

@A+

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

//

// Prozedur Dlg_Standard

// OHNE E_R_G

// Info

//

//

// 05.02.2003 AI Erstellung der Prozedur

// 15.09.2009 MS Neubewertung erweitert Auswahl zwischen Basis-EK und Eff-EK

// 19.10.2009 MS Windialogbox öffnet sich jetzt immer mit _WinDialogCenter

// 09.11.2010 MS AnzahlVonBis hinzugefuegt

// 18.11.2010 PW Dynamische Auswahl

// 03.08.2011 ST Controlling: Auftragsarten hinzugefügt

// 28.03.2012 AI BUG: Standard + Standard_Small hatten falschen Parent

// 16.07.2014 AH "Mat_Bestand" mit aFixDatum

// 31.07.2014 ST Prüfung auf Abschlussdatum bei Materialbestandänderung hinzugefügt Projekt 1326/39

// 06.08.2014 AH NEU: "ToolTip"

// 20.04.2016 ST Neu: "Dlg_Barcode"

// 03.06.2016 AH Neu: Dlg.Matz: Pflichtfelder Netto-/Bruttogewicht

// 05.04.2022 AH ERX

// 2023-01-26 MR Edit Bestandsänderugn berücksichtigt jetzt auch die Menge 2465/17

//

//

// Subprozeduren

// SUB EvtInit(aEvt : event) : logic

```

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

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

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

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

// SUB EvtClicked(aEvt : event) : logic

// SUB EvtChanged(aEvt : event) : logic;

// SUB EvtClose(aEvt : event) : logic

// SUB EvtKeyItem(aEvt : event; aKey : int; aID : int) : logic;

//

//

// SUB KWJahr(aFrage : alpha; varaKW : word; varaJahr : word) : logic;

// SUB Standard(aFrage : alpha; var aText : alpha; opt aPW : logic; opt aMaxChars : int) : logic;

// SUB Standard_Small(aFrage : alpha; varaText : alpha; opt aEinReturn : logic; opt aMaxChars : int) : logic;

// SUB DatumVonBis(aFrage : alpha; varaVon : date; varaBis : date; optaVorVon : date; optaVorBis : date) : logic;

// SUB Datum(aFrage : alpha; varaDat : date; optaVor : date) : logic;

// SUB Anzahl(aFrage : alpha; varaZahl : int; optaVorgabe : int; optaXPos : int; optaYPos : int) : logic;

// SUB AnzahlVonBis(aFrage : alpha; var aZahl1 : int; var aZahl2 : int; opt aVorgabe1 : int; opt aVorgabe2 : int) : logic;

// SUB Zeit(aFrage : alpha; varaZeit : Time; optaVorgabe : Time) : logic;

// SUB Menge(aFrage : alpha; varaMenge : float; optaVorgabe : float) : logic;

// SUB InfoBetrieb(aTitel : alpha; aText : alpha; opt aNegativ : logic;) : logic;

// SUB MATZ(aTyp : alpha; var amitVersand : var aBehalten : logic; logic; var aStk : int; var aNetto : float; var aBrutto : float) : logic;

// SUB PosErfassung() : int;

// SUB Mat_Bestand(var aStk : int; var aGew: float; var aPreis : float; var aGrund :alpha; var aDatum; aErfassung : date) : logic;

// SUB Mat_Neubewertung(var aPreis : float; var aAbwertEff : logic; var aNurAb : logic; var aGrund :alpha; var aDatum; aErfassung : date) : logic;

// sub Auswahl ( aCteLst : handle; opt aIndex : int; opt aTitle : alpha ) : int

// sub Auswahl_EvtMouseItem ( aEvt : event; aButton : int; aHitTest : int; altem : int; ald : int ) : logic

```

```

//  sub Auswahl_EvtKeyItem ( aEvt : event; aKey : int; ald : int ): logic
//  SUB Tooltip(aFrage : alpha);
//
//=====

@I:Def_Global

LOCAL begin

  vDialog : alpha;

  vNoMDI  : logic;

  vFrage  : alpha(128);

  vText   : alpha(128);

  vKW     : word;

  vJahr   : word;


  vDatum1 : date;

  vDatum2 : date;


  vZahl1  : int;

  vZahl2  : int;

  vZahl3  : int;

  vMenge  : float;


  vZeit   : Time;

end;

```

```

//=====

// EvtInit

//

//=====

Sub EvtInit(
    aEvt : event;
) : logic
local begin
    Erx : int;
    vA : alpha(400);
    vP : int;
    vX : int;
    vY : int;
    vOK : logic;
end;
begin
    WinsearchPath(aEvT:obj); // 27.08.2018 AH

    // alle über übersetzen
    Lib_GuiCom:TranslateObject( aEvt:Obj );

    if (vNoMDI=false) then begin
        vX # gFrmMain->wpAreaLeft;
        vY # 80 + gFrmMain->wpAreaTop;
    end;
end;

```

```

vP # gMDI;

if (vP<>0) and (HdlInfo(vP,_hdlExists)>0) then begin

    if (vP->wpname=Lib_GuiCom:GetAlternativeName('Mdi.Workbench')) or (vP->wpname=Lib_GuiCom:G

        vP # gFrmMain;

        vX # 0;

        vY # 0;

    end;

end;

if (vP<>0) and (HdlInfo(vP,_hdlExists)>0) then

    vX # vX + vP->wpAreaLeft + ((vP->wpAreaRight - vP->wpAreaLeft)/2);

    if (vP<>0) and (HdlInfo(vP,_hdlExists)>0) then

        vY # vY + vP->wpAreaTop + ((vP->wpAreaBottom - vP->wpAreaTop)/2);

    vX # vX - ((aEvt:Obj->wparearight - aEvt:Obj->wparealeft)/2);

    vY # vY - ((aEvt:Obj->wpareaBottom - aEvt:Obj->wpareaTop)/2);

    Lib_GuiCOM:ObjSetPos(aEvt:obj, vX,vY);

end;


//debug(vDialog);

//todo(vDialog)


case (vDialog) of

'Mat_Bestand' : begin

    $ed.NettoGewicht->wpDecimals # Set.Stellen.Gewicht;

    $ed.BruttoGewicht->wpDecimals # Set.Stellen.Gewicht;

end;

```

```

'Matz' : begin

$ed.Menge->wpDecimals # Set.Stellen.Menge;

$ed.Netto->wpDecimals # Set.Stellen.Gewicht;

$ed.Brutto->wpDecimals # Set.Stellen.Gewicht;

$Lb.Kommission->wpcaption # AInt(Auf.P.nummer)+'/'+AInt(Auf.P.Position);

$Lb.KundenSW->wpcaption # Auf.P.KundenSW

vA # "Auf.P.Güte";

vA # vA + ' ' +ANum(Auf.P.Dicke,Set.Stellen.Dicke);

vA # vA + ' x ' +ANum(Auf.P.Breite,Set.Stellen.Breite);

if ("Auf.P.Länge"<>0.0) then vA # vA + ' x ' +ANum("Auf.P.Länge","Set.Stellen.Länge");

vA # vA + ' ' ;

if (Auf.P.AusfOben<>"") then vA # vA + ' O:' +Auf.P.AusfOben;

if (Auf.P.AusfUnten<>"") then vA # vA + ' U:' +Auf.P.AusfUnten;


$lb.Auftragsinfo->wpcaption # vA;

Erx # RecLink(818,401,9,_recFirst); // Verwiegungsart holen

if (Erx<=_rLocked) then

    $lb1.AufVerwiegungsart->wpcaption # VWa.Bezeichnung.L1

else

    $lb1.AufVerwiegungsart->wpcaption # "";

$lb1.AufStueck->wpcaption # AInt(Auf.P.Prd.Rest.Stk);

if (VWa.NettoYN) then begin

    $lb1.AufNetto->wpcaption # ANum(Auf.P.Prd.Rest.Gew,Set.Stellen.Gewicht);

    $lb1.AufBrutto->wpcaption # "";

end

```

else begin

\$lb1.AufNetto->wpcaption # ";

\$lb1.AufBrutto->wpcaption # ANum(Auf.P.Prd.Rest.Gew,Set.Stellen.Gewicht);

end;

vMenge # Lib_Einheiten:WandleMEH(401, Auf.P.Prd.Rest.Stk, Auf.P.Prd.Rest.Gew, Auf.P.Prd.Rest, A

\$lb1.AufMenge->wpcaption # ANum(vMenge,Set.Stellen.Menge);

\$lb.Materialnr->wpcaption # AInt(Mat.Nummer);

vA # "Mat.Güte";

vA # vA + ' ' +ANum(Mat.Dicke,Set.Stellen.Dicke);

vA # vA + ' x ' +ANum(Mat.Breite,Set.Stellen.Breite);

if ("Mat.Länge"<>0.0) then vA # vA + ' x ' +ANum("Mat.Länge","Set.Stellen.Länge");

vA # vA + ' ';

if ("Mat.AusführungOben"<>") then vA # vA + ' O:'+"Mat.AusführungOben";

if ("Mat.AusführungUnten"<>") then vA # vA + ' U:'+"Mat.AusführungUnten";

\$lb.MaterialInfo->wpcaption # vA;

vOK # ("Auf.P.Güte"="Mat.Güte") and (Auf.P.Dicke=Mat.Dicke) and (Auf.P.Breite=Mat.Breite) and ("Au

if (vOK=false) then begin

\$lb.Materialinfo->wpColFg # _WinColwhite;

\$lb.Materialinfo->wpColBkg # _WinColRed;

end;

Exr # RecLink(818,200,10,_recFirst); // Verwieungsart holen

if (Exr<=_rLocked) then

```

    $lb1.UrVerwiegungsart->wpcaption # Vwa.Bezeichnung.L1

else

    $lb1.UrVerwiegungsart->wpcaption # "";

    $lb1.UrStueck->wpcaption # AInt(Mat.Bestand.Stk);

    $lb1.UrNetto->wpcaption # ANum(Mat.Gewicht.Netto,Set.Stellen.Gewicht);

    $lb1.UrBrutto->wpcaption # ANum(Mat.Gewicht.Brutto,Set.Stellen.Gewicht);


vMenge # Lib_Einheiten:WandleMEH(200, Mat.Bestand.Stk, Mat.Bestand.Gew, Mat.Bestand.Gew, 'kg

$lb1.UrMenge->wpcaption    # ANum(vMenge,Set.Stellen.Menge);


$lb1.MEH1->wpcaption # Auf.P.MEH.Preis;

$lb1.MEH2->wpcaption # Auf.P.MEH.Preis;

$lb1.MEH3->wpcaption # Auf.P.MEH.Preis;

end;


'VSB' : begin

    $Lb.Kommission->wpcaption # AInt(Auf.P.nummer)+'/'+AInt(Auf.P.Position);

    $Lb.KundenSW->wpcaption # Auf.P.KundenSW

    vA # "Auf.P.Güte";

    vA # vA + ' ' +ANum(Auf.P.Dicke,Set.Stellen.Dicke);

    vA # vA + ' x ' +ANum(Auf.P.Breite,Set.Stellen.Breite);

    if ("Auf.P.Länge"<>0.0) then vA # vA + ' x ' +ANum("Auf.P.Länge","Set.Stellen.Länge");

    vA # vA + ' ' ;

    if (Auf.P.AusfOben<>") then vA # vA + '  O:' +Auf.P.AusfOben;

    if (Auf.P.AusfUnten<>") then vA # vA + '  U:' +Auf.P.AusfUnten;

```



```
if(Auf.P.Projektnummer <> 0) then // MS 08.04.2011 Prj. 1342/2
```

```
Lib_Strings.Append(var vA, 'Projekt: ' + AInt(Auf.P.Projektnummer), ' ');
```

```
$lb.Auftragsinfo->wpcaption # vA;
```

```
Erx # RecLink(818,401,9,_recFirst); // Verwiegungsart holen
```

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

```
    $lb1.AufVerwiegungsart->wpcaption # VWa.Bezeichnung.L1
```

```
else
```

```
    $lb1.AufVerwiegungsart->wpcaption # ";
```

```
$lb1.AufStueck->wpcaption # AInt(Auf.P.Prd.Rest.Stk);
```

```
if (VWa.NettoYN) then begin
```

```
    $lb1.AufNetto->wpcaption # ANum(Auf.P.Prd.Rest.Gew,Set.Stellen.Gewicht);
```

```
    $lb1.AufBrutto->wpcaption # ";
```

```
end
```

```
else begin
```

```
    $lb1.AufNetto->wpcaption # ";
```

```
    $lb1.AufBrutto->wpcaption # ANum(Auf.P.Prd.Rest.Gew,Set.Stellen.Gewicht);
```

```
end;
```

```
vMenge # Lib_Einheiten:WandleMEH(401, Auf.P.Prd.Rest.Stk, Auf.P.Prd.Rest.Gew, Auf.P.Prd.Rest, A
```

```
$lb1.AufMenge->wpcaption # ANum(vMenge,Set.Stellen.Menge);
```

```
$lb.Materialnr->wpcaption # AInt(Mat.Nummer);
```

```
vA # "Mat.Güte";
```

```
vA # vA + ' ' +ANum(Mat.Dicke,Set.Stellen.Dicke);
```

```
vA # vA + ' x ' +ANum(Mat.Breite,Set.Stellen.Breite);
```

```
if ("Mat.Länge"<>0.0) then vA # vA + ' x '+ANum("Mat.Länge","Set.Stellen.Länge");
```

```
vA # vA + ' ';
```

```
if ("Mat.AusführungOben"<>") then vA # vA + ' O: '+'Mat.AusführungOben";
```

```
if ("Mat.AusführungUnten"<>") then vA # vA + ' U: '+'Mat.AusführungUnten";
```

```
if(Mat.EK.Projektnr <> 0) then // MS 08.04.2011 Prj. 1342/2
```

```
Lib_Strings:Append(var vA, 'Projekt: ' + AInt(Mat.EK.Projektnr), ' ');
```

```
$lb.MaterialInfo->wpcaption # vA;
```

```
vOK # ("Auf.P.Güte"="Mat.Güte") and (Auf.P.Dicke=Mat.Dicke) and (Auf.P.Breite=Mat.Breite) and ("Auf.P.Länge"=Mat.Länge)
```

```
if (vOK=false) then begin
```

```
    $lb.Materialinfo->wpColFg # _WinColwhite;
```

```
    $lb.Materialinfo->wpColBkg # _WinColRed;
```

```
end;
```

```
Erx # RecLink(818,200,10,_recFirst); // Verwieungsart holen
```

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

```
    $lb1.UrVerwiegungsart->wpcaption # Vwa.Bezeichnung.L1
```

```
else
```

```
    $lb1.UrVerwiegungsart->wpcaption # "";
```

```
$lb1.UrStueck->wpcaption # AInt(Mat.Bestand.Stk);
```

```
$lb1.UrNetto->wpcaption # ANum(Mat.Gewicht.Netto,Set.Stellen.Gewicht);
```

```
$lb1.UrBrutto->wpcaption # ANum(Mat.Gewicht.Brutto,Set.Stellen.Gewicht);
```

```
vMenge # Lib_Einheiten:WandleMEH(200, Mat.Bestand.Stk, Mat.Bestand.Gew, Mat.Bestand.Gew, 'kg')
```

```
$lb1.UrMenge->wpcaption # ANum(vMenge,Set.Stellen.Menge);
```

\$lb1.MEH1->wpcaption # Auf.P.MEH.Preis;

\$lb1.MEH2->wpcaption # Auf.P.MEH.Preis;

\$lb1.MEH3->wpcaption # Auf.P.MEH.Preis;

aEvt:obj->wpcaption # Translate('Material VSB setzen');

\$lb.Zuordnung->wpcaption # Translate('VSB setzen');

\$lb1.Lfs.P.Menge->wpvisible # false;

\$Lb1.AufMenge->wpvisible # false;

\$Lb1.MEH3->wpvisible # false;

\$Lb1.UrMenge->wpvisible # false;

\$Lb1.MEH2->wpvisible # false;

\$ed.Menge->wpvisible # false;

\$lb1.MEH1->wpvisible # false;

end;

'DatumVonBis' : begin

\$lbText->wpCaption # vFrage;

\$edDate1->wpCaptiondate # vdatum1;

\$edDate2->wpCaptiondate # vdatum2;

\$edDate1->WinFocusSet(true);

end;

'Datum' : begin

\$lbText->wpCaption # vFrage;

\$edDate1->wpCaptiondate # vdatum1;

\$edDate1->WinFocusSet(true);

end;

'Zeit' : begin

\$lbText->wpCaption # vFrage;

\$edTime1->wpCaptiontime # vZeit;

\$edTime1->WinFocusSet(true);

end;

'Menge' : begin

\$lbText->wpCaption # vFrage;

\$FloatEdit1->wpCaptionfloat # vMenge;

\$FloatEdit1->WinFocusSet(true);

end;

'Anzahl' : begin

\$lbText->wpCaption # vFrage;

\$IntEdit1->wpCaptionInt # vZahl1;

\$IntEdit1->WinFocusSet(true);

end;

'AnzahlVonBis' : begin

\$lbText->wpCaption # vFrage;

\$IntEdit1->wpCaptionInt # vZahl1;

\$IntEdit2->wpCaptionInt # vZahl2;

\$IntEdit1->WinFocusSet(true);

end;

'NrPosPos' : begin

\$lbText->wpCaption # vFrage;

\$IntEdit1->wpCaptionInt # vZahl1;

\$IntEdit2->wpCaptionInt # vZahl2;

```
$IntEdit3->wpCaptionInt # vZahl3;
```

```
$IntEdit1->WinFocusSet(true);
```

```
end;
```

```
'Standard' : begin
```

```
  $lbText->wpCaption # vFrage;
```

```
  $edText->wpcaption # vText;
```

```
  $edText->WinFocusSet(true);
```

```
end;
```

```
'Standard_Small' : begin
```

```
  $lbText->wpCaption # vFrage;
```

```
  $edText->wpcaption # vText;
```

```
  $edText->WinFocusSet(true);
```

```
end;
```

```
'InfoBetrieb' : begin
```

```
  /*
```

```
  $lbText->wpCaption # vFrage;
```

```
  $edText->wpcaption # vText;
```

```
  $edText->WinFocusSet(true);
```

```
  */
```

```
end;
```

```
'KW' : begin
```

```
  $lbText->wpCaption # vFrage;
```

```
  $edKW->wpcaption # AInt(vKW);
```

```
  $edJahr->wpcaption # AInt(vJahr);
```

```
$edKW->WinFocusSet(true);
```

```
end;
```

```
'Con_Generieren' : begin
```

```
Lib_GuiCom:Disable($cb.AlleGueten);
```

```
Lib_GuiCom:Disable($cb.MarkierteGueten);
```

```
Lib_GuiCom:Disable($cb.MarkierteArtikelgruppen);
```

```
Lib_GuiCom:Disable($cb.AlleArtikelgruppen);
```

```
Lib_GuiCom:Disable($cb.MarkierteArtikelnummer);
```

```
Lib_GuiCom:Disable($cb.AlleArtikelnummer);
```

```
$rbArtikel->wpCheckState # _WinStateChkUnchecked;
```

```
$rbMaterial->wpCheckState # _WinStateChkUnchecked;
```

```
$cbErl->wpCheckState # _WinStateChkchecked;
```

```
end;
```

```
'Mat_Neubewertung' : begin
```

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

```
end;
```

```
end;
```

```
end;
```

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

```
// EvtMenuCommand
```

```
//
```

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

```
sub EvtMenuCommand(
```

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

```
    aMenuItem : int     // Auslösender Menüpunkt / Toolbar-Button
```

```
) : logic
```

```
local begin
```

```
    vX : float;
```

```
    vI : int;
```

```
end;
```

```
begin
```

```
if (aMenuItem->wpname='Mnu.Save') then begin
```

```
    aEvt:obj->WinDialogResult(_WinIDOK);
```

```
    aEvt:Obj->WinClose();
```

```
end;
```

```
if (aMenuItem->wpname='Mnu.Ktx.Errechnen') then begin
```

```
    if (aEvt:Obj->wpname='ed.Stueck') then begin
```

```
        vX # $ed.Netto->wpcaptionfloat;
```

```
        if (vDialog='Matz') or (vDialog='VSB') then
```

```
            vI # Lib_Berechnungen:STK_aus_KgDBLWgrArt(vX, Mat.Dicke, Mat.Breite, "Mat.länge", Mat.Wareng
```

```
        if (vI=0) then
```

```
            vI # Lib_Berechnungen:STK_aus_KgDBLWgrArt(vX, Auf.P.Dicke, Auf.P.Breite, "Auf.P.länge", Auf.P.V
```

```

    $ed.Stueck->wpcaptionint # vl;

end;

if (aEvt:Obj->wpname='ed.Netto') then begin

    vl # $ed.Stueck->wpcaptionint;

    if (vDialog='Matz') or (vDialog='VSB') then

        vX # Lib_Berechnungen:KG_aus_StkDBLWgrArt(vl, Mat.Dicke, Mat.Breite, "Mat.länge", Mat.Warengn

    if (vX=0.0) then

        vX # Lib_Berechnungen:KG_aus_StkDBLWgrArt(vl, Auf.P.Dicke, Auf.P.Breite, "Auf.P.länge", Auf.P.W

    $ed.Netto->wpcaptionfloat # vX;

end;

if (aEvt:Obj->wpname='ed.Brutto') then begin

    vl # $ed.Stueck->wpcaptionint;

    if (vDialog='Matz') or (vDialog='VSB') then

        vX # Lib_Berechnungen:KG_aus_StkDBLWgrArt(vl, Mat.Dicke, Mat.Breite, "Mat.länge", Mat.Warengn

    if (vX=0.0) then

        vX # Lib_Berechnungen:KG_aus_StkDBLWgrArt(vl, Auf.P.Dicke, Auf.P.Breite, "Auf.P.länge", Auf.P.W

    $ed.Brutto->wpcaptionfloat # vX;

end;

end;

RETURN true;

end;

//=====

// EvtMenuCommand_MatB

```



```
//
```

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

```
sub EvtMenuCommand_MatB(
```

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

```
    aMenuItem : int     // Auslösender Menüpunkt / Toolbar-Button
```

```
) : logic
```

```
local begin
```

```
    vX : float;
```

```
    vI : int;
```

```
end;
```

```
begin
```

```
    if (aMenuItem->wpname='Mnu.Ktx.Errechnen') then begin
```

```
        if (aEvt:Obj->wpname='ed.Stueck') then begin
```

```
            vX # $ed.NettoGewicht->wpcaptionfloat;
```

```
            vI # Lib_Berechnungen:STK_aus_KgDBLWgrArt(vX, Mat.Dicke, Mat.Breite, "Mat.länge", Mat.Warengru
```

```
            $ed.Stueck->wpcaptionint # vI;
```

```
        end;
```

```
        if (aEvt:Obj->wpname='ed.NettoGewicht') then begin
```

```
            vI # $ed.Stueck->wpcaptionint;
```

```
            vX # Lib_Berechnungen:KG_aus_StkDBLWgrArt(vI, Mat.Dicke, Mat.Breite, "Mat.länge", Mat.Warengru
```

```
            $ed.NettoGewicht->wpcaptionfloat # vX;
```

```
        end;
```

```
        if (aEvt:Obj->wpname='ed.BruttoGewicht') then begin
```

```
            vI # $ed.Stueck->wpcaptionint;
```

```

vX # Lib_Berechnungen:KG_aus_StkDBLWgrArt(vl, Mat.Dicke, Mat.Breite, "Mat.länge", Mat.Warengru
$ed.BruttoGewicht->wpcaptionfloat # vX;

end;

end;

RETURN true;

end;

//=====

// Refrehslfm

//

//=====

sub Refreshlfm

local begin

vM1,vM2 : float;

vObj   : int;

end;

begin

// 03.06.2016 AH:

if (vDialog='Matz') or (vDialog='VSB') then begin

vM1 # cnvfa($Lb1.UrNetto->wpCaption)

vObj # $ed.Netto;

vM2 # vObj->wpCaptionFloat;

```

```

if (vM2=0.0) and (vM1<>0.0) then

    vObj->wpColBkg # _WinColLightYellow;

else

    vObj->wpColBkg # _WinColWindow; // "standard" Hintergrundfarbe

vObj # $ed.Brutto;

vM2 # vObj->wpCaptionFloat;

if (vM2=0.0) and (vM1<>0.0) then

    vObj->wpColBkg # _WinColLightYellow;

else

    vObj->wpColBkg # _WinColWindow; // "standard" Hintergrundfarbe

end;

end;

//=====

// EvtFocusInit

//      Fokus auf Objekt neu gesetzt

//=====

sub EvtFocusInit (

    aEvt          : event;    // Ereignis

    aFocusObject  : int      // vorheriges Objekt

) : logic

local begin

    vTmp  : int;

end;

begin

```

```
RefreshIfm();
```

```
if (aEvt:Obj->wpname='OK') and (vDialog='Standard_Small') then begin
```

```
  if (aEvt:Obj->wpcustom='END') then begin
```

```
    gSelected # 1;
```

```
    vText # $edText->wpCaption;
```

```
    vTmp # aEvt:Obj->WinInfo(_WinFrame);
```

```
    vTmp->winClose();
```

```
  end;
```

```
  RETURN true;
```

```
end;
```

```
if (aEvt:Obj->Wininfo(_WinType)=_WinTypeCheckbox) then
```

```
  aEvt:Obj->wpColBkg # Set.Col.Field.Cursor
```

```
else
```

```
  aEvt:Obj->wpColFocusBkg # Set.Col.Field.Cursor;
```

```
//  aEvt:Obj->wpColBkg # _WinColCyan;
```

```
// else
```

```
//  aEvt:Obj->wpColFocusBkg # ColFocus;
```

```
end;
```

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

```
// EvtFocusTerm
```

```
//      Fokus von Objekt wegnehmen
```

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

```
sub EvtFocusTerm (
```

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

```
    aFocusObject   : int      // nachfolgendes Objekt
```

```
) : logic
```

```
begin
```

```
    aEvt:Obj->wpColBkg # _WinColParent;
```

```
    RETURN true;
```

```
end;
```

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

```
// EvtFocusTermBarcode
```

```
//      Fokus von Objekt wegnehmen
```

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

```
sub EvtFocusTermBarcode (
```

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

```
    aFocusObject   : int      // nachfolgendes Objekt
```

```
) : logic
```

```
local begin
```

```
    vWin : int;
```

```
end;
```

```
begin
```

```
    vWin # aEvt:Obj->WinInfo(_WinFrame);
```

```
vText # aEvt:Obj->wpCaption;  
  
vWin->winClose();  
  
RETURN true;  
  
end;
```

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

```
// EvtClicked
```

```
//
```

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

```
Sub EvtClicked(  
    aEvt : event;  
    ) : logic  
local begin  
    vHdl : int;  
    vWin : int;  
end;  
begin
```

```
    vWin # aEvt:Obj->WinInfo(_WinFrame);
```

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

```
        'Bt.OK' : begin
```

```
            gSelected # 0;
```

```

if ($Bt.OK->wpcustom<>") then begin
    vHdl # vWin->Winsearch($Bt.OK->wpcustom);
    if (vHdl<>0) then begin
        case WinInfo(vHdl,_wintype) of
            _WinTypeDataList : gSelected # vHdl->wpCurrentInt;
            _WinTypeRecList : gSelected # vHdl->wpDbRecId;
        end;
    end;
end;
vWin->winClose();
end;

```

```

'Bt.Abbruch' : begin
    gSelected # 0;
    vWin->winClose();
end;

```

```

'OK' : begin
    RefreshIfm();
    case vDialog of
        'DatumVonBis' : begin
            vDatum1 # $edDate1->wpCaptiondate;
            vDatum2 # $edDate2->wpCaptiondate;
        end;
    end;
end;

```

'Datum' : begin

 vDatum1 # \$edDate1->wpCaptiondate;

end;

'Zeit' : begin

 vZeit # \$edTime1->wpCaptionTime;

end;

'Anzahl' : begin

 vZahl1 # \$IntEdit1->wpCaptionInt;

end;

'AnzahlVonBis' : begin

 vZahl1 # \$IntEdit1->wpCaptionInt;

 vZahl2 # \$IntEdit2->wpCaptionInt;

end;

'NrPosPos' : begin

 vZahl1 # \$IntEdit1->wpCaptionInt;

 vZahl2 # \$IntEdit2->wpCaptionInt;

 vZahl3 # \$IntEdit3->wpCaptionInt;

end;

'Menge' : begin

 vMenge # \$FloatEdit1->wpCaptionFloat;

end;

'Standard' : begin

 vText # \$edText->wpCaption;

end;

'Standard_Small' : begin

 vText # \$edText->wpCaption;


```

end;

'KW' : begin

    vKW # cnvIA($edKW->wpCaption);

    vJahr # cnvIA($edJahr->wpCaption);

end;

end;

end; //case

end;


RETURN true;

end;


//=====

// EvtChanged

//

//=====

sub EvtChanged(

    aEvt          : event;  // Ereignis

) : logic;

begin

    // 25.06.2014

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


    if (aEvt:Obj->wpname='cb.Versand') then begin

        if ($cb.Versand->wpCheckState=_WinStateChkChecked) then begin

```

```

$ed.Stueck->wpcaptionint  # cnvia($ed.Stueck->wpcustom);

$ed.Netto->wpcaptionfloat  # cnvfa($ed.Netto->wpcustom);

$ed.Brutto->wpcaptionfloat  # cnvfa($ed.Brutto->wpcustom);

$ed.Menge->wpcaptionfloat  # cnvfa($ed.Menge->wpcustom);

$ed.Datum->wpcaptionDate  # cnvda($ed.Datum->wpcustom);

Lib_GuiCom:Disable($ed.Stueck);

Lib_GuiCom:Disable($ed.Netto);

Lib_GuiCom:Disable($ed.Brutto);

Lib_GuiCom:Disable($ed.Menge);

Lib_GuiCom:Disable($ed.Datum);


Lib_GuiCom:Enable($ed.DatumVon);

Lib_GuiCom:Enable($ed.DatumBis);

Lib_GuiCom:Enable($ed.Zusatz);

end

else begin

Lib_GuiCom:Enable($ed.Stueck);

Lib_GuiCom:Enable($ed.Netto);

Lib_GuiCom:Enable($ed.Brutto);

Lib_GuiCom:Enable($ed.Menge);

Lib_GuiCom:Enable($ed.Datum);


Lib_GuiCom:Disable($ed.DatumVon);

Lib_GuiCom:Disable($ed.DatumBis);

Lib_GuiCom:Disable($ed.Zusatz);

end;

```

end;

// ST 2010-03-18

if (vDialog = 'Con_Generieren') then begin

if (aEvt:Obj->wpName = 'cbErl') AND (\$cbErl->wpCheckState = _WinStateChkchecked) then begin

\$cbAuf->wpCheckState # _WinStateChkUnchecked;

\$cbBest->wpCheckState # _WinStateChkUnchecked;

\$cbAng->wpCheckState # _WinStateChkUnchecked;

end;

if (aEvt:Obj->wpName = 'cbAuf') AND (\$cbAuf->wpCheckState = _WinStateChkchecked) then begin

\$cbErl->wpCheckState # _WinStateChkUnchecked;

\$cbBest->wpCheckState # _WinStateChkUnchecked;

\$cbAng->wpCheckState # _WinStateChkUnchecked;

end;

if (aEvt:Obj->wpName = 'cbBest') AND (\$cbBest->wpCheckState = _WinStateChkchecked) then begin

\$cbAuf->wpCheckState # _WinStateChkUnchecked;

\$cbErl->wpCheckState # _WinStateChkUnchecked;

\$cbAng->wpCheckState # _WinStateChkUnchecked;

end;

if (aEvt:Obj->wpName = 'cbAng') AND (\$cbAng->wpCheckState = _WinStateChkchecked) then begin

\$cbAuf->wpCheckState # _WinStateChkUnchecked;

\$cbBest->wpCheckState # _WinStateChkUnchecked;

\$cbErl->wpCheckState # _WinStateChkUnchecked;

end;

if (\$cbErl->wpCheckState = _WinStateChkUnchecked) and

(\$cbAuf->wpCheckState = _WinStateChkUnchecked) and

(\$cbBest->wpCheckState = _WinStateChkUnchecked) and

(\$cbAng->wpCheckState = _WinStateChkUnchecked) then aEvt:obj->wpCheckState # _WinStateChkCh

if (aEvt:Obj->wpName = 'cb.AlleKunden') AND (\$cb.MarkierteKunden->wpCheckState = _WinStateChkCh

\$cb.AlleKunden->wpCheckState # _WinStateChkchecked;

\$cb.MarkierteKunden->wpCheckState # _WinStateChkUnchecked;

end;

if (aEvt:Obj->wpName = 'cb.MarkierteKunden') AND (\$cb.AlleKunden->wpCheckState = _WinStateChkCh

\$cb.AlleKunden->wpCheckState # _WinStateChkUnchecked;

\$cb.MarkierteKunden->wpCheckState # _WinStateChkchecked;

end;

if (aEvt:Obj->wpName = 'cb.AlleVertreter') AND (\$cb.MarkierteVertreter->wpCheckState = _WinStateChkCh

\$cb.AlleVertreter->wpCheckState # _WinStateChkchecked;

\$cb.MarkierteVertreter->wpCheckState # _WinStateChkUnchecked;

end;

if (aEvt:Obj->wpName = 'cb.MarkierteVertreter') AND (\$cb.AlleVertreter->wpCheckState = _WinStateChkCh

\$cb.AlleVertreter->wpCheckState # _WinStateChkUnchecked;

\$cb.MarkierteVertreter->wpCheckState # _WinStateChkchecked;

end;

if (aEvt:Obj->wpName = 'cb.AlleAuftragsarten') AND (\$cb.MarkierteAuftragsarten->wpCheckState = _Wi

\$cb.AlleAuftragsarten->wpCheckState # _WinStateChkchecked;

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

```
end;
```

```
if (aEvt:Obj->wpName = 'cb.MarkierteAuftragsarten') AND ($cb.AlleAuftragsarten->wpCheckState = _WinStateChkUnchecked;
```

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

```
$cb.MarkierteAuftragsarten->wpCheckState # _WinStateChkchecked;
```

```
end;
```

```
if (aEvt:Obj->wpName = 'cb.AlleWarengruppen') AND ($cb.MarkierteWarengruppen->wpCheckState = _WinStateChkUnchecked;
```

```
$cb.AlleWarengruppen->wpCheckState # _WinStateChkchecked;
```

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

```
end;
```

```
if (aEvt:Obj->wpName = 'cb.MarkierteWarengruppen') AND ($cb.AlleWarengruppen->wpCheckState = _WinStateChkUnchecked;
```

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

```
$cb.MarkierteWarengruppen->wpCheckState # _WinStateChkchecked;
```

```
end;
```

```
if (aEvt:Obj->wpName = 'cb.AlleArtikelgruppen') AND ($cb.MarkierteArtikelgruppen->wpCheckState = _WinStateChkUnchecked;
```

```
$cb.AlleArtikelgruppen->wpCheckState # _WinStateChkchecked;
```

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

```
// Material ausblenden
```

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

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

```
Lib_GuiCom:Disable($cb.AlleGueten);
```

```

Lib_GuiCom:Disable($cb.MarkierteGueten);

$rbArtikel->wpCheckState # _WinStateChkchecked;

$rbMaterial->wpCheckState # _WinStateChkUnchecked;

end;

if (aEvt:Obj->wpName = 'cb.MarkierteArtikelgruppen') AND ($cb.AlleArtikelgruppen->wpCheckState = _V)

$cb.AlleArtikelgruppen->wpCheckState      # _WinStateChkUnchecked;

$cb.MarkierteArtikelgruppen->wpCheckState # _WinStateChkchecked;

// Material ausblenden

$cb.AlleGueten->wpCheckState      # _WinStateChkUnchecked;

$cb.MarkierteGueten->wpCheckState # _WinStateChkUnchecked;

Lib_GuiCom:Disable($cb.AlleGueten);

Lib_GuiCom:Disable($cb.MarkierteGueten);

$rbArtikel->wpCheckState # _WinStateChkchecked;

$rbMaterial->wpCheckState # _WinStateChkUnchecked;

end;

```

```

if (aEvt:Obj->wpName = 'cb.AlleArtikelnummer') AND ($cb.MarkierteArtikelnummer->wpCheckState = _V)

$cb.AlleArtikelnummer->wpCheckState      # _WinStateChkchecked;

$cb.MarkierteArtikelnummer->wpCheckState # _WinStateChkUnchecked;

// Material ausblenden

$cb.AlleGueten->wpCheckState      # _WinStateChkUnchecked;

$cb.MarkierteGueten->wpCheckState # _WinStateChkUnchecked;

Lib_GuiCom:Disable($cb.AlleGueten);

Lib_GuiCom:Disable($cb.MarkierteGueten);

```

```
$rbArtikel->wpCheckState # _WinStateChkchecked;
```

```
$rbMaterial->wpCheckState # _WinStateChkUnchecked;
```

```
end;
```

```
if (aEvt:Obj->wpName = 'cb.MarkierteArtikelnummer') AND ($cb.AlleArtikelnummer->wpCheckState = _V
```

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

```
    $cb.MarkierteArtikelnummer->wpCheckState # _WinStateChkchecked;
```

```
// Material ausblenden
```

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

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

```
Lib_GuiCom:Disable($cb.AlleGueten);
```

```
Lib_GuiCom:Disable($cb.MarkierteGueten);
```

```
$rbArtikel->wpCheckState # _WinStateChkchecked;
```

```
$rbMaterial->wpCheckState # _WinStateChkUnchecked;
```

```
end;
```

```
if ((aEvt:Obj->wpName = 'cb.AlleGueten') AND ($cb.MarkierteGueten->wpCheckState = _WinStateChkchecked)
```

```
then begin
```

```
    $cb.AlleGueten->wpCheckState      # _WinStateChkchecked;
```

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

```
// Artikel ausblenden
```

```
$rbArtikel->wpCheckState # _WinStateChkUnchecked;
```

```
$rbMaterial->wpCheckState # _WinStateChkchecked;
```

```
end;
```

```
if ((aEvt:Obj->wpName = 'cb.MarkierteGueten') AND ($cb.AlleGueten->wpCheckState = _WinStateChkchecked) then begin  
    $cb.AlleGueten->wpCheckState      # _WinStateChkUnchecked;  
    $cb.MarkierteGueten->wpCheckState # _WinStateChkchecked;  
  
    // Artikel ausblenden  
  
    $rbArtikel->wpCheckState # _WinStateChkUnchecked;  
    $rbMaterial->wpCheckState # _WinStateChkchecked;  
  
end;
```

```
// Artikelselektion
```

```
if ($rbArtikel->wpCheckState = _WinStateChkchecked) then begin
```

```
    // Material ausblenden
```

```
    $cb.AlleGueten->wpCheckState      # _WinStateChkUnchecked;  
    $cb.MarkierteGueten->wpCheckState # _WinStateChkUnchecked;  
    Lib_GuiCom:Disable($cb.AlleGueten);  
    Lib_GuiCom:Disable($cb.MarkierteGueten);
```

```
    // Artikel einblenden
```

```
    Lib_GuiCom:Enable($cb.MarkierteArtikelgruppen);  
    Lib_GuiCom:Enable($cb.AlleArtikelgruppen);  
    Lib_GuiCom:Enable($cb.MarkierteArtikelnummer);  
    Lib_GuiCom:Enable($cb.AlleArtikelnummer);
```

```
end;
```

```
// Materialselektion
```

```
if ($rbMaterial->wpCheckState = _WinStateChkchecked) then begin
```

```
    // Artikel ausblenden
```



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

$cb.AlleArtikelgruppen->wpCheckState # _WinStateChkUnchecked;

$cb.MarkierteArtikelnummer->wpCheckState # _WinStateChkUnchecked;

$cb.AlleArtikelnummer->wpCheckState # _WinStateChkUnchecked;

Lib_GuiCom:Disable($cb.MarkierteArtikelgruppen);

Lib_GuiCom:Disable($cb.AlleArtikelgruppen);

Lib_GuiCom:Disable($cb.MarkierteArtikelnummer);

Lib_GuiCom:Disable($cb.AlleArtikelnummer);


// Material einblenden

Lib_GuiCom:Enable($cb.AlleGueten);

Lib_GuiCom:Enable($cb.MarkierteGueten);

end;
```

```
$cb.AlleKunden->winupdate(_WinUpdFld2Obj);

$cb.MarkierteKunden->winupdate(_WinUpdFld2Obj);

$cb.AlleVertreter->winupdate(_WinUpdFld2Obj);

$cb.MarkierteVertreter->winupdate(_WinUpdFld2Obj);

$cb.AlleAuftragsarten->winupdate(_WinUpdFld2Obj);

$cb.MarkierteAuftragsarten->winupdate(_WinUpdFld2Obj);

$cb.AlleWarengruppen->winupdate(_WinUpdFld2Obj);

$cb.MarkierteWarengruppen->winupdate(_WinUpdFld2Obj);

$cb.AlleArtikelgruppen->winupdate(_WinUpdFld2Obj);

$cb.MarkierteArtikelgruppen->winupdate(_WinUpdFld2Obj);

$cb.AlleArtikelnummer->winupdate(_WinUpdFld2Obj);
```

```
$cb.MarkierteArtikelnummer->winupdate(_WinUpdFld2Obj);  
$cb.AlleGueten->winupdate(_WinUpdFld2Obj);  
$cb.MarkierteGueten->winupdate(_WinUpdFld2Obj);  
$rbMaterial->winupdate(_WinUpdFld2Obj);  
$rbArtikel->winupdate(_WinUpdFld2Obj);  
end;
```

```
if (vDialog = 'Mat_Neubewertung') then begin
```

```
  Lib_GuiCom:Enable($ed.Preis);  
  Lib_GuiCom:Enable($ed.AenderungProz);  
  Lib_GuiCom:Enable($ed.AenderungEUR);  
  Lib_GuiCom:Enable($cb.nurAbwertung);
```

```
if ($ed.Preis->wpCaptionFloat <> 0.0) then begin
```

```
  $ed.AenderungEUR->wpCaptionFloat # 0.0;  
  Lib_GuiCom:Disable($ed.AenderungEUR);
```

```
  $ed.AenderungProz->wpCaptionFloat # 0.0;
```

```
  Lib_GuiCom:Disable($ed.AenderungProz);
```

```
end;
```

```
if ($ed.AenderungEUR->wpCaptionFloat<> 0.0) then begin
```

```
  $ed.Preis->wpCaptionFloat # 0.0;
```

```
  Lib_GuiCom:Disable($ed.Preis);
```

```
$ed.AenderungProz->wpCaptionFloat # 0.0;
```

```
Lib_GuiCom:Disable($ed.AenderungProz);
```

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

```
Lib_GuiCom:Disable($cb.nurAbwertung);
```

```
end;
```

```
if ($ed.AenderungProz->wpCaptionFloat<> 0.0) then begin
```

```
    $ed.Preis->wpCaptionFloat # 0.0;
```

```
    Lib_GuiCom:Disable($ed.Preis);
```

```
$ed.AenderungEur->wpCaptionFloat # 0.0;
```

```
Lib_GuiCom:Disable($ed.AenderungEur);
```

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

```
Lib_GuiCom:Disable($cb.nurAbwertung);
```

```
end;
```

```
end;
```

```
RETURN true;
```

```
end;
```

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

```
// EvtClose
```

```
//      Schliessen eines Fensters
```

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

```
sub EvtClose(
```

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

```
): logic
```

```
begin
```

```
    case vDialog of
```

```
        'DatumVonBis' : begin
```

```
            vDatum1 # $edDate1->wpCaptiondate;
```

```
            vDatum2 # $edDate2->wpCaptiondate;
```

```
        end;
```

```
        'Datum' : begin
```

```
            vDatum1 # $edDate1->wpCaptiondate;
```

```
        end;
```

```
        'Zeit' : begin
```

```
            vZeit # $edTime1->wpCaptionTime;
```

```
        end;
```

```
        'Anzahl' : begin
```

```
            vZahl1 # $IntEdit1->wpCaptionInt;
```

```
        end;
```

```
        'AnzahlVonBis' : begin
```

```
            vZahl1 # $IntEdit1->wpCaptionInt;
```

```
            vZahl2 # $IntEdit2->wpCaptionInt;
```

```
        end;
```

```
        'NrPosPos' : begin
```

```

    vZahl1 # $IntEdit1->wpCaptionInt;

    vZahl2 # $IntEdit2->wpCaptionInt;

    vZahl3 # $IntEdit3->wpCaptionInt;

end;

'Menge' : begin

    vMenge # $FloatEdit1->wpCaptionFloat;

end;

'Standard' : begin

    vText # $edText->wpCaption;

end;

'Standard_Small' : begin

    vText # $edText->wpCaption;

end;

'KW' : begin

    vKW  # cnvIA($edKW->wpCaption);

    vJahr # cnvIA($edJahr->wpCaption);

end;

end;


RETURN true;

end;


//=====

// EvtKeyItem

//

```

//=====

sub EvtKeyItem(

 aEvt : event; // Ereignis

 aKey : int; // Taste

 aID : int; // RecID bei RecList, Node-Deskriptor bei TreeView, Focus-Objekt bei Frame und

) : logic;

begin

 if (aKey=_WinKeyF2) then begin

 aEvt:obj->WinDialogResult(_winidok);

 aEvt:Obj->WinClose();

 end;

 if (aKey=_WinKeyJ) then begin

 aEvt:obj->WinDialogResult(_WinIdYes);

 aEvt:Obj->WinClose();

 end;

 if (aKey=_WinKeyN) then begin

 aEvt:obj->WinDialogResult(_WinIdNo);

 aEvt:Obj->WinClose();

 end;

 RETURN(true);

end;

```

//=====

// KWJahr

//

//=====

sub KWJahr(

    aFrage : alpha;

    var aKW : word;

    var aJahr : word;

) : logic;

local begin

    vID    : int;

    vPrefix : alpha;

end;

begin

    vDialog # 'KW';

    vFrage # aFrage;

    vKW    # aKW;

    vJahr  # aJahr;

    vPrefix # gPrefix;

    // gPrefix # ";

    // vId # WinDialog('Dlg.Kalenderwoche',_WinDialogCenter,gMDI);//gFrmMain);

    vId # WinDialog('Dlg.Kalenderwoche',0,gMDI);//gFrmMain);

    // gPrefix # vPrefix;

    If (vId = _WinIdOk) then begin

        aKW    # vKW;

        aJahr  # vJahr;

```

```

    RETURN true;

end

else begin

    RETURN false;

end;

end;

//=====

//=====

// Standard

//

//=====

sub Standard(

    aFrage      : alpha;

    var aText   : alpha;

    opt aPW     : logic;

    opt aMaxChars : int;

) : logic;

local begin

    Erx : int;

    vID  : int;

    vPrefix : alpha;

    vMDI  : int;

    vHdl  : int;

end;

```


begin

vDialog # 'Standard';

vFrage # aFrage;

vText # aText;

vPrefix # gPrefix;

// gPrefix # '';

// vld # WinDialog('Dlg.Standard',_WinDialogCenterScreen,gMDI);//gFrmMain);

// 28.03.2012 AI: richtigen Parent suchen...

vMDI # gMDI;

if (gMDI=0) then vMDI # gFrmMain;

if (gMDI=gMDINotifier) or (gMDI=gMdiWorkbench) or (gMDI=gMdiMenu) then vMDI # gFrmMain;

// if (WinfocusGet()<>0) then

vHdl # WinOpen('Dlg.Standard',_WinOpenDialog)

// else

// vHdl # WinOpen('Dlg.Standard',_WinOpenDialog | _WinDialogNoActivate);

//vld # WinDialog('Dlg.Standard',0,gMDI)

// gMDI # vMDI;

if (aPW) then begin

Erx # vHDL->winsearch('edText');

Erx->wpPassword # y;

end;

if (aMaxChars<>0) then begin

```

    Erx # vHDL->winsearch('edText');

    Erx->wpLengthMax # aMaxChars;

end;

vID    # vHdl->Winialogrun(0,vMDI);//gFrmMain);

// else

//   vId # WinDialog('Dlg.Standard',0,gFrmMain);

//   gPrefix # vPrefix;

If (vId = _WinIdOk) then begin

    aText # vText;

    vHdl->winclose();

    RETURN true;

end;

aText # "";

vHdl->winclose();

RETURN false;

end;

//=====

// Standard_Small

//

//=====

sub Standard_Small(

```

```

aFrage      : alpha;

var aText    : alpha;

opt aEinReturn : logic;

opt aMaxChars : int;

) : logic;

local begin

    vID      : int;

    vMDI     : int;

    vHdl     : int;

    vHdl2    : int;

end;

begin

    vDialog # 'Standard_Small';

    vNoMDI  # y;

    vFrage  # aFrage;

    vText   # aText;

    gSelected # 0;


    // 28.03.2012 AI: richtigen Parent suchen...

    vMDI # gMDI;

    if (gMDI=0) then vMDI # gFrmMain;

    if (gMDI=gMDINotifier) or (gMDI=gMdiWorkbench) or (gMDI=gMdiMenu) then vMDI # gFrmMain;


    // vld # WinDialog('Dlg.Standard.Small', _WinDialogCenter,gFrmMain)

    vHdl # WinOpen('Dlg.Standard.Small',_WinOpenDialog);

    if (aEinReturn) then begin

```

```

vHdl2 # Winsearch(vHdl,'OK');

vHdl2->wpcustom # 'END';

end;

if (aMaxChars<>0) then begin

    vHdl2 # vHDL->winsearch('edText');

    vHdl2->wpLengthMax # aMaxChars;

end;


vID # Windialogrun(vHdl,_WinDialogCenter,vMDI);//gFrmMain);

vHdl->winclose();

If (vId = _WinIdOk) or (gSelected<>0) then begin

    gSelected # 0;

    aText # vText;

    RETURN true;

end

else begin

    aText # "";

    RETURN false;

end;

end;


//=====

// Datum: Von... bis...

//

//=====

```

```

sub DatumVonBis(
    aFrage : alpha;
    var aVon : date; // Pointer zu Datumsfelder
    var aBis : date;
    opt aVorVon : date; // Vorgaben für Feldinhalt
    opt aVorBis : date;
) : logic;
local begin
    vID : int;
    vPrefix : alpha;
end;
begin
    vDialog # 'DatumVonBis';
    vFrage # aFrage;

    if (aVorVon != 00.00.00) then
        vDatum1 # aVorVon;
    else
        vDatum1 # 00.00.00;
    if (aVorBis != 00.00.00) then
        vDatum2 # aVorBis;
    else
        vDatum2 # 00.00.00;

    vPrefix # gPrefix;
    // gPrefix # "";

```

```
// vld # WinDialog('Dlg.DatumVonBis', _WinDialogCenter, gMDI);//gFrmMain);
```

```
    vld # WinDialog('Dlg.DatumVonBis', 0, gMDI);//gFrmMain);
```

```
// gPrefix # vPrefix;
```

```
    If (vld = _WinIdOk) then begin
```

```
        aVon # vDatum1;
```

```
        aBis # vDatum2;
```

```
        RETURN true;
```

```
    end
```

```
    else begin
```

```
        RETURN false;
```

```
    end;
```

```
end;
```

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

```
// Datum
```

```
//
```

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

```
sub Datum(
```

```
    aFrage : alpha;
```

```
    var aDat : date; // Pointer zu Datumsfelder
```

```
    opt aVor : date; // Vorgaben für Feldinhalt
```

```
) : logic;
```

```
local begin
```

```
    vID : int;
```

```

    vPrefix : alpha;

end;

begin

    vDialog # 'Datum';

    vFrage # aFrage;


    if (aVor != 00.00.00) then

        vDatum1 # aVor;

    else

        vDatum1 # 00.00.00;


    vPrefix # gPrefix;

// gPrefix # ";

// vId # WinDialog('Dlg.Datum', _WinDialogCenter,gMDI);//gFrmMain);

    vId # WinDialog('Dlg.Datum', 0,gMDI);//gFrmMain);

// gPrefix # vPrefix;


    If (vId = _WinIdOk) then begin

        aDat # vDatum1;

        RETURN true;

    end

    else begin

        RETURN false;

    end;

end;

```

```

//=====

// Anzahl abfragen

//

//=====

sub Anzahl(

    aFrage      : alpha;

    var aZahl    : int; // Pointer zu Integer

    opt aVorgabe : int; // Vorgabe für Feld

    opt aXPos    : int;

    opt aYPos    : int;

) : logic;

local begin

    vID      : int;

    vPrefix : alpha;

    vHdl     : int;

end;

begin

    vDialog # 'Anzahl';

    vFrage # aFrage;


    if (aVorgabe != 0) then

        vZahl1 # aVorgabe;

    else

        vZahl1 # 0;

    vPrefix # gPrefix;

```



```

// gPrefix # ";

// vId # WinDialog('Dlg.Anzahl',_WinDialogCenterScreen,gFrmMain);

vHdl # WinOpen('Dlg.Anzahl',_winopendialog);

if (aXPos=0) and (aYPos=0) then begin
    if (gMDI<>0) then
        vID # vHdl->WinDialogRun(_WinDialogCenter, gMDI)
    else
        vID # vHdl->WinDialogRun(_WinDialogCenter,gFrmMain);
end

else begin

    Lib_guiCom:ObjSetPos(vHdl,aXPos,aYPos);

    if (gMDI<>0) then
        vID # vHdl->WinDialogRun(0,gMDI)
    else
        vID # vHdl->WinDialogRun(0,gFrmMain);

end;

vHdl->winclose();

// gPrefix # vPrefix;

If (vId = _WinIdOk) then begin
    aZahl # vZahl1;

    RETURN true;
end

else begin

    RETURN false;
end

```

end;

end;

//=====

// AnzahlVonBis abfragen

//

//=====

sub AnzahlVonBis(

 aFrage : alpha;

 var aZahl1 : int; // Pointer zu Integer

 var aZahl2 : int; // Pointer zu Integer

 opt aVorgabe1 : int; // Vorgabe für Feld

 opt aVorgabe2 : int; // Vorgabe für Feld

) : logic;

local begin

 vID : int;

 vPrefix : alpha;

 vHdl : int;

end;

begin

 vDialog # 'AnzahlVonBis';

 vFrage # aFrage;

 if (aVorgabe1 <> 0) then

 vZahl1 # aVorgabe1;

else

vZahl1 # 0;

if (aVorgabe2 <> 0) then

vZahl2 # aVorgabe2;

else

vZahl2 # 0;

vPrefix # gPrefix;

vHdl # WinOpen('Dlg.AnzahlVonBis', _winopendialog);

if (gMDI<>0) then

vID # vHdl->WinDialogRun(0,gMDI)

else

vID # vHdl->WinDialogRun(0,gFrmMain);

vHdl->winclose();

if (vId = _WinIdOk) then begin

aZahl1 # vZahl1;

aZahl2 # vZahl2;

RETURN true;

end

else begin

RETURN false;

end;

end;

```

//=====

// NrPosPos abfragen

//

//=====

sub NrPosPos(
    aFrage      : alpha;

    var aZahl1   : int; // Pointer zu Integer

    var aZahl2   : word;

    var aZahl3   : word;

    opt aVorgabe1 : int;

    opt aVorgabe2 : word;

    opt aVorgabe3 : word;

) : logic;

local begin

    vID      : int;

    vPrefix : alpha;

    vHdl     : int;

end;

begin

    vDialog # 'NrPosPos';

    vFrage # aFrage;


    if (aVorgabe1 <> 0) then

        vZahl1 # aVorgabe1;

```

else

vZahl1 # 0;

if (aVorgabe2 <> 0) then

vZahl2 # aVorgabe2;

else

vZahl2 # 0;

if (aVorgabe3 <> 0) then

vZahl3 # aVorgabe3;

else

vZahl3 # 0;

vPrefix # gPrefix;

vHdl # WinOpen('Dlg.NrPosPos', _winopendialog);

if (gMDI<>0) then

vID # vHdl->WinDialogRun(0,gMDI)

else

vID # vHdl->WinDialogRun(0,gFrmMain);

vHdl->winclose();

if (vId = _WinIdOk) then begin

aZahl1 # vZahl1;

aZahl2 # vZahl2;

aZahl3 # vZahl3;

```
    RETURN true;

end

else begin

    RETURN false;

end;

end;
```

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

```
// Zeit abfragen
```

```
//
```

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

```
sub Zeit(

    aFrage      : alpha;

    var aZeit   : Time; // Pointer

    opt aVorgabe : Time; // Vorgabe für Feld

) : logic;

local begin

    vID      : int;

    vPrefix : alpha;

end;

begin

    vDialog # 'Zeit';

    vFrage # aFrage;

    if (aVorgabe != 0:0) then
```

```

    vZeit # aVorgabe;

else

    vZeit # 0:0;


vPrefix # gPrefix;

// gPrefix # ";

// vld # WinDialog('Dlg.Zeit', _WinDialogCenter,gMDI);//gFrmMain);

    vld # WinDialog('Dlg.Zeit', 0,gMDI);//gFrmMain);

// gPrefix # vPrefix;


If (vld = _WinIdOk) then begin

    aZeit # vZeit;

    RETURN true;

end

else begin

    RETURN false;

end;

end;


//=====

// Menge abfragen

//

//=====

sub Menge(

    aFrage      : alpha;

```

```

var aMenge : float; // Pointer

opt aVorgabe : float; // Vorgabe für Feld

) : logic;

local begin

    vID : int;

    vPrefix : alpha;

end;

begin

    vDialog # 'Menge';

    vFrage # aFrage;


    if (aVorgabe != 0.0) then

        vMenge # aVorgabe;

    else

        vMenge # 0.0;


    vPrefix # gPrefix;

// gPrefix # "";

// vId # WinDialog('Dlg.Menge', _WinDialogCenter,gMDI);//gFrmMain);

    vId # WinDialog('Dlg.Menge', 0,gMDI);//gFrmMain);

// gPrefix # vPrefix;


    If (vId = _WinIdOk) then begin

        aMenge # vMenge;

        RETURN true;

    end

```



```
else begin  
    RETURN false;  
end;  
end;
```

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

```
// InfoBetrieb
```

```
//  Zeigt eine Information für den Betrieb an:
```

```
//    - Größere Schrift
```

```
//    - Farbiger
```

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

```
sub InfoBetrieb(  
    aTitel      : alpha;  
    aText       : alpha;  
    opt aNegativ : logic;  
    ) : logic;
```

```
local begin
```

```
    vID   : int;
```

```
    vMDI  : int;
```

```
    vHdl  : int;
```

```
    vHdl2 : int;
```

```
end;
```

```
begin
```

```
    vDialog # 'Dlg.InfoBetrieb';
```

```
    vNoMDI  # y;
```

```

vHdl # WinOpen('Dlg.InfoBetrieb',_WinOpenDialog);

// Titel setzen

vHdl->wpCaption # aTitel;

// Text setzen

vHdl2 # Winsearch(vHdl,'lbText');

vHdl2->wpCaption # aText;

// Farben setzen

if (aNegativ) then begin

    // ROT

    vHdl->wpColBkg # ColorRgbMake(255,45,45);

end

else begin

    // GRÜN

    vHdl->wpColBkg # ColorRgbMake(45,255,45);

end;

vID # Windialogrun(vHdl,_WinDialogCenter,gFrmMain);

vHdl->winclose();

RETURN true;

end;

```

```
//=====

// MATZ abfragen

//

//=====

sub MATZ(

    aTyp      : alpha;

    var aMitVersand : logic;

    var aBehalten  : logic;

    var aStk      : int;

    var aNetto    : float;

    var aBrutto   : float;

    var aMenge    : float;

    var aDatum    : date;

    var aVonDat   : date;

    var aBisDat   : date;

    var aZusatz   : alpha;

) : logic;

local begin

    Erx    : int;

    vID    : int;

    vPrefix : alpha;

    vOK    : logic;

    vHdl   : int;

end;
```

begin

vDialog # 'Matz';

if (aTyp='VSB') then vDialog # 'VSB';

vPrefix # gPrefix;

// gPrefix # ";

vHdl # WinOpen('Dlg.Matz',_WinOpenDialog);

if (aMitVersand) then begin

 Erx # vHdl->Winsearch('cb.Versand');

 Erx->wpvisible # true;

 Erx->wpCheckState # _WinStateChkChecked;

 Erx # vHdl->Winsearch('lb.DatumVon');

 Erx->wpvisible # true;

 Erx # vHdl->Winsearch('lb.DatumBis');

 Erx->wpvisible # true;

 Erx # vHdl->Winsearch('lb.Zusatz');

 Erx->wpvisible # true;

 Erx # vHdl->Winsearch('ed.DatumVon');

 Erx->wpvisible # true;

 Erx->wpcaptiondate # aVonDat;

 Erx # vHdl->Winsearch('ed.DatumBis');

 Erx->wpvisible # true;

 Erx->wpcaptiondate # aBisDat;

 Erx # vHdl->Winsearch('ed.Zusatz');

 Erx->wpvisible # true;

```
Erx->wpcaption    # aZusatz;

end;

if (aBehalten) then begin

    Erx    # vHdl->Winsearch('cb.BEHALTEN');

    Erx->wpvisible    # true;

    Erx->wpCheckState # _WinStateChkChecked;

end;


Erx    # vHdl->Winsearch('ed.Stueck');

Erx->wpcaptionint # aStk;

Erx->wpcustom # AInt(aStk);

Erx    # vHdl->Winsearch('ed.Brutto');

Erx->wpcaptionfloat # aBrutto;

Erx->wpDecimals # Set.Stellen.Gewicht; // 2022-12-01 AH

Erx->wpcustom # cnvaf(aBrutto);

Erx    # vHdl->Winsearch('ed.Netto');

Erx->wpcaptionfloat # aNetto;

Erx->wpDecimals # Set.Stellen.Gewicht; // 2022-12-01 AH

Erx->wpcustom # cnvaf(aNetto);

Erx    # vHdl->Winsearch('ed.Menge');

Erx->wpcaptionfloat # aMenge;

Erx->wpDecimals # Set.Stellen.Menge; // 2022-12-01 AH

Erx->wpcustom # cnvaf(aMenge);

Erx    # vHdl->Winsearch('ed.Datum');

Erx->wpcaptiondate # aDatum;

Erx->wpcustom # cnvad(aDatum);
```

```

// Dialog starten

REPEAT

    vID    # vHdl->Windialogrun(0,gMDI);//gFrmMain);

// gPrefix # vPrefix;

    if (vID<>_winIdok) then begin // Abbruch?

        vHdl->winclose();

        RETURN false;

    end

else begin          // Sichern?

    // 03.06.2016 AH:

    Erx    # vHdl->Winsearch('ed.Netto');

    if (Erx->wpColBkg=_WinColLightYellow) then begin

        Msg(1200,Translate('Nettogewicht'),0,0,0);

        CYCLE;

    end;

    Erx    # vHdl->Winsearch('ed.Brutto');

    if (Erx->wpColBkg=_WinColLightYellow) then begin

        Msg(1200,Translate('Bruttogewicht'),0,0,0);

        CYCLE;

    end;

    Erx    # vHdl->Winsearch('ed.Stueck');

    aStk   # Erx->wpcaptionint;

```

```

Erx    # vHdl->Winsearch('ed.Brutto');

aBrutto # Erx->wpcaptionfloat;

Erx    # vHdl->Winsearch('ed.Netto');

aNetto # Erx->wpcaptionfloat;

Erx    # vHdl->Winsearch('ed.Menge');

aMenge # Erx->wpcaptionfloat;

Erx    # vHdl->Winsearch('ed.Datum');

aDatum # Erx->wpcaptiondate;


Erx    # vHdl->Winsearch('ed.DatumVon');

aVonDat # Erx->wpcaptiondate;

Erx    # vHdl->Winsearch('ed.DatumBis');

aBisDat # Erx->wpcaptiondate;

Erx    # vHdl->Winsearch('ed.Zusatz');

aZusatz # Erx->wpcaption

Erx    # vHdl->Winsearch('cb.Versand');

aMitVersand # (Erx->wpCheckState=_WinStateChkChecked);

Erx    # vHdl->Winsearch('cb.BEHALTEN');

aBehalten # (Erx->wpCheckState=_WinStateChkChecked);


//    if (aMenge=0.0) then CYCLE;


vHdl->winclose();

RETURN true;

end;
```

UNTIL (1=1);

end;

//=====

// PosErfassung abfragen

//

//=====

sub PosErfassung(

 aAufpr : logic;

 aKalk : logic;

) : int;

local begin

 Erx : int;

 vID : int;

 vPrefix : alpha;

 vHdl : int;

end;

begin

 vPrefix # gPrefix;

 vHDL # WinOpen(Lib_GuiCom:GetAlternativeName('Dlg.PosErfassung'),_WinOpenDialog);

 if (aAufpr) then begin

 Erx # vHDL->Winsearch('cb.Aufpreise');


```

    Erx->wpCheckState # _WinStateChkChecked;

end;

if (aKalk) then begin

    Erx # vHDL->Winsearch('cb.Kalkulation');

    Erx->wpCheckState # _WinStateChkChecked;

end;


// Dialog starten

vID # vHDL->Windialogrun(0,gMDI);//gFrmMain);

// gPrefix # vPrefix;


If (vId = _WinIdNo) then begin

    vHDL->winclose();

    RETURN 0;

end;


vID # 1;

Erx  # vHDL->Winsearch('cb.Aufpreise');

if (Erx<>0) then

    if (Erx->wpCheckState=_WinStateChkchecked) then vID # vID + 2;

Erx  # vHDL->Winsearch('cb.Kalkulation');

if (Erx<>0) then

    if (Erx->wpCheckState=_WinStateChkchecked) then vID # vID + 4;


vHDL->winclose();

RETURN vID;

```

```

end;

//=====

// Material-Bestandsänderung

//

//=====

sub Mat_Bestand(

    var aStk    : int;

    var aNetto  : float;

    var aBrutto : float;

    var aPreis  : float;

    var aMenge  : float;

    var aGrund  : alpha;

    var aDat    : date;

    aEKEdit    : logic;

    aPEH       : alpha;

    opt aFixDatum : logic;

) : logic;

local begin

    Erx    : int;

    vHdl   : int;

    vId    : int;

    vPrefix : alpha;

    vA     : alpha(200);

end;

```

begin

// AFX

vA # AInt(astk)+'|'+ANum(aNetto,2)+'|'+ANum(aBrutto,2)+'|'+ANum(aPreis,2)+'|'+aGrund+'|'+cnvad(aDat);

// Übergabeparameter: "aStk|aNetto|aBrutto|aPreis|aGrund|aDatum"

Erx # RunAFX('Dlg.Mat.Bestand',vA);

if (Erx>0) then RETURN true;

if (Erx<0) then RETURN false;

vDialog # 'Mat_Bestand';

vPrefix # gPrefix;

vHdl # WinOpen('Dlg.Mat.Bestandsaenderung',_WinOpenDialog);

Erx # vHdl->WinSearch('ed.Stueck');

Erx->wpCaptionInt # aStk;

Erx # vHdl->WinSearch('ed.NettoGewicht');

Erx->wpCaptionFloat # aNetto;

Erx # vHdl->WinSearch('ed.BruttoGewicht');

Erx->wpCaptionFloat # aBrutto;

Erx # vHdl->WinSearch('lb.MEH');

Erx->wpCaption # Mat.MEH;

Erx # vHdl->WinSearch('ed.Menge');

if(Mat.MEH <> 't' and Mat.MEH <> 'kg') and (Mat.MEH<>'Stk') then

Erx->wpCaptionFloat # aMenge;

else

Lib_GuiCom:Disable(Erx);

Erx # vHdl->WinSearch('ed.Preis');

Erx->wpCaptionFloat # aPreis;

if (aEKEdit=false) then

Lib_GuiCom:Disable(Erx);

Erx # vHdl->WinSearch('ed.Datum');

Erx->wpCaptionDate # aDat;

if (aFixDatum) then Erx->wpdisabled # true;

Erx # vHdl->WinSearch('lb.WAEPEH');

if (aPEH="") then begin

aPEH # "Set.HausWährung.Kurz" + ' / ' + mat.meh; // 2022-09-05 AH ' / t';

if (Mat.MEH='kg') then aPEH # "Set.HausWährung.Kurz" + ' / t';

end;

Erx->wpCaption # aPEH;

// Dialog starten

REPEAT

vID # vHdl->Windialogrun(0,gMDI);//gFrmMain);

if (vID != _winIdOk) then begin // Abbruch?

vHdl->winclose();

RETURN false;

end

else begin // Sichern?

 Erx # vHdl->WinSearch('ed.Stueck');

 aStk # Erx->wpCaptionInt;

 Erx # vHdl->WinSearch('ed.NettoGewicht');

 aNetto # Erx->wpCaptionFloat;

 Erx # vHdl->WinSearch('ed.BruttoGewicht');

 aBrutto # Erx->wpCaptionFloat;

 Erx # vHdl->WinSearch('ed.Menge');

 aMenge # Erx->wpCaptionFloat;

 Erx # vHdl->WinSearch('ed.Preis');

 aPreis # Erx->wpCaptionFloat;

 Erx # vHdl->WinSearch('ed.Datum');

 aDat # Erx->wpCaptiondate;

 Erx # vHdl->WinSearch('ed.Grund');

 aGrund # Erx->wpCaption;

if (Mat.MEH='Stk') then // 2023-05-04 AH

 aMenge # cnvfi(aStk);

if (Lib_Faktura:Abschlusstest(aDat) = false) then begin

 Msg(001400 ,Translate('Werstellungsdatum') + '|' + CnvAd(aDat),0,0,0);

 CYCLE;

end;

vHdl->winclose();

RETURN true;

end;

UNTIL (1=1);

end;

//=====

// Material-Neubewertung

//

//=====

sub Mat_Neubewertung(

var aPreis : float;

var aAbwertEff : logic;

var aNurAb : logic;

var aGrund : alpha;

var aFix : logic;

var aDatum : date;

var aAenderungEUR : float;

var aAenderungProz : float;

) : logic;

local begin

Erx : int;

vHdl : int;

vId : int;

```

vPrefix : alpha;

vA      : alpha;

end;

begin

    // AFX

    vA # ANum(aPreis,2)+'|';

    if (aNurAB) then vA # vA + '1|'

    else vA # vA + '0|';

    vA # vA + aGrund;

    if (aFix) then vA # vA + '1|'

    else vA # vA + '0|';

    vA # vA + cnvad(aDatum)+'|';

    vA # vA + Anum(aAenderungEUR,2) + '|' + Anum(aAenderungPROZ,2) + '|';

    // Übergabeparameter: "aPreis|aNurAb|aGrund|aFix|aDatum|aAenderungEUR|aAenderungProz|"

    Erx # RunAFX('Dlg.Mat.Bewertung',vA);

    if (Erx>0) then RETURN true;

    if (Erx<0) then RETURN false;

    vDialog # 'Mat_Neubewertung';

    vPrefix # gPrefix;

    vHdl # WinOpen('Dlg.Mat.Neubewertung',_WinOpenDialog);

    Erx # vHdl->WinSearch('cb.NurAbwertung');

    if (aNurAb) then Erx->wpCheckState # _WinStateChkChecked

```

```
else Erx->wpCheckState # _WinStateChkUnchecked;

Erx # vHdl->WinSearch('cb.Fix');

if (aFix) then Erx->wpCheckState # _WinStateChkChecked

else Erx->wpCheckState # _WinStateChkUnchecked;

Erx # vHdl->WinSearch('ed.Preis');

Erx->wpCaptionFloat # aPreis;

Erx # vHdl->WinSearch('ed.Datum');

Erx->wpCaptionDate # aDatum;

Erx # vHdl->WinSearch('rb.Grundpreis');

Erx->wpCheckState # _WinStateChkChecked;

Erx # vHdl->WinSearch('rb.Effektivpreis');

Erx->wpCheckState # _WinStateChkUnchecked;
```

```
// Dialog starten
```

```
REPEAT
```

```
  vID # vHdl->Windialogrun(0,gMDI);//gFrmMain);
```

```
  if (vID <> _winIdOk) then begin // Abbruch?
```

```
    vHdl->winclose();
```

```
    RETURN false;
```

```
  end
```

```
  else begin // Sichern?
```

```
    Erx # vHdl->WinSearch('cb.NurAbwertung');
```

```
    aNurAb # (Erx->wpCheckState=_WinStateChkChecked);
```

```
    Erx # vHdl->WinSearch('cb.Fix');
```

```
    aFix # (Erx->wpCheckState=_WinStateChkChecked);
```

```
    Erx # vHdl->WinSearch('ed.Preis');
```



```

aPreis # Erx->wpCaptionFloat;

Erx    # vHdl->WinSearch('ed.Grund');

aGrund # Erx->wpCaption;

Erx    # vHdl->WinSearch('ed.Datum');

aDatum # Erx->wpCaptiondate;

Erx    # vHdl->WinSearch('rb.Grundpreis');

if(Erx->wpCheckState = _WinStateChkChecked) then

    aAbwertEff # false;

else

    aAbwertEff # true;

Erx    # vHdl->WinSearch('ed.AenderungEUR');

aAenderungEUR # Erx->wpCaptionFLoat;

Erx    # vHdl->WinSearch('ed.AenderungProz');

aAenderungProz # Erx->wpCaptionFLoat;

if (Lib_Faktura:Abschlusstest(aDatum) = false) then begin

    Msg(001400 ,Translate('Werstellungsdatum') + '|' + CnvAd(aDatum),0,0,0);

    CYCLE;

end;

vHdl->winclose();

RETURN true;

end;

UNTIL (1=1);

```

end;

//=====

// Controlling - Kennzahlen generieren

//

//=====

sub Con_Generieren(

var aJahr : int;

var aTyp : int;

var aNurMitUms : logic;

var aKndAlle : logic;

var aKndMark : logic;

var aVerAlle : logic;

var aVerMark : logic;

var aAArAlle : logic;

var aAArMark : logic;

var aWgrAlle : logic;

var aWgrMark : logic;

var aKstAlle : logic;

var aKstMark : logic;

var aArgAlle : logic;

var aArgMark : logic;

var aArtAlle : logic;

var aArtMark : logic;

```

var aGteAlle : logic;

var aGteMark : logic;

var aConTyp : alpha;

) : logic;

local begin

    Erx : int;

    vHdl : int;

    vId : int;

    vPrefix : alpha;

    vA : alpha(200);

    vCheck : logic;

end;

begin

    vDialog # 'Con_Generieren';

    vPrefix # gPrefix;

    vHdl # WinOpen('Dlg.Con.Generieren',_WinOpenDialog);

    Erx # vHdl->WinSearch('edJahr');

    Erx->wpCaption # AInt(DateYear(SysDate()+1900);

    // Dialog starten

    REPEAT

        vID # vHdl->Windialogrun(0,gMDI);//gFrmMain);

        if (vID != _winIdOk) then begin // Abbruch?

            vHdl->winclose();

```

```

RETURN false;

end

else begin // Sichern?

    Erx    # vHdl->WinSearch('cbErl');

    if (Erx->wpCheckState=_WinStateChkUnchecked) then aConTyp # 'Erl';

    Erx    # vHdl->WinSearch('cbAuf');

    if (Erx->wpCheckState=_WinStateChkUnchecked) then aConTyp # 'Auf';

    Erx    # vHdl->WinSearch('cbAng');

    if (Erx->wpCheckState=_WinStateChkUnchecked) then aConTyp # 'Ang';

    Erx    # vHdl->WinSearch('cbBest');

    if (Erx->wpCheckState=_WinStateChkUnchecked) then aConTyp # 'Best';

    if (aConTyp='') then CYCLE;


    Erx    # vHdl->WinSearch('edJahr');

    aJahr   # Cnvla(Erx->wpCaption);


    if (aJahr = 0) then

        CYCLE;


    Erx    # vHdl->WinSearch('cb.AlleKunden');

    aKndAlle # (Erx->wpCheckState=_WinStateChkChecked);


    Erx    # vHdl->WinSearch('cb.MarkierteKunden');

    aKndMark # (Erx->wpCheckState=_WinStateChkChecked);


    Erx    # vHdl->WinSearch('cb.AlleVertreter');

```

aVerAlle # (Erx->wpCheckState=_WinStateChkChecked);

Erx # vHdl->WinSearch('cb.MarkierteVertreter');

aVerMark # (Erx->wpCheckState=_WinStateChkChecked);

Erx # vHdl->WinSearch('cb.AlleAuftragsarten');

aAArAlle # (Erx->wpCheckState=_WinStateChkChecked);

Erx # vHdl->WinSearch('cb.MarkierteAuftragsarten');

aAArMark # (Erx->wpCheckState=_WinStateChkChecked);

Erx # vHdl->WinSearch('cb.AlleWarengruppen');

aWgrAlle # (Erx->wpCheckState=_WinStateChkChecked);

Erx # vHdl->WinSearch('cb.MarkierteWarengruppen');

aWgrMark # (Erx->wpCheckState=_WinStateChkChecked);

/*

Erx # vHdl->WinSearch('cb.AlleKostenstellen');

aKstAlle # (Erx->wpCheckState=_WinStateChkChecked);

Erx # vHdl->WinSearch('cb.MarkierteKostenstellen');

aKstMark # (Erx->wpCheckState=_WinStateChkChecked);

*/

Erx # vHdl->WinSearch('cb.AlleArtikelgruppen');

```
aArgAlle # (Erx->wpCheckState=_WinStateChkChecked);
```

```
Erx      # vHdl->WinSearch('cb.MarkierteArtikelgruppen');
```

```
aArgMark # (Erx->wpCheckState=_WinStateChkChecked);
```

```
Erx      # vHdl->WinSearch('cb.AlleArtikelnummer');
```

```
aArtAlle # (Erx->wpCheckState=_WinStateChkChecked);
```

```
Erx      # vHdl->WinSearch('cb.MarkierteArtikelnummer');
```

```
aArtMark # (Erx->wpCheckState=_WinStateChkChecked);
```

```
Erx      # vHdl->WinSearch('cb.AlleGueten');
```

```
aGteAlle # (Erx->wpCheckState=_WinStateChkChecked);
```

```
Erx      # vHdl->WinSearch('cb.MarkierteGueten');
```

```
aGteMark # (Erx->wpCheckState=_WinStateChkChecked);
```

```
// Vorbelegung des Types
```

```
Erx      # vHdl->WinSearch('rbArtikel');
```

```
vCheck   # (Erx->wpCheckState=_WinStateChkChecked);
```

```
if (vCheck) then
```

```
    aTyp   # 250;
```

```
Erx      # vHdl->WinSearch('rbMaterial');
```

```
vCheck   # (Erx->wpCheckState=_WinStateChkChecked);
```

if (vCheck) then

 aTyp # 200;

if (aTyp = 0) then

 aTyp # 200;

// Kennzahlen auf Umsatzbasis generieren

Erx # vHdl->WinSearch('cb.mitUmsatz');

aNurMitUms # (Erx->wpCheckState=_WinStateChkChecked);

vHdl->winclose();

RETURN true;

end;

UNTIL (1=1);

end;

//=====

// Controlling - Kennzahlen als Vorgaben generieren

//

//=====

sub Con_GenerierenVorgaben(

 var aJahr : int;

 var aFaktMenge : float;

```

var aFaktUmsatz : float;

var aFaktDB    : float;

) : logic;

local begin

    Erx    : int;

    vHdl   : int;

    vId    : int;

    vPrefix : alpha;

    vA     : alpha(200);

end;

begin

    vDialog # 'Con_GenerierenVorgaben';

    vPrefix # gPrefix;

    vHdl # WinOpen('Dlg.Con.GenerierenVorgaben',_WinOpenDialog);

    Erx # vHdl->WinSearch('edJahr');

    Erx->wpCaption # AInt(DateYear(SysDate())+1901);

    // Dialog starten

    REPEAT

        vID # vHdl->Windialogrun(0,gMDI);//gFrmMain);

        if (vID != _winIdOk) then begin // Abbruch?

            vHdl->winclose();

            RETURN false;

        end

```


else begin // Sichern?

 Erx # vHdl->WinSearch('edJahr');

 aJahr # Cnvla(Erx->wpCaption);

 if (aJahr = 0) then

 CYCLE;

 Erx # vHdl->WinSearch('edMenge');

 aFaktMenge # Erx->wpCaptionFloat;

 Erx # vHdl->WinSearch('edUmsatz');

 aFaktUmsatz # Erx->wpCaptionFloat;

 Erx # vHdl->WinSearch('edDb');

 aFaktDB # Erx->wpCaptionFloat;

 vHdl->winclose();

 RETURN true;

end;

UNTIL (1=1);

end;

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

```
// Auswahl
```

```
//      Dynamische Auswahl anhand von verketteter Cte-Liste
```

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

```
sub Auswahl ( aCteLst : handle; opt aIndex : int; opt aTitle : alpha ) : int
```

```
local begin
```

```
    vDlg : handle;
```

```
    vLst : handle;
```

```
    vItem : handle;
```

```
end;
```

```
begin
```

```
    if ( aTitle = " ) then
```

```
        aTitle # 'Bitte auswählen...';
```

```
    vDialog # 'Auswahl';
```

```
    vFrage # aTitle;
```

```
    vDlg # WinOpen( 'Dlg.Auswahl', _winOpenDialog );
```

```
    if ( vDlg <= 0 ) then
```

```
        RETURN -1;
```

```
    vLst # vDlg->WinSearch( 'DL.Std.Auswahl' );
```

```
    if ( vLst <= 0 ) then
```

```
        RETURN -1;
```

```
    FOR vItem # aCteLst->CteRead( _cteFirst );
```

```

LOOP vItem # aCteLst->CteRead( _cteNext, vItem );

WHILE ( vItem != 0 ) DO BEGIN

    vLst->WinLstDatLineAdd( vItem->spName );

END;

vLst->wpCurrentInt # aIndex;


aIndex # vDlg->WinDialogRun();

vDlg->WinClose();


RETURN aIndex;

end;


//=====

// Auswahl_EvtMouseItem

//      Mausclick in der SFX Auswahl

//=====

sub Auswahl_EvtMouseItem ( aEvt : event; aButton : int; aHitTest : int; altem : int; ald : int ) : logic
begin

    if ( ( aButton & _winMouseLeft > 0 ) and ( aButton & _winMouseDouble > 0 ) ) then begin

        $Dlg.Auswahl->WinDialogResult( ald );

        $Dlg.Auswahl->WinClose();

    end;

    RETURN true;

end;

```

```

//=====

// Auswahl_EvtKeyItem

//      Tastendruck in der SFX Auswahl

//=====

sub Auswahl_EvtKeyItem ( aEvt : event; aKey : int; ald : int ): logic
begin
    if ( aKey = _winKeyReturn ) then begin
        $Dlg.Auswahl->WinDialogResult( ald );
        $Dlg.Auswahl->WinClose();
    end;

    RETURN true;
end;

//=====

// Tooltip

//=====

SUB Tooltip(aFrage : alpha(4096));
local begin
    vWin    : int;
    vPrefix : alpha;
    vHdl    : int;
end;
begin
    vDialog # 'Tooltip';

```

```
vPrefix # gPrefix;
```

```
vWin # WinOpen('Dlg.ToolTip',_WinOpenDialog)
```

```
vHdl # Winsearch(vWin,'lbText');
```

```
vHdl->wpcaption # aFrage;
```

```
vWin->Windialogrun(_WinDialogCenter,gFrmMain);
```

```
vWin->winclose();
```

```
end;
```

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

```
// TooltipRTF
```

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

```
SUB TooltipRTF(
```

```
    aTxt      : int;
```

```
    opt aCaption : alpha);
```

```
local begin
```

```
    vWin  : int;
```

```
    vPrefix : alpha;
```

```
    vHdl  : int;
```

```
    vWinBon : int;
```

```
    vZList : int;
```

```
end;
```

```
begin
```

```
    vWinBon # VarInfo(Windowbonus);
```

vZList # gZLList;

vDialog # 'Tooltip';

vPrefix # gPrefix;

vWin # WinOpen('Dlg.ToolTipRTF',_WinOpenDialog)

if (aCaption<>") then vWin->wpCaption # aCaption;

vHdl # Winsearch(vWin,'RtfEdit1');

vHdl->wpdbTextBuf # aTxt;

vHdl->WinRtfLoad(_WinStreamBufText,0,aTxt);

// vHdl->wpcaption # aFrage;

vWin->Windialogrun(_WinDialogCenter,gFrmMain);

vWin->winclose();

if (vWinBon<>0) then Varinstance(Windowbonus, vWinBon);

if (gMDI<>0) then begin

gMDI->winfocusset(true);

winsleep(100);

gZLList # vZList;

end;

end;

```
//=====

// Barcode

//

//=====

sub Barcode(

    aFrage      : alpha;

    var aText    : alpha;

) : logic;

local begin

    vID      : int;

    vMDI     : int;

    vHdl     : int;

end;

begin

    vDialog # 'Standard';

    vFrage  # aFrage;

    vText   # aText;

    vMDI # gMDI;

    if (gMDI=0) then vMDI # gFrmMain;

    if (gMDI=gMDINotifier) or (gMDI=gMdiWorkbench) or (gMDI=gMdiMenu) then vMDI # gFrmMain;

    vHdl # WinOpen('Dlg.Barcode',_WinOpenDialog)

    vID   # vHdl->Windialogrun(0,vMDI);

    aText # vText;
```

```
vHdl->winclose();
```

```
RETURN true;
```

```
end;
```

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

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