

```

class ZCL_SHLP_DEMO_HELPER definition
  public
  final
  create public .

public section.
  type-pools SHLP .

  types:
    gtt_shlp_descr TYPE STANDARD TABLE OF shlp_descr WITH DEFAULT KEY .

  methods ADD_FIELDS_TO_SHLP_INTERFACE
    changing
      !CH_INTERFACE type DDSHIFACES .
  methods ADD_FIELDS_TO_SHLP_FIELDDDESCR
    changing
      !CH_FIELDDDESCR type DDFIELDS .
  methods ADD_FIELDS_TO_SHLP_FIELDPROP
    changing
      !CH_FIELDPROP type DDSHFPROPS .
  methods FILL_CUSTOMER_FAVORITES_INFO
    importing
      !IN_SHLP_TAB type GTT_SHLP_DESCR
    changing
      !CH_RECORD_TAB type LXE_TT_SEAHLPRES
      !CH_SHLP type SHLP_DESCR
      !CH_CALLCONTROL type DDSHF4CTRL .
  PROTECTED SECTION.
private section.

  types:
    BEGIN OF gty_debia .
      INCLUDE TYPE m_debia.
      TYPES: f_movie TYPE zcustomer_faves-f_movie,
             f_sport TYPE zcustomer_faves-f_sport.
      TYPES: END OF gty_debia .
  types:
    BEGIN OF gty_debid.
      INCLUDE TYPE m_debid.
      TYPES: f_movie TYPE zcustomer_faves-f_movie,
             f_sport TYPE zcustomer_faves-f_sport.
      TYPES: END OF gty_debid .
  types:
    BEGIN OF gty_debie.
      INCLUDE TYPE m_debie.
      TYPES: f_movie TYPE zcustomer_faves-f_movie,
             f_sport TYPE zcustomer_faves-f_sport.
      TYPES: END OF gty_debie .
  types:
    BEGIN OF gty_debil.
      INCLUDE TYPE m_debil.
      TYPES: f_movie TYPE zcustomer_faves-f_movie,
             f_sport TYPE zcustomer_faves-f_sport.
      TYPES: END OF gty_debil .

  methods GET_DATA_CUST_FAVORITES
    changing

```

```
!CH_TABLE type ANY TABLE .
ENDCLASS.
```

```
CLASS ZCL_SHLP_DEMO_HELPER IMPLEMENTATION.
```

```
* <SIGNATURE>-----
-----+
* | Instance Public Method ZCL_SHLP_DEMO_HELPER-
>ADD_FIELDS_TO_SHLP_FIELDDDESCR
* +-----
-----+
* | [<-->] CH_FIELDDDESCR          TYPE          DDFIELDS
* +-----
-----</SIGNATURE>
```

```
METHOD add_fields_to_shlp_fielddescr.
```

```
DATA: lv_position TYPE dfies-position,
      lv_offset   TYPE dfies-offset,
      lv_leng     TYPE dfies-leng,
      lv_intlen   TYPE dfies-intlen,
      lt_dd031    TYPE SORTED TABLE OF dd031
                  WITH UNIQUE KEY tabname fieldname as4local as4vers
```

```
position.
```

```
*- Get the last field
```

```
DATA(lt_fdesc) = ch_fielddescr[].
SORT lt_fdesc BY position DESCENDING.
READ TABLE lt_fdesc INTO DATA(lwa_last) INDEX 1.
```

```
CHECK sy-subrc = 0.
```

```
SELECT * FROM dd031 INTO TABLE lt_dd031
        WHERE tabname = 'ZCUSTOMER_FAVES' AND
               as4local = 'A'.
```

```
lv_position = lwa_last-position.
lv_offset = lwa_last-offset + lwa_last-intlen.
```

```
DO 2 TIMES.
```

```
ADD 1 TO lv_position.
```

```
APPEND INITIAL LINE TO ch_fielddescr ASSIGNING FIELD-
SYMBOL(<fs_fielddescr>).
```

```
CASE sy-index.
```

```
WHEN 1. "Field F_MOVIE
READ TABLE lt_dd031 ASSIGNING FIELD-SYMBOL(<fs_dd031>)
        WITH KEY fieldname = 'F_MOVIE'.
```

```
CHECK sy-subrc = 0.
```

```
<fs_fielddescr>-fieldtext = 'Favorite Movie'(001).
<fs_fielddescr>-reptext = 'Favorite Movie'(001).
```

```

WHEN 2. "Field F_SPORT
  READ TABLE lt_dd031 ASSIGNING <fs_dd031>
    WITH KEY fieldname = 'F_SPORT'.

  CHECK sy-subrc = 0.

  <fs_fielddescr>-fieldtext = 'Favorite Sport'(201).
  <fs_fielddescr>-reptext = 'Favorite Sport'(201).

ENDCASE.

IF sy-index <> 1.
  lv_offset = lv_offset + lv_intlen.
ENDIF.

<fs_fielddescr>-fieldname = <fs_dd031>-fieldname.
lv_leng = <fs_dd031>-leng.
lv_intlen = <fs_dd031>-intlen.

<fs_fielddescr>-langu = sy-langu.
<fs_fielddescr>-position = lv_position.
<fs_fielddescr>-offset = lv_offset.
<fs_fielddescr>-leng = lv_leng.
<fs_fielddescr>-intlen = lv_intlen.
<fs_fielddescr>-outputlen = lv_leng.
<fs_fielddescr>-decimals = <fs_dd031>-decimals.
<fs_fielddescr>-datatype = <fs_dd031>-datatype.
<fs_fielddescr>-inttype = <fs_dd031>-inttype.
<fs_fielddescr>-scrlen1 = '10'.
<fs_fielddescr>-scrlen2 = '15'.
<fs_fielddescr>-scrlen3 = '20'.
<fs_fielddescr>-lowercase = 'X'.
ENDDO.

ENDMETHOD.

* <SIGNATURE>-----
* -----+
* | Instance Public Method ZCL_SHLP_DEMO_HELPER->ADD_FIELDS_TO_SHLP_FIELDPROP
* +-----+
* | [<-->] CH_FIELDPROP                TYPE                DDSHFPROPS
* +-----+
* -----</SIGNATURE>
METHOD add_fields_to_shlp_fieldprop.

  DATA(lt_fprop) = ch_fieldprop[].
  SORT lt_fprop BY shlplispos DESCENDING.
  READ TABLE lt_fprop INTO DATA(lwa_fprop) INDEX 1.

  CHECK sy-subrc = 0.

  DO 2 TIMES.
    ADD 1 TO lwa_fprop-shlplispos.

```

APPEND INITIAL LINE TO ch_fieldprop ASSIGNING FIELD-
SYMBOL(<fs_fieldprop>).

```
CASE sy-index.  
  WHEN 1. "Field F_MOVIE  
    <fs_fieldprop>-fieldname = 'F_MOVIE'.  
  WHEN 2. "Field F_SPORT  
    <fs_fieldprop>-fieldname = 'F_SPORT'.  
ENDCASE.
```

```
<fs_fieldprop>-shlpoutput = abap_true.  
<fs_fieldprop>-shlplispos = lwa_fprop-shlplispos.  
ENDDO.
```

ENDMETHOD.

```
* <SIGNATURE>-----  
-----+  
* | Instance Public Method ZCL_SHLP_DEMO_HELPER->ADD_FIELDS_TO_SHLP_INTERFACE  
* +-----  
-----+  
* | [<-->] CH_INTERFACE                TYPE                DDSHIFACES  
* +-----  
-----</SIGNATURE>  
METHOD add_fields_to_shlp_interface.
```

APPEND INITIAL LINE TO ch_interface ASSIGNING FIELD-
SYMBOL(<fs_interface>).

```
<fs_interface>-shlpfield = 'F_MOVIE'.  
APPEND INITIAL LINE TO ch_interface ASSIGNING <fs_interface>.  
<fs_interface>-shlpfield = 'F_SPORT'.
```

ENDMETHOD.

```
* <SIGNATURE>-----  
-----+  
* | Instance Public Method ZCL_SHLP_DEMO_HELPER->FILL_CUSTOMER_FAVORITES_INFO  
* +-----  
-----+  
* | [--->] IN_SHLP_TAB                TYPE                GTT_SHLP_DESCR  
* | [<-->] CH_RECORD_TAB              TYPE                LXE TT_SEAHLPRES  
* | [<-->] CH_SHLP                    TYPE                SHLP_DESCR  
* | [<-->] CH_CALLCONTROL              TYPE                DDSHF4CTRL  
* +-----  
-----</SIGNATURE>  
METHOD fill_customer_favorites_info.
```

```
DATA: lt_debia TYPE STANDARD TABLE OF gty_debia,  
      lt_debid TYPE STANDARD TABLE OF gty_debid,  
      lt_debie TYPE STANDARD TABLE OF gty_debie,  
      lt_debil TYPE STANDARD TABLE OF gty_debil.
```

```
DATA: lv_vkorg_in_shelp TYPE xfeld.
```

```
FIELD-SYMBOLS: <fs_results> TYPE STANDARD TABLE.
```

```

CASE ch_shlp-shlpname.
  WHEN 'DEBIA'.
    ASSIGN lt_debia TO <fs_results>.
  WHEN 'DEBID'.
    ASSIGN lt_debid TO <fs_results>.
  WHEN 'DEBIE'.
    ASSIGN lt_debie TO <fs_results>.
  WHEN 'DEBIL'.
    ASSIGN lt_debil TO <fs_results>.
ENDCASE.

```

```

CHECK <fs_results> IS ASSIGNED.

```

*- Get field KUNNR (Customer favorites are based on KUNNR)

```

CALL FUNCTION 'F4UT_PARAMETER_VALUE_GET'
  EXPORTING
    parameter      = 'KUNNR'
    fieldname      = 'KUNNR'
  TABLES
    shlp_tab       = in_shlp_tab
    record_tab     = ch_record_tab
    results_tab    = <fs_results>
  CHANGING
    shlp           = ch_shlp
    callcontrol    = ch_callcontrol
  EXCEPTIONS
    parameter_unknown = 1
    OTHERS          = 2.

```

*- Get the customer favorites

```

CALL METHOD me->get_data_cust_favorites
  CHANGING
    ch_table = <fs_results>.

```

*- Assign values back to RECORD_TAB

```

CALL FUNCTION 'F4UT_PARAMETER_RESULTS_PUT'
  EXPORTING
    parameter      = 'F_MOVIE'
    fieldname      = 'F_MOVIE'
  TABLES
    shlp_tab       = in_shlp_tab
    record_tab     = ch_record_tab
    source_tab     = <fs_results>
  CHANGING
    shlp           = ch_shlp
    callcontrol    = ch_callcontrol
  EXCEPTIONS
    parameter_unknown = 1
    OTHERS          = 2.

```

```

CALL FUNCTION 'F4UT_PARAMETER_RESULTS_PUT'
  EXPORTING
    parameter      = 'F_SPORT'
    fieldname      = 'F_SPORT'
  TABLES
    shlp_tab       = in_shlp_tab

```

```

        record_tab      = ch_record_tab
        source_tab      = <fs_results>
CHANGING
        shlp            = ch_shlp
        callcontrol     = ch_callcontrol
EXCEPTIONS
        parameter_unknown = 1
        OTHERS          = 2.

```

ENDMETHOD.

```

* <SIGNATURE>-----
-----+
* | Instance Private Method ZCL_SHLP_DEMO_HELPER->GET_DATA_CUST_FAVORITES
* +-----
-----+
* | [(<-->)] CH_TABLE                TYPE                ANY TABLE
* +-----
-----</SIGNATURE>

```

METHOD get_data_cust_favorites.

```

TYPES: BEGIN OF lty_lookup,
        kunnr TYPE knal-kunnr,
        END OF lty_lookup.

```

```

TYPES: BEGIN OF lty_input,
        kunnr TYPE zcustomer_faves-kunnr,
        f_movie TYPE zcustomer_faves-f_movie,
        f_sport TYPE zcustomer_faves-f_sport,
        END OF lty_input.

```

```

DATA: lt_lookup TYPE STANDARD TABLE OF lty_lookup,
      lt_input  TYPE SORTED TABLE OF lty_input WITH NON-UNIQUE KEY kunnr.

```

```

*- Build table of customers for lookup
  LOOP AT ch_table ASSIGNING FIELD-SYMBOL(<fs_result>).
    ASSIGN COMPONENT 'KUNNR' OF STRUCTURE <fs_result> TO FIELD-
SYMBOL(<fs_kunnr>).

```

```

        APPEND INITIAL LINE TO lt_lookup ASSIGNING FIELD-SYMBOL(<fs_lookup>).
        <fs_lookup>-kunnr = <fs_kunnr>.
    ENDLOOP.

```

```

CHECK lt_lookup[] IS NOT INITIAL.
SORT lt_lookup BY kunnr.
DELETE ADJACENT DUPLICATES FROM lt_lookup.

```

```

*- Get promo/mktg records
  SELECT kunnr f_movie f_sport FROM zcustomer_faves
        INTO TABLE lt_input
        FOR ALL ENTRIES IN lt_lookup
        WHERE kunnr = lt_lookup-kunnr.

```

```

CHECK lt_input[] IS NOT INITIAL.

```

```

*- Assign values to output structure

```

```
LOOP AT ch_table ASSIGNING <fs_result>.
  ASSIGN COMPONENT 'KUNNR' OF STRUCTURE <fs_result> TO <fs_kunnr>.

*- Get customer favorites and map to output structure
  READ TABLE lt_input ASSIGNING FIELD-SYMBOL(<fs_input>)
    WITH KEY kunnr = <fs_kunnr>.
  IF sy-subrc = 0.
    ASSIGN COMPONENT 'F_MOVIE' OF STRUCTURE <fs_result> TO FIELD-
SYMBOL(<fs_f_movie>).
    ASSIGN COMPONENT 'F_SPORT' OF STRUCTURE <fs_result> TO FIELD-
SYMBOL(<fs_f_sport>).
    <fs_f_movie> = <fs_input>-f_movie.
    <fs_f_sport> = <fs_input>-f_sport.
  ENDIF.
ENDLOOP.

ENDMETHOD.
ENDCLASS.
```