

Makro Installation

1. Aktuelles Makro [Makro Parameter](#)

ACHTUNG! Bitte beachten Sie, dass Sie das Makro als **Zip-Datei** (.ZIP) herunterladen und es **NICHT in Ihrem Browser** öffnen und kopieren sollten.

2. In der Vorerfassung für einen beliebigen Klienten im Eingangs- oder Ausgangsrechnungsbuch auf einen bestehenden Datensatz im Bearbeitungsmodus gehen und mit der rechten Maustaste auf das blaue Kästchen klicken und anschließend Feldeigenschaften auswählen.

Achtung: Berechtigung zum Erstellen/Ändern von Makros muss vorhanden sein.



2.1 Wenn die Feldeigenschaften nicht sichtbar sind, überprüfen Sie bitte Ihre BMD-Rechte oder führen Sie folgende Schritte aus:

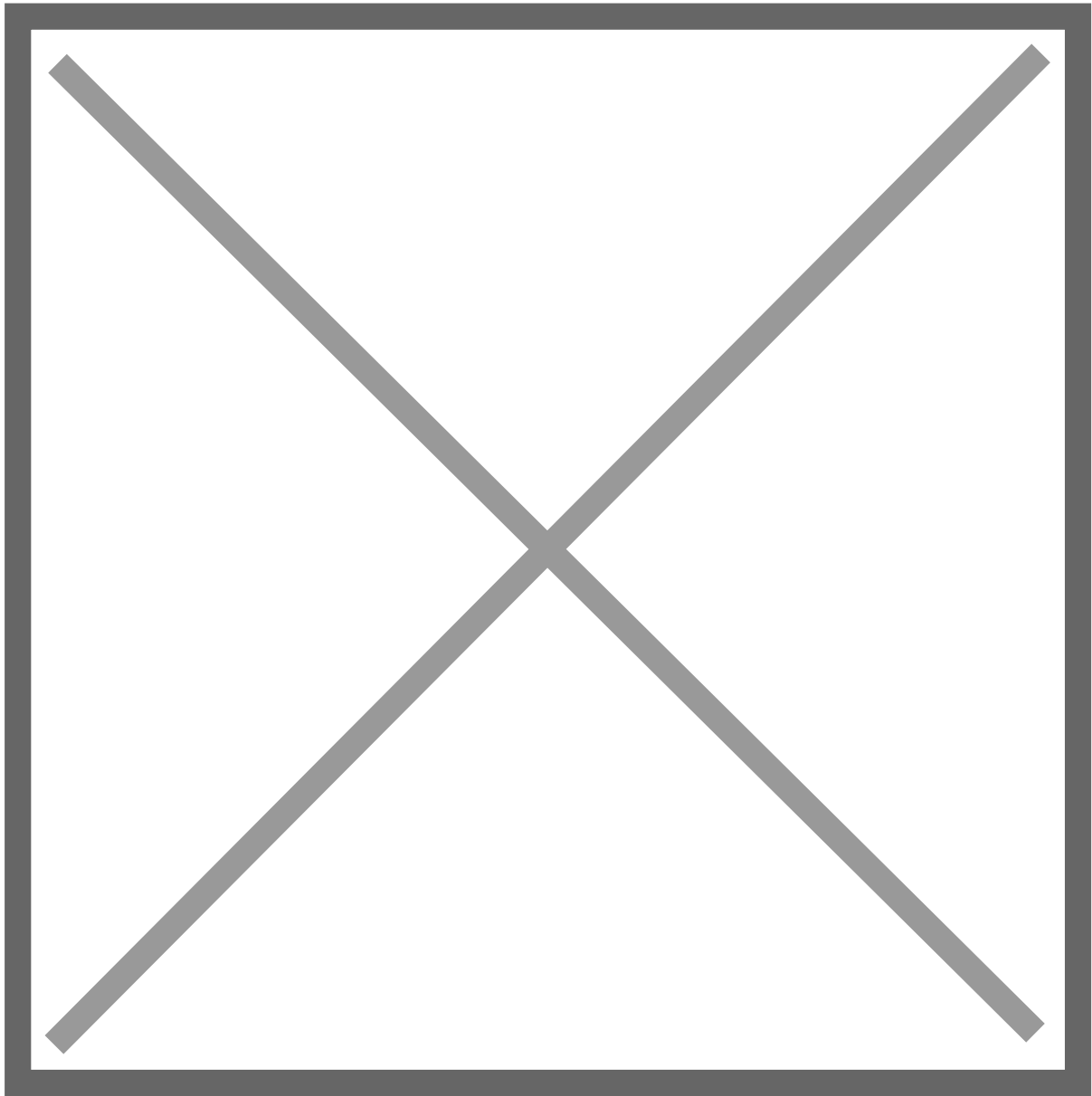
- In der Vorerfassung - rechte Seite der Maske, befinden sich 3 Punkte zur Feltauswahl.
- Wenn Sie diese auswählen, gelangen Sie zum Feltauswahlfenster.



- Wählen Sie die Bezeichnung "Lieferanten" aus, im unteren Teil befinden sich die Feldeigenschaften.

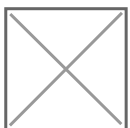


3. Feldeigenschaften Feldmakro anlegen: **Doppelklick links neben der Checkbox**
"Makro"



Checkbox Makro

4. Wenn Sie das Makro zum **ersten Mal** erstellen, gehen Sie bitte in den Makro-Editor, schalten Sie in den **Expertenmodus** und fügen Sie den Makrotext per Copy and Paste ein.
5. Optionen richtig einstellen ab ca. Zeile 20



6. Anschließend **kompilieren und speichern** Sie das Makro und es sollten keine Fehler auftreten.

Makro Parameter

- **do_show_kdlf** steuert die Anzeige des Pop-ups bei der Neuanlage oder Änderung eines Kunden bzw. Lieferanten:
- **1:** anzeigen,
- **0:** nicht anzeigen
- **do_auth** Aktivierung des Makros oder der kundenspezifischen Steuerung:
- **1:** Steuerung über das Frei Feld - FWF_FREIFELDA01_50,
- **0:** makro immer aktiv
- **do_modify_buchungstext** Buchungstext vom Kunden/Lieferanten übernehmen:
- **1:** buchungstext ändern auf Name des Lieferanten/Kunden
- **0:** keine buchungstext ändern auf Name des Lieferanten/Kunden

```
// ===== // Erstelldatum :
12.07.2019 01:50:36 // Ersteller : arewa // Eingerichtet von/am : AREWA // Beschreibung :
Feldmakro Lieferant // Version 1.0 vom 12.07.2019 // Version 2.0 vom 12.07.2020 // Version 3.0
vom 12.10.2022 mit konditionen // Version 3.1 vom 20.01.2023 ohne FibuMacroObject für bmd.com
+ matchcode setzen bei kdlf // Version 3.2 vom 13.03.2023 no BMD_CRLF // Version 3.3 vom
25.04.2023 do_show_kdlf geht wieder // Version 3.4 vom 10.05.2023 logging print dump_logfile //
Version 3.5 vom 30.08.2023 no exec_sql // Version 3.6 vom 02.11.2023 import kdlf file handle +
unique file + delete // Version 3.7 vom 23.11.2023 disable
VorerfassenModel.SetValue('MCA_VEB_UIDNR', per_uidnr); // Version 3.8 vom 08.12.2023
firmengruppen // Version 3.9 vom 01.02.2024 update person auch wenn iban, zahlssperre // Version
3.10 vom 21.11.2024 take lowest personenkreis // Version 3.11 vom 19.12.2022 kontoinhaber,
zahlssperre2 // ===== Var
FirmenNr :String; FibuNr :String; BS :String; kdlf_string :String; vorerf_typ: Integer;
VorerfassenModel :MacroObject; do_show_kdlf: Integer; do_modify_buchungstext: Integer; do_auth:
Integer; created_list: MacroObject; fh_bmd_log: TBMDInt;
//----- // Parameter festlegen - Kunde/Lieferant
nach Änderung anzeigen JA/NEIN //----- function
initParams; begin do_show_kdlf := 1; // 1: anzeigen, 0: nicht anzeigen do_auth := 0; // 1: Steuerung
über FWF_FREIFELDA01_50, 0: makro immer aktiv do_modify_buchungstext := 1; // 1:
buchungstext ändern auf Name des Lieferanten/Kunden end; //
=====
===== //
=====
===== function get_auth (FirmenNr: String; FibuNr: String); begin
Result := 15; if not do_auth then begin exit; end; //BMD_FirmenwerteFibu :=
FibuMacroObject.CreateFMacroFirmenwerteFibu('eindeutige FirmenwerteFibuObj-ID');
//BMD_FirmenwerteFibu.Load(FirmenNr, FibuNr); //Result := BMD_TEXTTONUMDEF
(BMD_FirmenwerteFibu.GetValue('MCA_FWF_FREIFELDA01_50')) sqlText := 'SELECT
FWF_FREIFELDA01_50 FROM FIBU.FWF_FIRMENWERTEFIBU WHERE FWF_FIRMENNR = :firmennr
AND FWF_FIBUNR = :fibunr'; iQuery := MacroObject.CreateMacroQuery('fetch_fwf_freifeld');
iQuery.setSQLText (sqlText); iQuery.setParamAsString('firmennr', FirmenNr);
iQuery.setParamAsString('fibunr', FibuNr); iQuery.Open(); if (iQuery.RecordCount() > 0) then begin
```

```

Result := BMD_TEXTTONUMDEF (iQuery.getAsString ('FWF_FREIFELDA01_50')); end; iQuery.close();
end; //

=====
===== procedure fetch_json (var json_string, json_learned, fibunr,
name, zusatzname, adresse, ort, plz, land_id, uid, uid_status, iban1, iban2, iban3, uid_datum,
kondition, zahlssperre, account_holder); begin json := MacroObject.CreateMacroJsonVal();
json.ParseJson(json_string); if json.KeyExists ('N1') then begin name := json.GetValueAsString('N1');
end; if json.KeyExists ('N2') then begin zusatzname := json.GetValueAsString('N2'); end; if
json.KeyExists ('A') then begin adresse := json.GetValueAsString('A'); end; if json.KeyExists ('H')
then begin account_holder := json.GetValueAsString('H'); end; if json.KeyExists ('C') then begin ort
:= json.GetValueAsString('C'); end; if json.KeyExists ('P') then begin plz :=
json.GetValueAsString('P'); end; if json.KeyExists ('CC') then begin land_id :=
json.GetValueAsString('CC'); end; if json.KeyExists ('U') then begin uid_status :=
json.GetValueAsString('U'); end; if json.KeyExists ('TS') then begin uid_timestamp :=
json.GetValueAsString('TS'); end; if json.KeyExists ('I1') then begin iban1 :=
json.GetValueAsString('I1'); end; if json.KeyExists ('I2') then begin iban2 :=
json.GetValueAsString('I2'); end; if json.KeyExists ('I3') then begin iban3 :=
json.GetValueAsString('I3'); end; if json.KeyExists ('Z') then begin zahlssperre :=
json.GetValueAsString('Z'); end; if json.KeyExists ('KD') then begin kondition :=
json.GetValueAsString('KD'); end; else begin kondition := '0'; end; if (uid_status = '1') then begin
uid_datum := uid_timestamp; uid_status := 'GÜLTIG, geprüft am ' + uid_timestamp; end; else if
(uid_status = '4') then begin uid_status := 'UNGÜLTIG'; end else if (BMD_LEN (uid_status) > 0) then
begin // nur wenn es einen status gibt uid_status := 'NICHT GEPRÜFT'; end;
json.ParseJson(json_learned); if json.KeyExists ('U') then begin uid := json.GetValueAsString('U');
end; end; //

=====
===== function show_kdlf (FirmenNr: String; FibuNr: String;
personenid: String); begin if do_show_kdlf = 1 then begin IParams := 'STP_CUS_COMPANYNO=' +
FirmenNr + ',STP_CUS_CUSTOMERID=' + personenid; if vorerf_typ = 3 then begin // ER IParams :=
IParams + ',KLI_ISTLIEFERANT=1'; end;
BMD_STARTFUNCTION('MCS_MDKUNDE_EDIT_CREATE',IParams); end; end; //

=====
===== function get_next_free_kdlf_nummer (FirmenNr: String;
buchsymbol: String; FibuNr: String); begin iQuery :=
MacroObject.CreateMacroQuery('fetch_nummernkreis'); sqlText := 'SELECT
PNK_FIRMENKATEGORIELFDNR, PNK_VONNR, PNK_BISNR FROM
BUERO.PNK_PERSONNUMMERNKREIS INNER JOIN FIBU.VBV_VORERFASBUCH on VBVB_FIRMENNR =
PNK_FIRMENNR and VBVB_FIBUNR=:fibunr and VBVB_BUS_BUCHSYMSYMBOL=:buchsymbol WHERE
PNK_FIRMENNR = :firmennr AND PNK_PERSONKZNR = iif (VBVB_VORERFASSTYP = 4, 1, iif
(VBVB_VORERFASSTYP = 3, 2, 99999)) ORDER BY PNK_FIRMENKATEGORIELFDNR'; // WHERE
PNK_FIRMENKATEGORIELFDNR=1 iQuery := MacroObject.CreateMacroQuery('fetch_kreise');
iQuery.setSQLText (sqlText); iQuery.setParamAsString('firmennr', Firmennr);
iQuery.setParamAsString('fibunr', FibuNr); iQuery.setParamAsString('buchsymbol', buchsymbol);
iQuery.Open(); if (iQuery.RecordCount() > 0) then begin from_konto :=
iQuery.getAsString('PNK_VONNR'); to_konto := iQuery.getAsString('PNK_BISNR'); iQuery.close();
sqlText := 'SELECT min(P.PER_PERSONENNR)+1 as MIN_NR FROM BUERO.PER_PERSON P WHERE

```

```

P.PER_PERSONENNR between :from and :to and not exists (select 1 from BUERO.PER_PERSON C,
BUERO.KLI_KUNDE_LIEFERANT A WHERE C.PER_PERSONENNR = P.PER_PERSONENNR+1 AND
C.PER_FIRMENNR in ( SELECT FIG_FIRMENNR FROM BUERO.FIG_FIRMENGRUPPENZU A WHERE
A.FIG_GRUPPE_FIRMENNR = (select B.FIG_GRUPPE_FIRMENNR FROM
BUERO.FIG_FIRMENGRUPPENZU B WHERE B.FIG_FIRMENNR = :firmennr) union select :firmennr )
AND C.PER_PERSONENART = "KL" AND C.PER_JAHR >= 0 AND C.PER_ISTAKTUELLUNTERNR >= 1
AND A.KLI_FIRMENNR = C.PER_FIRMENNR AND A.KLI_KLID = C.PER_PERSONENID AND
A.KLI_KLUNTERNR = C.PER_PERSONUNTERNR); iQuery :=
MacroObject.CreateMacroQuery('next_nummer'); iQuery.setSQLText (sqlText);
iQuery.setParamAsString('firmennr', Firmennr); iQuery.setParamAsString('from', from_konto);
iQuery.setParamAsString('to', to_konto); iQuery.Open(); if (iQuery.RecordCount() > 0) then begin
next_kdlf := iQuery.getAsString('MIN_NR'); iQuery.close(); Result := next_kdlf; end; end; end; //
=====
===== procedure get_KDLF (var personenid, kontonr, uid, uid_datum,
name, zusatzname, strasse, plz, land_id, iban1,iban2,iban3, kondition, zahlsperr, account_holder);
begin Result := ''; personenid := ''; sqlText := 'SELECT PER_NAME as NAME, isnull
(PER_ZUSATZNAME, '') as ZUSATZNAME, isnull (PER_UIDNR, '') as VATREGNO, PER_PERSONENID
as PERSONENID, isnull (BAI_IBAN, '') as IBAN, isnull (BAI_KONTOINHABER, '') as account_holder,
isnull (ADR_STRASSE, '') as ADRESSE, ADR_PLZ as PLZ, isnull (KOD_KONDLFDNR, 0) as KOND,
isnull (LAN_ISO_CODE, "AT") as COUNTRY, isnull (iif (KTO_KONTENGRUPPE = 2, iif (isnull
(KLI_HATZAHLSPERRE, 0) = 1, 1, PEK_KUN_ZAHLSPERRE), iif (isnull (KLI_HATZAHLSPERRE, 0) = 1,
1, PEK_LIE_ZAHLSPERRE)), 0) as ZAHLSPERRE FROM FIBU.PEK_PERSONENKONTO P1 INNER JOIN
BUERO.KLI_KUNDE_LIEFERANT KL ON KL.KLI_FIRMENNR = P1.PEK_FIRMENNR AND KL.KLI_KLID =
P1.PEK_KLID INNER JOIN BUERO.PER_PERSON ON PER_FIRMENNR = PEK_FIRMENNR AND
PER_PERSONENID = P1.PEK_KLID INNER JOIN FIBU.KTO_KONTO ON KTO_FIRMENNR =
PEK_FIRMENNR AND KTO_FIBUNR = PEK_FIBUNR AND KTO_KONTONR = PEK_KONTONR INNER JOIN
BUERO.ADR_ADRESSE ON ADR_FIRMENNR = PEK_FIRMENNR AND ADR_ADRESSLFDNR =
PER_ADRESSLFDNR LEFT JOIN BUERO.BAI_BANKINSTITUTIONSZU ON
BAI_BANKINSTLFDNR=PER_SENDER_BANKINSTLFDNR LEFT JOIN BUERO.KOD_KONDITIONEN ON
KOD_KONDNR=KLI_LIEF_KONDNR LEFT JOIN BUERO.LAN_LAND ON LAN_LANDNR = PER_LANDNR
WHERE PEK_FIRMENNR = :firmennr AND PEK_FIBUNR = :fibunr AND PEK_KONTONR = :kontonr';
//BMD_SHOWMESSAGE ('get_KDLF firmennr: ' + Firmennr + ' fibunr: ' + FibuNr + ' kontonr: ' +
kontonr); iQuery := MacroObject.CreateMacroQuery('get_person'); iQuery.setSQLText (sqlText);
iQuery.setParamAsString('firmennr', Firmennr); iQuery.setParamAsString('kontonr', KontoNr);
iQuery.setParamAsString('fibunr', FibuNr); iQuery.Open(); if (iQuery.RecordCount() > 0) then begin
uid := iQuery.getAsString('VATREGNO'); name := iQuery.getAsString('NAME'); zusatzname :=
iQuery.getAsString('ZUSATZNAME'); personenid := iQuery.getAsString('PERSONENID'); strasse :=
iQuery.getAsString('ADRESSE'); iban1 := iQuery.getAsString('IBAN'); kondition :=
iQuery.getAsString('KOND'); zahlsperr := iQuery.getAsString('ZAHLSPERRE'); land_id :=
iQuery.getAsString('COUNTRY'); plz := iQuery.getAsString('PLZ'); account_holder :=
iQuery.getAsString('account_holder'); end; iQuery.close(); end; //
=====
===== //-----
function dump_logfile (logfile: String); begin Result := ''; if (BMD_FILEEXISTS (logfile)) then begin
BMD_ASSIGNFILE (fh_bmd_log, logfile); BMD_OPENFILE_RESET (fh_bmd_log); while not BMD_EOF
(fh_bmd_log) do begin line := BMD_READLN (fh_bmd_log); //BMD_SHOWMESSAGE (line); Result :=

```

```

Result + line; end; BMD_CLOSEFILE (fh_bmd_log); end; end; //
=====
===== function import_KDLF (kontonr: String; uid: String; uid_datum:
String; name: String; zusatzname: String; strasse: String; plz: String; land_id: String; iban1: String;
iban2: String; iban3: String; kondition: String; zahlsperrre: String; kontoinhaber: String); begin
//gFileName := 'kdlf_import_' + kotonr + '.csv'; sequence := BMD_GETSEQID(); gFileName :=
'kdlf_import_' + sequence + '.csv'; sql_exists := 'SELECT count(1) as COUNT FROM
FIBU.PEK_PERSONENKONTO P1 INNER JOIN BUERO.KLI_KUNDE_LIEFERANT KL ON KL.KLI_FIRMENNR
= P1.PEK_FIRMENNR AND KL.KLI_KLID = P1.PEK_KLID INNER JOIN BUERO.PER_PERSON ON
PER_FIRMENNR = PEK_FIRMENNR AND PER_PERSONENID = P1.PEK_KLID INNER JOIN
FIBU.KTO_KONTO ON KTO_FIRMENNR = PEK_FIRMENNR AND KTO_FIBUNR = PEK_FIBUNR AND
KTO_KONTONR = PEK_KONTONR WHERE PEK_FIRMENNR = :firmennr AND PEK_FIBUNR = :fibunr
AND PEK_KONTONR = :kontonr'; temp:= BMD_GETENVIRONMENTVAR('temp'); filename := temp
+'\'+ gFileName; //if (BMD_FILEEXISTS(filename)) then begin // BMD_DELETEFILE(filename);
//DeleteFile //end; //BMD_CREATEFILE(filename); IFileNr := 1;
BMD_ASSIGNFILE(IFileNr,filename,1,1,1,1200); // unicode //rc := BMD_OPENFILE_APPEND(IFileNr);
//if (rc = '-1') then begin // BMD_SHOWMESSAGE('Fehler open file ' + filename); // exit; //end;
header_line1 :=
'NTCS_CONSTID_INFO%10%MCA_KTO_KONTONR%MCA_PER_NAME%MCA_PER_ZUSATZNAME%MCA
_ADR_ORTSNAME%MCA_ADR_PLZ%MCA_ADR_STRASSE%MCA_PER_UIDNR%MCA_PER_LANDNR%MCA
_A_BAI_IBAN%MCA_PER_UIDDATUM%MCA_KLI_KONDNR%MCA_KLI_HATZAHLSPERRE%MCA_PEK_LIE_
ZAHLSPERRE%MCA_PER_MATCHCODE%MCA_BAI_KONTOINHABER'; header_line2 := 'kto-
nr;Name;zusatzname;Ort;PLZ;Strasse;UID;Land;iban-nr;UID-
Datum;kondition;zahlsperre1;zahlsperre2;matchcode;kontoinhaber'; //header_line1 :=
'NTCS_CONSTID_INFO%10%MCA_KTO_KONTONR%MCA_PER_NAME%MCA_PER_ZUSATZNAME%MCA
_ADR_ORTSNAME%MCA_ADR_PLZ%MCA_ADR_STRASSE%MCA_PER_UIDNR%MCA_PER_LANDNR%MCA
_A_BAI_IBAN%MCA_PER_UIDDATUM%MCA_KLI_KONDNR%MCA_KLI_HATZAHLSPERRE%MCA_PER_MAT
CHCODE'; //header_line2 := 'kto-nr;Name;zusatzname;Ort;PLZ;Strasse;UID;Land;iban-nr;UID-
Datum;kondition;zahlsperre;matchcode'; BMD_WRITELN(IFileNr, header_line1);
BMD_WRITELN(IFileNr, header_line2); line:= kotonr + ';' + name + ';' + zusatzname + ';;' + +plz
+ ';' + strasse + ';' + uid + ';' + land_id + ';' + iban1 + ';' + uid_datum + ';' + kondition + ';' +
zahlsperre + ';' + zahlsperre + ';' + BMD_UPPER (name) + ';' + kontoinhaber;
BMD_WRITELN(IFileNr, line); BMD_CLOSEFILE(IFileNr); if(BMD_FILEEXISTS(filename))then begin
if((FirmenNr<>'') AND (FibuNr<>'') AND (filename<>''))then begin //BMD_SHOWMESSAGE
(dump_logfile (filename)); logfile := temp +'\Kontenimport' + FirmenNr + '_' + kotonr + '.log';
IParam := 'STP_FIB_COMPANYNO=' + FirmenNr + ',STP_FIB_FIBUNO=' + FibuNr +
',STP_IMP_FILENAME=' + filename + ',STP_LOG_FILENAME=' + logfile + ','; IParam := IParam +
'SILENT=1,'; IParam := IParam + 'STP_IMP_SEPARATOR=,,'; IParam := IParam +
'STP_IMP_ENCODING=3,'; IParam := IParam + 'STP_USEFIXEDLENGTH=0,'; IParam := IParam +
'STP_IGNORE_FIRSTLINE=1';
BMD_STARTFUNCTION('MCS_FRMFIBUVARCSVPERSONENKONTENIMPEXP_CREATE',IParam); //Start
Var. Personenimport BMD_DELETEFILE(filename); uid := ''; uid_status := ''; uid_timestamp := '';
name := ''; zusatzname := ''; adresse := ''; plz := ''; land_id := ''; iban1 := ''; iban2 := ''; iban3 :=
''; kondition := ''; personenid := ''; zahlsperrre := ''; get_KDLF (personenid, kotonr, uid,
uid_timestamp, name, zusatzname, adresse, plz, land_id, iban1, iban2, iban3, kondition,
zahlsperre, kontoinhaber); if personenid = '' then begin BMD_SHOWMESSAGE('Fehler:

```

```

Anlegen/Ändern Konto ' + kontonr + ' failed logfile: ' + dump_logfile (logfile)); end; Result :=
personenid; end; end; VorerfassenModel.SetValue('MCA_VEB_KONTONR', kontonr); end; //
=====
===== function store_created_kdlf (uid: String; name: String;
vorerf_typ: String; personenid: String); begin if uid = '' then begin // search for name search_string
:= FirmenNr + ';' + FibuNr + ';' + vorerf_typ + ';' + name; end; else begin search_string :=
FirmenNr + ';' + FibuNr + ';' + vorerf_typ + ';' + 'UID' + uid; end; created_list.SetKeyValue
(search_string, personenid); end; //
=====
===== // bei überspringen leeres feld --> gibt es den lieferanten
schon? //
=====
===== function KDLF_exists (search_uid: String; search_name: String;
vorerf_typ: String); begin sqlText := 'SELECT PEK_KONTONR as KONTONR, PER_NAME as NAME,
PER_ZUSATZNAME as ZUSATZNAME, PER_UIDNR as VATREGNO, KTO_KONTENGRUPPE FROM
FIBU.PEK_PERSONENKONTO P1 INNER JOIN BUERO.KLI_KUNDE_LIEFERANT KL ON KL.KLI_FIRMENNR
= P1.PEK_FIRMENNR AND KL.KLI_KLID = P1.PEK_KLID INNER JOIN BUERO.PER_PERSON ON
PER_FIRMENNR = PEK_FIRMENNR AND PER_PERSONENID = P1.PEK_KLID INNER JOIN
FIBU.KTO_KONTO ON KTO_FIRMENNR = PEK_FIRMENNR AND KTO_FIBUNR = PEK_FIBUNR AND
KTO_KONTONR = PEK_KONTONR WHERE PEK_FIRMENNR = :firmennr AND PEK_FIBUNR = :fibunr
AND (LEN (:uid) > 5 AND PER_UIDNR = :uid or (isnull (:uid, '') = '' and PER_NAME = :name)) AND
(KTO_KONTENGRUPPE = 4 OR (KTO_KONTENGRUPPE = 3 and :vorerf_typ = 3) OR
(KTO_KONTENGRUPPE = 2 and :vorerf_typ = 4))'; // Kontengruppe KUNDE=2, LIEFERANT=3, both
= 4 // vorerf_typ 4 = AR (Kunden), 3: ER (Lieferanten) Result := '0'; iQuery :=
MacroObject.CreateMacroQuery('get_konto'); iQuery.setSQLText (sqlText);
iQuery.setParamAsString('firmennr', Firmennr); iQuery.setParamAsString('fibunr', FibuNr);
iQuery.setParamAsString('uid', search_uid); iQuery.setParamAsString('name', search_name);
iQuery.setParamAsString('vorerf_typ', vorerf_typ); iQuery.Open(); if (iQuery.RecordCount() > 0)
then begin per_kontonr := iQuery.getAsString('KONTONR'); per_name :=
iQuery.getAsString('NAME'); per_uidnr := iQuery.getAsString('VATREGNO'); per_zusatzname :=
iQuery.getAsString('ZUSATZNAME'); kontengruppe := iQuery.getAsString('KTO_KONTENGRUPPE');
rc := BMD_ASKVALUEBOOL (kdlf_string + 'nummer ' + per_kontonr + bmd_lineBreak() +
bmd_lineBreak() + 'Name: <' + per_name + '>' + 'UID : <' + per_uidnr + '>', kdlf_string + '
übernehmen ?'); if (rc = 1) then begin VorerfassenModel.SetValue('MCA_VEB_KONTONR',
per_kontonr); Result := per_kontonr; end else begin Result := '0'; end; end; else begin
//BMD_SHOWMESSAGE ('no kdlf firmennr: ' + Firmennr + ' fibunr: ' + FibuNr + ' vorerf_typ: ' +
vorerf_typ + ' uid: ' + search_uid + ' name: ' + search_name + ' sql: ' + sqlText); end; end; //
=====
===== //BMD_SHOWMESSAGE ('start'); IF
(BMD_GETAPPLICATIONTYPE() <> 'BMDNTCS') THEN BEGIN exit; END; VorerfassenModel :=
MacroObject.CreateMacroModel('TBMDMDFibuVorerfassBuchungMgr'); //CreateMacroModel
lkontobezeichnung := VorerfassenModel.GetValue('MCU_VEB_ADD_KONTOBEZEICHNUNG');
//GetValue FibuNr := VorerfassenModel.GetValue('MCA_VEB_FIBUNR'); FirmenNr :=
VorerfassenModel.GetValue('MCA_VEB_FIRMENNR'); uid :=
VorerfassenModel.GetValue('MCA_VEB_UIDNR'); text :=
VorerfassenModel.GetValue('MCA_VEB_BUCHUNGSTEXT'); id :=

```

```

VorerfassenModel.GetValue('MCA_VEB_VORERFASSID'); rg_file :=
VorerfassenModel.GetValue('MCA_VEB_FREIFELDA01_20'); json :=
VorerfassenModel.GetValue('MCA_VEB_FREIFELDA03_300'); json_learned :=
VorerfassenModel.GetValue('MCA_VEB_FREIFELDA01_50'); periode :=
VorerfassenModel.GetValue('MCA_VEB_PERIODE'); blind_flag :=
VorerfassenModel.GetValue('MCA_VEB_FREIFELDN6_10'); fh_bmd_log := 4; if (BMD_LEN (text) >
100) then begin if (BMD_POS (text, ' ') > 0) then begin text := BMD_COPY (text, 1, 101); end else
begin text := BMD_COPY (text, 1, 100); end; end; personenid :=
VorerfassenModel.GetValue('MCA_VEB_PER_PERSONENID'); // alte personenid BS :=
VorerfassenModel.GetValue('MCA_VEB_BUS_BUCHSYMSYMBOL'); field_input_value :=
FIELD_INPUT_VALUE; field_old_value := FIELD_OLD_VALUE; KontoNr := field_input_value;
KontoNr_numeric := BMD_TEXTTONUMDEF (field_input_value); field_output_value :=
field_input_value; per_uidnr := ""; per_name := ""; initParams; AUTH_CRE_KUNDEN := 1;
AUTH_UPD_KUNDEN := 2; AUTH_CRE_LIEF := 4; AUTH_UPD_LIEF := 8; BMD_VORERF_TYP_KUNDEN
:= '4'; BMD_VORERF_TYP_LIEF := '3'; // Änderung Blindsätze unterbinden if (blind_flag = '99') and
(BMD_LEN (field_input_value) > 0) and (KontoNr_numeric > 0) then begin BMD_SHOWMESSAGE
('Änderung bereits verbuchten Blindsatz nicht erlaubt - bitte löschen vor Import');
FIELD_OUTPUT_VALUE := ""; exit; end; vorerf_typ :=
VorerfassenModel.GetValue('MCA_VEB_VBV_VORERFASSTYP'); kdLf_string := 'Lieferanten'; if
((vorerf_typ <> BMD_VORERF_TYP_LIEF) and (vorerf_typ <> BMD_VORERF_TYP_KUNDEN)) then
begin // Kassa exit; end; else if (vorerf_typ = BMD_VORERF_TYP_KUNDEN) then begin kdLf_string :=
'Kunden'; end; auth := get_auth (FirmenNr, FibuNr); auth_create := 0; auth_modify := 0; if
(vorerf_typ = BMD_VORERF_TYP_KUNDEN) then begin if auth and AUTH_CRE_KUNDEN then begin
auth_create := 1; end; if auth and AUTH_UPD_KUNDEN then begin auth_modify := 1; end; end; if
(vorerf_typ = BMD_VORERF_TYP_LIEF) then begin if auth and AUTH_CRE_LIEF then begin
auth_create := 1; end; if auth and AUTH_UPD_LIEF then begin auth_modify := 1; end; end; if (not
auth_create) and (not auth_modify) then begin //BMD_SHOWMESSAGE ('no auth'); exit; end;
adresse := ""; ort := ""; plz := ""; land_id := ""; uid_status := ""; iban1 := ""; iban2 := ""; iban3 := "";
per_adresse := ""; per_zusatzname := ""; per_iban1 := ""; per_iban2 := ""; per_iban3 := ""; uid_datum
:= ""; name := ""; zusatzname := ""; kondition := ""; zahlsperrre := ""; per_zahlsperrre := "";
kontoinhaber := ""; per_kontoinhaber := ""; per_plz := ""; per_land_id := ""; per_kondition := "";
personenid := ""; uid_timestamp := ""; done_list :=
MacroObject.FindGlobalMacroStringList('Logisthai_done'); if not done_list then begin done_list :=
MacroObject.CreateGlobalMacroStringList('Logisthai_done'); end; created_list :=
MacroObject.FindGlobalMacroStringList('Logisthai_created'); if not created_list then begin
created_list := MacroObject.CreateGlobalMacroStringList('Logisthai_created'); end; fetch_json (json,
json_learned, fibunr, name, zusatzname, adresse, ort, plz, land_id, uid, uid_status, iban1, iban2,
iban3, uid_datum, kondition, zahlsperrre, kontoinhaber); if (BMD_LEN (field_input_value) > 0) and
(KontoNr_numeric > 0) then begin if not auth_modify then begin //BMD_SHOWMESSAGE ('no auth');
exit; end; //BMD_SHOWMESSAGE ('field changed'); // field update && check update kundenstamm
get_KDLF (personenid, kontonr, per_uidnr, uid_timestamp, per_name, per_zusatzname,
per_adresse, per_plz, per_land_id, per_iban1, per_iban2, per_iban3, per_kondition, per_zahlsperrre,
per_kontoinhaber); if (field_input_value <> field_old_value) then begin if do_modify_buchungstext
then begin VorerfassenModel.SetValue('MCA_VEB_BUCHUNGSTEXT', per_name); end;
//VorerfassenModel.SetValue('MCA_VEB_UIDNR', per_uidnr);
VorerfassenModel.SetValue('MCA_VEB_FREIFELDNK2', 1); // set change id end if zahlsperrre = " then

```

```

begin zahlssperre := per_zahlssperre; end; if kondition = '' then begin kondition := per_kondition;
end; // update_stamm if (((BMD_LEN (uid) > 0) or (BMD_LEN (name) > 0)) and (BMD_LEN (name) >
0) and ( (per_uidnr <> uid) or (per_name <> name) or (adresse <> per_adresse) or (plz <>
per_plz) or (iban1 <> per_iban1) or (land_id <> per_land_id) or ((kondition <> per_kondition) AND
(kondition <> '')) or ((zahlssperre <> per_zahlssperre) AND (zahlssperre <> '')) or ((kontoinhaber <>
per_kontoinhaber) AND (per_kontoinhaber <> '')) )) then begin current_date := BMD_DATE(); //
BMD_SHOWMESSAGE('per_uidnr: ' + per_uidnr + ' uid: ' + uid + ' per_name: ' + per_name + '
name: ' + name + ' adresse: ' + adresse + ' per_adresse: ' + per_adresse + ' kondition: ' +
kondition + ' per_kondition: ' + per_kondition); // wurde mit aktuellem tagesdatum + firmennr +
fibunr + periode + kdlf + neuem updatewert schon einmal nach einem update gefragt
//done_string := current_date + ';' + FirmenNr + ';' + FibuNr + ';' + Periode + ';' +
field_input_value + ';' + name; //done :=
BMD_GETINTPARAMVALUE('LOGISTHAI_KDLF',done_string,''); //done := ''; // test only
//BMD_SHOWMESSAGE('per_uidnr: ' + per_uidnr + ' uid: ' + uid + ' per_name: ' + per_name + '
name: ' + name + ' adresse: ' + adresse + ' per_adresse: ' + per_adresse + ' kondition: ' +
kondition + ' per_kondition: ' + per_kondition); done_string := FirmenNr + ';' + FibuNr + ';' +
Periode + ';' + field_input_value + ';' + name; done := done_list.GetKeyValue (done_string); if
(done <> '') then begin //BMD_SHOWMESSAGE('Schon mal gefragt und du wolltest nicht, Jetzt
Pech'); end; else begin question := ''; if (per_name <> name) then begin question := question +
'Name <' + per_name + '> auf <' + name + '>' + bmd_lineBreak(); end; if (per_uidnr <> uid) then
begin question := question + 'UID <' + per_uidnr + '> auf <' + uid + '>' + bmd_lineBreak(); end; if
(adresse <> per_adresse) then begin question := question + 'Adresse <' + per_adresse + '> auf
<' + adresse + '>' + bmd_lineBreak(); end; if (plz <> per_plz) then begin question := question +
'PLZ <' + per_plz + '> auf <' + plz + '>' + bmd_lineBreak(); end; if (iban1 <> per_iban1) then
begin question := question + 'IBAN <' + per_iban1 + '> auf <' + iban1 + '>' + bmd_lineBreak();
end; if (land_id <> per_land_id) then begin question := question + 'Land <' + per_land_id + '> auf
<' + land_id + '>' + bmd_lineBreak(); end; if ((kondition <> per_kondition) AND (kondition <> ''))
then begin question := question + 'Kondition <' + per_kondition + '> auf <' + kondition + '>' +
bmd_lineBreak(); end; if ((zahlssperre <> per_zahlssperre) AND (zahlssperre <> '')) then begin
per_zahlssperre_string := 'Nein'; zahlssperre_string := 'Nein'; if (per_zahlssperre <> '0') then begin
per_zahlssperre_string := 'Ja'; end; if (zahlssperre <> '0') then begin zahlssperre_string := 'Ja'; end;
question := question + 'Zahlssperre <' + per_zahlssperre_string + '> auf <' + zahlssperre_string +
'>' + bmd_lineBreak(); end; if ((kontoinhaber <> per_kontoinhaber) AND (kontoinhaber <> ''))
then begin question := question + 'Kontoinhaber <' + per_kontoinhaber + '> auf <' +
kontoinhaber + '>' + bmd_lineBreak(); end; rc := BMD_ASKVALUEBOOL (question,
'Stammdatenupdate - ' + KontoNr + ' - ' + per_name + ' ?'); done_list.SetKeyValue (done_string,
'1'); if (rc = 1) then begin personenid := import_KDLF (KontoNr, uid, uid_datum, name, zusatzname,
adresse, plz, land_id, iban1, iban2, iban3, kondition, zahlssperre, kontoinhaber); if
do_modify_buchungstext then begin VorerfassenModel.SetValue('MCA_VEB_BUCHUNGSTEXT',
name); end; show_kdlf (FirmenNr, FibuNr, personenid); end; FIELD_KEEP_FOCUS:=True; end; end;
end; else if auth_create then begin // feld leer --> neuen kunden anlegen if (((BMD_LEN (uid) > 0) or
(BMD_LEN (plz) > 0)) and (BMD_LEN (name) > 0)) then begin rc := KDLF_exists (uid, text,
vorerf_typ); if (rc = '0') then begin next_nummer := get_next_free_kdlf_nummer (FirmenNr, BS,
FibuNr); if (next_nummer > 0) then begin text := name; if adresse <> '' then begin text := text +
' ' + adresse; end; if ort <> '' then begin text := text + ' ' + ort; end; if uid <> '' then begin text
:= text + ' UID ' + uid + ' : ' + uid_status; end; if iban1 <> '' then begin text := text + ' ' + iban1;

```

```
end; text := text + ' Nummer: '; rc := BMD_ASKVALUEBOOL (kdlf_string + 'Nummer ' +
next_nummer + bmd_lineBreak() + bmd_lineBreak() + name + bmd_lineBreak() + 'UID <' + uid +
'>: ' + uid_status + bmd_lineBreak() + adresse + bmd_lineBreak() + ort + bmd_lineBreak() +
'IBAN: ' + iban1, kdlf_string + ' anlegen ?'); if (rc = 1) then begin personenid := import_KDLF
(next_nummer, uid, uid_datum, name, zusatzname, adresse, plz, land_id, iban1, iban2, iban3,
kondition, zahlsperre, kontoinhaber); store_created_kdlf (uid, name, vorerf_typ, personenid);
field_output_value := next_nummer; show_kdlf (FirmenNr, FibuNr, personenid); end; end; else
begin BMD_SHOWMESSAGE ('Keine next free number'); end; end; else begin field_output_value :=
rc; end; FIELD_KEEP_FOCUS:=True; end; end; result := field_output_value;
FIELD_OUTPUT_VALUE:=result;
```

TEST MAKRO

Zeile 1

Zeile 2

Zeile 3

Created 2026-07-01 07:34:31 UTC by Margarita Molina

Updated 2026-09-16 10:35:37 UTC by Gabriela Milcheva