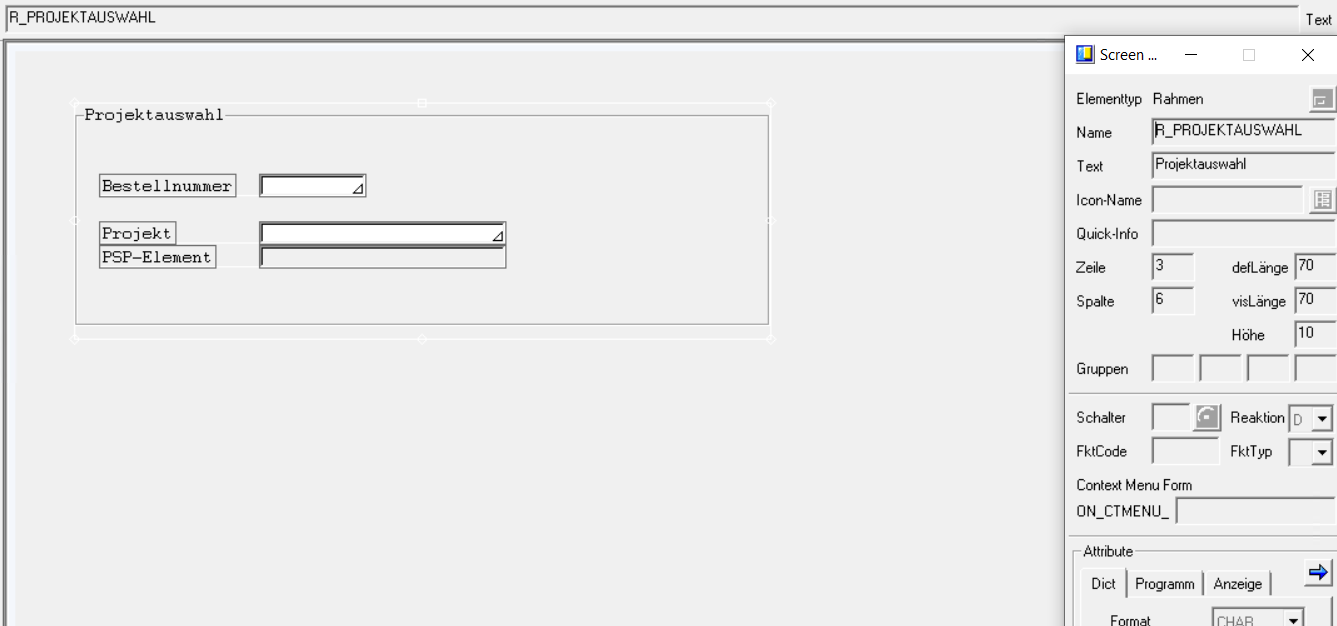
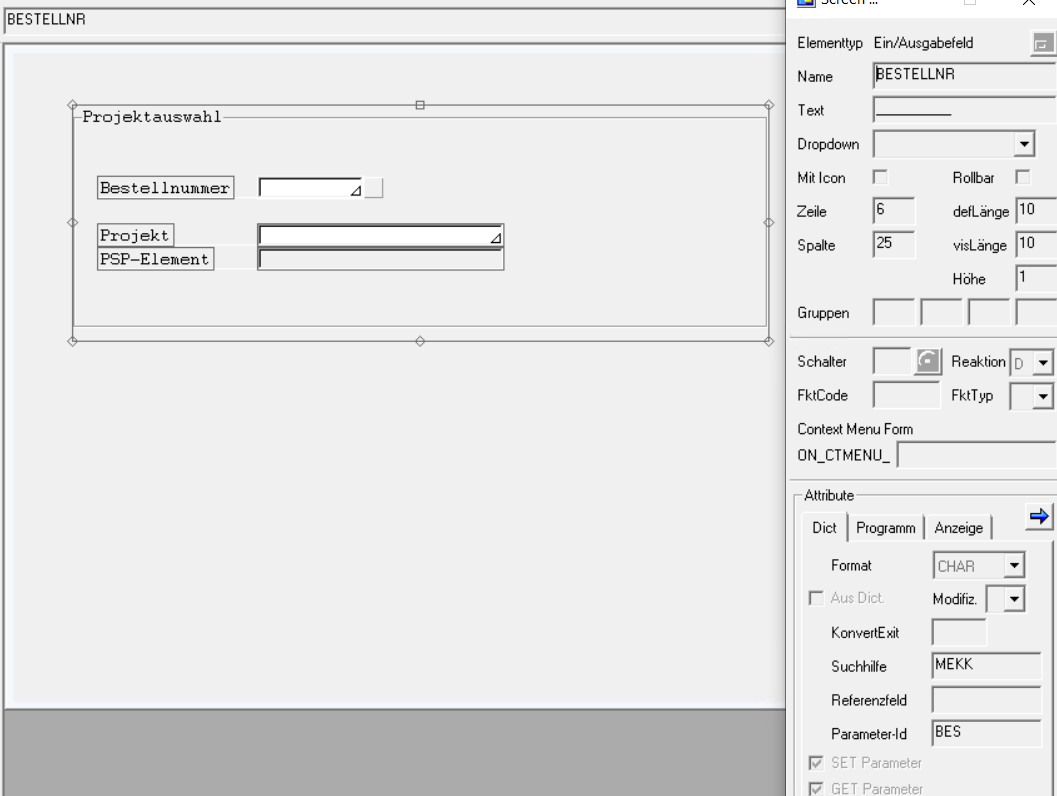
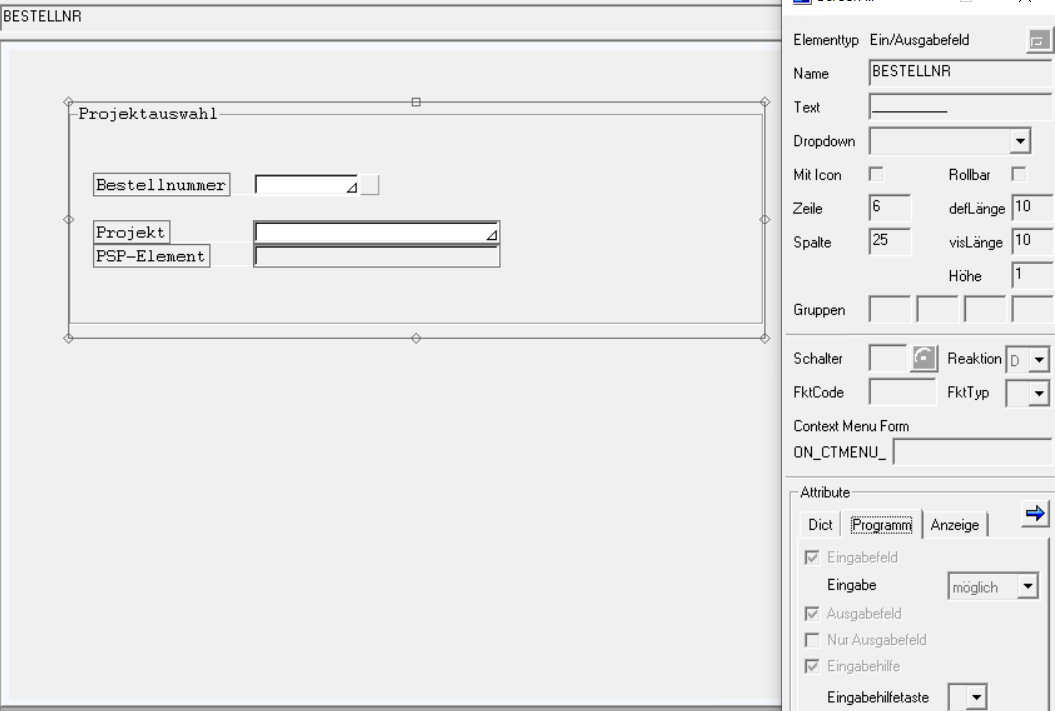
*Modulepool 🡪*

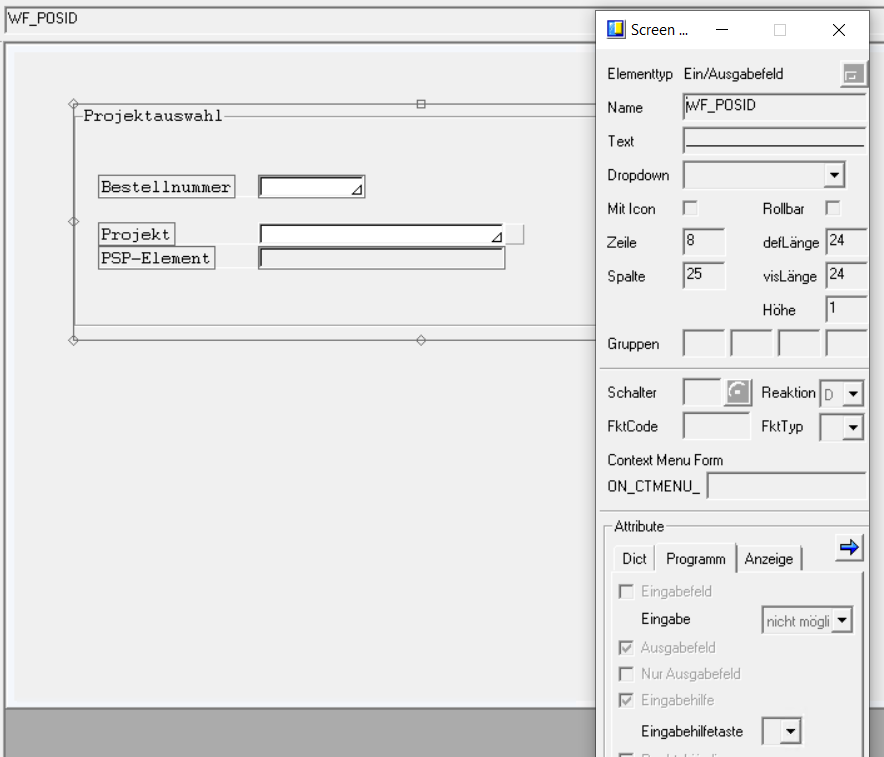
*\*&--------------------------------------------------------------------------------------------------\**  
*\*& Modulpool : /AFM/BAUMOD \**  
*\*& Transaction: /AFM/BAUMOD \**  
*\*&--------------------------------------------------------------------------------------------------\**  
*\*& Responsible: Olaf Meyer \**  
*\*& BEGIS mbH \**  
*\*& Roßstr. 92 \**  
*\*& 40476 Düsseldorf/Germany \**  
*\*& Phone: +49 211 49277-100 \**  
*\*& Departm. : KC-PM \**  
*\*& Date : 05.08 \**  
*\*&--------------------------------------------------------------------------------------------------\**  
*\*& Program Description: \**  
*\*& Modulpool zwecks Abwicklung des Prozesses "Bauen & Modernisieren". \**  
*\*& Angelegt als Rechentool zur Verfolgung der Abschlagszahlungsketten AZ SR EZ FE \**  
*\*& u. zur Ermittlung der für die Erstellung der jeweiligen Rechnungen notwendigen Daten. \* \**  
*\*&--------------------------------------------------------------------------------------------------\**  
*\*& Last Change Vers Programmer Reason \**  
*\*&------------- ---- ---------- --------------------------------------------------------------------\**  
*\*& 08.05 1.0 S. Malina first create \**  
*\* 08.14 1.1 S. MAlina jeweilige Gegenforderungen auf den zu buchenden Rech.betrag (MIRO) \**  
*\* aufgeschlagen, da zusätzliche Buchung(FB01) der \**  
*\* Gegenforderungen sonst auf dem jew.PSP doppelt abgezogen werden \* \**  
*\* 03.16 1.2 S.Malina Für Bauleistungsversicherung \**  
*\* (Gegenforderungszeile im Reiter Gegenforderungen) \**  
*\* die Umsatzsteuer stets auf NULL setzen. \**  
*\* - GEB nicht auf kum.Leistungsstand ermitteln, \**  
*\* sondern auf kum.Leistungsstand abzüglich der GF für \**  
*\* Energiekosten und Schlussreinigung (also auf die Zwischensumme II \**  
*\* des Rechnungsanhang-Dokumentes der MesseN) \**  
*\* - aufgrund Performence-Problemen erfolgt das Einlesen der FI-Belege \**  
*\* im Subscreen 1037 (Freigabe) jetzt nur noch bei Aufruf des Dynpros \**  
*\*& 09.16 2.0 O.Meyer diverses + neue Anforderungen \**  
*\*& 10.17 2.1 O.Meyer Design Festwerte, Dynpro's dynamisieren \**  
*\*& 07.18 2.2 O.Meyer Korrektur 180718(KSG) - Einzelrechn.(ER) muss auch F-43 möglich sein\**  
*\*& MWST Gegenforderung überschreibar!(MSN) \**  
*\*& 09.18 2.3 O.Meyer Korrektur 270918(KSG) - PSP f.Bestelldruck/Nachtrag ermittlung \**  
*\*&--------------------------------------------------------------------------------------------------\**  
  
INCLUDE /afm/baumod\_top.  
  
INCLUDE /afm/baumod\_pbo\_00.  
INCLUDE /afm/baumod\_pai\_00.  
INCLUDE /afm/baumod\_f00.  
  
INCLUDE /afm/baumod\_pbo\_01.  
INCLUDE /afm/baumod\_pai\_01.  
INCLUDE /afm/baumod\_f01.  
  
INCLUDE /afm/baumod\_pbo\_02.  
INCLUDE /afm/baumod\_pai\_02.  
INCLUDE /afm/baumod\_f02.  
  
INCLUDE /afm/baumod\_pbo\_03.  
INCLUDE /afm/baumod\_pai\_03.  
INCLUDE /afm/baumod\_f03.  
  
INCLUDE /afm/baumod\_druck. *"Druck aus Wordvorlagen Windows-Server*  
INCLUDE /afm/baumod\_f\_bds.  
*\*INCLUDE /afm/baumod\_f\_ukt.*  
INCLUDE /afm/baumod\_f\_oaer.  
INCLUDE /afm/baumod\_f\_beg.  
include /afm/baumod\_pai\_04. *"ADD ckn08072020*  
  
INCLUDE /afm/baumod\_check\_inst\_pay. *"BEJK20190702*  
  
*\*&-----------------------------------------------------------------------\**  
*\* FUNCTION CODES FOR TABSTRIP 'PROJ'*  
CONSTANTS: BEGIN OF c\_proj,  
 tab1 LIKE sy-ucomm VALUE 'PROJ\_VERTR',  
 tab2 LIKE sy-ucomm VALUE 'PROJ\_RECHN',  
 END OF c\_proj.  
*\* DATA FOR TABSTRIP 'PROJ'*  
CONTROLS: proj TYPE TABSTRIP.  
DATA: BEGIN OF g\_proj,  
 subscreen LIKE sy-dynnr,  
 prog LIKE sy-repid VALUE '/AFM/BAUMOD',  
 pressed\_tab LIKE sy-ucomm VALUE c\_proj-tab1,  
 END OF g\_proj.  
DATA: ok\_code LIKE sy-ucomm.  
  
  
*\* OUTPUT MODULE FOR TABSTRIP 'PROJ': SETS ACTIVE TAB*  
MODULE proj\_active\_tab\_set OUTPUT.  
 proj-activetab = g\_proj-pressed\_tab.  
 CASE g\_proj-pressed\_tab.  
 WHEN c\_proj-tab1.  
 g\_proj-subscreen = '1002'.  
 WHEN c\_proj-tab2.  
 g\_proj-subscreen = '1003'.  
 WHEN OTHERS.  
*\* DO NOTHING*  
 ENDCASE.  
  
ENDMODULE. *"PROJ\_ACTIVE\_TAB\_SET OUTPUT*  
  
  
*\* INPUT MODULE FOR TABSTRIP 'PROJ': GETS ACTIVE TAB*  
MODULE proj\_active\_tab\_get INPUT.  
 ok\_code = sy-ucomm.  
 CASE ok\_code.  
 WHEN c\_proj-tab1.  
 g\_proj-pressed\_tab = c\_proj-tab1.  
 WHEN c\_proj-tab2.  
 g\_proj-pressed\_tab = c\_proj-tab2.  
 WHEN OTHERS.  
*\* DO NOTHING*  
 ENDCASE.  
ENDMODULE. *"PROJ\_ACTIVE\_TAB\_GET INPUT*

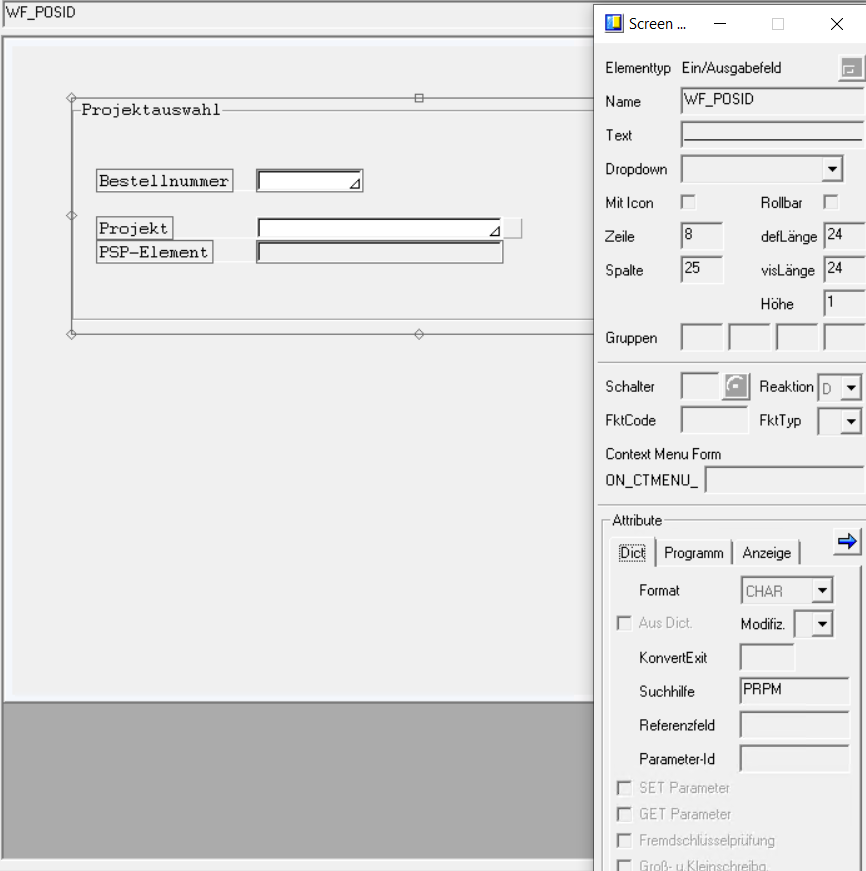
*Tcode : /AFM/BAUMOD*

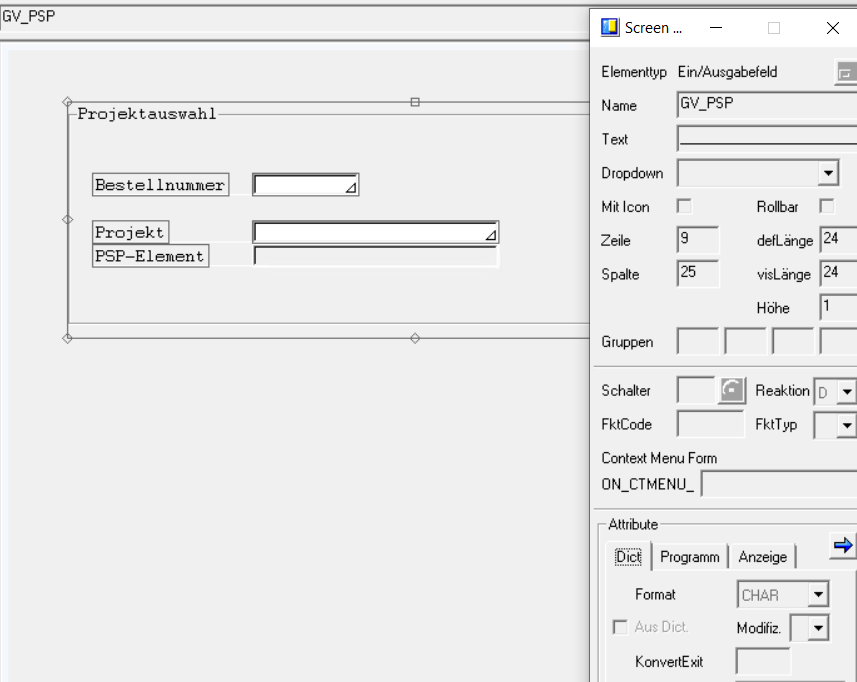
*Screen : 1000*

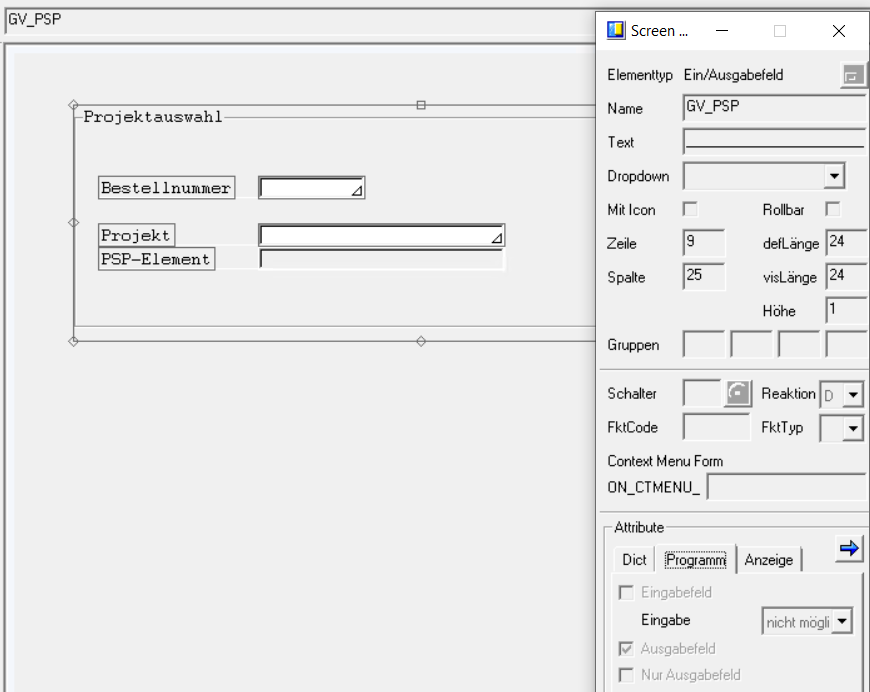
*Initial Screen :🡪   
  
*

**

**

**

**

**

*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\**  
PROCESS BEFORE OUTPUT.  
*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\**  
 MODULE check\_licence.  
 MODULE status\_1000.  
  
*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\**  
PROCESS AFTER INPUT.  
*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\**  
  
 CHAIN.  
 FIELD: bestellnr.  
 MODULE check\_eingabe.  
 MODULE new\_nr ON CHAIN-REQUEST.  
 ENDCHAIN.  
  
 MODULE exit AT EXIT-COMMAND.  
  
 MODULE user\_command\_1000.

*\*&---------------------------------------------------------------------\**  
*\*& Module CHECK\_LICENCE OUTPUT*  
*\*&---------------------------------------------------------------------\**  
*\* text*  
*\*----------------------------------------------------------------------\**  
MODULE check\_licence OUTPUT.  
 DATA lv\_tabname TYPE tabname16.  
 DATA ls\_mandt TYPE /afm/db\_begismnd.  
  
*\* Prüfen wie Tabelle heißt*  
 SELECT SINGLE mandt\_tab FROM /afm/baure\_steu INTO lv\_tabname.  
 IF sy-subrc = 0.  
*\* Auslesen des Customizings*  
 SELECT SINGLE \* FROM (lv\_tabname) INTO ls\_mandt.  
  
 IF ls\_mandt-afm\_baumod IS NOT INITIAL.  
 IF ls\_mandt-validfrom <= sy-datum AND ls\_mandt-validto >= sy-datum.  
 ELSE. *"ENDE*  
 MESSAGE e001(/afm/bau).  
*\* Keine aktive Produktlizenz!*  
 EXIT.  
 ENDIF.  
 ELSE. *"ENDE*  
 MESSAGE e001(/afm/bau).  
 EXIT.  
 ENDIF.  
 ELSE.  
 MESSAGE e001(/afm/bau).  
 EXIT.  
 ENDIF.  
  
ENDMODULE. *" CHECK\_LICENCE OUTPUT*

*\*&---------------------------------------------------------------------\**  
*\*& Module STATUS\_1000 OUTPUT*  
*\*&---------------------------------------------------------------------\**  
*\* text*  
*\*----------------------------------------------------------------------\**  
  
MODULE status\_1000 OUTPUT.  
 SET PF-STATUS 'STATUS\_1000'.  
 SET TITLEBAR 'TITEL\_1000'.  
 CLEAR t\_ucomm.  
  
*\*hinzugefügt von harshal*  
 CASE sy-ucomm.  
 WHEN 'BACK'.  
*\* clear projekt und psp-element*  
 CLEAR :gv\_psp,wf\_posid.  
 erster\_aufruf = 'X'. *"Und sorge dafür, dass im Startscreen neu gelesen wird*  
 ENDCASE.  
ENDMODULE. *" STATUS\_1000 OUTPUT*

*\*&---------------------------------------------------------------------\**  
*\*& Module CHECK\_EINGABE INPUT*  
*\*&---------------------------------------------------------------------\**  
MODULE check\_eingabe INPUT.  
  
 IF bestellnr EQ space AND sy-ucomm NE 'CANCEL'.  
 MESSAGE e000(su) WITH  
 'Bitte die Bestellnummer eingeben!'.  
 ENDIF.  
  
 IF bestellnr NE space.  
*\*{ INSERT TSTK903920 1*  
 DATA lv\_best(10) TYPE n.  
 MOVE bestellnr TO lv\_best.  
 MOVE lv\_best TO bestellnr.  
*\*} INSERT*  
 SELECT SINGLE ebeln FROM ekko INTO ekko-ebeln  
 WHERE ebeln EQ bestellnr.  
 IF sy-subrc NE 0.  
 MESSAGE e000(su) WITH 'Die Bestellung ist nicht vorhanden.'  
 'Bitte Eingabe prüfen.'.  
 ENDIF.  
 ENDIF.  
  
ENDMODULE. *" CHECK\_EINGABE INPUT*

*\*&---------------------------------------------------------------------\**  
*\*& Module NEW\_NR INPUT*  
*\*&---------------------------------------------------------------------\**  
*\* Bei Änderung der Bestellnummer das Flag für ersten Aufruf*  
*\* wieder zurücksetzen, Tabelleninhalte etc. löschen.*  
*\*----------------------------------------------------------------------\**  
MODULE new\_nr INPUT.  
  
 erster\_aufruf = 'X'.  
  
 CLEAR gesperrt.  
  
 CLEAR: wt\_ford,  
 wt\_ford[],  
 wt\_leist,  
 wt\_leist[],  
 wt\_projre,  
 wt\_projre[],  
 wt\_langtext,  
 wt\_langtext[],  
 wt\_text,  
 wt\_text[],  
 wt\_hilf,  
 wt\_hilf[],  
 re\_max,  
 id,  
 rechnung,  
 art,  
 anlnr,  
 wa\_bauvertr,  
 wa\_bauvertr\_act,  
 anz\_re,  
 adresse,  
*\* gegenstand,*  
*\* vergabeart,*  
 faell\_ac,  
 faell\_sr,  
 vor\_buerg\_ein,  
 vor\_buerg\_aus,  
 vor\_bem,  
 vor\_betrag,  
 erf\_buerg\_ein,  
 erf\_buerg\_aus,  
 erf\_bem,  
 erf\_betrag,  
 gew\_buerg\_ein,  
 gew\_buerg\_aus,  
 gew\_buerg\_bem,  
 gew\_betrag,  
 gew\_sr\_am,  
 gew\_abn\_am,  
 gew\_maeng,  
 gew\_frist,  
 gew\_nachweis,  
 gew\_bem,  
 psp,  
 wf\_posid.  
  
*\* Clearen der Felder in 1033 Freigabe*  
 CLEAR: leist\_n, leist\_b, einbeh\_n, einbeh\_b, gegenford\_n, gegenford\_b,  
 az\_n, az\_b, bish\_vz\_n, bish\_vz\_b, vz\_n, vz\_b, re\_betr\_n,  
 re\_betr\_b, zus\_einb\_n, zus\_einb\_b, enth\_mwst, enth\_mwst\_betr,  
 z\_freig\_n, z\_freig\_b, bem.  
*\* Clearen Felder in 1031 Leistung*  
 CLEAR: proz\_act, gpreis\_net\_akt, gpreis\_bru\_akt, auf\_akt,  
 gpreis\_auf\_net\_akt, gpreis\_auf\_bru\_akt.  
*\*{ INSERT TSTK904259 1*  
 CLEAR rekurz.  
*\*} INSERT*  
  
*\* Clearen Felder in 1035 Buchen*  
 CLEAR: referenz, re\_betr, steu\_betr, re\_betr\_net, xblnr, zuordnung, text,  
 sich\_einb, zus\_einb, re\_erfolgt, gebucht, zeb\_gebucht, monat, monat\_zeb, zeb\_aufgeloest, seb\_aufgeloest,  
 monat\_head, budat\_head. *"ADD ckn15052020*  
  
*\* Textcontainer clearen*  
 IF NOT editor IS INITIAL. *"nur wenn Textcontainer existiert*  
 CALL METHOD editor->delete\_text  
*\* importing*  
 EXCEPTIONS  
 OTHERS = 1.  
 ENDIF.  
  
ENDMODULE. *" NEW\_NR INPUT*

*\*&---------------------------------------------------------------------\**  
*\*& Module EXIT INPUT*  
*\*&---------------------------------------------------------------------\**  
MODULE exit INPUT.  
  
 CASE sy-ucomm.  
  
 WHEN 'EXIT'.  
  
 IF gesperrt NE 'X'. *"d.h. nicht bereits durch anderen User ge-*  
 *"sperrt, sondern durch diesen Lauf. Somit*  
 *"muß hier wieder entsperrt werden*  
 PERFORM dequeue USING bestellnr. *" psp.*  
 ENDIF.  
 PERFORM free\_objects.  
 LEAVE PROGRAM.  
 ENDCASE.  
  
ENDMODULE. *" EXIT INPUT*

*\*&---------------------------------------------------------------------\**  
*\*& Module USER\_COMMAND\_1000 INPUT*  
*\*&---------------------------------------------------------------------\**  
MODULE user\_command\_1000 INPUT.  
  
 CASE sy-ucomm.  
  
 WHEN 'EXIT'.  
 PERFORM popup\_to\_decide USING answer.  
 CASE answer.  
 WHEN 'A'.  
 subscr\_re = '1035'.  
 sy-ucomm = 'PROJ\_RECHN'.  
 LEAVE TO SCREEN '1001'.  
  
 WHEN '1'.  
 PERFORM datensicherung. *" Sicherung Bauvertragsdaten*  
 IF gesperrt NE 'X'. *"d.h. nicht bereits durch anderen User ge-*  
 *"sperrt, sondern durch diesen Lauf. Somit*  
 *"muß hier wieder entsperrt werden*  
 PERFORM dequeue USING bestellnr. *" psp.*  
 ENDIF.  
 PERFORM free\_objects.  
 LEAVE PROGRAM.  
  
 WHEN '2'.  
 PERFORM free\_objects.  
 IF gesperrt NE 'X'.  
 PERFORM dequeue USING bestellnr. *" psp.*  
 ENDIF.  
 LEAVE PROGRAM.  
 ENDCASE.  
  
 WHEN 'CANCEL'.  
 IF gesperrt IS INITIAL.  
  
 PERFORM popup\_to\_decide USING answer.  
 CASE answer.  
 WHEN 'A'.  
*\* G\_PROJ-SUBSCREEN = '1003'.*  
 subscr\_re = '1035'.  
 sy-ucomm = 'PROJ\_RECHN'.  
 LEAVE TO SCREEN '1001'.  
  
 WHEN '1'.  
 PERFORM datensicherung. *" Sicherung Bauvertragsdaten*  
 PERFORM free\_objects.  
 IF gesperrt NE 'X'.  
 PERFORM dequeue USING bestellnr. *" psp.*  
 ENDIF.  
 LEAVE PROGRAM.  
  
 WHEN '2'.  
 PERFORM free\_objects.  
 IF gesperrt NE 'X'.  
 PERFORM dequeue USING bestellnr. *" psp.*  
 ENDIF.  
 LEAVE PROGRAM.  
 ENDCASE.  
  
 ELSE.  
 PERFORM free\_objects.  
 LEAVE PROGRAM.  
 ENDIF.  
  
 WHEN 'READ'.  
 PERFORM read\_data.  
*\* Sperre bei Anzeige macht keinen Sinn!*  
*\* PERFORM enqueue USING bestellnr. " psp.*  
 PERFORM check\_installment\_payment *"BEJK20190702 siehe FORM HEADER*  
 TABLES wt\_projre.  
 LEAVE TO SCREEN 1001.  
  
 WHEN 'OK\_CODE'.  
*\*hinzugefügt von harshal*  
*\* clear projekt und psp-element*  
*\*\* CLEAR :gv\_psp,wf\_posid. "DEL ckn20210817 - Enter leert es auch im main screen*  
 PERFORM read\_data.  
 LEAVE TO SCREEN 1000.  
  
  
  
 ENDCASE.  
  
ENDMODULE. *" USER\_COMMAND\_1000 INPUT*

*\*&---------------------------------------------------------------------\**  
*\*& Form dequeue*  
*\*&---------------------------------------------------------------------\**  
*\* Tabellen entsperren*  
*\*----------------------------------------------------------------------\**  
*\* -->P\_BESTELLNR text*  
*\* -->P\_PSP text*  
*\*----------------------------------------------------------------------\**  
FORM dequeue USING p\_bestellnr.  
 *" P\_PSP.*  
  
 CALL FUNCTION 'DEQUEUE\_/AFM/BAUMOD'  
 EXPORTING  
*\* MODE\_/AFM/BAUPROJ = 'E'*  
 mandt = sy-mandt  
*\* PSPNR = p\_psp*  
 ebeln = p\_bestellnr  
*\* X\_PSPNR = ' '*  
*\* X\_EBELN = ' '*  
*\* \_SCOPE = '3'*  
*\* \_SYNCHRON = ' '*  
*\* \_COLLECT = ' '*  
 .  
  
  
 IF sy-subrc NE 0.  
 MESSAGE w000(su) WITH 'Nicht alle zugehörigen Tabellen konnten'  
 'entsperrt werden!'.  
 ENDIF.  
  
ENDFORM. *" dequeue*

*\*&---------------------------------------------------------------------\**  
*\*& Form FREE\_OBJECTS\_BDS*  
*\*&---------------------------------------------------------------------\**  
*\* Instanzen für Auftrags- und Nachtragsdokumente abbauen/freigeben*  
*\*----------------------------------------------------------------------\**  
*\* -->P\_GO\_CONTAINER\_B/N text*  
*\* -->P\_GO\_DOCUMENT\_B/N text*  
*\* -->P\_GO\_CONTROL\_B/N text*  
*\* -->P\_GO\_WORDDOC\_B/N text*  
*\*----------------------------------------------------------------------\**  
 FORM free\_objects\_bds USING io\_container TYPE REF TO cl\_gui\_custom\_container  
 io\_document TYPE REF TO i\_oi\_document\_proxy  
 io\_control TYPE REF TO i\_oi\_container\_control  
 io\_worddoc TYPE REF TO i\_oi\_word\_processor\_document.  
  
  
 IF NOT io\_document IS INITIAL.  
 CALL METHOD io\_document->close\_document.  
 ENDIF.  
  
 IF NOT io\_control IS INITIAL.  
 CALL METHOD io\_control->destroy\_control  
 EXPORTING  
 no\_flush = ' '.  
 ENDIF.  
  
 FREE io\_worddoc.  
 CLEAR io\_worddoc.  
  
 IF NOT io\_document IS INITIAL.  
 CALL METHOD io\_document->release\_document.  
 ENDIF.  
  
 FREE: io\_document,  
 io\_control.  
  
 CLEAR: io\_document,  
 io\_control.  
  
 IF NOT io\_container IS INITIAL.  
 CALL METHOD io\_container->free  
 EXCEPTIONS  
 cntl\_error = 1  
 cntl\_system\_error = 2  
 OTHERS = 3.  
 IF sy-subrc <> 0.  
 MESSAGE ID sy-msgid TYPE sy-msgty NUMBER sy-msgno  
 WITH sy-msgv1 sy-msgv2 sy-msgv3 sy-msgv4.  
 ENDIF.  
  
 FREE io\_container.  
 ENDIF.  
  
 ENDFORM. *" FREE\_OBJECTS\_BDS*

*Include 🡪 /AFM/BAUMOD\_PAI\_01*

*\*&---------------------------------------------------------------------\**  
*\*& Form popup\_to\_decide*  
*\*&---------------------------------------------------------------------\**  
*\* Abfragen bzgl.Datensicherung bei Verlassen des Programmes*  
*\*----------------------------------------------------------------------\**  
FORM popup\_to\_decide USING answer.  
  
*\*case bukrs.*  
  
*\* when '1001'.*  
 CALL FUNCTION 'POPUP\_TO\_DECIDE'  
 EXPORTING  
*\* DEFAULTOPTION = '1'*  
 textline1 = TEXT-001  
 textline2 = TEXT-002  
 textline3 = ' '  
 text\_option1 = 'Ja'  
 text\_option2 = 'Nein'  
*\* ICON\_TEXT\_OPTION1 = ' '*  
*\* ICON\_TEXT\_OPTION2 = ' '*  
 titel = 'Sicherungsabfrage'  
 start\_column = 25  
 start\_row = 15  
*\* CANCEL\_DISPLAY = 'X'*  
 IMPORTING  
 answer = answer.  
  
*\* when '9992'.*  
*\**  
*\* CALL FUNCTION 'POPUP\_TO\_DECIDE'*  
*\* EXPORTING*  
*\*\* DEFAULTOPTION = '1'*  
*\* TEXTLINE1 = text-001*  
*\* TEXTLINE2 = text-002*  
*\* TEXTLINE3 = text-008*  
*\* TEXT\_OPTION1 = 'Ja'*  
*\* TEXT\_OPTION2 = 'Nein'*  
*\*\* ICON\_TEXT\_OPTION1 = ' '*  
*\*\* ICON\_TEXT\_OPTION2 = ' '*  
*\* TITEL = 'Sicherungsabfrage'*  
*\* START\_COLUMN = 25*  
*\* START\_ROW = 15*  
*\*\* CANCEL\_DISPLAY = 'X'*  
*\* IMPORTING*  
*\* ANSWER = answer*  
*\* .*  
*\**  
*\* when others.*  
*\* leave program.*  
*\*endcase.*  
  
ENDFORM. *" popup\_to\_decide*

Include 🡪 /AFM/BAUMOD\_F01

*\*&---------------------------------------------------------------------\**  
*\*& Form datensicherung*  
*\*&---------------------------------------------------------------------\**  
FORM datensicherung.  
  
 DATA: ls\_bauproj LIKE /afm/bauproj.*"BEJK20190717*  
*\* Sicherung Bauvertragsdaten*  
 MOVE: psp TO wa\_bauvertr\_act-pspnr,  
 bestellnr TO wa\_bauvertr\_act-ebeln,  
 kreditor TO wa\_bauvertr\_act-lifnr,  
 anlnr TO wa\_bauvertr\_act-anlnr,  
  
*\* Teilweise werden die Daten zu PAI schon in die wt\_bauvertr\_act*  
*\* geschrieben, aber nur bei Änderung deren Feldinhalte.*  
*\* gegenstand to wa\_bauvertr\_act-gegenstand,*  
 erfuell\_sich TO wa\_bauvertr\_act-erfuell\_sich,  
 gewaehrleist TO wa\_bauvertr\_act-gewaehrleist,  
 gewaehrl\_dauer TO wa\_bauvertr\_act-gewaehrl\_dauer,  
 straf\_nach TO wa\_bauvertr\_act-straf\_nach,  
 straf\_max TO wa\_bauvertr\_act-straf\_max,  
*\* write aufschlag to wa\_bauvertr\_act-aufschlag.*  
 aufschlag TO wa\_bauvertr\_act-aufschlag.  
  
 CASE radio\_betr.  
 WHEN 'X'.  
 MOVE vertr\_straf\_betr TO wa\_bauvertr\_act-vertr\_straf\_betr.  
 CLEAR wa\_bauvertr\_act-vertr\_straf.  
 WHEN OTHERS.  
 MOVE vertr\_straf\_betr TO wa\_bauvertr\_act-vertr\_straf.  
 CLEAR wa\_bauvertr\_act-vertr\_straf\_betr.  
 ENDCASE.  
  
 MOVE: *"vertr\_straf\_betr to wa\_bauvertr\_act-vertr\_straf,*  
 radio\_proz TO wa\_bauvertr\_act-straf\_in\_proz,  
 radio\_betr TO wa\_bauvertr\_act-straf\_als\_betr,  
 radio\_kal TO wa\_bauvertr\_act-straf\_pro\_kal,  
 radio\_wer TO wa\_bauvertr\_act-straf\_pro\_werk,  
 anz\_re TO wa\_bauvertr\_act-anz\_re,  
 adresse TO wa\_bauvertr\_act-adresse,  
 faell\_ac TO wa\_bauvertr\_act-faell\_acontore,  
 faell\_sr TO wa\_bauvertr\_act-faell\_schlussre,  
 waers TO wa\_bauvertr\_act-waers.  
*\* vergabeart to wa\_bauvertr\_act-vergabeart.*  
  
 MOVE: sum\_net TO wa\_bauvertr\_act-gpreis\_netto,  
 sum\_bru TO wa\_bauvertr\_act-gpreis\_brutto,  
 sum\_auf\_net TO wa\_bauvertr\_act-gpreis\_auf\_netto,  
 sum\_auf\_bru TO wa\_bauvertr\_act-gpreis\_auf\_brutt,  
 mwst TO wa\_bauvertr\_act-mwst,  
 bukrs TO wa\_bauvertr\_act-bukrs.  
  
*\* Sicherung Bauleistungsversicherung*  
 READ TABLE wt\_ford WITH KEY key = 'BAUL'.  
 IF sy-subrc = 0.  
 MOVE: wt\_ford-bez TO wa\_bauvertr\_act-bauleist\_bez, *"ADD ckn24062020*  
 wt\_ford-als TO wa\_bauvertr\_act-bauleist\_als, *"ADD ckn24062020*  
 wt\_ford-proz TO wa\_bauvertr\_act-bauleist\_vers,  
 wt\_ford-netto\_betr TO wa\_bauvertr\_act-bauleist\_betr\_n,  
 wt\_ford-brutto\_betr TO wa\_bauvertr\_act-bauleist\_betr\_b,  
 wt\_ford-mwst TO wa\_bauvertr\_act-bauleist\_vers\_mwst.  
 ENDIF.  
*\* Sicherung Kostenumlage*  
 READ TABLE wt\_ford WITH KEY key = 'KOST'.  
 IF sy-subrc = 0.  
 MOVE: wt\_ford-bez TO wa\_bauvertr\_act-kostenum\_bez, *"ADD ckn24062020*  
 wt\_ford-als TO wa\_bauvertr\_act-kostenum\_als, *"ADD ckn24062020*  
 wt\_ford-proz TO wa\_bauvertr\_act-kostenumlage,  
 wt\_ford-netto\_betr TO wa\_bauvertr\_act-kostenum\_betr\_n,  
 wt\_ford-brutto\_betr TO wa\_bauvertr\_act-kostenum\_betr\_b,  
 wt\_ford-mwst TO wa\_bauvertr\_act-kostenumlage\_mwst.  
 ENDIF.  
*\* Sicherung Kostenumlage Schlussreinigung*  
 READ TABLE wt\_ford WITH KEY key = 'REIN'.  
 IF sy-subrc = 0.  
 MOVE: wt\_ford-bez TO wa\_bauvertr\_act-reinigung\_bez, *"ADD ckn24062020*  
 wt\_ford-als TO wa\_bauvertr\_act-reinigung\_als, *"ADD ckn24062020*  
 wt\_ford-proz TO wa\_bauvertr\_act-reinigung,  
 wt\_ford-netto\_betr TO wa\_bauvertr\_act-rein\_betr\_n,  
 wt\_ford-brutto\_betr TO wa\_bauvertr\_act-rein\_betr\_b,  
 wt\_ford-mwst TO wa\_bauvertr\_act-reinigung\_mwst.  
 ENDIF.  
  
*\*&---------------------------------------------------------------------\**  
  
*\* Sicherung Vertragserfüllungsdaten*  
*\* Die Daten werden sofort unter PAI in die wt\_bauvertr\_act geschrieben,*  
*\* wenn der Inhalt eines der Felder im Dynpro geändert wird.*  
  
*\*move: VOR\_BUERG\_EIN to wa\_bauvertr\_act-vor\_buerg\_ein,*  
*\* VOR\_BUERG\_AUS to wa\_bauvertr\_act-vor\_buerg\_aus,*  
*\* VOR\_BEM to wa\_bauvertr\_act-vor\_bem,*  
*\* VOR\_BETRAG to wa\_bauvertr\_act-vor\_betrag,*  
*\* ERF\_BUERG\_EIN to wa\_bauvertr\_act-erf\_buerg\_ein,*  
*\* ERF\_BUERG\_AUS to wa\_bauvertr\_act-erf\_buerg\_aus,*  
*\* ERF\_BEM to wa\_bauvertr\_act-erf\_bem,*  
*\* ERF\_BETRAG to wa\_bauvertr\_act-erf\_betrag,*  
*\* GEW\_BUERG\_EIN to wa\_bauvertr\_act-gew\_buerg\_ein,*  
*\* GEW\_BUERG\_AUS to wa\_bauvertr\_act-gew\_buerg\_aus,*  
*\* GEW\_BUERG\_BEM to wa\_bauvertr\_act-gew\_buerg\_bem,*  
*\* GEW\_BETRAG to wa\_bauvertr\_act-gew\_betrag,*  
*\* GEW\_SR\_AM to wa\_bauvertr\_act-gew\_sr\_am,*  
*\* GEW\_ABN\_AM to wa\_bauvertr\_act-gew\_abn\_am,*  
*\* GEW\_MAENG to wa\_bauvertr\_act-gew\_maeng,*  
*\* GEW\_FRIST to wa\_bauvertr\_act-gew\_frist,*  
*\* GEW\_NACHWEIS to wa\_bauvertr\_act-gew\_nachweis,*  
*\* GEW\_BEM to wa\_bauvertr\_act-gew\_bem.*  
  
*\* Sichern der Daten zur Ausbezahlung des Gewährleistungseinbehaltes*  
 MOVE: radio\_voll TO wa\_bauvertr\_act-gew\_eb\_kein,  
 radio\_teil TO wa\_bauvertr\_act-gew\_eb\_teilw,  
 anteil\_betr TO wa\_bauvertr\_act-gew\_eb\_teil\_betr,  
 radio\_null TO wa\_bauvertr\_act-gew\_eb\_voll.  
*\* Betrag für teilw.Einbeh.auch nur für den Fall sichern*  
 IF radio\_teil NE 'X'.  
 CLEAR wa\_bauvertr\_act-gew\_eb\_teil\_betr.  
 ENDIF.  
  
 IF ws\_steuerung-set\_protocol IS NOT INITIAL. *"ADD ckn29052020 - Es gibt bei UKT einen Abbruch - Vermutlich hier durch*  
*\*START 20190717 BEJK Protokollierung*  
  
 SELECT SINGLE \* FROM /afm/bauproj INTO ls\_bauproj  
 WHERE ebeln = wa\_bauvertr\_act-ebeln.  
 IF sy-subrc <> 0.  
 wa\_bauvertr\_act-uname\_create = sy-uname.  
 CALL FUNCTION 'ABI\_TIMESTAMP\_CONVERT\_INTO'  
 EXPORTING  
 iv\_date = sy-datlo  
 iv\_time = sy-timlo  
 IMPORTING  
 ev\_timestamp = wa\_bauvertr\_act-time\_create.  
 ELSEIF ls\_bauproj <> wa\_bauvertr\_act.  
 wa\_bauvertr\_act-uname\_change = sy-uname.  
 CALL FUNCTION 'ABI\_TIMESTAMP\_CONVERT\_INTO'  
 EXPORTING  
 iv\_date = sy-datlo  
 iv\_time = sy-timlo  
 IMPORTING  
 ev\_timestamp = wa\_bauvertr\_act-time\_change.  
 ENDIF.  
  
*\*ENDE 20190717 BEJK*  
 ENDIF. *"ADD ckn29052020 - Es gibt bei UKT einen Abbruch - Vermutlich hier durch*  
  
 *"ADD ckn20210810 - Start - ALLES SICHERN UND VERGLEICHEN!*  
 IF wa\_bauproj\_comp IS NOT INITIAL.  
 IF wa\_bauproj\_comp <> wa\_bauvertr\_act.  
  
 DATA: lv\_zero TYPE xfeld.  
 lv\_zero = ''.  
 PERFORM compare\_save CHANGING lv\_zero.  
  
 IF lv\_zero IS NOT INITIAL.  
 CALL FUNCTION 'POPUP\_TO\_INFORM'  
 EXPORTING  
 titel = 'Kopfdaten geleert'  
 txt1 = 'Unerwartete Änderung der Vertragsdetails'  
 txt2 = 'bitte die Bestellung bei Herrn Grassmüller melden'  
*\* TXT3 = ' '*  
*\* TXT4 = ' '*  
 .  
  
 ENDIF.  
 ENDIF.  
 ENDIF.  
 *"ADD ckn20210810 - Ende*  
  
  
 MODIFY /afm/bauproj FROM wa\_bauvertr\_act.  
  
 wa\_bauproj\_comp = wa\_bauvertr\_act. *"ADD ckn20210810*  
  
  
*\*&---------------------------------------------------------------------\**  
  
*\* Sicherung Gegenforderungen*  
*\* vorher bisherige Einträge löschen, dann aktuelle Einträge sichern*  
 SELECT \* FROM /afm/bauprojbel WHERE ebeln EQ bestellnr.  
 DELETE /afm/bauprojbel.  
 ENDSELECT.  
  
 LOOP AT wt\_ford WHERE key NE 'BAUL' AND key NE 'KOST' AND key NE 'REIN'.  
 MOVE: psp TO /afm/bauprojbel-pspnr,  
 bestellnr TO /afm/bauprojbel-ebeln,  
 wt\_ford-key TO /afm/bauprojbel-zaehler.  
 MOVE-CORRESPONDING wt\_ford TO /afm/bauprojbel.  
 INSERT /afm/bauprojbel.  
 ENDLOOP.  
  
*\*&---------------------------------------------------------------------\**  
  
*\* Inhalt des Textcontainers erst in Itab sichern, dann Wegschreiben als Langtext.*  
 IF NOT editor IS INITIAL.  
 PERFORM save\_text\_to\_table CHANGING wt\_text.  
 ENDIF.  
  
*\*----------------------------------------------------------------------\**  
*\* Baurechnung hinzufügen und entfernen \**  
*\*----------------------------------------------------------------------\**  
*\*Neue Rechnungen einfügen*  
 LOOP AT wt\_projre.  
 MOVE-CORRESPONDING wt\_projre TO /afm/bauprojre.  
 MODIFY /afm/bauprojre.  
 ENDLOOP.  
  
*\* Gelöschte entfernen*  
 LOOP AT wt\_projre\_clear.  
 MOVE-CORRESPONDING wt\_projre\_clear TO /afm/bauprojre.  
 DELETE /afm/bauprojre.  
 ENDLOOP.  
  
  
  
  
*\* -------------------------------------------------------------------- \**  
*\* Sicherung des Beauftragungsschreibens \**  
*\* -------------------------------------------------------------------- \**  
 DATA lv\_comp\_id TYPE bds\_compid.  
  
 CONCATENATE 'Auftragsschreiben Bestell.-Nr:' space gv\_objectkey '.doc' INTO lv\_comp\_id.  
  
 TYPES tyt\_errors TYPE STANDARD TABLE OF REF TO i\_oi\_error  
 WITH NON-UNIQUE DEFAULT KEY.  
  
 DATA lo\_error TYPE REF TO i\_oi\_error.  
 DATA lv\_changed TYPE i.  
  
*\*&---------------------------------------------------------------------\**  
  
 CLEAR lv\_changed.  
  
 IF NOT go\_document\_b IS INITIAL.  
 CALL METHOD go\_document\_b->has\_changed  
*\* EXPORTING*  
*\* no\_flush = ' '*  
 IMPORTING  
 error = lo\_error  
*\* retcode =*  
 ret\_value = lv\_changed.  
 ENDIF.  
  
 IF NOT lv\_changed IS INITIAL.  
 PERFORM save\_document USING go\_document\_b  
 gv\_doc\_open\_b  
 lv\_comp\_id  
 c\_class\_b  
 gv\_exist\_b  
 gv\_kz\_vorl\_b  
 'B'  
 gv\_source *"nur genutzt im kleinen Bauunterhalt*  
 bestellnr. *"nur genutzt im kleinen Bauunterhalt*  
 ENDIF.  
  
*\* Die Nachtragsschreiben zu den jeweiligen Bestell-Nachtragspositionen werden direkt*  
*\* im Subscreen 1026 mittels Drucktaste gesichert*  
*\**  
  
 *"ADD ckn24062020 - Start*  
 IF gt\_vert\_h IS NOT INITIAL.  
 SORT gt\_vert\_h BY ebeln feld timestamp DESCENDING.  
 DELETE ADJACENT DUPLICATES FROM gt\_vert\_h COMPARING ebeln feld.  
 INSERT /afm/bau\_vert\_h FROM TABLE gt\_vert\_h.  
 REFRESH gt\_vert\_h.  
 ENDIF.  
 *"ADD ckn24062020 - Ende*  
  
ENDFORM. *" datensicherung*

*\*&---------------------------------------------------------------------\**  
*\*& Form save\_text\_to\_table*  
*\*&---------------------------------------------------------------------\**  
*\* Text aus dem Textcontainer sichern*  
*\*----------------------------------------------------------------------\**  
*\* <--P\_WT\_TEXT text*  
*\*----------------------------------------------------------------------\**  
FORM save\_text\_to\_table CHANGING p\_wt\_text.  
  
 CLEAR: wt\_langtext,  
 wt\_langtext[],  
 wt\_text,  
 wt\_text[].  
  
*\* retrieve table from control*  
 CALL METHOD editor->get\_text\_as\_r3table  
 IMPORTING  
 table = p\_wt\_text  
*\* table = p\_my\_table*  
 EXCEPTIONS  
 OTHERS = 1.  
 IF sy-subrc NE 0.  
 CALL FUNCTION 'POPUP\_TO\_INFORM'  
 EXPORTING  
 titel = repid  
 txt2 = space  
 txt1 = TEXT-003.  
 ENDIF.  
  
*\* finally perform a flush to execute the calls in the control queue*  
 CALL METHOD cl\_gui\_cfw=>flush  
 EXCEPTIONS  
 OTHERS = 1.  
 IF sy-subrc NE 0.  
 CALL FUNCTION 'POPUP\_TO\_INFORM'  
 EXPORTING  
 titel = repid  
 txt2 = space  
 txt1 = TEXT-002.  
 ENDIF.  
  
  
 LOOP AT wt\_text INTO wa\_text.  
 MOVE: '/' TO wt\_langtext-tdformat,  
 wa\_text-line TO wt\_langtext-tdline.  
 APPEND wt\_langtext. CLEAR wt\_langtext.  
 ENDLOOP.  
  
  
*\* Langtext-Schlüssel*  
 CONCATENATE '/AFM/' bestellnr wf\_posid  
 INTO wf\_tdname\_lt.  
  
  
*\* Langtext (Vertragsgrundlage) sichern*  
 CLEAR thead.  
 MOVE: 'ST' TO thead-tdid,  
 'D' TO thead-tdspras,  
 wf\_tdname\_lt TO thead-tdname,  
 'TEXT' TO thead-tdobject,  
 '072' TO thead-tdlinesize.  
  
 CALL FUNCTION 'SAVE\_TEXT'  
 EXPORTING  
 client = sy-mandt  
 header = thead  
*\* INSERT = 'X'*  
 savemode\_direct = 'X'  
*\* OWNER\_SPECIFIED = ' '*  
*\* LOCAL\_CAT = ' '*  
*\* IMPORTING*  
*\* FUNCTION =*  
*\* NEWHEADER =*  
 TABLES  
 lines = wt\_langtext  
*\* EXCEPTIONS*  
*\* ID = 1*  
*\* LANGUAGE = 2*  
*\* NAME = 3*  
*\* OBJECT = 4*  
*\* OTHERS = 5*  
 .  
 IF sy-subrc <> 0.  
*\* message i000(SU) with ''.*  
 ENDIF.  
  
  
*\**  
*\* CALL FUNCTION 'EDIT\_TEXT'*  
*\* EXPORTING*  
*\*\* DISPLAY = ' '*  
*\*\* EDITOR\_TITLE = wf\_title*  
*\* HEADER = thead*  
*\*\* PAGE = ' '*  
*\* WINDOW = ' '*  
*\* SAVE = 'X'*  
*\* LINE\_EDITOR = ' '*  
*\*\* CONTROL = control*  
*\*\* PROGRAM = ' '*  
*\*\* LOCAL\_CAT = ' '*  
*\*\* IMPORTING*  
*\*\* FUNCTION =*  
*\*\* NEWHEADER =*  
*\*\* RESULT =*  
*\* TABLES*  
*\* LINES = wt\_langtext*  
*\*\* EXCEPTIONS*  
*\*\* ID = 1*  
*\*\* LANGUAGE = 2*  
*\*\* LINESIZE = 3*  
*\*\* NAME = 4*  
*\*\* OBJECT = 5*  
*\*\* TEXTFORMAT = 6*  
*\*\* COMMUNICATION = 7*  
*\*\* OTHERS = 8*  
*\* .*  
*\* IF SY-SUBRC <> 0.*  
*\*\* MESSAGE ID SY-MSGID TYPE SY-MSGTY NUMBER SY-MSGNO*  
*\*\* WITH SY-MSGV1 SY-MSGV2 SY-MSGV3 SY-MSGV4.*  
*\* ENDIF.*  
  
  
  
  
ENDFORM. *" save\_text\_to\_table*

*\*&---------------------------------------------------------------------\**  
*\*& Form COMPARE\_SAVE*  
*\*&---------------------------------------------------------------------\**  
*\* text*  
*\*----------------------------------------------------------------------\**  
*\* <--P\_LV\_ZERO text*  
*\*----------------------------------------------------------------------\**  
FORM compare\_save CHANGING cv\_zero TYPE xfeld.  
  
 DATA: ls\_save TYPE /afm/bauproj\_s,  
 lt\_save TYPE TABLE OF /afm/bauproj\_s.  
  
 REFRESH: lt\_save.  
  
 ls\_save-ebeln = wa\_bauvertr\_act-ebeln.  
*\* GET TIME STAMP FIELD ls\_save-timestamp.*  
 ls\_save-change\_user = sy-uname.  
 ls\_save-change\_date = sy-datum.  
 ls\_save-change\_time = sy-uzeit.  
  
*"DEL ckn20210817 - Start - PSPNR auch nicht auswerten*  
*\* IF wa\_bauproj\_comp-pspnr IS NOT INITIAL AND wa\_bauvertr\_act-pspnr IS INITIAL.*  
*\* cv\_zero = 'X'.*  
*\* ls\_save-nulled\_field = 'PSPNR'.*  
*\* GET TIME STAMP FIELD ls\_save-timestamp.*  
*\* APPEND ls\_save TO lt\_save.*  
*\* ENDIF.*  
*"DEL ckn20210817 - Ende*  
  
*\*\*Änderung durch Harshal*  
*\*\*Pflichtunterweisung Arbeitssicherheit und Energiemanagement -email subject KSG Hannover GmbH*  
*\*\*Kommentieren Sie die nicht benötigten Felder aus und setzen Sie den Wert des Feldes Null.*  
  
*\* IF wa\_bauproj\_comp-lifnr IS NOT INITIAL AND wa\_bauvertr\_act-lifnr IS INITIAL.*  
*\* cv\_zero = 'X'.*  
*\* ENDIF.*  
*\**  
*\* IF wa\_bauproj\_comp-kostl IS NOT INITIAL AND wa\_bauvertr\_act-kostl IS INITIAL.*  
*\* cv\_zero = 'X'.*  
*\* ENDIF.*  
*\**  
*\**  
*\* IF wa\_bauproj\_comp-gpreis\_netto IS NOT INITIAL AND wa\_bauvertr\_act-gpreis\_netto IS INITIAL.*  
*\* cv\_zero = 'X'.*  
*\* ENDIF.*  
*\**  
*\**  
*\* IF wa\_bauproj\_comp-mwst IS NOT INITIAL AND wa\_bauvertr\_act-mwst IS INITIAL.*  
*\* cv\_zero = 'X'.*  
*\* ENDIF.*  
  
*\**  
*\* IF wa\_bauproj\_comp-gpreis\_brutto IS NOT INITIAL AND wa\_bauvertr\_act-gpreis\_brutto IS INITIAL.*  
*\* cv\_zero = 'X'.*  
*\* ENDIF.*  
*\**  
*\**  
*\* IF wa\_bauproj\_comp-gpreis\_auf\_netto IS NOT INITIAL AND wa\_bauvertr\_act-gpreis\_auf\_netto IS INITIAL.*  
*\* cv\_zero = 'X'.*  
*\* ENDIF.*  
*\**  
*\**  
*\* IF wa\_bauproj\_comp-gpreis\_auf\_brutt IS NOT INITIAL AND wa\_bauvertr\_act-gpreis\_auf\_brutt IS INITIAL.*  
*\* cv\_zero = 'X'.*  
*\* ENDIF.*  
*\**  
*\**  
*\* IF wa\_bauproj\_comp-aufschlag IS NOT INITIAL AND wa\_bauvertr\_act-aufschlag IS INITIAL.*  
*\* cv\_zero = 'X'.*  
*\* ENDIF.*  
*\**  
*\**  
*\* IF wa\_bauproj\_comp-erfuell\_sich IS NOT INITIAL AND wa\_bauvertr\_act-erfuell\_sich IS INITIAL.*  
*\* cv\_zero = 'X'.*  
*\* ENDIF.*  
*\**  
*\**  
*\* IF wa\_bauproj\_comp-erf\_sich\_betr\_n IS NOT INITIAL AND wa\_bauvertr\_act-erf\_sich\_betr\_n IS INITIAL.*  
*\* cv\_zero = 'X'.*  
*\* ENDIF.*  
  
  
*\* IF wa\_bauproj\_comp-erf\_sich\_betr\_b IS NOT INITIAL AND wa\_bauvertr\_act-erf\_sich\_betr\_b IS INITIAL.*  
*\* cv\_zero = 'X'.*  
*\* ENDIF.*  
*\**  
*\**  
*\* IF wa\_bauproj\_comp-gewaehrleist IS NOT INITIAL AND wa\_bauvertr\_act-gewaehrleist IS INITIAL.*  
*\* cv\_zero = 'X'.*  
*\* ENDIF.*  
*\**  
*\* IF wa\_bauproj\_comp-gewaehrl\_betr\_n IS NOT INITIAL AND wa\_bauvertr\_act-gewaehrl\_betr\_n IS INITIAL.*  
*\* cv\_zero = 'X'.*  
*\* ENDIF.*  
*\**  
*\* IF wa\_bauproj\_comp-gewaehrl\_betr\_b IS NOT INITIAL AND wa\_bauvertr\_act-gewaehrl\_betr\_b IS INITIAL.*  
*\* cv\_zero = 'X'.*  
*\* ENDIF.*  
  
 IF wa\_bauproj\_comp-gewaehrl\_dauer IS NOT INITIAL AND wa\_bauvertr\_act-gewaehrl\_dauer IS INITIAL.  
 cv\_zero = 'X'.  
 ls\_save-nulled\_field = 'GEWAEHRL\_DAUER'.  
 GET TIME STAMP FIELD ls\_save-timestamp.  
 APPEND ls\_save TO lt\_save.  
 ENDIF.  
  
*\* IF wa\_bauproj\_comp-straf\_in\_proz IS NOT INITIAL AND wa\_bauvertr\_act-straf\_in\_proz IS INITIAL.*  
*\* cv\_zero = 'X'.*  
*\* ls\_save-nulled\_field = 'STRAF\_IN\_PROZ'.*  
*\* GET TIME STAMP FIELD ls\_save-timestamp.*  
*\* APPEND ls\_save TO lt\_save.*  
*\* ENDIF.*  
  
*\* IF wa\_bauproj\_comp-straf\_als\_betr IS NOT INITIAL AND wa\_bauvertr\_act-straf\_als\_betr IS INITIAL.*  
*\* cv\_zero = 'X'.*  
*\* ls\_save-nulled\_field = 'STRAF\_ALS\_BETR'.*  
*\* GET TIME STAMP FIELD ls\_save-timestamp.*  
*\* APPEND ls\_save TO lt\_save.*  
*\* ENDIF.*  
  
*\* IF wa\_bauproj\_comp-vertr\_straf IS NOT INITIAL AND wa\_bauvertr\_act-vertr\_straf IS INITIAL.*  
*\* cv\_zero = 'X'.*  
*\* ls\_save-nulled\_field = 'VERTR\_STRAF'.*  
*\* GET TIME STAMP FIELD ls\_save-timestamp.*  
*\* APPEND ls\_save TO lt\_save.*  
*\* ENDIF.*  
  
*\* IF wa\_bauproj\_comp-vertr\_straf\_betr IS NOT INITIAL AND wa\_bauvertr\_act-vertr\_straf\_betr IS INITIAL.*  
*\* cv\_zero = 'X'.*  
*\* ls\_save-nulled\_field = 'VERTR\_STRAF\_BETR'.*  
*\* APPEND ls\_save TO lt\_save.*  
*\* ENDIF.*  
  
*\*\*\*hinzugefügt von harshal*  
  
*\* IF wa\_bauproj\_comp-straf\_in\_proz IS NOT INITIAL AND wa\_bauvertr\_act-straf\_in\_proz IS INITIAL.*  
*\* IF wa\_bauproj\_comp-vertr\_straf IS NOT INITIAL AND wa\_bauvertr\_act-vertr\_straf IS INITIAL.*  
*\* cv\_zero = 'X'.*  
*\* ls\_save-nulled\_field = 'STRAF\_IN\_PROZ'.*  
*\* GET TIME STAMP FIELD ls\_save-timestamp.*  
*\* APPEND ls\_save TO lt\_save.*  
*\* ENDIF.*  
*\* ENDIF.*  
  
*\* IF wa\_bauproj\_comp-straf\_als\_betr IS NOT INITIAL AND wa\_bauvertr\_act-straf\_als\_betr IS INITIAL.*  
*\* IF wa\_bauproj\_comp-vertr\_straf IS NOT INITIAL AND wa\_bauvertr\_act-vertr\_straf IS INITIAL and*  
*\* wa\_bauproj\_comp-vertr\_straf\_betr IS NOT INITIAL AND wa\_bauvertr\_act-vertr\_straf\_betr IS INITIAL.*  
*\* cv\_zero = 'X'.*  
*\* ls\_save-nulled\_field = 'VERTR\_STRAF\_BETR'.*  
*\* GET TIME STAMP FIELD ls\_save-timestamp.*  
*\* APPEND ls\_save TO lt\_save.*  
*\* ENDIF.*  
*\* ENDIF.*  
 IF wa\_bauproj\_comp-vertr\_straf IS INITIAL AND wa\_bauvertr\_act-vertr\_straf IS INITIAL and  
 wa\_bauproj\_comp-vertr\_straf\_betr IS NOT INITIAL AND wa\_bauvertr\_act-vertr\_straf\_betr IS INITIAL.  
 cv\_zero = 'X'.  
 ls\_save-nulled\_field = 'VERTR\_STRAF\_BETR'.  
 GET TIME STAMP FIELD ls\_save-timestamp.  
 APPEND ls\_save TO lt\_save.  
 ENDIF.  
  
 IF wa\_bauproj\_comp-vertr\_straf IS not INITIAL AND wa\_bauvertr\_act-vertr\_straf IS INITIAL and  
 wa\_bauproj\_comp-vertr\_straf\_betr IS INITIAL AND wa\_bauvertr\_act-vertr\_straf\_betr IS INITIAL.  
 cv\_zero = 'X'.  
 ls\_save-nulled\_field = 'VERTR\_STRAF'.  
 GET TIME STAMP FIELD ls\_save-timestamp.  
 APPEND ls\_save TO lt\_save.  
 ENDIF.  
  
*\* IF wa\_bauproj\_comp-straf\_pro\_kal IS NOT INITIAL AND wa\_bauvertr\_act-straf\_pro\_kal IS INITIAL.*  
*\* cv\_zero = 'X'.*  
*\* ls\_save-nulled\_field = 'STRAF\_PRO\_KAL'.*  
*\* GET TIME STAMP FIELD ls\_save-timestamp.*  
*\* APPEND ls\_save TO lt\_save.*  
*\* ENDIF.*  
*\**  
*\* IF wa\_bauproj\_comp-straf\_pro\_werk IS NOT INITIAL AND wa\_bauvertr\_act-straf\_pro\_werk IS INITIAL.*  
*\* cv\_zero = 'X'.*  
*\* ls\_save-nulled\_field = 'STRAF\_PRO\_WERK'.*  
*\* GET TIME STAMP FIELD ls\_save-timestamp.*  
*\* APPEND ls\_save TO lt\_save.*  
*\* ENDIF.*  
  
 IF wa\_bauproj\_comp-straf\_nach IS NOT INITIAL AND wa\_bauvertr\_act-straf\_nach IS INITIAL.  
 cv\_zero = 'X'.  
 ls\_save-nulled\_field = 'STRAF\_NACH'.  
 GET TIME STAMP FIELD ls\_save-timestamp.  
 APPEND ls\_save TO lt\_save.  
 ENDIF.  
  
 IF wa\_bauproj\_comp-straf\_max IS NOT INITIAL AND wa\_bauvertr\_act-straf\_max IS INITIAL.  
 cv\_zero = 'X'.  
 ls\_save-nulled\_field = 'STRAF\_MAX'.  
 GET TIME STAMP FIELD ls\_save-timestamp.  
 APPEND ls\_save TO lt\_save.  
 ENDIF.  
  
*\* IF wa\_bauproj\_comp-bauleist\_vers IS NOT INITIAL AND wa\_bauvertr\_act-bauleist\_vers IS INITIAL.*  
*\* cv\_zero = 'X'.*  
*\* ls\_save-nulled\_field = 'PSPNR'.*  
*\* APPEND ls\_save TO lt\_save.*  
*\* ENDIF.*  
  
*\* IF wa\_bauproj\_comp-bauleist\_vers\_mwst IS NOT INITIAL AND wa\_bauvertr\_act-bauleist\_vers\_mwst IS INITIAL.*  
*\* cv\_zero = 'X'.*  
*\* ENDIF.*  
*\**  
*\* IF wa\_bauproj\_comp-bauleist\_betr\_n IS NOT INITIAL AND wa\_bauvertr\_act-bauleist\_betr\_n IS INITIAL.*  
*\* cv\_zero = 'X'.*  
*\* ENDIF.*  
*\**  
*\* IF wa\_bauproj\_comp-bauleist\_betr\_b IS NOT INITIAL AND wa\_bauvertr\_act-bauleist\_betr\_b IS INITIAL.*  
*\* cv\_zero = 'X'.*  
*\* ENDIF.*  
*\**  
*\* IF wa\_bauproj\_comp-bauleist\_bez IS NOT INITIAL AND wa\_bauvertr\_act-bauleist\_bez IS INITIAL.*  
*\* cv\_zero = 'X'.*  
*\* ENDIF.*  
*\**  
*\* IF wa\_bauproj\_comp-bauleist\_als IS NOT INITIAL AND wa\_bauvertr\_act-bauleist\_als IS INITIAL.*  
*\* cv\_zero = 'X'.*  
*\* ENDIF.*  
*\**  
*\* IF wa\_bauproj\_comp-kostenumlage IS NOT INITIAL AND wa\_bauvertr\_act-kostenumlage IS INITIAL.*  
*\* cv\_zero = 'X'.*  
*\* ENDIF.*  
*\**  
*\* IF wa\_bauproj\_comp-kostenumlage\_mwst IS NOT INITIAL AND wa\_bauvertr\_act-kostenumlage\_mwst IS INITIAL.*  
*\* cv\_zero = 'X'.*  
*\* ENDIF.*  
*\**  
*\* IF wa\_bauproj\_comp-kostenum\_betr\_n IS NOT INITIAL AND wa\_bauvertr\_act-kostenum\_betr\_n IS INITIAL.*  
*\* cv\_zero = 'X'.*  
*\* ENDIF.*  
*\**  
*\* IF wa\_bauproj\_comp-kostenum\_betr\_b IS NOT INITIAL AND wa\_bauvertr\_act-kostenum\_betr\_b IS INITIAL.*  
*\* cv\_zero = 'X'.*  
*\* ENDIF.*  
*\**  
*\* IF wa\_bauproj\_comp-kostenum\_bez IS NOT INITIAL AND wa\_bauvertr\_act-kostenum\_bez IS INITIAL.*  
*\* cv\_zero = 'X'.*  
*\* ENDIF.*  
*\**  
*\* IF wa\_bauproj\_comp-kostenum\_als IS NOT INITIAL AND wa\_bauvertr\_act-kostenum\_als IS INITIAL.*  
*\* cv\_zero = 'X'.*  
*\* ENDIF.*  
*\**  
*\* IF wa\_bauproj\_comp-reinigung IS NOT INITIAL AND wa\_bauvertr\_act-reinigung IS INITIAL.*  
*\* cv\_zero = 'X'.*  
*\* ENDIF.*  
*\**  
*\* IF wa\_bauproj\_comp-reinigung\_mwst IS NOT INITIAL AND wa\_bauvertr\_act-reinigung\_mwst IS INITIAL.*  
*\* cv\_zero = 'X'.*  
*\* ENDIF.*  
*\**  
*\* IF wa\_bauproj\_comp-rein\_betr\_n IS NOT INITIAL AND wa\_bauvertr\_act-rein\_betr\_n IS INITIAL.*  
*\* cv\_zero = 'X'.*  
*\* ENDIF.*  
*\**  
*\* IF wa\_bauproj\_comp-rein\_betr\_b IS NOT INITIAL AND wa\_bauvertr\_act-rein\_betr\_b IS INITIAL.*  
*\* cv\_zero = 'X'.*  
*\* ENDIF.*  
*\**  
*\* IF wa\_bauproj\_comp-reinigung\_bez IS NOT INITIAL AND wa\_bauvertr\_act-reinigung\_bez IS INITIAL.*  
*\* cv\_zero = 'X'.*  
*\* ENDIF.*  
*\**  
*\* IF wa\_bauproj\_comp-reinigung\_als IS NOT INITIAL AND wa\_bauvertr\_act-reinigung\_als IS INITIAL.*  
*\* cv\_zero = 'X'.*  
*\* ENDIF.*  
*\**  
*\* IF wa\_bauproj\_comp-anz\_re IS NOT INITIAL AND wa\_bauvertr\_act-anz\_re IS INITIAL.*  
*\* cv\_zero = 'X'.*  
*\* ENDIF.*  
*\**  
*\* IF wa\_bauproj\_comp-adresse IS NOT INITIAL AND wa\_bauvertr\_act-adresse IS INITIAL.*  
*\* cv\_zero = 'X'.*  
*\* ENDIF.*  
*\**  
*\* IF wa\_bauproj\_comp-faell\_acontore IS NOT INITIAL AND wa\_bauvertr\_act-faell\_acontore IS INITIAL.*  
*\* cv\_zero = 'X'.*  
*\* ENDIF.*  
*\**  
*\* IF wa\_bauproj\_comp-faell\_schlussre IS NOT INITIAL AND wa\_bauvertr\_act-faell\_schlussre IS INITIAL.*  
*\* cv\_zero = 'X'.*  
*\* ENDIF.*  
*\**  
*\* IF wa\_bauproj\_comp-vor\_buerg\_ein IS NOT INITIAL AND wa\_bauvertr\_act-vor\_buerg\_ein IS INITIAL.*  
*\* cv\_zero = 'X'.*  
*\* ENDIF.*  
*\**  
*\* IF wa\_bauproj\_comp-vor\_buerg\_aus IS NOT INITIAL AND wa\_bauvertr\_act-vor\_buerg\_aus IS INITIAL.*  
*\* cv\_zero = 'X'.*  
*\* ENDIF.*  
*\**  
*\* IF wa\_bauproj\_comp-vor\_bem IS NOT INITIAL AND wa\_bauvertr\_act-vor\_bem IS INITIAL.*  
*\* cv\_zero = 'X'.*  
*\* ENDIF.*  
*\**  
*\* IF wa\_bauproj\_comp-vor\_betrag IS NOT INITIAL AND wa\_bauvertr\_act-vor\_betrag IS INITIAL.*  
*\* cv\_zero = 'X'.*  
*\* ENDIF.*  
*\**  
*\* IF wa\_bauproj\_comp-erf\_buerg\_ein IS NOT INITIAL AND wa\_bauvertr\_act-erf\_buerg\_ein IS INITIAL.*  
*\* cv\_zero = 'X'.*  
*\* ENDIF.*  
*\**  
*\* IF wa\_bauproj\_comp-erf\_buerg\_aus IS NOT INITIAL AND wa\_bauvertr\_act-erf\_buerg\_aus IS INITIAL.*  
*\* cv\_zero = 'X'.*  
*\* ENDIF.*  
*\**  
*\* IF wa\_bauproj\_comp-erf\_bem IS NOT INITIAL AND wa\_bauvertr\_act-erf\_bem IS INITIAL.*  
*\* cv\_zero = 'X'.*  
*\* ENDIF.*  
*\**  
*\* IF wa\_bauproj\_comp-erf\_betrag IS NOT INITIAL AND wa\_bauvertr\_act-erf\_betrag IS INITIAL.*  
*\* cv\_zero = 'X'.*  
*\* ENDIF.*  
*\**  
*\* IF wa\_bauproj\_comp-gew\_sr\_am IS NOT INITIAL AND wa\_bauvertr\_act-gew\_sr\_am IS INITIAL.*  
*\* cv\_zero = 'X'.*  
*\* ENDIF.*  
*\**  
*\* IF wa\_bauproj\_comp-gew\_abn\_am IS NOT INITIAL AND wa\_bauvertr\_act-gew\_abn\_am IS INITIAL.*  
*\* cv\_zero = 'X'.*  
*\* ENDIF.*  
*\**  
*\* IF wa\_bauproj\_comp-gew\_maeng IS NOT INITIAL AND wa\_bauvertr\_act-gew\_maeng IS INITIAL.*  
*\* cv\_zero = 'X'.*  
*\* ENDIF.*  
*\**  
*\* IF wa\_bauproj\_comp-gew\_frist IS NOT INITIAL AND wa\_bauvertr\_act-gew\_frist IS INITIAL.*  
*\* cv\_zero = 'X'.*  
*\* ENDIF.*  
*\**  
*\* IF wa\_bauproj\_comp-gew\_nachweis IS NOT INITIAL AND wa\_bauvertr\_act-gew\_nachweis IS INITIAL.*  
*\* cv\_zero = 'X'.*  
*\* ENDIF.*  
*\**  
*\* IF wa\_bauproj\_comp-gew\_bem IS NOT INITIAL AND wa\_bauvertr\_act-gew\_bem IS INITIAL.*  
*\* cv\_zero = 'X'.*  
*\* ENDIF.*  
*\**  
*\* IF wa\_bauproj\_comp-gew\_buerg\_ein IS NOT INITIAL AND wa\_bauvertr\_act-gew\_buerg\_ein IS INITIAL.*  
*\* cv\_zero = 'X'.*  
*\* ENDIF.*  
*\**  
*\* IF wa\_bauproj\_comp-gew\_buerg\_aus IS NOT INITIAL AND wa\_bauvertr\_act-gew\_buerg\_aus IS INITIAL.*  
*\* cv\_zero = 'X'.*  
*\* ENDIF.*  
*\**  
*\* IF wa\_bauproj\_comp-gew\_buerg\_bem IS NOT INITIAL AND wa\_bauvertr\_act-gew\_buerg\_bem IS INITIAL.*  
*\* cv\_zero = 'X'.*  
*\* ENDIF.*  
*\**  
*\* IF wa\_bauproj\_comp-gew\_betrag IS NOT INITIAL AND wa\_bauvertr\_act-gew\_betrag IS INITIAL.*  
*\* cv\_zero = 'X'.*  
*\* ENDIF.*  
*\**  
*\* IF wa\_bauproj\_comp-waers IS NOT INITIAL AND wa\_bauvertr\_act-waers IS INITIAL.*  
*\* cv\_zero = 'X'.*  
*\* ENDIF.*  
*\**  
*\* IF wa\_bauproj\_comp-anlnr IS NOT INITIAL AND wa\_bauvertr\_act-anlnr IS INITIAL.*  
*\* cv\_zero = 'X'.*  
*\* ENDIF.*  
*\**  
*\* IF wa\_bauproj\_comp-bukrs IS NOT INITIAL AND wa\_bauvertr\_act-bukrs IS INITIAL.*  
*\* cv\_zero = 'X'.*  
*\* ENDIF.*  
*\**  
*\* IF wa\_bauproj\_comp-gew\_eb\_kein IS NOT INITIAL AND wa\_bauvertr\_act-gew\_eb\_kein IS INITIAL.*  
*\* cv\_zero = 'X'.*  
*\* ENDIF.*  
*\**  
*\* IF wa\_bauproj\_comp-gew\_eb\_teilw IS NOT INITIAL AND wa\_bauvertr\_act-gew\_eb\_teilw IS INITIAL.*  
*\* cv\_zero = 'X'.*  
*\* ENDIF.*  
*\**  
*\* IF wa\_bauproj\_comp-gew\_eb\_voll IS NOT INITIAL AND wa\_bauvertr\_act-gew\_eb\_voll IS INITIAL.*  
*\* cv\_zero = 'X'.*  
*\* ENDIF.*  
*\**  
*\* IF wa\_bauproj\_comp-gew\_eb\_teil\_betr IS NOT INITIAL AND wa\_bauvertr\_act-gew\_eb\_teil\_betr IS INITIAL.*  
*\* cv\_zero = 'X'.*  
*\* ENDIF.*  
  
 IF lt\_save IS NOT INITIAL.  
*\* INSERT /afm/bauproj\_s FROM TABLE lt\_save.*  
 MODIFY /afm/bauproj\_s FROM TABLE lt\_save.  
 ENDIF.  
  
ENDFORM.

*\*&---------------------------------------------------------------------\**  
*\*& Form SAVE\_DOCUMENT*  
*\*&---------------------------------------------------------------------\**  
*\* -->P\_GO\_DOCUMENT\_B text*  
*\* -->P\_GV\_DOC\_OPEN\_B text*  
*\* -->P\_LV\_COMP\_ID text*  
*\* -->P\_C\_CLASS\_B text*  
*\* -->P\_GV\_EXIST\_B text*  
*\* -->P\_GV\_KZ\_VORL\_B text*  
*\* -->P\_0669 text*  
*\*----------------------------------------------------------------------\**  
 FORM save\_document USING io\_document TYPE REF TO i\_oi\_document\_proxy  
 iv\_doc\_open  
 iv\_comp\_id  
 ic\_class  
 iv\_exist  
 iv\_kz\_vorl  
 iv\_art  
 iv\_source  
 iv\_ebeln.  
  
  
 TYPES: tyt\_errors TYPE STANDARD TABLE OF REF TO i\_oi\_error  
 WITH NON-UNIQUE DEFAULT KEY.  
  
 DATA lt\_data TYPE STANDARD TABLE OF bapiconten.  
  
*\*----------------------------------------------------------------------\**  
  
 IF NOT io\_document IS INITIAL.  
 REFRESH lt\_data.  
*\* Aktuellen Inhalt holen*  
 CALL METHOD io\_document->save\_document\_to\_table  
 CHANGING  
 document\_size = lv\_doc\_size  
 document\_table = lt\_data.  
 ENDIF.  
  
 IF iv\_doc\_open EQ 'X'.  
  
 CASE iv\_kz\_vorl.  
  
 WHEN 'X'.  
 CLEAR: ls\_component,  
 ls\_signature.  
 REFRESH: lt\_components,  
 lt\_signature.  
  
 ls\_signature-prop\_name = 'BDS\_DOCUMENTTYPE'.  
 ls\_signature-prop\_value = 'BDS\_TEXT'.  
 ls\_signature-doc\_count = '00000001'.  
 ls\_signature-doc\_ver\_no = '00000001'.  
 ls\_signature-doc\_var\_id = '00000001'.  
 ls\_signature-comp\_count = '00000001'.  
 APPEND ls\_signature TO lt\_signature.  
  
 ls\_signature-prop\_name = 'BDS\_DOCUMENTCLASS'.  
 ls\_signature-prop\_value = 'DOC'.  
 APPEND ls\_signature TO lt\_signature.  
  
 ls\_signature-prop\_name = 'LANGUAGE'.  
 ls\_signature-prop\_value = sy-langu.  
 APPEND ls\_signature TO lt\_signature.  
  
 CASE iv\_art.  
 WHEN 'N'. *"mehrere je Bestellung möglich*  
  
*\* Attribute BDS\_STATE und BDS\_KEYWORD können frei verwendet werden*  
  
 ls\_signature-prop\_name = 'DESCRIPTION'.  
  
*\* CONCATENATE gv\_objectkey '\_NA zu' gv\_sel\_pos\_na INTO ls\_signature-prop\_value.*  
 CONCATENATE bestellnr '\_NA\_' gv\_sel\_pos\_na INTO ls\_signature-prop\_value.  
*\* ls\_signature-prop\_value = gv\_objectkey.*  
 APPEND ls\_signature TO lt\_signature.  
  
*\* später entscheiden, ob evtl. nur BDS\_STATE genutzt wird, also ohne Schlagworte*  
 ls\_signature-prop\_name = 'BDS\_STATE'. *"genutzt werden soll*  
 ls\_signature-prop\_value = gv\_sel\_pos\_na.  
 APPEND ls\_signature TO lt\_signature.  
  
 CONCATENATE iv\_comp\_id '.doc' INTO ls\_component-comp\_id.  
 ls\_component-mimetype = 'application/vnd.ms-word'.  
 ls\_component-comp\_size = lv\_doc\_size.  
 ls\_component-doc\_count = '00000001'.  
 ls\_component-comp\_count = '00000001'.  
 APPEND ls\_component TO lt\_components.  
  
 WHEN OTHERS. *"Beauftragung*  
  
*\* bzgl.Kleinem Bauunterhalt (Extradruck aus separater Transaktion*  
 IF iv\_source = 'KL\_BAU'.  
 gv\_objectkey = iv\_ebeln.  
 ENDIF.  
  
 ls\_signature-prop\_name = 'DESCRIPTION'.  
 ls\_signature-prop\_value = gv\_objectkey.  
 APPEND ls\_signature TO lt\_signature.  
  
 ls\_component-comp\_id = iv\_comp\_id.  
 ls\_component-mimetype = 'application/vnd.ms-word'.  
 ls\_component-comp\_size = lv\_doc\_size.  
 ls\_component-doc\_count = '00000001'.  
 ls\_component-comp\_count = '00000001'.  
 APPEND ls\_component TO lt\_components.  
  
 ENDCASE.  
  
 CALL FUNCTION 'BDS\_BUSINESSDOCUMENT\_CREA\_TAB'  
 EXPORTING  
*\* LOGICAL\_SYSTEM =*  
 classname = ic\_class  
 classtype = c\_cltyp  
 client = sy-mandt  
 object\_key = gv\_objectkey  
*\* BINARY\_FLAG = 'X'*  
*\* IMPORTING*  
*\* OBJECT\_KEY =*  
 TABLES  
 signature = lt\_signature  
 components = lt\_components  
 content = lt\_data  
*\* ASCII\_CONTENT =*  
 EXCEPTIONS  
 nothing\_found = 1  
 parameter\_error = 2  
 not\_allowed = 3  
 error\_kpro = 4  
 internal\_error = 5  
 not\_authorized = 6  
 OTHERS = 7  
 .  
 IF sy-subrc <> 0.  
 MESSAGE ID sy-msgid TYPE sy-msgty NUMBER sy-msgno  
 WITH sy-msgv1 sy-msgv2 sy-msgv3 sy-msgv4.  
 ELSE.  
 MOVE lt\_signature[] TO gt\_signatures[]. *"nach der Erstellung des Dok.die Daten zur Vorlage mit den aktuellen Daten überschreiben*  
 MOVE lt\_components[] TO gt\_components[].  
  
 CLEAR iv\_kz\_vorl.  
 iv\_exist = 'X'.  
  
 DATA lv\_text(51) TYPE c.  
 CASE iv\_art.  
 WHEN 'B'.  
 IF gv\_vorlage = c\_class\_b.  
 MOVE text-399 TO lv\_text. *"PE Bestellung pauschal aus Vorlage heraus gesichert*  
 ELSEIF gv\_vorlage = c\_class\_tim.  
 MOVE text-400 TO lv\_text. *"TIM Bestellung VOB aus Vorlage heraus gesichert*  
 ENDIF.  
 WHEN 'N'.  
 MOVE text-401 TO lv\_text. *"Nachtrag zur Bestellung aus Vorlage heraus gesichert*  
 gv\_objectkey+11(2) = gv\_sel\_pos\_na+3(2). *" + Posnr 0020 z.B.*  
 ENDCASE.  
  
*\* 'Beauftragungsschreiben aus Vorlage heraus gesichert:'*  
 CALL FUNCTION 'POPUP\_TO\_INFORM'  
 EXPORTING  
 titel = ''  
 txt1 = lv\_text  
 txt2 = gv\_objectkey.  
 CLEAR gv\_objectkey+11(5).  
 ENDIF.  
  
 WHEN OTHERS. *" ==> Update des bereits bestehenden Dokumentes*  
*\*\*\* ACHTUNG: Nachtragsschreiben 'N' können mehrfach vorkommen zu einer Bestellung*  
*\*\*\* ==> hier den Objektschlüssel mit fortlaufender Nummer versehen*  
*\*\*\* oder besser über Schlagwort und Namen differenzieren, damit alle Nachträge.*  
*\*\*\* weiterhin unter einem Objektschlüssel, d.h. einer Abschlagskettenbestellung zu finden sind*  
 IF iv\_art EQ 'N'.  
 READ TABLE gt\_signatures INTO ls\_signature INDEX 1.  
 MOVE ls\_signature-doc\_id TO lv\_doc\_id.  
 MOVE ls\_signature-doc\_count TO lv\_doc\_count.  
  
 CALL METHOD cl\_bds\_document\_set=>update\_with\_table  
 EXPORTING  
*\* logical\_system =*  
 classname = ic\_class  
 classtype = c\_cltyp  
*\* client =*  
 doc\_id = ls\_signature-doc\_id  
 doc\_ver\_no = ls\_signature-doc\_ver\_no  
 doc\_var\_id = ls\_signature-doc\_var\_id  
 object\_key = gv\_objectkey  
*\* x\_force\_update =*  
 CHANGING  
 components = gt\_components *"it\_components*  
 content = lt\_data  
 signature = gt\_signatures  
*\* properties =*  
 EXCEPTIONS  
 nothing\_found = 1  
 internal\_error = 2  
 error\_kpro = 3  
 parameter\_error = 4  
 not\_authorized = 5  
 not\_allowed = 6  
 OTHERS = 7.  
 IF sy-subrc <> 0.  
 MESSAGE ID sy-msgid TYPE sy-msgty NUMBER sy-msgno  
 WITH sy-msgv1 sy-msgv2 sy-msgv3 sy-msgv4.  
 ELSE.  
 CALL FUNCTION 'POPUP\_TO\_INFORM'  
 EXPORTING  
 titel = ''  
 txt1 = text-402 *"Nachtrag zur Bestellung wurde gesichert*  
 txt2 = gv\_objectkey.  
 ENDIF.  
 ELSE. *"'B'*  
*\* bzgl.Kleinem Bauunterhalt (Extradruck aus separater Transaktion*  
 IF iv\_source = 'KL\_BAU'.  
 gv\_objectkey = iv\_ebeln.  
 ENDIF.  
  
 READ TABLE gt\_signatures INDEX 1 INTO ls\_signature.  
  
 CALL METHOD cl\_bds\_document\_set=>update\_with\_table  
 EXPORTING  
*\* logical\_system =*  
 classname = ic\_class  
 classtype = c\_cltyp  
*\* client =*  
 doc\_id = ls\_signature-doc\_id  
 doc\_ver\_no = ls\_signature-doc\_ver\_no  
 doc\_var\_id = ls\_signature-doc\_var\_id  
 object\_key = gv\_objectkey  
 x\_force\_update = 'X'  
 CHANGING  
 components = gt\_components *"it\_components*  
 content = lt\_data  
 signature = gt\_signatures  
*\* properties =*  
 EXCEPTIONS  
 nothing\_found = 1  
 internal\_error = 2  
 error\_kpro = 3  
 parameter\_error = 4  
 not\_authorized = 5  
 not\_allowed = 6  
 OTHERS = 7.  
 IF sy-subrc <> 0.  
 MESSAGE ID sy-msgid TYPE sy-msgty NUMBER sy-msgno  
 WITH sy-msgv1 sy-msgv2 sy-msgv3 sy-msgv4.  
 ELSE.  
 IF gv\_vorlage = c\_class\_b.  
 CALL FUNCTION 'POPUP\_TO\_INFORM'  
 EXPORTING  
 titel = ''  
 txt1 = text-403 *"PE Bestellung pauschal wurde gesichert*  
 txt2 = gv\_objectkey.  
 ELSEIF gv\_vorlage = c\_class\_tim.  
 CALL FUNCTION 'POPUP\_TO\_INFORM'  
 EXPORTING  
 titel = ''  
 txt1 = text-404 *"TIM Bestellung VOB wurde gesichert*  
 txt2 = gv\_objectkey.  
 ENDIF.  
 ENDIF.  
 ENDIF.  
  
 ENDCASE.  
  
 ENDIF. *" iv\_doc\_open EQ 'X'*  
  
 ENDFORM. *" SAVE\_DOCUMENT*

*Include 🡪 /AFM/BAUMOD\_F00*

*\*&---------------------------------------------------------------------\**  
*\*& Form dequeue*  
*\*&---------------------------------------------------------------------\**  
*\* Tabellen entsperren*  
*\*----------------------------------------------------------------------\**  
*\* -->P\_BESTELLNR text*  
*\* -->P\_PSP text*  
*\*----------------------------------------------------------------------\**  
FORM dequeue USING p\_bestellnr.  
 *" P\_PSP.*  
  
 CALL FUNCTION 'DEQUEUE\_/AFM/BAUMOD'  
 EXPORTING  
*\* MODE\_/AFM/BAUPROJ = 'E'*  
 mandt = sy-mandt  
*\* PSPNR = p\_psp*  
 ebeln = p\_bestellnr  
*\* X\_PSPNR = ' '*  
*\* X\_EBELN = ' '*  
*\* \_SCOPE = '3'*  
*\* \_SYNCHRON = ' '*  
*\* \_COLLECT = ' '*  
 .  
  
  
 IF sy-subrc NE 0.  
 MESSAGE w000(su) WITH 'Nicht alle zugehörigen Tabellen konnten'  
 'entsperrt werden!'.  
 ENDIF.  
  
ENDFORM. *" dequeue*

*Include🡪 /AFM/BAUMOD\_F00*

*\*&---------------------------------------------------------------------\**  
*\*& Form exit*  
*\*&---------------------------------------------------------------------\**  
FORM free\_objects.  
  
*\* Abbau Control Texteditor.*  
 IF NOT editor IS INITIAL.  
 CALL METHOD editor->free  
 EXCEPTIONS  
 OTHERS = 1.  
 IF sy-subrc NE 0.  
 CALL FUNCTION 'POPUP\_TO\_INFORM'  
 EXPORTING  
 titel = repid  
 txt2 = space  
 txt1 = TEXT-005.  
 ENDIF.  
*\* free ABAP object also*  
 FREE editor.  
 ENDIF.  
  
*\* Abbau Container zum textcontrol*  
 IF NOT textedit\_custom\_container IS INITIAL.  
 CALL METHOD textedit\_custom\_container->free  
 EXCEPTIONS  
 OTHERS = 1.  
 IF sy-subrc <> 0.  
*\* MESSAGE E002 WITH F\_RETURN.*  
 ENDIF.  
*\* free ABAP object also*  
 FREE textedit\_custom\_container.  
 ENDIF.  
  
  
*\* Dokumentenverwaltungs-Instanzen abbauen*  
 IF gv\_doc\_open\_b EQ 'X'.  
 PERFORM free\_objects\_bds USING go\_container\_b  
 go\_document\_b  
 go\_control\_b  
 go\_worddoc\_b.  
*\* IF NOT go\_document\_b IS INITIAL.*  
*\* CALL METHOD GO\_DOCUMENT\_B->CLOSE\_DOCUMENT.*  
*\* ENDIF.*  
*\**  
*\**  
*\* IF NOT go\_control\_b IS INITIAL.*  
*\* CALL METHOD go\_control\_b->destroy\_control*  
*\* EXPORTING*  
*\* no\_flush = ' '.*  
*\* ENDIF.*  
*\**  
*\* FREE go\_worddoc\_b.*  
*\* CLEAR go\_worddoc\_b.*  
*\**  
*\* IF NOT go\_document\_b IS INITIAL.*  
*\* CALL METHOD go\_document\_b->release\_document.*  
*\* ENDIF.*  
*\**  
*\* FREE: go\_document\_b,*  
*\* go\_control\_b.*  
*\**  
*\* CLEAR: go\_document\_b,*  
*\* go\_control\_b.*  
*\**  
*\**  
*\* IF NOT go\_container\_b IS INITIAL.*  
*\* CALL METHOD go\_container\_b->free*  
*\* EXCEPTIONS*  
*\* cntl\_error = 1*  
*\* cntl\_system\_error = 2*  
*\* OTHERS = 3*  
*\* .*  
*\* if sy-subrc <> 0.*  
*\* MESSAGE ID sy-msgid TYPE sy-msgty NUMBER sy-msgno*  
*\* WITH sy-msgv1 sy-msgv2 sy-msgv3 sy-msgv4.*  
*\* endif.*  
*\**  
*\* FREE go\_container\_b.*  
*\* ENDIF.*  
 ENDIF.  
 IF gv\_doc\_open\_n EQ 'X'.  
 PERFORM free\_objects\_bds USING go\_container\_n  
 go\_document\_n  
 go\_control\_n  
 go\_worddoc\_n.  
 ENDIF.  
  
*\* Synchronisation der Automation Queue*  
 CALL METHOD cl\_gui\_cfw=>flush  
 EXCEPTIONS  
 OTHERS = 1.  
 IF sy-subrc NE 0.  
 MESSAGE i000(su) WITH 'Fehler bei Flush'.  
 ENDIF.  
  
ENDFORM. *" exit*

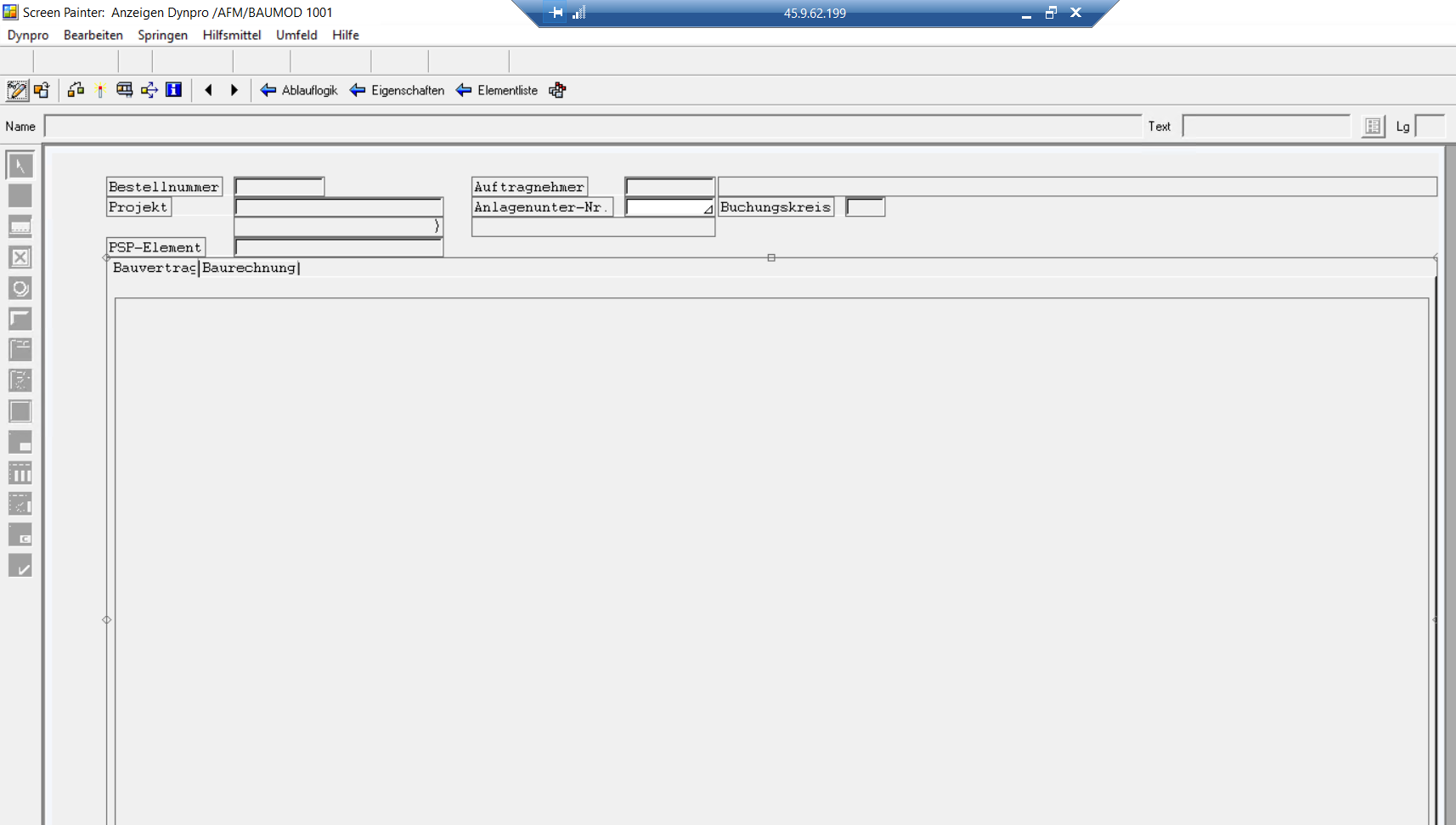
\*\*/AFM/BAUMOD\_F00  
*\*&---------------------------------------------------------------------\**  
*\*& Form read\_data*  
*\*&---------------------------------------------------------------------\**  
*\* Lesen der Standard-Defaultwerte bzw. der explizit zum Projekt \**  
*\* erfassten Defaultwerte aus Tab./AFM/BAU\_STAMMD. \**  
*\* Lesen der Daten aus der Bestellung (Gesamtleistungssumme, Kontierung \**  
*\*----------------------------------------------------------------------\**  
FORM read\_data.  
  
 DATA: h\_bestellnr(10) TYPE n,  
 lv\_pattern TYPE /afm/bau\_kstgrp\_muster, *"BEJK20190715 Anpassung Customizing Kostengruppen*  
 lv\_kstgrp TYPE /afm/kostengruppe\_n, *"BEJK20190715 Anpassung Customizing Kostengruppen*  
 lv\_stringlength TYPE i, *"BEJK20190715 Anpassung Customizing Kostengruppen*  
 lv\_index TYPE i, *"BEJK20190715 Anpassung Customizing Kostengruppen*  
 lv\_char TYPE c LENGTH 1. *"BEJK20190715 Anpassung Customizing Kostengruppen*  
  
*\*\*\*\*\*\* Ablauf nur beim ersten Aufruf \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\**  
 IF erster\_aufruf = 'X'.  
  
 MOVE bestellnr TO h\_bestellnr.  
 MOVE h\_bestellnr TO bestellnr.  
  
 CLEAR: treffer\_psp,  
 wf\_kbetr.  
  
*\* Daten aus der Steuerungstabelle holen*  
 SELECT SINGLE \* FROM /afm/baure\_steu INTO ws\_steuerung WHERE key\_entry EQ '1'.  
 IF sy-subrc NE 0.  
 MESSAGE w000(su) WITH 'Keinen Eintrag in /AFM/BAURE\_STEU gefunden. Bitte pflegen!'.  
*\* message w001(/AFM/BAU) with '/AFM/BAURE\_STEU'.*  
 ENDIF.  
  
*\* Konstanten aus der Tabelle /AFM/BAUMOD\_CONS ziehen*  
 SELECT SINGLE \* FROM /afm/baumod\_cons INTO ws\_const WHERE key\_entry EQ '1'.  
 IF sy-subrc NE 0.  
 MESSAGE w000(su) WITH 'Keinen Eintrag in /AFM\_BAUMOD\_CONS gefunden. Bitte pflegen!'.  
*\* message w001(/AFM/BAU) with '/AFM\_BAUMOD\_CONS'.*  
 ENDIF.  
  
*\* dient hier nur zur Vorbelegung im Dynpro. Wird in den Transaktionen aktuell aus ws\_cons übergeben*  
 blart = ws\_const-blart\_f43. *"Belegart*  
 blart\_gegenf = ws\_const-blart\_fb01. *"Belegart Gegenforderungen*  
 blart\_sich = ws\_const-blart\_f43. *"Belegart Einbehalte*  
 blart\_rekurz = ws\_const-blart\_miro\_guts.  
  
*\* Projekt aus der Bestellung ziehen. Erste Position stellt den Hauptauftrag dar*  
 DATA lv\_pos1 TYPE ebelp.  
 SELECT SINGLE MIN( ebelp ) FROM ekpo INTO lv\_pos1  
 WHERE ebeln EQ bestellnr  
 AND loekz EQ space.  
*\*\* hinzugefügt von Harshal*  
 CLEAR: ekkn, psp. *"ADD ckn/hrl 20210817 - Damit Bestellung\_alt nicht Bestellung\_neu 'überschreibt' weil nichts zu B2 in EKKN ist*  
 SELECT SINGLE \* FROM ekkn WHERE ebeln EQ bestellnr  
 AND ebelp EQ lv\_pos1.  
*\* and zekkn eq '1'.*  
*\* nicht mehr das PSP, sondern die Proejktdefinition ziehen (PSP kein Schlüssel mehr in den Abschlagskettentabellen*  
 SELECT SINGLE psphi FROM prps INTO psp WHERE pspnr EQ ekkn-ps\_psp\_pnr.  
  
*\* Sofern das Gebäude mit ins Dynpro gesetzt werden sollte (wie bei LBBW)*  
*\* Gebäude lesen aus zugehöriger führender Projektedition: ### UPD*  
*\* data: wf\_psphi type ps\_psphi,*  
*\* wf\_post1 type ps\_post1.*  
*\**  
*\* select single psphi from prps into wf\_psphi*  
*\* where pspnr eq psp.*  
*\* select single post1 from proj into wf\_post1*  
*\* where pspnr eq wf\_psphi.*  
*\* move wf\_post1 to gebaeude.*  
*\*} INSERT*  
  
*\* CALL FUNCTION 'CONVERSION\_EXIT\_ABPSP\_OUTPUT'*  
 CALL FUNCTION 'CONVERSION\_EXIT\_KONPD\_OUTPUT'  
 EXPORTING  
 input = psp *"NUMC 8*  
 IMPORTING  
 output = wf\_posid. *"CHAR 24*  
 CASE psp.  
 WHEN space.  
 SELECT SINGLE bukrs FROM ekko INTO bukrs  
 WHERE ebeln EQ bestellnr.  
 WHEN OTHERS.  
*\* Buchungskreis des Projektes ziehen*  
 SELECT SINGLE vbukr post1 FROM proj INTO (bukrs, psp\_bez)  
 WHERE pspnr EQ psp.  
 ENDCASE.  
  
*\* zwecks Ausgabe im Einstiegs- und Hauptdynpro erstes PSP ziehen*  
 CLEAR lv\_pos1.  
 SELECT SINGLE MIN( ebelp ) FROM ekpo INTO lv\_pos1  
 WHERE ebeln EQ bestellnr  
 AND loekz EQ space.  
 SELECT SINGLE ps\_psp\_pnr FROM ekkn INTO gv\_psp  
 WHERE ebeln EQ bestellnr  
 AND ebelp EQ lv\_pos1.  
 IF sy-subrc = 0. *"ADD ckn20210817 - Wenn GV\_PSP nicht gefunden wurde gibt es hier einen ShortDump!*  
*\* and zekkn eq '1'. "erstes Kontierungs-PSP (muß stets dasgleiche sein)*  
 CALL FUNCTION 'CONVERSION\_EXIT\_ABPSP\_OUTPUT'  
 EXPORTING  
 input = gv\_psp *"NUMC 8*  
 IMPORTING  
 output = gv\_psp. *"CHAR 24*  
 ELSE. *"ADD ckn20210817*  
 CLEAR: gv\_psp. *"ADD ckn20210817*  
 ENDIF. *"ADD ckn20210817*  
  
*\*START 20190715 BEJK Anpassung Customizing pro Kostengruppe*  
 lv\_stringlength = strlen( ws\_steuerung-kstgrp\_muster ).  
 WHILE lv\_index < lv\_stringlength.  
 lv\_char = ws\_steuerung-kstgrp\_muster+lv\_index(1).  
 IF lv\_char = '1'.  
 CONCATENATE gv\_psp+lv\_index(1) '00' INTO lv\_kstgrp.  
 EXIT.  
 ENDIF.  
 ADD 1 TO lv\_index.  
 ENDWHILE.  
*\*ENDE 20190715 BEJK*  
  
  
*\*\*\*\* Sollen auf bestimmte PSP´s und Stufe (z.B. nur Kontierung auf Gewerkeebene erlaubt) abgefragt werden,*  
*\*\*\*\* nachfolgendes Coding anpassen*  
  
*\*\* Prüfung der in der Bestellung als Kontierungselemente eingetragenen*  
*\*\* PSP-Elemente:*  
*\*\* Generell nur auf Gewerkeebene zulassen mit Ausnahme von Elemente wie folgende Bsp.:*  
*\*\* B2.1300/00/0000*  
*\*\* B2.1300/00/1111*  
*\*\* B2.1300/00/8888*  
*\*\* B2.1300/00/9999, also an der 11-ten Stelle 0000, 1111, 8888, 9999*  
  
*\* data: wf\_pspcheck type ps\_posid,*  
*\* wf\_stufe type ps\_stufe,*  
*\* wf\_ohne\_edition(24) type c,*  
*\* wf\_ebelp type ebelp,*  
*\* ls\_psp\_kont type /AFM/PSP\_KONTIER.*  
*\**  
*\* select \* from ekkn where ebeln eq bestellnr.*  
*\* move ekkn-ebelp to wf\_ebelp.*  
*\**  
*\* CALL FUNCTION 'CONVERSION\_EXIT\_ABPSP\_OUTPUT'*  
*\* EXPORTING*  
*\* INPUT = ekkn-ps\_psp\_pnr "NUMC 8*  
*\* IMPORTING*  
*\* OUTPUT = wf\_pspcheck. "CHAR 24*  
*\**  
*\**  
*\*\* Planungselemente P\* und W\* ausschliessen, Budget-Psp B\* zulassen*  
*\* case wf\_pspcheck+0(1).*  
*\* when 'B'.*  
*\* when others.*  
*\**  
*\* message w000(SU) with 'Es sind ausschließlich B\*-Projekte zugelassen!'*  
*\* 'Bitte erst die Bestellung in Pos.' wf\_ebelp 'ändern.'.*  
*\* leave to screen 1000.*  
*\* endcase.*  
*\**  
  
*\*\*\*\* case wf\_pspcheck+11(4).*  
*\*\*\*\* when '0000' or '1111' or '8888' or '9999'.*  
*\*\*\*\*\* => keine Prüfung auf die Ebene*  
*\*\*\*\**  
*\*\*\*\* when others. "Check, ob PSP auf Gewerkebene liegt (Level 3)*  
*\*\*\*\* select single stufe from prps into wf\_stufe*  
*\*\*\*\* where pspnr eq ekkn-ps\_psp\_pnr.*  
*\*\*\*\* if sy-subrc = 0.*  
*\*\*\*\* if wf\_stufe ne '4'.*  
*\*\*\*\**  
*\*\*\*\* CALL FUNCTION 'CONVERSION\_EXIT\_ABPSN\_INPUT'*  
*\*\*\*\* EXPORTING*  
*\*\*\*\* INPUT = wf\_pspcheck*  
*\*\*\*\* IMPORTING*  
*\*\*\*\* OUTPUT = wf\_ohne\_edition.*  
*\*\*\*\**  
*\*\*\*\**  
*\*\*\*\* select single \* from /AFM/PSP\_KONTIER into ls\_psp\_kont*  
*\*\*\*\* where posid eq wf\_ohne\_edition.*  
*\*\*\*\* if sy-subrc ne 0.*  
*\*\*\*\* message w000(SU) with 'Das Kontierungs-PSP in Pos.' wf\_ebelp*  
*\*\*\*\* 'liegt nicht auf Gewerkeebene u.ist nicht Inhalt' 'd./AFM/PSP\_KONTIER.Bitte erst d.Bestellung ändern!.'.*  
*\*\*\*\* leave to screen 1000.*  
*\*\*\*\* endif.*  
*\*\*\*\* endif.*  
*\*\*\*\* endif.*  
*\*\*\*\* endcase.*  
*\**  
*\* clear: wf\_pspcheck,*  
*\* wf\_stufe,*  
*\* wf\_ebelp.*  
*\* endselect.*  
*\*} INSERT*  
  
*\* Auftragnehmer aus der Bestellung ziehen*  
 DATA lv\_bukrs TYPE bukrs.  
 SELECT SINGLE lifnr bukrs FROM ekko INTO (kreditor, lv\_bukrs)  
 WHERE ebeln EQ bestellnr.  
 SELECT SINGLE \* FROM lfa1  
 WHERE lifnr EQ kreditor.  
 IF sy-subrc = 0.  
*\* Adresse des Kreditors -> für Subscreen 1022*  
 IF kreditor NE space.  
 CONCATENATE:  
 lfa1-pstlz lfa1-ort01 INTO plz\_ort SEPARATED BY ' ',  
 lfa1-name1 plz\_ort lfa1-stras INTO adresse  
 SEPARATED BY ', '.  
 MOVE adresse TO wf\_kred\_bez.  
*\* bzgl. Formularausgabe:*  
 MOVE: lfa1-name1 TO firma,  
 lfa1-stras TO stras.  
*\* Länderschlüssel zum Bukrs ziehen*  
*\* move lfa1-land1 to wf\_land.*  
 SELECT SINGLE land1 FROM t001 INTO wf\_land  
 WHERE bukrs EQ lv\_bukrs.  
 ELSE.  
 MESSAGE i000(su) WITH  
 'Auftragnehmer existiert nicht!'.  
 CLEAR: kreditor.  
 ENDIF.  
 ENDIF.  
  
*\* Test zwecks gleicher MWST-Auslesung, bisher wird für Gegenforderungen*  
*\* MWST-Satz aus Stammdatentab. gezogen, sollte aber auch aus der Bestellung kommen*  
  
*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\**  
*\* Daten zur Bestellung und Projekt ziehen, sofern bereits zur Be- \**  
*\* stellung Projektdaten gepflegt worden sind. \**  
*\* Tab. /AFM/BAUPROJ, /AFM/BAUPROJRE und /AFM/BAUPROJBEL \**  
*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\**  
  
*\* Bestellpositionen einlesen*  
 CLEAR: wt\_leist,  
 wt\_leist[],  
 wf\_kbetr.  
  
*\* Ermittlung der ersten Position der Bestellung, da nicht gewährleistet ist,*  
*\* daß diese immer mit 00010 beziffert ist.*  
 DATA wf\_pos1 TYPE ebelp.  
 CLEAR wf\_pos1.  
  
*\* ----------------------------------------------------------------------------------- "STM28072014*  
*\* Wurden die BAdI´s ME\_PROCESS\_PO\_CUST und ME\_GUI\_PO\_CUST umgesetzt, werden*  
*\* in die jeweiligen Bestellpositionen über ein Kundenfeld als Nachtragsposition*  
*\* zu einer Hauptposition zugeordnet und nachfolgend hier darüber erkannt.*  
*\* Sofern nicht umgesetzt, gilt weiterhin: Die 1.Position stellt den Hauptauftrag (HA)*  
*\* dar, ab der 2-ten Position handelt es sich um Nachträge (NA)*  
 DATA ls\_badi\_bau TYPE /afm/me\_badi\_bau.  
  
*\* Speichern der Belegkonditionsnummer KNUMV, um später aus EKPO den*  
*\* Gesamtleistungsnettowert nach Abzug der Rabatte auslesen zu können*  
 SELECT SINGLE knumv FROM ekko INTO wf\_knumv WHERE ebeln EQ bestellnr.  
  
 SELECT \* FROM ekpo WHERE ebeln EQ bestellnr  
 AND loekz EQ space.  
*\* Daten in die Itab wt\_leist zur Anzeige schreiben*  
 MOVE: ekpo-ebelp TO wt\_leist-ebelp,  
 ekpo-txz01 TO wt\_leist-txz01,  
 ekpo-brtwr TO wt\_leist-gpreis\_netto.  
 st\_kennz\_te = st\_kennz = ekpo-mwskz.  
 wt\_leist-mwskz = ekpo-mwskz.  
  
 CASE ws\_steuerung-incl\_badi\_nachtr.  
 WHEN 'X'.  
 SELECT SINGLE \* FROM /afm/me\_badi\_bau INTO ls\_badi\_bau  
 WHERE ebeln EQ ekpo-ebeln  
 AND ebelp EQ ekpo-ebelp  
 AND badi\_na\_zu\_ebelp > 0.  
 IF sy-subrc = 0.  
*\* => NA*  
 MOVE 'NA' TO wt\_leist-art.  
 MOVE ls\_badi\_bau-badi\_na\_zu\_ebelp TO wt\_leist-badi\_na\_zu\_ebelp.  
 MOVE ls\_badi\_bau-badi\_begdat TO wt\_leist-badi\_begdat.  
 ELSE.  
*\* => HA*  
 MOVE 'HA' TO wt\_leist-art.  
 ENDIF.  
  
 WHEN OTHERS.  
 SELECT MIN( ebelp ) FROM ekpo INTO wf\_pos1  
 WHERE ebeln EQ bestellnr  
 AND loekz EQ space.  
 IF ekpo-ebelp EQ wf\_pos1. *" 1.Pos. = Hauptauftrag*  
 MOVE 'HA' TO wt\_leist-art.  
 ELSEIF ekpo-ebelp GT wf\_pos1. *" alle folgenden Pos. sind Nachträge*  
 MOVE 'NA' TO wt\_leist-art.  
 ENDIF.  
 ENDCASE.  
  
*\* Nachlaß/Rabatt mit Konditionsart RA00 = Rabatt auf netto in den Kopfdaten*  
*\* => Gilt für alle Positionen, also auch später erfasste Nachlässe*  
*\* if ekpo-ebelp = wf\_pos1.*  
  
*\* Eintrag in den Kopfdaten => betrifft alle Positionen !!!*  
*\* Berücksichtigung prozentualer Rabatte RA00 und RA01: AUSSCHLIESSLICH einmalige Verwendung je Kondition*  
*\* Ein 2-ter Eintrag würde Wert verfälschen, da der zweite sich bereits auf verminderten Betrag durch die erste Kondition beziehen würde*  
*\* Ergänzung 'RB00' (absol.Betrag) wäre aufwendiger, da im Screen und dessen Vearbeitung ein prozentualer Wert erwartet wird.*  
 CLEAR aufschlag.  
 SELECT SINGLE \* FROM konv WHERE knumv EQ wf\_knumv  
 AND kschl = 'RA00'  
 AND kposn EQ ekpo-ebelp.  
 IF sy-subrc = 0.  
 aufschlag = aufschlag + ( konv-kbetr / 10 ).  
 MOVE aufschlag TO wt\_leist-aufschlag.  
 ENDIF.  
  
 SELECT SINGLE \* FROM konv WHERE knumv EQ wf\_knumv  
 AND kschl = 'RA01'  
 AND kposn EQ ekpo-ebelp.  
 IF sy-subrc = 0.  
 aufschlag = aufschlag + ( konv-kbetr / 10 ).  
 MOVE aufschlag TO wt\_leist-aufschlag.  
 ENDIF.  
  
*\* STM261107: Auf den Hauptauftrag (Position 10) sollen ebenfalls Minderungsmengen*  
*\* erfasst werden können. Es können keine vom Betrag her negativen zusätzlichen Positionen*  
*\* erfasst werden, sodaß per Konditionsart MM00 auf die Pos.10 der Bestellwert um den*  
*\* entsprechenden Negativbetrag vermindert wird.*  
 CLEAR wf\_mind.  
 SELECT kbetr FROM konv INTO konv-kbetr  
 WHERE knumv EQ wf\_knumv  
 AND kposn EQ ekpo-ebelp  
 AND kschl EQ 'MM00'.  
  
 ADD konv-kbetr TO wf\_mind.  
 ENDSELECT.  
  
 wt\_leist-gpreis\_netto = wt\_leist-gpreis\_netto + wf\_mind. *"STM261107*  
 ADD ekpo-netpr TO wf\_hauptauftr. *"Sichern als Basis für Kostenumlage u.Bauleist.versicherung*  
  
*\* \* bzgl.Steuersatz den Konditionssatz ermitteln A003 = Tab. Steuerkennz.*  
*\* select single \* from A003 where kappl eq 'TX' "Applikation Steuer*  
*\* and kschl eq 'MWVS' "Konditionsart Vorsteu*  
*\* .*  
*\* \* and aland eq 'DE'*  
*\* and mwskz eq ekpo-mwskz.*  
*\**  
*\* select single kbetr from konp into wf\_kbetr*  
*\* where knumh eq a003-knumh*  
*\* and kopos eq '01'.*  
  
  
  
*\* \* Vollen Mehrwertsteuersatz des Steuerkennzeichens aus der Bestellung ermitteln*  
 CLEAR wf\_kbetr.  
  
 CASE st\_kennz.  
 WHEN 'RE' OR 'RD'. *"STM25112015: hier brutto = netto setzen (EU und Drittland Steuerschuld)*  
  
 wt\_leist-gpreis\_auf\_netto =  
 wt\_leist-gpreis\_netto \* ( ( 100 + aufschlag ) / 100 ).  
  
*\* Bruttobeträge*  
 wt\_leist-gpreis\_brutto = wt\_leist-gpreis\_netto.  
 wt\_leist-gpreis\_auf\_brutt = wt\_leist-gpreis\_auf\_netto.  
  
 WHEN OTHERS.  
 SELECT \* FROM a003 WHERE mwskz EQ st\_kennz  
 AND aland EQ wf\_land.  
*\* and aland eq 'DE'.*  
 SELECT SINGLE \* FROM konp WHERE knumh EQ a003-knumh.  
 IF sy-subrc = 0.  
 ADD konp-kbetr TO wf\_kbetr.  
 ENDIF.  
 ENDSELECT.  
 mwst = wf\_kbetr / 10.  
 MOVE mwst TO wt\_leist-mwst.  
 wt\_leist-gpreis\_auf\_netto =  
 wt\_leist-gpreis\_netto \* ( ( 100 + aufschlag ) / 100 ).  
*\* Bruttobeträge*  
 wt\_leist-gpreis\_brutto =  
 wt\_leist-gpreis\_netto \* ( ( 100 + mwst ) / 100 ).  
 wt\_leist-gpreis\_auf\_brutt =  
 wt\_leist-gpreis\_auf\_netto \* ( ( 100 + mwst ) / 100 ).  
 ENDCASE.  
  
 APPEND wt\_leist. CLEAR wt\_leist.  
  
 ENDSELECT.  
  
*\* Summenbildung Bestellwert*  
 CLEAR: wf\_anz\_pos,  
 sum\_net\_1,  
 sum\_net\_2,  
 sum\_bru,  
 sum\_auf\_net,  
 sum\_auf\_bru.  
  
 DESCRIBE TABLE wt\_leist LINES wf\_anz\_pos.  
  
*\*Gruppenverarbeitung um Mehrwertsteuerkennzeichen zu trennen*  
 LOOP AT wt\_leist.  
  
 READ TABLE gt\_mwskz INTO ls\_mwskz WITH KEY mwskz = wt\_leist-mwskz.  
 IF sy-subrc <> 0.  
 ls\_mwskz-mwskz = wt\_leist-mwskz.  
 APPEND ls\_mwskz TO gt\_mwskz.  
 ENDIF.  
 ENDLOOP.  
  
 DATA lv\_tabix\_mwskz LIKE sy-tabix.  
  
 CLEAR: sum\_net\_1, sum\_bru, sum\_auf\_net, sum\_auf\_bru, sum\_net\_2,  
 sum\_bru, sum\_auf\_net, sum\_auf\_bru, mwst\_2, mwst\_2\_1, mwst\_1, mwst\_1\_1.  
  
 LOOP AT gt\_mwskz INTO ls\_mwskz.  
 lv\_tabix\_mwskz = sy-tabix.  
 LOOP AT wt\_leist WHERE mwskz = ls\_mwskz-mwskz.  
 IF lv\_tabix\_mwskz = 1.  
 ADD: wt\_leist-gpreis\_netto TO sum\_net\_1,  
 wt\_leist-gpreis\_brutto TO sum\_bru,  
 wt\_leist-gpreis\_auf\_netto TO sum\_auf\_net,  
 wt\_leist-gpreis\_auf\_brutt TO sum\_auf\_bru.  
 IF sy-tabix = 1.  
 mwst\_1 = wt\_leist-mwst.  
 mwst\_1\_1 = wt\_leist-mwst.  
 ENDIF.  
 ELSE.  
 ADD: wt\_leist-gpreis\_netto TO sum\_net\_2,  
 wt\_leist-gpreis\_brutto TO sum\_bru,  
 wt\_leist-gpreis\_auf\_netto TO sum\_auf\_net,  
 wt\_leist-gpreis\_auf\_brutt TO sum\_auf\_bru.  
 mwst\_2 = wt\_leist-mwst.  
 mwst\_2\_1 = wt\_leist-mwst .  
 ENDIF.  
 ENDLOOP.  
 ENDLOOP.  
  
*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\**  
*\* Baurechnungen lesen \**  
*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\**  
  
 CLEAR: wt\_projre\_storno,  
 wt\_projre\_storno[].  
  
 DATA ws\_rbkp TYPE rbkp.  
 DATA ws\_bkpf TYPE bkpf.  
 CLEAR ws\_rbkp.  
 CLEAR ws\_bkpf.  
  
 DATA lv\_tabix TYPE sytabix.  
  
*\*\*\* Prüfen, ob die Belegnummern (MM und FI) den den Status VORERFASST haben und ggf.*  
*\*\*\*\* bei höherem Systemjahr als Belegjahr auf Aktualsierung (neue Nr.+neues Jahr) prüfen*  
*\*\*\*\* und in den Tabellen /AFM/BAUPROJRE und wt\_projre aktual. (Nr.+Jahr plus Haken in wt\_projre)*  
*\* RBKP Materialbelegskopftabelle*  
*\* BKPF FI-Belegekopftabelle*  
*\*\* da es sich um eine kleine Datenmenge mit eindeutigem Schlüsselzugriff handelt, werden die Betrachtungen bzgl.*  
*\*\* Jahreswechsel bei vorerfassten Belegen, Stornierungen hier nacheinander vorgenommen*  
  
*\* Der Folgebeleg FI-17-er Nr. zum MM-Beleg wird zwar mit gelöscht,*  
*\* aber der neue aktualisierten Beleg in XBLNR hat auch einen neuen FI-Folgebeleg erzeugt und muß zum*  
*\* MM-Beleg aus der BKPF mit AWKEY gelesen werden (analog nach MIRO)*  
*\* select \* from /AFM/BAUPROJRE into corresponding fields of table wt\_projre*  
*\* where pspnr eq psp*  
*\* and ebeln eq bestellnr.*  
 SELECT \* FROM /afm/bauprojre INTO CORRESPONDING FIELDS OF TABLE wt\_projre  
 WHERE ebeln EQ bestellnr.  
  
 LOOP AT wt\_projre. *" where pspnr eq psp*  
 *"and ebeln eq bestellnr.*  
 IF wt\_projre-miro\_mm\_belnr NE space.  
 MOVE sy-tabix TO lv\_tabix.  
  
 SELECT SINGLE \* FROM rbkp INTO ws\_rbkp  
 WHERE belnr EQ wt\_projre-miro\_mm\_belnr  
 AND gjahr EQ wt\_projre-gjahr  
 AND rbstat EQ '2' *"gelöschter Beleg*  
 AND tcode EQ 'MIR7' *"vorerfasst*  
 AND stblg EQ space *"kein stornierter beleg, sonst stünde hier der Referenzbeleg*  
 AND xblnr NE space. *"neue Nr. + neue Jahr Bsp.Eintrag: &&51056016222011*  
  
 IF sy-subrc = 0 AND ws\_rbkp-xblnr CS '&&'. *"Achtung: Inhalt zwecks Erkennung bei jedem Kunden anders.*  
 *"=> es handelt sich um einen vorerfassten Beleg, der durch neuen im neuen Jahr ersetzt worden ist*  
 MOVE: ws\_rbkp-xblnr+2(10) TO wt\_projre-miro\_mm\_belnr,  
 ws\_rbkp-xblnr+12(4) TO wt\_projre-gjahr.  
  
*\* Folgebeleg FI ermitteln zum neuen MM-Beleg*  
 DATA: lv\_refbel TYPE awkey,  
 lv\_blart TYPE blart,  
 lv\_budat TYPE budat.  
  
 SELECT SINGLE blart budat FROM \*rbkp INTO (lv\_blart, lv\_budat)  
 WHERE belnr EQ wt\_projre-miro\_mm\_belnr *"neu erzeugte MM-Nr.*  
 AND gjahr EQ wt\_projre-gjahr.  
  
*\* neues Buchungsdatum übernehmen*  
 MOVE lv\_budat TO wt\_projre-budat.  
  
 CONCATENATE wt\_projre-miro\_mm\_belnr wt\_projre-gjahr INTO lv\_refbel.  
 SELECT SINGLE belnr FROM bkpf INTO wt\_projre-belnr\_miro *"=> neuer FI-Folgebeleg*  
 WHERE bukrs EQ wt\_projre-bukrs  
 AND gjahr EQ wt\_projre-gjahr  
 AND blart EQ lv\_blart  
 AND budat EQ lv\_budat  
 AND awkey EQ lv\_refbel.  
*\* ob der Beleg auch das Kennzeichen für gebucht erhalten muß, wird weiter unten abgefragt (RBSTAT = 5)*  
  
 ENDIF.  
 IF wt\_projre-belnr\_f43 NE space. *"ist FI-Beleg auch vorerfasst und inzwischen aktualisiert worden*  
 SELECT SINGLE \* FROM bkpf INTO ws\_bkpf  
 WHERE bukrs EQ wt\_projre-bukrs  
 AND belnr EQ wt\_projre-belnr\_f43  
 AND gjahr EQ wt\_projre-gjahr\_fi  
 AND bstat EQ 'W' *"Vorerfaßter Beleg mit Wechsel der Belegidentifikation (V=vorerfasst)*  
 AND xblnr NE space.  
 IF sy-subrc = 0.  
 MOVE: ws\_bkpf-xblnr+0(10) TO wt\_projre-belnr\_f43,  
 ws\_bkpf-xblnr+10(4) TO wt\_projre-gjahr\_fi.  
 ENDIF.  
 ENDIF.  
  
*\* BELNR\_GEGENF: wird nicht betrachtet, da die Gegenforderungen nicht vorerfasst werden und demzufolge keinem*  
*\* Belegnummernwechsel unterliegen können.*  
*\* Sollte dies zukünftig geschehen, müßte auch hier noch der neue Referenzbeleg eingelesen werden.*  
  
*\* modify wt\_projre index lv\_tabix.*  
 UPDATE /afm/bauprojre FROM wt\_projre.  
 COMMIT WORK.  
 ENDIF.  
  
 ENDLOOP.  
  
*\*BEJK20191029 start Auskommentiert, da über Check\_Installment\_Payment implementiert*  
*\* CLEAR wt\_projre.*  
*\* REFRESH wt\_projre.*  
*\**  
*\* CLEAR ws\_rbkp.*  
*\* CLEAR ws\_bkpf.*  
*\**  
*\* SELECT \* FROM /afm/bauprojre INTO CORRESPONDING FIELDS OF wt\_projre*  
*\* WHERE ebeln EQ bestellnr.*  
*\* SELECT SINGLE \* FROM rbkp INTO ws\_rbkp*  
*\* WHERE belnr EQ wt\_projre-miro\_mm\_belnr*  
*\* AND gjahr EQ wt\_projre-gjahr*  
*\* AND ( rbstat EQ '2' OR stblg NE space ).*  
*\*\* rbstat = '2' => gelöschter Beleg (z.B. nach Vorerfassung)*  
*\*\* stblg gefüllt => Beleg wurde mit diesem Beleg im Feld "Storniert mit" storniert*  
*\* IF sy-subrc = 0.*  
*\* MOVE wt\_projre TO wt\_projre\_storno.*  
*\* APPEND wt\_projre\_storno. CLEAR wt\_projre\_storno.*  
*\* CLEAR wt\_projre. "BEJK20190710 Darstellung bei Löschung am Anfang*  
*\* ELSE.*  
*\*\* ist der FI-Beleg zum Gew.EB storniert oder vorerfasster Beleg gelöscht (BSTAT=Z)*  
*\* SELECT SINGLE \* FROM bkpf INTO ws\_bkpf*  
*\* WHERE bukrs EQ wt\_projre-bukrs "STM17122014*  
*\* AND belnr EQ wt\_projre-belnr\_f43*  
*\* AND gjahr EQ wt\_projre-gjahr*  
*\* AND ( stblg NE space OR bstat = 'Z' ).*  
*\* IF sy-subrc = 0.*  
*\* CLEAR wt\_projre-belnr\_f43.*  
*\* ENDIF.*  
*\* APPEND wt\_projre. CLEAR wt\_projre.*  
*\* ENDIF.*  
*\* ENDSELECT.*  
*\*BEJK20191029 ende*  
  
*\*{ INSERT C42K912790*  
*\* Alle evtl.stornierten oder gelöschten Belege aus der Tabelle /AFM/BAUPROJRE löschen*  
*\* Pflegedialog auf die Tabelle zwecks Löschung der Rechnungseinträge damit obsolet*  
  
*\*BEJK20191029 start Auskommentiert, da über Check\_Installment\_Payment implementiert*  
*\* DELETE /afm/bauprojre FROM TABLE wt\_projre\_storno.*  
*\*BEJK20191029 ende*  
  
*\* Wurden nach der Vorerfassung die errechneten Buchungsbeträge geändert gebucht, sind die*  
*\* Werte in der /AFMBAUPROJRE nicht mehr aktuell bzw. richtig => an dieser Stelle werden die*  
*\* abhängigen Werte abhängig von den tatsächlich gebuchten Werte für die /AFM/BAUPROJRE neu ermittelt.*  
*\* ------------------------------------------------------------------------------------------ \**  
*\* A C H T U N G*  
*\* die nachträgliche Aktualisierung der Beträge von bereits im Tool erfassten Rechnungen (AZ´s)*  
*\* über die Rechnungsbeträge funktioniert NUR, wenn keine Sicherheitseinbehalte gebucht werden*  
*\* /AFM/BAURE\_STEU-SICH\_EB\_BUCHEN weil bei Buchung der Sich.EB wird der Einbehaltsbetrag in der MIRO*  
*\* mit hinzu addiert, während ohne Buchung der Sich.EB der um den EB gekürzte Betrag erfasst wird*  
*\* wt\_projre-freig\_betr\_b ist dann der falsche Vergleichsbetrag bei Abgleich mit dem CO-Beleg*  
*\**  
*\* MUSS NOCH ANGEPASST WERDEN--- sollte jetzt passen, aber noch T E S T E N bevor folgende IF-Abfrage entfernt wird*  
*\**  
 IF ws\_steuerung-sich\_eb\_buchen EQ space. *"############################################# TESTEN,wenn die IF-Abfrage raus ist*  
  
 DATA: wf\_index TYPE sy-tabix.  
 DATA: ws\_projre TYPE /afm/bauprojre.  
 DATA: wf\_einbeh\_kum\_n TYPE /afm/einbehalt\_kum,  
 wf\_ges\_leist\_net TYPE /afm/g\_lstand\_net,  
 wf\_az\_kum\_net TYPE /afm/az\_kum\_net.  
  
 CLEAR wf\_index.  
 CLEAR ws\_rbkp.  
 LOOP AT wt\_projre WHERE ebeln EQ bestellnr  
 AND miro\_mm\_belnr NE space.  
  
 CLEAR wt\_projre-miro\_durchgebuch.  
  
 MOVE sy-tabix TO wf\_index.  
 SELECT SINGLE \* FROM rbkp INTO ws\_rbkp  
 WHERE belnr EQ wt\_projre-miro\_mm\_belnr  
 AND gjahr EQ wt\_projre-gjahr.  
  
 MOVE: ws\_rbkp-zterm TO wt\_projre-zahlungsbed,  
 ws\_rbkp-xblnr TO wt\_projre-referenz.  
  
*\* Belegstatus abfragen*  
 CASE ws\_rbkp-rbstat.  
 WHEN '5'. *"gebucht A wäre vorerfasst, B voerfasst vollständig gesichert*  
 MOVE 'X' TO wt\_projre-miro\_durchgebuch.  
 WHEN OTHERS.  
 ENDCASE.  
  
*\* Buchungsbeträge abfragen*  
 CASE wt\_projre-re\_art.  
 WHEN 'AZ' OR 'ER'. *"bei ER ergibt Einbehaltssumme 0*  
 IF ws\_rbkp-rmwwr GT wt\_projre-einzel\_re\_betr\_b OR ws\_rbkp-rmwwr LT wt\_projre-einzel\_re\_betr\_b.  
  
*\* ws\_rbkp-rmwwr: Rechnungsbrutto*  
*\* ws\_rbkp-wmwst1: Rechnungssteuerbetrag*  
  
*\* Freigabebetrag: Wert nach Abzug des Sicherheitseinbehaltes*  
 wt\_projre-freig\_betr\_n = ws\_rbkp-rmwwr - ws\_rbkp-wmwst1.  
 wt\_projre-freig\_betr\_b = ws\_rbkp-rmwwr.  
  
*\* Aus gebuchtem Betrag den Rechnungsbetrag (also incl.Sicherheitseinbehalt) ermitteln*  
 IF wt\_projre-erf\_sich\_proz = 'X'. *"Sich.Einbehalte in Prozent erfasst*  
 wt\_projre-einzel\_re\_betr\_n = wt\_projre-freig\_betr\_n / ( ( 100 - wt\_projre-erfuell\_sich ) / 100 ).  
 ELSE.  
 wt\_projre-einzel\_re\_betr\_n = wt\_projre-freig\_betr\_n + wt\_projre-erf\_sich\_betr\_n.  
 ENDIF.  
 wt\_projre-einzel\_re\_betr\_b = wt\_projre-einzel\_re\_betr\_n \* ( ( 100 + wt\_projre-mwst ) / 100 ).  
*\* Einzelne Einbehalte*  
 IF wt\_projre-erf\_sich\_proz = 'X'. *"Sich.Einbehalte in Prozent erfasst*  
 wt\_projre-einbeh\_einz\_n = ( ( wt\_projre-erfuell\_sich / 100 ) \* wt\_projre-einzel\_re\_betr\_n ).  
 ELSE.  
 wt\_projre-einbeh\_einz\_n = wt\_projre-erf\_sich\_betr\_n.  
 ENDIF.  
 wt\_projre-einbeh\_einz\_b = wt\_projre-einbeh\_einz\_n \* ( ( 100 + wt\_projre-mwst ) / 100 ).  
 ENDIF. *"Betragsunterschied Buchungswert zu berechnetem Wert*  
  
 WHEN 'SR'.  
 DATA: wf\_vgl TYPE /afm/freig\_net.  
  
*\*\*\*\* ACHTUNG: SR darf nicht gekürzt werden, sondern müßte storniert/gelöscht werden!!!*  
  
 wf\_vgl = ( ( wt\_projre-freig\_betr\_n + wt\_projre-einbeh\_einz\_n ) \* ( ( 100 + wt\_projre-mwst ) / 100 ) ).  
*\* bei SR wird der Gew.Einbehalt auf den Freigabebetrag addiert, sodaß dieser auch beim Vergleich berechneter Wert zu*  
*\* tatsächlich gebuchtem Wert berücksichtigt bzw. aufaddiert werden muß. Hier zunächst mit Netto-Werten, um evtl.*  
*\* rundungsdifferenzen auszuschließen*  
 IF ws\_rbkp-rmwwr GT wf\_vgl OR ws\_rbkp-rmwwr LT wf\_vgl.  
  
*\* Freigabebetrag: Wert nach Abzug des Gew.einbehaltes*  
 wt\_projre-freig\_betr\_n = ws\_rbkp-rmwwr - ws\_rbkp-wmwst1 - wt\_projre-einbeh\_einz\_n.  
 wt\_projre-freig\_betr\_b = wt\_projre-freig\_betr\_n \* ( ( 100 + wt\_projre-mwst ) / 100 ).  
  
*\* Aus gebuchtem Betrag den Rechnungsbetrag (also incl.) ermitteln*  
 IF wt\_projre-gew\_sich\_proz = 'X'. *"Gew.Einbehalte in Prozent erfasst*  
 wt\_projre-einzel\_re\_betr\_n = wt\_projre-freig\_betr\_n / ( ( 100 - wt\_projre-gewaehrleist ) / 100 ).  
 ELSE.  
 wt\_projre-einzel\_re\_betr\_n = wt\_projre-freig\_betr\_n + wt\_projre-gewaehrl\_betr\_n.  
 ENDIF.  
 wt\_projre-einzel\_re\_betr\_b = wt\_projre-einzel\_re\_betr\_n \* ( ( 100 + wt\_projre-mwst ) / 100 ).  
*\* Einzelne Einbehalte*  
 IF wt\_projre-erf\_sich\_proz = 'X'. *"Gew.Einbehalte in Prozent erfasst*  
 wt\_projre-einbeh\_einz\_n = ( ( wt\_projre-gewaehrleist / 100 ) \* wt\_projre-einzel\_re\_betr\_n ).  
 ELSE.  
 wt\_projre-einbeh\_einz\_n = wt\_projre-gewaehrl\_betr\_n.  
 ENDIF.  
 wt\_projre-einbeh\_einz\_b = wt\_projre-einbeh\_einz\_n \* ( ( 100 + wt\_projre-mwst ) / 100 ).  
 ENDIF.  
 ENDCASE.  
  
 MODIFY wt\_projre INDEX wf\_index.  
  
 CLEAR ws\_rbkp.  
 ENDLOOP.  
  
 CLEAR: wf\_einbeh\_kum\_n,  
 wf\_ges\_leist\_net,  
 wf\_az\_kum\_net.  
  
*\*\* Loop um alle kumulierten Werte zu aktualisieren bzw. aufzusummieren*  
 LOOP AT wt\_projre WHERE pspnr EQ psp  
 AND ebeln EQ bestellnr.  
  
*\* Gesamtleistungsstand*  
 wf\_ges\_leist\_net = wf\_ges\_leist\_net + wt\_projre-einzel\_re\_betr\_n.  
 wt\_projre-ges\_leist\_net\_1 = wt\_projre-ges\_leist\_net\_1.  
 wt\_projre-ges\_leist\_bru\_1 = wt\_projre-ges\_leist\_net\_1 \* ( ( 100 + wt\_projre-mwst\_1 ) / 100 ).  
 wt\_projre-ges\_leist\_net\_2 = wt\_projre-ges\_leist\_net\_2.  
 wt\_projre-ges\_leist\_bru\_2 = wt\_projre-ges\_leist\_net\_2 \* ( ( 100 + wt\_projre-mwst\_2 ) / 100 ).  
  
*\* Sicherheits-Einbehalte kumuliert*  
 wf\_einbeh\_kum\_n = wf\_einbeh\_kum\_n + wt\_projre-einbeh\_einz\_n.  
 wt\_projre-einbeh\_kum\_n = wf\_einbeh\_kum\_n.  
 wt\_projre-einbeh\_kum\_b = wt\_projre-einbeh\_kum\_n \* ( ( 100 + wt\_projre-mwst ) / 100 ).  
  
*\* Abschlagszahlungen kumuliert*  
 wf\_az\_kum\_net = wf\_az\_kum\_net + wt\_projre-freig\_betr\_n.  
 wt\_projre-az\_kum\_net = wf\_az\_kum\_net.  
  
  
*\* erst am ende, wenn alle Werte aktualisiert sind, aktuellen prozentualen Stand ermitteln*  
 wt\_projre-prozent\_akt = ( wt\_projre-ges\_leist\_net\_1 + wt\_projre-ges\_leist\_net\_2 ) / sum\_auf\_net.  
  
 MODIFY wt\_projre INDEX sy-tabix.  
  
 CLEAR ws\_projre.  
 MOVE wt\_projre TO ws\_projre.  
*\* select single \* from /AFM/BAUPROJRE where pspnr eq wt\_projre-pspnr*  
*\* and ebeln eq wt\_projre-ebeln.*  
 SELECT SINGLE \* FROM /afm/bauprojre WHERE ebeln EQ wt\_projre-ebeln.  
 IF sy-subrc = 0.  
 MODIFY /afm/bauprojre FROM ws\_projre.  
 ENDIF.  
 ENDLOOP.  
  
 ELSE.  
  
 CLEAR wf\_index.  
 CLEAR ws\_rbkp.  
 LOOP AT wt\_projre WHERE pspnr EQ psp  
 AND ebeln EQ bestellnr  
 AND miro\_mm\_belnr NE space.  
  
 CLEAR wt\_projre-miro\_durchgebuch.  
  
 MOVE sy-tabix TO wf\_index.  
 SELECT SINGLE \* FROM rbkp INTO ws\_rbkp  
 WHERE belnr EQ wt\_projre-miro\_mm\_belnr  
 AND gjahr EQ wt\_projre-gjahr.  
  
 MOVE: ws\_rbkp-zterm TO wt\_projre-zahlungsbed,  
 ws\_rbkp-xblnr TO wt\_projre-referenz.  
  
*\* Belegstatus abfragen*  
 CASE ws\_rbkp-rbstat.  
 WHEN '5'. *"gebucht A wäre vorerfasst*  
 MOVE 'X' TO wt\_projre-miro\_durchgebuch.  
 WHEN OTHERS.  
 ENDCASE.  
  
 MODIFY wt\_projre INDEX wf\_index.  
 CLEAR ws\_rbkp.  
 ENDLOOP.  
  
 ENDIF. *"ws\_steuerung-SICH\_EB\_BUCHEN eq space. ###UPD*  
  
  
 DESCRIBE TABLE wt\_projre LINES re\_max.  
*\* Vorbelegen Rechnungsart, Bezeichnung und ID für erste Rechnung*  
 IF re\_max NE 0.  
 READ TABLE wt\_projre INDEX 1.  
 MOVE: wt\_projre-bez TO rechnung,  
 wt\_projre-re\_art TO art.  
*\* id = 1. "BEJK20190708 ids an zaehler anpassen AUSKOMMENTIERT*  
 id = wt\_projre-zaehler. *"BEJK20190708 ids an zaehler anpassen*  
 ENDIF.  
  
*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\**  
*\*STM15022013: Aufgrund nachträglicher wilder Wechsel von Steuerkennzeichen, PSP-Elementen und somit auch Buchungskreisen innerhalb*  
*\* der Bestellungen erfolgt hier bei Programmstart ein Update auf die beiden Tabellen /AFM/BAUPROJ und /AFM/BAUPROJRE*  
*\* bzgl. Buchungskreis (aus PSP der 1.Pos. der Bestellung ohne Löschkennzeichen)*  
*\* Bewusst erst hier platziert, damiert die vorherige Ermittlung von evtl. stornierten Belegen noch zum evtl. anderen Bukrs*  
*\* funktioniert*  
 DATA ls\_bauproj TYPE /afm/bauproj.  
 DATA lt\_bauprojre TYPE TABLE OF /afm/bauprojre WITH HEADER LINE.  
  
 SELECT SINGLE \* FROM /afm/bauproj INTO ls\_bauproj  
 WHERE ebeln EQ bestellnr.  
 IF sy-subrc = 0.  
 IF ls\_bauproj-bukrs NE bukrs.  
 MOVE bukrs TO ls\_bauproj-bukrs.  
 UPDATE /afm/bauproj FROM ls\_bauproj.  
 IF sy-subrc = 0.  
 COMMIT WORK AND WAIT.  
 ENDIF.  
 ENDIF.  
 ENDIF.  
 SELECT \* FROM /afm/bauprojre INTO TABLE lt\_bauprojre  
 WHERE ebeln EQ bestellnr.  
 IF sy-subrc = 0.  
 LOOP AT lt\_bauprojre.  
 IF lt\_bauprojre-bukrs NE bukrs.  
 MOVE bukrs TO lt\_bauprojre-bukrs.  
 MODIFY lt\_bauprojre INDEX sy-tabix.  
 ENDIF.  
 ENDLOOP.  
 UPDATE /afm/bauprojre FROM TABLE lt\_bauprojre.  
 IF sy-subrc = 0.  
 COMMIT WORK AND WAIT.  
 ENDIF.  
 ENDIF.  
  
*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\**  
*\* Textcontainer erzeugen. Für bestehendes Projekt könnte Langtext*  
*\* bestehen, sodaß hier zwecks Füllen bereits der Textcontainer erzeugt*  
*\* werden muß*  
*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\**  
 IF editor IS INITIAL. *"Nur create object wenn noch leer*  
  
*\* initilize local variable with sy-repid, since sy-repid doesn't work*  
*\* as parameter directly.*  
 repid = sy-repid.  
  
*\* create control container*  
 CREATE OBJECT textedit\_custom\_container  
 EXPORTING  
 container\_name = 'TEXTEDITOR'  
 EXCEPTIONS  
 cntl\_error = 1  
 cntl\_system\_error = 2  
 create\_error = 3  
 lifetime\_error = 4  
 lifetime\_dynpro\_dynpro\_link = 5.  
 IF sy-subrc NE 0.  
*\* add your handling*  
 ENDIF.  
  
*\* create calls constructor, which initializes, creats and links*  
*\* a TextEdit Control*  
 CREATE OBJECT editor  
 EXPORTING  
 parent = textedit\_custom\_container  
 wordwrap\_mode = cl\_gui\_textedit=>wordwrap\_at\_fixed\_position  
 wordwrap\_to\_linebreak\_mode = cl\_gui\_textedit=>true  
 EXCEPTIONS  
 OTHERS = 1.  
 IF sy-subrc NE 0.  
 CALL FUNCTION 'POPUP\_TO\_INFORM'  
 EXPORTING  
 titel = repid  
 txt2 = space  
 txt1 = TEXT-001.  
 ENDIF.  
  
 ENDIF. *"ende create object*  
  
*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\**  
*\* Bauvertragsdaten lesen \**  
*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\**  
  
 SELECT \* FROM /afm/bau\_ggn\_key INTO TABLE gt\_ggn\_key. *"#NoWhere ADD ckn24062020*  
  
 SELECT SINGLE \* FROM /afm/bauproj WHERE ebeln EQ bestellnr.  
 IF sy-subrc = 0.  
 MOVE-CORRESPONDING /afm/bauproj TO wa\_bauvertr\_act.  
  
 *"ADD ckn20210810 - Start - Für einen Abgleich sichern*  
 MOVE-CORRESPONDING /afm/bauproj TO wa\_bauproj\_comp.  
 *"ADD ckn20210810 - Ende*  
  
*\* Feld Anlagenunternummer im Kopfbereich belegen*  
 MOVE wa\_bauvertr\_act-anlnr TO anlnr.  
  
*\* Vertrags-/Detailwerte ins Dynpro 1022 setzen*  
 MOVE: wa\_bauvertr\_act-erfuell\_sich TO erfuell\_sich,  
 wa\_bauvertr\_act-gewaehrleist TO gewaehrleist,  
 wa\_bauvertr\_act-gewaehrl\_dauer TO gewaehrl\_dauer,  
 wa\_bauvertr\_act-straf\_nach TO straf\_nach,  
 wa\_bauvertr\_act-straf\_max TO straf\_max,  
*\* wa\_bauvertr\_act-gegenstand to gegenstand,*  
 wa\_bauvertr\_act-anz\_re TO anz\_re,  
 wa\_bauvertr\_act-adresse TO adresse,  
 wa\_bauvertr\_act-faell\_acontore TO faell\_ac,  
 wa\_bauvertr\_act-faell\_schlussre TO faell\_sr,  
*\* wa\_bauvertr\_act-vergabeart to vergabeart,*  
 wa\_bauvertr\_act-bukrs TO bukrs,  
 wa\_bauvertr\_act-waers TO waers.  
*\* wa\_bauvertr\_act-aufschlag to aufschlag. "Aufschlag stets nur aus Bestellung ziehen.*  
  
 IF wa\_bauvertr\_act-straf\_in\_proz EQ 'X'.  
 MOVE: wa\_bauvertr\_act-vertr\_straf TO vertr\_straf\_betr,  
 'X' TO radio\_proz.  
 CLEAR radio\_betr.  
 ELSE.  
 MOVE: wa\_bauvertr\_act-vertr\_straf\_betr TO vertr\_straf\_betr,  
 'X' TO radio\_betr.  
 CLEAR radio\_proz.  
 ENDIF.  
  
 IF wa\_bauvertr\_act-straf\_pro\_kal EQ 'X'.  
 MOVE 'X' TO radio\_kal.  
 CLEAR radio\_wer.  
 ELSE.  
 MOVE 'X' TO radio\_wer.  
 CLEAR radio\_kal.  
 ENDIF.  
  
*\* Vertragserfüllungsdaten lesen.*  
 MOVE: wa\_bauvertr\_act-vor\_buerg\_ein TO vor\_buerg\_ein,  
 wa\_bauvertr\_act-vor\_buerg\_aus TO vor\_buerg\_aus,  
 wa\_bauvertr\_act-vor\_bem TO vor\_bem,  
 wa\_bauvertr\_act-vor\_betrag TO vor\_betrag,  
 wa\_bauvertr\_act-erf\_buerg\_ein TO erf\_buerg\_ein,  
 wa\_bauvertr\_act-erf\_buerg\_aus TO erf\_buerg\_aus,  
 wa\_bauvertr\_act-erf\_bem TO erf\_bem,  
 wa\_bauvertr\_act-erf\_betrag TO erf\_betrag,  
 wa\_bauvertr\_act-gew\_buerg\_ein TO gew\_buerg\_ein,  
 wa\_bauvertr\_act-gew\_buerg\_aus TO gew\_buerg\_aus,  
 wa\_bauvertr\_act-gew\_buerg\_bem TO gew\_buerg\_bem,  
 wa\_bauvertr\_act-gew\_betrag TO gew\_betrag,  
 wa\_bauvertr\_act-gew\_sr\_am TO gew\_sr\_am,  
 wa\_bauvertr\_act-gew\_abn\_am TO gew\_abn\_am,  
 wa\_bauvertr\_act-gew\_maeng TO gew\_maeng,  
 wa\_bauvertr\_act-gew\_frist TO gew\_frist,  
 wa\_bauvertr\_act-gew\_nachweis TO gew\_nachweis,  
 wa\_bauvertr\_act-gew\_bem TO gew\_bem.  
  
*\* Daten zur Ausbezahlung des Gewährleistungseingehaltes lesen.*  
 IF wa\_bauvertr\_act-gew\_eb\_kein EQ 'X'.  
 MOVE 'X' TO radio\_voll.  
 CLEAR: radio\_teil,  
 anteil\_betr,  
 radio\_null.  
 ELSEIF wa\_bauvertr\_act-gew\_eb\_teilw EQ 'X'.  
 MOVE: 'X' TO radio\_teil,  
 wa\_bauvertr\_act-gew\_eb\_teil\_betr TO anteil\_betr.  
 CLEAR: radio\_voll,  
 radio\_null.  
  
 ELSEIF wa\_bauvertr\_act-gew\_eb\_voll EQ 'X'.  
 MOVE 'X' TO radio\_null.  
 CLEAR: radio\_teil,  
 anteil\_betr,  
 radio\_voll.  
 ENDIF.  
  
*\* MOVE: 'BAUL' TO wt\_ford-key,*  
*\* 'Bauleistungsversicherung' TO wt\_ford-bez,*  
*\* '%' TO wt\_ford-als,*  
*\* /afm/bauproj-bauleist\_vers TO wt\_ford-proz,*  
*\* /afm/bauproj-bauleist\_betr\_n TO wt\_ford-netto\_betr,*  
*\* mwst to wt\_ford-mwst,*  
*\* /afm/bauproj-bauleist\_betr\_b TO wt\_ford-brutto\_betr.*  
*\*\*\* 180708 MSN !!!*  
*\* Gegenforderungen / Belastungsanzeigen lesen*  
 MOVE: 'BAUL' TO wt\_ford-key,  
*\* 'Bauleistungsversicherung' TO wt\_ford-bez, "DEL ckn24062020*  
*\* '%' TO wt\_ford-als, "DEL ckn24062020*  
 /afm/bauproj-bauleist\_bez TO wt\_ford-bez, *"ADD ckn24062020*  
 /afm/bauproj-bauleist\_als TO wt\_ford-als, *"ADD ckn24062020*  
 /afm/bauproj-bauleist\_vers TO wt\_ford-proz,  
 /afm/bauproj-bauleist\_betr\_n TO wt\_ford-netto\_betr,  
 /afm/bauproj-bauleist\_vers\_mwst TO wt\_ford-mwst,  
 /afm/bauproj-bauleist\_betr\_b TO wt\_ford-brutto\_betr.  
 IF wt\_ford-bez IS INITIAL. PERFORM fill\_bez USING 'BAUL' CHANGING wt\_ford-bez. ENDIF. *"ADD ckn24062020*  
 IF wt\_ford-als IS INITIAL. wt\_ford-als = 'P'. ENDIF. *"ADD ckn24062020*  
 APPEND wt\_ford. CLEAR wt\_ford.  
 MOVE: 'KOST' TO wt\_ford-key,  
*\* 'Energiekosten' TO wt\_ford-bez, "DEL ckn24062020*  
*\* '%' TO wt\_ford-als, "DEL ckn24062020*  
 /afm/bauproj-kostenum\_bez TO wt\_ford-bez, *"ADD ckn24062020*  
 /afm/bauproj-kostenum\_als TO wt\_ford-als, *"ADD ckn24062020*  
 /afm/bauproj-kostenumlage TO wt\_ford-proz,  
 /afm/bauproj-kostenum\_betr\_n TO wt\_ford-netto\_betr,  
 /afm/bauproj-kostenumlage\_mwst TO wt\_ford-mwst,  
 /afm/bauproj-kostenum\_betr\_b TO wt\_ford-brutto\_betr.  
 IF wt\_ford-bez IS INITIAL. PERFORM fill\_bez USING 'KOST' CHANGING wt\_ford-bez. ENDIF. *"ADD ckn24062020*  
 IF wt\_ford-als IS INITIAL. wt\_ford-als = 'P'. ENDIF. *"ADD ckn24062020*  
 APPEND wt\_ford. CLEAR wt\_ford.  
 MOVE: 'REIN' TO wt\_ford-key,  
*\* 'Schlussreinigung' TO wt\_ford-bez, "DEL ckn24062020*  
*\* '%' TO wt\_ford-als, "DEL ckn24062020*  
 /afm/bauproj-reinigung\_bez TO wt\_ford-bez, *"ADD ckn24062020*  
 /afm/bauproj-reinigung\_als TO wt\_ford-als, *"ADD ckn24062020*  
 /afm/bauproj-reinigung TO wt\_ford-proz,  
 /afm/bauproj-rein\_betr\_n TO wt\_ford-netto\_betr,  
 /afm/bauproj-reinigung\_mwst TO wt\_ford-mwst,  
 /afm/bauproj-rein\_betr\_b TO wt\_ford-brutto\_betr.  
 IF wt\_ford-bez IS INITIAL. PERFORM fill\_bez USING 'REIN' CHANGING wt\_ford-bez. ENDIF. *"ADD ckn24062020*  
 IF wt\_ford-als IS INITIAL. wt\_ford-als = 'P'. ENDIF. *"ADD ckn24062020*  
 APPEND wt\_ford. CLEAR wt\_ford.  
  
 SELECT \* FROM /afm/bauprojbel WHERE ebeln EQ bestellnr.  
  
 MOVE: /afm/bauprojbel-zaehler TO wt\_ford-key,  
 /afm/bauprojbel-bez TO wt\_ford-bez,  
 'Betrag' TO wt\_ford-als,  
 /afm/bauprojbel-netto\_betr TO wt\_ford-netto\_betr,  
 /afm/bauprojbel-mwst TO wt\_ford-mwst,  
 /afm/bauprojbel-brutto\_betr TO wt\_ford-brutto\_betr,  
 /afm/bauprojbel-als TO wt\_ford-als, *"ADD ckn24062020*  
 /afm/bauprojbel-proz TO wt\_ford-proz. *"ADD ckn24062020*  
 APPEND wt\_ford. CLEAR wt\_ford.  
 ENDSELECT.  
  
*\*----------------------------------------------------------------\**  
*\* Textcontainer mit Langtext füllen \**  
*\*----------------------------------------------------------------\**  
  
*\* Langtext Vertragsgrundlage einlesen.*  
 CLEAR wt\_langtext.  
 REFRESH wt\_langtext.  
  
*\* Langtext-Schlüssel*  
 CONCATENATE '/AFM/' bestellnr wf\_posid INTO wf\_tdname\_lt.  
  
*\* Langtext (Vertragsgrundlage)*  
 CLEAR thead.  
 MOVE: 'ST' TO thead-tdid,  
 'D' TO thead-tdspras,  
 wf\_tdname\_lt TO thead-tdname,  
 'TEXT' TO thead-tdobject,  
 '072' TO thead-tdlinesize.  
  
 CALL FUNCTION 'READ\_TEXT'  
 EXPORTING  
 id = thead-tdid  
 language = thead-tdspras  
 name = thead-tdname  
 object = thead-tdobject  
 IMPORTING  
 header = thead  
 TABLES  
 lines = wt\_langtext  
 EXCEPTIONS  
 id = 1  
 language = 2  
 name = 3  
 not\_found = 4  
 object = 5  
 reference\_check = 6  
 wrong\_access\_to\_archive = 7  
 OTHERS = 8.  
 IF sy-subrc <> 0.  
 IF sy-subrc NE 4. *"wahrscheinlich kein Text gepflegt*  
 MESSAGE i000(su) WITH 'Fehler beim Einlesen der'  
 'Vertragsgrundlage.'.  
 ENDIF.  
 ENDIF.  
  
*\* Den Inhalt aus wt\_langtext (nur die Spalte TDLINE) erst in Hilfstab.*  
*\* schieben, um dann den Rumpf in die wt\_text\_in zu setzen. Die Methode füllen des Textcontrols erfordert diesen Typ.*  
  
 CLEAR: wt\_hilf,  
 wt\_hilf[].  
  
 LOOP AT wt\_langtext.  
 MOVE wt\_langtext-tdline TO wt\_hilf-line.  
 APPEND wt\_hilf. CLEAR wt\_hilf.  
 ENDLOOP.  
  
 MOVE wt\_hilf[] TO wt\_text\_in.  
  
*\* send R/3 table to control*  
 CALL METHOD editor->set\_text\_as\_r3table  
 EXPORTING  
 table = wt\_text\_in  
 EXCEPTIONS  
 OTHERS = 1.  
 IF sy-subrc NE 0.  
 CALL FUNCTION 'POPUP\_TO\_INFORM'  
 EXPORTING  
 titel = repid  
 txt2 = space  
 txt1 = TEXT-003.  
 ENDIF.  
  
 ELSE. *" es existieren noch keine gespeicherten Bauprojektdaten:*  
*\* keine Gegenforderungen => Füllen mit Defaultwerten*  
*\* keine Vertragdaten => Defaultwerte aus Tab. /AFM/BAU\_STAMMD ziehen*  
  
*\* Defaultwerte zum Projekt bzw. Standardwerte*  
*\*START 20190715 BEJK Anpassung Customizing pro Kostengruppe*  
 IF lv\_kstgrp IS NOT INITIAL.  
 SELECT SINGLE \* FROM /afm/bau\_stammd  
 WHERE pspnr = psp  
 AND kstgrp = lv\_kstgrp.  
 IF sy-subrc <> 0.  
 SELECT SINGLE \* FROM /afm/bau\_stammd  
 WHERE pspnr = psp.  
 ENDIF.  
 ELSE.  
*\*ENDE 20190715 BEJK*  
 SELECT SINGLE \* FROM /afm/bau\_stammd WHERE pspnr EQ psp.  
 ENDIF.*"BEJK20190715 Anpassung Customizing pro Kostengruppe*  
 IF sy-subrc = 0.  
 treffer\_psp = 'X'.  
 MOVE-CORRESPONDING /afm/bau\_stammd TO wa\_bauvertr.  
 MOVE wa\_bauvertr-anlnr TO anlnr.  
 ELSE.  
 SELECT \* FROM /afm/bau\_stammd CLIENT SPECIFIED INTO it\_baumod  
 WHERE mandt = sy-mandt  
 AND kstgrp = lv\_kstgrp.*"BEJK20190715 Anpassung Customizing pro Kostengruppe*  
 CALL FUNCTION 'CONVERSION\_EXIT\_KONPD\_OUTPUT'  
 EXPORTING  
 input = it\_baumod-pspnr *"jetzt Projektdefinition !!!*  
 IMPORTING  
 output = it\_posid  
*\* PSELT =*  
 .  
 MOVE it\_posid TO it\_baumod-posid. *"CHAR 24*  
 APPEND it\_baumod.  
 ENDSELECT.  
  
 SORT it\_baumod DESCENDING BY posid.  
 LOOP AT it\_baumod.  
*\* CALL FUNCTION 'CONVERSION\_EXIT\_ABPSP\_OUTPUT'*  
*\* EXPORTING*  
*\* INPUT = it\_baumod-pspnr "NUMC 8*  
*\* IMPORTING*  
*\* OUTPUT = it\_posid. "CHAR 24*  
  
*\* len = strlen( it\_posid ).*  
 len = strlen( it\_baumod-posid ).  
*\* if it\_posid = wf\_posid(len).*  
 IF it\_baumod-posid = wf\_posid(len).  
 treffer\_psp = 'X'.  
 MOVE-CORRESPONDING it\_baumod TO wa\_bauvertr.  
 MOVE wa\_bauvertr-anlnr TO anlnr.  
 EXIT.  
 ENDIF.  
 ENDLOOP.  
*\*START 20190715 BEJK Anpassung Customizing pro Kostengruppe*  
 IF anlnr IS INITIAL.  
 FREE it\_baumod.  
 SELECT \* FROM /afm/bau\_stammd CLIENT SPECIFIED INTO it\_baumod  
 WHERE mandt = sy-mandt  
 AND kstgrp IS NULL.  
 CALL FUNCTION 'CONVERSION\_EXIT\_KONPD\_OUTPUT'  
 EXPORTING  
 input = it\_baumod-pspnr *"jetzt Projektdefinition !!!*  
 IMPORTING  
 output = it\_posid.  
 MOVE it\_posid TO it\_baumod-posid. *"CHAR 24*  
 APPEND it\_baumod.  
 ENDSELECT.  
  
 SORT it\_baumod DESCENDING BY posid.  
 LOOP AT it\_baumod.  
 len = strlen( it\_baumod-posid ).  
 IF it\_baumod-posid = wf\_posid(len).  
 treffer\_psp = 'X'.  
 MOVE-CORRESPONDING it\_baumod TO wa\_bauvertr.  
 MOVE wa\_bauvertr-anlnr TO anlnr.  
 EXIT.  
 ENDIF.  
 ENDLOOP.  
 ENDIF.  
*\*ENDE 20190715 BEJK*  
 ENDIF.  
  
*\* wenn keine Werte zum PSP ermittelt werden konnten, die Defaultwerte des Dummy-PSP A.ZZ-Z einlesen*  
 IF treffer\_psp NE 'X'.  
 CALL FUNCTION 'CONVERSION\_EXIT\_KONPD\_INPUT' *"in /AFM/BAU\_STAMMD sind nur noch die Projektdefinitionen gepflegt (realistischer)*  
 EXPORTING  
 input = ws\_const-default\_psp  
 IMPORTING  
 output = psp\_elem  
*\* PROJWA =*  
*\* EXCEPTIONS*  
*\* NOT\_FOUND = 1*  
*\* OTHERS = 2*  
 .  
 IF sy-subrc <> 0.  
*\* MESSAGE ID SY-MSGID TYPE SY-MSGTY NUMBER SY-MSGNO*  
*\* WITH SY-MSGV1 SY-MSGV2 SY-MSGV3 SY-MSGV4.*  
 ENDIF.  
  
*\*START 20190715 BEJK Anpassung Customizing pro Kostengruppe*  
 IF lv\_kstgrp IS NOT INITIAL.  
 SELECT SINGLE \* FROM /afm/bau\_stammd  
 WHERE pspnr = psp\_elem  
 AND kstgrp = lv\_kstgrp.  
 IF sy-subrc <> 0.  
 SELECT SINGLE \* FROM /afm/bau\_stammd  
 WHERE pspnr = psp\_elem.  
 ENDIF.  
 ELSE.  
*\*ENDE 20190715 BEJK*  
 SELECT SINGLE \* FROM /afm/bau\_stammd WHERE pspnr EQ psp\_elem.  
 ENDIF.*"BEJK20190715 Anpassung Customizing pro Kostengruppe*  
 IF sy-subrc = 0.  
 MOVE-CORRESPONDING /afm/bau\_stammd TO wa\_bauvertr.  
 ENDIF.  
 ENDIF.  
  
*\* Itab wt\_ford bzw. TC tc\_ford mit Bauleistungsvericherungs- und Kostenumlage -Prozentsätzen vorbelegen*  
 MOVE: 'KOST' TO wt\_ford-key,  
*\* 'Energiekosten' TO wt\_ford-bez, "DEL ckn24062020*  
*\* '%' TO wt\_ford-als, "DEL ckn24062020*  
 'P' TO wt\_ford-als,  
 wa\_bauvertr-kostenumlage TO wt\_ford-proz,  
*\* wa\_bauvertr-mwst to wt\_ford-mwst.*  
*\* MWST-Wert aus Bestellung einsetzen*  
 mwst TO wt\_ford-mwst.  
 PERFORM fill\_bez USING 'KOST' CHANGING wt\_ford-bez. *"ADD ckn24062020*  
*\* Werte für Kostenumlage und Bauleistungsversicherung werden erst bei*  
*\* der Schlußrechnung ermittelt, und zwar auf den vorliegenden Leistungsstand nach Abzug Aufschlag.*  
 APPEND wt\_ford. CLEAR wt\_ford.  
  
 MOVE: 'BAUL' TO wt\_ford-key,  
*\* 'Bauleistungsversicherung' TO wt\_ford-bez, "DEL ckn24062020*  
*\* '%' TO wt\_ford-als, "DEL ckn24062020*  
 'P' TO wt\_ford-als,  
 wa\_bauvertr-bauleist\_vers TO wt\_ford-proz,  
*\* wa\_bauvertr-mwst TO wt\_ford-mwst.*  
*\* 0 TO wt\_ford-mwst."STM09032016*  
 mwst TO wt\_ford-mwst.*"BEJK20190731 bzgl. Nachfrage Grassmüller*  
 PERFORM fill\_bez USING 'BAUL' CHANGING wt\_ford-bez. *"ADD ckn24062020*  
*\* CLEAR wt\_ford-mwst."BEJK20190731 bzgl. Nachfrage Grassmüller*  
 APPEND wt\_ford. CLEAR wt\_ford.  
  
 MOVE: 'REIN' TO wt\_ford-key,  
*\* 'Schlussreinigung' TO wt\_ford-bez, "DEL ckn24062020*  
*\* '%' TO wt\_ford-als, "DEL ckn24062020*  
 'P' TO wt\_ford-als,  
 wa\_bauvertr-reinigung TO wt\_ford-proz,  
*\* wa\_bauvertr-mwst to wt\_ford-mwst.*  
 mwst TO wt\_ford-mwst.  
 PERFORM fill\_bez USING 'REIN' CHANGING wt\_ford-bez. *"ADD ckn24062020*  
 APPEND wt\_ford. CLEAR wt\_ford.  
  
*\* Defaultwerte ins Dynpro 1022 setzen*  
 MOVE: wa\_bauvertr-erfuell\_sich TO erfuell\_sich,  
 wa\_bauvertr-gewaehrleist TO gewaehrleist,  
 wa\_bauvertr-gewaehrl\_dauer TO gewaehrl\_dauer,  
 wa\_bauvertr-straf\_nach TO straf\_nach,  
 wa\_bauvertr-straf\_max TO straf\_max.  
  
 IF wa\_bauvertr-straf\_in\_proz EQ 'X'.  
 MOVE: wa\_bauvertr-vertr\_straf TO vertr\_straf\_betr,  
 'X' TO radio\_proz.  
 CLEAR radio\_betr.  
 ELSE.  
 MOVE: wa\_bauvertr-vertr\_straf\_betr TO vertr\_straf\_betr,  
 'X' TO radio\_betr.  
 CLEAR radio\_proz.  
 ENDIF.  
  
 IF wa\_bauvertr-straf\_pro\_kal EQ 'X'.  
 MOVE 'X' TO radio\_kal.  
 CLEAR radio\_wer.  
 ELSE.  
 MOVE 'X' TO radio\_wer.  
 CLEAR radio\_kal.  
 ENDIF.  
  
 ENDIF. *"sy-subrc auf /AFM/BAUPROJ*  
  
*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\**  
*\*\* Daten zur Bestellung und Projekt ziehen, sofern bereits zur Be- \**  
*\*\* stellung Projektdaten gepflegt worden sind. \**  
*\*\* Tab. /AFM/BAUPROJ, /AFM/BAUPROJRE und /AFM/BAUPROJBEL \**  
*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\**  
*\**  
*\*\* Bestellpositionen einlesen*  
*\* clear: wt\_leist,*  
*\* wt\_leist[].*  
*\**  
*\*\* Speichern der Belegkonditionsnummer KNUMV, um später aus EKPO den*  
*\*\* Gesamtleistungsnettowert nach Abzug der Rabatte auslesen zu können*  
*\* select single knumv from ekko into wf\_knumv*  
*\* where ebeln eq bestellnr.*  
*\**  
*\* select \* from ekpo where ebeln eq bestellnr.*  
*\*\* daten in die Itab wt\_leist zur Anzeige schreiben*  
*\* move: ekpo-ebelp to wt\_leist-ebelp,*  
*\* ekpo-txz01 to wt\_leist-txz01,*  
*\*\* ekpo-netpr to wt\_leist-gpreis\_netto.*  
*\* ekpo-brtwr to wt\_leist-gpreis\_netto.*  
*\**  
*\*\* Nachlaß/Rabatt RA00 = Rabatt auf netto*  
*\* select single \* from konv where knumv eq wf\_knumv*  
*\* and kschl = 'RA00'.*  
*\* aufschlag = konv-kbetr / 10.*  
*\* move aufschlag to wt\_leist-aufschlag.*  
*\**  
*\**  
*\* if ekpo-ebelp eq '00010'. "Pos.10 = Hauptauftrag*  
*\* move 'HA' to wt\_leist-art.*  
*\**  
*\**  
*\* elseif ekpo-ebelp gt '00010'. "alle folgenden sind Nachträge*  
*\* move 'NA' to wt\_leist-art.*  
*\* endif.*  
*\**  
*\* add ekpo-netpr to wf\_hauptauftr. "Sichern als Basis für Kosten-*  
*\* "umlage u.Bauleist.versicherung*  
*\**  
*\*\* bzgl.Steuersatz den Konditionssatz ermitteln A003 = Tab. Steuerkennz.*  
*\* select single \* from A003 where kappl eq 'TX' "Applikation*  
*\*Steuer*  
*\* and kschl eq 'MWVS' "Konditionsart Vorsteu*  
*\*.*  
*\*\* and aland eq 'DE'*  
*\* and mwskz eq ekpo-mwskz.*  
*\**  
*\* select single kbetr from konp into wf\_kbetr*  
*\* where knumh eq a003-knumh*  
*\* and kopos eq '01'.*  
*\**  
*\*\* Mehrwertsteuersatz*  
*\* mwst = wf\_kbetr / 10.*  
*\* move mwst to wt\_leist-mwst.*  
*\**  
*\*\* Nettobetrag nach Aufschlag*  
*\* wt\_leist-gpreis\_auf\_netto =*  
*\* wt\_leist-gpreis\_netto \* ( ( 100 + aufschlag ) / 100 )*  
*\*.*  
*\**  
*\*\*Bruttobeträge*  
*\* wt\_leist-gpreis\_brutto =*  
*\* wt\_leist-gpreis\_netto \* ( ( 100 + mwst ) / 100 )*  
*\*.*  
*\* wt\_leist-gpreis\_auf\_brutt =*  
*\* wt\_leist-gpreis\_auf\_netto \* ( ( 100 + mwst ) / 100 )*  
*\*.*  
*\**  
*\**  
*\* append wt\_leist. clear wt\_leist.*  
*\**  
*\* endselect.*  
*\**  
*\*\* Summenbildung Bestellwert*  
*\* clear: wf\_anz\_pos,*  
*\* sum\_net,*  
*\* sum\_bru,*  
*\* sum\_auf\_net,*  
*\* sum\_auf\_bru.*  
*\**  
*\* write tc\_leist-lines to wf\_anz\_pos.*  
*\**  
*\* loop at wt\_leist.*  
*\* add: wt\_leist-gpreis\_netto to sum\_net,*  
*\* wt\_leist-gpreis\_brutto to sum\_bru,*  
*\* wt\_leist-gpreis\_auf\_netto to sum\_auf\_net,*  
*\* wt\_leist-gpreis\_auf\_brutt to sum\_auf\_bru.*  
*\* endloop.*  
  
 MOVE aufschlag TO auf\_akt. *"Vorbelegung Aufschlag im Subscreen 1031*  
 *"Bereich Neuer Stand*  
  
*\* Vorbelegung Buchungsperiode, Textfeld und Fälligkeitstermin für F-43*  
 MOVE: sy-datum+4(2) TO monat,  
 monat TO monat\_zeb,  
 kreditor TO text,  
 '20991231' TO faellig.  
  
 subscr\_re = '1031'.  
  
 DATA lv\_wrt\_kum TYPE c LENGTH 1.  
  
 SELECT SINGLE wrt\_kum FROM /afm/baure\_steu INTO lv\_wrt\_kum.  
  
 IF lv\_wrt\_kum IS NOT INITIAL.  
 cbkum = 'X'.  
 ENDIF.  
  
 *"ADD ckn08072020 - Start*  
 IF gs\_bau\_excep IS INITIAL.  
 SELECT SINGLE \* FROM /afm/bau\_excep INTO gs\_bau\_excep.  
 ENDIF.  
 *"ADD ckn08072020 - Ende*  
  
 ELSE.  
*\* id = '0001'.*  
 subscr\_re = '1031'.  
  
 SELECT SINGLE wrt\_kum FROM /afm/baure\_steu INTO lv\_wrt\_kum.  
  
 IF lv\_wrt\_kum IS NOT INITIAL.  
 cbkum = 'X'.  
 ENDIF.  
  
 ENDIF. *"Aufruf 1*  
  
 CLEAR erster\_aufruf.  
  
ENDFORM. *" read\_data*

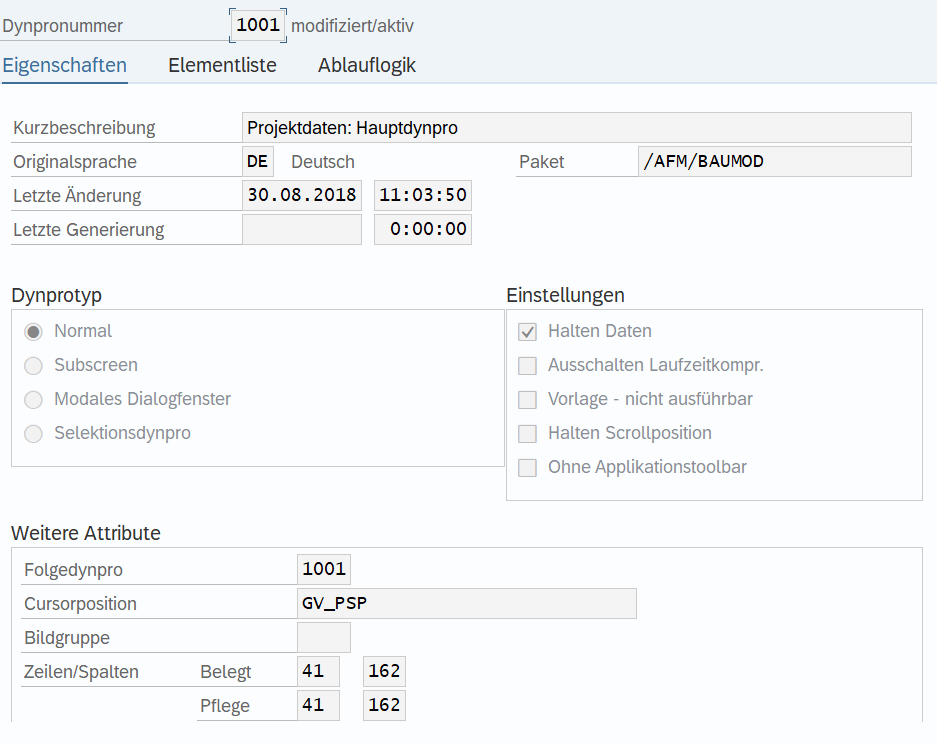
*\*&---------------------------------------------------------------------\**  
*\*& Form FILL\_BEZ*  
*\*&---------------------------------------------------------------------\**  
*\* Fülle Bez mit dem Eintrag vom Customizing oder Festwerten*  
*\*----------------------------------------------------------------------\**  
*\* -->P\_2019 text*  
*\* <--P\_WT\_FORD\_BEZ text*  
*\*----------------------------------------------------------------------\**  
FORM fill\_bez USING uv\_key TYPE /afm/key\_gegenf  
 CHANGING cv\_bez TYPE bez40.  
  
 DATA: ls\_ggn\_key TYPE /afm/bau\_ggn\_key.  
  
 READ TABLE gt\_ggn\_key WITH KEY ggn\_key = uv\_key INTO ls\_ggn\_key.  
 IF sy-subrc = 0.  
 cv\_bez = ls\_ggn\_key-bez.  
 ELSE.  
 CASE uv\_key.  
 WHEN 'BAUL'.  
 cv\_bez = 'Bauleistungsversicherung'.  
 WHEN 'KOST'.  
 cv\_bez = 'Energiekosten'.  
 WHEN 'REIN'.  
 cv\_bez = 'Schlussreinigung'.  
 ENDCASE.  
 ENDIF.  
  
ENDFORM.

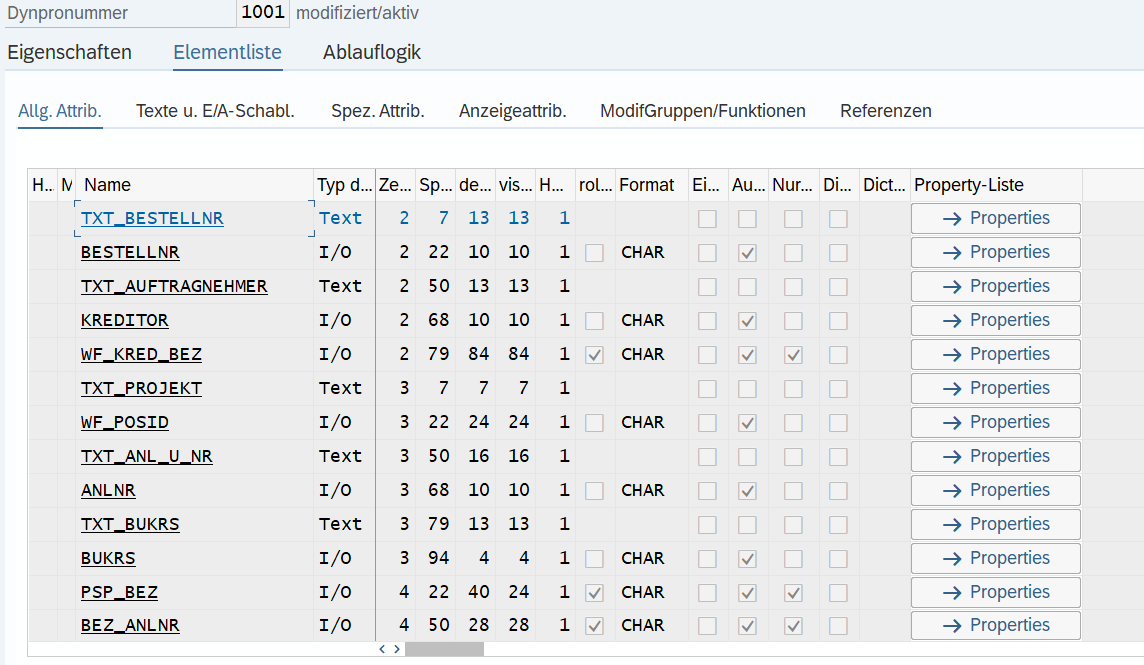
Include 🡪 /AFM/BAUMOD\_PBO\_01

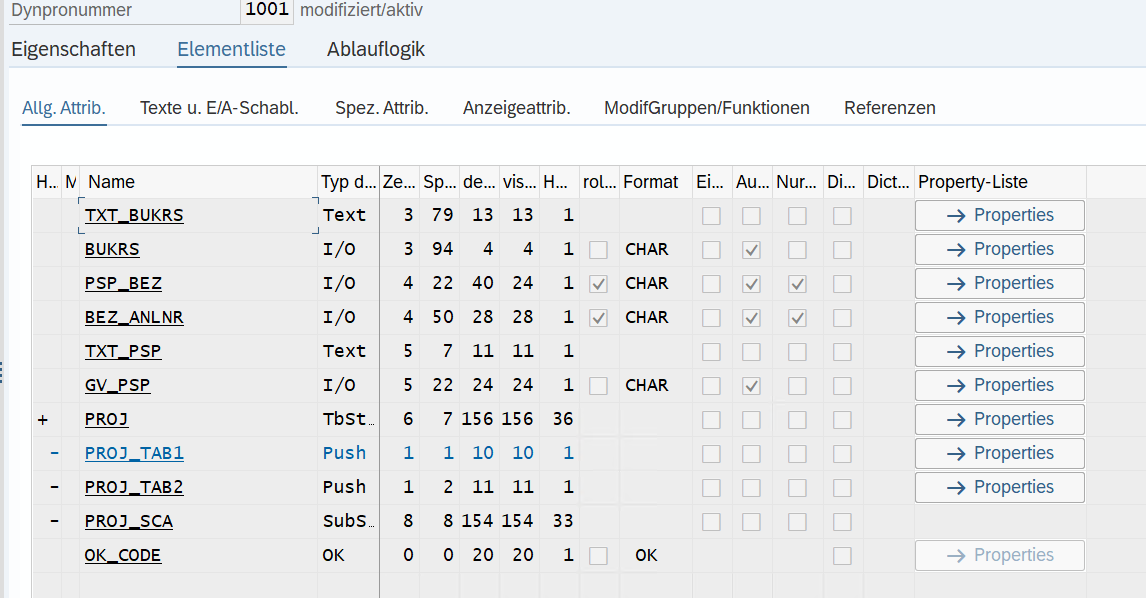
FORM check\_installment\_payment  
*\*Table und Data Definitionen*  
 TABLES ct\_projre LIKE wt\_projre[].  
  
 DATA: lt\_rbkp TYPE TABLE OF rbkp,  
 ls\_rbkp TYPE rbkp,  
 ls\_projre TYPE /afm/bauprojre,  
 ls\_projr2 TYPE /afm/bauprojre,  
 lv\_ans TYPE c LENGTH 1,  
 lv\_cache TYPE boolean,  
 lv\_nonaz TYPE boolean,  
 lv\_tabix LIKE sy-tabix,  
 lv\_subrc LIKE sy-subrc,  
 lv\_total TYPE bwert,  
 lv\_rmwwr TYPE rmwwr,  
 lv\_count TYPE i,  
 lv\_para TYPE c LENGTH 1.  
  
*\* Auf Verbuchen im Hintergrund warten (MIRO etc.)*  
 WAIT UP TO 1 SECONDS.  
  
*\*Auf gelöschte Belege prüfen (customizing egal)*  
 PERFORM check\_miro\_change  
 TABLES ct\_projre.  
  
*\*Prüfen ob Parameter gesetzt ist, wenn nicht beenden.*  
 SELECT SINGLE chk\_inst\_pay FROM /afm/baure\_steu INTO lv\_para  
 WHERE chk\_inst\_pay = 'X'.  
 IF sy-subrc <> 0.  
 EXIT.  
 ENDIF.  
  
*\*Sortierung Tabelle für Verarbeitungsreihenfolge*  
 SORT ct\_projre BY zaehler ASCENDING.  
  
*\*Ermittlung Gesamtbestellsumme*  
 READ TABLE ct\_projre INTO ls\_projre INDEX 1.  
 SELECT SUM( netwr ) FROM ekpo INTO lv\_total  
 WHERE ebeln = ls\_projre-ebeln.  
 LOOP AT ct\_projre INTO ls\_projre WHERE re\_art <> 'AZ'.  
 lv\_nonaz = 'X'.  
 ENDLOOP.  
 LOOP AT ct\_projre INTO ls\_projre.  
  
*\*Logik nur bei Abschlagszahlungen oder wenn Meldung mit Ja beantwortet wurde*  
 IF ( ls\_projre-re\_art = 'AZ' OR ( ls\_projre-re\_art = 'SR' AND ls\_projre-gegenf\_net IS INITIAL ) OR lv\_ans = '1' ) AND ( lv\_ans <> '2' AND lv\_ans <> 'A' ).  
  
*\*Bei Abschlagszahlungen gucken ob Bruttosumme in MIRO abweicht*  
 IF ls\_projre-re\_art = 'AZ' OR ( ls\_projre-re\_art = 'SR' AND ls\_projre-gegenf\_net IS INITIAL ).  
 IF lv\_nonaz IS INITIAL.  
 lv\_rmwwr = ls\_projre-ein\_re\_betr\_n\_1 \* ( 1 + ls\_projre-mwst\_1 / 100 ).  
 SELECT SINGLE \* FROM rbkp INTO ls\_rbkp  
 WHERE belnr = ls\_projre-miro\_mm\_belnr  
 AND gjahr = ls\_projre-gjahr  
 AND rmwwr <> lv\_rmwwr.  
 lv\_subrc = sy-subrc.  
 ELSEIF ls\_projre-re\_art = 'SR'.  
 lv\_rmwwr = ls\_projre-ein\_re\_betr\_n\_1 \* ( 1 + ls\_projre-mwst\_1 / 100 ).  
 SELECT SINGLE \* FROM rbkp INTO ls\_rbkp  
 WHERE belnr = ls\_projre-miro\_mm\_belnr  
 AND gjahr = ls\_projre-gjahr  
 AND rmwwr <> lv\_rmwwr.  
 lv\_subrc = sy-subrc.  
 ELSE.  
 lv\_subrc = 4.  
 ENDIF.  
 ENDIF.  
  
*\*Abfrage nur einmal bei abweichen, Baustein ändert tabix*  
 lv\_tabix = sy-tabix.  
 IF lv\_subrc = 0 AND lv\_cache IS INITIAL.  
 lv\_cache = 'X'.  
 CALL FUNCTION 'POPUP\_TO\_CONFIRM'  
 EXPORTING  
 titlebar = 'Information'  
 text\_question = TEXT-201  
 display\_cancel\_button = ' '  
 popup\_type = 'I'  
 IMPORTING  
 answer = lv\_ans  
 EXCEPTIONS  
 text\_not\_found = 1  
 OTHERS = 2.  
 ENDIF.  
  
*\*Abfrage lv\_ans = '1' für Durchläufe bis Fenster notwendig*  
 IF lv\_ans = '1'.  
  
*\*Vorherige Tabellenzeile zur kummulierung holen*  
 IF lv\_tabix > 1.  
 lv\_tabix = lv\_tabix - 1.  
 READ TABLE ct\_projre INTO ls\_projr2 INDEX lv\_tabix.  
 ENDIF.  
  
*\*Bei Abschlagszahlungen mit Abweichung mehr Werte anpassen*  
 IF ( ls\_projre-re\_art = 'AZ' OR ( ls\_projre-re\_art = 'SR' AND ls\_projre-gegenf\_net IS INITIAL ) ) AND lv\_subrc = 0.  
 ls\_projre-ein\_re\_betr\_n\_1 = ls\_rbkp-rmwwr / ( 1 + ls\_projre-mwst\_1 / 100 ).  
 IF ls\_projre-re\_art <> 'SR'. *"BEJK20190808 If und Else fuer Gewaehrleistung*  
 ls\_projre-einbeh\_einz\_n = ls\_projre-ein\_re\_betr\_n\_1 \* ( ls\_projre-erfuell\_sich / 100 ).  
 ls\_projre-freig\_betr\_n = ls\_projre-ein\_re\_betr\_n\_1 \* ( 1 - ls\_projre-erfuell\_sich / 100 ).  
 ls\_projre-freig\_betr\_b = ls\_projre-freig\_betr\_n \* ( 1 + ls\_projre-mwst\_1 / 100 ).  
 ENDIF.  
 ADD 1 TO lv\_count.  
 ENDIF.  
  
*\*Kummulierte Werte von allen Zeilen ab der ersten geänderten anpassen*  
 IF ls\_projre-re\_art <> 'SR'.  
 ls\_projre-einbeh\_einz\_b = ls\_projre-einbeh\_einz\_n \* ( 1 + ls\_projre-mwst\_1 / 100 ).  
 ls\_projre-einbeh\_kum\_n = ls\_projr2-einbeh\_kum\_n + ls\_projre-einbeh\_einz\_n.  
 ls\_projre-einbeh\_kum\_b = ls\_projr2-einbeh\_kum\_b + ls\_projre-einbeh\_einz\_b.  
 ELSE.  
 ls\_projre-einbeh\_einz\_n = ( ls\_projr2-ges\_leist\_net\_1 + ls\_projre-ein\_re\_betr\_n\_1 + ls\_projre-ges\_leist\_net\_2 ) \* ( ls\_projre-gewaehrleist / 100 ).  
 ls\_projre-einbeh\_einz\_b = ls\_projre-einbeh\_einz\_n \* ( 1 + ls\_projre-mwst\_1 / 100 ).  
 ls\_projre-freig\_betr\_n = ls\_projre-ein\_re\_betr\_n\_1 - ls\_projre-einbeh\_einz\_n.  
 ls\_projre-freig\_betr\_b = ls\_projre-freig\_betr\_n \* ( 1 + ls\_projre-mwst\_1 / 100 ).  
 ls\_projre-einbeh\_kum\_n = ls\_projr2-einbeh\_kum\_n.  
 ls\_projre-einbeh\_kum\_b = ls\_projr2-einbeh\_kum\_b.  
 ENDIF.  
 ls\_projre-ges\_leist\_net\_1 = ls\_projr2-ges\_leist\_net\_1 + ls\_projre-ein\_re\_betr\_n\_1.  
 ls\_projre-ges\_leist\_bru\_1 = ls\_projr2-ges\_leist\_bru\_1 + ls\_projre-ein\_re\_betr\_n\_1 \* ( 1 + ls\_projre-mwst\_1 / 100 ).  
*\* ls\_projre-prozent\_akt = ( ls\_projre-ges\_leist\_net\_1 + ls\_projre-ges\_leist\_net\_2 ) / lv\_total \* 100. "DEL ckn23062020*  
 *" ADD ckn23062020 - Start*  
 TRY .  
 ls\_projre-prozent\_akt = ( ls\_projre-ges\_leist\_net\_1 + ls\_projre-ges\_leist\_net\_2 ) / lv\_total \* 100.  
 CATCH cx\_root INTO DATA(oref).  
 *"Teilen durch 0 oder Overflow der Prozent\_akt Variable werden alle hier abgefangen*  
 MESSAGE oref->get\_text( ) TYPE 'E'.  
 ENDTRY.  
 *" ADD ckn23062020 - Ende*  
  
 IF ls\_projre-re\_art <> 'SR'.  
 ls\_projre-az\_kum\_net = ls\_projr2-az\_kum\_net + ls\_projre-ein\_re\_betr\_n\_1.  
 ls\_projre-az\_kum\_bru = ls\_projr2-az\_kum\_bru + ls\_projre-ein\_re\_betr\_n\_1 \* ( 1 + ls\_projre-mwst\_1 / 100 ).  
 ELSE.  
 ls\_projre-az\_kum\_net = ls\_projr2-az\_kum\_net + ls\_projre-freig\_betr\_n + ls\_projre-einbeh\_einz\_n.  
 ls\_projre-az\_kum\_bru = ls\_projr2-az\_kum\_bru + ls\_projre-freig\_betr\_b + ls\_projre-einbeh\_einz\_b.  
 ENDIF.  
*\*START 20190716 BEJK Protokollierung*  
 ls\_projre-uname\_change = sy-uname.  
 CALL FUNCTION 'ABI\_TIMESTAMP\_CONVERT\_INTO'  
 EXPORTING  
 iv\_date = sy-datlo  
 iv\_time = sy-timlo  
 IMPORTING  
 ev\_timestamp = ls\_projre-time\_change.  
*\*ENDE 20190716 BEJK*  
*\*Interne Tabelle modifizieren (speichern in Hauptprogramm)*  
 MODIFY ct\_projre FROM ls\_projre.  
 ENDIF.  
 ENDIF.  
 ENDLOOP.  
  
*\*Erfolgsmeldung ausgeben.*  
 IF lv\_ans = 1.  
 MESSAGE s006(/afm/bau) WITH lv\_count.  
 ENDIF.  
ENDFORM. *"check\_installment\_payment*

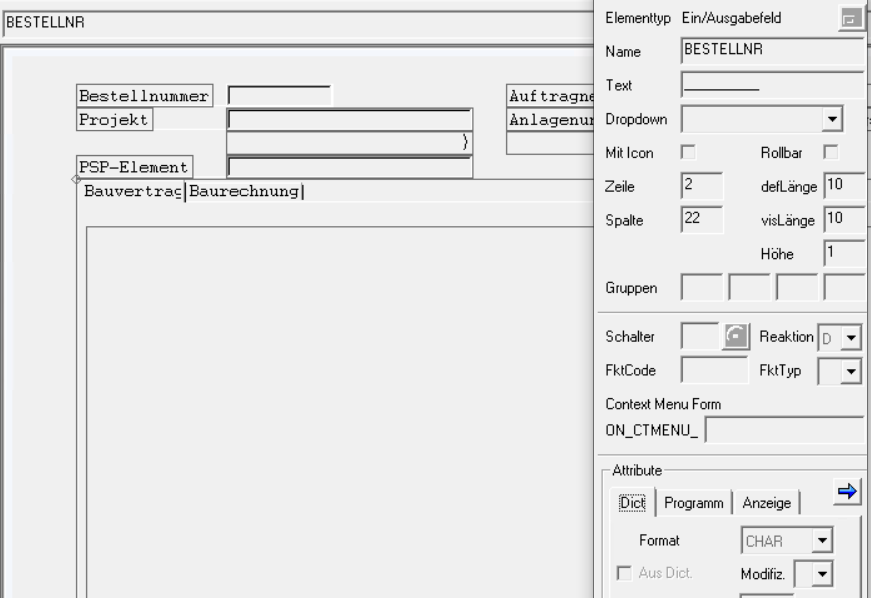
*\*&---------------------------------------------------------------------\**  
*\*& Form check\_miro\_change*  
*\*&---------------------------------------------------------------------\**  
*\* text*  
*\*----------------------------------------------------------------------\**  
*\* -->CT\_PROJRE text*  
*\*----------------------------------------------------------------------\**  
FORM check\_miro\_change  
 TABLES ct\_projre LIKE wt\_projre[].  
  
 DATA: ls\_projre TYPE /afm/bauprojre,  
 ls\_projr2 TYPE /afm/bauprojre,  
 ls\_projr3 TYPE /afm/bauprojre,  
 lt\_projre TYPE TABLE OF /afm/bauprojre,  
 ls\_rbkp TYPE rbkp,  
 lv\_total TYPE bwert,  
 lv\_change TYPE boolean,  
 lv\_delete TYPE boolean,  
 lv\_nonaz TYPE boolean,  
 lv\_tabix LIKE sy-tabix,  
 lv\_tabix2 LIKE sy-tabix,  
 lv\_lines TYPE i.  
  
*\*Sortierung Tabelle für Verarbeitungsreihenfolge*  
 SORT ct\_projre BY zaehler ASCENDING.  
  
*\*Ermittlung Gesamtbestellsumme*  
 READ TABLE ct\_projre INTO ls\_projre INDEX 1.  
 SELECT SUM( netwr ) FROM ekpo INTO lv\_total  
 WHERE ebeln = ls\_projre-ebeln.  
 LOOP AT ct\_projre INTO ls\_projre WHERE re\_art <> 'AZ'.  
 lv\_nonaz = 'X'.  
 ENDLOOP.  
 IF lv\_nonaz IS INITIAL.  
 LOOP AT ct\_projre INTO ls\_projre.  
 lv\_tabix2 = sy-tabix.  
 IF sy-tabix > 1.  
 lv\_tabix = sy-tabix - 1.  
 READ TABLE ct\_projre INTO ls\_projr2 INDEX lv\_tabix.  
 ENDIF.  
 SELECT SINGLE \* FROM rbkp INTO ls\_rbkp  
 WHERE belnr = ls\_projre-miro\_mm\_belnr  
 AND gjahr = ls\_projre-gjahr  
 AND rbstat = '2'.  
 IF sy-subrc = 0.  
 lv\_change = 'X'.  
 lv\_delete = 'X'.  
 READ TABLE ct\_projre INTO ls\_projr3 INDEX lv\_tabix2.  
 APPEND ls\_projr3 TO lt\_projre.  
 DELETE ct\_projre INDEX lv\_tabix2.  
 ENDIF.  
 IF lv\_change = 'X' AND lv\_delete <> 'X'.  
*\* Kummulierte Werte von allen Zeilen ab der ersten geänderten anpassen*  
 ls\_projre-einbeh\_einz\_b = ls\_projre-einbeh\_einz\_n \* ( 1 + ls\_projre-mwst\_1 / 100 ).  
 ls\_projre-einbeh\_kum\_n = ls\_projr2-einbeh\_kum\_n + ls\_projre-einbeh\_einz\_n.  
 ls\_projre-einbeh\_kum\_b = ls\_projr2-einbeh\_kum\_b + ls\_projre-einbeh\_einz\_b.  
 ls\_projre-ges\_leist\_net\_1 = ls\_projr2-ges\_leist\_net\_1 + ls\_projre-ein\_re\_betr\_n\_1.  
 ls\_projre-ges\_leist\_bru\_1 = ls\_projr2-ges\_leist\_bru\_1 + ls\_projre-ein\_re\_betr\_n\_1 \* ( 1 + ls\_projre-mwst\_1 / 100 ).  
 ls\_projre-prozent\_akt = ( ls\_projre-ges\_leist\_net\_1 + ls\_projre-ges\_leist\_net\_2 ) / lv\_total \* 100.  
 ls\_projre-az\_kum\_net = ls\_projr2-az\_kum\_net + ls\_projre-ein\_re\_betr\_n\_1.  
 ls\_projre-az\_kum\_bru = ls\_projr2-az\_kum\_bru + ls\_projre-ein\_re\_betr\_n\_1 \* ( 1 + ls\_projre-mwst\_1 / 100 ).  
*\* START 20190716 BEJK Protokollierung*  
 ls\_projre-uname\_change = sy-uname.  
 CALL FUNCTION 'ABI\_TIMESTAMP\_CONVERT\_INTO'  
 EXPORTING  
 iv\_date = sy-datlo  
 iv\_time = sy-timlo  
 IMPORTING  
 ev\_timestamp = ls\_projre-time\_change.  
*\* ENDE 20190716 BEJK*  
 MODIFY ct\_projre FROM ls\_projre.  
 ENDIF.  
 lv\_delete = ' '.  
 ENDLOOP.  
 ENDIF.  
 lv\_lines = lines( ct\_projre ).  
 IF lv\_lines = 0 AND lv\_change = 'X'.  
 gv\_delrechnung = 'X'.  
 ENDIF.  
  
 DELETE /afm/bauprojre FROM TABLE lt\_projre.  
  
ENDFORM. *"check\_miro\_change*

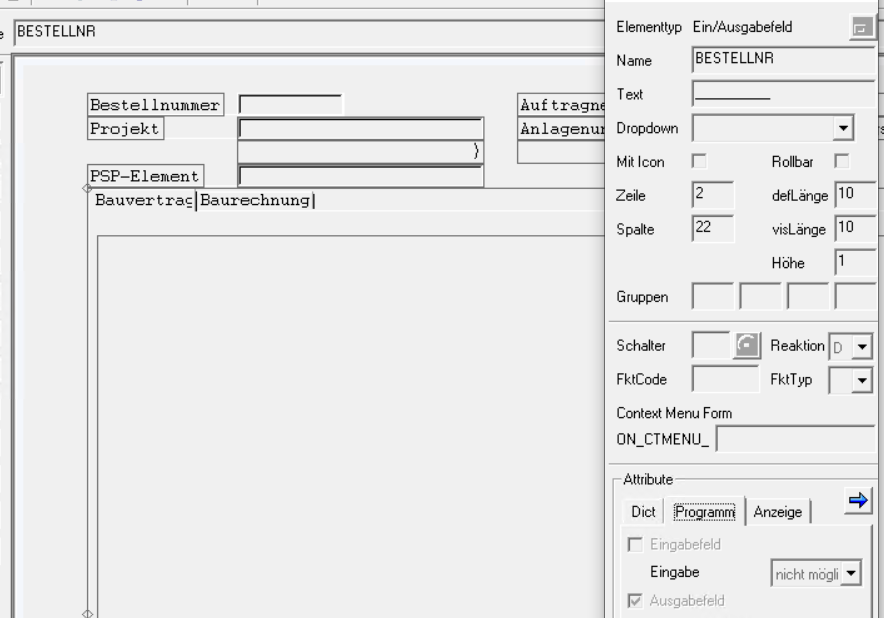
*Screen: 🡪 1001*

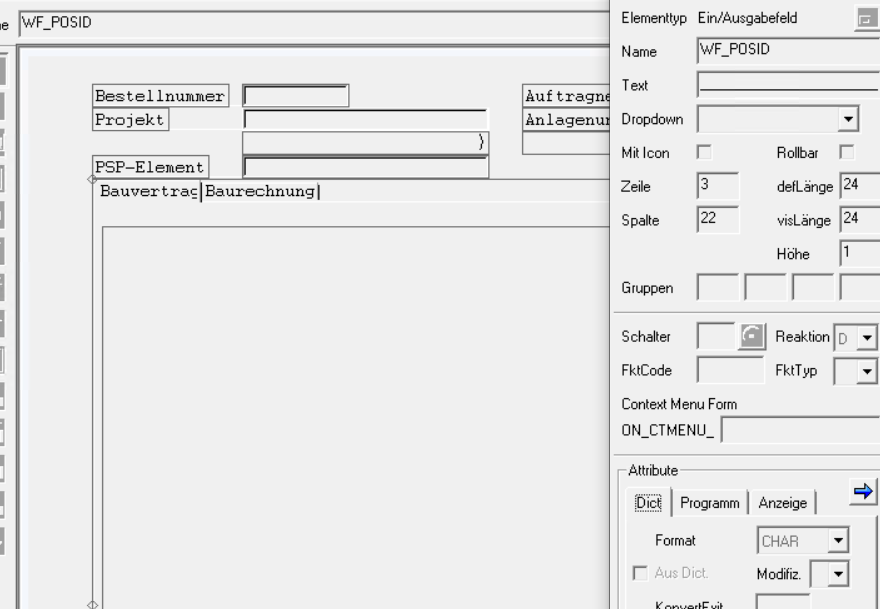


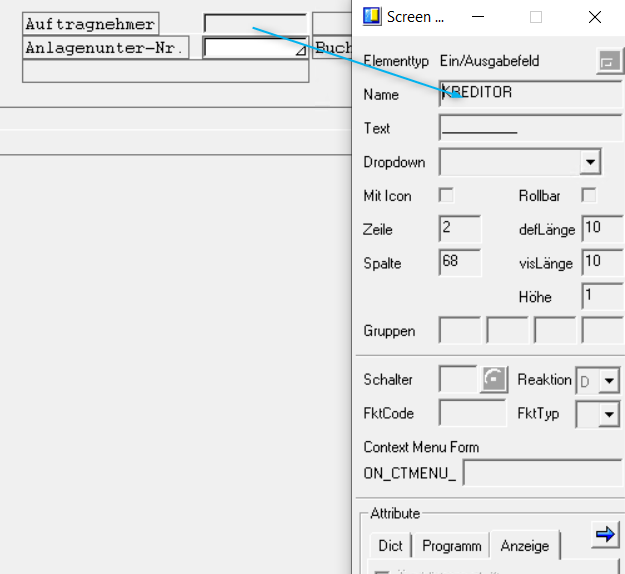


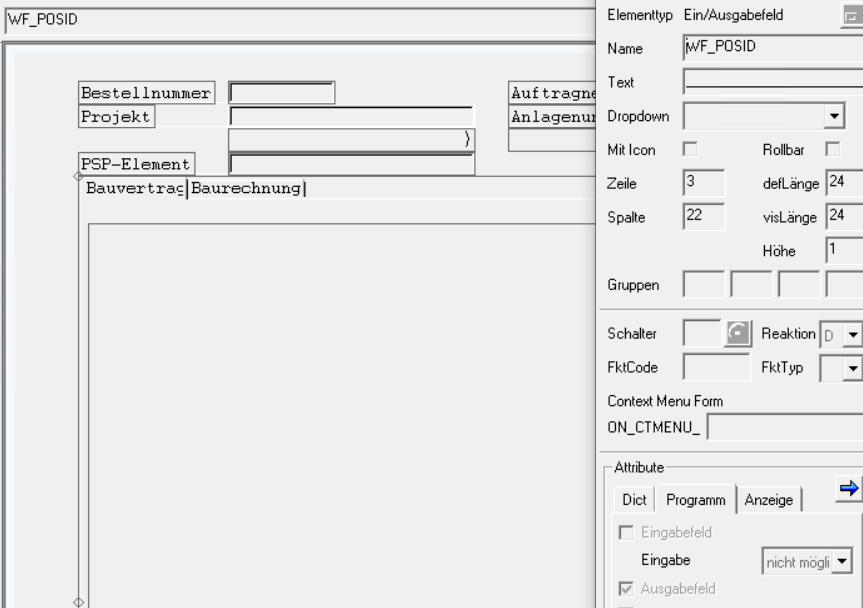


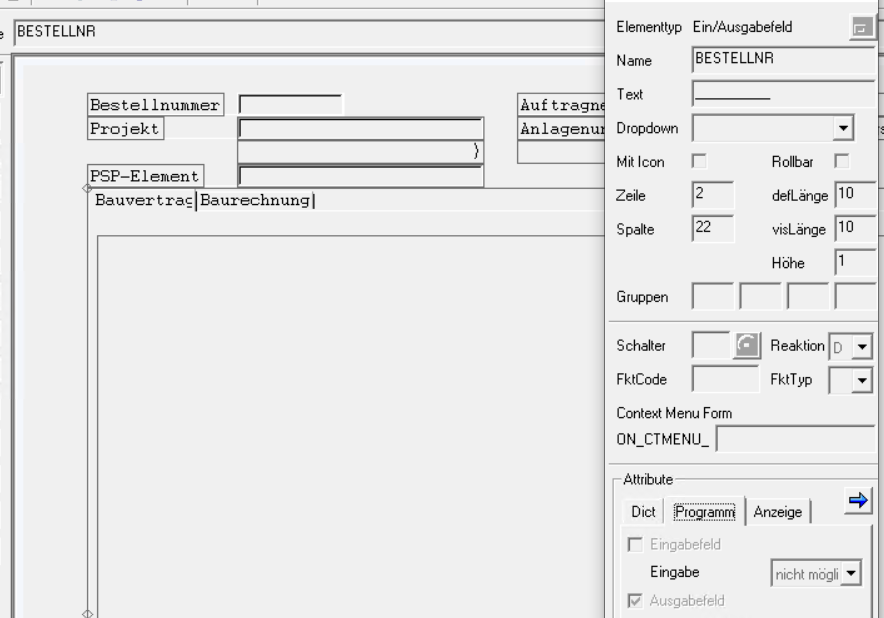


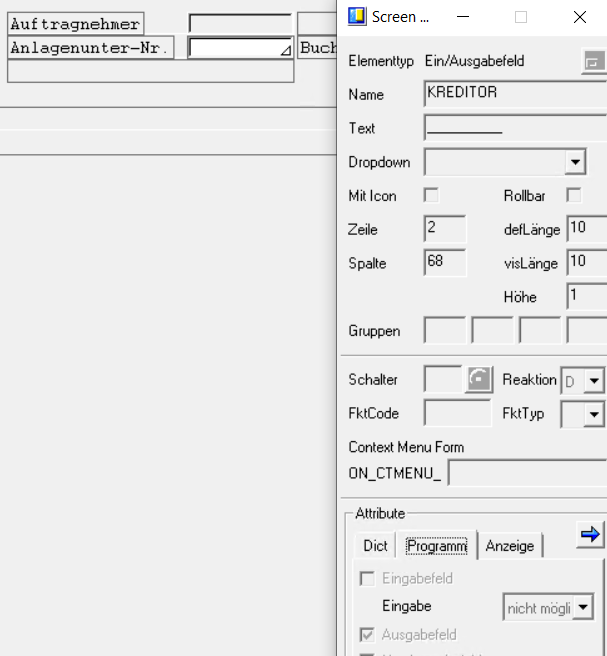


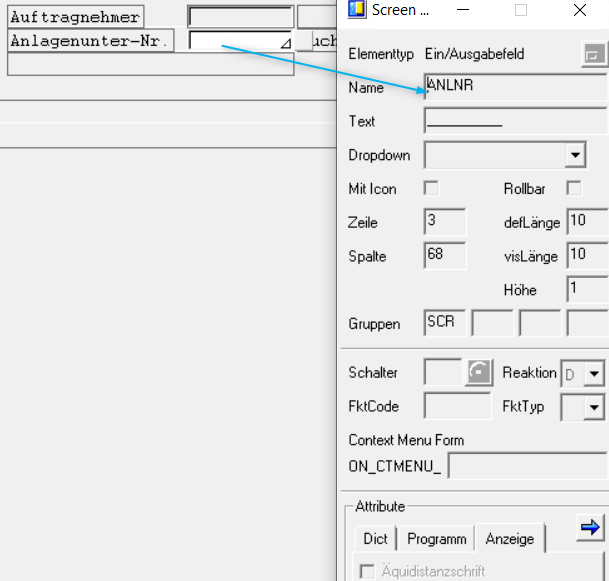


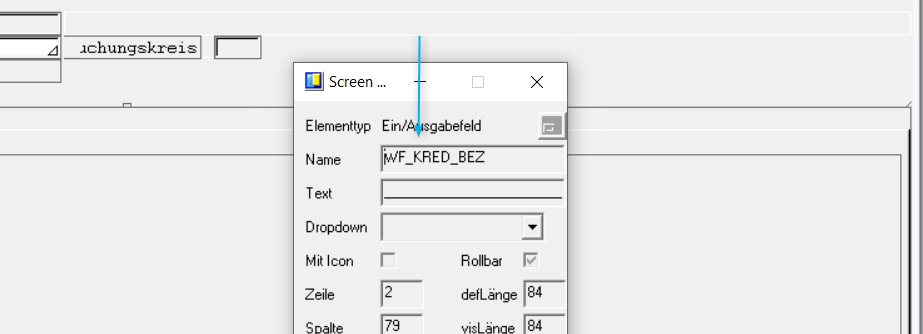


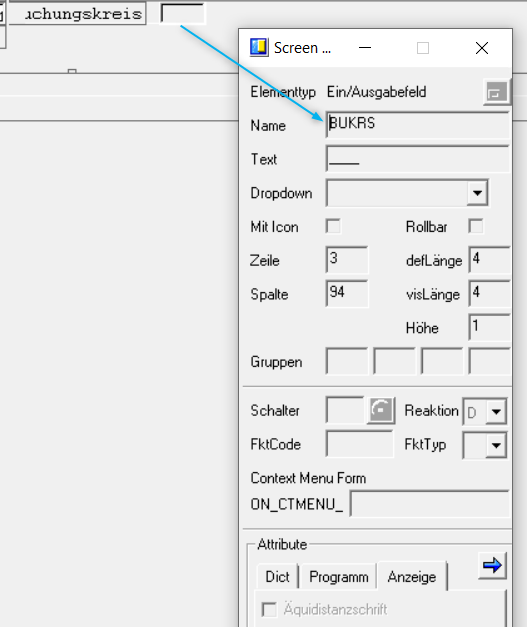


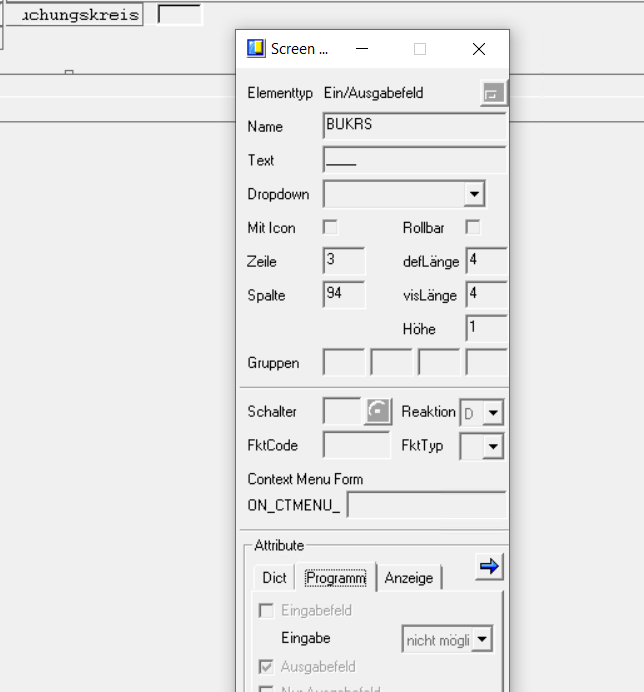


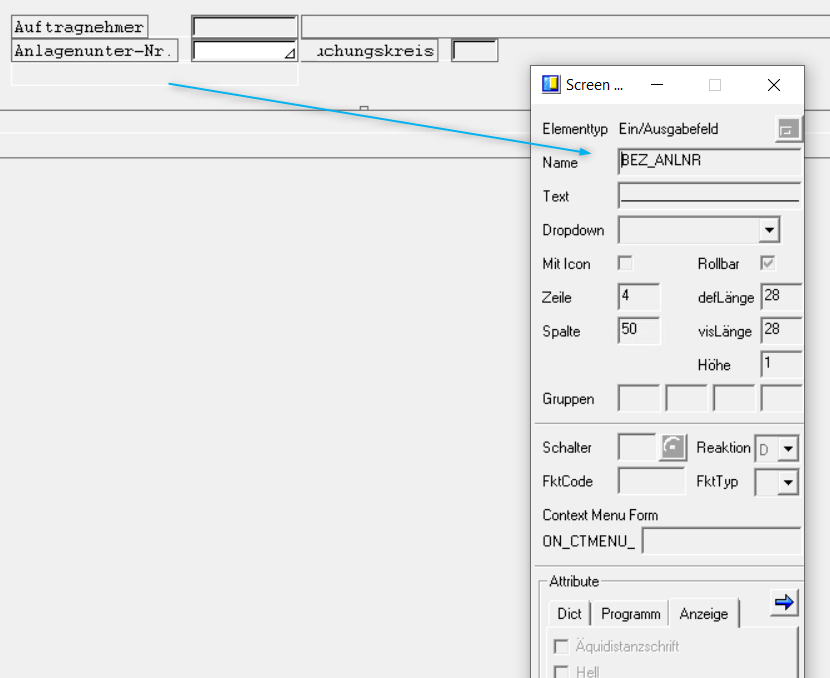




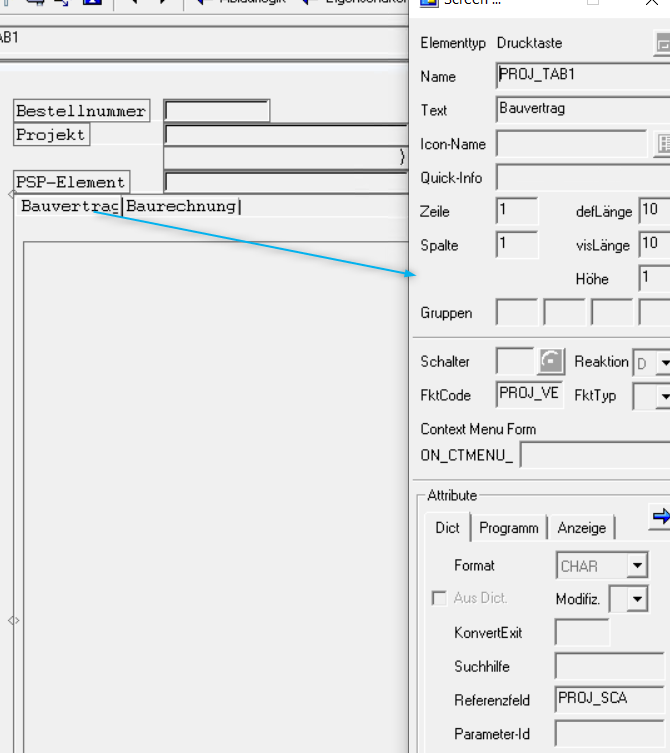


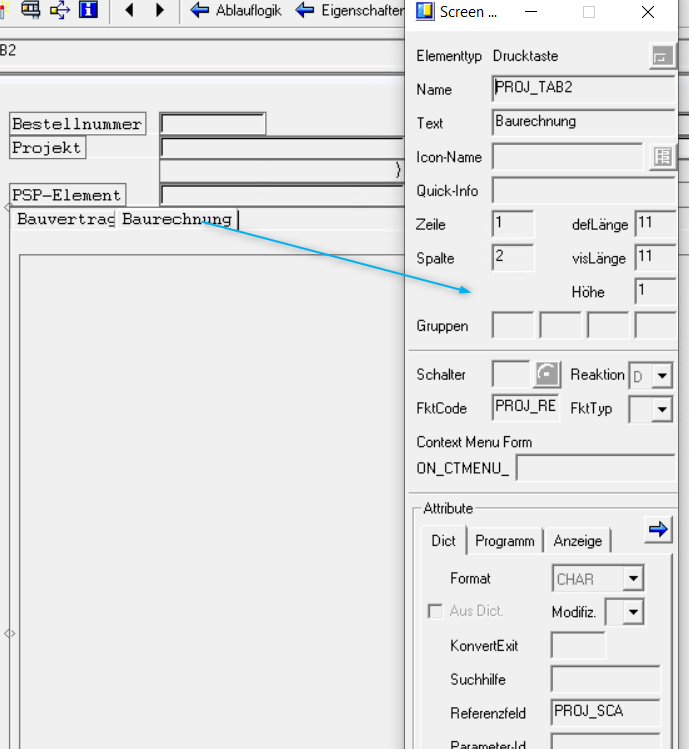


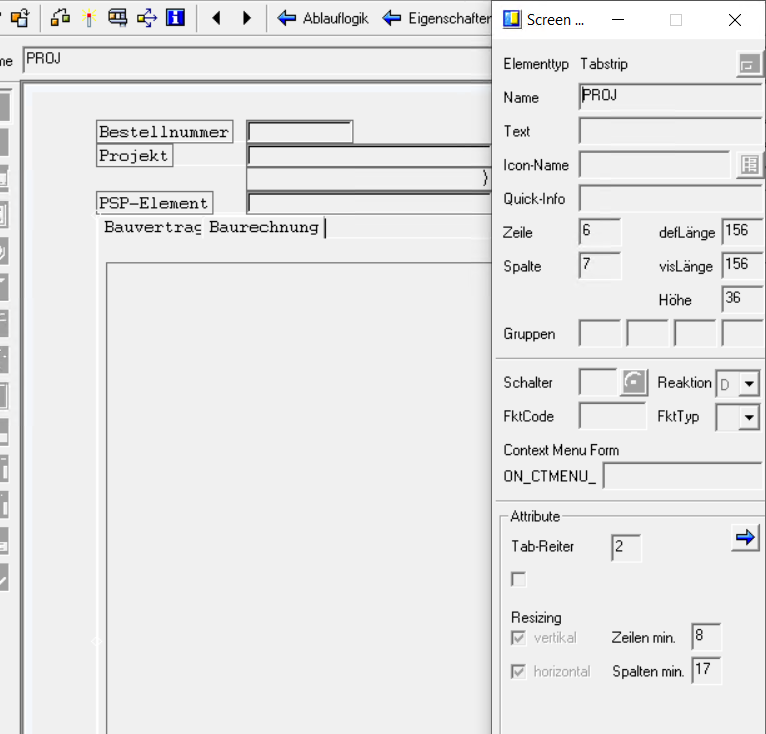


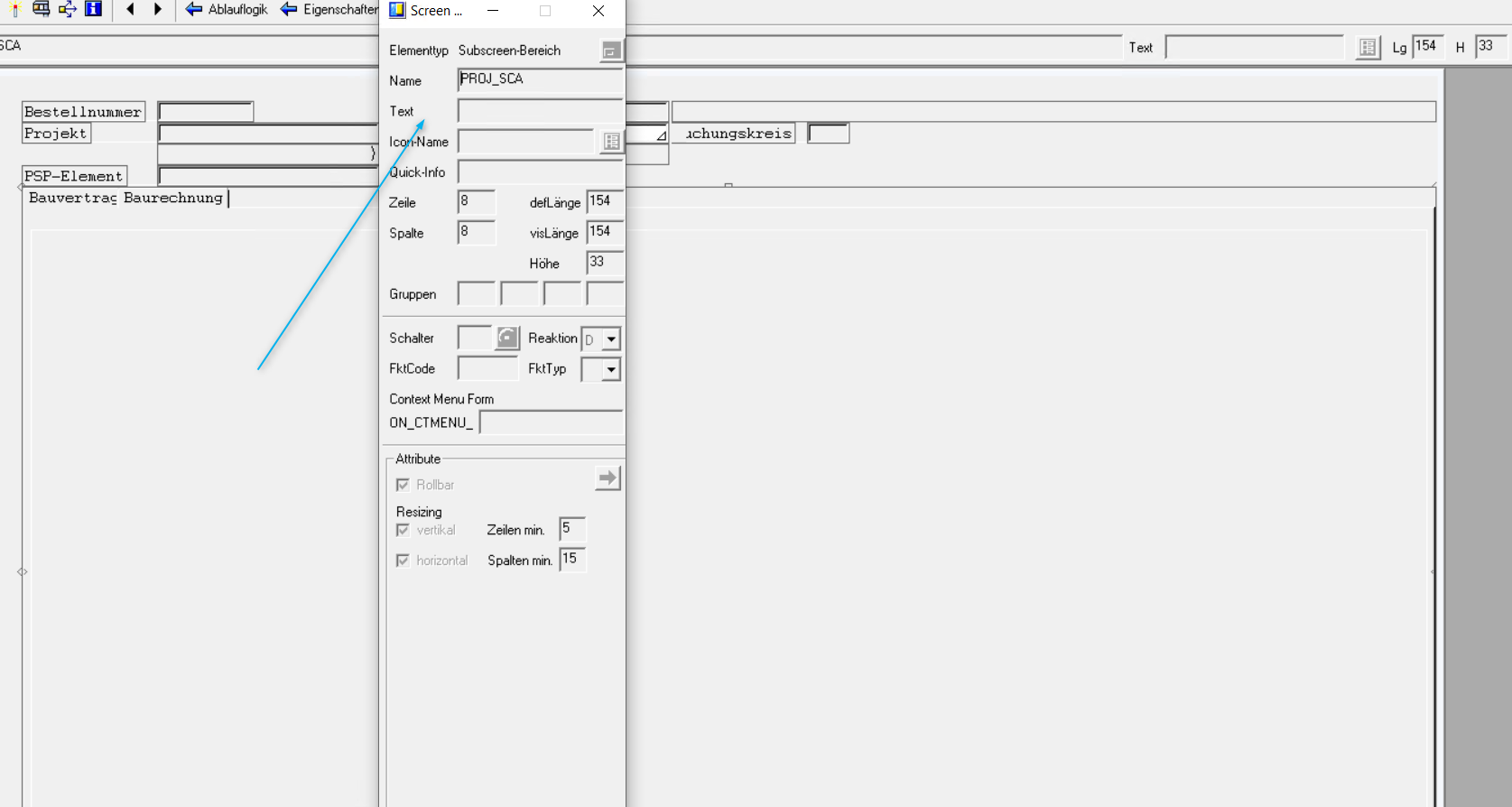








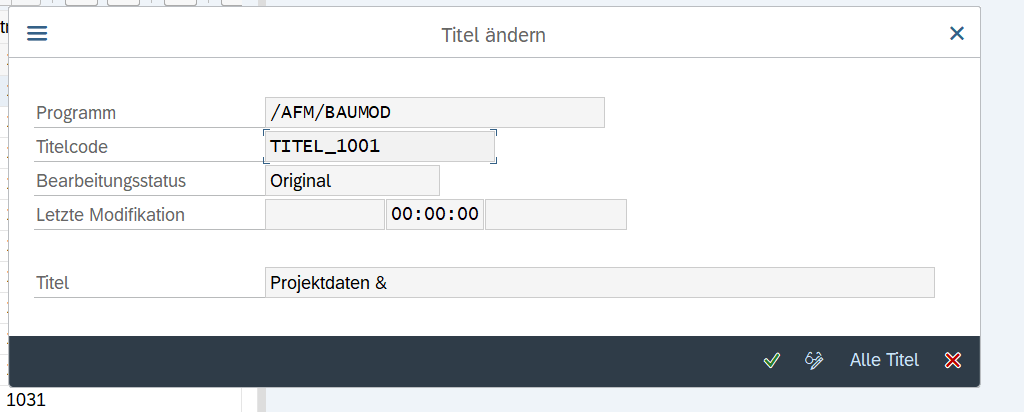


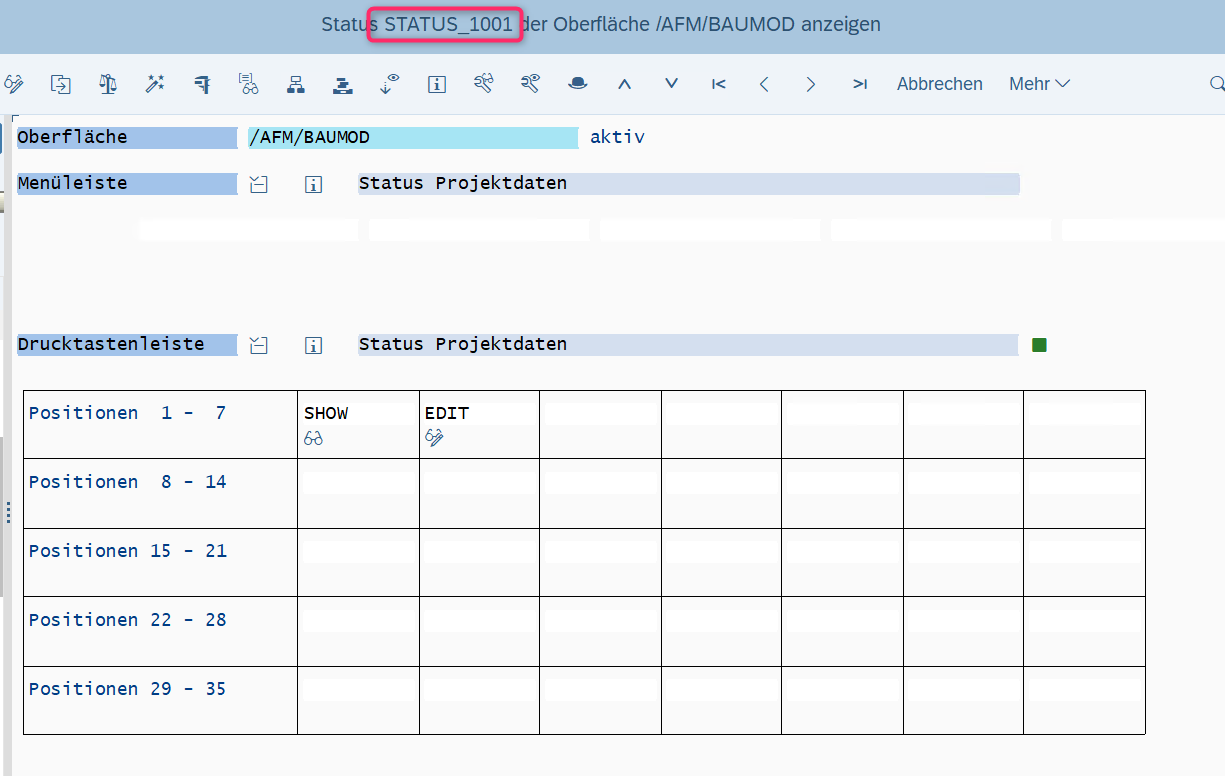


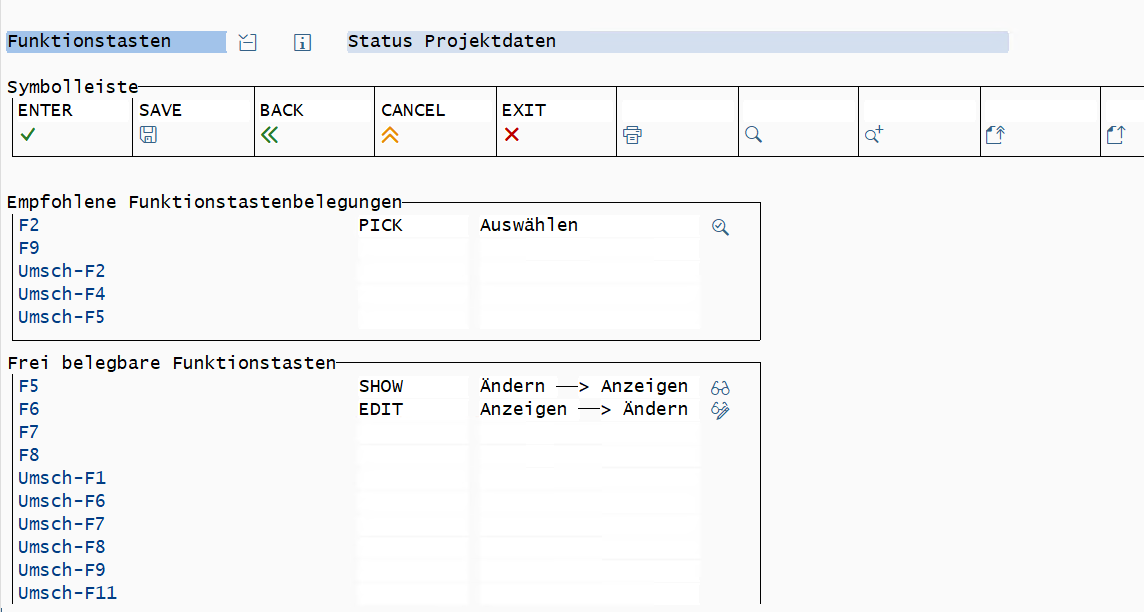
*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\**  
PROCESS BEFORE OUTPUT.  
*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\**  
  
 MODULE status\_1001.  
  
 MODULE modify\_screen\_1001.  
  
*\* PBO FLOW LOGIC FOR TABSTRIP 'PROJ'*  
 MODULE proj\_active\_tab\_set.  
 CALL SUBSCREEN proj\_sca  
 INCLUDING g\_proj-prog g\_proj-subscreen.  
  
  
*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\**  
PROCESS AFTER INPUT.  
*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\**  
  
*\* PAI FLOW LOGIC FOR TABSTRIP 'PROJ'*  
 CALL SUBSCREEN proj\_sca.  
 MODULE proj\_active\_tab\_get.  
  
 MODULE user\_command\_1001.

*\*&---------------------------------------------------------------------\**  
*\*& Module STATUS\_1001 OUTPUT*  
*\*&---------------------------------------------------------------------\**  
MODULE status\_1001 OUTPUT.  
  
 SET PF-STATUS 'STATUS\_1001'.  
 SET TITLEBAR '1001' WITH ''.  
  
*\*&---------------------------------------------------------------------\**  
  
 PERFORM status.  
  
*\*&---------------------------------------------------------------------\**  
  
 CLEAR wt\_baumod\_anl.  
 REFRESH wt\_baumod\_anl.  
  
*\* "STM07022012 Hierarchiesuche nach Umstellung auf Projektdef. aufgrund Mehrfachkontierung raus!!!*  
  
 IF anlnr NE space.  
 SELECT SINGLE \* FROM /afm/bau\_stammd WHERE pspnr EQ psp  
 AND ( anlnr NE space OR  
 anlnr2 NE space OR  
 anlnr3 NE space ).  
*\* nur wenn zum expliziten PSP auch mindestens eine Anlagenunternr.*  
*\* gepflegt ist, ansonsten nach höher liegendem PSP-Eintrag suchen (siehe*  
*\* else-Zweig) und dessen Anlagenunternummern ziehen.*  
 IF sy-subrc = 0.  
 IF /afm/bau\_stammd-anlnr EQ anlnr.  
 MOVE /afm/bau\_stammd-bez\_anlnr TO bez\_anlnr.  
 ELSEIF /afm/bau\_stammd-anlnr2 EQ anlnr.  
 MOVE /afm/bau\_stammd-bez\_anlnr2 TO bez\_anlnr.  
 ELSEIF /afm/bau\_stammd-anlnr3 EQ anlnr.  
 MOVE /afm/bau\_stammd-bez\_anlnr3 TO bez\_anlnr.  
 ELSE.  
 CLEAR bez\_anlnr.  
 ENDIF.  
  
 ENDIF.  
 ENDIF.  
  
ENDMODULE. *" STATUS\_1001 OUTPUT*

*\*&---------------------------------------------------------------------\**  
*\*& Form status*  
*\*&---------------------------------------------------------------------\**  
FORM status.  
  
 DATA lv\_subrc LIKE sy-subrc.  
  
 CLEAR t\_ucomm.  
  
*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\**  
*\* Felder zum ausblenden in die Tabelle t\_ucomm*  
*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\**  
  
 CASE sy-ucomm.  
  
 WHEN 'SHOW'.  
 APPEND 'SHOW' TO t\_ucomm.  
 CLEAR flag\_edit. *"Änderung nicht möglich*  
 SET PF-STATUS 'STATUS\_1001' EXCLUDING t\_ucomm.  
 SET TITLEBAR 'TITEL\_1001' WITH 'anzeigen'.  
  
 WHEN 'EDIT'.  
  
 PERFORM enqueue USING bestellnr CHANGING lv\_subrc.  
  
 IF lv\_subrc <> 0.  
 gesperrt = 'X'.  
 ELSE.  
 CLEAR gesperrt.  
 ENDIF.  
  
 IF gesperrt EQ space. *"kein Sperrobjekt aktiv=>Änderungen zulassen*  
 SET PF-STATUS 'STATUS\_1001' EXCLUDING t\_ucomm.  
*\* Berechtigungsabfrage auf Änderungsfunktion des Berecht.Objektes Z\_BAUMOD:*  
 CALL FUNCTION 'AUTHORITY\_CHECK'  
 EXPORTING  
*\* NEW\_BUFFERING = 3*  
 user = sy-uname  
 object = 'Z\_BAUMOD'  
 field1 = 'ACTVT'  
 value1 = '02'  
 EXCEPTIONS  
 user\_dont\_exist = 1  
 user\_is\_authorized = 2  
 user\_not\_authorized = 3  
 user\_is\_locked = 4  
 OTHERS = 5.  
 IF sy-subrc = 2.  
 gv\_auth\_check = 'X'.  
 SET TITLEBAR 'TITEL\_1001' WITH 'ändern'.  
 WRITE 'X' TO flag\_edit. *"Änderung möglich*  
 APPEND 'EDIT' TO t\_ucomm.  
 ELSE.  
 MESSAGE i000(38) WITH 'Sie haben keine Änderungsberechtigung!'.  
 APPEND 'SHOW' TO t\_ucomm.  
 ENDIF.  
 ELSE.  
*\* wenn Sperrobjekt aktiv ist, keine Änderungen zulassen=> Icon EDIT eingeblendet lassen*  
 APPEND 'SHOW' TO t\_ucomm.  
 ENDIF.  
 SET PF-STATUS 'STATUS\_1001' EXCLUDING t\_ucomm.  
  
 WHEN 'READ'.  
 IF flag\_edit EQ 'X'.  
 APPEND 'EDIT' TO t\_ucomm.  
 SET TITLEBAR 'TITEL\_1001' WITH 'ändern'.  
 ELSE.  
 APPEND 'SHOW' TO t\_ucomm.  
 SET TITLEBAR 'TITEL\_1001' WITH 'anzeigen'.  
 ENDIF.  
  
 SET PF-STATUS 'STATUS\_1001' EXCLUDING t\_ucomm.  
  
 WHEN 'BACK'. *"OTHERS.*  
 IF flag\_edit EQ 'X' AND gv\_auth\_check = 'X'.  
 APPEND 'EDIT' TO t\_ucomm.  
 SET PF-STATUS 'STATUS\_1001' EXCLUDING t\_ucomm.  
 SET TITLEBAR 'TITEL\_1001' WITH 'ändern'.  
 ELSE.  
 APPEND 'SHOW' TO t\_ucomm.  
 APPEND 'CALC' TO t\_ucomm.  
 SET PF-STATUS 'STATUS\_1001' EXCLUDING t\_ucomm.  
 SET TITLEBAR 'TITEL\_1001' WITH 'anzeigen'.  
 ENDIF.  
  
 WHEN 'OK\_CODE'.  
 IF flag\_edit EQ 'X' AND gv\_auth\_check = 'X'.  
 APPEND 'EDIT' TO t\_ucomm.  
 SET PF-STATUS 'STATUS\_1001' EXCLUDING t\_ucomm.  
 SET TITLEBAR 'TITEL\_1001' WITH 'ändern'.  
 ELSE.  
 APPEND 'SHOW' TO t\_ucomm.  
 SET PF-STATUS 'STATUS\_1001' EXCLUDING t\_ucomm.  
 SET TITLEBAR 'TITEL\_1001' WITH 'anzeigen'.  
 ENDIF.  
  
 WHEN OTHERS.  
 IF flag\_edit EQ 'X' AND gv\_auth\_check = 'X'.  
 APPEND 'EDIT' TO t\_ucomm.  
 SET PF-STATUS 'STATUS\_1001' EXCLUDING t\_ucomm.  
 SET TITLEBAR 'TITEL\_1001' WITH 'ändern'.  
 ELSE.  
 APPEND 'SHOW' TO t\_ucomm.  
 SET PF-STATUS 'STATUS\_1001' EXCLUDING t\_ucomm.  
 SET TITLEBAR 'TITEL\_1001' WITH 'anzeigen'.  
 ENDIF.  
  
 ENDCASE.  
  
 MODIFY SCREEN.  
  
ENDFORM. *" status*

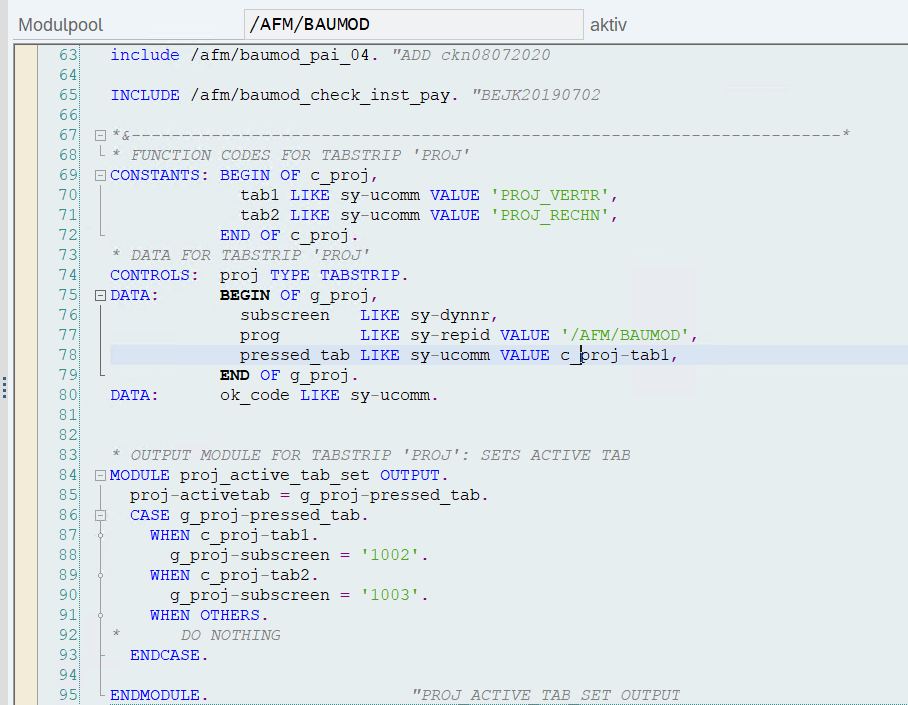
**

**

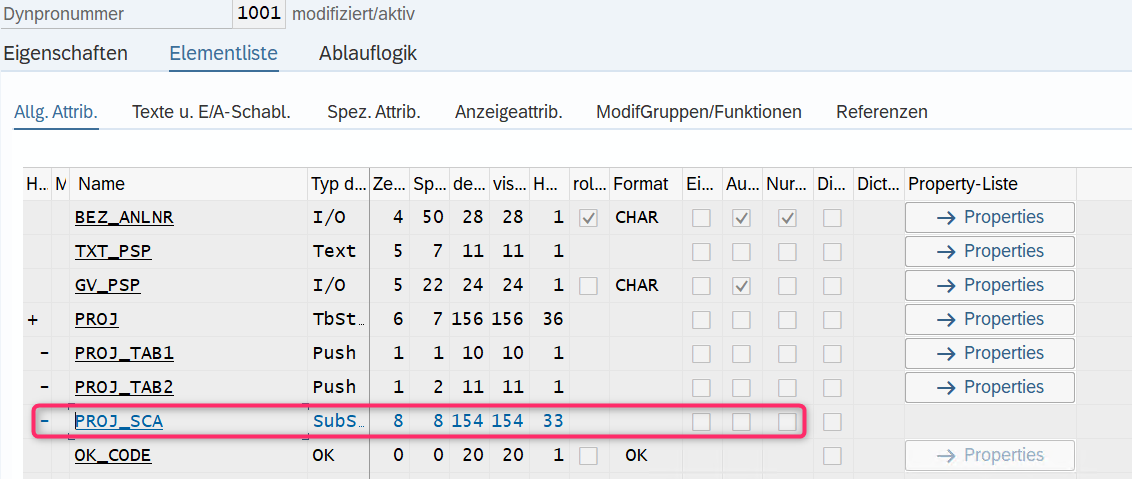
**

*\*&---------------------------------------------------------------------\**  
*\*& Form enqueue*  
*\*&---------------------------------------------------------------------\**  
*\* Tabellen bzgl.jew.Datensatz gegen Schreibzugriff sperren*  
*\*----------------------------------------------------------------------\**  
*\* --> p1 text*  
*\* <-- p2 text*  
*\*----------------------------------------------------------------------\**  
FORM enqueue USING p\_bestellnr CHANGING subrc. *" p\_psp.*  
  
 DATA: sperr\_1,  
 sperr\_2,  
 sperr\_3.  
  
 CALL FUNCTION 'ENQUEUE\_/AFM/BAUMOD'  
 EXPORTING  
*\* MODE\_/AFM/BAUPROJ = 'E'*  
 mandt = sy-mandt  
*\* PSPNR = p\_psp*  
 ebeln = p\_bestellnr  
*\* X\_PSPNR = ' '*  
*\* X\_EBELN = ' '*  
 \_scope = '1'  
*\* \_WAIT = ' '*  
*\* \_COLLECT = ' '*  
 EXCEPTIONS  
 foreign\_lock = 1  
 system\_failure = 2  
 OTHERS = 3.  
 subrc = sy-subrc.  
 CASE sy-subrc.  
 WHEN 0. *"Sperre erfolgreich durchgeführt*  
 WHEN 1. *"bereits gesperrt*  
 sperr\_1 = 'X'.  
*\* gesperrt = 'X'.*  
  
*\*\* System verwendet hier Klasse MC und Nr.601 MSGTYP 'E'*  
*\*\* => hierbei wird der System-User ins Feld msgv1 gestellt*  
*\* wf\_user = sy-msgv1.*  
*\* message i000(SU) with 'Projekt wird z.Zt. bearbeitet durch'*  
*\* wf\_user*  
*\* ' => '*  
*\* 'Nur Anzeige der Daten möglich!'.*  
 WHEN 2.  
 MESSAGE e000(su) WITH 'Interner Fehler beim Sperren!'.  
 LEAVE TO SCREEN 1000.  
 WHEN OTHERS.  
 CLEAR flag\_edit.  
 MESSAGE i000(su) WITH 'Sonstiger Fehler beim Sperren!'.  
 LEAVE TO SCREEN 1000.  
 ENDCASE.  
  
 CASE sy-subrc.  
 WHEN 0. *"Sperre erfolgreich durchgeführt*  
 WHEN 1. *"bereits gesperrt*  
 sperr\_1 = 'X'.  
 WHEN 2.  
 MESSAGE e000(su) WITH 'Interner Fehler beim Sperren!'.  
 LEAVE TO SCREEN 1000.  
 WHEN OTHERS.  
 CLEAR flag\_edit.  
 MESSAGE i000(su) WITH 'Sonstiger Fehler beim Sperren!'.  
 LEAVE TO SCREEN 1000.  
 ENDCASE.  
  
 IF sperr\_1 = 'X'.  
 gesperrt = 'X'.  
 CLEAR flag\_edit.  
*\* System verwendet bei diesen Sperrbausteinen die Klasse MC*  
*\* und die Nr.601 MSGTYP 'E'*  
*\* => hierbei wird der System-User ins Feld msgv1 gestellt*  
 wf\_user = sy-msgv1.  
 MESSAGE i000(su) WITH 'Projekt wird z.Zt. bearbeitet durch'  
 wf\_user  
 ' => '  
 'Nur Anzeige der Daten möglich!'.  
 ENDIF.  
  
ENDFORM. *" enqueue*

*\*&---------------------------------------------------------------------\**  
*\*& Module MODIFY\_SCREEN\_1001 OUTPUT*  
*\*&---------------------------------------------------------------------\**  
MODULE modify\_screen\_1001 OUTPUT.  
  
*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\**  
*\* Daten Anzeigen*  
*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\**  
 IF sy-ucomm EQ 'SHOW'. *"zur Ansicht*  
 CLEAR flag\_edit.  
*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\**  
*\* Daten Ändern*  
*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\**  
 ELSEIF sy-ucomm EQ 'EDIT'.  
 IF gesperrt EQ space. *"kein Sperrobjekt aktiv*  
 MOVE 'X' TO flag\_edit.  
 PERFORM check\_installment\_payment *"BEJK20190702 siehe FORM HEADER*  
 TABLES wt\_projre.  
 ENDIF.  
 ENDIF.  
  
 LOOP AT SCREEN.  
 IF flag\_edit EQ 'X' AND gv\_auth\_check = 'X'.  
 IF screen-group1 EQ 'SCR'.  
 MOVE '1' TO screen-input. *" Eingabe möglich*  
 MODIFY SCREEN.  
 ENDIF.  
 ENDIF.  
 ENDLOOP.  
  
ENDMODULE. *" MODIFY\_SCREEN\_1001 OUTPUT*

**

*\* OUTPUT MODULE FOR TABSTRIP 'PROJ': SETS ACTIVE TAB*  
MODULE proj\_active\_tab\_set OUTPUT.  
 proj-activetab = g\_proj-pressed\_tab.  
 CASE g\_proj-pressed\_tab.  
 WHEN c\_proj-tab1.  
 g\_proj-subscreen = '1002'.  
 WHEN c\_proj-tab2.  
 g\_proj-subscreen = '1003'.  
 WHEN OTHERS.  
*\* DO NOTHING*  
 ENDCASE.  
  
ENDMODULE. *"PROJ\_ACTIVE\_TAB\_SET OUTPUT*



*\* INPUT MODULE FOR TABSTRIP 'PROJ': GETS ACTIVE TAB*  
MODULE proj\_active\_tab\_get INPUT.  
 ok\_code = sy-ucomm.  
 CASE ok\_code.  
 WHEN c\_proj-tab1.  
 g\_proj-pressed\_tab = c\_proj-tab1.  
 WHEN c\_proj-tab2.  
 g\_proj-pressed\_tab = c\_proj-tab2.  
 WHEN OTHERS.  
*\* DO NOTHING*  
 ENDCASE.  
ENDMODULE. *"PROJ\_ACTIVE\_TAB\_GET INPUT*

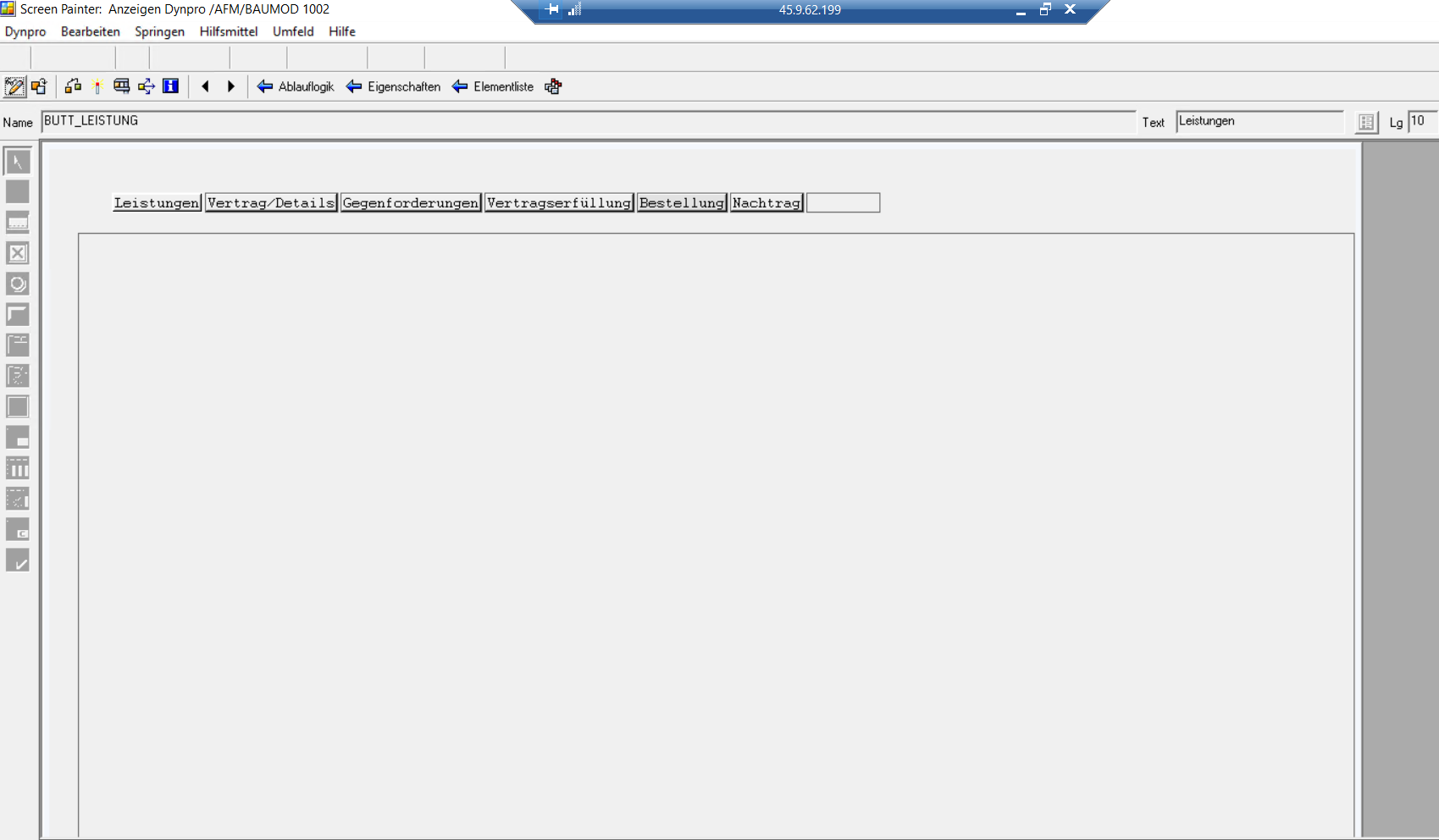
*Include 🡪 /AFM/BAUMOD\_PAI\_01*

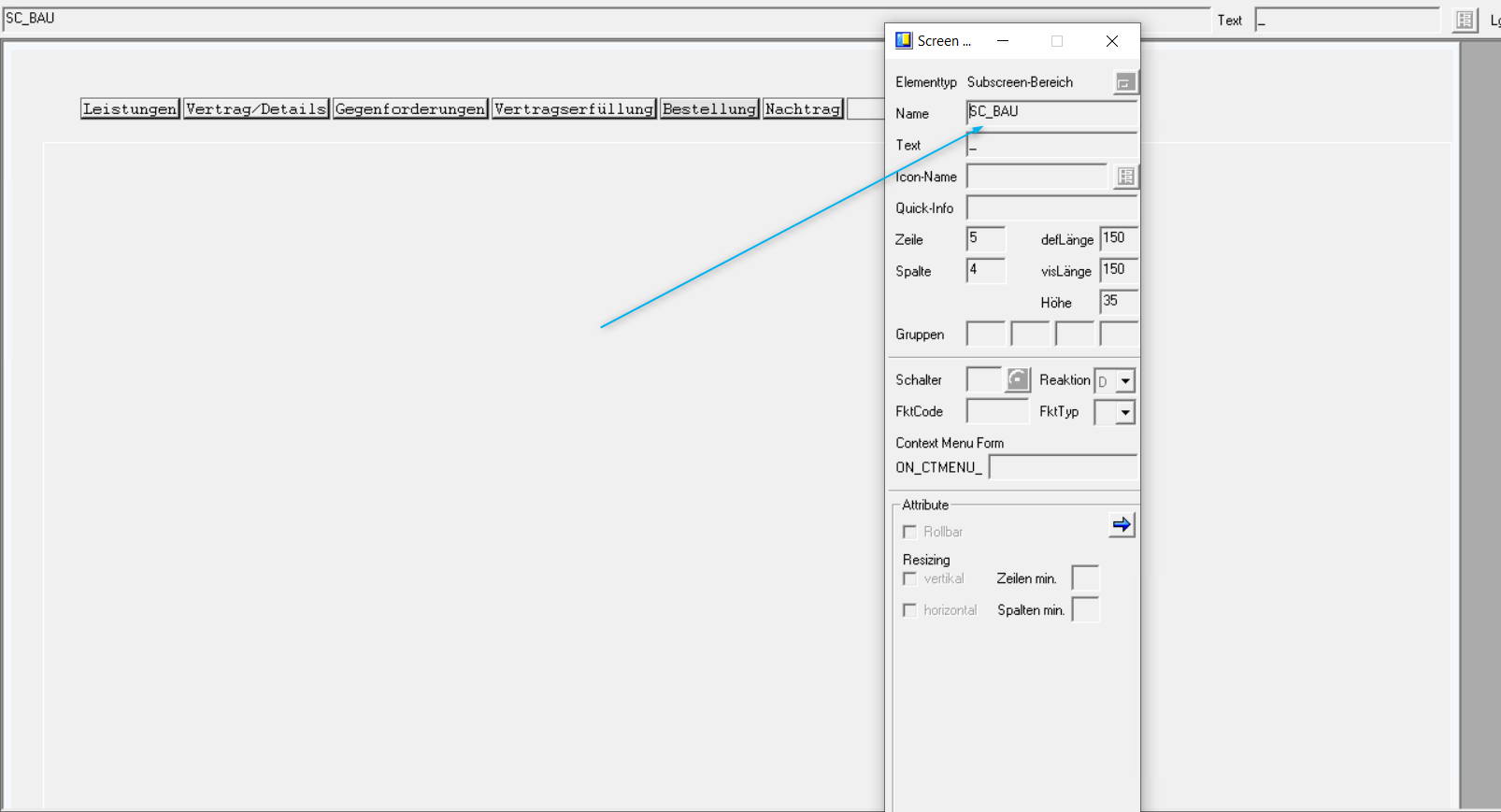
*\*----------------------------------------------------------------------\**  
*\* INCLUDE /AFM/BAUMOD\_PAI\_01 \**  
*\*----------------------------------------------------------------------\**  
*\*&---------------------------------------------------------------------\**  
*\*& Module USER\_COMMAND\_1001 INPUT*  
*\*&---------------------------------------------------------------------\**  
MODULE user\_command\_1001 INPUT.  
  
*\* PSP-Element bzgl. Suchhilfe /AFM/baumod, bzw.an den darin enthaltenen*  
*\* Suchhilfe\_Exit /AFM/BAUMOD\_ANL an Parameter ID übergeben.*  
 SET PARAMETER ID '/AFM/PSP' FIELD psp.  
  
*\* Lesen der Bezeichnung zur ausgesuchten Anlagenunternummer*  
*\* if anlnr ne space.*  
*\* select single \* from /AFM/BAU\_STAMMD where anlnr eq anlnr.*  
*\* if sy-subrc = 0.*  
*\* move /AFM/BAU\_STAMMD-anlnr to bez\_anlnr*  
*\* endif.*  
  
*\*if anlnr ne space.*  
*\* get parameter id '/AFM/BEZ\_ANLNR' field bez\_anlnr.*  
*\*endif.*  
  
 CASE sy-ucomm.  
  
 WHEN 'BACK' OR 'CANCEL'.  
 IF gv\_doc\_open\_b = 'X'.  
 PERFORM close\_and\_cleanup USING go\_document\_b  
 go\_worddoc\_b  
 go\_control\_b  
 gv\_doc\_open\_b  
 gv\_kz\_vorl\_b  
 gv\_url\_b  
 'B'.  
 ENDIF.  
  
*\* Bestehende Sperren zu Bestellung auflösen*  
 PERFORM dequeue USING bestellnr. *" psp.*  
 clear flag\_edit.  
 LEAVE TO SCREEN 1000.  
  
 WHEN 'EXIT'.  
 PERFORM popup\_to\_decide USING answer.  
 CASE answer.  
 WHEN 'A'.  
*\* G\_PROJ-SUBSCREEN = '1003'.*  
 subscr\_re = '1035'.  
 sy-ucomm = 'PROJ\_RECHN'.  
 LEAVE TO SCREEN '1001'.  
  
 WHEN '1'.  
 PERFORM datensicherung. *" Sicherung Bauvertragsdaten*  
 PERFORM free\_objects.  
 IF gesperrt NE 'X'. *"d.h. nicht bereits durch anderen User ge-*  
 *"sperrt, sondern durch diesen Lauf. Somit*  
 *"muß hier wieder entsperrt werden*  
 PERFORM dequeue USING bestellnr. *" psp.*  
 ENDIF.  
 LEAVE PROGRAM.  
  
 WHEN '2'.  
 PERFORM free\_objects.  
 IF gesperrt NE 'X'.  
 PERFORM dequeue USING bestellnr. *" psp.*  
 ENDIF.  
 LEAVE PROGRAM.  
 ENDCASE.  
  
*\* WHEN 'CANCEL'.*  
*\*\* perform exit.*  
*\* LEAVE TO SCREEN 1000.*  
  
*\* Absprung in die jeweilige Bestellung*  
 WHEN 'PICK'.  
 GET CURSOR FIELD fld.  
 IF fld = 'BESTELLNR'.  
 SET PARAMETER ID 'BES' FIELD bestellnr.  
 CALL TRANSACTION 'ME23N'.  
  
 erster\_aufruf = 'X'.  
 PERFORM read\_data.  
  
 ENDIF.  
  
 WHEN 'SAVE'.  
 PERFORM datensicherung. *" Sicherung Bauvertragsdaten*  
  
*\*\*\* Absprung per Drucktaste in die Stammdatenpflege in der Tabelle*  
*\*\*\* /AFM/BAU\_STAMMD auf Wunsch entfernt*  
*\**  
*\* when 'STAMMDATEN'.*  
*\*\* Absprung in Pflegedialog der allgemeinen Stammdaten.*  
*\*\* dazu Parametertransaktion bzgl. SM30 angelegt*  
*\* call transaction '/AFM/BAUMODSTAMM'.*  
*\**  
*\*\*\*\*\*\*\* Stammdaten neu ziehen*  
*\*\* wenn noch keine Rechnung erfasst wurde, also bei der ersten Bearbei-*  
*\*\* tung die Stammdaten aus dem Tool heraus gepflegt werden, diese dann*  
*\*\* auch gleich übernehmen:*  
*\**  
*\* CALL FUNCTION 'CONVERSION\_EXIT\_ABPSP\_INPUT'*  
*\* EXPORTING*  
*\* INPUT = wf\_posid "CHAR 24*  
*\* IMPORTING*  
*\* OUTPUT = psp\_int. "NUMC 8*  
*\**  
*\*select single \* from /AFM/BAUPROJRE where ebeln eq bestellnr*  
*\* and pspnr eq psp\_int.*  
*\**  
*\* if sy-subrc ne 0. "noch keine Rechnung => Erstbearbeitung*  
*\**  
*\*\* Defaultwerte zum Projekt bzw. Standardwerte*  
*\*clear: wa\_bauvertr,*  
*\* it\_baumod,*  
*\* it\_baumod[],*  
*\* wt\_ford,*  
*\* wt\_ford[],*  
*\* treffer\_psp.*  
*\**  
*\* select single \* from /AFM/BAU\_STAMMD where pspnr eq psp.*  
*\* if sy-subrc = 0.*  
*\* treffer\_psp = 'X'.*  
*\* move-corresponding /AFM/BAU\_STAMMD to wa\_bauvertr.*  
*\* move wa\_bauvertr-anlnr to anlnr.*  
*\* else.*  
*\*\* select single \* from /AFM/BAU\_STAMMD where pspnr eq '00000168'.*  
*\* SELECT \* FROM /AFM/BAU\_STAMMD client specified into it\_baumod*  
*\* WHERE MANDT = sy-mandt.*  
*\**  
*\* CALL FUNCTION 'CONVERSION\_EXIT\_ABPSP\_OUTPUT'*  
*\* EXPORTING*  
*\* INPUT = it\_baumod-pspnr "NUMC 8*  
*\* IMPORTING*  
*\* OUTPUT = it\_posid. "CHAR 24*  
*\**  
*\* move it\_posid to it\_baumod-posid. "CHAR 24*  
*\**  
*\* append it\_baumod.*  
*\* endselect.*  
*\**  
*\* sort it\_baumod descending by posid.*  
*\**  
*\* loop at it\_baumod.*  
*\*\* len = strlen( it\_posid ).*  
*\* len = strlen( it\_baumod-posid ).*  
*\*\* if it\_posid = wf\_posid(len).*  
*\* if it\_baumod-posid = wf\_posid(len).*  
*\* treffer\_psp = 'X'.*  
*\* move-corresponding it\_baumod to wa\_bauvertr.*  
*\* move /AFM/BAU\_STAMMD-anlnr to anlnr.*  
*\* exit.*  
*\* endif.*  
*\* endloop.*  
*\* endif.*  
*\*\* wenn keine Werte zum PSP ermittelt werden konnten, die Defaultwerte*  
*\*\* des Dummy-PSP A.ZZ-Z einlesen*  
*\* if treffer\_psp ne 'X'.*  
*\* CALL FUNCTION 'CONVERSION\_EXIT\_ABPSP\_INPUT'*  
*\* EXPORTING*  
*\* INPUT = 'A.ZZ-Z' "CHAR 24*  
*\* IMPORTING*  
*\* OUTPUT = psp\_elem. "NUMC 8*  
*\**  
*\**  
*\* select single \* from /AFM/BAU\_STAMMD where pspnr eq psp\_elem.*  
*\* if sy-subrc = 0.*  
*\* move-corresponding /AFM/BAU\_STAMMD to wa\_bauvertr.*  
*\* move /AFM/BAU\_STAMMD-anlnr to anlnr.*  
*\* endif.*  
*\* endif.*  
*\**  
*\**  
*\**  
*\**  
*\*\* Itab wt\_ford bzw. TC tc\_ford mit Bauleistungsvericherungs- und*  
*\*Kosten-*  
*\*\* umlage -Prozentsätzen vorbelegen.*  
*\**  
*\* move: 'KOST' to wt\_ford-key,*  
*\* 'Kostenumlage' to wt\_ford-bez,*  
*\* '%' to wt\_ford-als,*  
*\* wa\_bauvertr-kostenumlage to wt\_ford-proz,*  
*\*\* wa\_bauvertr-mwst to wt\_ford-mwst.*  
*\*\* MWST-Wert aus Bestellung einsetzen*  
*\* mwst to wt\_ford-mwst.*  
*\*\* Werte für Kostenumlage und Bauleistungsversicherung werden erst bei*  
*\*\* der Schlußrechnung ermittelt, und zwar auf den vorliegenden*  
*\*Leistungs-*  
*\*\* stand nach Abzug Aufschlag.*  
*\**  
*\* append wt\_ford. clear wt\_ford.*  
*\**  
*\* move: 'BAUL' to wt\_ford-key,*  
*\* 'Bauleistungsversicherung' to wt\_ford-bez,*  
*\* '%' to wt\_ford-als,*  
*\* wa\_bauvertr-bauleist\_vers to wt\_ford-proz,*  
*\*\* wa\_bauvertr-mwst to wt\_ford-mwst.*  
*\* mwst to wt\_ford-mwst.*  
*\**  
*\* append wt\_ford. clear wt\_ford.*  
*\**  
*\**  
*\*\* Defaultwerte ins Dynpro 1022 setzen*  
*\* move: wa\_bauvertr-erfuell\_sich to erfuell\_sich,*  
*\* wa\_bauvertr-gewaehrleist to gewaehrleist,*  
*\* wa\_bauvertr-gewaehrl\_dauer to gewaehrl\_dauer,*  
*\* wa\_bauvertr-straf\_nach to straf\_nach,*  
*\* wa\_bauvertr-straf\_max to straf\_max.*  
*\*\* write wa\_bauvertr-aufschlag to aufschlag.*  
*\**  
*\*\* replace ',' with '.' into aufschlag.*  
*\**  
*\* if wa\_bauvertr-straf\_in\_proz eq 'X'.*  
*\* move: wa\_bauvertr-vertr\_straf to vertr\_straf\_betr,*  
*\* 'X' to radio\_proz.*  
*\* clear radio\_betr.*  
*\* else.*  
*\* move: wa\_bauvertr-vertr\_straf\_betr to vertr\_straf\_betr,*  
*\* 'X' to radio\_betr.*  
*\* clear radio\_proz.*  
*\* endif.*  
*\**  
*\* if wa\_bauvertr-straf\_pro\_kal eq 'X'.*  
*\* move 'X' to radio\_kal.*  
*\* clear radio\_wer.*  
*\* else.*  
*\* move 'X' to radio\_wer.*  
*\* clear radio\_kal.*  
*\* endif.*  
*\**  
*\* endif. "sy-subrc*  
*\*\*\*\* ende Stammdaten neu ziehen*  
*\**  
*\**  
  
 ENDCASE.  
  
*\* CASE sy-ucomm.*  
*\*\* CASE fcode.*  
*\* WHEN 'RAUF' OR 'RUNTER' OR 'ERSTE' OR 'LETZTE'.*  
*\* clear fcode.*  
*\* PERFORM SUPRESS\_F31(SAPLIQS0).*  
*\* WHEN OTHERS.*  
*\* ENDCASE.*  
  
ENDMODULE. *" USER\_COMMAND\_1001 INPUT*

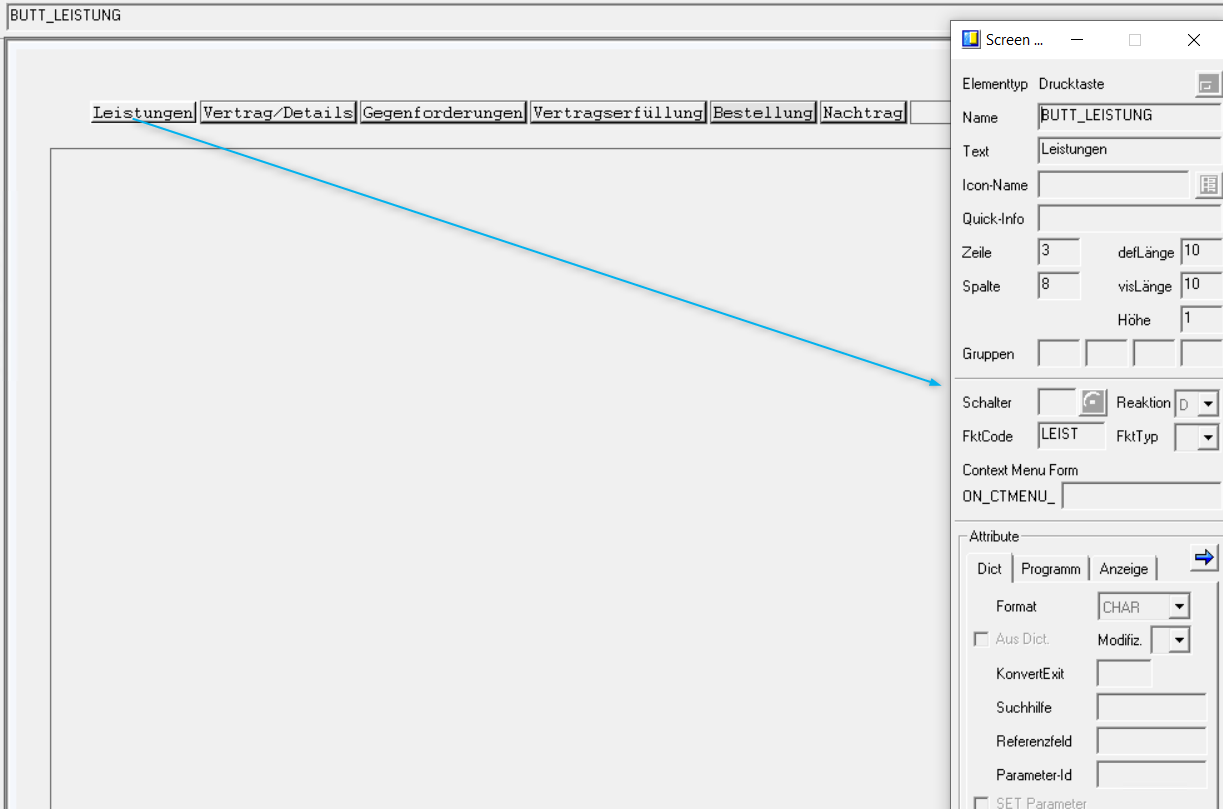
*include🡪 /AFM/BAUMOD\_F\_BDS*

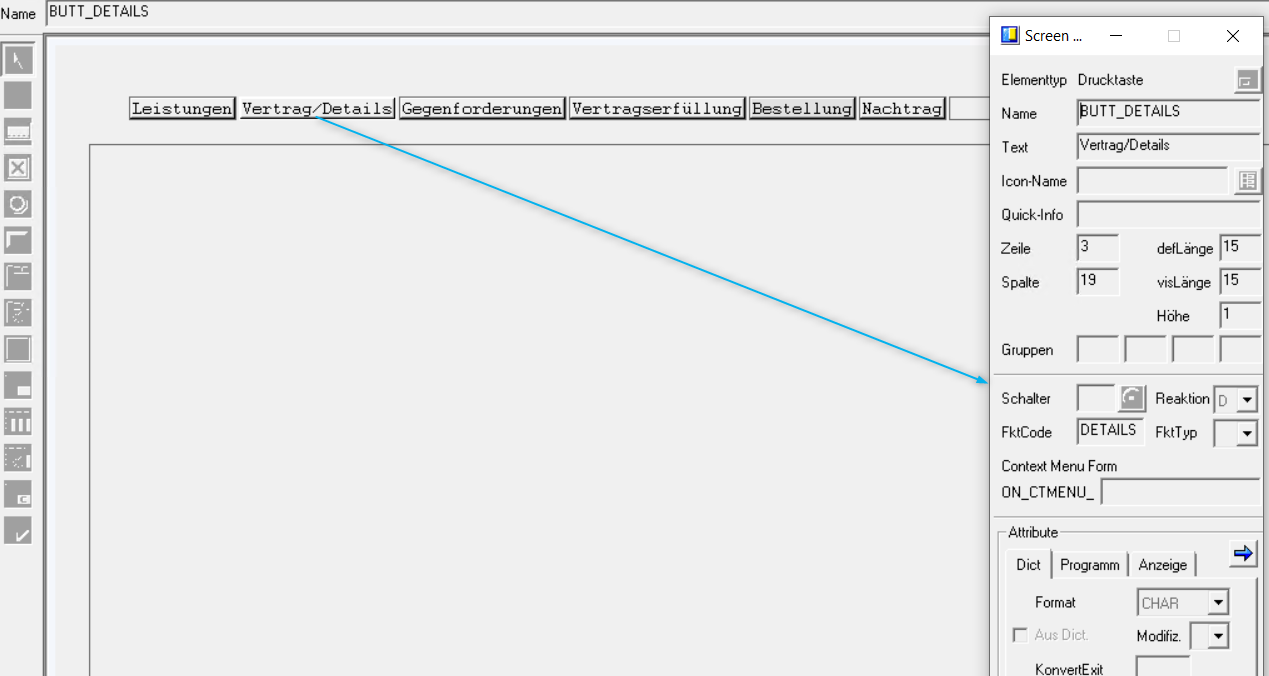
*\*&---------------------------------------------------------------------\**  
*\*& Form CLOSE\_AND\_CLEANUP*  
*\*&---------------------------------------------------------------------\**  
*\* -->P\_GO\_DOCUMENT\_N text*  
*\* -->P\_GO\_WORDDOC\_N text*  
*\* -->P\_GO\_CONTROL\_N text*  
*\* -->P\_GV\_DOC\_OPEN\_N text*  
*\* -->P\_GV\_KZ\_VORL\_N text*  
*\* -->P\_GV\_URL\_N text*  
*\* -->P\_0847 text*  
*\*----------------------------------------------------------------------\**  
 FORM close\_and\_cleanup USING io\_document TYPE REF TO i\_oi\_document\_proxy  
 io\_worddoc TYPE REF TO i\_oi\_word\_processor\_document  
 io\_control TYPE REF TO i\_oi\_container\_control  
 iv\_doc\_open  
 iv\_kz\_vorl  
 iv\_url  
 iv\_art.  
  
 REFRESH gt\_signatures.  
 REFRESH gt\_components.  
 REFRESH gt\_uris.  
  
 IF NOT io\_document IS INITIAL.  
 CALL METHOD io\_document->close\_document.  
 ENDIF.  
  
 CLEAR iv\_doc\_open.  
 CLEAR iv\_kz\_vorl. *"Ob Vorlage oder nicht immer clearen*  
 CLEAR iv\_url.  
  
 IF NOT io\_control IS INITIAL.  
 CALL METHOD io\_control->destroy\_control  
 EXPORTING  
 no\_flush = ' '.  
 ENDIF.  
  
 FREE io\_worddoc.  
 CLEAR io\_worddoc.  
  
 IF NOT io\_document IS INITIAL.  
 CALL METHOD io\_document->release\_document.  
 ENDIF.  
  
 FREE: io\_document,  
 io\_control.  
  
 CLEAR: io\_document,  
 io\_control.  
  
 ENDFORM. *" CLOSE\_AND\_CLEANUP*

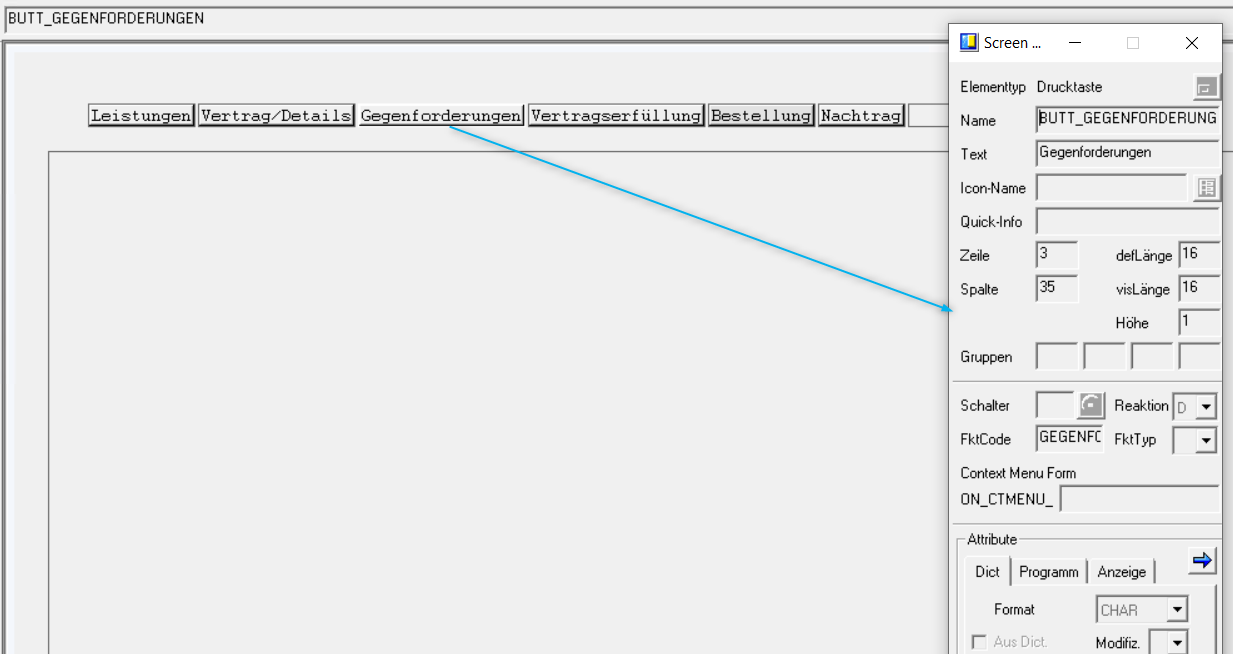
*Screen: 🡪 1002*

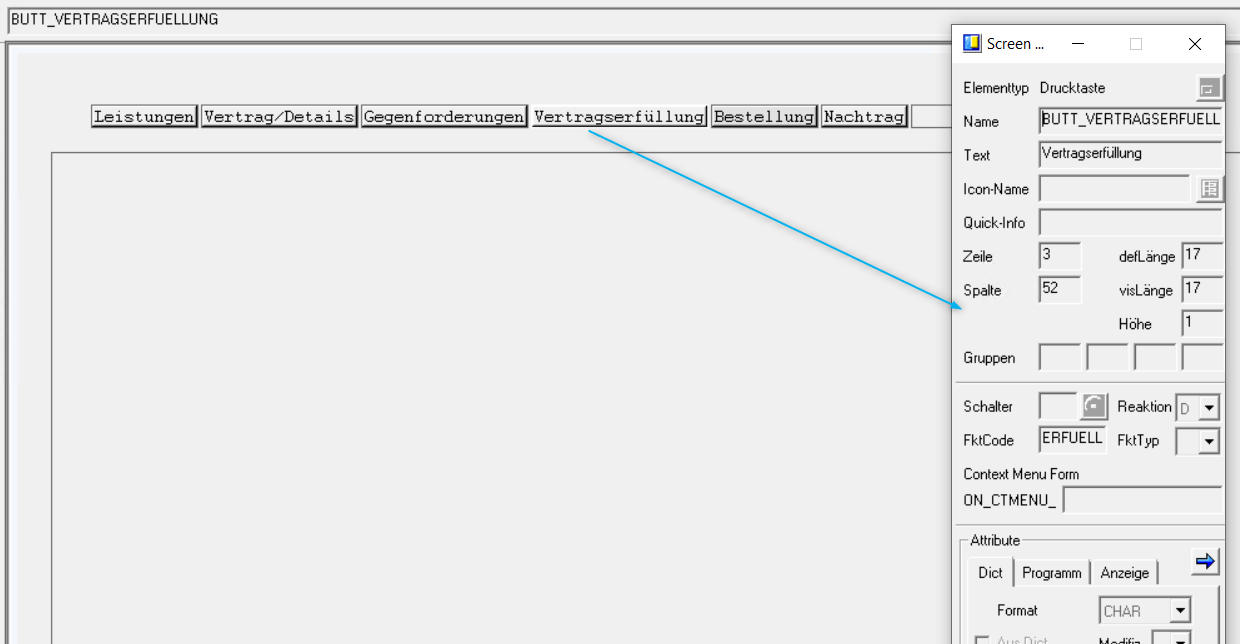


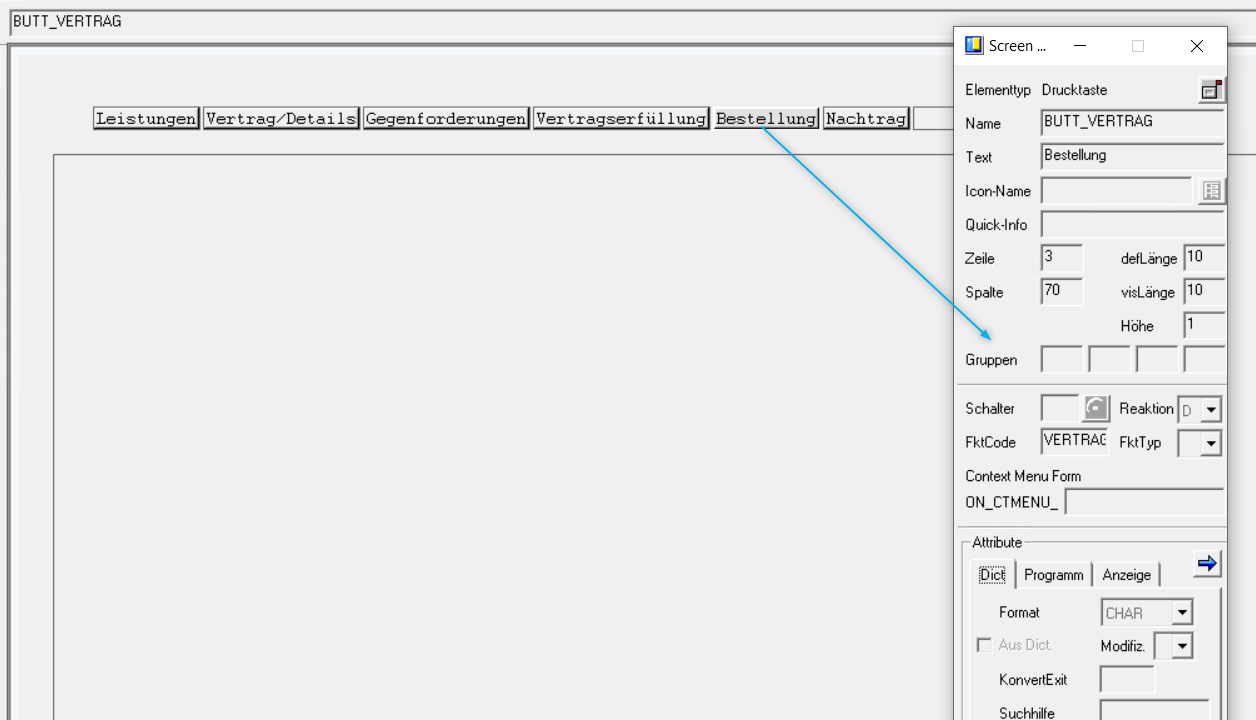


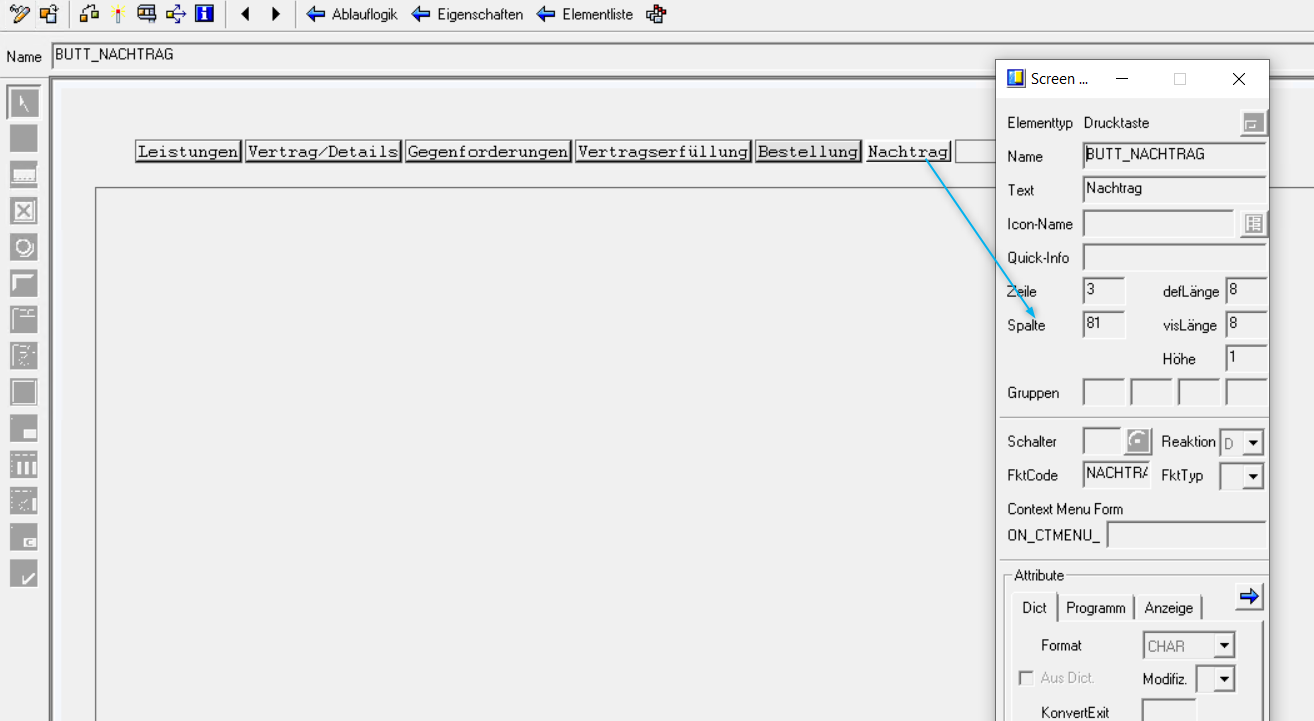


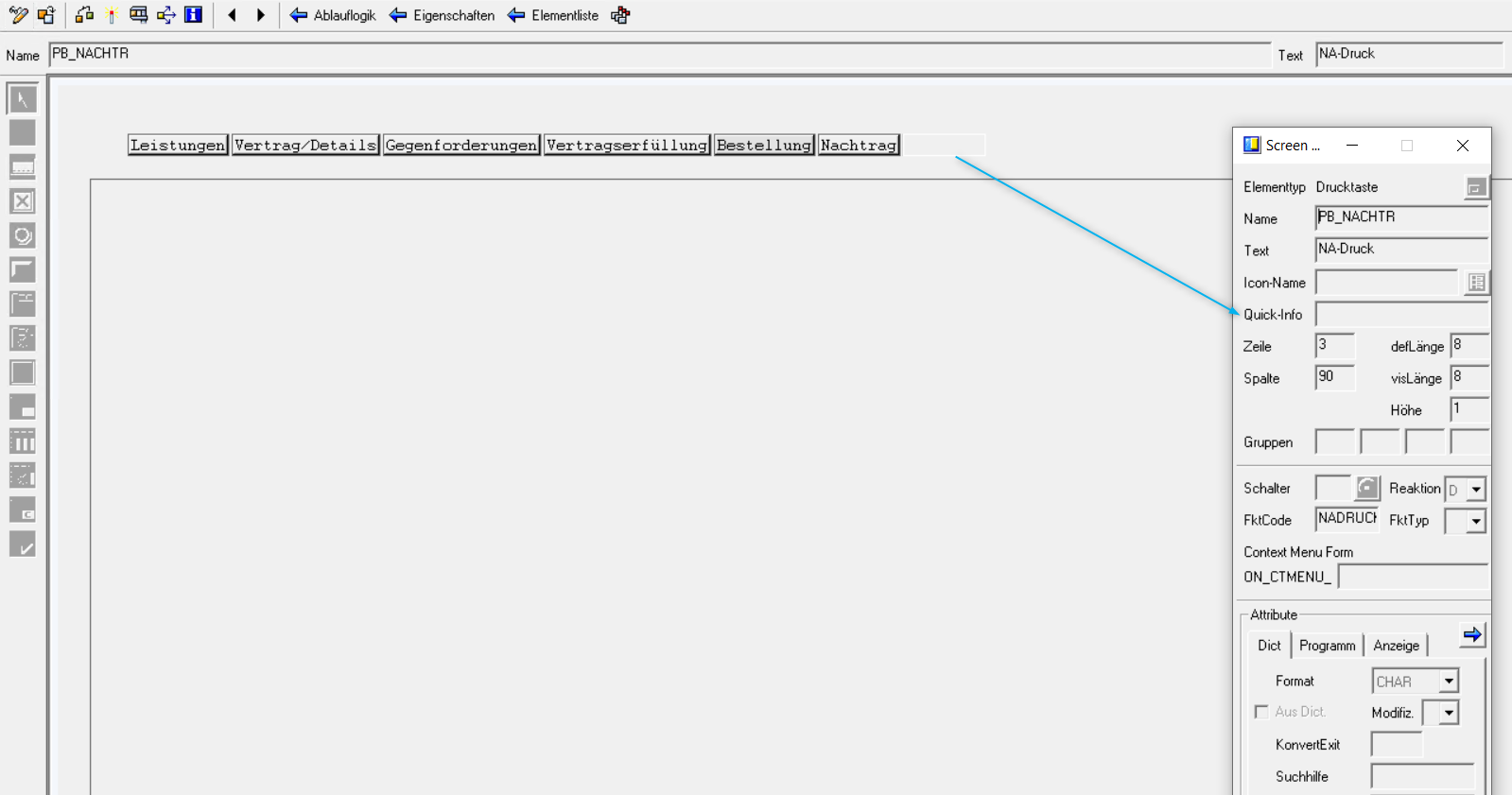


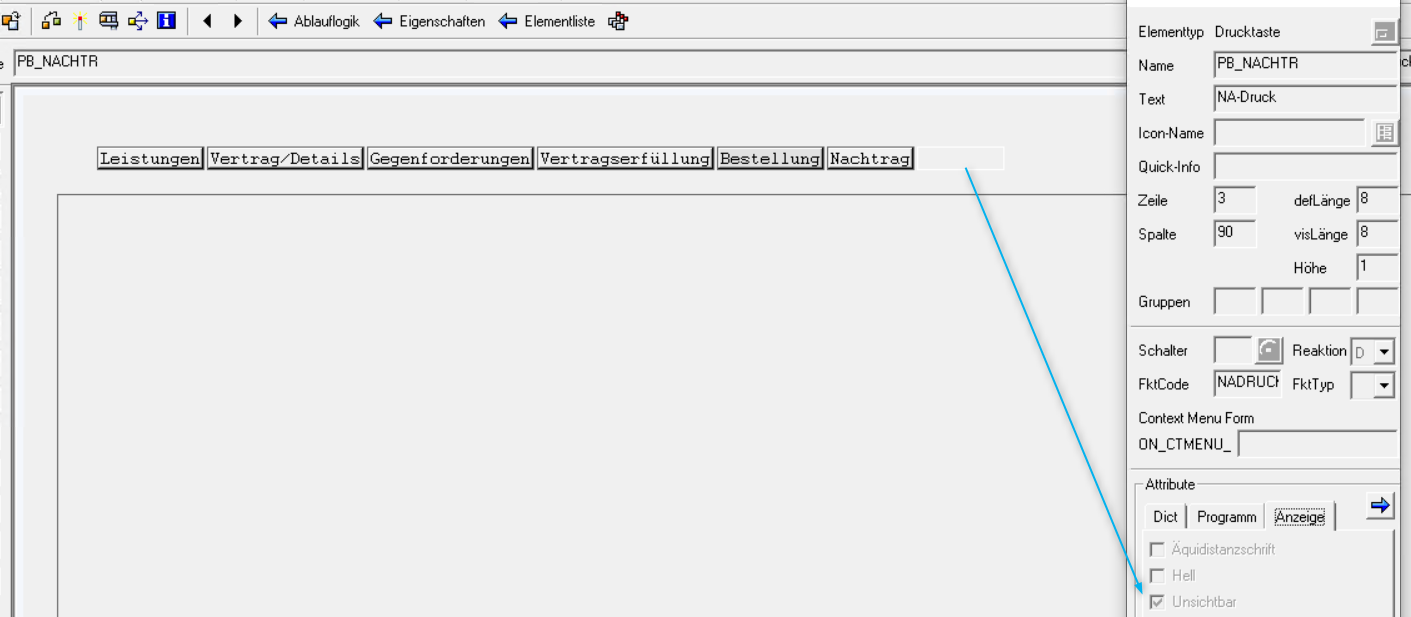


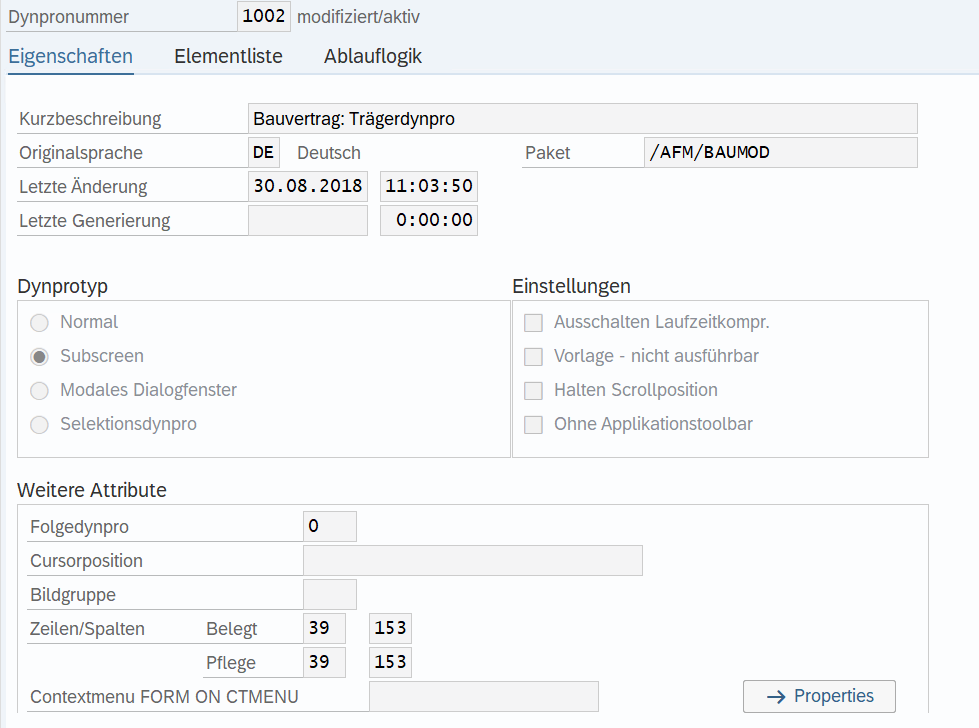












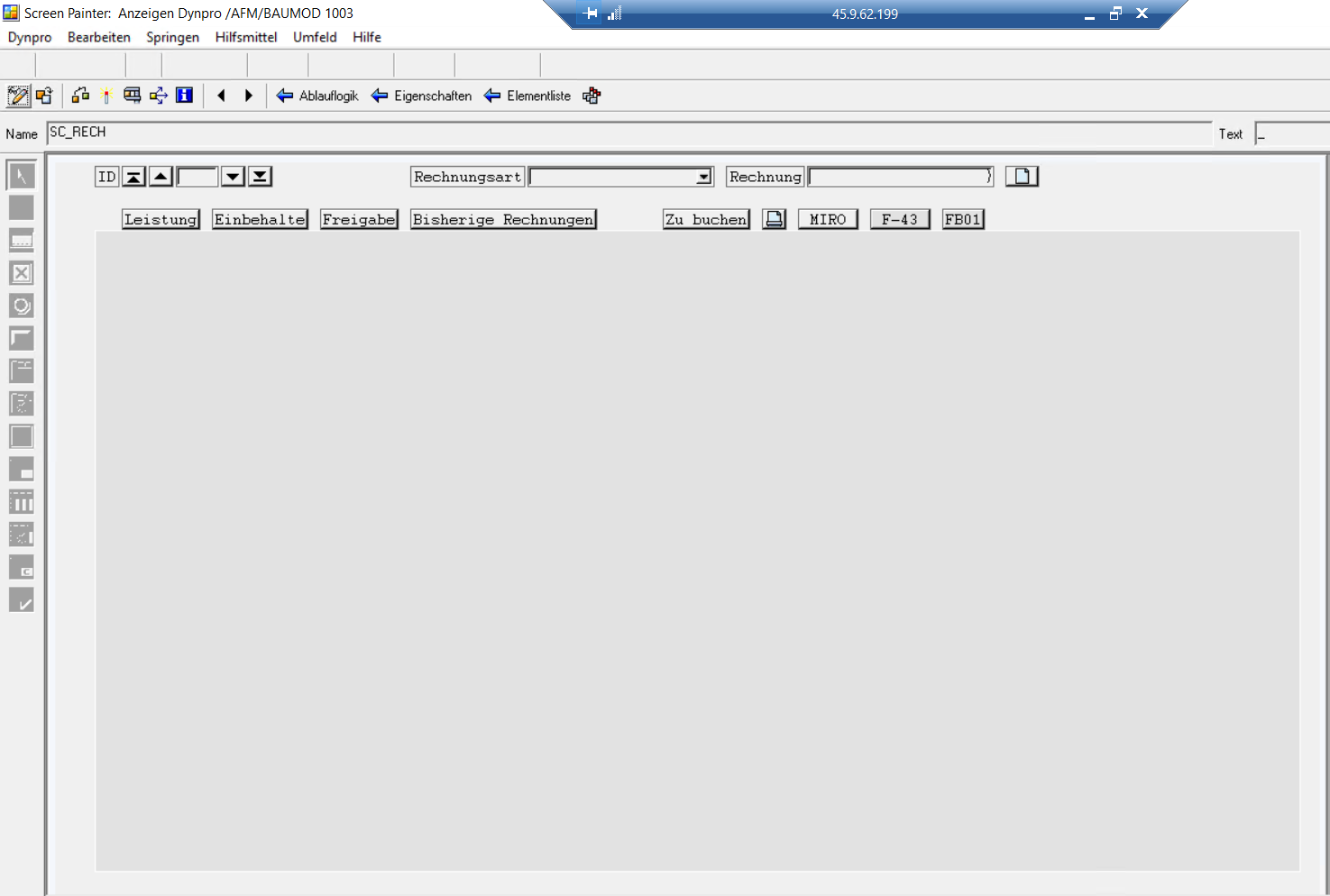


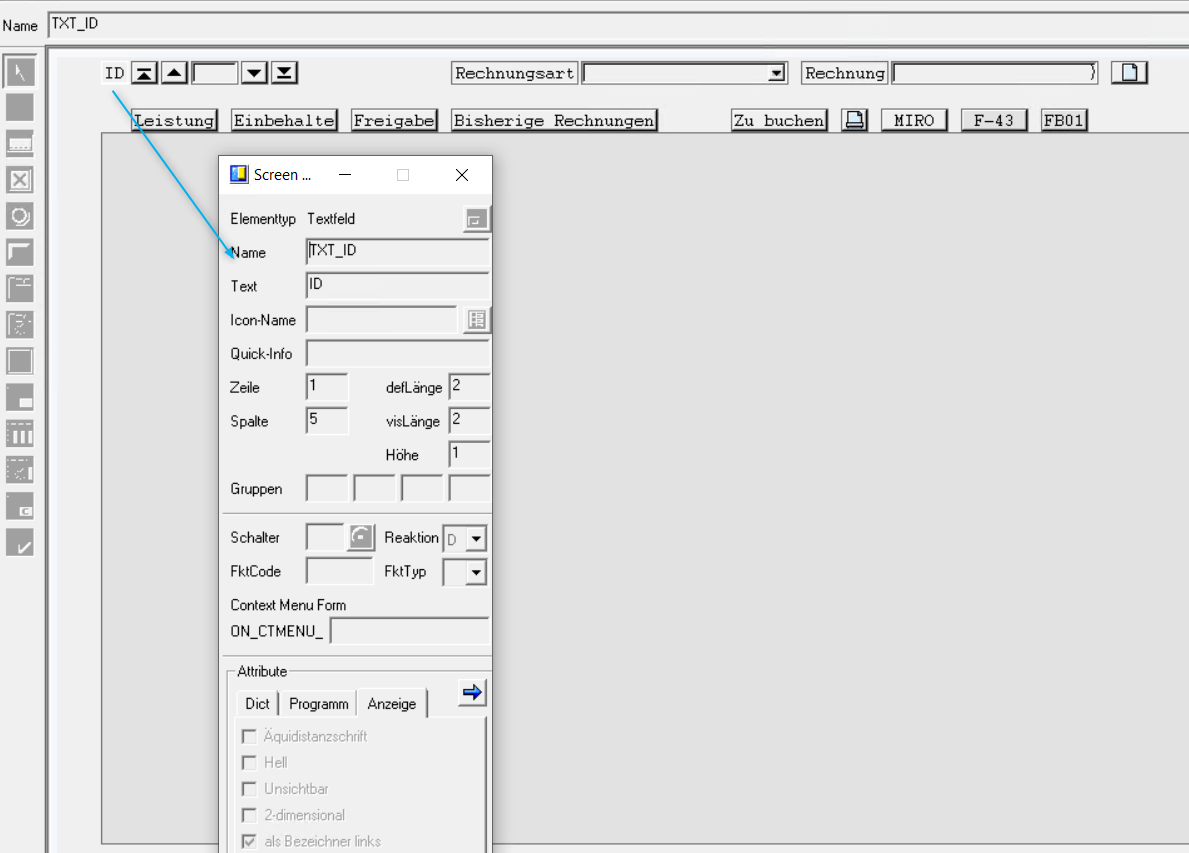
*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\**  
PROCESS BEFORE OUTPUT.  
*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\**  
  
*\* MODULE STATUS\_1002.*  
  
 MODULE modify\_screen\_1002.  
  
 CALL SUBSCREEN sc\_bau INCLUDING sy-repid subscr\_ba.  
  
  
*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\**  
PROCESS AFTER INPUT.  
*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\**  
  
 CALL SUBSCREEN sc\_bau.  
  
 MODULE user\_command\_1002.

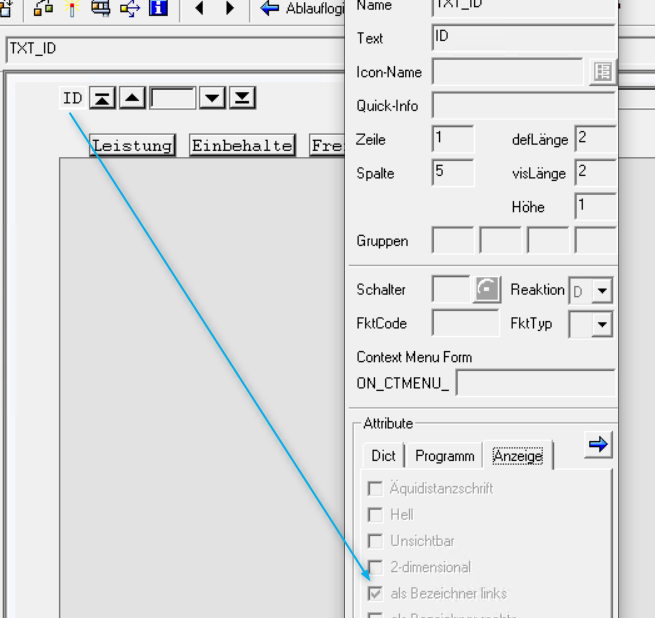
include 🡪 /AFM/BAUMOD\_PBO\_02

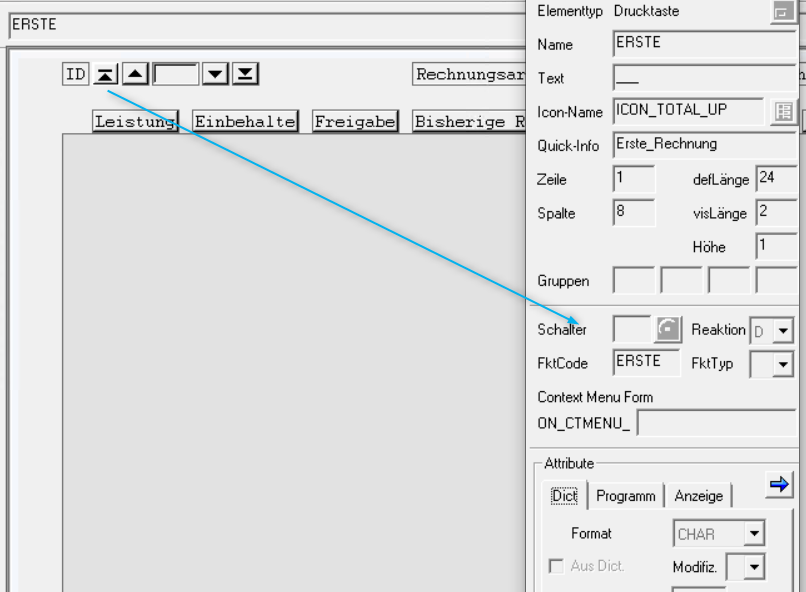
*\*&---------------------------------------------------------------------\**  
*\*& Module MODIFY\_SCREEN\_1002 OUTPUT*  
*\*&---------------------------------------------------------------------\**  
*\* Dunkelschalten der jeweils gedrückten Drucktasten*  
*\*----------------------------------------------------------------------\**  
MODULE modify\_screen\_1002 OUTPUT.  
  
 LOOP AT SCREEN.  
 IF subscr\_ba = '1021' AND screen-name EQ 'BUTT\_LEISTUNG'.  
 screen-input = '0'.  
 ELSEIF subscr\_ba = '1022' AND screen-name EQ 'BUTT\_DETAILS'.  
 screen-input = '0'.  
 ELSEIF subscr\_ba = '1023' AND screen-name EQ 'BUTT\_GEGENFORDERUNGEN'.  
 screen-input = '0'.  
 ELSEIF subscr\_ba = '1024' AND screen-name EQ 'BUTT\_VERTRAGSERFUELLUNG'.  
 screen-input = '0'.  
*\* ELSEIF subscr\_ba = '1025' AND screen-name EQ 'BUTT\_VERTRAG'.*  
*\* screen-input = '0'.*  
 ELSEIF subscr\_ba = '1027' AND screen-name EQ 'BUTT\_VERTRAG'.  
 screen-input = '0'.  
 ELSEIF subscr\_ba = '1026' AND screen-name EQ 'BUTT\_NACHTRAG'.  
 screen-input = '0'.  
 ENDIF.  
 MODIFY SCREEN.  
 ENDLOOP.  
  
 IF ws\_steuerung-butt\_vertrag IS INITIAL.  
 LOOP AT SCREEN.  
 IF screen-name EQ 'BUTT\_VERTRAG'.  
 screen-input = '0'.  
 MODIFY SCREEN.  
 EXIT.  
 ENDIF.  
 ENDLOOP.  
 ENDIF.  
  
 IF ws\_steuerung-butt\_nachtrag IS INITIAL.  
 LOOP AT SCREEN.  
 IF screen-name EQ 'BUTT\_NACHTRAG'.  
 screen-input = '0'.  
 MODIFY SCREEN.  
 EXIT.  
 ENDIF.  
 ENDLOOP.  
 ENDIF.  
  
ENDMODULE. *" MODIFY\_SCREEN\_1002 OUTPUT*

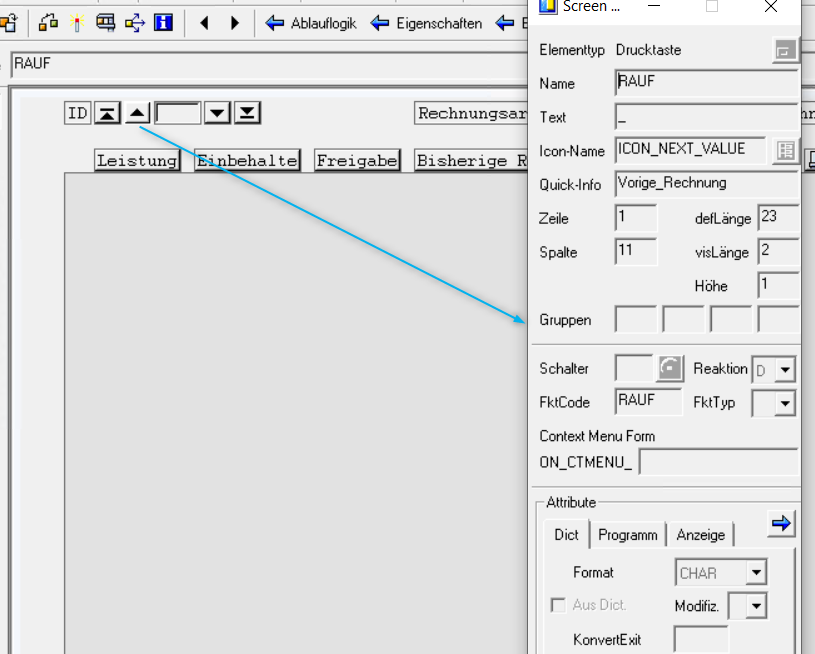
*Include 🡪 /AFM/BAUMOD\_PAI\_02  
\*----------------------------------------------------------------------\**  
*\* INCLUDE /AFM/BAUMOD\_PAI\_02 \**  
*\*----------------------------------------------------------------------\**  
*\*&---------------------------------------------------------------------\**  
*\*& Module USER\_COMMAND\_1002 INPUT*  
*\*&---------------------------------------------------------------------\**  
MODULE user\_command\_1002 INPUT.  
  
 MOVE sy-ucomm TO fcode.  
  
 CASE fcode.  
 WHEN 'LEIST'.  
 subscr\_ba = '1021'.  
 WHEN 'DETAILS'.  
 subscr\_ba = '1022'.  
 WHEN 'GEGENFORD'.  
 subscr\_ba = '1023'.  
 WHEN 'ERFUELL'.  
 subscr\_ba = '1024'.  
 WHEN 'VERTRAG'.  
 subscr\_ba = '1027'.  
 WHEN 'VERTR'.  
 subscr\_ba = '1027'.  
 WHEN 'NACHTRAG'.  
 subscr\_ba = '1026'.  
*\* when 'SAVE'.*  
*\* perform datensicherung.*  
*\* when 'DRUCK'. " OLE Wordanbindung hier ansetzen*  
*\* perform druck\_ole.*  
*\* when 'NADRUCK'. " OLE Wordanbindung hier ansetzen*  
*\* perform druck\_ole\_nachtrag.*  
  
*\* when 'BEAUFTRAGUNG'.*  
  
*\* DATA: gt\_rspar TYPE TABLE OF rsparams,*  
*\* gs\_rspar LIKE rsparams.*  
*\**  
*\* gs\_rspar-selname = 'SEL'.*  
*\* gs\_rspar-kind = 'S'.*  
*\* gs\_rspar-sign = 'I'.*  
*\* gs\_rspar-option = 'EQ'.*  
*\* gs\_rspar-low = bestellnr.*  
*\* APPEND gs\_rspar TO gt\_rspar.*  
*\**  
*\* SUBMIT /AFM/BEAUFTRAGUNG\_BDS USING SELECTION-SCREEN 1000 WITH SELECTION-TABLE gt\_rspar.*  
  
 WHEN OTHERS.  
  
 ENDCASE.  
  
ENDMODULE. *" USER\_COMMAND\_1002 INPUT*

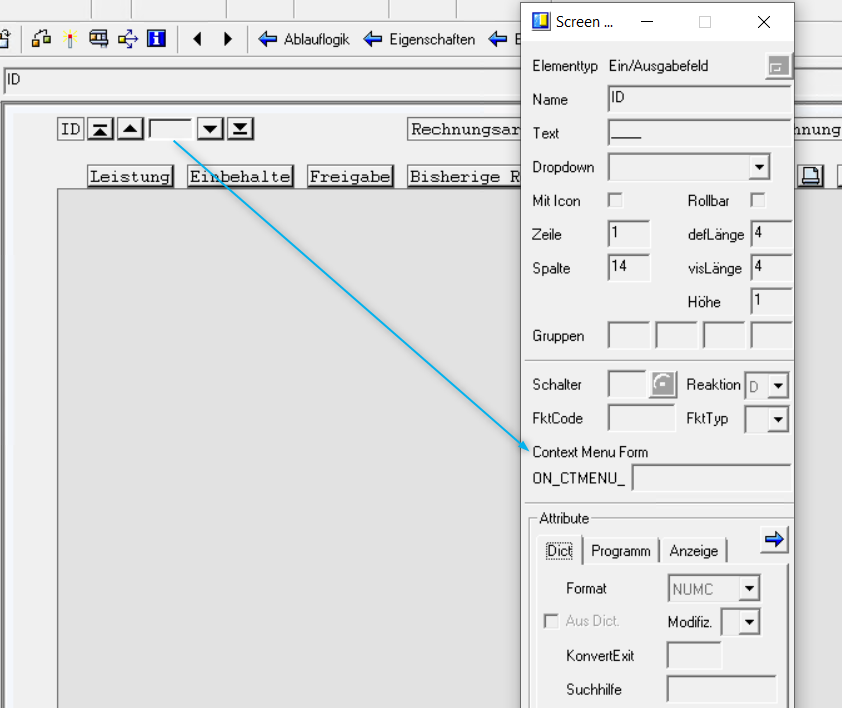
*Screen: 🡪 1003  
*

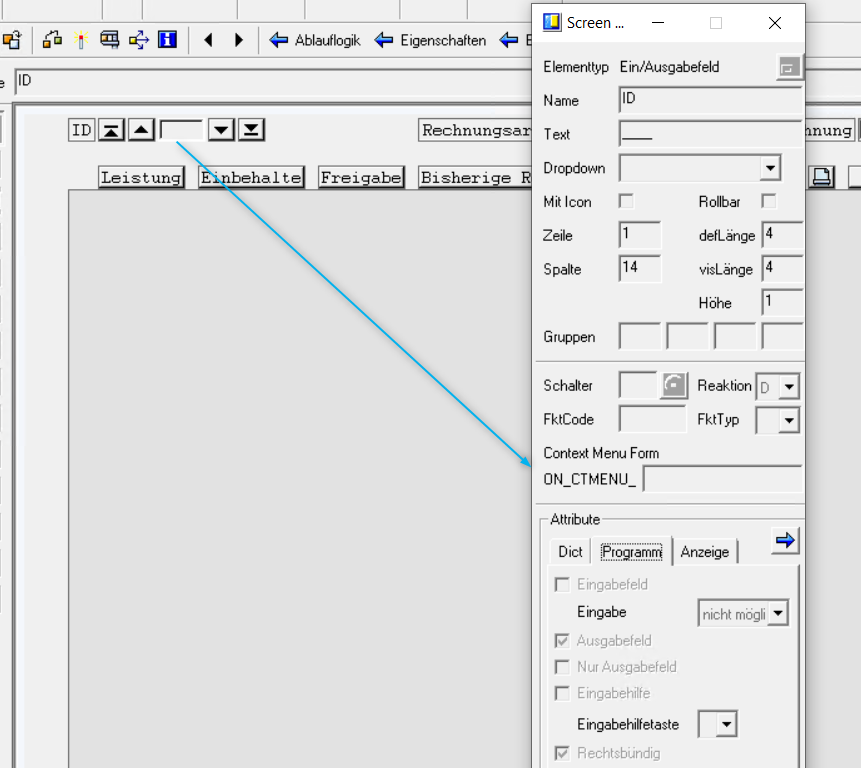
**

**

**

**

**

**