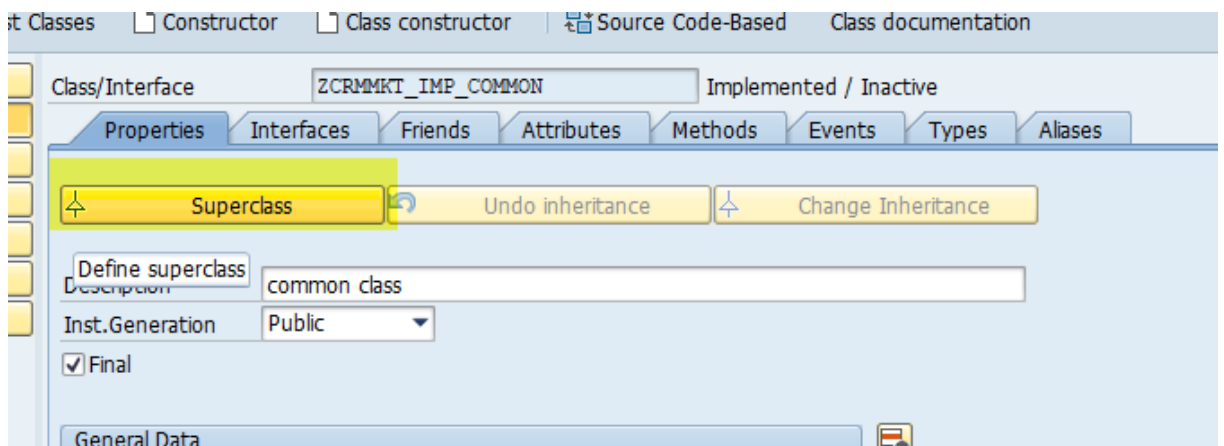
Search

Display

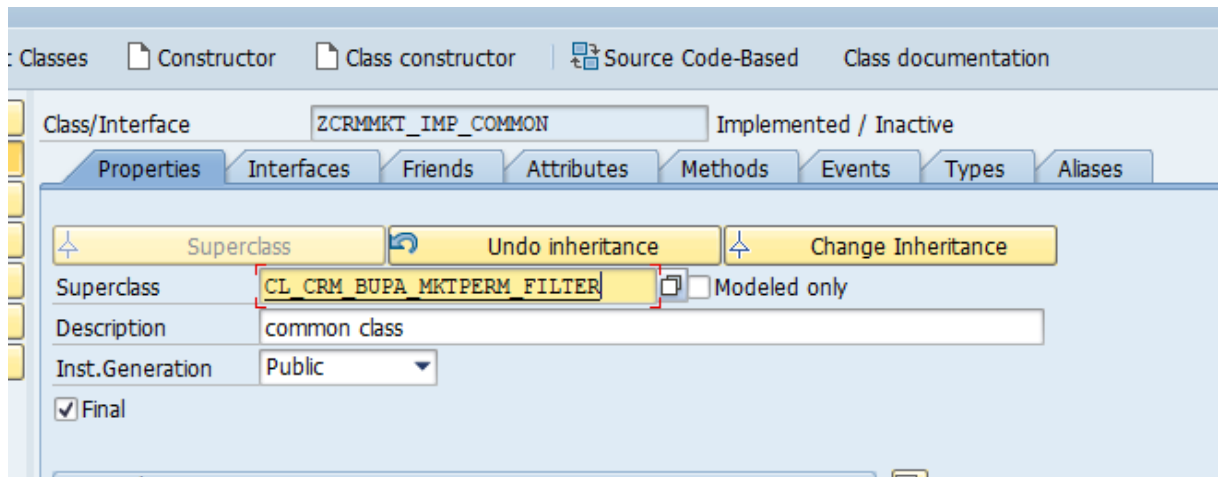
Change

Create

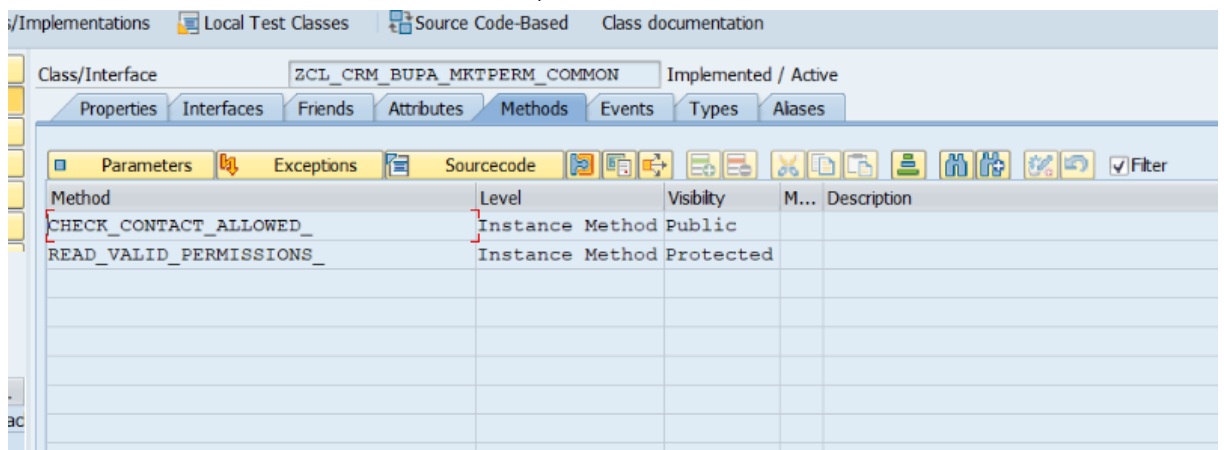
4. Go to the tab *Properties* and click on *Superclass*.



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, protected.





7. Select the method CHECK\_CONTACT\_ALLOWED\_, click on [Parameters](#), and enter the parameters as follows:

Class/Interface: ZCL\_CRM\_BUPA\_MKTPERM\_COMMON Implemented / Active

Parameters of Method: CHECK\_CONTACT\_ALLOWED\_

Parameter	Type	P...	O...	Typing Method	Associated Type	Default Value
IV_USAGE_DATE	Importing	<input type="checkbox"/>	<input checked="" type="checkbox"/>	Type	DATS	SY-DATLO
IV_CHANNEL	Importing	<input type="checkbox"/>	<input type="checkbox"/>	Type	CRMT_BU_MKTPERM_CHAN	
IT_STD_VALUE	Importing	<input type="checkbox"/>	<input checked="" type="checkbox"/>	Type	ZMKT_VALUES_T	
IV_CHECK_ACCOUNT_CONTACT	Importing	<input type="checkbox"/>	<input checked="" type="checkbox"/>	Type	CRMT_BOOLEAN	ABAP_TRUE
CT_MKTPERM_APPLICABLE	Changing	<input type="checkbox"/>	<input type="checkbox"/>	Type	ZMKT_APPLICABLE_T	
RT_VALID_PERMISSIONS	Returning	<input checked="" type="checkbox"/>	<input type="checkbox"/>	Type	CRMT_BUT_MKTPERM_ALL_T	

8. Select the method READ\_VALID\_PERMISSIONS\_, click on [Parameters](#), and enter the parameters as follows:

Class/Interface: ZCL\_CRM\_BUPA\_MKTPERM\_COMMON Implemented / Active

Parameters of Method: READ\_VALID\_PERMISSIONS\_

Parameter	Type	P...	O...	Typing Method	Associated Type	Default Value
IV_USAGE_DATE	Importing	<input type="checkbox"/>	<input checked="" type="checkbox"/>	Type	DATS	SY-DATLO
IV_OPTID	Importing	<input type="checkbox"/>	<input checked="" type="checkbox"/>	Type	CRMT_BU_MKTPERM_OPTID	
IV_IS_OPT_OUT	Importing	<input type="checkbox"/>	<input checked="" type="checkbox"/>	Type	CRMT_BU_MKTPERM_OPTOUT	
IV_CHANNEL	Importing	<input type="checkbox"/>	<input type="checkbox"/>	Type	CRMT_BU_MKTPERM_CHAN	
IV_CHECK_ACCOUNT	Importing	<input type="checkbox"/>	<input checked="" type="checkbox"/>	Type	CRMT_BOOLEAN	ABAP_TRUE
IS_APPLICABLE	Importing	<input type="checkbox"/>	<input type="checkbox"/>	Type	CRMT_BUT_MKP_APPLICABLE	
IT_PERMISSIONS	Importing	<input type="checkbox"/>	<input checked="" type="checkbox"/>	Type	CRMT_BUT_MKTPERM_ALL_T	
RT_VALID_PERMISSIONS	Returning	<input checked="" type="checkbox"/>	<input type="checkbox"/>	Type	CRMT_BUT_MKTPERM_ALL_T	

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
```

```
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
```

```

PARTNERGUID = GV_PARTNERGUID
CONTPGUID = Ic_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.

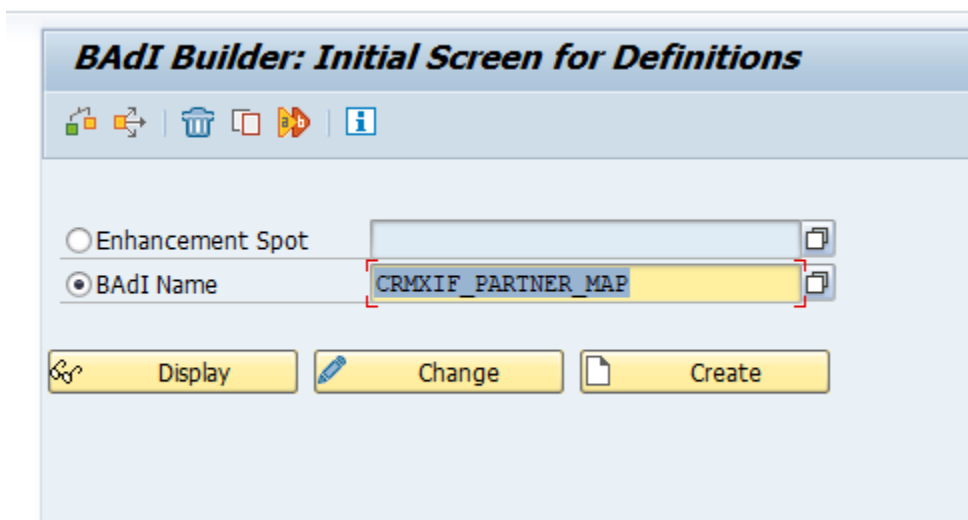
* delete redundant contact permission
LOOP AT LT_VALID_PERMISSIONS INTO LS_VALID_PERMISSIONS WHERE
PARTNERGUID = GV_PARTNERGUID AND
CONTPGUID = GV_CONTPGUID AND
CHANNEL = IV_CHANNEL AND
PERMISSION = cl_crm_bupa_mktperm_tools=>GC_GIVEN.
    READ TABLE LT_VALID_PERMISSIONS WITH KEY
        PARTNERGUID = 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.

```

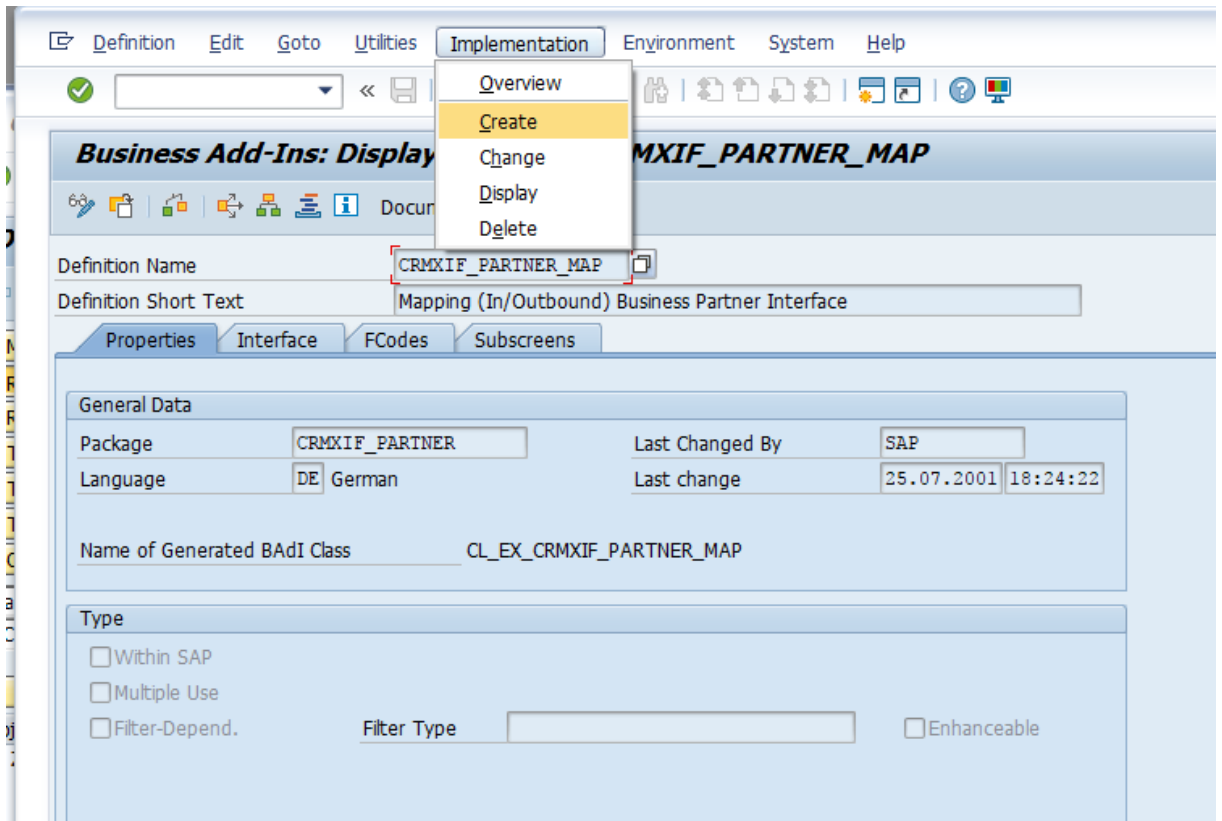
12. Activate your implementation.

## 2.3 BAdI CRMXIF\_PARTNER\_MAP (Marketing Permissions for Accounts)

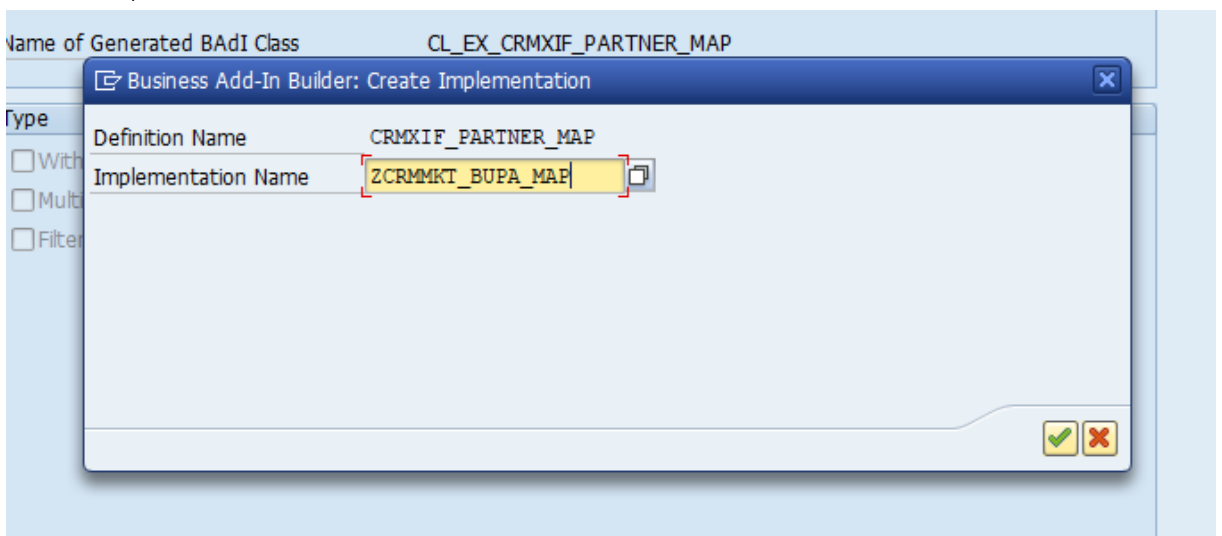
1. Start transaction SE18, select BAdI *Name CRMXIF\_PARTNER\_MAP*, and choose *Display*.



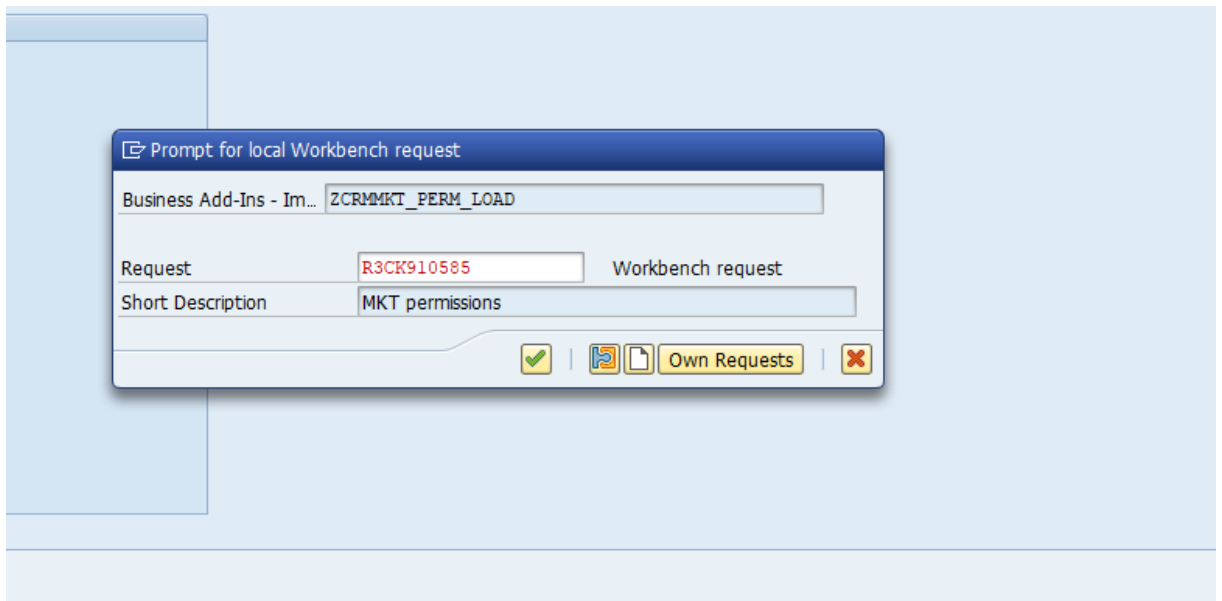
2. Choose *Implementation -> Create*.



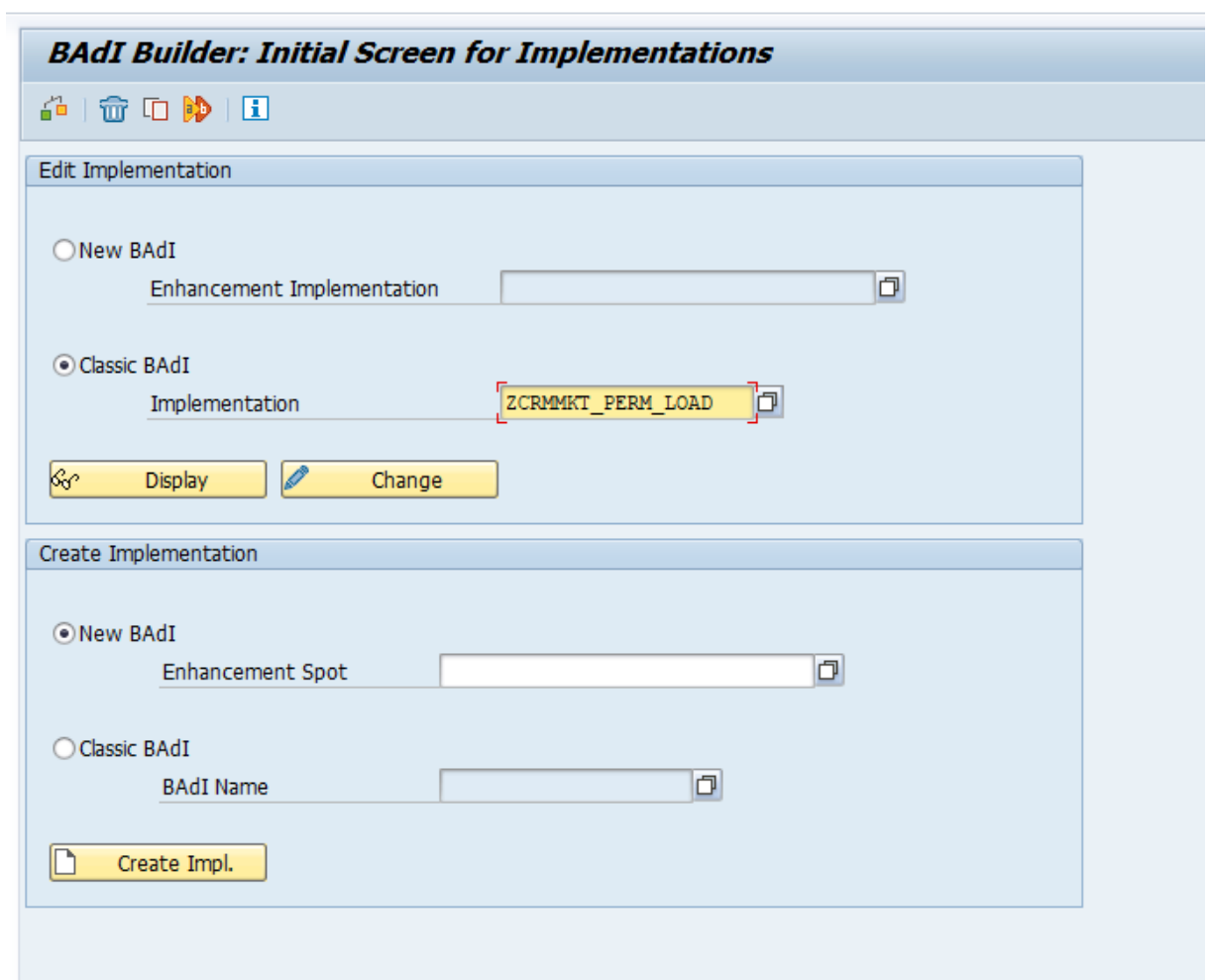
3. Enter an Implementation Name Z or Y<CRMMKT\_BUPA\_MAP> and click *OK*.



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.



7. Start transaction SE19.
8. Search for the implementation, for example Z or Y<CRMMKT\_PERM\_LOAD>.



9. Click on *Display*.

The screenshot shows the SAP Business Add-In Builder interface. The title bar reads "Business Add-In Builder: Display Implementation ZCRMMKT\_PERM\_LOAD". The menu bar includes "Implementation", "Edit", "Go to", "Guides", "Environment", "System", and "Help". The toolbar contains various icons for file operations and navigation. Below the title bar, there are tabs for "Definition", "Documenta", and "Documentation". The main area is divided into two sections: "Properties" and "Interface". The "Properties" section contains the following fields:

Implementation Name	ZCRMMKT_PERM_LOAD	<input type="checkbox"/> Active
Implementation Short Text	Implementation BAdI CRMXIF_PARTNER_MAP	
Definition Name	CRMXIF_PARTNER_MAP	
Runtime Behavior	Implementation will not be called	

The "Interface" section contains the following fields:

General Data	
Package	ZCRMMKT_IMP
Language	EN English
Last Changed By	HOHO
Last change	20.02.2018 14:34:34

The "Type" section contains the following options:

- ☐ Within SAP
- ☐ Multiple Use
- ☐ Filter-Depend.

10. Go to the tab *Interface*.

**Business Add-In Builder: Display Implementation ZCRMMKT\_PERM\_LOAD**

Definition Documenta Documentation

Implementation Name: ZCRMMKT\_PERM\_LOAD Inactive

Implementation Short Text: implementation BAdI CRMXIF\_PARTNER\_MAP

Definition Name: CRMXIF\_PARTNER\_MAP

Runtime Behavior: Implementation will not be called

Properties Interface

Interface name: IF\_EX\_CRMXIF\_PARTNER\_MAP

Name of implementing class: ZCL\_IM\_CRMMKT\_PERM\_LOAD

Method	Implemen...	Description
CHANGE_MAPPED_DATA_OUT	ABAP Co...	Change Mapping in Outbound Process
CHANGE_MAPPED_DATA_IN	ABAP Co...	Change Mapping in Inbound Processin

Default implementation class

Example implementation class

11. Double click on the *Implementing Class*.

**Class Builder: Display Class ZCL\_IM\_CRMMKT\_PERM\_LOAD**

Local Definitions/Implementations Source Code

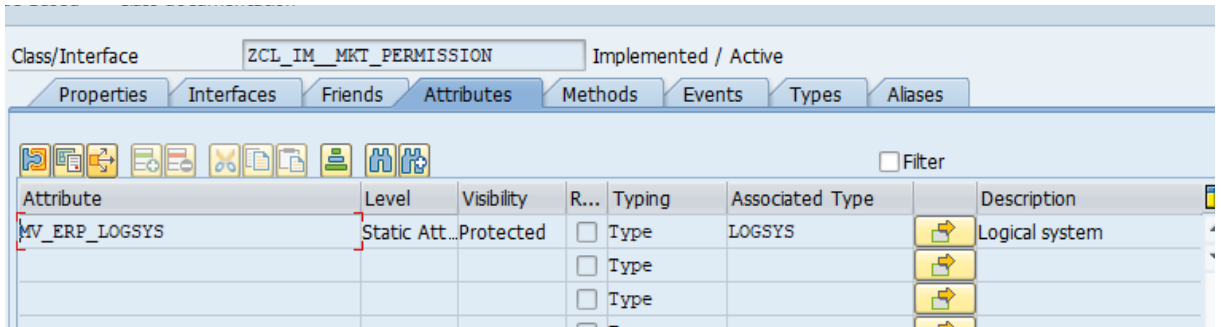
Class/Interface: ZCL\_IM\_CRMMKT\_PERM\_LOAD Implemented / Inactive

Properties Interfaces Friends Attributes Methods Events Types Aliases

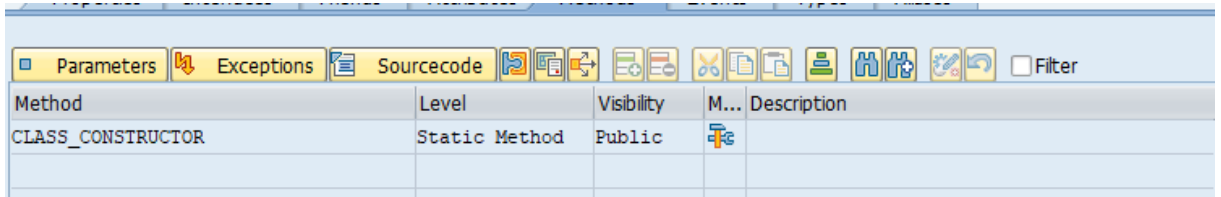
Parameters Exceptions Sourcecode

Method	Level	Visibility	M...	Description
IF_EX_CRMXIF_PARTNER_MAP~CHANGE_MAP...	Instance Method	Public		Change Mapping in Outbound Processing
IF_EX_CRMXIF_PARTNER_MAP~CHANGE_MA...	Instance Method	Public		Change Mapping in Inbound Processing

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_ersites TYPE TABLE OF smof_erpsh.
DATA ls_ersites TYPE smof_erpsh.

CALL FUNCTION 'SMOF_READ_SMOFERPSH'
EXPORTING
  i_sitetypeid = 'SMOF_ERPSITE'
TABLES
  t_ersites = lt_ersites.

  CHECK LINES( lt_ersites ) = 1.
  READ TABLE lt_ersites INTO ls_ersites INDEX 1.
  mv_erp_logsys = ls_ersites-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,

```



```

        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

```

```

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.
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'.
  Is_value-VALUE = Is_pager-CONTACT-DATA-PAGER_NO.
  Is_value-STD = Is_pager-CONTACT-DATA-STD_NO.
  APPEND Is_value TO It_value.
WHEN OTHERS.
ENDCASE.

IF It_value IS INITIAL.
  CONTINUE.
ENDIF.

CALL METHOD LO_ZCL_CRM_BUPA_MKTPERM->CHECK_CONTACT_ALLOWED_
EXPORTING
  IV_CHANNEL      = Is_channel-DATA-CHANNEL
  IT_STD_VALUE    = It_value
  IV_CHECK_ACCOUNT_CONTACT = abap_false
CHANGING
  CT_MKTPERM_APPLICABLE = It_mkp_applicable
RECEIVING
  RT_VALID_PERMISSIONS = It_valid_permissions.

IF It_valid_permissions IS INITIAL OR It_mkp_applicable IS INITIAL.
  CONTINUE.
ENDIF.

*****
LOOP AT It_valid_permissions INTO Is_valid_permission.
  IF Is_valid_permission-VALUE IS INITIAL AND Is_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 Is_channel-DATA-CHANNEL EQ 'TEL'.
    IF Is_valid_permission-VALUE IS INITIAL.
      READ TABLE It_value INTO Is_value WITH KEY
        STD = 'X'.
      IF sy-subrc EQ 0.
        IF Is_value-R_3_USER = '2' OR Is_value-R_3_USER = '3'.
          <mkt_permission>-DATA-CHANNEL = 'TE2'.
        ENDIF.
      ELSE.
        READ TABLE Is_address-DATA-COMMUNICATION-PHONE-PHONE INTO Is_phone
          WITH KEY CONTACT-DATA-TEL_NO = Is_valid_permission-value.
        IF sy-subrc EQ 0.
          IF Is_phone-CONTACT-DATA-R_3_USER = '2' OR Is_phone-CONTACT-DATA-R_3_USER = '3'.
            <mkt_permission>-DATA-CHANNEL = 'TE2'.
          ENDIF.
        ENDIF.
      ENDIF.
    ENDIF.
  ENDIF.

  *check against mobile phone
  IF <mkt_permission>-DATA-CHANNEL IS INITIAL.
    <mkt_permission>-DATA-CHANNEL = Is_channel-DATA-CHANNEL.
  ENDIF.

  *   simulate CPI conversion of time stamp; add initial time to timestamp valid from date is not in future
  CONVERT DATE Is_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

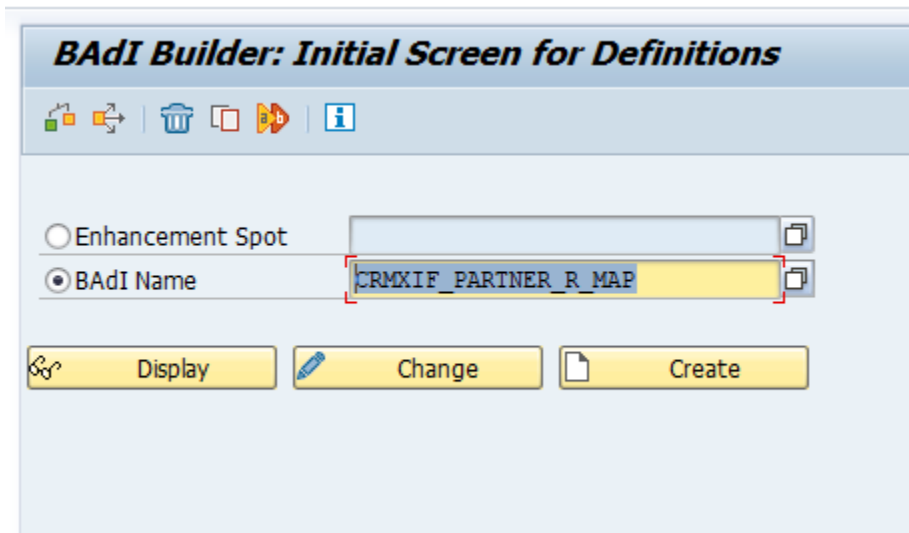
*move all redundant entries
    LOOP AT <mkt_permission_t> ASSIGNING <mkt_permission_channel>.
*move all older entries
    LOOP AT <mkt_permission_t> ASSIGNING <mkt_permission>
        WHERE DATA-CHANNEL EQ <mkt_permission_channel>-DATA-CHANNEL
        *
            AND DATA-VALID_FROM LT <mkt_permission_channel>-DATA-VALID_FROM
            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.

```

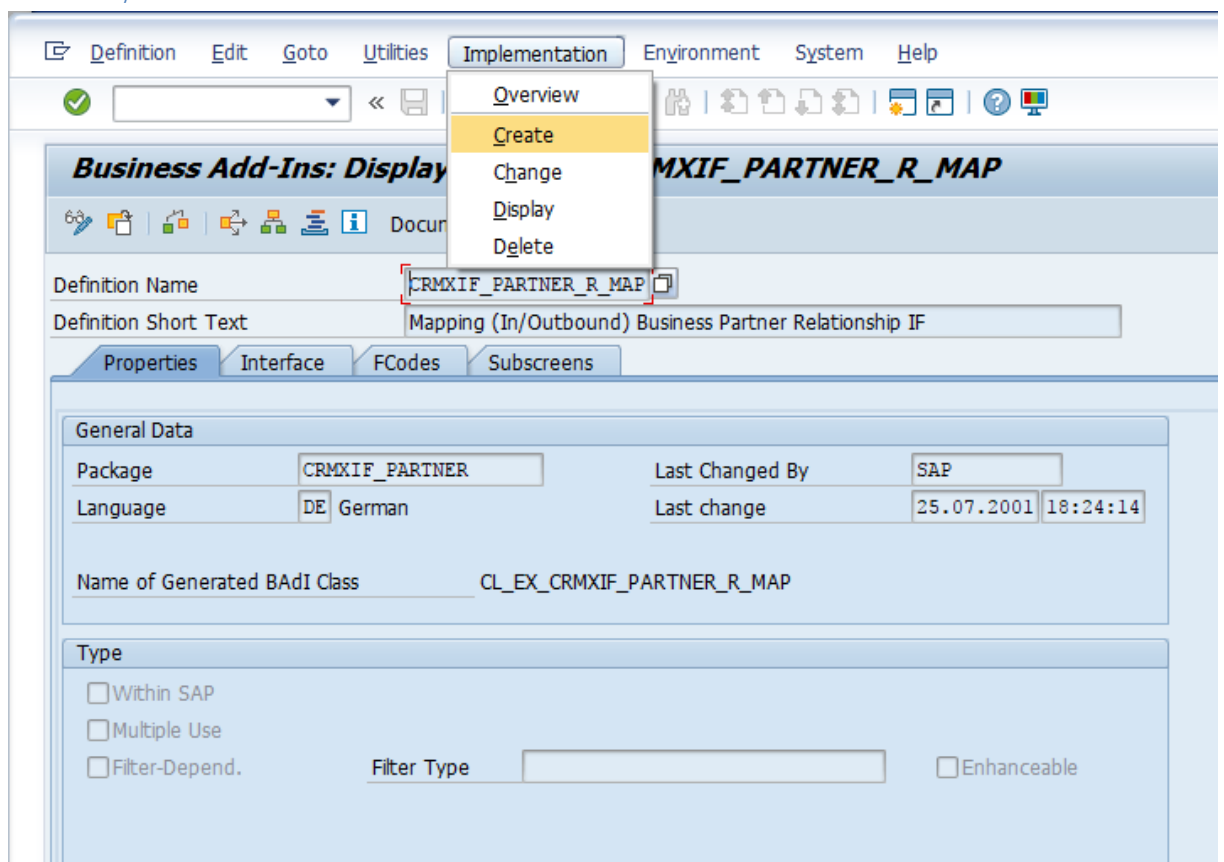
15. Activate your implementation.

## 2.4 BAdI CRMXIF\_PARTNER\_R\_MAP (Marketing Permissions for Contacts)

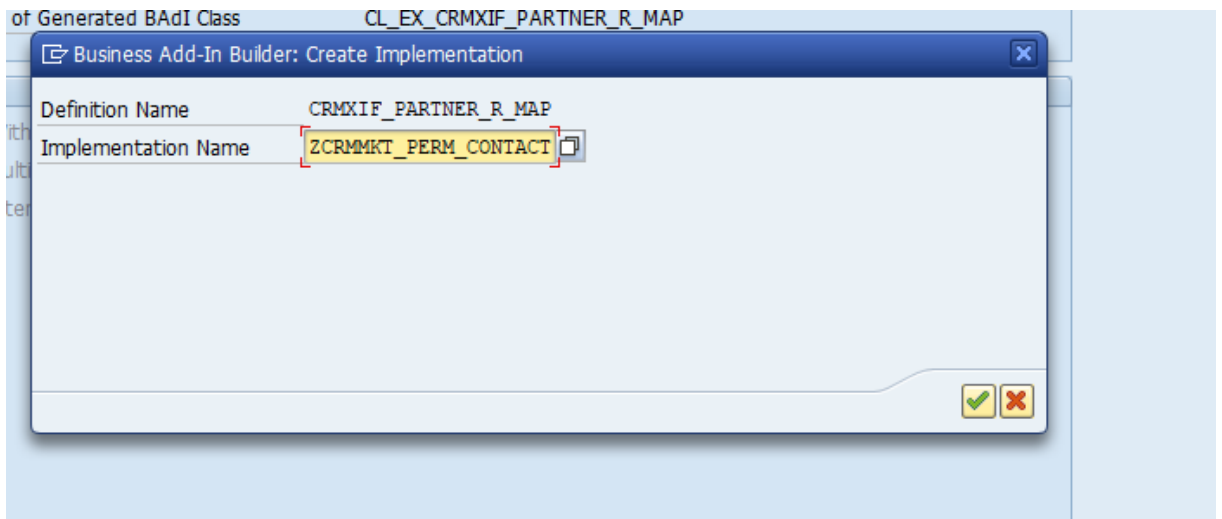
1. Start SE18, select BAdI Name `CRMXIF_PARTNER_R_MAP`, and click on *Display*.



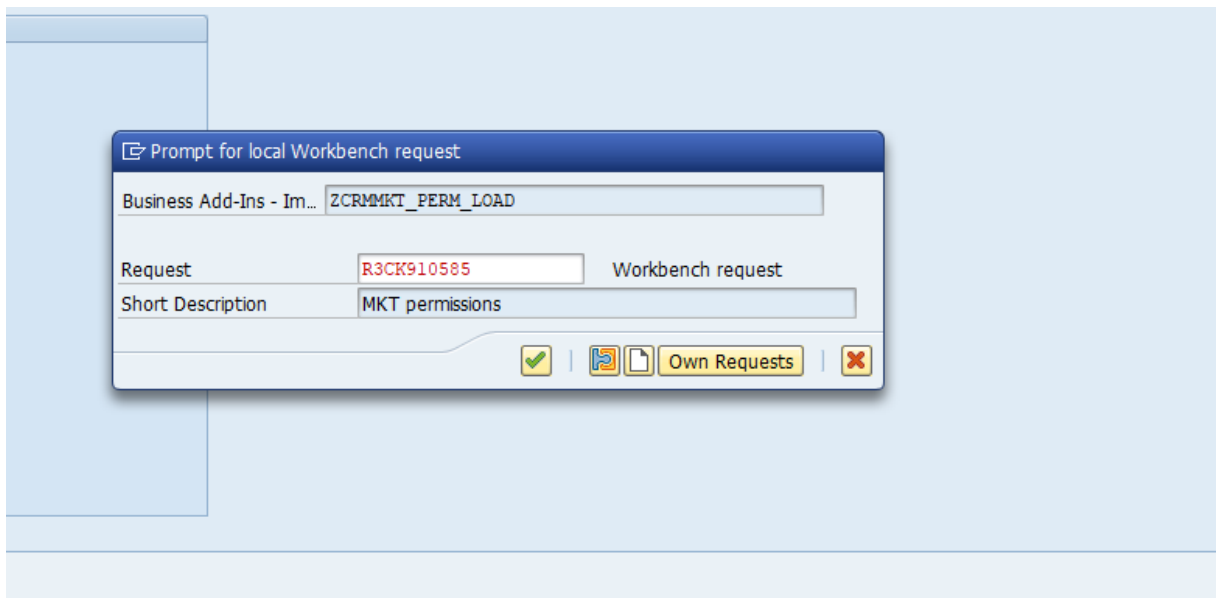
2. Choose *Implementation-> Create*.



3. Enter an Implementation Name Z or Y<CRMMKT\_PERM\_CONTACT> and click *OK*.



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.

**BAdI Builder: Initial Screen for Implementations**

☐ New BAdI  
 Enhancement Implementation

☒ Classic BAdI  
 Implementation

---

**Create Implementation**

☒ New BAdI  
 Enhancement Spot

☐ Classic BAdI  
 BAdI Name

9. Click on *Display*.

**Business Add-In Builder: Display Implementation ZCRMMKT\_PERM\_CONTACT**

Definition Documenta Documentation

Implementation Name  ☐ nactive

Implementation Short Text

Definition Name

Runtime Behavior

**Properties** **Interface**

**General Data**

Package

Language  English

Last Changed By

Last change

**Type**

☐ Within SAP

☐ Multiple Use

☐ Filter-Depend.

10. Go to the tab *Interface*.



**Business Add-In Builder: Display Implementation ZCRMMKT\_PERM\_CONTACT**

Definition Documenta Documentation

Implementation Name: ZCRMMKT\_PERM\_CONTACT Inactive

Implementation Short Text: Implementing CRMXIF\_PARTNER\_R\_MAP

Definition Name: CRMXIF\_PARTNER\_R\_MAP

Runtime Behavior: Implementation will not be called

Properties Interface

Interface name: IF\_EX\_CRMXIF\_PARTNER\_REL\_MAP

Name of implementing class: ZCL\_IM\_CRMMKT\_PERM\_CONTACT

Method	Implemen...	Description
CHANGE_MAPPED_DATA_OUT	ABAP Co...	Change Mapping in Outbound Process
CHANGE_MAPPED_DATA_IN	ABAP Co...	Change Mapping in Inbound Processin

Default implementation class

Example implementation class

- Double click on the *Implementing Class*.

**Class Builder: Display Class ZCL\_IM\_CRMMKT\_PERM\_CONTACT**

Local Definitions/Implementations Source Code-Based

Class/Interface: ZCL\_IM\_CRMMKT\_PERM\_CONTACT Implemented / Inactive

Properties Interfaces Friends Attributes Methods Events Types Aliases

Parameters Exceptions Sourcecode

Method	Level	Visibility	M...	Description
IF_EX_CRMXIF_PARTNER_REL_MAP~CHANGE	Instance Method	Public		Change Mapping in Outbound Processing
IF_EX_CRMXIF_PARTNER_REL_MAP~CHANG...	Instance Method	Public		Change Mapping in Inbound Processing

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



*\* 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\_zeit,  
      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

lv\_org\_guid = lv\_org\_guid

lv\_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.

* 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.
    ENDIF.
  ENDIF.

```

```

DELETE <mkt_permission_t>.
ENDLOOP.

LOOP AT It_channel INTO Is_channel.
  clear It_mkp_applicable.
  clear It_valid_permissions.
  clear It_value.
  CASE Is_channel-DATA-CHANNEL.
*   TEL -> PHONE
    WHEN 'TEL'.
      LOOP AT It_phone INTO Is_phone.
        Is_value-VALUE = Is_phone-CONTACT-DATA-TEL_NO.
        Is_value-STD = Is_phone-CONTACT-DATA-STD_NO.
        Is_value-R_3_USER = Is_phone-CONTACT-DATA-R_3_USER.
        APPEND Is_value TO It_value.
      ENDLOOP.
*   FAX -> FAX
    WHEN 'FAX'.
      Is_value-VALUE = Is_fax-CONTACT-DATA-FAX_NO.
      Is_value-STD = Is_fax-CONTACT-DATA-STD_NO.
      APPEND Is_value TO It_value.
*   INT -> EMAIL
    WHEN 'INT'.
      Is_value-VALUE = Is_email-CONTACT-DATA-E_MAIL.
      Is_value-STD = Is_email-CONTACT-DATA-STD_NO.
      APPEND Is_value TO It_value.
*   PAG -> SMS
    WHEN 'PAG'.
      Is_value-VALUE = Is_pager-CONTACT-DATA-PAGER_NO.
      Is_value-STD = Is_pager-CONTACT-DATA-STD_NO.
      APPEND Is_value TO It_value.
    WHEN OTHERS.
  ENDCASE.

  IF It_value IS INITIAL.
    CONTINUE.
  ENDIF.

  CALL METHOD LO_ZCL_CRM_BUPA_MKTPERM->CHECK_CONTACT_ALLOWED_
    EXPORTING
      IV_CHANNEL      = Is_channel-DATA-CHANNEL
      IT_STD_VALUE     = It_value
      IV_CHECK_ACCOUNT_CONTACT = abap_true
    CHANGING
      CT_MKTPERM_APPLICABLE = It_mkp_applicable
    RECEIVING
      RT_VALID_PERMISSIONS = It_valid_permissions.

  IF It_valid_permissions IS INITIAL OR It_mkp_applicable IS INITIAL.
    CONTINUE.
  ENDIF.
  LOOP AT It_valid_permissions INTO Is_valid_permission.
    IF Is_valid_permission-VALUE IS INITIAL AND It_value IS INITIAL.
      CONTINUE.
    ENDIF.
    APPEND INITIAL LINE TO <mkt_permission_t> ASSIGNING <mkt_permission>.
    lv_index = sy-TABIX.
*check against mobile phone
    IF Is_channel-DATA-CHANNEL EQ 'TEL'.
      IF Is_valid_permission-VALUE IS INITIAL.
        READ TABLE It_value INTO Is_value WITH KEY
          STD = 'X'.
        IF sy-subrc EQ 0.
          IF Is_value-R_3_USER = '2' OR Is_value-R_3_USER = '3'.
            <mkt_permission>-DATA-CHANNEL = 'TE2'.
          ENDIF.
        ELSE.
          READ TABLE Is_address-DATA-COMMUNICATION-PHONE-PHONE INTO Is_phone
            WITH KEY CONTACT-DATA-TEL_NO = Is_valid_permission-value.
          IF sy-subrc EQ 0.
            IF Is_phone-CONTACT-DATA-R_3_USER = '2' OR Is_phone-CONTACT-DATA-R_3_USER = '3'.
              <mkt_permission>-DATA-CHANNEL = 'TE2'.
            ENDIF.
          ELSE.
            <mkt_permission>-DATA-CHANNEL = 'TE1'.
          ENDIF.
        ENDIF.
      ENDIF.
    ELSE.
      <mkt_permission>-DATA-CHANNEL = 'TE1'.
    ENDIF.
  ENDLOOP.

```

```

        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

```

```

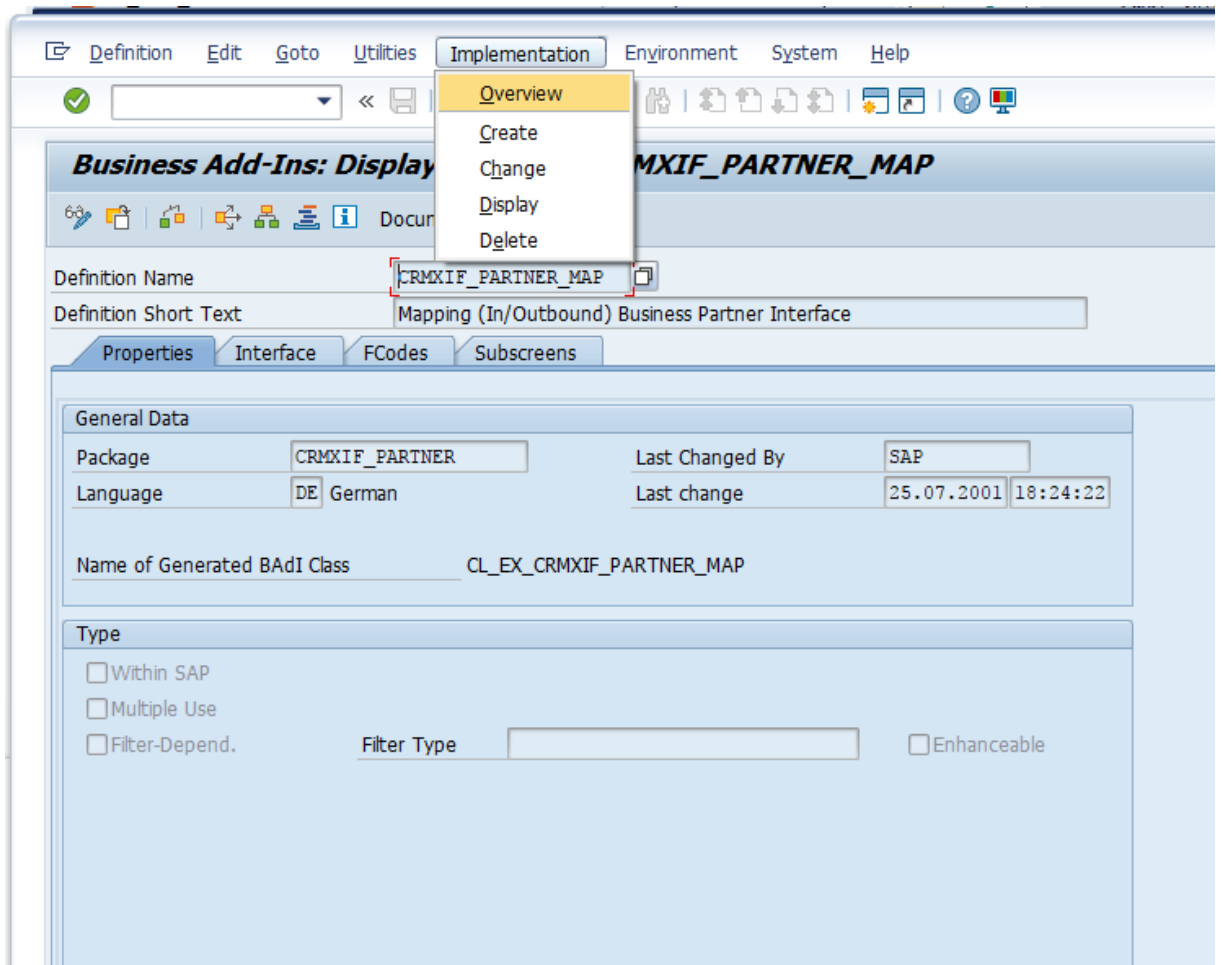
*move all redundant entries
LOOP AT <mkt_permission_t> ASSIGNING <mkt_permission_channel>.
LOOP AT <mkt_permission_t> ASSIGNING <mkt_permission>
WHERE DATA-CHANNEL EQ <mkt_permission_channel>-DATA-CHANNEL
AND DATA-VALUE EQ <mkt_permission_channel>-DATA-VALUE.
*move all older entries
IF <mkt_permission>-DATA-VALID_FROM 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.

```

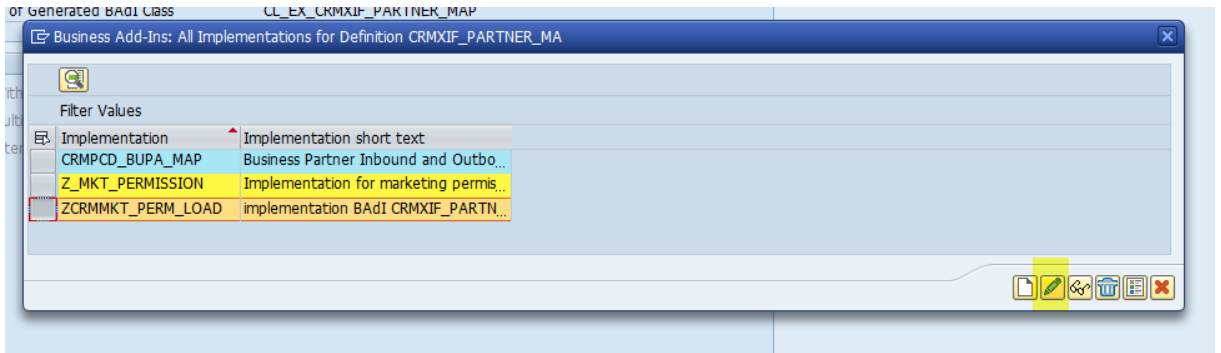
15. Activate your implementation.

## 2.5 Activate BAdI Implementations

1. Start transaction SE18.
2. Display BAdI CRMXIF\_PARTNER\_MAP.
3. Choose *Implementation->Overview*.

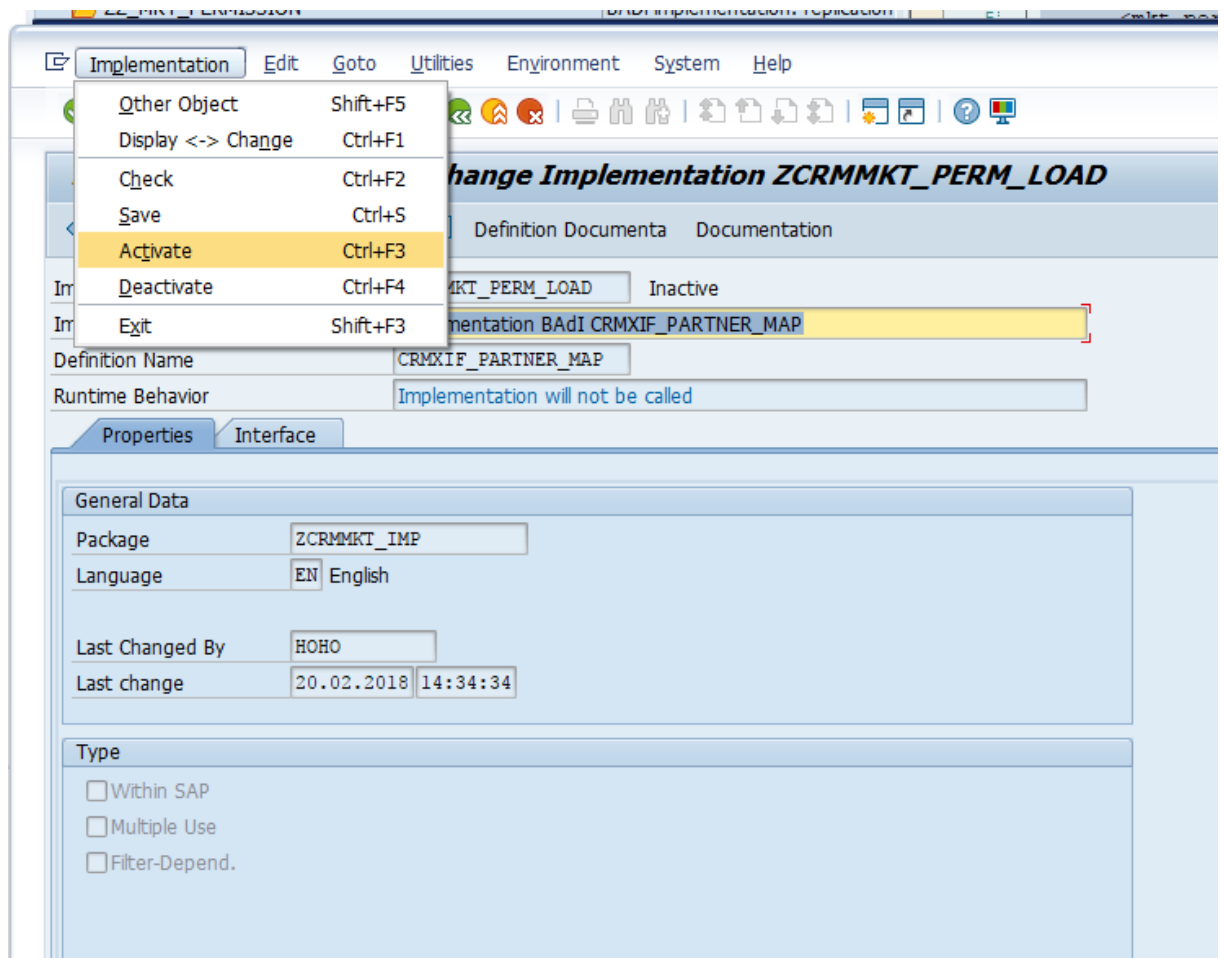


4. Select the implementation and click on *Change* (F6).

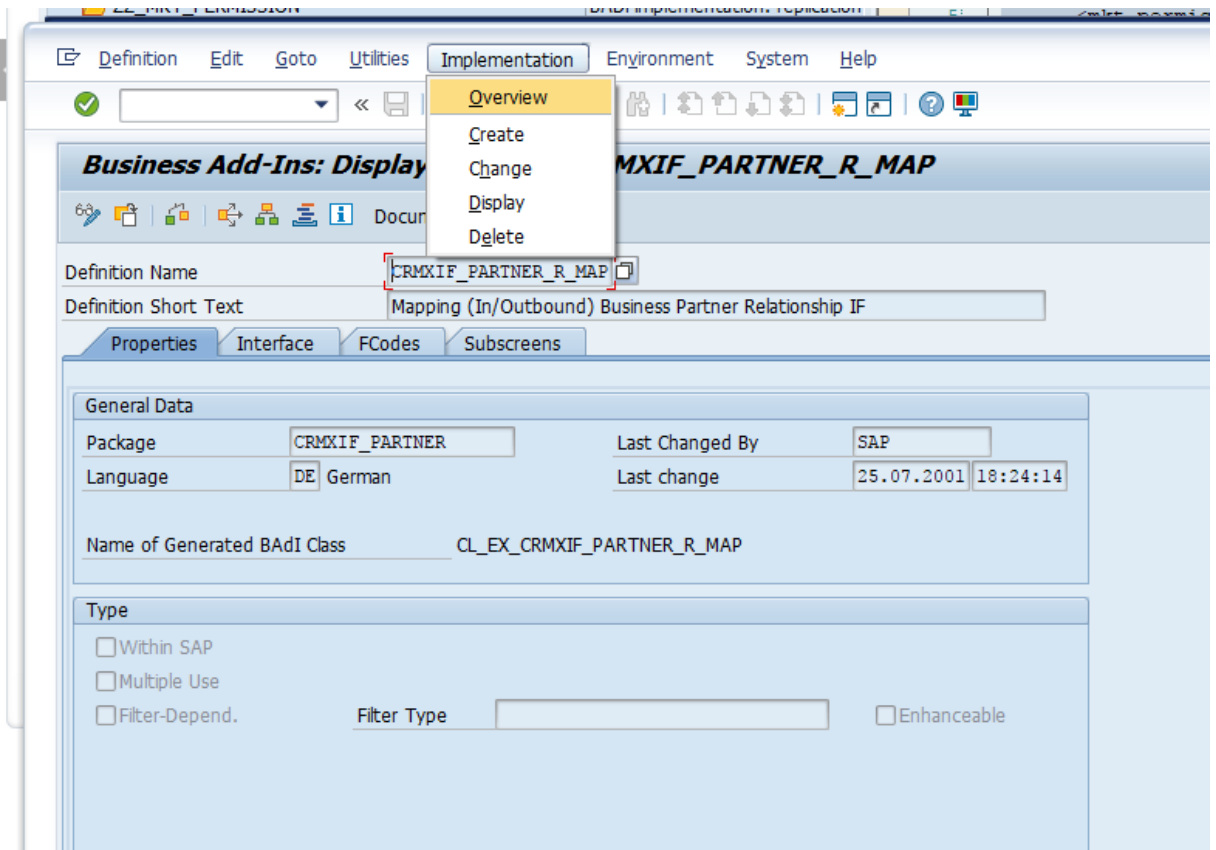


5. Choose *Implementation->Activate* (Ctrl + F3).

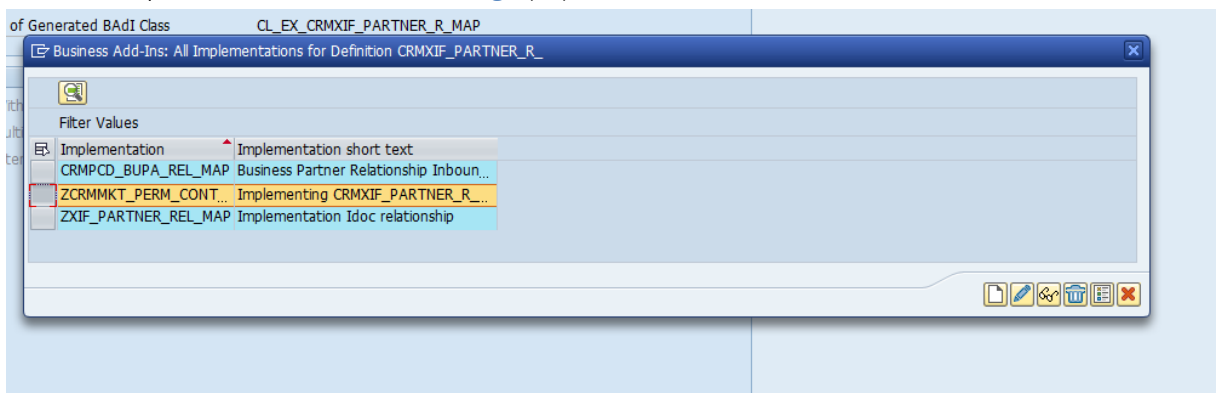




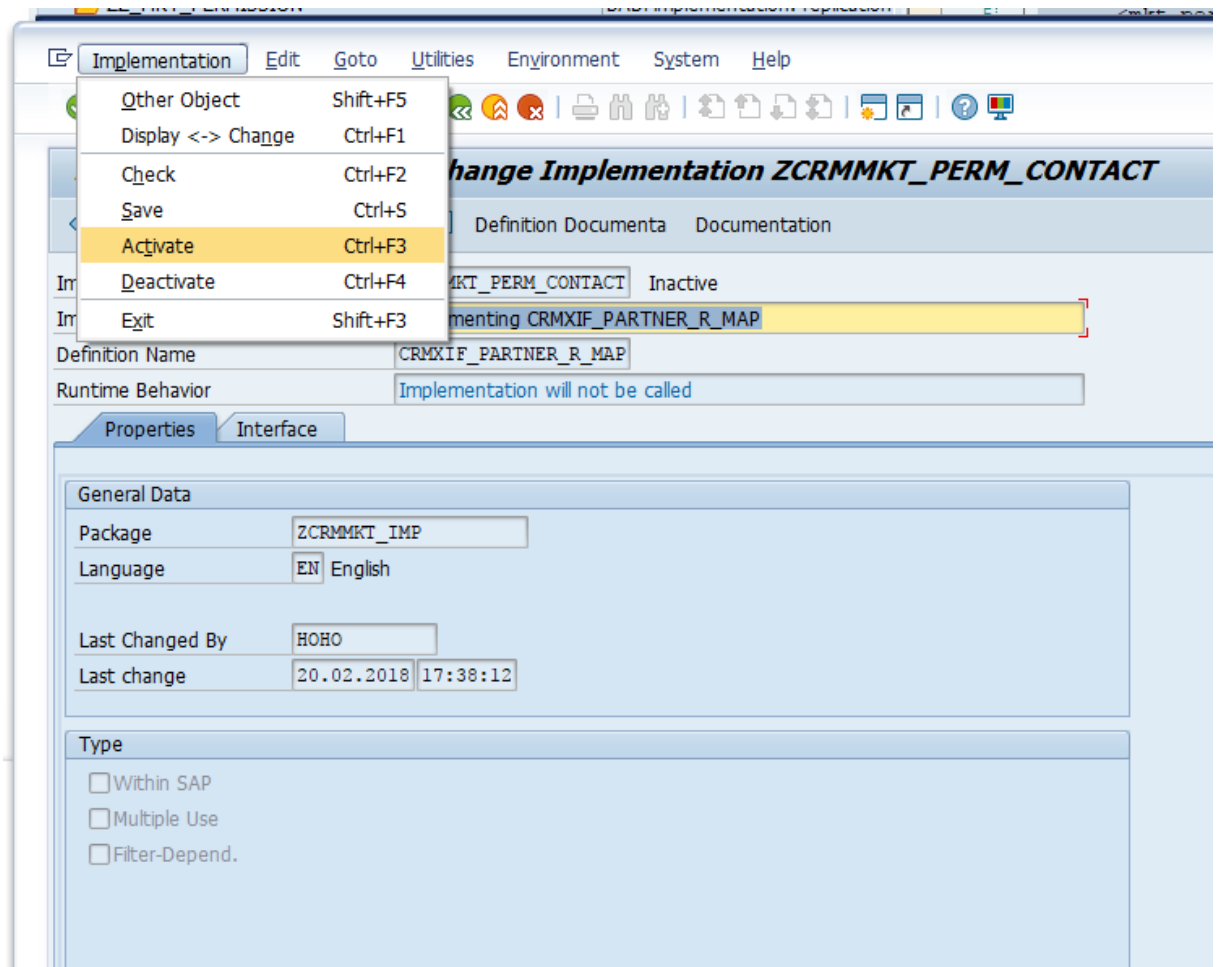
6. Go back to SE18.
7. Display BAdI CRMXIF\_PARTNER\_R\_MAP.
8. Choose *Implementation->Overview*.



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



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



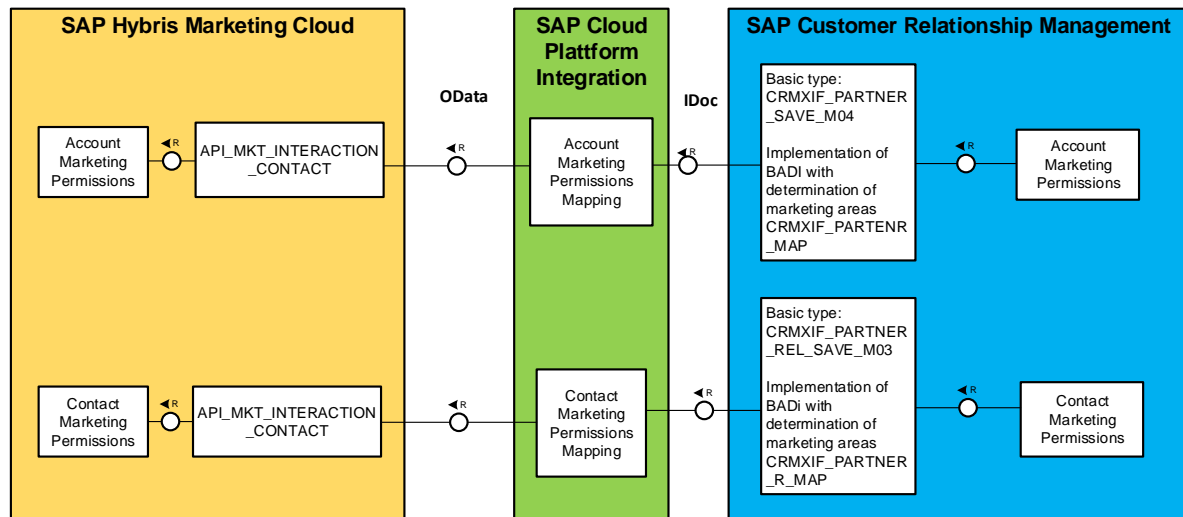
## 3 Extended Integration Scenario for Marketing Areas Determination

### 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,



## 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*.

SAP Sales Professional

Home

My Appointments Today

No result found

Workflow Tasks

Prices

Products

Competitor Products

Employees

Product Catalog

Activity Journal Templates

Activity Journal Templ. Types

Document Templates

Organizational Model

Billing Documents

Organizational Model

Territories

Rule Policies

Rule Policies for Rule Builder

Job Monitoring

Surveys

Mail Forms

Web Links

No result found

CRM Links

No result found

SAP Jam Feeds

Home

Worklist

Calendar

E-Mail Inbox

Account Management

Activities

Sales Cycle

Pipeline Performance

Sales Operations

Reports

Dashboards

Create

Appointment

Interaction Log

Task

E-Mail

Contact

Lead

Opportunity

Quotation

Interactive Report

Recent Items

contact 2 HOHO

test contact 506622...

HOHO MP

BP\_COMP

BP\_HEAD

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*.

**Organizational Unit: Marketing, 18.04.2018 - 31.12.9999**

Save | Cancel | Set Preview Period

- Sales
- Service
- ▾ Marketing
  - Marketing North
  - ▾ Marketing South
    - Area München
    - Area Stuttgart
    - ▾ Area Mannheim
      - Hoang Ti No Ho
    - Marketing West
    - Marketing East

▼ Organizational Unit Details | Edit | Change Period General Data | Change Period Address

---

**General Data** Validity

Hierarchy Path: Global Company > Headquarters  
 Description: Marketing  
 Code: MKT\_ORG

---

**Address** Validity

Street/House No.:  
 City:  
 Postal code:  
 Country:  
 Region:  
 Phone:  
 Fax:

---

**Functions** Belongs To

Sales Organization: ☐  
 Sales Office: ☐  
 Sales Group: ☐  
 Service Organization: ☐  
 Service Team: ☐  
**Marketing Organization: ☒**

---

**Allow Org. Unit to be Determined**

**Marketing: ☒**  
 Sales: ☐  
 Service: ☐

▼ Attributes | Edit List

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*.

Position: Area Mannheim, 18.04.2018 - 31.12.9999

Save | Cancel | Set Preview Period

- Sales
- Service
- ▾ Marketing
  - Marketing North
  - ▾ Marketing South
    - Area München
    - Area Stuttgart
    - ▾ Area Mannheim
      - Hoang Ti No Ho
      - Marketing West
      - Marketing East

Position Details | Edit | Change Period General Data

General Data		Validity
Description:	Area Mannheim	Valid From: 16.04.2018
Code:	MKT_MA	Valid To: 31.12.9999
Head of Org. Unit:	<input type="checkbox"/>	

Role Assignment		Validity
Business Role:	Marketing Professional	Valid From: 16.04.2018
		Valid To: 31.12.9999

Functional Assignments	
Sales Organization:	
Sales Office:	
Sales Group:	
Service Organization:	
Service Team:	
Marketing Organization:	

Marketing: ☒  
 Sales: ☐  
 Service: ☐

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.

Organizational Unit: Global Company, 18.04.2018 - 31.12.9999

Save | Cancel | Set Preview Period

- Sales
- Service
- ▾ Marketing
  - Marketing North
  - Marketing South
  - ▾ Marketing West
    - ▾ Area Düsseldorf
      - Hoang Ti No Ho
      - Area Aachen
  - Marketing East

Organizational Unit Details | Edit | Change Period General Data

**Organizational Unit: Marketing West, 18.04.2018 - 31.12.9999**

Save | Cancel | Set Preview Period

- ▶ Sales
- ▶ Service
- ▼ Marketing
  - ▶ Marketing North
  - ▼ Marketing South
    - ▶ Area München
    - Area Stuttgart
    - ▼ Area Mannheim
      - ▶ Hoang Ti No Ho
  - ▶ Marketing West
  - ▶ Marketing East

▼ Organizational Unit Details   Edit | Change Period General Data   Change Period

General Data

Hierarchy Path:   Global Company > Headquarters >

### 3.4 Maintain Employee Responsible in Master Data

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



← → SAP https://ldcir3c.wdf.sap.corp:44317/sap(bD1biZjPTA4MiZkPW1pbG==)/bc/bsp/sap/crm\_u

HDM ▾ SAP Hybris Marketing Cloud w... A2X - Wiki@SAP A2A-B2B Service Interface...

## SAP Sales Professional

**Organizational Unit: Marketing for foods, 17.04.2018**

Save | Cancel | Set Preview Period

Home

Worklist

Calendar

E-Mail Inbox

Account Management ▸

Activities ▸

Sales Cycle ▸

Pipeline Performance ▸

Sales Operations ▸

Reports

Dashboards ▸

Create

Appointment

Interaction Log

Task

E-Mail

Contact

▼ Organizational Unit

Organizational Unit Position Employee User

Description

Accounts

Contacts

Account Hierarchies

Account Plans

▶ Hoang Ti No Ho

▶ Ingo Fenske

▶ Cow

▶ Pick

▶ Fish

▶ Marketing for drinks

▼ Organizational Unit Details Edit | Change Period Go

**General Data**

2) Search for your account or contact.

**Search: Accounts**

**Search Criteria**

Search for:

Name 1 / Last Name	is	<input type="text"/>	+ -
City	is	<input type="text"/>	+ -
Role	is	<input type="text"/>	+ -
Account ID	is	50662245	+ -

Maximum Number of Results:

Save Search As:  ☐ Include View

**Result List: 1 Account Found**

☐ Corporate Account
 ☐ Individual Account
 ☐ Group
 
☐ Employee Responsible
 ☐ Marketing Attribute

ID	Name	Phone	Street
50662245	HOHO MP	+49 (621) 777777	Lampertheimerstrasse 1

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

Vertieb Sud      Distribution channel 01      Product Division 10

**Relationships**

Actions	Relationship	Name	Standard
<input type="button" value="New"/>	Has the Employee Responsible	Matthew Carter / Palo Alto CA 94304	<input type="checkbox"/>
<input type="button" value="New"/>	Has the Employee Responsible	Hoang Ti No Ho / D-68305 Mannheim	<input type="checkbox"/>

**Attachments**

**Business Appointments**

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

**Corporate Account: HOHO D-68305 MANNHEIM - Relationships**

**Relationships**

Relationship:

Partner:

Partner ID:

Standard: ☐

Valid From:

Valid To:

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

Corporate Account: HOHO D-68305 MANNHEIM - Relationships

Back

Relationships

Relationship:

Partner:

Partner ID:

Standard: ☐

Valid From:

Valid To:

Relationship List:

- CRMM03 Has Correspondence Recipient
- BUR020 Has Department
- CRMM04 Has Dunning Recipient
- CRME02 Has End Customer
- ICM003 Has External Person
- CRMM02 Has Payee
- CRME03- Has Service Center
- CRME01- Has Trader
- CRMH00 Has an undefined relationship with
- CRMM01 Has the Alternative Payer
- CRMH04 Has the Bill-To Party
- CRMS01 Has the Carrier
- BUR011 Has the Employee Responsible**
- ICM002 Has the External Organization
- CRMS09- Has the Remanufacturer
- CRMH06 Has the Responsible Business Partner
- ICM001 Has the Responsible Organization
- CHM005 Has the Responsible Partner Contact
- CRMS03- Has the Responsible Scrapper
- CHM002 Has the Ship-To Party/Recipient

- 7) Select the employee responsible.

Professional

Corporate Account: HOHO D-68305 MANNHEIM - Relationships

Back

Relationships

Relationship:

Partner:

Partner ID:

Standard: ☐

Valid From:

Valid To:

Partner Function Assignments

Insert

Sales Org. ID	Sales Org.	Distr. Channel ID	Distr. Channel

- 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.

Transparent Table  Active

Short Description

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

Field	Key	Ini...	Data element	Data Type	Length	Deci...	Short Description
CLIENT	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	MANDT	CLNT	3	0	Client
ZZMKT_ORG	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	CRM_MKTPL_MORG	CHAR	14	0	Marketing Organization
ZZYMKT_AREA	<input type="checkbox"/>	<input type="checkbox"/>	CHAR40	CHAR	40	0	Character field of length 40

### Dictionary: Display Technical Settings

Revised<->Active

Name  ☐ Transparent Table

Short Descript.

Last Changed

Status

General Properties DB-Specific Properties

Logical Storage Parameters

Data Class  Organization and customizing

Size Category  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

☐ Log Data Changes

☐ Write access only with JAVA

2. Create a new append structure Z or Y<MKT\_AREAS\_APPEND>.

Append Structure:  Active

Short Description:

Attributes Components Entry help/check Currency/quantity fields

Predefined Type Show Appending Obj 1 / 5

Component	Typing Method	Component Type	Data Type	Length	Deci...	Short Description
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. Start transaction SE11.
4. Select Data type and enter CRMT\_BUT\_MKTPERM\_DATA\_CHNG.
5. Click on *Display*.

ABAP Dictionary: Initial Screen

Database table

View

☒ Data type

Type Group

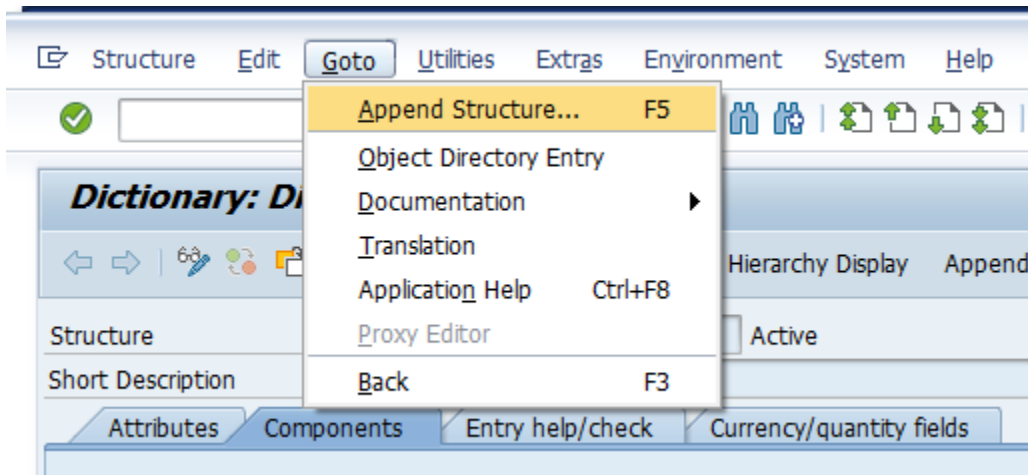
Domain

Search help

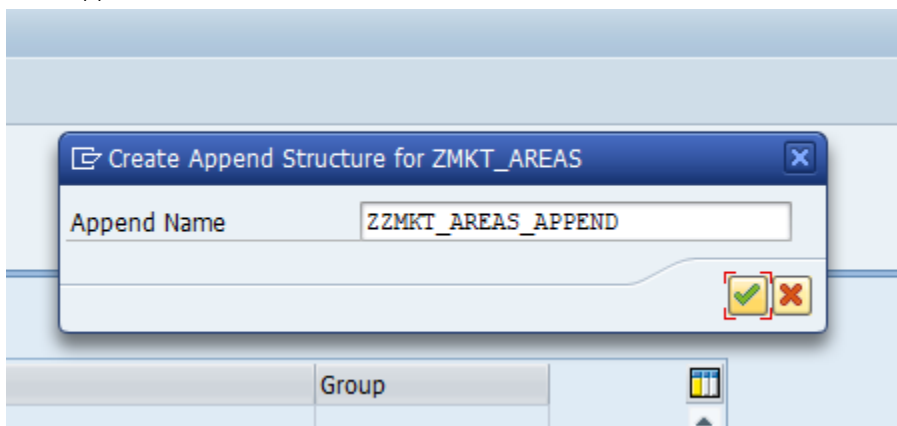
Lock object

Display Change Create

6. Choose *Goto->Append Structure*.



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*.

Structure

CRMT\_BUT\_MKTPERM\_DATA\_CHNG

Active

Short Description



Marketing Permission: Data for update (w/o partner/ contact)

Attributes


Components

Entry help/check

Currency/quantity fields

  Predefined Type

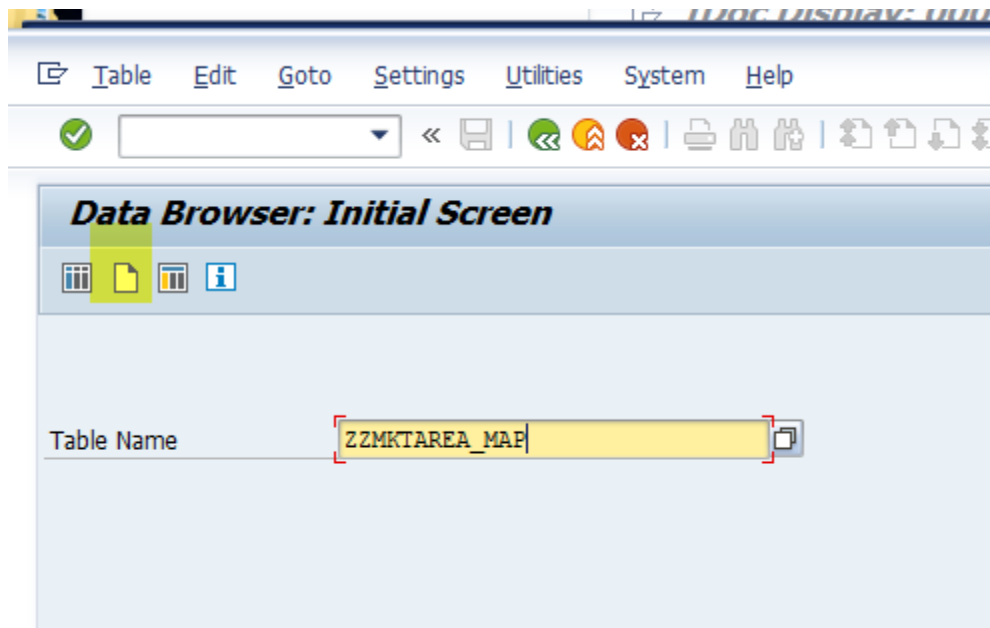
1 / 11

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

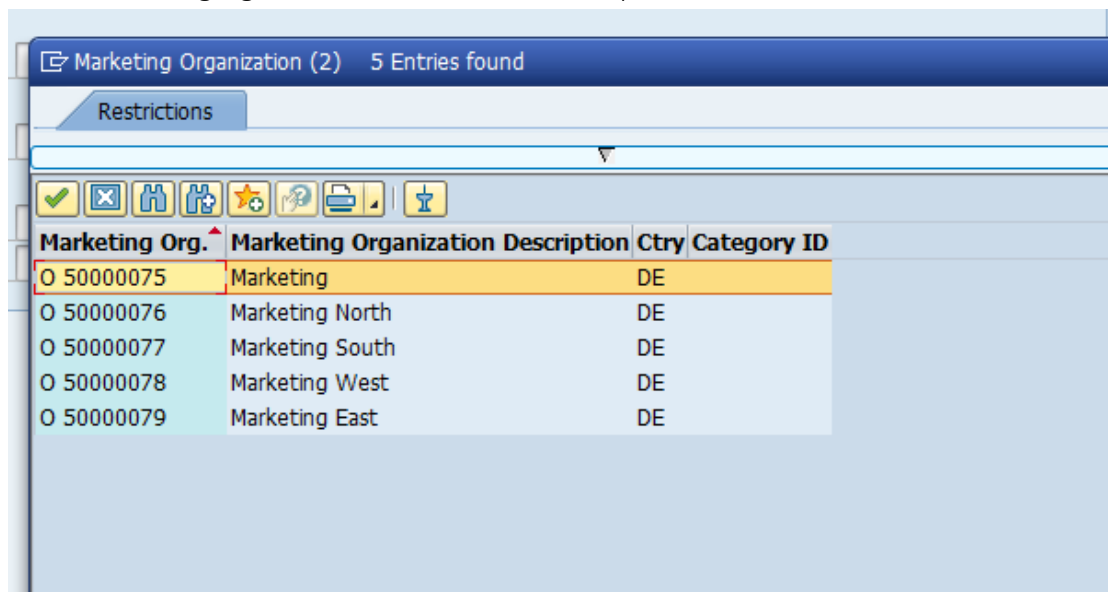
## 3.6 Maintain the Mapping Table Z or Y<MKTAREA\_MAP>

1. Start the transaction SE16.

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.

**Table ZZMKTAREA\_MAP Insert**

Reset

CLIENT 082

ZZMKT ORG O 50000182

ZZYMKT AREA GLOBAL\_MT\_AUD\_LEAF

5. Save your entries.

**Data Browser: Table ZZMKTAREA\_M**

CLIE...	ZZMKT_ORG	ZZYMKT_AR...
277	O 50000075	CXXGLOBAL
277	O 50000076	DK_AREA_A
277	O 50000077	DK_AREA_B
277	O 50000078	AM_AREA_A
277	O 50000079	AM_AREA_B

## 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.



**ALE Interfaces for Function Module**

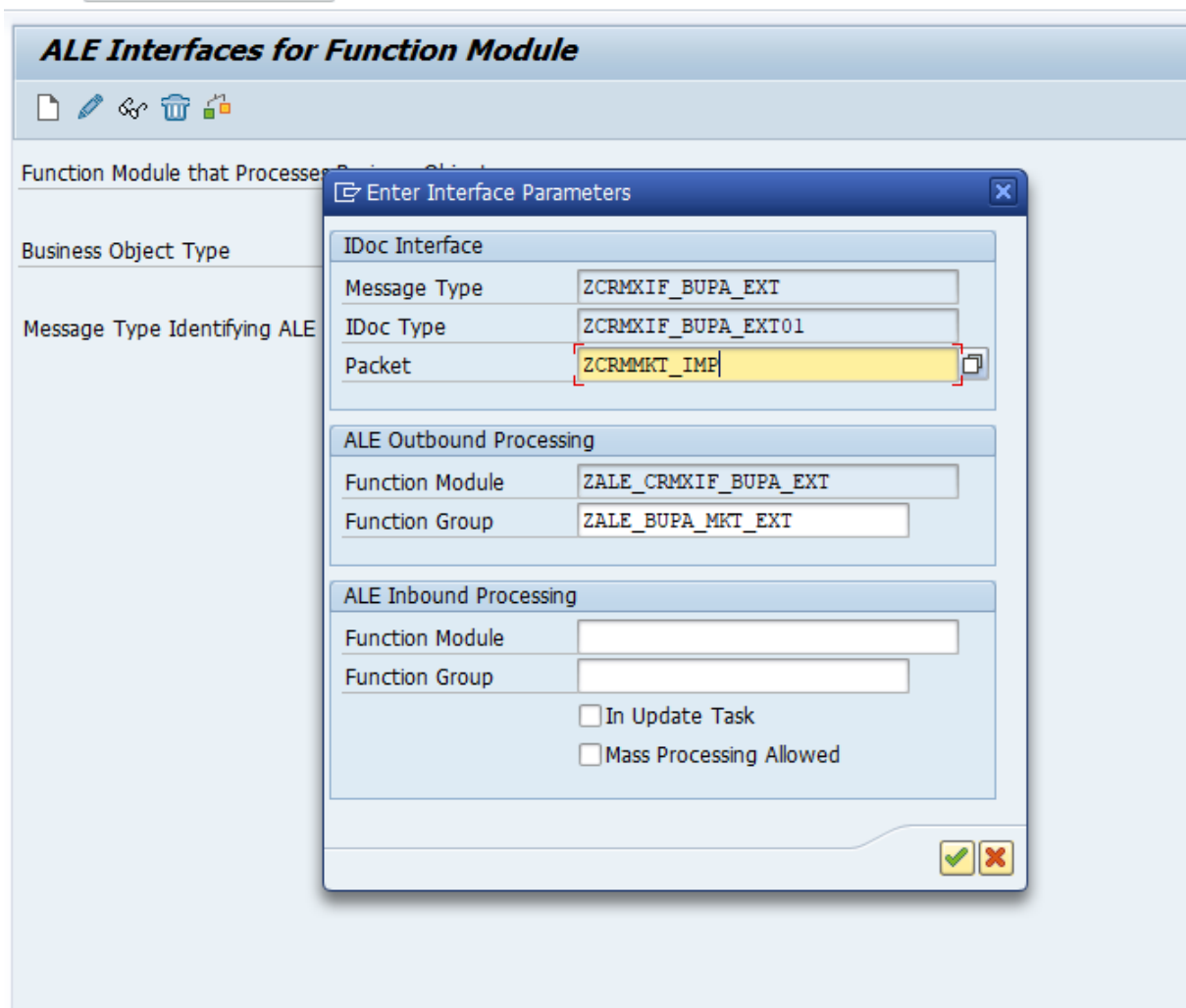
Function Module that Processes Business Object: CRMXIF\_PARTNER\_SAVE

Business Object Type: BUS1006

Message Type Identifying ALE Interface

Message Type
CRMXIF_PARTNER_SAVE
CRMXIF_PARTNER_SAVE_M
ZCRMXIF_BUPA_EXT
ZCRMXIF_PARTNER_SAVE
ZCRMXIF_PARTNER_SAVE_M01

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>



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.

**ALE Interfaces for Function Module**

Function Module that Processes Business Object

CRMIF\_PARTNER\_REL\_SAVE

Business Object Type

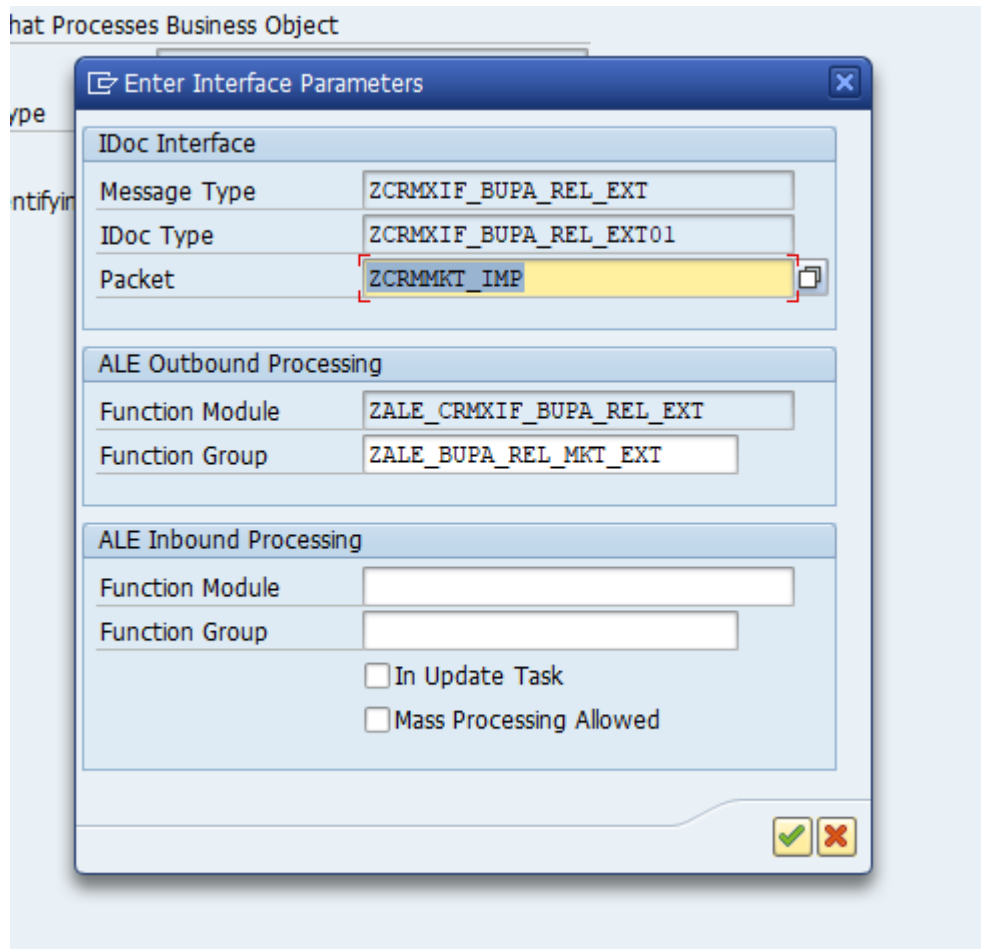
BUS1006080

Message Type Identifying ALE Interface

List of Message Types

Message Type
CRMIF_PARTNER_REL_SAVE
CRMIF_PARTNER_REL_SAVE_M
ZCRMIF_BUPA_REL_EXT

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<CRMIF\_BUPA\_REL\_EXT>  
 IDoc type: Z or Y<CRMIF\_BUPA\_REL\_EXT01>  
 Package: Z or Y<CRMMKT\_IMP> (see 2.2.1)  
 Function group: Z or Y<ALE\_BUPA\_REL\_MKT\_EXT>



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

Class/Interface: ZCL_CRM_BUPA_MKTPERM_COMMON   Implemented / Active							
Properties   Interfaces   Friends   Attributes   Methods   Events   Types   Aliases							
Filter							
Attribute	Level	Visibility	R...	Typing	Associated Type	Description	Initial value
GV_PARTNERNO	Instance Attribute	Protected	<input type="checkbox"/>	Type	BU_PARTNER	Business Partner Number	
GT_MKT_AREAS	Instance Attribute	Public	<input type="checkbox"/>	Type	ZMKT_AREAS_T	Table of Marketing Areas	
			<input type="checkbox"/>	Type			
			<input type="checkbox"/>	Type			

- Go to the tab *Methods* and add a new constructor:

CONSTRUCTOR: Instance Method, Public

Properties Interfaces Friends Attributes Methods Events Types Aliases				
Parameters Exceptions Sourcecode [Icons] Filter				
Method	Level	Visibility	M...	Description
CHECK_CONTACT_ALLOWED_	Instance Method	Public		
READ_VALID_PERMISSIONS_	Instance Method	Protected		
CONSTRUCTOR	Instance Method	Public		

- Select CONSTRUCTOR and click on *Parameters* button

Properties Interfaces Friends Attributes Methods Events Types Aliases				
Parameters Exceptions Sourcecode [Icons] Filter				
Goto Parameters				
Method	Level	Visibility	M...	Description
CHECK_CONTACT_ALLOWED_	Instance Method	Public		
READ_VALID_PERMISSIONS_	Instance Method	Protected		
CONSTRUCTOR	Instance Method	Public		

- Add three new parameters for the constructor.

IV\_PARTNERNO: Optional, BU\_PARTNER

IV\_PARTNERGUID: BU\_PARTNER\_GUID

IV\_CONTPGUID: BU\_PARTNER\_GUID

Parameters of Method CONSTRUCTOR						
Methods Exceptions Sourcecode Properties [Icons]						
Parameter	P...	O...	Typing Method	Associated Type	Default Value	Description
IV_PARTNERNO	<input type="checkbox"/>	<input checked="" type="checkbox"/>	Type	BU_PARTNER		Business Partner Number
IV_PARTNERGUID	<input type="checkbox"/>	<input type="checkbox"/>	Type	BU_PARTNER_GUID		Business Partner GUID
IV_CONTPGUID	<input type="checkbox"/>	<input type="checkbox"/>	Type	BU_PARTNER_GUID		Business Partner GUID
	<input type="checkbox"/>	<input type="checkbox"/>	Type			
	<input type="checkbox"/>	<input type="checkbox"/>	Type			

- 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,
```

```

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.

```

## 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*

```

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.

```



```

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.
*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>.

```

```

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.
  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.

*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-ZZMKT_AREA = ls_mkt_area-ZZMKT_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.
    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'.

```

#### 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\_zeit,  
      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.

```

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
  lv_org_guid = lv_org_guid
  lv_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.

```

*\*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.

```

IF It_valid_permissions IS INITIAL OR It_mkp_applicable IS INITIAL.
  CONTINUE.
ENDIF.
LOOP AT It_valid_permissions INTO Is_valid_permission.
  IF Is_valid_permission-VALUE IS INITIAL AND It_value IS INITIAL.
    CONTINUE.
  ENDIF.
  APPEND INITIAL LINE TO <mkt_permission_t> ASSIGNING <mkt_permission>.
  lv_index = sy-TABIX.
  *check against mobile phone
  IF Is_channel-DATA-CHANNEL EQ 'TEL'.
    IF Is_valid_permission-VALUE IS INITIAL.
      READ TABLE It_value INTO Is_value WITH KEY
        STD = 'X'.
      IF sy-subrc EQ 0.
        IF Is_value-R_3_USER = '2' OR Is_value-R_3_USER = '3'.
          <mkt_permission>-DATA-CHANNEL = 'TE2'.
        ENDIF.
      ENDIF.
    ELSE.
      READ TABLE Is_address-DATA-COMMUNICATION-PHONE-PHONE INTO Is_phone
        WITH KEY CONTACT-DATA-TEL_NO = Is_valid_permission-value.
      IF sy-subrc EQ 0.
        IF Is_phone-CONTACT-DATA-R_3_USER = '2' OR Is_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 = Is_channel-DATA-CHANNEL.
  ENDIF.
*
* simulate CPI conversion of time stamp; add initial time to timestamp valid from date is not in future
  CONVERT DATE Is_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 current UTC date is smaller / valid from date is in future => current UTC should be passed
  IF lv_date_valid_from > lv_date_utc.
    Is_valid_permission-valid_from = lv_date_utc.
  ENDIF.

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

<mkt_permission>-DATA-ORIGIN = Is_valid_permission-ORIGIN.
<mkt_permission>-DATA-KEY-RECORD_GUID = Is_valid_permission-RECORD_GUID.

IF Is_valid_permission-VALUE IS INITIAL.
  READ TABLE It_value INTO Is_value WITH KEY STD = 'X'.
  IF sy-subrc EQ 0.
    <mkt_permission>-DATA-VALUE = Is_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.

*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-ZZMKT_AREA = ls_mkt_area-ZZMKT_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.

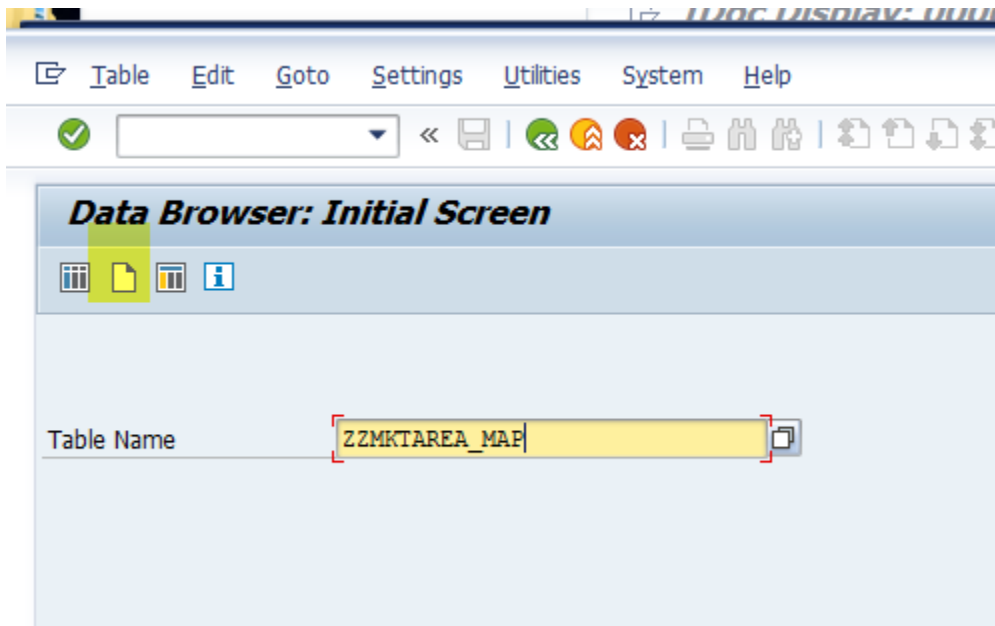
*move all redundant entries
* ----- marketing permissions -----
    ENDIF.
ENDLOOP.
endmethod.

```

## 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](#).





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

The screenshot shows the 'Marketing Organization (2) 5 Entries found' screen. A 'Restrictions' tab is active. Below the tab is a toolbar with various icons. The main area displays a table with the following data:

Marketing Org.	Marketing Organization Description	Ctry	Category ID
O 50000075	Marketing	DE	
O 50000076	Marketing North	DE	
O 50000077	Marketing South	DE	
O 50000078	Marketing West	DE	
O 50000079	Marketing East	DE	

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

**Table ZZMKTAREA\_MAP Insert**

Reset

CLIENT 082

ZZMKT ORG O 50000182

ZZYMKT AREA GLOBAL\_MT\_AUD\_LEAF

5. Save your entries.

**Data Browser: Table ZZMKTAREA\_M**

CLIE...	ZZMKT_ORG	ZZYMKT_AR...
277	O 50000075	CXXGLOBAL
277	O 50000076	DK_AREA_A
277	O 50000077	DK_AREA_B
277	O 50000078	AM_AREA_A
277	O 50000079	AM_AREA_B

### 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:

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:

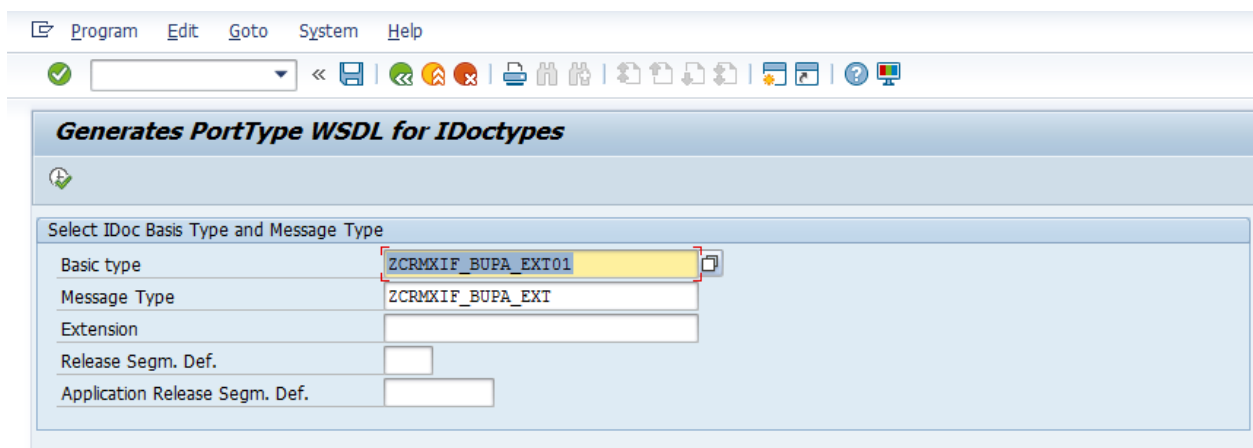
Create a new site according to the setup guide

## 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">
```

- 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>
```

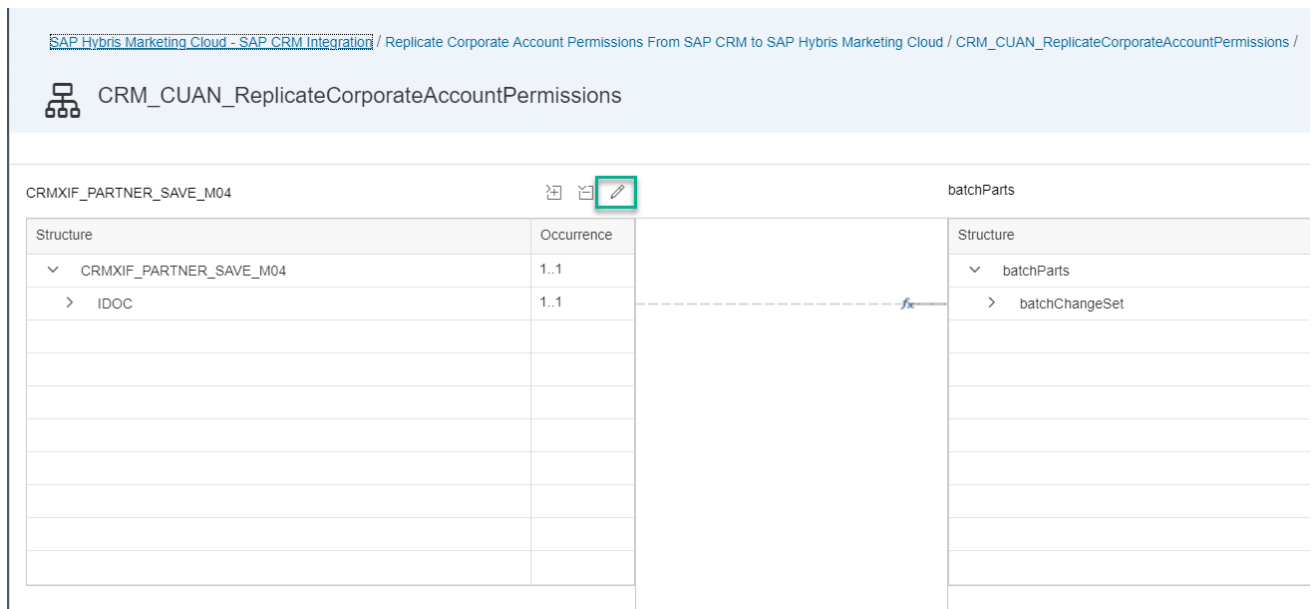
### 3. Adapt iFlow SAP Cloud Platform Integration (CPI)

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*.

The screenshot shows the SAP Cloud Platform Integration (CPI) Web UI. The top navigation bar includes 'SAP', 'SAP Cloud Platform Integration', and 'Data Services Integration'. The main header shows 'Design / SAP Hybris Marketing Cloud - SAP CRM Integration /' with an 'Edit' button highlighted. Below the header, the package name 'SAP Hybris Marketing Cloud - SAP CRM Integration' is displayed, along with a description: 'This integration content provides an out-of-the-box integration of SAP CRM with SAP Hybris Marketing Cloud.' and metadata: 'Vendor: SAP', 'Version: 2.5', 'Mode: Editable'. The 'Artifacts (12)' tab is selected, showing a table of integration flows. The table has columns for 'Name', 'Type', 'Version', and 'Actions'. The flow 'Replicate Contact Permissions From SAP CRM to SAP Hybris Marketing Cloud' is highlighted with a red box. Below the table, a note states: 'With this integration flow you can replicate Contact Permissions from SAP CRM to SAP Hybris Marketing Cloud Created'.

Name	Type	Version	Actions
CRM_ActivityReplication_Appointment_yMKT Replication of all the activities from SAP Hybris Marketing Cloud to SAP CRM Unmodified   Update Available	Integration Flow	1.0.3	[Link]
CRM_Business_Partner_Relationship_yMKT Business Partner Relationship Integration Unmodified   Update Available	Integration Flow	1.0.5	[Link]
CRM_Business_Partner_yMKT Business Partner Integration Unmodified   Update Available	Integration Flow	1.0.5	[Link]
CRM_Lead_Replication_yMKT Replication of leads from SAP Hybris Marketing Cloud to SAP CRM Unmodified   Update Available	Integration Flow	2.0.1	[Link]
CRM_OneOrder_Replication_Confirmation_yMKT Confirmation of replication of activities and leads from CRM to SAP Hybris Marketing Cloud. Unmodified   Update Available	Integration Flow	1.0.2	[Link]
Replicate Contact Permissions From SAP CRM to SAP Hybris Marketing Cloud With this integration flow you can replicate Contact Permissions from SAP CRM to SAP Hybris Marketing Cloud Created	Integration Flow	1.0.0	[Link]

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



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.

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

A photograph of dandelion seeds floating in the air against a clear blue sky. The seeds are in various stages of dispersal, with some fully blown and others still attached to their stems. The lighting is soft, suggesting a late afternoon or early morning setting.

[www.sap.com/contactsap](http://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](http://www.sap.com/corporate-en/legal/copyright/index.epx) for additional trademark information and notices.