

FUNCTION ZEWM_ASRS_USAGE_DETAILS_KHPM.

```
*"-----
*"*Local Interface:
*"  IMPORTING
*"      REFERENCE(IV_LGNUM) TYPE  /SCWM/LGNUM
*"      REFERENCE(IV_VARIANT) TYPE  VARIANT
*"      REFERENCE(IV_MODE) TYPE  /SCWM/DE_MON_FM_MODE
*"  EXPORTING
*"      REFERENCE(EV_RETURNCODE) TYPE  XFELD
*"      REFERENCE(EV_VARIANT) TYPE  VARIANT
*"      REFERENCE(ET_DATA) TYPE  ZSCWM_TT_ASRS_UTIL
*"  CHANGING
*"      REFERENCE(CT_TAB_RANGE) TYPE  RSDS_TRANGE
*"      REFERENCE(CT_FIELDCAT) TYPE  LVC_T_FCAT
*"-----
```

```
*-----*
* Program Name      : ZEWM_ASRS_USAGE_DETAILS
* Create Date       : 05/01/2024
* Author            : Adam Siraj
* RICEF #           : Inb_BR_16 ASRS Usage
*-----*
* Description       : This FM is used to display ASRS usage details
*-----*
```

```
* MODIFICATION HISTORY *
*-----*
* Mod. No.  Date      Name                Transport Number      Change Desc. *
*-----*
* 1          7/12/2023 Himaja              KSDK901137              Maintained constant values, select stmt and
*-----*
```

```
CONSTANTS :lc_dynnr   TYPE sydynnr VALUE '0200',
            lc_confirm TYPE /scwm/de_tostat VALUE 'C',
*... Start of Addition by P00123461 (Himaja) ...* TR:KSDK901137
            prt1       TYPE /scwm/de_procty VALUE '2010',
            prt2       TYPE /scwm/de_procty VALUE '3000',
            prt3       TYPE /scwm/de_procty VALUE '3010',
            prt4       TYPE /scwm/de_procty VALUE '3090',
            prt5       TYPE /scwm/de_procty VALUE '3060',
            prt6       TYPE /scwm/de_procty VALUE '3093'.
*... End of Addition by P00123461 (Himaja) ...*
```

```
DATA: lo_mon_stock    TYPE REF TO /scwm/cl_mon_stock,
      lv_error        TYPE c,
      lr_procty       TYPE STANDARD TABLE OF rsdsselopt,
      lr_procty_outb  TYPE STANDARD TABLE OF rsdsselopt,
      ls_procty_outb  TYPE rsdsselopt,
      ls_procty       TYPE rsdsselopt,
      ls_date         TYPE rsdsselopt,
      ls_data         TYPE zscwm_s_asrs_util,
```

```
      lv_time_stamp_low TYPE /scwm/de_confirmed_dt,
      lv_time_stamp_high TYPE /scwm/de_confirmed_dt.
```

```
CLEAR :et_data[].
```

```
CALL SELECTION-SCREEN lc_dynnr STARTING AT 10 3
                                ENDING AT 98 19 .
```

```
IF s_date[] IS INITIAL.
    MESSAGE TEXT-009 TYPE 'E' DISPLAY LIKE 'I'.
RETURN.
```

```

ENDIF.

LOOP AT s_date .

    READ TABLE s_time INTO DATA(ls_time).

    IF s_date-option = 'EQ' AND s_date-high IS INITIAL.
        CONVERT DATE s_date-low TIME ls_time-low INTO TIME STAMP lv_time_stamp_low TIME ZONE sy-zonl
        s_date-high = s_date-low.           "change done by Adam for single date input
        CONVERT DATE s_date-high TIME ls_time-high INTO TIME STAMP lv_time_stamp_high TIME ZONE sy-z
    ENDIF.
    IF s_date-option = 'BT' .
        CONVERT DATE s_date-low TIME ls_time-low INTO TIME STAMP lv_time_stamp_low TIME ZONE sy-zonl
        CONVERT DATE s_date-high TIME ls_time-high INTO TIME STAMP lv_time_stamp_high TIME ZONE sy-z
    ENDIF.

ENDLOOP.

LOOP AT s_aisle INTO DATA(lr_aisle).
    SHIFT lr_aisle-low RIGHT DELETING TRAILING space.
    SHIFT lr_aisle-high RIGHT DELETING TRAILING space.
    MODIFY s_aisle FROM lr_aisle TRANSPORTING low high.
ENDLOOP.

SELECT aisle FROM /scwm/tmfaisles INTO TABLE @DATA(lt_aisles)
           WHERE lgnum = @iv_lgnum
           AND aisle IN @s_aisle.

IF sy-subrc EQ 0.
* LOOP AT lt_aisles INTO DATA(ls_ale).
    LOOP AT lt_aisles ASSIGNING FIELD-SYMBOL(<ls_ale>).      "TR:KSDK901137
        SHIFT <ls_ale>-aisle LEFT DELETING LEADING space.
*     MODIFY lt_aisles FROM ls_ale TRANSPORTING aisle.
    ENDLOOP.
    SORT lt_aisles BY aisle.
    DELETE ADJACENT DUPLICATES FROM lt_aisles COMPARING aisle.
ENDIF.

*Outbound
ls_procty-sign = wmegc_sign_inclusive.  "'I'.
ls_procty-option = wmegc_option_eq.  "'EQ'.
ls_procty-low = '2010'.
APPEND ls_procty TO lr_procty.

ls_procty-sign = wmegc_sign_inclusive.  "'I'.
ls_procty-option = wmegc_option_eq.  "'EQ'.
ls_procty-low = '2100'.
APPEND ls_procty TO lr_procty.

*Inbound and reorgination
ls_procty-sign = wmegc_sign_inclusive.  "'I'.
ls_procty-option = wmegc_option_eq.  "'EQ'.
ls_procty-low = '3090'.
APPEND ls_procty TO lr_procty.

*Daifuku addition by Adam
ls_procty-sign = wmegc_sign_inclusive.  "'I'.
ls_procty-option = wmegc_option_eq.  "'EQ'.
ls_procty-low = '3010'.
APPEND ls_procty TO lr_procty.

ls_procty-sign = wmegc_sign_inclusive.  "'I'.
ls_procty-option = wmegc_option_eq.  "'EQ'.
ls_procty-low = '3093'.

```

```
APPEND ls_procty TO lr_procty.  
*end of Daifuku addition
```

```
SELECT lgnum,  
       procty,  
       trart,  
       tostat,  
       prsrc,  
       vltyp,  
       vlpla,  
       nltyp,  
       nlpla,  
       confirmed_at  
FROM /scwm/ordim_c INTO TABLE @DATA(lt_ordim_c)  
WHERE lgnum EQ @iv_lgnum AND  
       procty IN @lr_procty AND  
       tostat EQ @lc_confirm AND  
       confirmed_at GE @lv_time_stamp_low AND  
       confirmed_at LE @lv_time_stamp_high.
```

```
IF sy-subrc EQ 0.  
    DELETE lt_ordim_c WHERE trart = '7'.  
    DELETE lt_ordim_c WHERE trart = '5'.  
ENDIF.
```

```
IF lt_ordim_c[] IS NOT INITIAL.
```

```
    SELECT lgpla, aisle FROM /scwm/lagp INTO TABLE @DATA(lt_bin)  
    FOR ALL ENTRIES IN @lt_ordim_c  
    WHERE lgnum = @iv_lgnum  
    AND lgpla = @lt_ordim_c-vlpla .
```

```
*... Start of Addition by P00123461 (Himaja) ...*   TR:KSDK901137  
    IF sy-subrc = 0 .  
        SORT lt_bin BY aisle lgpla.  
    ENDIF.
```

```
*... End of Addition by P00123461 (Himaja) ...*
```

```
    SELECT lgpla, aisle FROM /scwm/lagp INTO TABLE @DATA(lt_bin2)  
    FOR ALL ENTRIES IN @lt_ordim_c  
    WHERE lgnum = @iv_lgnum  
    AND lgpla = @lt_ordim_c-nlpla .
```

```
*... Start of Addition by P00123461(Himaja) ...*   TR:KSDK901137  
    IF sy-subrc = 0 .  
        SORT lt_bin2 BY aisle lgpla.  
    ENDIF.
```

```
*... End of Addition by P00123461(Himaja) ...*
```

```
APPEND LINES OF lt_bin2[] TO lt_bin[].
```

```
*    SORT lt_bin. "commented by Himaja due to lt_bin is a table with an empty primary key  
*    SORT lt_bin BY aisle lgpla.
```

```
DELETE ADJACENT DUPLICATES FROM lt_bin COMPARING aisle lgpla.
```

```
*    LOOP AT lt_ordim_c INTO DATA(ls_ordim_c).
```

```
LOOP AT lt_ordim_c ASSIGNING FIELD-SYMBOL(<ls_ordim_c>).
```

```
    CLEAR <ls_ordim_c>-prsrc.
```

```
*    IF ls_ordim_c-procty = '3090' OR ls_ordim_c-procty = '3060' OR ls_ordim_c-procty = '3010'
```

```
    IF <ls_ordim_c>-procty = prt4 OR <ls_ordim_c>-procty = prt5 OR <ls_ordim_c>-procty = prt3
```

```
        READ TABLE lt_bin INTO DATA(ls_bin) WITH KEY lgpla = <ls_ordim_c>-nlpla.
```

```
    IF sy-subrc = 0.
```

```
        SHIFT ls_bin-aisle LEFT DELETING LEADING space.
```

```
        <ls_ordim_c>-prsrc = ls_bin-aisle.
```

```
    ENDIF.
```

```
ELSEIF <ls_ordim_c>-procty = prt1. "'2010' .
```

```
    CLEAR ls_bin.
```

```

    READ TABLE lt_bin INTO ls_bin WITH KEY lgpla = <ls_ordin_c>-vlpla.
    IF sy-subrc = 0.
        SHIFT ls_bin-aisle LEFT DELETING LEADING space.
        <ls_ordin_c>-prsrc = ls_bin-aisle.
    ENDIF.
ENDIF.
*
    MODIFY lt_ordin_c FROM ls_ordin_c TRANSPORTING prsrc . "commented by Himaja
    CLEAR ls_bin.
ENDLOOP.
ENDIF.

TYPES: BEGIN OF ty_outbound,
    lgnum      TYPE /scwm/lgnum,
    procty     TYPE /scwm/de_procty,
    trart      TYPE /scwm/lvs_trart,
    tostat     TYPE /scwm/de_tostat,
    prsrc      TYPE /scwm/de_prsrc,
    vltyp      TYPE /scwm/ltap_vltyp,
    vlpla      TYPE /scwm/ltap_vlpla,
    nltyp      TYPE /scwm/ltap_nltyp,
    nlpla      TYPE /scwm/ltap_nlpla,
    confirmed_at TYPE /scwm/de_confirmed_dt,
END OF ty_outbound.

ls_procty_outb-sign = wmegc_sign_inclusive.  "'I'".
ls_procty_outb-option = wmegc_option_eq.  "'EQ'".
ls_procty_outb-low = '2010'.
APPEND ls_procty_outb TO lr_procty_outb.

ls_procty_outb-sign = wmegc_sign_inclusive.  "'I'".
ls_procty_outb-option = wmegc_option_eq.  "'EQ'".
ls_procty_outb-low = '2100'.
APPEND ls_procty_outb TO lr_procty_outb.

ls_procty_outb-sign = wmegc_sign_inclusive.  "'I'".
ls_procty_outb-option = wmegc_option_eq.  "'EQ'".
ls_procty_outb-low = '2011'.
APPEND ls_procty_outb TO lr_procty_outb.

DATA: lt_outbound TYPE TABLE OF ty_outbound.

SELECT lgnum,
    procty,
    trart,
    tostat,
    prsrc,
    vltyp,
    vlpla,
    nltyp,
    nlpla,
    confirmed_at
FROM /scwm/ordin_c INTO TABLE @lt_outbound
WHERE lgnum EQ @iv_lgnum AND
    procty IN @lr_procty_outb AND
    vltyp EQ '3000' AND
    tostat EQ @lc_confirm AND
    confirmed_at GE @lv_time_stamp_low AND
    confirmed_at LE @lv_time_stamp_high.

IF sy-subrc EQ 0.
    DELETE lt_ordin_c WHERE trart = '7'.
    DELETE lt_ordin_c WHERE trart = '5'.

```

```

ENDIF.

LOOP AT lt_aisles INTO DATA(ls_aisle).
    ls_data-aisle = ls_aisle-aisle.
*Outbound

*      LOOP AT lt_ordim_c INTO <ls_ordim_c> WHERE procty = prt1 "'2010'
*                                     AND prsrc = ls_aisle-aisle.
*          ls_data-no_out = ls_data-no_out + 1.
*      ENDLOOP.

LOOP AT lt_outbound INTO DATA(ls_outbound) WHERE prsrc CS ls_aisle-aisle.
    ls_data-no_out = ls_data-no_out + 1.
ENDLOOP.

*Inbound
LOOP AT lt_ordim_c INTO <ls_ordim_c> WHERE procty = prt4 "'3090'
                                AND prsrc = ls_aisle-aisle.
    IF <ls_ordim_c>-nltyp = '3000' AND <ls_ordim_c>-vltyp NE '3000'.
        ls_data-no_inb = ls_data-no_inb + 1.
    ELSEIF <ls_ordim_c>-nltyp = '3000' AND <ls_ordim_c>-vltyp = '3000'.
        ls_data-no_reorg = ls_data-no_reorg + 1.
    ENDIF.
ENDLOOP.

*Daifuku addition
LOOP AT lt_ordim_c INTO <ls_ordim_c> WHERE procty = prt3 "'3010'
                                AND prsrc = ls_aisle-aisle.
*      IF ls_ordim_c-nltyp = '3000' AND ls_ordim_c-vltyp NE '3000'.
*          ls_data-no_inb = ls_data-no_inb + 1.
*      ELSEIF ls_ordim_c-nltyp = '3000' AND ls_ordim_c-vltyp = '3000'.
*          ls_data-no_reorg = ls_data-no_reorg + 1.
*      ENDIF.
ENDLOOP.

LOOP AT lt_ordim_c INTO <ls_ordim_c> WHERE procty = prt6 "'3093'
                                AND prsrc = ls_aisle-aisle.
*      IF ls_ordim_c-nltyp = '3000' AND ls_ordim_c-vltyp NE '3000'.
*          ls_data-no_inb = ls_data-no_inb + 1.
*      ELSEIF ls_ordim_c-nltyp = '3000' AND ls_ordim_c-vltyp = '3000'.
*          ls_data-no_reorg = ls_data-no_reorg + 1.
*      ENDIF.
ENDLOOP.
IF ls_data-no_inb IS INITIAL.
    ls_data-no_inb = 0.
ENDIF.

IF ls_data-no_out IS INITIAL.
    ls_data-no_out = 0.
ENDIF.

IF ls_data-no_reorg IS INITIAL.
    ls_data-no_reorg = 0.
ENDIF.

ls_data-total = ls_data-no_inb + ls_data-no_out + ls_data-no_reorg.

APPEND ls_data TO et_data.
CLEAR ls_data.

ENDLOOP.

```

```

DATA : lv_result TYPE /scwm/ltap_vsolm.
CLEAR :lt_aisles[],lt_ordim_c[],lt_bin[],lt_bin2[].
IF et_data[] IS NOT INITIAL.
  LOOP AT et_data INTO ls_data.
    CLEAR lv_result.
    lv_result = ls_data-no_reorg / ( ls_data-no_inb + ls_data-no_out + ls_data-no_reorg ) .
    lv_result = ceil( lv_result * 100 ).
    MOVE lv_result TO ls_data-per_reorg.
*    ls_data-per_reorg = ls_data-per_reorg * 100 .
    MODIFY et_data FROM ls_data TRANSPORTING per_reorg.
  ENLOOP.
ELSE.
  MESSAGE TEXT-010 TYPE 'E' DISPLAY LIKE 'I'.
ENDIF.
ENDFUNCTION.

```