

CDA

ZCL\_CRM\_TASK\_TRANSFER\_REP

Release 740

System

---

---

Class: ZCL\_CRM\_TASK\_TRANSFER\_REP  
Status: Active  
Inh. from: ZCL\_ES\_ALV\_OM

Attributes

Description: Task transfer ALV report  
Instantiation: Public  
Final  
Not released  
Fixed pt.arithmetic  
Category: General Object Type  
Package: ZPCABAP  
Original lang.: EN  
Created by: P000597  
Created on: 21.02.2015

Attributes  
Public attributes  
Attrib.            Cat            Description

Ref. Type  
Init. value

---

C_LOG_OBJECT	Const	Application log: Object name (Application code)
TYPE BALOBJ_D		
'ZLO_CRM_AUDIT_CASES'		

---

---

Private attribute			Ref. Type
Attrib.	Cat	Description	
Init. value			

---

ZCL_CRM_FROM_CASE	Inst	BOR case	TYPE REF TO
ZCL_CRM_BOR_CASE			
ZCL_CRM_TO_CASE	Inst	BOR case	TYPE REF TO
ZCL_CRM_BOR_CASE			
ZCL_CRM_BOL_CORE	Stat.	BOL Core	TYPE REF TO
CL_CRM_BOL_CORE			
DATA	Stat.		TYPE REF TO DATA

LOGGERStat. Message Log - HandlerTYPE REF TO

CL\_S\_AUT\_BAL\_LOG

Methods

Public methods

CLASS\_CONSTRUCTOR

Description: CLASS\_CONSTRUCTOR

Static method

METHOD class\_constructor.

bol\_core = cl\_crm\_bol\_core=>get\_instance( ).

bol\_core->start\_up( iv\_appl\_name = 'CRM\_CMG'

iv\_display\_mode\_support = abap\_true ).

logger = cl\_s\_aut\_bal\_log=>create( im\_object = c\_log\_object

).

ENDMETHOD.

09.03.201518:21:01

1

ZCL\_CRM\_TASK\_TRANSFER\_REPRelease 740System

CDA

SET\_CASES

Description: Sets the "from" and "to" cases.

Instance mthd

Importing parameter

P\_FROM TYPE RSOBJS\_NODE\_ID (From Case external ID)

P\_TO TYPE RSOBJS\_NODE\_ID (To Case external ID)

Exceptions

CX\_BAPI\_ERROR (BAPI Error Exception: Migrated Error Messages

Type "E")

method SET\_CASES.

DATA lx\_error TYPE REF TO cx\_bapi\_error.

from\_case = NEW zcl\_crm\_bor\_case( ).

to\_case = NEW zcl\_crm\_bor\_case( ).

TRY .

```

        from_case->set_instance_by_id( p_from ).
        to_case->set_instance_by_id( p_to ).
    CATCH cx_bapi_error INTO lx_error.
        RAISE EXCEPTION lx_error.
    ENDTRY.
endmethod.

Private methods
GET_DATA
Description: Fills in the data table.
Instance mthd

CHANGING parameter
    PT_DATA TYPE CRMT_CMG_RECORD_TAB    (CRM Case Management: Case
Record Entries)

Exceptions
    CX_BAPI_ERROR    (BAPI Error Exception: Migrated Error Messages
Type "E")
METHOD get_data.
    DATA: lo_case      TYPE REF TO cl_crm_bol_entity,
           lv_cguid     TYPE          crmt_genil_object_guid,
           lo_tasks     TYPE REF TO if_bol_bo_col,
           lo_task      TYPE REF TO if_bol_bo_property_access,
           lx_error     TYPE REF TO cx_crm_genil_model_error,
           lo_iterator  TYPE REF TO if_bol_bo_col_iterator,
           lrf_data     TYPE REF TO data,
           lt_return    TYPE          bapiret2_t,
           lw_return    TYPE          bapiret2.
    FIELD-SYMBOLS <lw_cmg_record> TYPE crmt_cmg_record.
    TRY.
        lv_cguid = me->from_case->av_objid.
        lo_case  = bol_core->get_root_entity( iv_object_name =
'CmgCase'
                                              iv_object_guid =
lv_cguid ).
        lo_tasks ?= lo_case->get_related_entities(
                                              iv_relation_name =
'CmgRecordRel' ).
        CATCH cx_crm_genil_model_error INTO lx_error.
            lw_return-id      = '00'.
            lw_return-type    = 'E'.
            lw_return-number  = 368.

```

---

09.03.2015	18:21:01
------------	----------

---

2

---

ZCL_CRM_TASK_TRANSFER_REP	Release 740	System
---------------------------	-------------	--------

---

CDA

---

```

        lw_return-message_v1 = lx_error->get_text( ).
        lw_return-message    = lx_error->get_text( ).
        APPEND lw_return TO lt_return.
        RAISE EXCEPTION TYPE cx_bapi_error
        EXPORTING
            status = lt_return.
    ENDTRY.
    lo_iterator = lo_tasks->get_iterator( ).
    lo_task = lo_tasks->get_first( ).
    CREATE DATA lrf_data TYPE crmt_cmig_record.
    ASSIGN lrf_data->* TO <lw_cmig_record>.
    WHILE lo_task IS BOUND.
        lo_task->get_properties( IMPORTING
                                es_attributes = <lw_cmig_record>
                                ).
        IF <lw_cmig_record>-bor_type = 'BUS2000125'.
            APPEND <lw_cmig_record> TO pt_data.
        ENDIF.
        lo_task = lo_iterator->get_next( ).
        CLEAR <lw_cmig_record>.
    ENDWHILE.
ENDMETHOD.

TRANSFER_TASKS
Description: Transfers tasks from the source to the target cases.
Instance mthd

METHOD transfer_tasks.
    DATA: lo_selections TYPE REF TO cl_salv_selections,
           lo_from_case  TYPE REF TO if_scmg_case_api,
           lo_to_case     TYPE REF TO if_scmg_case_api,
           lt_seletion    TYPE          salv_t_row,
           lw_row         TYPE          crmt_cmig_record,
           lt_sp_poid     TYPE          srm_list_poid,
           lw_sp_poid     TYPE          srm_poid,
           lv_guid_16     TYPE          crmt_object_guid,
           lv_guid_32     TYPE          scmig_case_guid,
           lt_return      TYPE          bapiret2_t,
           lt_messages    TYPE          scmig_t_attr_return_value,
           lv_fail        TYPE          abap_bool.
    FIELD-SYMBOLS: <li_index> TYPE          int4,
                   <lt_data>  TYPE STANDARD TABLE.
    lo_selections = alv->get_selections( ).
    lt_seletion = lo_selections->get_selected_rows( ).
    IF lt_seletion IS INITIAL.
        MESSAGE 'Please, select at least one task.'(M03) TYPE 'I'.
        RETURN.
    ENDIF.
    " Getting an API case instance for the From Case
    lv_guid_16 = from_case->av_objid.
    lv_guid_32 = cl_crm_isu_guid=>convert_16_to_32( lv_guid_16 ).
    cl_scmig_case_api=>if_scmig_case_api~open_case(
        EXPORTING
            im_case_guid      = lv_guid_32
            im_enqueue       = abap_true

```

09.03.2015

18:21:01

3

ZCL\_CRM\_TASK\_TRANSFER\_REP      Release 740      System  
CDA

---

```

        im_update_task = abap_true
    RECEIVING
        re_case         = lo_from_case
    EXCEPTIONS
        failed           = 1
        enqueue_failed   = 2
        invalid_guid     = 3
        cx_srm_gsp_back   = 4
        no_authority     = 5
        illegal_case_type = 6
        OTHERS           = 7      ).
    IF sy-subrc <> 0.
        m_add_syst_msg.
        MESSAGE 'It is not possible to change the source
case.' (M01)
            TYPE 'E'.
    ENDIF.
    " Getting an API case instance for the To Case
    lv_guid_16 = to_case->av_objid.
    lv_guid_32 = cl_crm_isu_guid=>convert_16_to_32( lv_guid_16 ).
    cl_scmg_case_api=>if_scmg_case_api~open_case(
        EXPORTING
            im_case_guid      = lv_guid_32
            im_enqueue        = abap_true
            im_update_task    = abap_true
        RECEIVING
            re_case           = lo_to_case
        EXCEPTIONS
            failed            = 1
            enqueue_failed    = 2
            invalid_guid      = 3
            cx_srm_gsp_back    = 4
            no_authority      = 5
            illegal_case_type  = 6
            OTHERS            = 7      ).
    IF sy-subrc <> 0.
        lo_from_case->dequeue( ).
        m_add_syst_msg.
        MESSAGE 'It is not possible to change the target
case.' (M02)
            TYPE 'E'.
    ENDIF.
    ASSIGN data->* TO <lt_data>.
    LOOP AT lt_seletion ASSIGNING <li_index>.
        READ TABLE <lt_data> INTO lw_row INDEX <li_index>.
        IF sy-subrc = 0.

```

```

CLEAR lt_sp_poid.
lw_sp_poid-id      = 'BOR_OBJECT_TYPE'.
lw_sp_poid-value = lw_row-bor_type.
APPEND lw_sp_poid TO lt_sp_poid.
lw_sp_poid-id      = 'BOR_OBJECT_ID'.
lw_sp_poid-value = lw_row-guid.
INSERT lw_sp_poid INTO TABLE lt_sp_poid.
" Removing tasks from source case
lo_from_case->element_delete(
    EXPORTING

```

---

	09.03.2015	18:21:01	
--	------------	----------	--

---

4			
---	--	--	--

---

	ZCL_CRM_TASK_TRANSFER_REP	Release 740	System
--	---------------------------	-------------	--------

---

CDA			
-----	--	--	--

---

```

    "im_anchor = 'CA'
    im_sp_poid = lt_sp_poid
    im_sps_id  = 'CRM_ACTIVITY_TASK'
    "im_id     = lw_row-model_id
EXCEPTIONS
    failed                = 1
    invalid_parameters    = 2
    OTHERS                 = 3          ).
IF sy-subrc <> 0.
    m_add_syst_msg.
    lv_fail = abap_true.
    EXIT.
ENDIF.
" Adding tasks to the Target Case
lo_to_case->element_insert(
    EXPORTING
        im_anchor = 'CA'
        im_sp_poid = lt_sp_poid
        im_sps_id  = 'CRM_ACTIVITY_TASK'
    EXCEPTIONS
        failed                = 1
        not_enqueued          = 2
        invalid_parameters    = 3
        OTHERS                 = 4          ).
IF sy-subrc <> 0.
    m_add_syst_msg.
    lv_fail = abap_true.
    EXIT.
ENDIF.
ENDIF.
ENDLOOP.
IF lv_fail = abap_false.

```

```

lo_from_case->save(
    IMPORTING
        ex_messages = lt_messages
    EXCEPTIONS
        failed      = 1
        OTHERS      = 2
    ).
IF sy-subrc <> 0.
    m_add_syst_msg.
ENDIF.
CLEAR lt_messages.
lo_to_case->save(
    IMPORTING
        ex_messages = lt_messages
    EXCEPTIONS
        failed      = 1
        OTHERS      = 2
    ).
IF sy-subrc <> 0.
    m_add_syst_msg.
ENDIF.
COMMIT WORK AND WAIT.
LOOP AT lt_selction ASSIGNING <li_index>.
    DELETE <lt data> INDEX <li_index>.

```

---

09.03.2015

18:21:01

5

```

CDA          ZCL_CRM_TASK_TRANSFER_REP          Release 740          System

```

```

        ENDLOOP.
ELSE.
    ROLLBACK WORK.
    logger->show( im_title = text-m02 ).
ENDIF.
lo_from_case->close_case( ).
lo_to_case->close_case( ).
alv->refresh( ).
ENDMETHOD.

```

## Redefined Methods

```
ON_USER_COMMAND
method ON_USER_COMMAND.
    IF e_salv_function = 'TRANSFER'.
        me->transfer_tasks( ).
    ENDIF.
    IF e_salv_function = 'HISTORY'.
        logger->show( im_title = 'Processing messages'(T02) ).
    ENDIF.
```

```

endmethod.

PROCESS_FUNCTIONS

METHOD process_functions.
  DATA: lo_functions TYPE REF TO CL_SALV_FUNCTIONS,
         ls_tooltip   TYPE          string.
  alv->set_screen_status( pfstatus = 'ZCRMR006_ST_GUI'
                        report    = me->repid
                        set_functions = alv->c_functions_all
  ).
ENDMETHOD.

PROCESS_TOP_OF_LIST

METHOD process_top_of_list.
  DATA: lo_grid    TYPE REF TO cl_salv_form_layout_grid,
         lo_grid_1  TYPE REF TO cl_salv_form_layout_grid,
         lo_flow    TYPE REF TO cl_salv_form_layout_flow,
         lo_label   TYPE REF TO cl_salv_form_label,
         lo_text    TYPE REF TO cl_salv_form_text,
         lv_value   TYPE          swc_value,
         ls_text    TYPE          string.
  lo_grid = NEW cl_salv_form_layout_grid( ).
  IF me->title IS NOT INITIAL.
    lo_grid->create_header_information(
      row      = 1
      column   = 1
      text     = me->title
      tooltip  = me->title ).
  ENDIF.
  *... in the cell [2,1] create a grid
  lo_grid_1 = lo_grid->create_grid(
    row      = 2
    column   = 1 ).

```

6

09.03.2015

18:21:01

CDA

ZCL\_CRM\_TASK\_TRANSFER\_REP

Release 740

System

```

*... in the cell [1,1] of the second grid create a label
  lo_flow = lo_grid_1->create_flow(
    row      = 2
    column   = 1 ).
  lo_label = lo_flow->create_label(
    text     = 'Source Case:'(t01)
    tooltip  = text-t01 ).

```



```

TRY .
    lv_value = from_case->get_plain_attribute( 'EXT_KEY' ).
    CATCH cx_bapi_error.
        "Ignore
ENDTRY.
SHIFT lv_value LEFT DELETING LEADING '0'.
lo_text = lo_flow->create_text(
    text      = lv_value
    tooltip   = lv_value ).
lo_label = lo_flow->create_label(
    text      = 'Target Case:'(t03)
    tooltip   = text-t03 ).
TRY .
    lv_value = to_case->get_plain_attribute( 'EXT_KEY' ).
    CATCH cx_bapi_error.
        "Ignore
ENDTRY.
SHIFT lv_value LEFT DELETING LEADING '0'.
lo_text = lo_flow->create_text(
    text      = lv_value
    tooltip   = lv_value ).
alv->set_top_of_list( lo_grid ).
ENDMETHOD.

```

## PUBLISH\_ALV

```

METHOD publish_alv.
    DATA: lx_error TYPE REF TO cx_bapi_error,
           lw_return TYPE          bapiret2.
    TRY .
        me->get_data( CHANGING pt_data = itab ).
        CATCH cx_bapi_error INTO lx_error.
        LOOP AT lx_error->status INTO lw_return.
            m_add_bapi_msg.
        ENDLOOP.
        READ TABLE lx_error->status INTO lw_return
            WITH KEY type = 'E'.
        IF sy-subrc = 0.
            MESSAGE
                ID lw_return-id
                TYPE lw_return-type
                NUMBER lw_return-number
                WITH lw_return-message_v1
                    lw_return-message_v2
                    lw_return-message_v3
                    lw_return-message_v4.
        ENDIF.
    ENDTRY.

```

09.03.2015

18:21:01

7

---

```

ENDTRY.
GET REFERENCE OF itab INTO DATA.
TRY.
    cl_salv_table=>factory(
        EXPORTING
            list_display = abap_false
        IMPORTING
            r_salv_table = alv
        CHANGING
            t_table      = itab ).
CATCH cx_salv_msg INTO alv_msg.
MESSAGE alv_msg TYPE 'I'.
EXIT.
ENDTRY.
me->process_functions( ).
me->set_columns( ).
me->process_layout( i_variant ).
me->register_events( ).
me->process_report_headers( ).
alv->display( ).
ENDMETHOD.

REGISTER_EVENTS

METHOD register_events.
    DATA: lo_events TYPE REF TO cl_salv_events_table.
    lo_events = alv->get_event( ).
    SET HANDLER me->on_user_command FOR lo_events.
ENDMETHOD.

SET_COLUMNS
method SET_COLUMNS.
    data lo_selections TYPE REF TO CL_SALV_SELECTIONS.
    CALL METHOD SUPER->SET_COLUMNS.
    lo_selections = alv->get_selections( ).
    lo_selections->set_selection_mode(

if_salv_c_selection_mode=>multiple ).
endmethod.

```

#### Local Types

```

*** use this source file for any type of declarations (class
*** definitions, interfaces or type declarations) you need for
*** components in the private section

```

#### Local class definitions

```

*** use this source file for the definition and implementation of
*** local helper classes, interface definitions and type
*** declarations

```

#### Macros

```

*** use this source file for any macro definitions you need

```

\*\*\* in the implementation part of the class

---

8	09.03.2015	18:21:01	
---	------------	----------	--

---

CDA	ZCL_CRM_TASK_TRANSFER_REP	Release 740	System
-----	---------------------------	-------------	--------

---



---

```

DEFINE m_add_bapi_msg.
  logger->add_entry( im_msgid   = lw_return-id
                    im_msgtype = lw_return-type
                    im_msgno   = lw_return-number
                    im_msgv1   = lw_return-message_v1
                    im_msgv2   = lw_return-message_v2
                    im_msgv3   = lw_return-message_v3
                    im_msgv4   = lw_return-message_v4 ).
END-OF-DEFINITION.
DEFINE m_add_syst_msg.
  logger->add_entry( im_msgid   = sy-msgid
                    im_msgtype = sy-msgty
                    im_msgno   = sy-msgno
                    im_msgv1   = sy-msgv1
                    im_msgv2   = sy-msgv2
                    im_msgv3   = sy-msgv3
                    im_msgv4   = sy-msgv4 ).
END-OF-DEFINITION.

```

<hr/>			
9	09.03.2015	18:21:01	
CDA	ZCL_CRM_TASK_TRANSFER_REP	Release 740	System
<hr/>			

	Overview
	Attributes
1	Attributes
1	Public attributes
1	Private attribute
	Methods
1	Public methods
1	CLASS_CONSTRUCTOR
1	SET_CASES
1	Private methods
2	GET_DATA
2	TRANSFER_TASKS
3	Redefined Methods
6	

6

6

6

7

8

8

8       Local Types

8

8       Local class definitions

8

8       Macros

8