

@A+

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

//

// Prozedur Adr_V2_Main

// OHNE E_R_G

// Info AdrVpg ist nur Verpackung + VorlageAuftrag/Bestellung

//

//

// 08.02.2021 AH Erstellung der Prozedur

// 27.07.2021 AH ERX

// 12.07.2022 HA Quick Jump

//

// Subprozeduren

// SUB EvtInit(aEvt : event) : logic

// SUB Pflichtfelder();

// SUB RefreshIfm(optaName : alpha)

// SUB RecInit(opt aBehalten : logic);

// 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 AusUnterlage()

// SUB AusUmverpackung()

// SUB AusZwischenlage()

```

// SUB AusVorlageAuf()

// SUB AusVerwiegungsart()

// SUB AusEtikettentyp()

// SUB RefreshMode(optaNoRefresh : logic);

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

// SUB EvtClicked(aEvt : event) : logic

// SUB EvtChanged(aEvt : event) : logic

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

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

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

// SUB EvtClose(aEvt : event) : logic

// SUB EvtPosChanged ( aEvt : event; aRect : rect; aClientSize : point; aFlags : int ) : logic

// SUB Skizzendaten();

// SUB EvtTimer...

//

//=====

@I:Def_Global

@I:Def_Rights

define begin

cTitle : 'Verpackungsvorschriften'

cFile : 105

cMenuName : 'Adr.V.Bearbeiten'

cPrefix : 'Adr_V2'

cZList : $ZL.Adr.Verpackungen

cKey : 1

```

```
cTxtRtf(a) : '~105.'+CnvAI(Adr.V.Adressnr,_FmtNumLeadZero | _FmtNumNoGroup,0,7)+'.'+CnvAi(a,_FmtNumNoGroup,0,7)
end;
```

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

```
// EvtInit
```

```
//      Initialisieren der Applikation
```

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

```
sub EvtInit(
```

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

```
): logic
```

```
local begin
```

```
    vHdl : int;
```

```
    vPar : int;
```

```
    vRect : Rect;
```

```
end;
```

```
begin
```

```
    WinSearchPath(aEvt:Obj);
```

```
    gTitle  # Translate(cTitle);
```

```
    gFile   # cFile;
```

```
    gMenuName # cMenuName;
```

```
    gPrefix  # cPrefix;
```

```
    gZLList  # cZList;
```

```
    gKey     # cKey;
```

```
$lb.Kunde1->wpcustom # aint(Adr.Nummer);
```

```
// Verpackungstitel setzen

if(Set.Vpg1.Titel <> "") then

    $lAdr.V.VpgText1 -> wpcaption # Set.Vpg1.Titel;

if(Set.Vpg2.Titel <> "") then

    $lAdr.V.VpgText2 -> wpcaption # Set.Vpg2.Titel;

if(Set.Vpg3.Titel <> "") then

    $lAdr.V.VpgText3 -> wpcaption # Set.Vpg3.Titel;

if(Set.Vpg4.Titel <> "") then

    $lAdr.V.VpgText4 -> wpcaption # Set.Vpg4.Titel;

if(Set.Vpg5.Titel <> "") then

    $lAdr.V.VpgText5 -> wpcaption # Set.Vpg5.Titel;

if(Set.Vpg6.Titel <> "") then

    $lAdr.V.VpgText6 -> wpcaption # Set.Vpg6.Titel;
```

```
Lib_Guicom2:Underline($edAdr.V.Verwiegungsart);

Lib_Guicom2:Underline($edAdr.V.Etikettentyp);

Lib_Guicom2:Underline($edVorlageAuf);

Lib_Guicom2:Underline($edAdr.V.Zwischenlage);

Lib_Guicom2:Underline($edAdr.V.Unterlage);

Lib_Guicom2:Underline($edAdr.V.Umverpackung);

Lib_Guicom2:Underline($edAdr.V.RtfText1);
```

```
SetStdAusFeld('edVorlageAuf' , 'VorlageAuf');
```

```

SetStdAusFeld('edAdr.V.Zwischenlage' , 'Zwischenlage');

SetStdAusFeld('edAdr.V.Unterlage' , 'Unterlage');

SetStdAusFeld('edAdr.V.Umverpackung' , 'Umverpackung');

SetStdAusFeld('edAdr.V.Verwiegungsart', 'Verwiegungsart');

SetStdAusFeld('edAdr.V.Etikettentyp' , 'Etikettentyp');


RunAFX('Adr.V.Init.Pre', aint(aEvt:Obj));

App_Main:EvtInit(aEvt);

RunAFX('Adr.V.Init', aint(aEvt:Obj));

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($edAdr.V.lfdNr);

end;


//=====

```

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

```
sub RtfLoad();
```

```
begin
```

```
if (Adr.V.RtfText1<>0) then begin
```

```
    Lib_Texte:RtfTextRead($Adr.V.RTF, cTxtRtf(Adr.V.RtfText1));
```

```
end
```

```
else begin
```

```
    Lib_Texte:RtfTextRead($Adr.V.RTF, cTxtRtf(Adr.V.LfdNr));
```

```
end;
```

```
end;
```

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

```
// Refreshlfm
```

```
//      "Infomasken" refreshen
```

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

```
sub Refreshlfm(
```

```
    opt aName    : alpha;
```

```
    opt aChanged : logic;
```

```
)
```

```
local begin
```

```
    Erx    : int;
```

```
    vTxtHdl : int;
```

```
    vTmp    : int;
```

```
    vBuf100 : int;
```

end;

begin

\$lb.Kunde1->wpcaption # Adr.Stichwort;

\$lb.Kunde2->wpcaption # Adr.Stichwort;

\$lb.Verpackungsnr2->wpcaption # aint(Adr.V.lfdNr);

Erx # RecLink(818,105,4,_recFirst); // Verwiegungsart holen

if (Erx<=_rLocked) then

\$lb.Verwiegungsart->wpcaption # VWa.Bezeichnung.L1

else

\$lb.Verwiegungsart->wpcaption # ";

if (aName="") then begin

if (Adr.V.VorlageAuf<>0) then

\$edVorlageAuf->wpcaption # aint(Adr.V.VorlageAuf)+'/'+aint(Adr.V.VorlageAufPos)

else

\$edVorlageAuf->wpcaption # ";

end;

Erx # RecLink(840,105,3,_recFirst); // Etikettentyp holen

if (Erx<=_rLocked) then

\$lb.Etikettentyp->wpcaption # Eti.Bezeichnung

else

\$lb.Etikettentyp->wpcaption # ";

```
if (aName='edAdr.V.RtfText1') and (($edAdr.V.RtfText1->wpchanged) or (aChanged)) then begin  
    if (Adr.V.RtfText1<>0) then  
        RtfLoad();  
end;
```

```
vTxtHdl # $Adr.V.RTF->wpdbtextBuf;  
if (vTxtHdl=0) then begin  
    vTxtHdl # TextOpen(32);  
    $Adr.V.RTF->wpdbTextBuf # vTxtHdl;  
end;
```

```
if (Mode<>c_ModeEdit) and (aName='') then  
    RtfLoad();
```

```
// 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();
```

```
// dynamische Pflichtfelder einfärben
```



```
Lib_Pflichtfelder:PflichtfelderEinfuerben();  
end;
```

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

```
// Reclnit
```

```
//      Init für Änderung und Neuanlage
```

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

```
sub Reclnit(opt aBehalten : logic);
```

```
local begin
```

```
    Erx    : int;
```

```
    vNewLfdNr : int;
```

```
    vBuf105  : int;
```

```
    vTxtHdl  : int;
```

```
end;
```

```
begin
```

```
// Ankerfunktion?
```

```
if (aBehalten) then begin
```

```
    if (RunAFX('Adr.V.Reclnit', '1') < 0) then
```

```
        RETURN;
```

```
end
```

```
else begin
```

```
    if (RunAFX('Adr.V.Reclnit', '0') < 0) then
```

```
        RETURN;
```

```
end;
```

```
if (Mode=c_ModeNew) then begin
```

```
  vTxtHdl # $Adr.V.RTF->wpdbTextBuf;
```

```
  if (vTxtHdl<>0) then begin
```

```
    TextClear(vTxtHdl);
```

```
    $Adr.V.RTF->WinUpdate(_WinUpdBuf2Obj);
```

```
  end;
```

```
if (aBehalten = false) then begin // 25.03.2011 MS Vogel Bauer (Prj. 1161/326)
```

```
  RecBufClear(105);
```

```
//  Adr.V.EinkaufYN # true;
```

```
//  Adr.V.VerkaufYN # true;
```

```
if (w_AppendNr<>0) then begin
```

```
  RecRead(105,0,_recId,w_AppendNr);
```

```
  FOR Erx # RecLink(106,105,1,_recFirst)
```

```
  LOOP Erx # RecLink(106,105,1,_recNext)
```

```
  WHILE (Erx<=_rLocked) do begin
```

```
    Adr.V.AF.Verpacknr # 32000;
```

```
    RekInsert(106,_recUnlock,'MAN');
```

```
    Adr.V.AF.Verpacknr # Adr.V.lfdNr;
```

```
    RecRead(106,1,0);
```

```
  END;
```

```
w_BinKopieVonDatei # gFile;  
w_BinKopieVonRecID # RecInfo(gFile, _recid);  
w_AppendNr          # 0;  
end;
```

```
end
```

```
else begin
```

```
  SelRecInsert(gZLList->wpDbSelection, 105);
```

```
  FOR Erx # RecLink(106,105,1,_recFirst)
```

```
  LOOP Erx # RecLink(106,105,1,_recNext)
```

```
  WHILE (Erx<=_rLocked) do begin
```

```
    Adr.V.AF.Verpacknr # 32000;
```

```
    RekInsert(106,_recUnlock,'MAN');
```

```
    Adr.V.AF.Verpacknr # Adr.V.lfdNr;
```

```
    RecRead(106,1,0);
```

```
  END;
```

```
w_BinKopieVonDatei # gFile;
```

```
w_BinKopieVonRecID # RecInfo(gFile, _recid);
```

```
end;
```

```
// letzte Verpackung der Adresse lesen
```

```
vBuf105 # RekSave(105);
```

```
Erx # RecLink(105, 100, 33,_recLast);
```

```

if(Erx > _rLocked) then

    RecBufClear(105);

vNewLfdNr # Adr.V.lfdNr + 1;    // letzte Verpackungsnummer um 1 erhoehen

RekRestore(vBuf105);


// neuen Datensatz vorbelegen

Adr.V.lfdNr # vNewLfdNr;

Adr.V.Adressnr # Adr.Nummer;


// Focus setzen auf Feld:

$edAdr.V.lfdNr->WinFocusSet(true);

end

else begin


// Focus setzen auf Feld:

$edAdr.V.KundenArtNr->WinFocusSet(true);

end;


gMdi -> WinUpdate();

end;


//=====

// RecSave

//      vor Speicherung

```

//=====

sub RecSave() : logic;

local begin

 Erx : int;

 vNr : word

end;

begin

 // dynamische Pflichtfelder überprüfen

 if (!Lib_Pflichtfelder:PflichtfelderPruefenVorSpeichern()) then

 RETURN false;

 // logische Prüfung

 If (Adr.V.lfdnr=0) then begin

 Msg(001200,Translate('Verpackungsnummer'),0,0,0);

 \$NB.Main->wpcurrent # 'NB.Page1';

 \$edAdr.v.lfdNr->WinFocusSet(true);

 RETURN false;

 end;

 if (mode=c_ModeNew) then begin

 Erx # RecRead(gFile,1,_RecTest);

 if (Erx=_rOk) then begin

 Msg(105000,"",0,0,0);

 \$NB.Main->wpcurrent # 'NB.Page1';

 \$edAdr.v.lfdNr->WinFocusSet(true);

 RETURN false;

end;

end;

If (Adr.V.VorlageAuf<>0) then begin

if (Adr.V.EinkaufYN) then begin

 Erx # Ein_Data:read(Adr.V.VorlageAuf, Adr.V.VorlageAufPos,y);

 if (Erx<500) or (Ein.Vorgangstyp<>c_VorlageAuf) then begin

 Msg(001201,Translate('Vorlage'),0,0,0);

 \$NB.Main->wpcurrent # 'NB.Page1';

 \$edVorlageAuf->WinFocusSet(true);

 RETURN false;

 end;

end

else begin

 Erx # Auf_Data:read(Adr.V.VorlageAuf, Adr.V.VorlageAufPos,y);

 if (Erx<400) or (Auf.Vorgangstyp<>c_VorlageAuf) then begin

 Msg(001201,Translate('Vorlage'),0,0,0);

 \$NB.Main->wpcurrent # 'NB.Page1';

 \$edVorlageAuf->WinFocusSet(true);

 RETURN false;

 end;

end

end;

// Nummernvergabe

```

// Satz zurückspeichern & protokollieren

TRANSON;

if (Mode=c_ModeEdit) then begin

    Erx # RekReplace(gFile,_recUnlock,'MAN');

    if (Erx<>_rOk) then begin

        TRANSBRK;

        Msg(001000+Erx,gTitle,0,0,0);

        RETURN False;

    end;

    PtD_Main:Compare(gFile);

    Lib_Texte:RtfTextSave($Adr.V.RTF, cTxtRtf(Adr.V.LfdNr));

end

else begin

    // Ausführungen umnummerieren

    vNr # Adr.V.lfdNr;

    Adr.V.LfdNr # 32000;

    WHILE (RecLink(106,105,1,_recFirst)=_rOK) do begin

        RecRead(106,1,_RecLock);

        Adr.V.AF.Adressnr  # Adr.Nummer;

        Adr.V.AF.Verpacknr # vNr;

        Erx # RekReplace(106,_recUnlock,'MAN');

        If (Erx<>_Rok) then begin

            TRANSBRK;

            Msg(001000+Erx,gTitle,0,0,0);

            RETURN False;

```

end;

END;

Adr.V.lfdNr # vNr;

Adr.V.Anlage.Datum # Today;

Adr.V.Anlage.Zeit # Now;

Adr.V.Anlage.User # gUserName;

Erx # RekInsert(gFile,0,'MAN');

if (Erx<>_rOk) then begin

TRANSBRK;

Msg(001000+Erx,gTitle,0,0,0);

RETURN False;

end;

Lib_Texte:RtfTextSave(\$Adr.V.RTF, cTxtRtf(Adr.V.LfdNr));

end;

TRANSOFF;

RunAFX('Adr.V.RecSave.Post','');

// Weitermachen mit eingeben?

if (w_NoList = false) and (Mode = c_ModeNew) then begin

if (Msg(000005,",_WinIcoQuestion,_WinDialogYesNo,1)=_WinIdYes) then begin

Reclnit(true);

RETURN false;


```

end

else begin

    RETURN true;

end;

end; // Weitermachen mit eingeben?


RETURN true; // Speichern erfolgreich

end;


//=====

// RecCleanup

//      Aufräumen bei "Cancel"

//=====

sub RecCleanup() : logic
local begin

    vNr : int;

end;

begin

    // Ausführungen löschen

    vNr # Adr.V.lfdNr;

    if (Mode=c_ModeNew) then begin

        Adr.V.LfdNr # 32000;

        WHILE (RecLink(106,105,1,_recFirst)=_rOK) do

```

```

    RekDelete(106,0,'MAN');

end;

Adr.V.LfdNr # vNr;


RunAFX('Adr.V.Cleanup.Post', '');


RETURN true;

end;


//=====

// RecDel

//      Satz soll gelöscht werden

//=====

sub RecDel()

local begin

    Erx    : int;

    vBuf105 : int;

    vName   : alpha;

end;

begin

    // Prüfen, ob diese Verpackung wo anders als Einsatz-VPG dient....

    vBuf105 # RecBufCreate(105);

    vBuf105->Adr.V.EinsatzVPG.Adr # Adr.V.Adressnr;

    vBuf105->Adr.V.EinsatzVPG.Nr # Adr.V.lfdNr;

```

```
Erx # RecRead(vBuf105,5,0);
```

```
if (erx<=_rMultikey) then begin
```

```
Msg(105002,aint(vBuf105->Adr.V.Adressnr)+'/'+aint(vBuf105->adr.V.lfdNr),_WinIcoError, _WinDialogOk
```

```
RecBufDestroy(vBuf105);
```

```
RETURN;
```

```
end;
```

```
RecBufDestroy(vBuf105);
```

```
// Diesen Eintrag wirklich löschen?
```

```
if (Msg(000001,",_WinIcoQuestion,_WinDialogYesNo,2)=_WinIdYes) then begin
```

```
// Ausführungen löschen
```

```
TRANSON;
```

```
WHILE (RecLink(106,105,1,_recFirst)=_rOK) do
```

```
Erx # RekDelete(106,0,'MAN');
```

```
//RekDelete(106,0,'MAN');
```

```
if (Erx=_rLocked) then begin
```

```
TRANSBRK;
```

```
msg(105001,"0,0,0,0);
```

```
RETURN;
```

```
end;
```

```
// Aufpreise löschen
```

```
WHILE (RecLink(104,105,9,_recFirst)<=_rLocked) do begin
```

```
Erx # RekDelete(104);  
  
if (erx<>_rOK) then begin  
  
    TRANSBRK;  
  
    msg(105001,"0,0,0,0);  
  
    RETURN;  
  
end;  
  
END;
```

```
Erx # RekDelete(gFile,0,'MAN');  
  
if (Erx<>_rok) then begin  
  
    TRANSBRK;  
  
    msg(105001,"0,0,0,0);  
  
    RETURN;  
  
end;
```

```
vName # cTxtRtf(Adr.V.LfdNr);  
  
TxtDelete(vName,0);
```

```
if (gZLList->wpDbSelection<>0) then begin  
  
    SelRecDelete(gZLList->wpDbSelection,gFile);  
  
    RecRead(gFile, gZLList->wpDbSelection, 0);  
  
end;
```

```
TRANSOFF;
```

end;

end;

//=====

// EvtFocusInit

// Fokus auf Objekt neu gesetzt

//=====

sub EvtFocusInit (

 aEvt : event; // Ereignis

 aFocusObject : int // vorheriges Objekt

) : logic

begin

if (Adr.V.RtfText1=0) and

((Mode=c_modeedit) or (Mode=c_modenew)) and

(Wininfo(aEvt:Obj, _WinType)=_WinTypeRtfEdit) then begin

 aEvt:Obj->wpReadOnly # n;

// aEvt:Obj->wpColBkgApp # _WinColWindow;

 \$Adr.V.ToolbarRTF -> wpdisabled # false;

 \$Adr.V.ToolbarTXT -> wpdisabled # false;

 \$Adr.V.ToolbarRTF->wpObjLink # aEvt:Obj->WpName;

end

else begin

 if (\$Adr.V.ToolbarRTF->wpdisabled=false) then begin

 \$Adr.V.ToolbarRTF -> winupdate(_Winupdon);

```

$Adr.V.ToolbarRTF -> wpdisabled # true;

$Adr.V.ToolbarTXT -> wpdisabled # true;

end;

if (Wininfo(aEvt:Obj, _WinType)=_WinTypeRtfEdit) then begin

    aEvt:Obj->wpReadOnly # y;

//    aEvt:Obj->wpColBkgApp # _WinCol3DLight;

    end;

end;

/*

ST 2019-07-12 Deaktiviert, damit man Einträge auch leeren kann (Pug HErr Rosenbaum)

$edAdr.V.EinsatzVPG.Adr->wpreadonly # true;

$edAdr.V.EinsatzVPG.Nr->wpreadonly # true;

*/

// Auswahlfelder aktivieren

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

    Lib_GuiCom:AuswahlEnable(aEvt:Obj);

else

    Lib_GuiCom:AuswahlIDisable(aEvt:Obj);

end;

//=====

```

```

// EvtFocusTerm

//      Fokus vom Objekt wegbewegen

//=====

sub EvtFocusTerm (

    aEvt          : event;      // Ereignis

    aFocusObject   : int        // neu zu fokusierendes Objekt

) : logic

local begin

    Erx : int;

end;

begin

    if (aEvt:obj->wpname='edVorlageAuf') then begin

        Lib_Berechnungen:Int2AusAlpha($edVorlageAuf->wpcaption,var Adr.V.VorlageAuf, var Adr.V.VorlageAuf);

//    MQU_Data:Autokorrektur(var "Adr.V.Güte");

    end;

    if ((aEvt:obj->wpname='edAdr.V.RingKgVon') and ($edAdr.V.RingKgVon->wpchanged)) then begin

        if (Adr.V.KgmmVon=0.0) and (Adr.V.Breite<>0.0) then Adr.V.KGmmVon # Rnd(Adr.V.RingKgVon / Adr.V.Breite);

        if (Adr.V.RAD=0.0) then begin

            Erx # RecLink(819,105,2,_recFirst); // Warengruppe holen

            Adr.V.RAD # Lib_Berechnungen:RAD_aus_KgStkBDichteRID(Adr.V.RingKgVon, 1, Adr.V.Breite, Wgr_Von);

        end;

        $edAdr.V.KgmmVon->winupdate();

//    $edAdr.V.RAD->winupdate();

    end;

end;

```

```
if ((aEvt:obj->wpname='edAdr.V.RingKgBis') and ($edAdr.V.RingKgBis->wpchanged)) then begin
    if (Adr.V.KgmmBis=0.0) and (Adr.V.Breite<>0.0) then Adr.V.KGmmBis # Rnd(Adr.V.RingKgBis / Adr.V.Breite)
    if (Adr.V.RADmax=0.0) then begin
        Erx # RecLink(819,105,2,_recFirst); // Warengruppe holen
        Adr.V.RADmax # Lib_Berechnungen:RAD_aus_KgStkBDichteRID(Adr.V.RingKgBis, 1, Adr.V.Breite, Wgr_Von)
    end;
    $edAdr.V.KgmmBis->winupdate();
//    $edAdr.V.RADMax->winupdate();
end;
```

```
if ((aEvt:obj->wpname='edAdr.V.KgmmVon') and ($edAdr.V.KgmmVon->wpchanged)) then begin
    if (Adr.V.RingkgVon=0.0) and (Adr.V.Breite<>0.0) then Adr.V.RingKgVon # Rnd(Adr.V.kgmmVon * Adr.V.Breite)
    if (Adr.V.RAD=0.0) then begin
        Erx # RecLink(819,105,2,_recFirst); // Warengruppe holen
        Adr.V.RAD # Lib_Berechnungen:RAD_aus_KgStkBDichteRID(Adr.V.RingKgVon, 1, Adr.V.Breite, Wgr_Von)
    end;
//    $edAdr.V.RAD->winupdate();
    $edAdr.V.RingKgVon->winupdate();
end;
```

```
if ((aEvt:obj->wpname='edAdr.V.KgmmBis') and ($edAdr.V.KgmmBis->wpchanged)) then begin
    if (Adr.V.Ringkgbis=0.0) and (Adr.V.Breite<>0.0) then Adr.V.RingKgBis # Rnd(Adr.V.kgmmbis * Adr.V.Breite)
    if (Adr.V.RADmax=0.0) then begin
        Erx # RecLink(819,105,2,_recFirst); // Warengruppe holen
        Adr.V.RADmax # Lib_Berechnungen:RAD_aus_KgStkBDichteRID(Adr.V.RingKgBis, 1, Adr.V.Breite, Wgr_Von)
    end;
end;
```



```

end;

// $edAdr.V.RADMax->winupdate();

$edAdr.V.RingKgBis->winupdate();

end;

if (Set.Adr.RgGewKgmmYN=false) then begin
    if ((aEvt:obj->wpname='edAdr.V.RAD') and ($edAdr.V.RAD->wpchanged) and (Adr.V.RAD<>0.0)) then begin
        if (Adr.V.RingKgVon=0.0) then
            Adr.V.RingKgVon # Lib_Berechnungen:Kg_aus_StkBDichteRIDRAD(1, Adr.V.Breite, Wgr_Data:GetDi
        if (Adr.V.KgmmVon=0.0) and (Adr.V.Breite<>0.0) then Adr.V.KGmmVon # Rnd(Adr.V.RingKgVon / Adr.V
        $edAdr.V.KgmmVon->winupdate();
        $edAdr.V.RingKgVon->winupdate();
    end;

    if ((aEvt:obj->wpname='edAdr.V.RADmax') and ($edAdr.V.RADmax->wpchanged) and (Adr.V.RADmax<>0.0)) then begin
        if (Adr.V.RingKgBis=0.0) then
            Adr.V.RingKgBis # Lib_Berechnungen:Kg_aus_StkBDichteRIDRAD(1, Adr.V.Breite, Wgr_Data:GetDi
        if (Adr.V.KgmmBis=0.0) and (Adr.V.Breite<>0.0) then Adr.V.KGmmBis # Rnd(Adr.V.RingKgBis / Adr.V
        $edAdr.V.KgmmBis->winupdate();
        $edAdr.V.RingKgBis->winupdate();
    end;
end;

// logische Prüfung von Verknüpfungen

RefreshIfm(aEvt:Obj->wpName);

```

```
RETURN true;
```

```
end;
```

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

```
// Auswahl
```

```
//      Auswahlliste öffnen
```

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

```
sub Auswahl(
```

```
    aBereich : alpha;
```

```
)
```

```
local begin
```

```
    Erx      : int;
```

```
    vA       : alpha;
```

```
    vFilter  : int;
```

```
    vQ       : alpha(4000);
```

```
    vQ2      : alpha(4000);
```

```
    vNr      : int;
```

```
    vHdl     : int;
```

```
    vTmp     : int;
```

```
    vSel     : int;
```

```
    vSelName : alpha;
```

```
end;
```

```
begin
```

case aBereich of

'RtfText1' : begin

gMDI # Lib_GuiCom:AddChildWindow(gMDI,'Adr.V.Verwaltung',here+':AusRtfText1');

VarInstance(WindowBonus, CnvIA(gMDI->wpCustom));

vQ # 'Adr.V.Adressnr = ' + aint(Adr.Nummer);

Lib_Sel:QRecList(0, vQ);

Lib_GuiCom:RunChildWindow(gMDI);

end;

'Unterlage' : begin

RecBufClear(838); // ZIELBUFFER LEEREN

gMDI # Lib_GuiCom:AddChildWindow(gMDI,'ULa.Verwaltung',here+':AusUnterlage');

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

vQ # ' ULa.Typ=0 OR ULa.Typ=1';

Lib_Sel:QRecList(0, vQ);

Lib_GuiCom:RunChildWindow(gMDI);

end;

'Umverpackung' : begin

RecBufClear(838); // ZIELBUFFER LEEREN

gMDI # Lib_GuiCom:AddChildWindow(gMDI,'ULa.Verwaltung',here+':AusUmverpackung');

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

vQ # ' ULa.Typ=0 OR ULa.Typ=3';

```
Lib_Sel:QRecList(0, vQ);
```

```
Lib_GuiCom:RunChildWindow(gMDI);
```

```
end;
```

```
'Zwischenlage' : begin
```

```
RecBufClear(838);      // ZIELBUFFER LEEREN
```

```
gMDI # Lib_GuiCom:AddChildWindow( gMDI,'ULa.Verwaltung',here+':AusZwischenlage');
```

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

```
vQ # ' ULa.Typ=0 OR ULa.Typ=2';
```

```
Lib_Sel:QRecList(0, vQ);
```

```
Lib_GuiCom:RunChildWindow(gMDI);
```

```
end;
```

```
'VorlageAuf' : begin
```

```
if (Adr.V.VerkaufYN) then begin
```

```
RecBufClear(401);      // ZIELBUFFER LEEREN
```

```
gMDI # Lib_GuiCom:AddChildWindow(gMDI,'Auf.P.Verwaltung',here+':AusVorlageAuf');
```

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

```
vQ # 'LinkCount(Kopf) > 0 ';
```

```
Lib_Sel:QAlpha(var vQ2, 'Auf.Vorgangstyp', '=', c_VorlageAuf);
```

```
vHdl # SelCreate(401, gZLList->wpdbkeyno);
```

```
vHdl->SelAddLink("",400, 401, 3, 'Kopf');
```

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

```

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

Erx # vHdl->SelDefQuery('Kopf', vQ2);

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;

//   vHdl # gZLList;

//   Lib_Sel:QRecList(vHdl, auf.p.auf 'BAG.VorlageYN=true');

Lib_GuiCom:RunChildWindow(gMDI);

RETURN;

end;

if (Adr.V.EinkaufYN) then begin

    RecBufClear(501);      // ZIELBUFFER LEEREN

    gMDI # Lib_GuiCom:AddChildWindow(gMDI,'Ein.P.Verwaltung',here+':AusVorlageEin');

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

    vQ # 'LinkCount(Kopf) > 0 ';

    Lib_Sel:QAlpha(var vQ2, 'Ein.Vorgangstyp', '=', c_VorlageAuf);

    vHdl # SelCreate(501, gZLList->wpdbkeyno);

    vHdl->SelAddLink("",500, 501, 3, 'Kopf');

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

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

    Erx # vHdl->SelDefQuery('Kopf', vQ2);

    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;


//      vHdl # gZLList;

//      Lib_Sel:QRecList(vHdl, auf.p.auf 'BAG.VorlageYN=true');

Lib_GuiCom:RunChildWindow(gMDI);

RETURN;

end;

end;


'Verwiegungsart' : begin

RecBufClear(818);      // ZIELBUFFER LEEREN

gMDI # Lib_GuiCom:AddChildWindow(gMDI,'VWa.Verwaltung',here+':AusVerwiegungsart');

Lib_GuiCom:RunChildWindow(gMDI);

end;


'Etikettentyp' : begin

RecBufClear(840);      // ZIELBUFFER LEEREN

gMDI # Lib_GuiCom:AddChildWindow( gMDI,'Eti.Verwaltung',here+':AusEtikettentyp');

Lib_GuiCom:RunChildWindow(gMDI);

end;

```

end;

end;

//=====

// AusRtfText1

//

//=====

sub AusRtfText1()

local begin

 vBuf105 : int;

 vTmp : int;

end;

begin

 if (gSelected<>0) then begin

 vBuf105 # RecBufCreate(105);

 RecRead(vBuf105,0,_RecId,gSelected);

 gSelected # 0;

 // Feldübernahme

 Adr.V.RtfText1 # vBuf105->Adr.v.lfdNr;

 RecBufDestroy(vBuf105);

 vTmp # WinFocusget(); // LastFocus-Feld refreshen

 if (vTmp<>0) then vTmp->Winupdate(_WinUpdFld2Obj);

 end;

```

Adr.Nummer # cnvia($lb.Kunde1->wpcustom);

RecRead(100,1,0);


// Focus setzen:

$edAdr.V.RtfText1->Winfocusset(false);

RefreshIfm('edAdr.V.RtfText1',y);

end;


//=====

// AusUnterlage

//

//=====

sub AusUnterlage()

local begin

    vTmp : int;

end;

begin

    if (gSelected<>0) then begin

        RecRead(838,0,_RecId,gSelected);

        gSelected # 0;

        // Feldübernahme

        Adr.V.Unterlage # ULa.Bezeichnung;

        Adr.V.StapelhAbzug # "ULa.Höhenabzug";

        vTmp # WinFocusget(); // LastFocus-Feld refreshen

        if (vTmp<>0) then vTmp->Winupdate(_WinUpdFld2Obj);

```



```

    $edAdr.V.StapelhAbzug->winupdate(_WinUpdFld2Obj);

end;

// Focus auf Editfeld setzen:

$edAdr.V.Unterlage->Winfocusset(false);

end;


//=====

// AusUmverpackung

//

//=====

sub AusUmverpackung()

local begin

    vTmp : int;

end;

begin

    if (gSelected<>0) then begin

        RecRead(838,0,_RecId,gSelected);

        gSelected # 0;

        // Feldübernahme

        Adr.V.Umverpackung # ULa.Bezeichnung;

        vTmp # WinFocusget(); // LastFocus-Feld refreshen

        if (vTmp<>0) then vTmp->Winupdate(_WinUpdFld2Obj);

    end;

    // Focus auf Editfeld setzen:

    $edAdr.V.Umverpackung->Winfocusset(false);

```

end;

//=====

// AusZwischenlage

//

//=====

sub AusZwischenlage()

local begin

 vTmp : int;

end;

begin

 if (gSelected<>0) then begin

 RecRead(838,0,_RecId,gSelected);

 gSelected # 0;

 // Feldübernahme

 Adr.V.Zwischenlage # ULa.Bezeichnung;

 vTmp # WinFocusget(); // LastFocus-Feld refreshen

 if (vTmp<>0) then vTmp->Winupdate(_WinUpdFld2Obj);

 end;

 // Focus auf Editfeld setzen:

 \$edAdr.V.Zwischenlage->Winfocusset(false);

end;

//=====

```
// AusVorlageAuf
```

```
//
```

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

```
sub AusVorlageAuf()
```

```
local begin
```

```
    vTmp : int;
```

```
end;
```

```
begin
```

```
    if (gSelected<>0) then begin
```

```
        RecRead(401,0,_RecId,gSelected);
```

```
        gSelected # 0;
```

```
        // Feldübernahme
```

```
        Adr.V.VorlageAuf # Auf.P.Nummer;
```

```
        Adr.V.VorlageAufPos # Auf.P.Position;
```

```
        vTmp # WinFocusget(); // LastFocus-Feld refreshen
```

```
        if (vTmp<>0) then vTmp->Winupdate(_WinUpdFld2Obj);
```

```
        RefreshIfm();
```

```
    end;
```

```
    // Focus auf Editfeld setzen:
```

```
    $edVorlageAuf->Winfocusset(false);
```

```
end;
```

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

```
// AusVorlageEin
```

```
//
```

```
//=====

sub AusVorlageEin()

local begin

    vTmp : int;

end;

begin

    if (gSelected<>0) then begin

        RecRead(501,0,_RecId,gSelected);

        gSelected # 0;

        // Feldübernahme

        Adr.V.VorlageAuf # Ein.P.Nummer;

        Adr.V.VorlageAufPos # Ein.P.Position;

        vTmp # WinFocusget(); // LastFocus-Feld refreshen

        if (vTmp<>0) then vTmp->Winupdate(_WinUpdFld2Obj);

        RefreshIfm();

    end;

    // Focus auf Editfeld setzen:

    $edVorlageAuf->Winfocusset(false);

end;
```

```
//=====

// AusVerwiegungsart

//

//=====

sub AusVerwiegungsart()
```

```

local begin

    vTmp : int;

end;

begin

    if (gSelected<>0) then begin

        RecRead(818,0,_RecId,gSelected);

        gSelected # 0;

        // Feldübernahme

        Adr.V.Verwiegungsart # VWa.Nummer;

        vTmp # WinFocusget(); // LastFocus-Feld refreshen

        if (vTmp<>0) then vTmp->Winupdate(_WinUpdFld2Obj);

    end;

    // Focus auf Editfeld setzen:

    $edAdr.V.Verwiegungsart->Winfocusset(false);

end;

```

```

//=====

```

```

// AusEtikettentyp

```

```

//

```

```

//=====

```

```

sub AusEtikettentyp()

```

```

local begin

```

```

    vTmp : int;

```

```

end;

```

```

begin

```

```

if (gSelected<>0) then begin

  RecRead(840,0,_RecId,gSelected);

  gSelected # 0;

  // Feldübernahme

  Adr.V.Etikettentyp # Eti.Nummer;

  vTmp # WinFocusget(); // LastFocus-Feld refreshen

  if (vTmp<>0) then vTmp->Winupdate(_WinUpdFld2Obj);

end;

// Focus auf Editfeld setzen:

$edAdr.V.Etikettentyp->Winfocusset(false);

end;

//=====

// RefreshMode

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

//=====

sub RefreshMode(opt aNoRefresh : logic);

local begin

  vHdl : int;

end

begin

  gMenu # gFrmMain->WinInfo(_WinMenu);

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

```

if (vHdl <> 0) then

 vHdl->wpDisabled # (Mode<>c_ModeList);

// Button & Menüs sperren

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

if (vHdl <> 0) then

 vHdl->wpDisabled # (vHdl->wpDisabled) or (w_context<>") or (Rechte[Rgt_Adr_V_Anlegen]=n);

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

if (vHdl <> 0) then

 vHdl->wpDisabled # (vHdl->wpDisabled) or (w_context<>") or (Rechte[Rgt_Adr_V_Anlegen]=n);

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

if (vHdl <> 0) then

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

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

if (vHdl <> 0) then

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

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

if (vHdl <> 0) then

 vHdl->wpDisabled # (Mode<>c_ModeList) or (w_Auswahlmode) or (Rechte[Rgt_Adr_V_Loeschen]=n);

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

if (vHdl <> 0) then

 vHdl->wpDisabled # (Mode<>c_ModeList) or (w_Auswahlmode) or (Rechte[Rgt_Adr_V_Loeschen]=n);

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

```

if (vHdl <> 0) then

    vHdl->wpDisabled # (Mode<>c_ModeList) and (Mode<>c_ModeView);

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

if (vHdl <> 0) then

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

        (Rechte[Rgt_Adr_V_Anlegen]=n);

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

if (vHdl <> 0) then

    vHdl->wpDisabled # Rechte[Rgt_Adr_Excel_Export]=false;

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

if (vHdl <> 0) then

    vHdl->wpDisabled # Rechte[Rgt_Adr_Excel_Import]=false;

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;

    vTmp : int;

end;

begin

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

    case (aMenuItem->wpName) of

        'Mnu.Aufpreise' : begin

            vHdl # winsearch(gMDI, 'NB.Main');

            //    vHdl->wpcustom # cnvai(Adr.V.Adressnr,_FmtNumNoGroup,0,5)+CnvAl(Winfocusget(),_FmtNumNoGroup);

            RecBufClear(104);

            // MUSTER

            gMDI # Lib_GuiCom:AddChildWindow(gMDI,'Adr.V.Z.Verwaltung','',y);

            Lib_GuiCom:RunChildWindow(gMDI);

        end;

        'Mnu.Copy' : begin

            w_AppendNr # RecInfo(gFile, _recId);

            App_Main:Action(c_ModeNew);

            RETURN true;

        end;
    end;
end;

```

'Mnu.Protokoll' : begin

PtD_Main:View(gFile,Adr.V.Anlage.Datum, Adr.V.Anlage.Zeit, Adr.V.Anlage.User);

end;

'Mnu.CUS.Felder' : begin

CUS_Main:Start(gFile, RecInfo(gFile, _recID));

end;

'Mnu.Druck.Stammblatt' : begin

lib_Dokumente:PrintForm(105,'Stammdatenblatt',n);

end;

// NEU: Serienmarkierung 2011-07-04 TM

'Mnu.Mark.Sel' : begin

Adr_V_Mark_Sel(); // Aufruf Selektionsdialog und -durchführung

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.VorlageAuf'   : Auswahl('VorlageAuf');

        'bt.RtfText1'     : Auswahl('RtfText1');

        'bt.Zwischenlage' : Auswahl('Zwischenlage');

        'bt.Unterlage'    : Auswahl('Unterlage');

        'bt.Umverpackung' : Auswahl('Umverpackung');

        'bt.Verwiegungsart' : Auswahl('Verwiegungsart');

        'bt.Etikettentyp' : Auswahl('Etikettentyp');

    end;

end;

```

```

//=====

// EvtChanged

//

//=====

sub EvtChanged(

    aEvt          : event;    // Ereignis

): logic

local begin

    vName  : alpha;

    vTxtHdl : int;

end;

begin

    if (aEvt:Obj->wpchanged=false) then RETURN true;

    if (aEvt:Obj->wpname='cbAdr.V.EinkaufYN') and (Adr.V.EinkaufYN) then begin

        Adr.V.VerkaufYN # n;

        $cbAdr.V.VerkaufYN->winupdate(_WinUpdFld2Obj);

        $edVorlageAuf->wpcaption # "";

    end;

    if (aEvt:Obj->wpname='cbAdr.V.VerkaufYN') and (Adr.V.VerkaufYN) then begin

        Adr.V.EinkaufYN # n;

        $cbAdr.V.EinkaufYN->winupdate(_WinUpdFld2Obj);

        $edVorlageAuf->wpcaption # "";

    end;

end;

```

```
if (aEvt:Obj->wpname='cbAdr.V.StehendYN') and (Adr.V.StehendYN) then begin
```

```
    Adr.V.LiegendYN # n;
```

```
    $cbAdr.V.LiegendYN->winupdate(_WinUpdFld2Obj);
```

```
end;
```

```
if (aEvt:Obj->wpname='cbAdr.V.LiegendYN') and (Adr.V.LiegendYN) then begin
```

```
    Adr.V.StehendYN # n;
```

```
    $cbAdr.V.StehendYN->winupdate(_WinUpdFld2Obj);
```

```
end;
```

```
end;
```

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

```
// EvtPageSelect
```

```
//      Seitenauswahl von Notebooks
```

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

```
sub EvtPageSelect(
```

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

```
    aPage         : int;
```

```
    aSelecting    : logic;
```

```
) : logic
```

```
begin
```

```
    RETURN true;
```

```
end;
```

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

```
// EvtLstDataInit
```

```
//
```

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

```
sub EvtLstDataInit(
```

```
    aEvt    : Event;
```

```
    aRecId  : int;
```

```
    Opt aMark : logic;
```

```
);
```

```
local begin
```

```
    Erx    : int;
```

```
    vBuf105 : int;
```

```
end;
```

```
begin
```

```
    if (w_context<>") then RecLink(100,105,10,_recFirst); // Adresse holen
```

```
    if (aMark) then begin
```

```
        if (RunAFX('Adr.V.EvtLstDataInit','y')<0) then RETURN;
```

```
    end else begin
```

```
        if (RunAFX('Adr.V.EvtLstDataInit','n')<0) then RETURN;
```

```
    end;
```

```
Gv.Alpha.01 # ";
```

```
if (Adr.V.EinsatzVPG.Adr<>0) and (Adr.V.EinsatzVPG.Nr<>0) then begin
```

```
    vBuf105 # RecBufCreate(105);
```

```
    vBuf105->Adr.v.Adressnr # Adr.V.EinsatzVPG.Adr;
```

```

vBuf105->Adr.V.lfdNr  # Adr.V.EinsatzVPG.Nr;

Erx # RecRead(vBuf105,1,0);

if (erx<=_rMultikey) then begin

    GV.Alpha.01 # anum(vBuf105->Adr.V.Dicke,Set.Stellen.Dicke);

    if (vBuf105->Adr.V.Breite<>0.0) then

        GV.Alpha.01 # Gv.Alpha.01 + ' x '+anum(vBuf105->Adr.V.Breite,Set.Stellen.Breite);

    if (vBuf105->"Adr.V.Länge"<>0.0) then

        GV.Alpha.01 # Gv.Alpha.01 + ' x '+anum(vBuf105->"Adr.V.Länge","Set.Stellen.Länge");

end;

RecBufDestroy(vBuf105);

end;

end;

//=====

// EvtLstSelect

//      Zeilenauswahl von RecList/DataList

//=====

sub EvtLstSelect(

    aEvt          : event;    // Ereignis

    aRecID        : int;

) : logic

begin

    RecRead(gFile,0,_recid,aRecID);

    if (w_context<>") then RecLink(100,105,10,_recFirst);  // Adresse holen

end;

```

```

//=====

// EvtClose

//      Schliessen eines Fensters

//=====

sub EvtClose(

    aEvt          : event;    // Ereignis

): logic

local begin

    vTxtHdl : int;

end;

begin

    vTxtHdl # $Adr.V.RTF->wpdbTextBuf;

    if (vTxtHdl<>0) then

        TextClose(vTxtHdl);

    RETURN true;

end;

//=====

// EvtPosChanged

//

//=====

```



```
sub EvtPosChanged ( aEvt : event; aRect : rect; aClientSize : point; aFlags : int ) : logic
```

```
local begin
```

```
    vRect : rect;
```

```
    vHdl : int;
```

```
end
```

```
begin
```

```
    if (gZLList=0) then RETURN true;    // WORKAROUND VogelBauer
```

```
    // WORKAROUND
```

```
    if (gMDI->wpname<>w_Name) then RETURN false;
```

```
    // Quickbar
```

```
    vHdl # Winsearch(gMDI,'gs.Main');
```

```
    if (vHdl<>0) then begin
```

```
        vRect      # vHdl->wpArea;
```

```
        vRect:right # aRect:right-aRect:left+2;
```

```
        vRect:bottom # aRect:bottom-aRect:Top+5;
```

```
        vHdl->wparea # vRect;
```

```
    end;
```

```
    RETURN true;
```

```
end;
```

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

```
// EvtTimer
```

```
//
```

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

```
sub EvtTimer(
```

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

```
    aTimerId      : int;
```

```
): logic
```

```
local begin
```

```
    vParent : int;
```

```
    vA      : alpha;
```

```
    vMode : alpha;
```

```
end;
```

```
begin
```

```
    App_Main:EvtTimer(aEvt,aTimerId);
```

```
    RETURN true;
```

```
end;
```

```
sub JumpTo(
```

```
    aName : alpha;
```

```
    aBuf  : int);
```

```
begin
```

```
if ((aName =^ 'edAdr.V.Verwiegungsart') AND (aBuf->Adr.V.Verwiegungsart<>0)) then begin
```

```
    RekLink(818,105,4,0); // Verweigerungsart holen
```

```
    Lib_Guicom2:JumpToWindow('VWa.Verwaltung');
```

```
    RETURN;
```

```
end;
```

```
if ((aName =^ 'edAdr.V.Etikettentyp') AND (aBuf->Adr.V.Etikettentyp<>0)) then begin
```

```
    RekLink(840,105,3,0); // Eetikettentyp holen
```

```
    Lib_Guicom2:JumpToWindow('Eti.Verwaltung');
```

```
    RETURN;
```

```
end;
```

```
if ((aName =^ 'edVorlageAuf') AND (aBuf<>0)) then begin //Kein Database ist gegeben
```

```
    RekLink(501,105,11,1); // Vorlageauftrag holen
```

```
    Lib_Guicom2:JumpToWindow('Auf.P.Verwaltung');
```

```
    RETURN;
```

```
end;
```

```
if ((aName =^ 'edAdr.V.Zwischenlage') AND (aBuf->Adr.V.Zwischenlage<>")) then begin
```

```
    todo('Zwischenlage')
```

```
    // RekLink(840,10,1,0); // Zwischenlage holen
```

```
    Lib_Guicom2:JumpToWindow('ULa.Verwaltung');
```

```
    RETURN;
```

```
end;
```

```
if ((aName =^ 'edAdr.V.Unterlage') AND (aBuf->Adr.V.Unterlage<>")) then begin
```

```
    todo('Unterlage')
```

```
    // RekLink(840,10,1,0); // Unterlage holen
```

```
    Lib_Guicom2:JumpToWindow('ULa.Verwaltung');
```

```
    RETURN;
```

```
end;
```

```
if ((aName =^ 'edAdr.V.Umverpackung') AND (aBuf->Adr.V.Umverpackung<>'')) then begin
  todo('Umverpackung')
  // RekLink(840,10,1,0); // Unterlage holen
  Lib_Guicom2:JumpToWindow('ULa.Verwaltung');
  RETURN;
end;
```

```
if ((aName =^ 'edAdr.V.RtfText1') AND (aBuf->Adr.V.RtfText1<>0)) then begin
  todo('Umverpackung')
  // RekLink(840,10,1,0); // Unterlage holen
  Lib_Guicom2:JumpToWindow('');
  RETURN;
end;
```

```
end;
```

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