SAP **1316186 - SAP Best Practices Baseline Turkey: BADI for Depreciation**

Version 4 **Validity:** 15.07.2013 - active   **Language** English

---

Header Data

Released On   15.07.2013 14:03:54
Release Status Released for Customer
Component   XX-CSC-TR Turkey
Priority   Correction with low priority
Category   Legal change

Symptom

This note is only valid for SAP Best Practices Baseline packages - Based on SAP ECC 6.00 -
Release V1.603 , V1.605. and V1.607. No support will be given by SAP Türkiye if applied in
any other R/3 system.
This note contains information, source code corrections and transport requests to handle
depreciation key for best practise requirements in Turkey. Please apply manual steps given below.

Other Terms

Import process, ERP Baseline, Turkey

Reason and Prerequisites

Best practises

Solution

```
1. Create package ZTR_FI
- Call transaction SE80
- Select package and enter package name as ZTR_FI
- Enter the name of the package as Turkey: FI Enhancements
- Save package
2. Create Implementation ZTR_FAA_DC_CUSTOMER
- Call transaction SE19
- Enter implementation name as ZTR_FAA_DC_CUSTOMER
- Click Create button
- Enter Definition name as FAA_DC_CUSTOMER
- Enter short text as Turkey: Fixed assets base value calculation
- Save the implementation in the package ZTR_FI
- Select Interface Tab
- Double click the method DEFINE_CHANGEOVER_YR
- Copy the following source code:
METHOD if_ex_faa_dc_customer~define_changeover_yr.
  CONSTANTS: c_con_x(1)          TYPE c VALUE 'X'.
  FIELD-SYMBOLS: <ls_yearseg>   TYPE faa_ee_s_yearseg,
                 <ls_hlpseg>    TYPE faa_dc_s_hlpseg,
                 <ls_org>       TYPE faa_org_ca.
  DATA: ld_sum_apc              TYPE f,
        ld_nbv                  TYPE f,
        ld_useful_life          TYPE faa_dc_useful_life ,
        ld_end_life             TYPE faa_dc_start_yr ,
        ld_periods              TYPE faa_dc_periods    ,
        p_gjahr                 TYPE gjahr,
        p_bukrs                 TYPE bukrs,
        ls_t001                 TYPE t001.
  SELECT SINGLE * FROM  t001
        INTO ls_t001
        WHERE  bukrs  = io_handle->mo_assetdata->ms_basic-orgunit.
  CHECK sy-subrc = 0.
  IF ls_t001-land1 = 'TR'.
    CASE is_hlpseg-keyref->ms_settings-changeover_meth.
      WHEN c_con_x.
        LOOP AT io_handle->mts_hlpseg ASSIGNING <ls_hlpseg>
            WHERE keyref->ms_settings-changeover_meth CA c_con_x
              AND changeover_yr                    IS INITIAL
              AND area                             EQ is_hlpseg-area
              AND use_max_periods                  IS INITIAL.
          CHECK <ls_hlpseg>-changeover_yr IS INITIAL.
          p_bukrs = io_handle->mo_assetdata->ms_basic-orgunit.
          p_gjahr = 0.
          IF NOT ( <ls_hlpseg>-cap_date IS INITIAL ).
            CALL FUNCTION 'FI_PERIOD_DETERMINE'
              EXPORTING
                i_budat = <ls_hlpseg>-cap_date
                i_bukrs = p_bukrs
              IMPORTING
```

```abap
                    e_gjahr = p_gjahr.
            ELSE.
              p_gjahr = <ls_hlpseg>-fyear.
            ENDIF.
            ld_useful_life = <ls_hlpseg>-useful_life / <ls_hlpseg>-periods.
            ld_end_life    = p_gjahr + ld_useful_life - 1.
            ld_periods     = ld_useful_life * <ls_hlpseg>-periods.
            IF <ls_hlpseg>-useful_life > ld_periods.
              ld_end_life = ld_end_life + 1.
            ENDIF.
            IF ld_end_life <= <ls_hlpseg>-fyear.
              cd_changeover_yr = <ls_hlpseg>-fyear.
            ENDIF.
          ENDLOOP.
      ENDCASE.
    ENDIF.
  ENDMETHOD.
- Save and activate the method
- Select Interface Tab
- Double click the method SET_PARAMETER
- Copy the following source code:
METHOD if_ex_faa_dc_customer~set_parameter.
  FIELD-SYMBOLS: <ls_parameter>      TYPE faa_ee_s_parameter.
  FIELD-SYMBOLS: <ls_hlpseg>         TYPE faa_dc_s_hlpseg.
  DATA: lt_parameter                 TYPE faa_ee_t_parameter.
  DATA: lt_anlb                      TYPE TABLE OF anlb.
  DATA: lt_afapl                   TYPE t093c-afapl.
  DATA: lt_afasl                   TYPE anlb-afasl.
  DATA: lt_metdeg                   TYPE t090naz-metdeg.
  DATA: lt_metstu                   TYPE t090naz-metstu.
  DATA: lt_afproz                   TYPE t090ns-afproz.
  DATA: lt_degfak                   TYPE t090nd-degfak.
  DATA: lt_maxprz                   TYPE t090nd-maxprz.
  DATA: lt_msfak                   TYPE anlz-msfak.
  DATA: lt_ndjar                   TYPE anlb-ndjar.
  DATA: lt_zuper                   TYPE ants-zuper.
  DATA: lt_faktor                   TYPE p DECIMALS 9.
  DATA: ls_parameter                TYPE faa_ee_s_parameter.
  DATA: ld_start_prd                TYPE numc3.
  DATA: ld_useful_life             TYPE faa_dc_useful_life.
  DATA: ld_useful_life_twice        TYPE faa_dc_useful_life.
  DATA: ld_decision_indicator       TYPE faa_dc_useful_life.
  DATA: ld_period_from_to           TYPE faa_dc_period_to.
  DATA: ls_t001                     TYPE t001.
  SELECT SINGLE * FROM   t001
         INTO ls_t001
         WHERE  bukrs = io_handle->mo_assetdata->ms_basic-orgunit.
  CHECK sy-subrc = 0.
  IF ls_t001-land1 = 'TR'.
    lt_parameter[] = it_parameter[].
    LOOP AT lt_parameter ASSIGNING <ls_parameter>.

      LOOP AT io_handle->mts_hlpseg ASSIGNING <ls_hlpseg>
        WHERE fyear       = <ls_parameter>-fyear
        AND   area        = <ls_parameter>-area
        AND    amount_type = <ls_parameter>-amount_type
        AND   period_to    = <ls_parameter>-period_to.
      ENDLOOP.
      SELECT SINGLE afapl FROM t093c INTO lt_afapl
          WHERE bukrs = io_handle->mo_assetdata->ms_basic-orgunit.
      SELECT SINGLE afasl FROM anlb INTO lt_afasl
          WHERE bukrs = io_handle->mo_leadobj->md_leadobj
          AND    anln1 = io_handle->mo_assetdata->ms_basic-asset
          AND    anln2 = io_handle->mo_assetdata->ms_basic-subnumber
          AND    afabe = <ls_parameter>-area.
      SELECT SINGLE ndjar FROM anlb INTO lt_ndjar
          WHERE bukrs = io_handle->mo_leadobj->md_leadobj
          AND    anln1 = io_handle->mo_assetdata->ms_basic-asset
          AND    anln2 = io_handle->mo_assetdata->ms_basic-subnumber
          AND    afabe = <ls_parameter>-area.
      lt_metdeg = space.
      SELECT SINGLE metdeg FROM t090naz INTO lt_metdeg
        WHERE afapl  = lt_afapl
        AND    afasl  = lt_afasl
        AND    afatyp = 'N'
        AND    phase  = '1'.
      SELECT SINGLE metstu FROM t090naz INTO lt_metstu
        WHERE afapl  = lt_afapl
        AND    afasl  = lt_afasl
        AND    afatyp = 'S'
        AND    phase  = '1'.
      SELECT SINGLE afproz FROM t090ns INTO lt_afproz
        WHERE afapl  = lt_afapl
        AND    metstu = lt_metstu
        AND    zgjahr = '9999'
        AND    daujar = '000'
        AND    dauper = '011'.
      CHECK sy-subrc = 0.
      CHECK <ls_parameter>-amount_type = 'S'.
      SELECT SINGLE degfak FROM t090nd INTO lt_degfak
        WHERE afapl  = lt_afapl
        AND    metdeg = lt_metdeg.
      SELECT SINGLE maxprz FROM t090nd INTO lt_maxprz
        WHERE afapl   = lt_afapl
        AND    metdeg = lt_metdeg.
```

```abap
      IF sy-subrc <> 0.
        lt_degfak = 1.
        lt_maxprz = 100.
      ENDIF.
      lt_zuper = <ls_hlpseg>-cap_date+4(2).
      IF <ls_hlpseg>-start_yr = <ls_hlpseg>-fyear.
        SELECT SINGLE msfak FROM anlz INTO lt_msfak
         WHERE bukrs  = io_handle->mo_leadobj->md_leadobj
         AND   anln1  = io_handle->mo_assetdata->ms_basic-asset
         AND   anln2  = io_handle->mo_assetdata->ms_basic-subnumber.
        lt_afproz = ( 100 / lt_ndjar ) * lt_degfak.
        lt_faktor = '1'.
        IF lt_msfak = '2'.
          lt_faktor = '1.125'.
        ENDIF.
        IF lt_msfak = '3'.
          lt_faktor = '1.15'.
        ENDIF.
        lt_afproz = lt_afproz * lt_faktor.
        IF lt_afproz <> 100
         OR lt_ndjar = 2.
          IF lt_afproz > lt_maxprz.
            lt_afproz = lt_maxprz.
          ENDIF.
        ENDIF.
        <ls_parameter>-percent_stated = lt_afproz / 100 *
                    ( 11 - ( 12 - lt_zuper ) ) / 12.
        ld_period_from_to = 0.
        IF <ls_parameter>-period_to <> 12.
          ld_period_from_to = <ls_parameter>-period_to - <ls_hlpseg>-period_from + 1.
        ENDIF.
        <ls_parameter>-percent_factor = ld_period_from_to / 11.
      ELSE.
        <ls_parameter>-percent_stated = '0'.
      ENDIF.
      APPEND <ls_parameter> TO ct_parameter.
    ENDLOOP.
  ENDIF.
ENDMETHOD.
```
- Save and activate the method
- Activate the implementation
3. Create Implementation ZTR_FAA_EE_CUSTOMER
- Call transaction SE19
- Enter implementation name as ZTR_FAA_EE_CUSTOMER
- Click Create button
- Enter Definition name as FAA_EE_CUSTOMER
- Enter short text as Turkey: Fixed assed depreciation calculation
- Save the implementation in the package ZTR_FI
- Select Interface Tab
- Double click the method SET_PERCENT_AMOUNT
- Copy the following source code:
```abap
METHOD if_ex_faa_ee_customer~set_percent_amount.
  FIELD-SYMBOLS:  <ls_segment>    TYPE faa_ee_s_segment,
                  <ls_parameter> TYPE faa_ee_s_parameter,
                  <ls_rounding>  TYPE faa_ee_s_rounding,
                  <ls_calcdata>  TYPE faa_ee_s_calcdata.
  DATA: ld_amount        TYPE faa_ee_s_calcdata-amount,
        ld_amount_prev   TYPE faa_ee_s_calcdata-amount,
        gc_round_near    TYPE faa_ee_s_rounding-method,
        gc_decimals      TYPE faa_ee_decimals.
  IF isx_idata-country = 'TR'.
  ASSIGN: isx_cdata-s_segment  to <ls_segment>.
    ASSIGN: <ls_segment>-ref_parameter->* TO <ls_parameter>.
    CHECK:  <ls_parameter>  IS ASSIGNED.
    ASSIGN: <ls_segment>-ref_calcdata->*  TO <ls_calcdata>,
            <ls_segment>-ref_rounding->*  TO <ls_rounding>.
    CHECK:  <ls_calcdata>    IS ASSIGNED,
            <ls_rounding>    IS ASSIGNED.
    ld_amount  =    <ls_calcdata>-basevalue
                 * <ls_parameter>-percent_factor
                 * <ls_parameter>-percent_stated.
    ld_amount_prev  =   <ls_calcdata>-baseval_prev
                     * <ls_parameter>-percent_factor
                     * <ls_parameter>-percent_stated.
    gc_round_near = 0.
    gc_decimals   = 0.
    CALL METHOD cl_faa_ee=>_round_value_prev
      EXPORTING
        id_place     = <ls_rounding>-place
        id_decimals  = gc_decimals
        id_method    = gc_round_near
      CHANGING
        cd_value     = ld_amount
        cd_value_add = ld_amount_prev.
    <ls_calcdata>-amount        = ld_amount.
    <ls_calcdata>-amount_prev   = ld_amount_prev.
    <ls_calcdata>-percent       = <ls_parameter>-percent_stated.
    cs_calcdata = <ls_calcdata>.
  ENDIF.
ENDMETHOD.
```
- Save and activate the method
- Activate the implementation

Validity

Software Component	From Rel.	To Rel.	And Subsequent
SAP_APPL	600	600	☐
	602	602	☐
	603	603	☐
	604	604	☐
	605	605	☐
	606	606	☐
	617	617	☐