BAPI\_ACC\_DOCUMENT\_POST

|  |
| --- |
| method /ifl/if\_dx\_step~execute.   data: lr\_context  type ref to zcl\_fi\_six\_mrx\_dx\_context,         lt\_return   type table of bapiret2,         lv\_obj\_type type awtyp,         lv\_obj\_key  type awkey,         lv\_obj\_sys  type awsys.    lr\_context ?= ir\_context.  \* Terminalkonfiguration lesen   read table lr\_context->trxdata assigning field-symbol(<ls\_trxdata>) index 1.   if sy-subrc is not initial.     raise exception type zcx\_fi\_six\_mrx       exporting         textid = zcx\_fi\_six\_mrx=>no\_trx\_data.   endif.   data(ls\_terminal) = zcl\_fi\_six\_mrx\_util=>get\_terminal\_config( exporting iv\_terminal\_id              = <ls\_trxdata>-terminal\_id                                                                           iv\_additional\_merchant\_data = <ls\_trxdata>-additional\_merchant\_data ).  \* Buchungsdatum ermitteln   data(lv\_budat) = conv budat( sy-datum ).   read table lr\_context->parameter into data(ls\_parameter) with key fieldname = 'BUDAT'.   if sy-subrc is initial and ls\_parameter-value is not initial.     lv\_budat = conv #( ls\_parameter-value ).   endif.  \* Mappen der Daten   me->map\_header( exporting ir\_status         = ir\_status                             is\_terminal       = ls\_terminal                             is\_trxdata        = <ls\_trxdata>                             iv\_sumproc        = conv abap\_bool( boolc( lines( lr\_context->trxdata ) > 1 ) )                             iv\_budat          = lv\_budat                   importing es\_documentheader = data(ls\_documentheader)                             es\_customercpd    = data(ls\_customercpd)                             et\_headtext       = data(lt\_headtext) ).    me->map\_items( exporting is\_terminal          = ls\_terminal                            it\_trxdata           = lr\_context->trxdata                  importing et\_accountgl         = data(lt\_accountgl)                            et\_accountreceivable = data(lt\_accountreceivable)                            et\_accountpayable    = data(lt\_accountpayable)                            et\_accounttax        = data(lt\_accounttax)                            et\_currencyamount    = data(lt\_currencyamount)                            et\_extension1        = data(lt\_extension1)                  changing  ct\_headtext          = lt\_headtext ).  \* FI-Beleg buchen   call function 'BAPI\_ACC\_DOCUMENT\_POST'     exporting       documentheader    = ls\_documentheader       customercpd       = ls\_customercpd     importing       obj\_type          = lv\_obj\_type       obj\_key           = lv\_obj\_key       obj\_sys           = lv\_obj\_sys     tables       accountgl         = lt\_accountgl       accountreceivable = lt\_accountreceivable       accountpayable    = lt\_accountpayable       accounttax        = lt\_accounttax       currencyamount    = lt\_currencyamount       extension1        = lt\_extension1       return            = lt\_return.    data(lv\_error) = abap\_false.   loop at lt\_return assigning field-symbol(<ls\_return>).     message id <ls\_return>-id type <ls\_return>-type number <ls\_return>-number           with <ls\_return>-message\_v1 <ls\_return>-message\_v2 <ls\_return>-message\_v3 <ls\_return>-message\_v4           into sy-msgli.     ir\_log->add\_sy( ).      if <ls\_return>-type ca 'EA'.       lv\_error = abap\_true.     endif.   endloop.   if lv\_error = abap\_true.     raise exception type zcx\_fi\_six\_mrx.   endif.  \* Belegnummer in den Statuseintrag übernehmen   ir\_status->gs\_status-objtype = gc\_objtype\_bkpf.   ir\_status->gs\_status-objkey  = lv\_obj\_key+10(4) && lv\_obj\_key(10) && lv\_obj\_key+14(4).  \* Texte speichern   loop at lt\_headtext assigning field-symbol(<ls\_headtext>).     data(ls\_header) = value thead( tdobject = 'BELEG'                                    tdname   = conv #( ir\_status->gs\_status-objkey )                                    tdid     = <ls\_headtext>-tdid                                    tdspras  = sy-langu ).      call function 'SAVE\_TEXT'       exporting         header   = ls\_header       tables         lines    = <ls\_headtext>-lines       exceptions         id       = 1         language = 2         name     = 3         object   = 4         others   = 5.     if sy-subrc is not initial.       raise exception type zcx\_fi\_six\_mrx         message id sy-msgid type sy-msgty number sy-msgno with sy-msgv1 sy-msgv2 sy-msgv3 sy-msgv4.     endif.   endloop.    call function 'COMMIT\_TEXT'. endmethod.  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  method map\_header.   clear:  es\_documentheader,           es\_customercpd,           et\_headtext.  \* Land vom Buchungskreis ermitteln   select single land1 from t001 into @data(lv\_land)     where bukrs = @ir\_status->gs\_status-bukrs.   if sy-subrc is not initial.     lv\_land = 'CH'.   endif.  \* Benutzer   es\_documentheader-username = sy-uname.  \* Kopftext   if is\_trxdata-trx\_type\_id = gc\_trx\_type\_id\_refund.     es\_documentheader-header\_txt = 'Storno'(001) && ':'.   endif.   if iv\_sumproc = abap\_true.     if es\_documentheader-header\_txt is initial.       es\_documentheader-header\_txt = 'Sammelbuchung'(002).     else.       es\_documentheader-header\_txt = es\_documentheader-header\_txt && space && 'Sammelbuchung'(002).     endif.   endif.   if es\_documentheader-header\_txt is initial.     es\_documentheader-header\_txt = is\_terminal-bktxt.   else.     es\_documentheader-header\_txt = es\_documentheader-header\_txt && space && is\_terminal-bktxt.   endif.  \* Organisationsdaten   es\_documentheader-comp\_code = ir\_status->gs\_status-bukrs.  \* Datum   es\_documentheader-doc\_date   = is\_trxdata-trx\_date.   es\_documentheader-pstng\_date = iv\_budat.  \* Dokumentart   es\_documentheader-doc\_type = is\_terminal-blart.  \* Referenzen   if iv\_sumproc = abap\_false.     es\_documentheader-ref\_doc\_no = is\_trxdata-additional\_merchant\_data.   else.     es\_documentheader-ref\_doc\_no = is\_terminal-bezeichnung.   endif.  \* Kopftexte   append initial line to et\_headtext assigning field-symbol(<ls\_headtext>).   <ls\_headtext>-tdid = gc\_tdid\_paymeth\_trxdatetime.   append initial line to <ls\_headtext>-lines assigning field-symbol(<ls\_line>).   <ls\_line>-tdformat = '/'.   <ls\_line>-tdline   = |{ is\_trxdata-product } / { is\_trxdata-trx\_date country = lv\_land } / { is\_trxdata-trx\_time country = lv\_land }|.    append initial line to et\_headtext assigning <ls\_headtext>.   <ls\_headtext>-tdid = gc\_tdid\_refnumber.   append initial line to <ls\_headtext>-lines assigning <ls\_line>.   <ls\_line>-tdformat = '/'.   if iv\_sumproc = abap\_false.     <ls\_line>-tdline   = |{ is\_trxdata-auth\_ref\_no } / { is\_trxdata-additional\_merchant\_data }|.   else.     <ls\_line>-tdline   = |{ is\_trxdata-auth\_ref\_no } / { is\_terminal-bezeichnung }|.   endif.    append initial line to et\_headtext assigning <ls\_headtext>.   <ls\_headtext>-tdid = gc\_tdid\_trxtype.   append initial line to <ls\_headtext>-lines assigning <ls\_line>.   <ls\_line>-tdformat = '/'.   if iv\_sumproc = abap\_false.     <ls\_line>-tdline   = |{ is\_trxdata-trx\_type }|.   else.     <ls\_line>-tdline   = 'Summenverarbeitung'(003).   endif.  \* CPD-Kunde   if is\_terminal-kunnr is not initial.     select single xcpdk, name1, ort01, land1 from kna1 into @data(ls\_kna1)       where kunnr = @is\_terminal-kunnr.     if ls\_kna1-xcpdk = abap\_true.       es\_customercpd-name = ls\_kna1-name1.       es\_customercpd-city = ls\_kna1-ort01.       es\_customercpd-country = ls\_kna1-land1.     endif.   endif. endmethod. |
| METHOD do\_post. \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*     DATA: lt\_return     TYPE TABLE OF bapiret2,           lv\_bapi\_error TYPE abap\_bool,           ls\_post       TYPE ty\_s\_post,           lv\_obj\_type   TYPE bapiache09-obj\_type,           lv\_obj\_key    TYPE bapiache09-obj\_key,           lv\_objid      TYPE bapiache09-obj\_key,           lv\_obj\_sys    TYPE bapiache09-obj\_sys. \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*     CLEAR: gt\_return,gv\_belnr.     " Allgemeine Daten abfüllen     CALL FUNCTION 'OWN\_LOGICAL\_SYSTEM\_GET'       IMPORTING         own\_logical\_system             = ls\_post-gs\_bapi\_header-obj\_sys       EXCEPTIONS         own\_logical\_system\_not\_defined = 1         OTHERS                         = 2.     "     ls\_post-gs\_bapi\_header-bus\_act   = 'RFBU'.     ls\_post-gs\_bapi\_header-obj\_key   = ''.     ls\_post-gs\_bapi\_header-obj\_type  = 'BKPFF'.     ls\_post-gs\_bapi\_header-username  = sy-uname. \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*     ls\_post-gs\_bapi\_header-header\_txt = is\_header-bktxt.     ls\_post-gs\_bapi\_header-comp\_code  = cv\_bukrs.     ls\_post-gs\_bapi\_header-doc\_date   = is\_header-bldat.     ls\_post-gs\_bapi\_header-pstng\_date = is\_header-budat.     ls\_post-gs\_bapi\_header-doc\_type   = is\_header-blart.     ls\_post-gs\_bapi\_header-fis\_period = is\_header-monat.     ls\_post-gs\_bapi\_header-ref\_doc\_no = is\_header-xblnr. \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*     "Buchungs Positionen erzeugen:     do\_adapt\_post\_pos( EXPORTING is\_header = is\_header                                  it\_data   = it\_data                         CHANGING cs\_post   = ls\_post ). \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*     " BAPI Verarbeitung     CALL FUNCTION 'BAPI\_ACC\_DOCUMENT\_CHECK' "Prüfen       EXPORTING         documentheader = ls\_post-gs\_bapi\_header       TABLES         accountgl      = ls\_post-gt\_bapi\_sachkonto         currencyamount = ls\_post-gt\_bapi\_wp         accounttax     = ls\_post-gt\_bapi\_tax         return         = lt\_return.      gt\_return = lt\_return.      LOOP AT lt\_return TRANSPORTING NO FIELDS WHERE type CA 'EA'."error or abort       RAISE post\_error.     ENDLOOP. \*     IF iv\_testlauf EQ abap\_true.       EXIT."Testlauf ende     ENDIF. \* \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* \*\* Echt Buchung     CLEAR: lt\_return,gt\_return.     CALL FUNCTION 'BAPI\_ACC\_DOCUMENT\_POST' "Buchen       EXPORTING         documentheader = ls\_post-gs\_bapi\_header       IMPORTING         obj\_type       = lv\_obj\_type         obj\_key        = lv\_obj\_key         obj\_sys        = lv\_obj\_sys       TABLES         accountgl      = ls\_post-gt\_bapi\_sachkonto         currencyamount = ls\_post-gt\_bapi\_wp         accounttax     = ls\_post-gt\_bapi\_tax         return         = lt\_return. \*\*     LOOP AT lt\_return TRANSPORTING NO FIELDS WHERE type CA 'EA'."error or abort       lv\_bapi\_error = abap\_true.       EXIT.     ENDLOOP.     gt\_return = lt\_return.     IF lv\_bapi\_error EQ abap\_true."Fehler       CALL FUNCTION 'BAPI\_TRANSACTION\_ROLLBACK'.       RAISE post\_error.     ELSE.       CALL FUNCTION 'BAPI\_TRANSACTION\_COMMIT' EXPORTING wait = 'X'.       gv\_belnr = ev\_belnr = lv\_obj\_key(10).       "Archivieren       IF iv\_xstring IS NOT INITIAL.         lv\_objid = lv\_obj\_key+10(4) && lv\_obj\_key(10) && lv\_obj\_key+14(4).         do\_add\_archive\_document( iv\_objid   = lv\_objid                                  iv\_xstring = iv\_xstring ).         do\_add\_archive\_document( EXPORTING iv\_objid   = lv\_objid                                            iv\_xstring = iv\_xstring                                  EXCEPTIONS                                    error      = 1                                    OTHERS     = 2 ).         IF sy-subrc <> 0.           MESSAGE 'Fehler: Dokument Anhang an Beleg fehlgeschlagen' TYPE 'S' DISPLAY LIKE 'E'.         ENDIF.       ENDIF.      ENDIF.    ENDMETHOD. |
| METHOD do\_post\_accrual. \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* \* VERBUCHUNG: Abgrenzungsbuchung                         \* \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* \* Lokale Datendefinition                   \* \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*   DATA: ls\_header      TYPE bapiache09,         lt\_gl          TYPE STANDARD TABLE OF bapiacgl09,         ls\_gl          LIKE LINE OF lt\_gl,         lt\_curr        TYPE STANDARD TABLE OF bapiaccr09,         ls\_curr        LIKE LINE OF lt\_curr,         lt\_belnr       TYPE STANDARD TABLE OF rf048\_d,         ls\_ktopl\_saknr TYPE zfigen\_s\_ktopl\_saknr,         ls\_return      TYPE bapiret2.    DATA: lv\_mwskz\_skb1      TYPE mwskz,         lv\_zuonr           TYPE acpi\_zuonr,         lv\_amount\_external TYPE bapicurr\_d,         lv\_with\_asset      TYPE abap\_bool.  \* \*\*\* Daten für den Anlageprozess   DATA: ls\_extension2        TYPE bapiparex,         lt\_extension2        TYPE STANDARD TABLE OF bapiparex,         ls\_extension2\_detail TYPE zfigen\_s\_asset\_acc\_doc\_ext2,         lt\_asset\_post\_memory TYPE zfigen\_t\_asset\_acc\_doc\_post,         ls\_asset\_post\_memory TYPE zfigen\_s\_asset\_acc\_doc\_post.   \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* \* Coding                                   \* \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  \*==========================================\* \* Export                                   \* \*==========================================\*   CLEAR: rv\_ok,          ev\_obj\_type,          ev\_obj\_key,          ev\_obj\_sys,          es\_belnr.   REFRESH: et\_return.  \*==========================================\* \* Periodendaten                            \* \*==========================================\*   TRY.       DATA(ls\_period) = zfiacccl\_services=>get\_period\_data(         EXPORTING           iv\_gjahr              = is\_head-gjahr           iv\_quarter            = is\_head-quarter ).     CATCH zcx\_bcgen.       CLEAR: ls\_period.   ENDTRY.    IF ls\_period IS INITIAL.     CLEAR: rv\_ok,            ls\_return.     zbcgencl\_services=>do\_compose\_message(       EXPORTING         iv\_type        = 'E'         iv\_id          = 'ZFI\_ACC'         iv\_number      = '029'       IMPORTING         es\_bapiret2    = ls\_return ).     INSERT ls\_return INTO TABLE et\_return.     RETURN.   ENDIF.  \*==========================================\* \* Gegenkonto                               \* \*==========================================\*   CLEAR: ls\_ktopl\_saknr.   CASE is\_head-acc\_type.     WHEN zfiacccl\_services=>mc\_accrual\_type-ta. \*     \*\*\* Transitorische Aktive       TRY.           ls\_ktopl\_saknr = zfiacccl\_parameter=>get\_gl\_ta(              EXPORTING                iv\_bukrs          = is\_head-bukrs                iv\_reference\_date = ls\_period-accrual\_date ).         CATCH zcx\_bcgen.           CLEAR: ls\_ktopl\_saknr.       ENDTRY.     WHEN zfiacccl\_services=>mc\_accrual\_type-tp. \*     \*\*\* Transitorische Passiven       TRY.           ls\_ktopl\_saknr = zfiacccl\_parameter=>get\_gl\_tp(              EXPORTING                iv\_bukrs          = is\_head-bukrs                iv\_reference\_date = ls\_period-accrual\_date ).         CATCH zcx\_bcgen.           CLEAR: ls\_ktopl\_saknr.       ENDTRY.     WHEN OTHERS.       CLEAR: ls\_ktopl\_saknr.   ENDCASE.    IF ls\_ktopl\_saknr IS INITIAL.     CLEAR: rv\_ok,            ls\_return.     zbcgencl\_services=>do\_compose\_message(       EXPORTING         iv\_type        = 'E'         iv\_id          = 'ZFI\_ACC'         iv\_number      = '030'       IMPORTING         es\_bapiret2    = ls\_return ).     INSERT ls\_return INTO TABLE et\_return.     RETURN.   ENDIF.  \*==========================================\* \* Departement und Text für die Zuordnung   \* \*==========================================\*   CLEAR: lv\_zuonr.   IF NOT is\_head-prctr IS INITIAL.     zfiacccl\_services=>get\_departement(       EXPORTING         iv\_prctr          = is\_head-prctr         iv\_reference\_date = ls\_period-accrual\_date         iv\_langu          = 'D'                       "Immer Deutsch       IMPORTING         ev\_short          = DATA(lv\_short)         ev\_shorttext      = DATA(lv\_shorttext) ).     lv\_zuonr = lv\_short && ` ` && lv\_shorttext.   ELSE.     CLEAR: lv\_zuonr.   ENDIF.  \*==========================================\* \* Anlagenverbuchung                        \* \* ==> Verbuchung findet über eine          \* \*     Stellvertretungsanlage statt, die    \* \*     mit der Transaktion ZFIACCKTOGR      \* \*     gepflegt werden kann.                \* \*==========================================\*   CLEAR: lv\_with\_asset.   IF NOT is\_head-equnr IS INITIAL.     lv\_with\_asset = abap\_true.     TRY.         DATA(ls\_ktogr\_anla) = zfiacccl\_services=>get\_post\_anla\_for\_equnr(           EXPORTING             iv\_equnr                  = is\_head-equnr             iv\_use\_wildcard\_not\_found = abap\_true ).       CATCH zcx\_bcgen. \*       \*\*\* Fehler! Keine Anlage aus der Kontogruppe ermittelt         CLEAR: rv\_ok,                ls\_return.         zbcgencl\_services=>do\_compose\_message(           EXPORTING             iv\_type        = 'E'             iv\_id          = 'ZFI\_ACC'             iv\_number      = '032'           IMPORTING             es\_bapiret2    = ls\_return ).         INSERT ls\_return INTO TABLE et\_return.         RETURN.     ENDTRY.   ELSE.     CLEAR: lv\_with\_asset,            ls\_ktogr\_anla.   ENDIF.  \*==========================================\* \* Verbuchungsdaten aufbauen                \* \*==========================================\* \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* \* Kopfdaten                                \* \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*   CLEAR: ls\_header.    ls\_header-bus\_act = 'RFBU'.   ls\_header-comp\_code = is\_head-bukrs.   ls\_header-doc\_date = sy-datum.   ls\_header-doc\_type = 'SB'.   ls\_header-pstng\_date = ls\_period-accrual\_date.   ls\_header-ref\_doc\_no = 'ABGRENZUNG'.   ls\_header-username = sy-uname.    ls\_header-reason\_rev = ls\_period-stgrd.   ls\_header-planned\_rev\_date = ls\_period-stodt.  \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* \* Sachkonten                               \* \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*   REFRESH: lt\_gl,            lt\_extension2,            lt\_asset\_post\_memory.   CLEAR: lv\_mwskz\_skb1.    CASE is\_head-acc\_type.     WHEN zfiacccl\_services=>mc\_accrual\_type-ta. \*     \*\*\* Soll       CLEAR: ls\_gl.       ls\_gl-itemno\_acc = 1.       ls\_gl-item\_text = is\_head-sgtxt.       ls\_gl-gl\_account = ls\_ktopl\_saknr-saknr.       IF NOT lv\_zuonr IS INITIAL.         ls\_gl-alloc\_nmbr = lv\_zuonr.       ENDIF.        INSERT ls\_gl INTO TABLE lt\_gl.  \*     \*\*\* Haben       ls\_gl-itemno\_acc = 2.       ls\_gl-item\_text = is\_head-sgtxt.       ls\_gl-gl\_account = is\_head-saknr.       ls\_gl-costcenter = is\_head-kostl.       ls\_gl-wbs\_element = is\_head-posid.       IF NOT lv\_zuonr IS INITIAL.         ls\_gl-alloc\_nmbr = lv\_zuonr.       ENDIF.  \*     \*\*\* Mehrwertsteuercode       TRY.           ls\_gl-tax\_code = zfiacccl\_services=>get\_saknr\_mwskz(             EXPORTING               iv\_ktopl = is\_head-ktopl               iv\_saknr = is\_head-saknr               iv\_bukrs = is\_head-bukrs ).         CATCH zcx\_bcgen.           CLEAR: ls\_gl-tax\_code.       ENDTRY.        INSERT ls\_gl INTO TABLE lt\_gl.      WHEN zfiacccl\_services=>mc\_accrual\_type-tp. \*     \*\*\* Soll       ls\_gl-itemno\_acc = 1.       ls\_gl-item\_text = is\_head-sgtxt.       ls\_gl-gl\_account = is\_head-saknr.       ls\_gl-costcenter = is\_head-kostl.       ls\_gl-wbs\_element = is\_head-posid.       IF NOT lv\_zuonr IS INITIAL.         ls\_gl-alloc\_nmbr = lv\_zuonr.       ENDIF.  \*     \*\*\* Mehrwertsteuercode       TRY.           ls\_gl-tax\_code = zfiacccl\_services=>get\_saknr\_mwskz(             EXPORTING               iv\_ktopl = is\_head-ktopl               iv\_saknr = is\_head-saknr               iv\_bukrs = is\_head-bukrs ).         CATCH zcx\_bcgen.           CLEAR: ls\_gl-tax\_code.       ENDTRY.  \*------------------------------------------------------------------\* \*     Anlagen ==> Nur für TP relevant                              \* \*------------------------------------------------------------------\*       IF lv\_with\_asset = abap\_true. \*       \*\*\* Positionsdaten \*       \*\*\* ==> Es wird nicht mit der effektiven Anlagennummer die Abgrenzung verbucht, sondern \*       \*\*\*     mit "Stellvertreter" Anlagen         ls\_gl-asset\_no = ls\_ktogr\_anla-anln1.         ls\_gl-sub\_number = ls\_ktogr\_anla-anln2.         ls\_gl-acct\_type = 'A'.  \*       \*\*\* Extension: Anlagebewegungsart kann nicht direkt an das BAPI mitgegeben werden, deswegen \*       \*\*\* die Extension2 abfüllen. Diese wird vom BADI BADI\_ACC\_DOCUMENT verarbeitet.         ls\_extension2\_detail-itemno\_acc = ls\_gl-itemno\_acc.         ls\_extension2\_detail-anbwa = '100'.      "Fix: Anlagezugang          ls\_extension2-structure = 'ZFIGEN\_S\_ASSET\_ACC\_DOC\_EXT2'.         ls\_extension2+30(\*) = ls\_extension2\_detail.         INSERT ls\_extension2 INTO TABLE lt\_extension2.  \*       \*\*\* Memory: Damit alle Positionen mit individuellen Kontierungen verbucht werden können, muss dies im \*       \*\*\* Substitutions BADI AC\_DOCUMENT angepasst werden. Die effektiven Daten müssen mittels Memory Transfer \*       \*\*\* an das BADI übergeben werden         CLEAR: ls\_asset\_post\_memory.         ls\_asset\_post\_memory-awitem = ls\_gl-itemno\_acc.                           "Notwendig         ls\_asset\_post\_memory-anln1 = ls\_gl-asset\_no.                              "Notwendig         ls\_asset\_post\_memory-anln2 = ls\_gl-sub\_number.                            "Notwendig         ls\_asset\_post\_memory-anbwa = ls\_extension2\_detail-anbwa.         ls\_asset\_post\_memory-bukrs = ls\_header-comp\_code.         ls\_asset\_post\_memory-ktopl = zfigencl\_services=>mc\_default\_ktopl.         ls\_asset\_post\_memory-saknr = ls\_gl-gl\_account.         ls\_asset\_post\_memory-kokrs = zfigencl\_services=>mc\_default\_kokrs.         ls\_asset\_post\_memory-kostl = ls\_gl-costcenter.                            "Notwendig Kostenstelle oder PSP Element         IF NOT ls\_gl-wbs\_element IS INITIAL.                                      "==> PSP Element im internen Format!           CALL FUNCTION 'CONVERSION\_EXIT\_ABPSP\_INPUT'             EXPORTING               input     = ls\_gl-wbs\_element             IMPORTING               output    = ls\_asset\_post\_memory-pspnr             EXCEPTIONS               not\_found = 1               OTHERS    = 2.           IF sy-subrc <> 0.             CLEAR: ls\_asset\_post\_memory-pspnr.           ENDIF.         ENDIF.         ls\_asset\_post\_memory-waers = is\_head-waers.         ls\_asset\_post\_memory-amount\_internal = is\_head-acc\_wrbtr.         ls\_asset\_post\_memory-shkzg = 'S'.         INSERT ls\_asset\_post\_memory INTO TABLE lt\_asset\_post\_memory.       ENDIF.        INSERT ls\_gl INTO TABLE lt\_gl.  \*     \*\*\* Haben       CLEAR: ls\_gl.       ls\_gl-itemno\_acc = 2.       ls\_gl-item\_text = is\_head-sgtxt.       ls\_gl-gl\_account = ls\_ktopl\_saknr-saknr.       IF NOT lv\_zuonr IS INITIAL.         ls\_gl-alloc\_nmbr = lv\_zuonr.       ENDIF.        INSERT ls\_gl INTO TABLE lt\_gl.     WHEN OTHERS.   ENDCASE.  \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* \* Betrag                                   \* \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*   REFRESH: lt\_curr.  \* \*\*\* Betrag ins externe Format umwandeln (speziell für Währungen ohne 2 Nachkommastellen)   CLEAR: lv\_amount\_external.   CALL FUNCTION 'BAPI\_CURRENCY\_CONV\_TO\_EXTERNAL'     EXPORTING       currency        = is\_head-waers       amount\_internal = abs( is\_head-acc\_wrbtr )     IMPORTING       amount\_external = lv\_amount\_external.  \* \*\*\* Soll   CLEAR: ls\_curr.   ls\_curr-itemno\_acc = 1.   ls\_curr-currency = is\_head-waers.   ls\_curr-amt\_doccur = lv\_amount\_external.   INSERT ls\_curr INTO TABLE lt\_curr.  \* \*\*\* Haben   CLEAR: ls\_curr.   ls\_curr-itemno\_acc = 2.   ls\_curr-currency = is\_head-waers.   ls\_curr-amt\_doccur = lv\_amount\_external \* -1.   INSERT ls\_curr INTO TABLE lt\_curr.  \*==================================================\* \* Daten ins Memory für das BADI AC\_DOCUMENT        \* \*==================================================\*   zfigencl\_asset=>do\_delete\_asset\_memory( ).   IF NOT lt\_asset\_post\_memory IS INITIAL.     zfigencl\_asset=>set\_export\_asset\_memory( it\_asset\_post = lt\_asset\_post\_memory ).   ENDIF.  \*==========================================\* \* Verbuchung oder Überprüfung              \* \*==========================================\* \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* \* Überprüfung                              \* \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*   IF iv\_test = abap\_true.     CALL FUNCTION 'BAPI\_ACC\_DOCUMENT\_CHECK'       EXPORTING         documentheader = ls\_header       TABLES         accountgl      = lt\_gl         currencyamount = lt\_curr         return         = et\_return         extension2     = lt\_extension2.      LOOP AT et\_return INTO ls\_return       WHERE type = 'S'.       rv\_ok = abap\_true.       EXIT.     ENDLOOP.      IF sy-subrc NE 0. \*     \*\*\* Die generelle Fehlermeldung eliminieren       DELETE et\_return         WHERE type = 'E'           AND id = 'RW'           AND number = '609'.     ENDIF. \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* \* Verbuchung                               \* \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*   ELSE.     CALL FUNCTION 'BAPI\_ACC\_DOCUMENT\_POST'       EXPORTING         documentheader = ls\_header       IMPORTING         obj\_type       = ev\_obj\_type         obj\_key        = ev\_obj\_key         obj\_sys        = ev\_obj\_sys       TABLES         accountgl      = lt\_gl         currencyamount = lt\_curr         return         = et\_return         extension2     = lt\_extension2.      LOOP AT et\_return INTO ls\_return       WHERE type = 'S'.       rv\_ok = abap\_true.       EXIT.     ENDLOOP.      IF NOT rv\_ok = abap\_true. \*     \*\*\* Die generelle Fehlermeldung eliminieren       DELETE et\_return         WHERE type = 'E'           AND id = 'RW'           AND number = '609'.     ENDIF.      IF iv\_with\_commit\_work = abap\_true.       IF rv\_ok = abap\_true.         CALL FUNCTION 'BAPI\_TRANSACTION\_COMMIT'           EXPORTING             wait = abap\_true.       ELSE.         CALL FUNCTION 'BAPI\_TRANSACTION\_ROLLBACK'.       ENDIF.     ENDIF.  \*   \*\*\* Schlussendlich noch die Belegnummer ermitteln     IF rv\_ok = abap\_true.       REFRESH: lt\_belnr.       CALL FUNCTION 'FI\_ACCBELNR\_GET'         TABLES           t\_belnr = lt\_belnr.        LOOP AT lt\_belnr INTO es\_belnr.         EXIT.       ENDLOOP.     ENDIF.   ENDIF.  \*==================================================\* \* Daten wieder aus dem Memory löschen              \* \*==================================================\*   IF NOT lt\_asset\_post\_memory IS INITIAL.     zfigencl\_asset=>do\_delete\_asset\_memory( ).   ENDIF. ENDMETHOD. |
| METHOD bapi\_book. \* Buchungs-BAPI ausführen      DATA: ls\_header         TYPE bapiache09,           ls\_cpd            TYPE bapiacpa09,           lt\_accgl          TYPE zsdttyp\_bapiacgl09,           ls\_accgl          LIKE LINE OF lt\_accgl,           lt\_accountpayable TYPE TABLE OF bapiacap09,           ls\_accountpayable LIKE LINE OF lt\_accountpayable,           lt\_curr           TYPE zsdttyp\_bapiaccr09,           ls\_curr           LIKE LINE OF lt\_curr,           lt\_tax            TYPE zsdttyp\_bapiactx09,           lt\_extension      TYPE zsdttyp\_bapiparex,           lt\_return         TYPE TABLE OF bapiret2.  \* Kopfdaten in BAPI-Struktur füllen     ls\_header-obj\_type   = mc\_awtyp.     ls\_header-bus\_act    = mc\_glvor.     ls\_header-username   = sy-uname.     ls\_header-header\_txt = cs\_book-bktxt.     ls\_header-comp\_code  = zodbcl\_tools=>mc\_ethz\_bukrs.     ls\_header-doc\_date   = cs\_book-bldat.     ls\_header-pstng\_date = sy-datum.     ls\_header-doc\_type   = cs\_book-blart.     ls\_header-ref\_doc\_no = cs\_book-xblnr.  \* CPD     IF cs\_book-cpd\_name1 IS NOT INITIAL.        ls\_cpd-name       = cs\_book-cpd\_name1.       ls\_cpd-name\_2     = cs\_book-cpd\_name2.       ls\_cpd-name\_3     = cs\_book-cpd\_name3.       ls\_cpd-postl\_code = cs\_book-cpd\_pstlz.       ls\_cpd-city       = cs\_book-cpd\_ort01.       ls\_cpd-street     = cs\_book-cpd\_stras.       IF cs\_book-cpd\_land1 IS INITIAL.         cs\_book-cpd\_land1 = `CH`.       ENDIF.       ls\_cpd-country    = cs\_book-cpd\_land1.  \*     \*\*\* Kombination von IBAN + Bankland/Nummer/Schlüssel ist nicht mehr erlaubt (SPS4, ab ca. Juni 2021) \*     \*\*\* Nur noch die IBAN abfüllen \*      SELECT SINGLE \* FROM tiban INTO @DATA(ls\_tiban) \*       WHERE iban = @cs\_book-iban. \* \*      ls\_cpd-bank\_acct  = ls\_tiban-bankn. \*      ls\_cpd-bank\_no    = ls\_tiban-bankl. \*      ls\_cpd-bank\_ctry  = ls\_tiban-banks.       ls\_cpd-iban       = cs\_book-iban.      ENDIF.  \* Position Kreditor     ls\_accountpayable-itemno\_acc = 1.     ls\_accountpayable-vendor\_no  = cs\_book-lifnr.     IF cs\_book-zterm IS NOT INITIAL.       ls\_accountpayable-pmnttrms = cs\_book-zterm. "Zahlungsbedingung     ENDIF.     IF cs\_book-zlsch IS NOT INITIAL.       ls\_accountpayable-pymt\_meth = cs\_book-zlsch.  "Zahlweg     ENDIF.     IF cs\_book-zlspr IS NOT INITIAL.       ls\_accountpayable-pmnt\_block = cs\_book-zlspr. "Zahlsperre     ENDIF.     IF cs\_book-partner\_bk IS NOT INITIAL.       ls\_accountpayable-partner\_bk = cs\_book-partner\_bk.     ENDIF.     APPEND ls\_accountpayable TO lt\_accountpayable.      CLEAR ls\_curr.     ls\_curr-itemno\_acc = 1.     ls\_curr-currency   = `CHF`.     ls\_curr-amt\_doccur = cs\_book-dmbtr.     ls\_curr-amt\_doccur = ls\_curr-amt\_doccur \* -1.     APPEND ls\_curr TO lt\_curr.  \* Position Soll     CLEAR ls\_accgl.     ls\_accgl-itemno\_acc  = 2.     ls\_accgl-item\_text   = cs\_book-sgtxt.     ls\_accgl-gl\_account  = cs\_book-hkont.     IF cs\_book-posid IS NOT INITIAL.       ls\_accgl-wbs\_element = cs\_book-posid.     ENDIF.     IF cs\_book-kostl IS NOT INITIAL.       ls\_accgl-costcenter = cs\_book-kostl.     ENDIF.     APPEND ls\_accgl TO lt\_accgl.  \* Währung und Betrag     CLEAR ls\_curr.     ls\_curr-itemno\_acc  = 2.     ls\_curr-currency    = `CHF`.     ls\_curr-amt\_doccur  = cs\_book-dmbtr.     APPEND ls\_curr TO lt\_curr.  \* BAPI ausführen     CALL FUNCTION 'BAPI\_ACC\_DOCUMENT\_POST'       EXPORTING         documentheader = ls\_header         customercpd    = ls\_cpd       TABLES         accountgl      = lt\_accgl         accountpayable = lt\_accountpayable         currencyamount = lt\_curr         accounttax     = lt\_tax         extension2     = lt\_extension         return         = lt\_return.      READ TABLE lt\_return INDEX 1 INTO DATA(ls\_return).      IF ls\_return-type = `S` AND ls\_return-id = `RW` AND ls\_return-number = `605`.       cs\_book-book\_belnr = ls\_return-message\_v2(10).       cs\_book-book\_gjahr = ls\_return-message\_v2+14(4).       cs\_book-light      = mc\_green.        COMMIT WORK AND WAIT.  \* Beilagen als IXOS speichern       attach\_pdf( CHANGING cs\_book = cs\_book ).      ELSE.        IF ls\_return-type CA 'EA'.          READ TABLE lt\_return INDEX 2 INTO ls\_return. \* Fehler werden rückgängig gemacht         ROLLBACK WORK. \* Protokolltabelle füllen         zbccl\_tools=>save\_bapi\_msg( it\_return = lt\_return ).       ENDIF.        cs\_book-error\_msg = ls\_return-message.       cs\_book-light     = mc\_red.        ROLLBACK WORK.      ENDIF.    ENDMETHOD. \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_   METHOD attach\_pdf. \* PDF-Beilagen als IXOS speichern      DATA: lv\_docart     TYPE saeobjart,           lv\_object\_id  TYPE saeobjid,           lv\_saearchivi TYPE saearchivi,           lv\_doc\_type   TYPE saedoktyp,           lv\_pdf\_merged TYPE xstring,           lt\_pdf        TYPE TABLE OF xstring,           lv\_pdf        TYPE xstring,           lv\_length     TYPE i,           lt\_toa\_dara   TYPE TABLE OF toa\_dara,           ls\_toa\_dara   LIKE LINE OF lt\_toa\_dara,           t1            TYPE i,           t2            TYPE i,           time          TYPE i,           time\_d        TYPE p DECIMALS 2.      CONSTANTS: co\_sap\_object TYPE saeanwdid VALUE 'BKPF'.      SELECT \* FROM zfi\_excel\_pdf INTO TABLE @DATA(lt\_attach)      WHERE book\_id = @cs\_book-book\_id.      IF sy-subrc = 0.        SORT lt\_attach BY zid.        SELECT SINGLE obart INTO @lv\_docart         FROM t003b        WHERE blart = @cs\_book-blart.        IF sy-subrc = 0.  \* Objekt-ID zusammensetzen         CONCATENATE zodbcl\_tools=>mc\_ethz\_bukrs                     cs\_book-book\_belnr                     cs\_book-book\_gjahr                INTO lv\_object\_id.  \* Ermittlung der Archivzuordnung         SELECT SINGLE archiv\_id FROM toaom           INTO @lv\_saearchivi          WHERE sap\_object = @co\_sap\_object            AND ar\_object  = @lv\_docart            AND ar\_status  = 'X'.          IF sy-subrc = 0.  \* Ermittlung Document Type           SELECT SINGLE doc\_type FROM toadv INTO @lv\_doc\_type            WHERE ar\_object = @lv\_docart.            IF sy-subrc = 0.  \* zu einem PDF mergen             LOOP AT lt\_attach INTO DATA(ls\_attach).               APPEND ls\_attach-data TO lt\_pdf.             ENDLOOP.  \* PDF-Merge durchführen             TRY.                 DATA(lo\_merger) = NEW cl\_rspo\_pdf\_merge( ).                 LOOP AT lt\_pdf INTO lv\_pdf.                   lo\_merger->add\_document( lv\_pdf ).                 ENDLOOP.                 lo\_merger->merge\_documents(                   IMPORTING merged\_document = lv\_pdf\_merged ).               CATCH cx\_rspo\_pdf\_merge. \* do nothing             ENDTRY.              lv\_length = xstrlen( lv\_pdf\_merged ).              ls\_toa\_dara-sap\_object = co\_sap\_object.             ls\_toa\_dara-ar\_object  = lv\_docart.             ls\_toa\_dara-object\_id  = lv\_object\_id.             ls\_toa\_dara-function   = 'DARA'.             ls\_toa\_dara-mandant    = sy-mandt.             APPEND ls\_toa\_dara TO lt\_toa\_dara.              IF lv\_length IS NOT INITIAL AND lv\_pdf\_merged IS NOT INITIAL.  \* PDF auf IXOS speichern               CALL FUNCTION 'ARCHIV\_CREATE\_OUTGOINGDOC\_MULT'                 EXPORTING                   pdflen                   = lv\_length                   documentclass            = 'PDF'                   document                 = lv\_pdf\_merged                 TABLES                   arc\_i\_tab                = lt\_toa\_dara                 EXCEPTIONS                   error\_archiv             = 1                   error\_communicationtable = 2                   error\_connectiontable    = 3                   error\_kernel             = 4                   error\_parameter          = 5                   error\_format             = 6                   OTHERS                   = 7.               IF sy-subrc <> 0. \* do nothing               ELSE.                  DELETE FROM zfi\_excel\_pdf WHERE book\_id = @cs\_book-book\_id.                  COMMIT WORK AND WAIT.                  GET RUN TIME FIELD t1.  \* Archiv-ID holen in Klassenattribut                 sy-subrc = 4.                  WHILE sy-subrc <> 0.                    GET RUN TIME FIELD t2.                    SELECT SINGLE \* FROM toa01 INTO ms\_toa01                    WHERE object\_id = lv\_object\_id.                   IF sy-subrc = 0.                     EXIT.                    ELSE. \* spätestens nach 1 Min. abbrechen, sonst gibt's Endlosschlaufe                     time = t2 - t1.                     time\_d = time / 1000000.                     IF time\_d > 60.                       EXIT.                     ENDIF.                   ENDIF.                  ENDWHILE.                ENDIF.              ENDIF.            ENDIF.          ENDIF.        ENDIF.      ELSE.        IF ms\_toa01 IS NOT INITIAL. "Multilink IXOS aus erstem Beleg \* Objekt-ID zusammensetzen         CONCATENATE zodbcl\_tools=>mc\_ethz\_bukrs                     cs\_book-book\_belnr                     cs\_book-book\_gjahr                INTO lv\_object\_id.          ms\_toa01-object\_id = lv\_object\_id.          MODIFY toa01 FROM ms\_toa01.         IF sy-subrc = 0.           COMMIT WORK AND WAIT.         ENDIF.        ENDIF.      ENDIF.    ENDMETHOD. |
| METHOD document\_post. \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* \* FI Dokument verbuchen                                  \* \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* \* Lokale Datendefinition                   \* \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*   DATA: lv\_post\_successful TYPE abap\_bool.    DATA: ls\_return LIKE LINE OF zficl\_exp\_fi\_document=>gt\_return,         ls\_extension2\_itemno TYPE zficl\_exp\_fi\_document=>ls\_ty\_extension2.    DATA: lt\_belnr TYPE STANDARD TABLE OF rf048\_d.    DATA: lt\_accgl TYPE STANDARD TABLE OF bapiacgl09,         lt\_accap TYPE STANDARD TABLE OF bapiacap09,         lt\_curr TYPE STANDARD TABLE OF bapiaccr09,         lt\_extension2 TYPE STANDARD TABLE OF bapiparex.    DATA: lv\_date\_adjusted TYPE abap\_bool,         lv\_date\_new TYPE datum.  \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* \* Coding                                   \* \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  \*==========================================\* \* Export                                   \* \*==========================================\*   CLEAR: ev\_obj\_type,          ev\_obj\_key,          ev\_obj\_sys,          ev\_ok,          es\_belnr.   REFRESH: et\_return.    IF zficl\_exp\_fi\_document=>gs\_header IS INITIAL.     RETURN.   ENDIF.  \*==========================================\* \* Check Buchungsdatum wenn:                \* \* - Erwünscht                              \* \* - Kreditorenbuchung vorhanden ist        \* \*==========================================\*   IF ( iv\_adjust\_posting\_date = abap\_true ) AND      ( NOT zficl\_exp\_fi\_document=>gt\_accap IS INITIAL ).      CLEAR: lv\_date\_adjusted,            lv\_date\_new.      zficl\_exp\_fi\_document=>adjust\_posting\_date(       EXPORTING         iv\_posting\_date\_current = zficl\_exp\_fi\_document=>gs\_header-pstng\_date         iv\_koart                = 'K'         iv\_bukrs                = zficl\_exp\_fi\_document=>gs\_header-comp\_code       IMPORTING          ev\_date\_adjusted       = lv\_date\_adjusted          ev\_posting\_date\_new    = lv\_date\_new ).      IF lv\_date\_adjusted = abap\_true.       zficl\_exp\_fi\_document=>gs\_header-pstng\_date = lv\_date\_new.     ENDIF.   ENDIF.  \*==========================================\* \* Tabellen in Standard Tabellen überführen \* \*==========================================\*   REFRESH: lt\_accgl.   lt\_accgl = zficl\_exp\_fi\_document=>gt\_accgl.   REFRESH: lt\_accap.   lt\_accap = zficl\_exp\_fi\_document=>gt\_accap.   REFRESH: lt\_curr.   lt\_curr = zficl\_exp\_fi\_document=>gt\_curr.   REFRESH: lt\_extension2.   LOOP AT zficl\_exp\_fi\_document=>gt\_extension2 INTO ls\_extension2\_itemno.     INSERT ls\_extension2\_itemno-bapiparex INTO TABLE lt\_extension2.   ENDLOOP.  \*==========================================\* \* Verbuchung anstossen                     \* \*==========================================\*   CALL FUNCTION 'BAPI\_ACC\_DOCUMENT\_POST'     EXPORTING       documentheader    = zficl\_exp\_fi\_document=>gs\_header \*     CUSTOMERCPD       = \*     CONTRACTHEADER    =     IMPORTING       obj\_type          = ev\_obj\_type       obj\_key           = ev\_obj\_key       obj\_sys           = ev\_obj\_sys     TABLES       accountgl         = lt\_accgl \*     ACCOUNTRECEIVABLE =       accountpayable    = lt\_accap \*     ACCOUNTTAX        =       currencyamount    = lt\_curr \*     CRITERIA          = \*     VALUEFIELD        = \*     EXTENSION1        =       return            = zficl\_exp\_fi\_document=>gt\_return \*     PAYMENTCARD       = \*     CONTRACTITEM      =       extension2        = lt\_extension2 \*     REALESTATE        = \*     ACCOUNTWT         =     .    et\_return = zficl\_exp\_fi\_document=>gt\_return.  \*============================================\* \* Verbuchung ok?                             \* \*============================================\*   CLEAR: lv\_post\_successful.   LOOP AT  zficl\_exp\_fi\_document=>gt\_return INTO ls\_return.     IF ls\_return-type = 'S'. \*     \*\*\* Alles ok: Verbuchung möglich!       lv\_post\_successful = abap\_true.       EXIT.     ENDIF.   ENDLOOP.    IF lv\_post\_successful = abap\_true.     ev\_ok = abap\_true.   ELSE.     CLEAR: ev\_ok.   ENDIF.  \*============================================\* \* Commit Work/Rollback wenn gewünscht        \* \* ==> Die Methode kann mehrfach aufgerufen   \* \*     werden. Evtl. soll der Commit erst     \* \*     nach der letzten Buchung durchgeführt  \* \*     werden. In diesem Fall muss die        \* \*     Methode COMMIT\_WORK selbstständig auf- \* \*     gerufen werden                         \* \*============================================\*   IF iv\_with\_commit\_work = abap\_true.     zficl\_exp\_fi\_document=>commit\_work( ev\_ok ).   ENDIF.    IF ev\_ok IS INITIAL.     RETURN.                        "Methode verlassen   ENDIF.  \*============================================\* \* Generierte Belegnnummer herausfinden       \* \*============================================\*   REFRESH: lt\_belnr.   CALL FUNCTION 'FI\_ACCBELNR\_GET'     TABLES       t\_belnr = lt\_belnr.    IF NOT lt\_belnr IS INITIAL.     READ TABLE lt\_belnr INTO es\_belnr INDEX 1.     IF sy-subrc NE 0.       CLEAR: es\_belnr.     ENDIF.   ELSE.     CLEAR: es\_belnr.   ENDIF.  ENDMETHOD. \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ METHOD adjust\_posting\_date. \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* \* Buchungsdatum anpassen                                 \* \* Im Moment ist das nur für die Verbuchung von           \* \* Kreditoren vorgesehen. Sollte eine Periode bereits     \* \* geschlossen sein, so wird das Buchungsdatum auf den    \* \* nächsten 1. des Folgemonates gesetzt                   \* \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* \* Lokale Datendefinition                   \* \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*   DATA: lv\_posting\_date TYPE datum,         lv\_gjahr TYPE gjahr,         lv\_month TYPE month,         lv\_period TYPE frper,         lv\_bukrs TYPE bukrs. \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* \* Coding                                   \* \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  \*==========================================\* \* Export                                   \* \*==========================================\*   CLEAR: ev\_date\_adjusted,          ev\_posting\_date\_new.  \*==========================================\* \* Import                                   \* \*==========================================\*   IF NOT iv\_koart = 'K'.                         "Nur bei Kreditorenverarbeitungen     RETURN.   ENDIF.    CLEAR: lv\_posting\_date.   IF NOT iv\_posting\_date\_current IS INITIAL.     lv\_posting\_date = iv\_posting\_date\_current.   ELSE.     lv\_posting\_date = sy-datum.   ENDIF.    CLEAR: lv\_bukrs.   IF NOT iv\_bukrs IS INITIAL.     lv\_bukrs = iv\_bukrs.   ELSE.     lv\_bukrs = zficl\_exp\_declaration=>co\_bukrs\_default.   ENDIF.  \*==========================================\* \* Sicherheitsabfrage: Wenn im neuen Monat  \* \* noch die Periode geschlossen wäre (zum   \* \* Beispiel im Januar, wenn erst am 6.1.    \* \* die Periode wieder geöffnet würde, so    \* \* darf natürlich nicht der erste Tag des   \* \* Folgemonates automatisch gesetzt werden  \* \* Deshalb hier die Regel:                  \* \* - Alle Abfragen vor dem 20. des Monates  \* \*   werden nicht berücksichtigt            \* \*==========================================\*   IF lv\_posting\_date+6(2) <= 20.     ev\_posting\_date\_new = lv\_posting\_date.     IF ev\_posting\_date\_new <> iv\_posting\_date\_current.       ev\_date\_adjusted = abap\_true.     ENDIF.     RETURN.              "Methode verlassen   ENDIF.  \*==========================================\* \* Geschäftsjahr/Monat ableiten             \* \*==========================================\*   lv\_gjahr = lv\_posting\_date+0(4).   lv\_month = lv\_posting\_date+4(2).   lv\_period = lv\_month.  \*==========================================\* \* Check Periode                            \* \*==========================================\*   CALL FUNCTION 'FI\_PERIOD\_CHECK'     EXPORTING       i\_bukrs          = lv\_bukrs       i\_gjahr          = lv\_gjahr       i\_koart          = iv\_koart       i\_monat          = lv\_period     EXCEPTIONS       error\_period     = 1       error\_period\_acc = 2       invalid\_input    = 3       OTHERS           = 4.    IF sy-subrc EQ 0. \*   \*\*\* Periode ist geöffnet     ev\_posting\_date\_new = lv\_posting\_date.   ELSE. \*   \*\*\* Buchungsperiode ist nicht geöffnet. Nun den ersten Tag des Folgemonates ermitteln     CALL FUNCTION 'END\_TIME\_DETERMINE'       EXPORTING         duration                   = 1         unit                       = 'MON'       IMPORTING         end\_date                   = ev\_posting\_date\_new       CHANGING         start\_date                 = lv\_posting\_date       EXCEPTIONS         factory\_calendar\_not\_found = 1         date\_out\_of\_calendar\_range = 2         date\_not\_valid             = 3         unit\_conversion\_error      = 4         si\_unit\_missing            = 5         parameters\_no\_valid        = 6         OTHERS                     = 7.      IF sy-subrc EQ 0. \*   \*\*\* Der 1. des Folgemonates       ev\_posting\_date\_new+6(2) = '01'.     ELSE.       ev\_posting\_date\_new = lv\_posting\_date.     ENDIF.   ENDIF.  \*==========================================\* \* Export                                   \* \*==========================================\*   IF ev\_posting\_date\_new <> iv\_posting\_date\_current.     ev\_date\_adjusted = abap\_true.   ENDIF.  ENDMETHOD. |
| METHOD do\_post. \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*     DATA: lt\_return        TYPE TABLE OF bapiret2,           lv\_bapi\_error    TYPE abap\_bool,           ls\_post          TYPE ty\_s\_post,           lv\_obj\_type      TYPE bapiache09-obj\_type,           lv\_obj\_key       TYPE bapiache09-obj\_key,           lv\_obj\_key\_doc   TYPE bapiache09-obj\_key,           lv\_obj\_sys       TYPE bapiache09-obj\_sys,           lv\_obj\_key\_txt   TYPE char30,           lv\_loginfo       TYPE zfi\_e\_kng\_loginfo,           ls\_return        TYPE bapiret2,           lv\_target\_docart TYPE char10,           lv\_gjahr         TYPE gjahr. \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*     IF gv\_read\_only EQ abap\_true.       ev\_error = abap\_true.       CALL FUNCTION 'BALW\_BAPIRETURN\_GET2'         EXPORTING           type   = 'S'           cl     = 'ZFIKNG\_BACKEND'           number = 6         IMPORTING           return = ls\_return.       APPEND ls\_return TO et\_return.       EXIT.     ENDIF. \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*     ls\_post = gs\_post."Ab hier nur noch mit ls\_post arbeiten \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*     " Allgemeine Daten abfüllen     CALL FUNCTION 'OWN\_LOGICAL\_SYSTEM\_GET'       IMPORTING         own\_logical\_system             = ls\_post-gs\_bapi\_header-obj\_sys       EXCEPTIONS         own\_logical\_system\_not\_defined = 1         OTHERS                         = 2.     "     ls\_post-gs\_bapi\_header-bus\_act   = 'RFBU'.     ls\_post-gs\_bapi\_header-obj\_key   = ''.     ls\_post-gs\_bapi\_header-obj\_type  = 'BKPFF'.     ls\_post-gs\_bapi\_header-username  = sy-uname. \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*     "Buchungs Positionen erzeugen:     do\_adapt\_post\_pos( CHANGING cs\_post = ls\_post ). \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*     " BAPI Verarbeitung     "Immer zuerst prüfen, grund bei einem echtlauf wird immer eine Belegnummer gezogen     "egal ob danach ein ROLLBACK kommt. >>  1799440     IF iv\_echtlauf EQ abap\_false AND iv\_inboxcheck EQ abap\_true.       IF ls\_post-gs\_bapi\_header-username IS INITIAL.         ls\_post-gs\_bapi\_header-username = sy-uname.       ENDIF.       IF ls\_post-gs\_bapi\_header-pstng\_date IS INITIAL.         ls\_post-gs\_bapi\_header-pstng\_date = sy-datum.       ENDIF.     ENDIF.     CALL FUNCTION 'BAPI\_ACC\_DOCUMENT\_CHECK' "Prüfen       EXPORTING         documentheader = ls\_post-gs\_bapi\_header       TABLES         accountgl      = ls\_post-gt\_bapi\_sachkonto \*       accountreceivable = ls\_recs\_bapi-bapiacar09         accountpayable = ls\_post-gt\_bapi\_kreditor         accounttax     = ls\_post-gt\_bapi\_tax         currencyamount = ls\_post-gt\_bapi\_wp \*       extension1     = ls\_recs\_bapi-bapiacextc         return         = lt\_return.      LOOP AT lt\_return TRANSPORTING NO FIELDS WHERE type CA 'EA'."error or abort       lv\_bapi\_error = abap\_true.       EXIT.     ENDLOOP.      IF iv\_echtlauf EQ abap\_false OR lv\_bapi\_error EQ abap\_true.       ev\_error  = lv\_bapi\_error.       et\_return = lt\_return.       EXIT."Testlauf ende     ENDIF.  \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* \* Echt Buchung     CLEAR lt\_return.     CALL FUNCTION 'BAPI\_ACC\_DOCUMENT\_POST' "Buchen       EXPORTING         documentheader = ls\_post-gs\_bapi\_header       IMPORTING         obj\_type       = lv\_obj\_type         obj\_key        = lv\_obj\_key         obj\_sys        = lv\_obj\_sys       TABLES         accountgl      = ls\_post-gt\_bapi\_sachkonto \*       accountreceivable = ls\_recs\_bapi-bapiacar09         accountpayable = ls\_post-gt\_bapi\_kreditor         accounttax     = ls\_post-gt\_bapi\_tax         currencyamount = ls\_post-gt\_bapi\_wp \*       extension1     = ls\_recs\_bapi-bapiacextc         return         = lt\_return. \*     LOOP AT lt\_return TRANSPORTING NO FIELDS WHERE type CA 'EA'."error or abort       lv\_bapi\_error = abap\_true.       EXIT.     ENDLOOP.     et\_return = lt\_return.     ev\_error  = lv\_bapi\_error.     IF lv\_bapi\_error EQ abap\_true."Fehler       CALL FUNCTION 'BAPI\_TRANSACTION\_ROLLBACK'.     ELSE.       CALL FUNCTION 'BAPI\_TRANSACTION\_COMMIT' EXPORTING wait = 'X'.       lv\_gjahr = lv\_obj\_key+14(4).       do\_finalize\_sucess\_post( iv\_belnr = lv\_obj\_key(10)                                iv\_bukrs = lv\_obj\_key+10(4)                                iv\_gjahr = lv\_gjahr ). \*      "Update Inbox DB \*      UPDATE zfi\_kng\_inbox SET belnr     = lv\_obj\_key(10) \*                               bukrs     = lv\_obj\_key+10(4) \*                               gjahr     = lv\_obj\_key+14(4) \*                               ver\_datum = sy-datum \*                               ver\_uzeit = sy-uzeit \*                               ver\_uname = sy-uname \*                               status    = 'S' \*                           WHERE invoice\_guid EQ gv\_invoice\_guid. \*      "Positionen \*      UPDATE zfi\_kng\_inboxpos SET belnr  = lv\_obj\_key(10) \*                                  bukrs  = lv\_obj\_key+10(4) \*                                  gjahr  = lv\_obj\_key+14(4) \*                                  status = 'S' \*                           WHERE invoice\_guid EQ gv\_invoice\_guid. \*      "Readonly setzen \*      gv\_read\_only = abap\_true. \*      "Update Inbox Log \*      CONCATENATE lv\_obj\_key(10) lv\_obj\_key+10(4) lv\_obj\_key+14(4) INTO lv\_obj\_key\_txt SEPARATED BY space. \*      lv\_loginfo = lv\_obj\_key\_txt. \*      zfi\_kng\_backend\_uty=>do\_write\_inbox\_log( \*        EXPORTING \*          iv\_invoice\_guid = gv\_invoice\_guid \*          iv\_log\_id       = zfi\_kng\_backend\_uty=>ct\_inboxlog-post \*          iv\_loginfo      = lv\_loginfo \*          iv\_no\_commit    = abap\_true \*        EXCEPTIONS \*          error           = 1 \*          OTHERS          = 2 ). \*      IF sy-subrc <> 0. \*        " \*      ENDIF. \*      COMMIT WORK. \*      "Entsperren \*      zfi\_kng\_backend\_uty=>do\_lock( EXPORTING iv\_invoice\_guid = gv\_invoice\_guid \*                                              iv\_action       = 'F' )."Free \*      "Update Archiv-Table \*      lv\_target\_docart = 'ZFIDOC' && gs\_inbox-doc\_type. \*      lv\_obj\_key\_doc   = lv\_obj\_key+10(4) && lv\_obj\_key(10) && lv\_obj\_key+14(4). \*      zfi\_kng\_backend\_uty=>do\_archive\_after\_post( iv\_objkey        = lv\_obj\_key\_doc \*                                                  iv\_invoice\_guid  = gv\_invoice\_guid \*                                                  iv\_target\_docart = lv\_target\_docart ). \*      "Update Callback (Schnittstellen informieren) \*      IF gs\_inbox-test\_backend NE abap\_true. \*        zfi\_kng\_backend\_uty=>do\_callback( iv\_invoice\_guid    = gv\_invoice\_guid \*                                          iv\_callback\_klasse = gs\_intypcus-callback\_class ). \*      ENDIF.     ENDIF.      ENDMETHOD. |
|  |
|  |