

1.1 Function Module Z_CAT_LEVEL_DETERMINATION

Use Catalog, Codegroup and Code for getting Category Levels. In general I don't like going to the tables I have done here, but I had some pressure...

```
function z_cat_level_determination.  
* "-----  
* "-----  
* " * "Local Interface:  
* " IMPORTING  
* " REFERENCE (IV_KATALOGART) TYPE COMT_CATALOG  
* " REFERENCE (IV_CODEGRUPPE) TYPE QCODEGRP  
* " REFERENCE (IV_CODE) TYPE QCODE  
* " REFERENCE (IV_TIMESTAMP) TYPE CRMT_TIMESTAMP OPTIONAL  
* " EXPORTING  
* " REFERENCE (EV_LEVELS) TYPE ZCATLEVELS  
* "-----  
* "-----  
  
data: lv_guid1 type crm_erms_cat_guid,  
lv_guid2 type crm_erms_cat_guid,  
lv_guid3 type crm_erms_cat_guid,  
lv_asp_guid type crm_erms_cat_guid,  
lv_conckey type zcatconckey,  
lv_objguid type crm_erms_cat_guid.  
  
data: ls_buffer type zs_buffer.  
  
clear: ev_levels.  
lv_conckey-katalogart = iv_katalogart.  
lv_conckey-codegruppe = iv_codegruppe.  
lv_conckey-code = iv_code.  
  
read table gt_buffer into ls_buffer  
with key conckey = lv_conckey.  
if sy-subrc <> 0 or  
not iv_timestamp is initial.  
  
clear ls_buffer.  
ls_buffer-conckey = lv_conckey.  
if iv_timestamp is initial.  
select single b~cat_guid c~asp_guid into (lv_guid3, lv_asp_guid  
)  
from crmc_erms_cat_ok as a  
inner join crmc_erms_cat_ln as b  
on a~obj_guid = b~obj_guid  
inner join crmc_erms_cat_ca as c  
on b~cat_guid = c~cat_guid  
inner join crmc_erms_cat_as as d  
on c~asp_guid = d~asp_guid  
where a~obj_extkey = lv_conckey  
and a~obj_ddic = 'SUBJECTCODECRM'  
and b~lnk_type = 'IS_CODE'  
and b~obj_lang = space  
and d~asp_state = 'R'. "I know this is ugly but have to ge  
t the active schema.  
else.  
select single b~cat_guid c~asp_guid into (lv_guid3, lv_asp_guid  
)  
from crmc_erms_cat_ok as a
```

```

inner join crmc_erms_cat_ln as b
  on a~obj_guid = b~obj_guid
inner join crmc_erms_cat_ca as c
  on b~cat_guid = c~cat_guid
inner join crmc_erms_cat_as as d
  on c~asp_guid = d~asp_guid
where a~obj_extkey = lv_conckey
  and a~obj_ddic = 'SUBJECTCODECRM'
  and b~lnk_type = 'IS_CODE'
  and b~obj_lang = space
  and d~val_from <= iv_timestamp
  and d~val_to >= iv_timestamp
  and ( d~asp_state = 'R'
        or d~asp_state = 'U').      "I know this is ugly but have
to get the valid.
endif.
if sy-subrc = 0.
select single pare_guid into lv_guid2
from crmc_erms_cat_hi
where node_guid = lv_guid3
  and tree_type = 'CAT'
  and pare_guid <> lv_asp_guid
  and tree_guid = lv_asp_guid.      "make sure we get only active
tree structure.
if sy-subrc = 0.
select single pare_guid into lv_guid1
from crmc_erms_cat_hi
where node_guid = lv_guid2
  and tree_type = 'CAT'
  and pare_guid <> lv_asp_guid
  and tree_guid = lv_asp_guid.      "make sure we get only act
ive tree structure.
endif.

* Now determine how many levels we found.
if lv_guid1 is initial.
if lv_guid2 is initial.
*   level3 was highest level found -> becomes 1.
select single cat_id into ls_buffer-level1
from crmc_erms_cat_ca
where cat_guid = lv_guid3
  and asp_guid = lv_asp_guid.
else.
*   level2 -> becomes 1, level3 -> becomes 2.
select single cat_id into ls_buffer-level1
from crmc_erms_cat_ca
where cat_guid = lv_guid2
  and asp_guid = lv_asp_guid.
select single cat_id into ls_buffer-level2
from crmc_erms_cat_ca
where cat_guid = lv_guid3
  and asp_guid = lv_asp_guid.
endif.
else.
* All 3 levels found
select single cat_id into ls_buffer-level1
from crmc_erms_cat_ca
where cat_guid = lv_guid1
  and asp_guid = lv_asp_guid.
select single cat_id into ls_buffer-level2

```

```

        from crmc_erms_cat_ca
        where cat_guid = lv_guid2
              and asp_guid = lv_asp_guid.
    select single cat_id into ls_buffer-level3
        from crmc_erms_cat_ca
        where cat_guid = lv_guid3
              and asp_guid = lv_asp_guid.
    insert ls_buffer into table gt_buffer.
  endif.
else.
    "should not happen
  endif.
endif.
ev_levels-level1 = ls_buffer-level1.
ev_levels-level2 = ls_buffer-level2.
ev_levels-level3 = ls_buffer-level3.

```

```
endfunction.
```

1.2 Report ZCATLEVELS_UPDATE.

```

*&-----
--*
*& Report  ZCATLEVELS_UPDATE
*&
*&-----
--*
*& Description:
*& This Program populates the fields ZZLEVEL1, ZZLEVEL2 and ZZLEVEL3
on
*& CRMD_CUSTOMER_H based on the Category Hierachy derived from the
*& Catalog, Code and Codegroup on Subject Profile
*&
*& INTENDED USE: Only as an Initial Load program.
*& BADI ZCRM_SERVICE_OS_BADI takes care of Delta Updates
*&-----
--*

```

```
report zcatlevels_update.
```

```
***START ----- Data Declaration ----- ***
```

```
tables: crmd_orderadm_h.
```

```

data: all_guid      type table of crmt_object_guid,
      lt_guid       type crmt_object_guid_tab,
      lv_count      type num4,
      lv_levels     type zcatlevels.

```

```

data: lt_reqobj     type crmt_object_name_tab,
      ls_reqobj     type crmt_object_name,
      lt_subject    type crmt_subject_wrkt,
      lt_inputf     type crmt_input_field_tab,
      ls_inputf     type crmt_input_field,
      lt_field      type crmt_input_field_names_tab,
      ls_field      type crmt_input_field_names,
      lt_customer_h type crmt_customer_h_comt,
      ls_customer_h type crmt_customer_h_com,
      lt_exception  type crmt_exception_t,

```

```

ls_exception    type crmt_exception,
lt_save         type crmt_object_guid_tab.

field-symbols: <fs_guid>      type crmt_object_guid,
               <fs_subject>   type crmt_subject_wrk,
               <fs_inputf>    type crmt_input_field.

***END ----- Data Declaration ----- ***

***START ----- Selection Screen ----- ***

select-options: so_num      for crmd_orderadm_h-object_id,
               so_type     for crmd_orderadm_h-process_type.
parameters:    p_count     type num4 default 50.

***END ----- Selection Screen ----- ***

form dump_exception.

  loop at lt_exception into ls_exception.
    write: / 'Order:', ls_exception-guid, sy-vline,
           'Class:', ls_exception-class, sy-vline,
           'Code:', ls_exception-code.
  endloop.
  clear: lt_exception.
endform.                    "dump_exception

***START ----- form process_entries. ----- ***
form process_entries.

  call function 'CRM_ORDER_READ'
    exporting
      it_header_guid          = lt_guid
      it_requested_objects    = lt_reqobj
    importing
*      ET_SERVICE_OS          = ET_SERVICE_OS
      et_subject              = lt_subject
      et_exception            = lt_exception
  exceptions
    document_not_found       = 1
    error_occurred           = 2
    document_locked          = 3
    no_change_authority      = 4
    no_display_authority     = 5
    no_change_allowed        = 6
    others                    = 7.
  if sy-subrc <> 0.
    write: / '****ORDER_READ Errors****'.
    perform dump_exception.
  endif.

  clear: lt_customer_h, lt_inputf, lt_exception.

  loop at lt_subject assigning <fs_subject>
    where ref_kind = 'A'
      and profile_type = 'A'.

```

```
call function 'Z_CAT_LEVEL_DETERMINATION'
  exporting
    iv_katalogart = <fs_subject>-katalogart
    iv_codegruppe = <fs_subject>-codegruppe
    iv_code       = <fs_subject>-code
    iv_timestamp  = <fs_subject>-timestamp
  importing
    ev_levels     = lv_levels.

if not lv_levels is initial.
  clear: ls_customer_h.
  ls_customer_h-zzcatlevel1 = lv_levels-level1.
  ls_customer_h-zzcatlevel2 = lv_levels-level2.
  ls_customer_h-zzcatlevel3 = lv_levels-level3.
  ls_customer_h-ref_guid    = <fs_subject>-ref_guid.
  insert ls_customer_h into table lt_customer_h.

  ls_inputf-ref_guid = <fs_subject>-ref_guid.
  insert ls_inputf into table lt_inputf.
endif.
endloop.

call function 'CRM_ORDER_MAINTAIN'
  exporting
    it_customer_h = lt_customer_h
  importing
    et_exception  = lt_exception
  changing
    ct_input_fields = lt_inputf
  exceptions
    error_occurred = 1
    document_locked = 2
    no_change_allowed = 3
    no_authority = 4
    others = 5.
if sy-subrc <> 0.
  write: / '****ORDER_MAINTAIN Errors****'.
  perform dump_exception.
endif.

clear lt_save.
loop at lt_inputf assigning <fs_inputf>.
  insert <fs_inputf>-ref_guid into table lt_save.
endloop.

call function 'CRM_ORDER_SAVE'
  exporting
    it_objects_to_save = lt_save
  importing
    et_exception = lt_exception
  exceptions
    document_not_saved = 1
    others = 2.
if sy-subrc <> 0.
  write: / '****ORDER_SAVE Errors****'.
  perform dump_exception.
endif.

call function 'BAPI_TRANSACTION_COMMIT'
  exporting
    wait = 'X'.
```

```
endform.                                "process_entries

***END   ----- form process_entries. ----- ***

*****START-OF-
SELECTION*****
start-of-selection.

    clear: lv_count, lt_guid, lt_reqobj, lt_field.

***POPULATE GLOBAL DATA*****
**
ls_reqobj = 'SUBJECT'.
append ls_reqobj to lt_reqobj.
ls_field-fieldname = 'ZZCATLEVEL1'.
append ls_field to lt_field.
ls_field-fieldname = 'ZZCATLEVEL2'.
append ls_field to lt_field.
ls_field-fieldname = 'ZZCATLEVEL3'.
append ls_field to lt_field.

ls_inputf-ref_kind = 'A'.
ls_inputf-objectname = 'CUSTOMER_H'.
ls_inputf-field_names = lt_field.
***END POPULATE GLOBAL DATA*****
**

***Get all relevant transactions.
select guid into table all_guid
  from crmd_orderadm_h
  where object_type = 'BUS2000116'
        and process_type in so_type
        and object_id in so_num.

loop at all_guid assigning <fs_guid>.
  lv_count = lv_count + 1.
  insert <fs_guid> into table lt_guid.

  if lv_count >= p_count.
    perform process_entries.

    clear: lv_count, lt_guid.
  endif.
endloop.

if lv_count > 0.
  perform process_entries.
endif.
```

1.3 BADI - CRM_SERVICE_OS_BADI

method if_ex_crm_service_os_badi~crm_srv_osset_merge.

```

data: lt_guid      type crmt_object_guid_tab,
      ls_osset     type crmt_srv_osset_wrkl,
      lt_subject   type crmt_srv_subject_wrkt,
      lt_header    type crmt_orderadm_h_wrkt,
      lt_reqobj    type crmt_object_name_tab,
      ls_guid      type crmt_object_guid,
      ls_subject   type crmt_srv_subject_wrk,
      ls_header    type crmt_orderadm_h_wrk,
      ls_reqobj    type crmt_object_name.

data: lv_catlevels type zcatlevels,
      ls_customer_h type crmt_customer_h_com,
      ls_customer_wrk type crmt_customer_h_wrk,
      lv_not_exists type flag,
      lt_inputf      type crmt_input_field_names_tab,
      ls_inputf      type crmt_input_field_names.

call function 'CRM_ORDERADM_H_READ_OW'
  exporting
    iv_orderadm_h_guid = iu_srv_osset_wrk-guid_hd
  importing
    es_orderadm_h_wrk = ls_header
  exceptions
    admin_header_not_found = 1
    others                  = 2.
if sy-subrc = 0
  and ls_header-object_type = 'BUS2000116'
  and iu_srv_osset_wrk-ref_kind = 'A'.

loop at iu_srv_osset_wrk-osset into ls_osset
  where profile_type = 'A'.

  loop at ls_osset-subject into ls_subject.
    call function 'Z_CAT_LEVEL_DETERMINATION'
      exporting
        iv_katalogart = ls_subject-katalogart
        iv_codegruppe = ls_subject-codegruppe
        iv_code        = ls_subject-code
      importing
        ev_levels      = lv_catlevels.
    if not lv_catlevels is initial.
      call function 'CRM_CUSTOMER_H_READ_OW'
        exporting
          iv_guid      = ls_header-guid
        importing
          es_customer_h_wrk = ls_customer_wrk
        exceptions
          header_not_found = 1
          others           = 2.
      if sy-subrc ne 0.
        lv_not_exists = 'X'.
      else.
        lv_not_exists = space.
  endloop
endloop

```

```
endif.

ls_customer_h-zzcatlevel1 = lv_catlevels-level1.
ls_customer_h-zzcatlevel2 = lv_catlevels-level2.
ls_customer_h-zzcatlevel3 = lv_catlevels-level3.
ls_customer_h-ref_guid    = ls_header-guid.

ls_inputf-fieldname = 'ZZCATLEVEL1'.
insert ls_inputf into table lt_inputf.
ls_inputf-fieldname = 'ZZCATLEVEL2'.
insert ls_inputf into table lt_inputf.
ls_inputf-fieldname = 'ZZCATLEVEL3'.
insert ls_inputf into table lt_inputf.
* ls_inputf-fieldname = 'REF_GUID'.
* insert ls_inputf into table lt_inputf.

if lv_not_exists = 'X'.
  call function 'CRM_CUSTOMER_H_CREATE_OW'
    exporting
      iv_guid          = ls_header-guid
    exceptions
      error_occurred = 1
      others          = 2.
else.
  ls_customer_h-mode          = 'B'. "always change
  ls_inputf-fieldname = 'MODE'.
  insert ls_inputf into table lt_inputf.
endif.

call function 'CRM_CUSTOMER_H_CHANGE_OW'
  exporting
    is_customer_h_com = ls_customer_h
  changing
    ct_input_field_names = lt_inputf
  exceptions
    fill_error          = 1
    put_error           = 2
    publish_error       = 3
    others              = 4.
endif.
endloop.
endloop.
endif.

endmethod.
```