

@A+

//==== Business-Control =====

//

// Prozedur Auf_SL_RV_Main

// OHNE E_R_G

// Info

//

//

// 15.12.2004 AI Erstellung der Prozedur

// 04.04.2022 AH ERX

// 15.07.2022 HA Quick Jump

//

// Subprozeduren

// SUB EvtInit(aEvt : event) : logic

// SUB Pflichtfelder();

// SUB CheckAufSL(aAnz : int; aBreite : float; aLaenge : float) : logic;

// SUB ReserviereSL(aAnz : int; aBreite : float; aLaenge : float; aWdh : int) : logic;

// SUB ReserviereSchrott(aMenge : float; aWdh : int) : logic;

// SUB UpdateMengen() : logic;

// SUB RefreshIfm(optaName : alpha)

// SUB Reclnit()

// SUB RecSave() : logic;

// SUB RecCleanup() : logic

// SUB RecDel()

// SUB EvtFocusInit(aEvt : event; aFocusObject : int) : logic

// SUB EvtFocusTerm(aEvt : event; aFocusObject : int) : logic

```

// SUB Auswahl(aBereich : alpha)

// SUB AusCharge()

// SUB RefreshMode(optaNoRefresh : logic);

// SUB EvtMenuCommand(aEvt : event; aMenuItem : int) : logic

// SUB EvtClicked(aEvt : event) : logic

// SUB EvtPageSelect(aEvt : event; aPage : int; aSelecting : logic) : logic

// SUB EvtMouseItem(aEvt : event; aButton : int; aHit : int; aItem : int; aID : int) : logic

// SUB EvtLstDataInit(aEvt : Event; aRecId : int);

// SUB EvtLstSelect(aEvt : event; aRecID : int) : logic

// SUB EvtClose(aEvt : event) : logic

//

//=====

@I:Def_Global

@I:Def_Rights

@I:Def_Aktionen


define begin

    cTitle : 'Reservierungen'

    cFile : 404

    cMenuName : 'Std.Bearbeiten'

    cPrefix : 'Auf_SL_RV'

    cZList : $ZL.Auf.SL.RV

    cKey : 4

end;

//=====

```

```

// EvtInit

//      Initialisieren der Applikation

//=====

sub EvtInit(

    aEvt : event;      // Ereignis

): logic

begin

    WinSearchPath(aEvt:Obj);

    gTitle  # Translate(cTitle);

    gFile   # cFile;

    gMenuName # cMenuName;

    gPrefix # cPrefix;

    gZLList # cZList;

    gKey    # cKey;


    $clmAuf.A.Menge->wpcaption # Translate('Menge')+ ' '+Auf.P.MEH.Einsatz;

    $clmGesamt->wpcaption # Translate('Gesamt')+ ' '+Auf.P.MEH.Einsatz;


    Lib_Guicom2:Underline($edAuf.A.Charge);


    SetStdAusFeld('edAuf.A.Charge' , 'Charge');


    App_Main:EvtInit(aEvt);

end;

```

```
//=====
```

```
// Pflichtfelder
```

```
//      Färbt Pflichtfelder richtig ein
```

```
//=====
```

```
sub Pflichtfelder();
```

```
begin
```

```
  if (Mode<>c_ModeNew) and (Mode<>c_ModeEdit) and
```

```
    (Mode<>c_ModeNew2) and (Mode<>c_ModeEdit2) then RETURN;// Pflichtfelder
```

```
  // Pflichtfelder
```

```
  //Lib_GuiCom:Pflichtfeld($);
```

```
end;
```

```
//=====
```

```
// CheckAufSL
```

```
//
```

```
//=====
```

```
sub CheckAufSL(aAnz : int; aBreite : float; aLaenge : float) : logic;
```

```
local begin
```

```
  Erx    : int;
```

```
  vThisID : int;
```

```
  vFound  : logic;
```

```
end;
```

```
begin
```

```
  vThisID # RecInfo(409,_RecID);
```

```
  vFound # n;
```

```

Erx # RecLink(409,401,15,_recFirst);

WHILE (vFound=n) and (Erx<=_rlocked) do begin

    if (Auf.SL.Breite=aBreite) and ("Auf.SL.Länge"=aLaenge) then begin

        vFound # y;

        BREAK;

    end;

    Erx # RecLink(409,401,15,_recNext);

END;
```

```

if (vFound) then begin

    if (aAnz+"Auf.SL.Prd.Plan.Stk">"Auf.SL.Stückzahl") then vFound # n;

end;
```

```

RecRead(409,0,_RecId,vThisID);
```

```

RETURN vFound;

end;
```

```
//=====
```

```
// ReserviereSL
```

```
//
```

```
//=====
```

```
sub ReserviereSL(aAnz : int; aBreite : float; aLaenge : float; aWdh : int) : logic;
```

```
local begin
```

```
    vThisID : int;
```

```

vFound    : logic;

vMenge    : float;

vStk      : int;

vGew      : float;

vProzent  : float;

end;

begin

todo('reserviere SL');

RETURN true;

/*****

vThisID # RecInfo(409,_RecID);

vFound # n;


RecLink(250,404,3,_recFirst);    // Artikel holen

if (Auf.A.Charge<>") then begin

    Art.C.ArtikelNr    # Auf.A.ArtikelNr;

    Art.C.Adressnr     # 0;

    Art.C.Anschriftnr  # 0;

    Art.C.Charge.Intern # Auf.A.Charge;

    Erx # RecRead(252,1,0);      // Charge holen

    if (Erx>_rLocked) then RecBufClear(252);

    end

else begin

    RecBufClear(252);

end;

```

if (Art.MEH='mm') or (Art.MEH='m') or (Art.MEH='lfdm') or (Art.MEH='lfdmm') then begin

 vMenge # cnvfi(aAnz) * aLaenge;

 if (Art.MEH='m') or (Art.MEH='lfdm') then

 vMenge # vMenge / 1000.0;

 end

else if (Art.MEH='kg') or (Art.MEH='t') then begin

 vMenge # cnvfi(aAnz) * "Art.GewichtProm" * aLaenge / 1000.0;

 if (Art.MEH='t') then

 vMenge # vMenge / 1000.0;

 end

else if (Art.MEH='qm') then begin

 vMenge # cnvfi(aAnz) * aBreite * aLaenge;

 if (Art.MEH='qm') then

 vMenge # vMenge / 1000000.0;

end;

Exr # RecLink(409,401,15,_recFirst);

WHILE (vFound=n) and (Exr<=_rlocked) do begin

 if (Auf.SL.Breite=aBreite) and ("Auf.SL.Länge"=aLaenge) then begin

 vFound # y;

 BREAK;

 end;

 Exr # RecLink(409,401,15,_recNext);

END;

if (vFound=n) or (aAnz+"Auf.SL.Prd.Plan.Stk">"Auf.SL.Stückzahl") then begin

RecRead(409,0,_RecId,vThisID);

RETURN false;

end;

vMenge # vMenge * cnvfi(aWdh);

if (Art.C.Charge.Intern<>"") and (aWdh<Art.C.Bestand.Stk) then begin

vProzent # (cnvfi(aWdh) / CnvFI(Art.C.Bestand.Stk) * 100.0);

Art_Data:SplitCharge(vProzent);

Auf.A.Charge # Art.C.Charge.Intern;

end;

RecLink(100,401,4,_recFirst); // Kunde holen

Erx # RecLink(835,401,5,_recFirst); // Auftragsart holen

if (Erx>_rLocked) then RecBufClear(835);

Auf.A.Aktionsnr # Auf.SL.Nummer;

Auf.A.AktionsPos # Auf.SL.Position;

Auf.A.AktionsPos2 # Auf.SL.lfdNr;

Aufx.A.Adressnummer # Adr.Nummer;

"Auf.A.Stückzahl" # aAnz*aWdh;

Auf.A.Dicke # Art.C.Dicke;

Auf.A.Breite # aBreite;

"Auf.A.Länge" # aLaenge;

Auf.A.Menge # vMenge;

Auf.A.MEH # Art.MEH;


```

Auf.A.MEH.Preis    # Auf.P.MEH.Preis;

Auf.A.Gewicht      # Lib_Einheiten:WandleMEH(404, "Auf.A.Stückzahl", 0.0, Auf.A.Menge, Auf.A.MEH, 'k

//todo(cnvauf(auf.a.gewicht)+'kg ');

//return false;

/*

// Umrechnen in Berechnungseinheit

Auf.A.Menge.Preis  # Lib_Einheiten:WandleMEH(404, vStk, vGew, vMenge, Auf.A.MEH, Auf.A.MEH.Pre

*/

Auf.A.AktionsTyp   # c_Akt_Prd_Plan;

Auf.A.Bemerkung    # c_AktBem_Prd_Plan;

Auf.A.AktionsDatum # today;

if (Auf_A_Data:NeuAnlegen(n,y)=false) then begin

    RecRead(409,0,_RecId,vThisID);

    RETURN false;

end;


// SL anpassen

/*

RecRead(409,1,_RecLock);

Auf.SL.Prd.Plan.Stk  # Auf.SL.Prd.Plan.Stk + "Auf.A.Stückzahl";

Auf.SL.Prd.Plan      # Auf.SL.Prd.Plan.Gew + Auf.A.Gewicht;

Auf.SL.Prd.Plan      # Auf.SL.Prd.Plan  + Auf.A.Menge;

RekReplace(409,_recUnlock,'AUTO');

*/

// DetailCharge + SumCharge reservieren

```

```

vStk # cnvif(Lib_Einheiten:WandleMEH(252, 0, 0.0, vMenge, Art.MEH, 'STK'));

vGew # Lib_Einheiten:WandleMEH(252, 0, 0.0, vMenge, Art.MEH, 'KG');

if (Art_Data:Reservierung(Auf.SL.ArtikelNr, Auf.SL.Adresse, aAnschrift, aCharge, 'AUF', Auf.SL.Nummer,

//if (Art_Data:Reservierung(vMenge, 'RV', y)=false) then begin

    RecRead(409,0,_RecId,vThisID);

    RETURN false;

end;

// AuftragsRestmenge mindern

if (AAr.ReserviereSLYN) then Art_Data:Auftrag(-1.0*vMenge);

RecRead(409,0,_RecId,vThisID);

RETURN true;

****/

end;

//=====

// ReserviereSchrott

//

//=====

sub ReserviereSchrott(aMenge : float; aWdh : int) : logic;

local begin

    vThisID : int;

```

```

vFound : logic;

vMenge : float;

vProzent : float;

end;

begin

todo('Reserviere Schrott');

RETURN true;

/*****

vThisID # RecInfo(409,_RecID);

vFound # n;

RecLink(250,404,3,_recFirst);  // Artikel holen

if (Auf.A.Charge<>") then begin

    Art.C.ArtikelNr    # Auf.A.ArtikelNr;

    Art.C.Adressnr     # 0;

    Art.C.Anschriftnr  # 0;

    Art.C.Charge.Intern # Auf.A.Charge;

    Erx # RecRead(252,1,0);      // Charge holen

    if (Erx>_rLocked) then RecBufClear(252);

    end

else begin

    RecBufClear(252);

end;

vMenge # aMenge * cnvfi(aWdh);

```

if (Art.C.Charge.Intern<>"") and (aWdh<Art.C.Bestand.Stk) then begin

vProzent # (cnvfl(aWdh) / CnvFl(Art.C.Bestand.Stk) * 100.0);

Art_Data:SplitCharge(vProzent);

Auf.A.Charge # Art.C.Charge.Intern;

end;

RecLink(100,401,4,_recFirst); // Kunde holen

Auf.A.Aktionsnr # Auf.SL.Nummer;

Auf.A.AktionsPos # Auf.SL.Position;

Auf.A.AktionsPos2 # 0;

Aufx.A.Adressnummer # Adr.Nummer;

"Auf.A.Stückzahl" # aWdh;

Auf.A.Breite # 0.0;

"Auf.A.Länge" # 0.0;

Auf.A.Menge # vMenge;

Auf.A.MEH # Art.MEH;

Auf.A.MEH.Preis # Auf.P.MEH.Preis;

Auf.A.AktionsTyp # c_Akt_Schrott;

Auf.A.Bemerkung # c_AktBem_BA_Plan;

Auf.A.AktionsDatum # today;

if (Auf_A_Data:NeuAnlegen())=false) then begin

RecRead(409,0,_RecId,vThisID);

RETURN false;

end;

```

if (Art_Data:Reservierung(vMenge, 'RV',y)=false) then begin
    RecRead(409,0,_RecId,vThisID);

    RETURN false;

end;

RecRead(409,0,_RecId,vThisID);

RETURN true;

****/

end;

```

```

//=====

// UpdateMengen

//

//=====

sub UpdateMengen() : logic;

local begin

    Erx          : int;

    vOk          : logic;

    vM1,vM2,vM3,vM4,vM5 : float;

    vMax         : float;

end;

begin

    vOk # y;

```

RecLink(250,404,3,_recFirst); // Artikel holen

\$lb.ArtikelNr->wpcaption # Art.Nummer;

\$lb.Stichwort->wpcaption # Art.Stichwort;

\$lb.EinsatzMEH->wpcaption # Art.MEH;

\$lb.EinsatzMEH2->wpcaption # Art.MEH;

\$lb.GesamtMEH->wpcaption # Art.MEH;

\$lb.RestMEH->wpcaption # Art.MEH;

\$lb.MEH1->wpcaption # Art.MEH;

\$lb.MEH2->wpcaption # Art.MEH;

\$lb.MEH3->wpcaption # Art.MEH;

\$lb.MEH4->wpcaption # Art.MEH;

\$lb.MEH5->wpcaption # Art.MEH;

if (Auf.A.Charge<>"") then begin

Art.C.ArtikelNr # Auf.A.ArtikelNr;

Art.C.Adressnr # 0;

Art.C.Anschriftnr # 0;

Art.C.Charge.Intern # Auf.A.Charge;

Erx # RecRead(252,1,0); // Charge holen

if (Erx>_rLocked) then RecBufClear(252);

end

else begin

RecBufClear(252);

end;

\$lb.Chargenbreite->wpcaption # ANum(Art.C.Breite,2);

\$lb.Chargenlaenge->wpcaption # ANum("Art.C.Länge",2);

```

$lb.Chargenmenge->wpcaption # ANum("Art.C.Verfügbar",2);

$lb.Stueck->wpcaption      # AInt("Art.C.Bestand.Stk");

if (Art.C.Bestand.Stk<>0) then

    vMax # "Art.C.Verfügbar" / CnvFI(Art.C.Bestand.Stk);

if (Art.MEH='mm') or (Art.MEH='m') or (Art.MEH='lfdm') or (Art.MEH='lfdmm') then begin

    vM1 # cnvfi($edAnzahl1->wpcaptionint) * ($edLaenge1->wpcaptionfloat);
    vM2 # cnvfi($edAnzahl2->wpcaptionint) * ($edLaenge2->wpcaptionfloat);
    vM3 # cnvfi($edAnzahl3->wpcaptionint) * ($edLaenge3->wpcaptionfloat);
    vM4 # cnvfi($edAnzahl4->wpcaptionint) * ($edLaenge4->wpcaptionfloat);
    vM5 # cnvfi($edAnzahl5->wpcaptionint) * ($edLaenge5->wpcaptionfloat);

//    vMax # "Art.C.Länge";

    if (Art.MEH='m') or (Art.MEH='lfdm') then begin

        vM1 # vM1 / 1000.0;
        vM2 # vM2 / 1000.0;
        vM3 # vM3 / 1000.0;
        vM4 # vM4 / 1000.0;
        vM5 # vM5 / 1000.0;

//        vMax # vMax / 1000.0;

    end;

end

else if (Art.MEH='qm') then begin

    vM1 # cnvfi($edAnzahl1->wpcaptionint) * ($edBreite1->wpcaptionfloat) * ($edLaenge1->wpcaptionfloat);
    vM2 # cnvfi($edAnzahl2->wpcaptionint) * ($edBreite2->wpcaptionfloat) * ($edLaenge2->wpcaptionfloat);
    vM3 # cnvfi($edAnzahl3->wpcaptionint) * ($edBreite3->wpcaptionfloat) * ($edLaenge3->wpcaptionfloat);

```

```

vM4 # cnvfi($SedAnzahl4->wpcaptionint) * ($SedBreite4->wpcaptionfloat) * ($SedLaenge4->wpcaptionfloat);
vM5 # cnvfi($SedAnzahl5->wpcaptionint) * ($SedBreite5->wpcaptionfloat) * ($SedLaenge5->wpcaptionfloat);

// vMax # "Art.C.Länge" * Art.C.Breite;

if (Art.MEH='qm') then begin

  vM1 # vM1 / 1000000.0;

  vM2 # vM2 / 1000000.0;

  vM3 # vM3 / 1000000.0;

  vM4 # vM4 / 1000000.0;

  vM5 # vM5 / 1000000.0;

// vMax # vMax / 1000000.0;

end;

end

else if (Art.MEH='kg') or (Art.MEH='t') then begin

  vM1 # cnvfi($SedAnzahl1->wpcaptionint) * "Art.GewichtProm" * ($SedLaenge1->wpcaptionfloat) / 1000.0;
  vM2 # cnvfi($SedAnzahl2->wpcaptionint) * "Art.GewichtProm" * ($SedLaenge2->wpcaptionfloat) / 1000.0;
  vM3 # cnvfi($SedAnzahl3->wpcaptionint) * "Art.GewichtProm" * ($SedLaenge3->wpcaptionfloat) / 1000.0;
  vM4 # cnvfi($SedAnzahl4->wpcaptionint) * "Art.GewichtProm" * ($SedLaenge4->wpcaptionfloat) / 1000.0;
  vM5 # cnvfi($SedAnzahl5->wpcaptionint) * "Art.GewichtProm" * ($SedLaenge5->wpcaptionfloat) / 1000.0;

// vMax # "Art.C.Länge" * Art.C.Breite;

if (Art.MEH='t') then begin

  vM1 # vM1 / 1000.0;

  vM2 # vM2 / 1000.0;

  vM3 # vM3 / 1000.0;

  vM4 # vM4 / 1000.0;

  vM5 # vM5 / 1000.0;

// vMax # vMax / 1000000.0;

```



```

    end;

end;

$lb.Summe5->wpcaption # ANum(vM5,2);

$lb.Summe4->wpcaption # ANum(vM4,2);

$lb.Summe3->wpcaption # ANum(vM3,2);

$lb.Summe2->wpcaption # ANum(vM2,2);

$lb.Summe1->wpcaption # ANum(vM1,2);

$lb.Einsatzmenge->wpcaption # ANum(vMax,2);


$lb.Gesamtsumme->wpcaption # ANum(vM1+vM2+vM3+vM4+vM5,2);

$lb.Rest->wpcaption      # ANum(vMax-(vM1+vM2+vM3+vM4+vM5),2);


if (vMax-(vM1+vM2+vM3+vM4+vM5)<0.0) then vOk # n;

if (Art.C.Bestand.Stk<$edStueck->wpcaptionint) then vOk # n;


RETURN vOk;

end;


//=====

// Refreshlfm

//      "Infomasken" refreshen

//=====

sub Refreshlfm(

    opt aName : alpha;

)

```

local begin

 vTmp : int;

end;

begin

 UpdateMengen();

 // veränderte Felder in Objekte schreiben

 if (aName<>") then begin

 vTmp # gMdi->winsearch(aName);

 if (vTmp<>0) then

 vTmp->winupdate(_WinUpdFld2Obj);

 end;

 // einfärben der Pflichtfelder

 if (Mode=c_ModeNew) or (Mode=c_ModeNew2) or

 (Mode=c_ModeEdit) or (Mode=c_ModeEdit2) then

 Pflichtfelder();

end;

//=====

// Reclnit

// Init für Änderung und Neuanlage

//=====

sub Reclnit()

local begin

 Erx : int;

 vHdl : int;

 vNr : int;

end;

begin

 \$NB.Page1->wpdisabled # n;

 vNr # Auf.SL.lfdNr;

 vHdl # \$DL.Stueckliste;

 vHdl->WinLstDatLineRemove(_WinLstDatLineAll);

 Erx # RecLink(409,401,15,_RecFirst);

 WHILE (Erx<=_rLocked) do begin

 vHdl->WinLstDatLineAdd(Auf.SL.ArtikelNr,_WinLstDatLineLast);

 vHdl->WinLstCellSet(Auf.SL.Bemerkung,_WinLstDatLineLast,2);

 vHdl->WinLstCellSet(AInt("Auf.SL.Stückzahl"),3,_WinLstDatLineLast);

 vHdl->WinLstCellSet(ANum("Auf.SL.Breite",Set.Stellen.Breite),4,_WinLstDatLineLast);

 vHdl->WinLstCellSet(ANum("Auf.SL.Länge","Set.Stellen.Länge"),5,_WinLstDatLineLast);

 vHdl->WinLstCellSet(ANum("Auf.SL.Menge",Set.Stellen.Menge),6,_WinLstDatLineLast);

 vHdl->WinLstCellSet(ANum(Auf.SL.Menge - Auf.SL.Prd.Plan,Set.Stellen.Menge),7,_WinLstDatLineLast);

 Erx # RecLink(409,401,15,_RecNext);

 END;

vHdl->Winupdate();

Auf.SL.lfdNr # vNr;

RecRead(409,1,0);

Auf.A.ArtikelNr # Auf.SL.ArtikelNr;

Auf.A.Nummer # Auf.SL.Nummer;

Auf.A.Position # Auf.SL.Position;

\$edStueck->wpcaptionint # 1;

\$edAnzahl1->wpcaptionint # 0;

\$edAnzahl2->wpcaptionint # 0;

\$edAnzahl3->wpcaptionint # 0;

\$edAnzahl4->wpcaptionint # 0;

\$edAnzahl5->wpcaptionint # 0;

\$edBreite1->wpcaptionfloat # 0.0;

\$edBreite2->wpcaptionfloat # 0.0;

\$edBreite3->wpcaptionfloat # 0.0;

\$edBreite4->wpcaptionfloat # 0.0;

\$edBreite5->wpcaptionfloat # 0.0;

\$edLaenge1->wpcaptionfloat # 0.0;

\$edLaenge2->wpcaptionfloat # 0.0;

\$edLaenge3->wpcaptionfloat # 0.0;

\$edLaenge4->wpcaptionfloat # 0.0;

```
$edLaenge5->wpcaptionfloat # 0.0;
```

```
$lb.Stueck->wpcaption      # ";
```

```
$lb.Chargenbreite->wpcaption # ";
```

```
$lb.Chargenlaenge->wpcaption # ";
```

```
$lb.Chargenmenge->wpcaption # ";
```

```
$lb.Einsatzmenge->wpcaption # ";
```

```
$lb.Summe1->wpcaption      # ";
```

```
$lb.Summe2->wpcaption      # ";
```

```
$lb.Summe3->wpcaption      # ";
```

```
$lb.Summe4->wpcaption      # ";
```

```
$lb.Summe5->wpcaption      # ";
```

```
$lb.Gesamtsumme->wpcaption # ";
```

```
$lb.Rest->wpcaption        # ";
```

```
$cb.schrott->wpCheckState  # _WinStateChkUnchecked;
```

```
// Felder Disablen durch:
```

```
if (Auf.P.MEH.Einsatz<>'qm') then begin
```

```
    Lib_GuiCom:Disable($edBreite1);
```

```
    Lib_GuiCom:Disable($edBreite2);
```

```
    Lib_GuiCom:Disable($edBreite3);
```

```
    Lib_GuiCom:Disable($edBreite4);
```

```
    Lib_GuiCom:Disable($edBreite5);
```

```
end
```

```
else begin
```

```
Lib_GuiCom:Enable($sedBreite1);  
Lib_GuiCom:Enable($sedBreite2);  
Lib_GuiCom:Enable($sedBreite3);  
Lib_GuiCom:Enable($sedBreite4);  
Lib_GuiCom:Enable($sedBreite5);  
  
end;  
  
// Focus setzen auf Feld:  
$sedAuf.A.Charge->WinFocusSet(true);  
  
end;
```

```
//=====
```

// RecSave

// vor Speicherung

```
//=====
```

```
sub RecSave() : logic;
```

```
local begin
```

```
  vCharge : alpha;
```

```
  vRest  : float;
```

```
end;
```

```
begin
```

```
if ($sedAnzahl1->wpcaptionint<>0) then begin
```

```
  if (CheckAufSL($sedAnzahl1->wpcaptionint,$sedBreite1->wpcaptionfloat,$sedLaenge1->wpcaptionfloat)=fa
```

```
    Msg(409002,'1',0,0,0);
```

```
        RETURN false;

    end;

end;

if ($SedAnzahl2->wpcaptionint<>0) then begin

    if (CheckAufSL($SedAnzahl2->wpcaptionint,$SedBreite2->wpcaptionfloat,$SedLaenge2->wpcaptionfloat)=fa

        Msg(409002,'2',0,0,0);

        RETURN false;

    end;

end;

if ($SedAnzahl3->wpcaptionint<>0) then begin

    if (CheckAufSL($SedAnzahl3->wpcaptionint,$SedBreite3->wpcaptionfloat,$SedLaenge3->wpcaptionfloat)=fa

        Msg(409002,'3',0,0,0);

        RETURN false;

    end;

end;

if ($SedAnzahl4->wpcaptionint<>0) then begin

    if (CheckAufSL($SedAnzahl4->wpcaptionint,$SedBreite4->wpcaptionfloat,$SedLaenge4->wpcaptionfloat)=fa

        Msg(409002,'4',0,0,0);

        RETURN false;

    end;

end;

if ($SedAnzahl5->wpcaptionint<>0) then begin

    if (CheckAufSL($SedAnzahl5->wpcaptionint,$SedBreite5->wpcaptionfloat,$SedLaenge5->wpcaptionfloat)=fa

        Msg(409002,'5',0,0,0);

        RETURN false;

    end;
```

end;

if (UpdateMengen())=false) then begin

Msg(409001,',',0,0,0);

RETURN false;

end;

vCharge # Auf.A.Charge;

TRANSON;

if (\$SedAnzahl1->wpcaptionint<>0) then begin

if (ReserviereSL(\$SedAnzahl1->wpcaptionint,\$SedBreite1->wpcaptionfloat,\$SedLaenge1->wpcaptionfloat, \$

TRANSBRK;

Auf.A.Charge # vCharge;

Msg(409002,'1',0,0,0);

RETURN false;

end;

end;

if (\$SedAnzahl2->wpcaptionint<>0) then begin

if (ReserviereSL(\$SedAnzahl2->wpcaptionint,\$SedBreite2->wpcaptionfloat,\$SedLaenge2->wpcaptionfloat, \$

TRANSBRK;

Auf.A.Charge # vCharge;

Msg(409002,'2',0,0,0);

RETURN false;

end;

end;


```
if ($SedAnzahl3->wpcaptionint<>0) then begin
```

```
    if (ReserviereSL($SedAnzahl3->wpcaptionint,$SedBreite3->wpcaptionfloat,$SedLaenge3->wpcaptionfloat, $
```

```
        TRANSBRK;
```

```
        Auf.A.Charge # vCharge;
```

```
        Msg(409002,'3',0,0,0);
```

```
        RETURN false;
```

```
    end;
```

```
end;
```

```
if ($SedAnzahl4->wpcaptionint<>0) then begin
```

```
    if (ReserviereSL($SedAnzahl4->wpcaptionint,$SedBreite4->wpcaptionfloat,$SedLaenge4->wpcaptionfloat, $
```

```
        TRANSBRK;
```

```
        Auf.A.Charge # vCharge;
```

```
        Msg(409002,'4',0,0,0);
```

```
        RETURN false;
```

```
    end;
```

```
end;
```

```
if ($SedAnzahl5->wpcaptionint<>0) then begin
```

```
    if (ReserviereSL($SedAnzahl5->wpcaptionint,$SedBreite5->wpcaptionfloat,$SedLaenge5->wpcaptionfloat, $
```

```
        TRANSBRK;
```

```
        Auf.A.Charge # vCharge;
```

```
        Msg(409002,'5',0,0,0);
```

```
        RETURN false;
```

```
    end;
```

```
end;
```

```
if ($cb.schrott->wpcheckState=_WinStateChkChecked) and ($lb.Rest->wpcaption<>") then begin
```

```

vRest # Cnvfa($lb.Rest->wpCaption);

if (ReserviereSchrott(vRest, $edStueck->wpcaptionint )=false) then begin

    TRANSBRK;

    Auf.A.Charge # vCharge;

    Msg(409002,'99',0,0,0);

    RETURN false;

end;

end;

/*

TRANSBRK;

msg(123123,'ERZWUNGENER FEHLER !!!',0,0,0);

RETURN false;

*/

TRANSOFF;


RETURN true; // Speichern erfolgreich

end;


//=====

// RecCleanup

//      Aufräumen bei "Cancel"

//=====

sub RecCleanup() : logic

begin

```

```

RETURN true;

end;

//=====

// RecDel

//      Satz soll gelöscht werden

//=====

sub RecDel() : int

local begin

    vBuf252 : int;

    Erx    : int;

end;

begin

    // Diesen Eintrag wirklich löschen?

    if (Msg(000001,",_WinIcoQuestion,_WinDialogYesNo,2)=_WinIdNo) then RETURN _rOK;

    Erx # RecLink(409,404,5,_RecFirst);    // Stückliste holen

    if (Erx<>_roK) then RETURN Erx;

    RecLink(250,404,3,_recFirst);          // Artikel holen

    if (Auf.A.Charge<>") then begin

        Erx # RecLink(252,404,4,_recFirst); // Charge holen

        if (Erx>_rLocked) then RecBufClear(252);

    end

```

else begin

RecBufClear(252);

end;

vBuf252 # RecBufCreate(252);

RecBufCopy(252,vBuf252);

TRANSON;

/**1910

if (Art_Data:Reservierung(Auf.A.Menge * (-1.0), 'RV',y)=false) then begin

TRANSBRK;

RecBufDestroy(vBuf252);

RETURN;

end;

***/

Erx # RecLink(835,401,5,_recFirst); // Auftragsart holen

if (Erx>_rLocked) then RecBufClear(835);

// AuftragRestsmenge erhöhen

if (AAr.ReserviereSLYN) then Art_Data:Auftrag(Auf.A.Menge);

if (Auf_A_Data:Entfernen())=false) then begin

TRANSBRK;

RecBufDestroy(vBuf252);

RETURN _rNorec;

end;

```

RecBufCopy(vBuf252,252);

RecBufDestroy(vBuf252);


// alles ok!

TRANSOFF;


RETURN _rok;

end;


//=====

// EvtFocusInit

//      Fokus auf Objekt neu gesetzt

//=====

sub EvtFocusInit (

    aEvt          : event;    // Ereignis

    aFocusObject   : int      // vorheriges Objekt

) : logic

begin

    // Auswahlfelder aktivieren

    if (Lib_Pflichtfelder:TypAuswahlFeld(aEvt:Obj)<>"") then

        Lib_GuiCom:AuswahlEnable(aEvt:Obj);

    else

        Lib_GuiCom:AuswahlDisable(aEvt:Obj);

```

end;

//=====

// EvtFocusTerm

// Fokus vom Objekt wegbewegen

//=====

sub EvtFocusTerm (

 aEvt : event; // Ereignis

 aFocusObject : int // neu zu fokusierendes Objekt

) : logic

begin

 // logische Prüfung von Verknüpfungen

 RefreshIfm(aEvt:Obj->wpName);

 RETURN true;

end;

//=====

// Auswahl

// Auswahlliste öffnen

//=====

sub Auswahl(

```

aBereich : alpha;

)

local begin

vA      : alpha;

vFilter : int;

vQ      : alpha(4090);

vHdl    : int;

Erx     : int;

end;


begin


case aBereich of

'Charge' : begin

    RecBufClear(252);      // ZIELBUFFER LEEREN

    gMDI # Lib_GuiCom:AddChildWindow(gMDI,'Art.C.Verwaltung','Auf_SL_RV_Main:AusCharge');

    VarInstance(WindowBonus,cnvIA(gMDI->wpcustom));

/*

vFilter # RecFilterCreate(252,1);

vFilter->RecFilterAdd(1,_FltAND,_FltEq,Auf.A.ArtikelNr);

vFilter->RecFilterAdd(2,_FltAND,_FltEq,0);

vFilter->RecFilterAdd(4,_FltAND,_FltAbove,");

gZLList->wpDbFilter # vFilter;

gKey # 1;

*/

vQ # "";

```

```

Lib_Sel:QAlpha(var vQ, 'Art.C.ArtikelNr'      , '=', Auf.A.ArtikelNr);

Lib_Sel:QInt(var vQ, 'Art.C.Adressnr'      , '=', 0);

Lib_Sel:QAlpha(var vQ, 'Art.C.Charge.Intern' , '>', '');

Lib_Sel:QDate(var vQ, 'Art.C.Ausgangsdatum' , '=', 0.0.0);

vHdl # SelCreate(252, gKey);

Erx # vHdl->SelDefQuery("", vQ);

if (Erx != 0) then Lib_Sel:QError(vHdl);

// speichern, starten und Name merken...

w_SelName # Lib_Sel:SaveRun(var vHdl,0,n);

// Liste selektieren...

gZLList->wpDbSelection # vHdl;


Lib_GuiCom:RunChildWindow(gMDI,gFrmMain,_Winaddhidden);

gMdi->WinUpdate(_WinUpdOn);

end;

end;

end;


//=====

// AusCharge

//

//=====

sub AusCharge()

begin

```



```

if (gSelected<>0) then begin

    RecRead(252,0,_RecId,gSelected);

    // Feldübernahme

    Auf.A.Charge # Art.C.Charge.Intern;

    gSelected # 0;

end;

// Focus auf Editfeld setzen:

$edAuf.A.Charge->Winfocusset(false);

// ggf. Labels refreshen

RefreshIfm();

end;


//=====

// RefreshMode

//      Setzt alle Menüs/Toolbars/Buttons passend zum Modus

//=====

sub RefreshMode(opt aNoRefresh : logic);

local begin

    d_MenuItem : int;

    vHdl : int;

end

begin

    gMenu # gFrmMain->WinInfo(_WinMenu);

```

\$NB.Page1->wpdisabled # mode=c_modeList;

// Button & Menüs sperren

vHdl # gMdi->WinSearch('New');

if (vHdl <> 0) then

vHdl->wpDisabled # (vHdl->wpDisabled) or (Rechte[Rgt_Auf_SL_RV_Anlegen]=n);

vHdl # gMenu->WinSearch('Mnu.New');

if (vHdl <> 0) then

vHdl->wpDisabled # (vHdl->wpDisabled) or (Rechte[Rgt_Auf_SL_RV_Anlegen]=n);

vHdl # gMdi->WinSearch('Edit');

if (vHdl <> 0) then

vHdl->wpDisabled # (y) or (vHdl->wpDisabled) or (w_Auswahlmode) or (Rechte[Rgt_Auf_SL_RV_Aende

vHdl # gMenu->WinSearch('Mnu.Edit');

if (vHdl <> 0) then

vHdl->wpDisabled # (y) or (vHdl->wpDisabled) or (w_Auswahlmode) or (Rechte[Rgt_Auf_SL_RV_Aende

vHdl # gMdi->WinSearch('Delete');

if (vHdl <> 0) then

vHdl->wpDisabled # ((Mode<>c_ModeList) and (Mode<>c_ModeView)) or (Rechte[Rgt_Auf_SL_RV_Lo

vHdl # gMenu->WinSearch('Mnu.Delete');

if (vHdl <> 0) then

vHdl->wpDisabled # ((Mode<>c_ModeList) and (Mode<>c_ModeView)) or (Rechte[Rgt_Auf_SL_RV_Lo

if (Mode<>c_ModeOther) and (Mode<>c_ModeList) and (aNoRefresh=false) then RefreshIfm();

```

end;

//=====

// EvtMenuCommand

//      Menüpunkt aufgerufen

//=====

sub EvtMenuCommand (

    aEvt      : event;    // Ereignis

    aMenuItem : int      // Menüeintrag

) : logic

local begin

    vHdl    : int;

    vMode   : alpha;

    vParent : int;

    vTmp    : int;

end;

begin

    if (Mode=c_ModeList) then RecRead(gFile,0,0,gZLList->wpdbrecid);

    case (aMenuItem->wpName) of

        'Mnu.Protokoll' : begin

            PtD_Main:View(gFile,Auf.A.Anlage.Datum, Auf.A.Anlage.Zeit, Auf.A.Anlage.User);

        end;

```

```
end; // case
```

```
end;
```

```
//=====
```

```
// EvtClicked
```

```
//      Button gedrückt
```

```
//=====
```

```
sub EvtClicked (
```

```
    aEvt          : event;    // Ereignis
```

```
) : logic
```

```
begin
```

```
if (Mode=c_ModeView) then RETURN true;
```

```
case (aEvt:Obj->wpName) of
```

```
    'bt.Charge' :  Auswahl('Charge');
```

```
    'bt.xxxxx'  :  Auswahl('...');
```

```
    'bt.xxxxx'  :  Auswahl('...');
```

```
end;
```

```
end;
```

```
//=====
```

```
// EvtPageSelect
```

```
//      Seitenauswahl von Notebooks
```

```
//=====
```

```
sub EvtPageSelect(
```

```
    aEvt      : event;    // Ereignis
```

```
    aPage     : int;
```

```
    aSelecting : logic;
```

```
) : logic
```

```
begin
```

```
    RETURN true;
```

```
end;
```

```
//=====
```

```
// EvtMouseItem
```

```
//      Mausklicks in Listen
```

```
//=====
```

```
sub EvtMouseItem(
```

```
    aEvt      : event;    // Ereignis
```

```
    aButton   : int;      // Button
```

```
    aHit      : int;      // Hitcode
```

```
    aItem     : int;      // Item
```

```
    aID       : int;      // ID
```

```
) : logic
```

```
local begin
```

```
    vHdl : int;
```

```

vA : alpha;

end;

begin

    if (altem=0) then RETURN false;

    if (aEvt:Obj->wpname='DL.Stueckliste') then begin
        if (aButton & _WinMouseDouble>0) then begin

            //    RecRead(409,0,_RecId,aEvt:obj->wpDbRecId); // Auf-SL lesen

            vHdl # $DL.Stueckliste;

            if ($edAnzahl1->wpcaptionint=0) then begin
                vHdl->WinLstCellGet(vA,4,_WinLstDatLineCurrent);
                $edBreite1->wpcaptionfloat # Cnvfa(vA);
                vHdl->WinLstCellGet(vA,5,_WinLstDatLineCurrent);
                $edLaenge1->wpcaptionfloat # Cnvfa(vA);
                $edAnzahl1->winfocusset();
            end
        else
            if ($edAnzahl2->wpcaptionint=0) then begin
                vHdl->WinLstCellGet(vA,4,_WinLstDatLineCurrent);
                $edBreite2->wpcaptionfloat # Cnvfa(vA);
                vHdl->WinLstCellGet(vA,5,_WinLstDatLineCurrent);
                $edLaenge2->wpcaptionfloat # Cnvfa(vA);
                $edAnzahl2->winfocusset();
            end
        end
    end
end

```

```

end

else

if ($edAnzahl3->wpcaptionint=0) then begin

    vHdl->WinLstCellGet(vA,4,_WinLstDatLineCurrent);

    $edBreite3->wpcaptionfloat # Cnvfa(vA);

    vHdl->WinLstCellGet(vA,5,_WinLstDatLineCurrent);

    $edLaenge3->wpcaptionfloat # Cnvfa(vA);

    $edAnzahl3->winfocusset();

end

else

if ($edAnzahl4->wpcaptionint=0) then begin

    vHdl->WinLstCellGet(vA,4,_WinLstDatLineCurrent);

    $edBreite4->wpcaptionfloat # Cnvfa(vA);

    vHdl->WinLstCellGet(vA,5,_WinLstDatLineCurrent);

    $edLaenge4->wpcaptionfloat # Cnvfa(vA);

    $edAnzahl4->winfocusset();

end

else

if ($edAnzahl5->wpcaptionint=0) then begin

    vHdl->WinLstCellGet(vA,4,_WinLstDatLineCurrent);

    $edBreite5->wpcaptionfloat # Cnvfa(vA);

    vHdl->WinLstCellGet(vA,5,_WinLstDatLineCurrent);

    $edLaenge5->wpcaptionfloat # Cnvfa(vA);

    $edAnzahl5->winfocusset();

end;

```

```

    gMdi->winupdate();

end;

end;

end;

//=====

// EvtLstDataInit

//

//=====

sub EvtLstDataInit(
    aEvt    : Event;
    aRecId  : int;
    Opt aMark : logic;
);
begin
    // Refreshmode();
end;

//=====

// EvtLstSelect

//      Zeilenauswahl von RecList/DataList

//=====

sub EvtLstSelect(

```



```

aEvt          : event;    // Ereignis

aRecID        : int;

) : logic

begin

    RecRead(gFile,0,_recid,aRecID);

// RefreshMode(y); // falls Menüs gesetzte werden sollen

end;


//=====

// EvtClose

//      Schliessen eines Fensters

//=====

sub EvtClose

(

aEvt          : event;    // Ereignis

): logic

begin

    RETURN true;

end;


sub JumpTo(

aName : alpha;

aBuf  : int);

begin

```

```
if ((aName =^ 'edAuf.A.Charge') AND (aBuf->Auf.A.Charge<>")) then begin
```

```
    RekLink(252,404,4,2);  // Charge holen
```

```
    Lib_Guicom2:JumpToWindow('Art.C.Verwaltung');
```

```
    RETURN;
```

```
end;
```

```
end;
```

```
//=====
```

```
//=====
```

```
//=====
```

```
//=====
```