```
@A+
//==== Business-Control ================================================
//
//  Prozedur    Con2_Main
//                  OHNE E_R_G
//  Info
//
//
//  14.12.2009  AI  Erstellung der Prozedur
//  18.03.2010  ST  Ausgliederung Recalc, Generierung von Kennzahlen
//  25.03.2010  ST  Serienmarkierung hinzugefügt
//  03.08.2011  ST  Auftragsart hinzugefügt
//  05.04.2022  AH  ERX
//  21.07.2022  HA  Quick Jump
//
//  Subprozeduren
//    SUB EvtInit(aEvt : event) : logic
//    SUB RefreshDataList();
//    SUB Edit(aMonat : int);
//    SUB Pflichtfelder();
//    SUB RefreshIfm(optaName : alpha; optaChanged : logic)
//    SUB RecInit()
//    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 AusAdresse()
//    SUB AusVertreter()
//    SUB AusWGr()
//    SUB AusAGr()
//    SUB AusArtikel()
//    SUB AusGuete()
//    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 EvtMouseItem(aEvt : event; aButton : int; aHitTest : int; aItem : int; aID : int) : logic;
//    SUB EvtKeyItem(aEvt : event; aKey : int; aID : int) : logic;
//
//=========================================================================
@I:Def_Global
@I:Def_Rights


define begin
  cTitle      : 'Controlling'
  cFile       :  950
```

cMenuName : 'Con.Bearbeiten'

cPrefix : 'Con2'

cZList : $ZL.Controlling

cKey : 1

cClmMonat : 1

cClmSollM : 2

cClmIstM : 3

cClmSimM : 4

cClmSollErl : 5

cClmIstErl : 6

cClmSimErl : 7

cClmSollErlDS : 8

cClmIstErlDS : 9

cClmSimErlDS : 10

cClmSollDB : 11

cClmIstDB : 12

cClmSimDB : 13

cClmSollDBProz : 14

cClmIstDBProz : 15

cClmSimDBProz : 16

cClmSollDBDS : 17

cClmIstDBDS : 18

```
  cClmSimDBDS   : 19
end;

declare RefreshDataList();
declare CalcSumProz(aDL : int; opt aRep : logic)

//=========================================================================
// EvtInit
//         Initialisieren der Applikation
//=========================================================================
sub EvtInit(
  aEvt  : event;        // Ereignis
): logic
begin
  WinSearchPath(aEvt:Obj);
  gTitle    # Translate(cTitle);
  gFile     # cFile;
  gMenuName # cMenuName;
  gPrefix   # cPrefix;
  gZLList   # cZList;
  gKey      # cKey;

  Lib_Guicom2:Underline($edCon.Adressnummer);
  Lib_Guicom2:Underline($edCon.Auftragsart);
  Lib_Guicom2:Underline($edCon.Vertreternr);
  Lib_Guicom2:Underline($edCon.Warengruppe);
```

```
  Lib_Guicom2:Underline($edCon.Artikelnummer);
  Lib_Guicom2:Underline($edCon.Artikelgruppe);

  SetStdAusFeld('edCon.Adressnummer'  ,'Adresse');
  SetStdAusFeld('edCon.Vertreternr'   ,'Vertreter');
  SetStdAusFeld('edCon.Warengruppe'   ,'WGr');
  SetStdAusFeld('edCon.Artikelgruppe' ,'AGr');
  SetStdAusFeld('edCon.Artikelnummer' ,'Artikel');
  SetStdAusFeld('edCon.Auftragsart'   ,'Auftragsart');

  gZLList->wpColFocusBkg    # Set.Col.RList.Cursor;
  gZLList->wpColFocusOffBkg # "Set.Col.RList.CurOff";

  RefreshDataList();

  Lib_GuiCom:RecallList($dl.Con.Tabelle);

  App_Main:EvtInit(aEvt);

end;


//=========================================================================
//=========================================================================
sub Filter(aUpdate : logic)
local begin
```

```
  vMenge  : logic;
  vErloes : logic;
  vDB     : logic;
  vSoll   : logic;
  vIst    : logic;
  vSim    : logic;
  vDS     : logic;
end;
begin
  $dl.Con.Tabelle->WinUpdate(_WinupdOff);

  vDS     # ($cb.DS->wpCheckState=_WinStateChkChecked);
  vSim    # ($cb.Sim->wpCheckState=_WinStateChkChecked);
  vSoll   # ($cb.Soll->wpCheckState=_WinStateChkChecked);
  vIst    # ($cb.Ist->wpCheckState=_WinStateChkChecked);
  vMenge  # ($cb.Menge->wpCheckState=_WinStateChkChecked);
  vErloes # ($cb.Erloes->wpCheckState=_WinStateChkChecked);
  // nur bei Erlösen auch DB zeigen...
  vDB     # ($cb.DB->wpCheckState=_WinStateChkChecked) and (Con.Typ=");

  $clm.SollM->wpvisible     # vMenge and vSoll;
  $clm.IstM->wpvisible      # vMenge and vIst;
  $clm.SimM->wpvisible      # vMenge and vSim;

  $clm.SollErl->wpvisible   # vErloes and vSoll;
  $clm.IstErl->wpvisible    # vErloes and vIst;
```

```
  $clm.SimErl->wpvisible    # vErloes and vSim;
  $clm.SollErlDS->wpvisible # vErloes and vSoll and vDS;
  $clm.IstErlDS->wpvisible  # vErloes and vIst and vDS;
  $clm.SimErlDS->wpvisible  # vErloes and vSim and vDS;

  $clm.SollDB->wpvisible    # vDB and vSoll;
  $clm.IstDB->wpvisible     # vDB and vIst;
  $clm.SimDB->wpvisible     # vDB and vSim;
  $clm.SollDBProz->wpvisible  # vDB and vSoll;
  $clm.IstDBProz->wpvisible   # vDB and vIst;
  $clm.SimDBProz->wpvisible   # vDB and vSim;
  $clm.SollDBDS->wpvisible  # vDB and vSoll and vDS;
  $clm.IstDBDS->wpvisible   # vDB and vIst and vDS;
  $clm.SimDBDS->wpvisible   # vDB and vSim and vDS;

  if (aUpdate) then $dl.Con.Tabelle->WinUpdate(_WinupdOn);

end;


//=========================================================================
//  RefreshDataList
//
//=========================================================================
sub RefreshDataList();
local begin
```

```
  vI    : int;
  vX    : float;
  vHdl  : handle;
  vPos  : int;
  vA : alpha;
end;
begin

  vHdl # $dl.Con.Tabelle;
  vPos # vHdl->wpcurrentint;

  vHdl->WpAutoUpdate # false;
  vHDL->WinLstDatLineRemove(_winLstDatLineall);

  $clm.SollM->wpCaption     # Translate('Soll-Menge')+' '+Con.MEH;
  $clm.IstM->wpCaption      # Translate('Ist-Menge')+' '+Con.MEH;
  $clm.SimM->wpCaption      # Translate('Sim.-Menge')+' '+Con.MEH;

  FOR vI # 1 loop inc(vI) while (vI<=13) do begin

    if (vI=13) then
      vHDL->WinLstDatLineAdd('SUMME')
    else
      vHDL->WinLstDatLineAdd(lib_Berechnungen:Monat_aus_Datum(DateMake(1,vI,2000)));

    // MENGEN
```

```
vHDL->WinLstCellSet(FldFloat(950, 2, vI), cClmSollM       ,_WinLstDatLineLast);
vHDL->WinLstCellSet(FldFloat(950, 3, vI), cClmIstM        ,_WinLstDatLineLast);
vHDL->WinLstCellSet(FldFloat(950, 4, vI), cClmSimM        ,_WinLstDatLineLast);

// ERLÖS
vHDL->WinLstCellSet(FldFloat(950, 5, vI), cClmSollErl     ,_WinLstDatLineLast);
vHDL->WinLstCellSet(FldFloat(950, 6, vI), cClmIstErl      ,_WinLstDatLineLast);
vHDL->WinLstCellSet(FldFloat(950, 7, vI), cClmSimErl      ,_WinLstDatLineLast);

// DB
vHDL->WinLstCellSet(FldFloat(950,11, vI), cClmSollDB      ,_WinLstDatLineLast);
vHDL->WinLstCellSet(FldFloat(950,12, vI), cClmIstDB       ,_WinLstDatLineLast);
vHDL->WinLstCellSet(FldFloat(950,13, vI), cClmSimDb       ,_WinLstDatLineLast);

// Prozent
CalcSumProz(vHdl, true);
vHDL->WinLstCellSet(FldFloat(950, 8, vI), cClmSollDBProz  ,_WinLstDatLineLast);
vHDL->WinLstCellSet(FldFloat(950, 9, vI), cClmIstDBProz   ,_WinLstDatLineLast);
vHDL->WinLstCellSet(FldFloat(950,10, vI), cClmSimDBProz   ,_WinLstDatLineLast);

// Durchschnitt Erlös
vX # 0.0;
if (FldFloat(950, 2, vI)<>0.0) then vX # FldFloat(950,5,vI) / FldFloat(950,2,vI);
vHDL->WinLstCellSet(vX, cClmSollErlDS,_WinLstDatLineLast);
vX # 0.0;
if (FldFloat(950, 3, vI)<>0.0) then vX # FldFloat(950,6,vI) / FldFloat(950,3,vI);
```

```
    vHDL->WinLstCellSet(vX, cClmIstErlDS,_WinLstDatLineLast);
    vX # 0.0;
    if (FldFloat(950, 4, vI)<>0.0) then vX # FldFloat(950,7,vI) / FldFloat(950,4,vI);
    vHDL->WinLstCellSet(vX, cClmSimErlDS,_WinLstDatLineLast);

    // Durchschnitt DB
    vX # 0.0;
    if (FldFloat(950, 2, vI)<>0.0) then vX # FldFloat(950,11,vI) / FldFloat(950,2,vI);
    vHDL->WinLstCellSet(vX, cClmSollDBDS  ,_WinLstDatLineLast);
    vX # 0.0;
    if (FldFloat(950, 3, vI)<>0.0) then vX # FldFloat(950,12,vI) / FldFloat(950,3,vI);
    vHDL->WinLstCellSet(vX, cClmIstDBDS   ,_WinLstDatLineLast);
    vX # 0.0;
    if (FldFloat(950, 4, vI)<>0.0) then vX # FldFloat(950,13,vI) / FldFloat(950,4,vI);
    vHDL->WinLstCellSet(vX, cClmSimDBDS   ,_WinLstDatLineLast);
  END;

  vHdl->wpcurrentint # vPos;

  vHdl->WpAutoUpdate # true;

end;

//=========================================================================
//  Edit
```

```
//
//=========================================================================
sub Edit(aMonat : int);
local begin
  vEvt  : event;
end;
begin
  if (Mode=c_modeview) then RETURN;

  if (aMonat<1) or (aMonat>12) then RETURN;
  if (Rechte[Rgt_CON_Aendern]=n) then RETURN;

  vEvt:obj # $dl.Con.Tabelle;

  Lib_DataList:EvtKeyItem(vEvt, _WinkeyReturn ,0, true);
end;


//=========================================================================
sub SumColumn(
  aDL   : int;
  aClm  : int) : float
local begin
  vI    : int;
  vF    : float;
  vSum  : float;
```

```
end
begin
  FOR vI # 1 loop inc(vI) while (vI<=12) do begin
    WinLstCellGet(aDL, vF, aClm, vI);
    vSum # vSum + vF;
  END;
  RETURN vSum;
end;



//=========================================================================
sub CalcSumProz(
  aDL       : int;
  opt aRep  : logic)
local begin
  vI        : int;
  vF1,vF2   : float;
  vM        : float;
  vX1, vX2  : float;
end;
begin

// Ums * Proz = DB -> Proz = DB / Ums

  vI # 13;
```

```
  // Soll-DB
  vF1 # FldFloat(950, 5, vI);
//debugx('SollDB% = '+anum(FldFloat(950, 11, vI),2)+'/'+anum(vF,2));
  if (vF1<>0.0) then vF1 # FldFloat(950, 11, vI) / vF1 * 100.0;
  vX1 # FldFloat(950, 8, vI);
  FldDef(950, 8, vI, vF1);
  WinLstCellSet(aDl, vF1, cClmSollDBProz, vI);

  // Sim-DB
  vF2 # FldFloat(950, 7, vI);
  if (vF2<>0.0) then vF2 # Rnd(FldFloat(950, 13, vI) / vF2,2) * 100.0;
  vX2 # FldFloat(950, 10, vI);
  FldDef(950,10, vI, vF2);
  WinLstCellSet(aDl, vF2, cClmSimDBProz, vI);

  if (aRep) then begin
    if (vX1<>vF1) or (vX2<>vF2) then begin
      RecRead(950,1,_recLock|_recnoload);
      RekReplace(950);
    end;
  end;

end;

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

```
//=========================================================================
sub AddSum(
  aDL   : int;
  aCol  : int;
  aDiff : float;
  aSbr  : int)
local begin
  vF    : float;
end;
begin
//debug('addsum '+anum(aDiff,2));
  WinLstCellGet(aDL, vF, aCol, 13);
  vF # vF + aDiff;
  WinLstCellSet(aDL, vF, aCol, 13);
  FldDef(950, aSbr, 13, vF);
end;



//=========================================================================
//=========================================================================
sub CalcDS(
  aDL   : int;
  aRow  : int)
local begin
  vF    : float;
  vM    : float;
```

```
end;
begin
  // Soll-DS-DB = DB/Menge
  WinLstCellGet(aDL, vM, cClmSollM, aRow);
  WinLstCellGet(aDL, vF, cClmSollDB, aRow);
  if (vM<>0.0) then vM # Rnd(vF / vM,2);
//debug('SollDBDS = '+anum(vF,2)+'/'+anum(vM,0));
  WinLstCellSet(aDL, vM, cClmSollDBDS, aRow);

  // Soll-DS-Erlös = Erl/Menge
  WinLstCellGet(aDL, vM, cClmSollM, aRow);
  WinLstCellGet(aDL, vF, cClmSollErl, aRow);
//debug('SollErlDS = '+anum(vF,2)+'/'+anum(vM,0));
  if (vM<>0.0) then vM # Rnd(vF / vM,2);
  WinLstCellSet(aDL, vM, cClmSollErlDS, aRow);

  // Sim-DS-DB = DB/Menge
  WinLstCellGet(aDL, vM, cClmSimM, aRow);
  WinLstCellGet(aDL, vF, cClmSimDB, aRow);
  if (vM<>0.0) then vM # Rnd(vF / vM,2);
  WinLstCellSet(aDL, vM, cClmSimDBDS, aRow);

  // Sim-DS-Erlös = Erl/Menge
  WinLstCellGet(aDL, vM, cClmSimM, aRow);
  WinLstCellGet(aDL, vF, cClmSimErl, aRow);
  if (vM<>0.0) then vM # Rnd(vF / vM,2);
```

```
  WinLstCellSet(aDL, vM, cClmSimErlDS, aRow);


end;



//=========================================================================
//=========================================================================
sub EvtLstEditCommit(
  aEvt                  : event;        // Ereignis
  aColumn               : handle;       // Spalte
  aKey                  : int;          // Taste
  aFocusObject          : handle;       // Deskriptor des Fokus-Objektes
) : logic;
local begin
  vRow  : int;
  vOK   : logic;
  vHdl  : int;
  vF    : float;
  vDiff : float;
end;
begin

  vOK # Lib_Datalist:EvtLstEditCommit(aEvt, aColumn, aKey, aFocusObject);
  if (vOK) and (aKey=_winkeyreturn) then begin
    vRow # aEvt:obj->wpcurrentint;
    vHdl # Wininfo(aEvt:obj, _WinLstEditObject);
```

```
    vF # vHdl->wpCaptionFloat;
    case (aColumn->wpname) of
      'clm.SollM' :     begin
          WinLstCellGet(aEvt:obj, vDiff, cClmSollM, vRow);
WinLstCellSet(aEvt:obj, vF, cClmSollM, vRow);
          FldDef(950, 2, vRow, vF);
          vDiff # vF - vDiff;
          AddSum(aEvt:Obj, cClmSollM, vDiff, 2);
          CalcDS(aEvt:Obj, vRow);
          CalcDS(aEvt:Obj, 13);
          CalcSumProz(aEvt:Obj);
      end;

      'clm.SimM' :      begin
          WinLstCellGet(aEvt:obj, vDiff, cClmSimM, vRow);
WinLstCellSet(aEvt:obj, vF, cClmSimM, vRow);
          FldDef(950, 4, vRow, vF);
          vDiff # vF - vDiff;
          AddSum(aEvt:Obj, cClmSimM, vDiff, 4);
          CalcDS(aEvt:Obj, vRow);
          CalcDS(aEvt:Obj, 13);
          CalcSumProz(aEvt:Obj);
      end;

      'clm.SollErl' :   begin
          WinLstCellGet(aEvt:obj, vDiff, cClmSollErl, vRow);
```

```
WinLstCellSet(aEvt:obj, vF, cClmSollErl, vRow);
        FldDef(950, 5, vRow, vF);
        vDiff # vF - vDiff;
        AddSum(aEvt:Obj, cClmSollErl, vDiff, 5);
        CalcDS(aEvt:Obj, vRow);
        CalcDS(aEvt:Obj, 13);
        CalcSumProz(aEvt:Obj);
    end;

    'clm.SimErl' :    begin
        WinLstCellGet(aEvt:obj, vDiff, cClmSimErl, vRow);
WinLstCellSet(aEvt:obj, vF, cClmSimErl, vRow);
        FldDef(950, 7, vRow, vF);
        vDiff # vF - vDiff;
        AddSum(aEvt:Obj, cClmSimErl, vDiff, 7);
        CalcDS(aEvt:Obj, vRow);
        CalcDS(aEvt:Obj, 13);
        CalcSumProz(aEvt:Obj);
    end;

    'clm.SollDBProz' :  begin
        WinLstCellGet(aEvt:obj, vDiff, cClmSollDBProz, vRow);
WinLstCellSet(aEvt:obj, vF, cClmSollDBProz, vRow);
        FldDef(950, 8, vRow, vF);

        // Soll-DB errechnen
```

```
        vF # Rnd(FldFloat(950, 5, vRow) * (vF / 100.0),2);
        FldDef(950, 11, vRow, vF);
        WinLstCellGet(aEvt:obj, vDiff, cClmSollDB, vRow);
        vDiff # vF - vDiff;
        WinLstCellSet(aEvt:obj, vF, cClmSollDB, vRow);
        AddSum(aEvt:Obj, cClmSollDB, vDiff, 11);
        CalcDS(aEvt:Obj, vRow);
        CalcDS(aEvt:Obj, 13);
        CalcSumProz(aEvt:Obj);
    end;

    'clm.SimDBProz' :   begin
        WinLstCellGet(aEvt:obj, vDiff, cClmSimDBProz, vRow);
WinLstCellSet(aEvt:obj, vF, cClmSimDBProz, vRow);
        FldDef(950, 10, vRow, vF);

        // Sim-DB errechnen
        vF # Rnd(FldFloat(950, 7, vRow) * (vF / 100.0),2);
        FldDef(950, 13, vRow, vF);
        WinLstCellGet(aEvt:obj, vDiff, cClmSimDB, vRow);
        vDiff # vF - vDiff;
        WinLstCellSet(aEvt:obj, vF, cClmSimDB, vRow);

        AddSum(aEvt:Obj, cClmSimDB, vDiff, 13);
        CalcDS(aEvt:Obj, vRow);
        CalcDS(aEvt:Obj, 13);
```

```
        CalcSumProz(aEvt:Obj);
      end;
    end;
  end;

  RETURN(vOK);
end;



//=========================================================================
// Pflichtfelder
//             Färbt Pflichtfelder richtig ein
//=========================================================================
sub Pflichtfelder();
begin
  if (Mode<>c_ModeNew) and (Mode<>c_ModeEdit) and
    (Mode<>c_ModeNew2) and (Mode<>c_ModeEdit2) then RETURN;// Pflichtfelder
  // Pflichtfelder
  //Lib_GuiCom:Pflichtfeld($);
end;



//=========================================================================
//  RefreshIfm
//           "Infomasken" refreshen
//=========================================================================
```

```
sub RefreshIfm(
  opt aName     : alpha;
  opt aChanged  : logic;
)
local begin
  Erx   : int;
  vName : alpha;
  vHdl  : handle;
  vHdl2 : handle;
  vTmp  : int;
  vDS   : logic;
end;
begin

  case (Con.Typ) of
    ''  : vName # 'Erlöse';
    'A' : vName # 'Auftragserfassung';
    'B' : vName # 'Bestellerfassung';
    'G' : vName # 'Angebotserfassung';
  end;
  $lbTyp->wpCaption # vName;
  $bt.Refresh->wpDisabled # mode <> c_modeView;
  $bt.Graph->wpDisabled # mode <> c_modeView;

  if (Con.Refreshdatum>1.1.1900) then begin
    $lbDatum->wpCaption # cnvad(Con.Refreshdatum);
```

```
  $lbZEit->wpcaption  # cnvat(Con.Refreshzeit);
end
else begin
  $lbDatum->wpCaption # '';
  $lbZeit->wpcaption # '';
end;

if (Mode=c_ModeView) and (aName='') then begin
  Filter(false);
  RefreshDataList();
end;

if (aName='') or (aName='edCon.Adressnummer') then begin
  Erx # RecLink(100,950,1,0);     // Adresse holen
  if (Erx<=_rLocked) then
    $Lb.Adresse->wpcaption # Adr.Stichwort
  else
    $Lb.Adresse->wpcaption # '';
end;
if (aName='') or (aName='edCon.Vertreternr') then begin
  Erx # RecLink(110,950,2,0);     // Vertreter holen
  if (Erx<=_rLocked) then
    $Lb.Vertreter->wpcaption # Ver.Stichwort
  else
    $Lb.Vertreter->wpcaption # '';
end;
```

```
if (aName='') or (aName='edCon.Warengruppe') then begin
  Erx # RecLink(819,950,3,0);     // WGr holen
  if (Erx<=_rLocked) then
    $Lb.Warengruppe->wpcaption # Wgr.Bezeichnung.L1
  else
    $Lb.Warengruppe->wpcaption # '';
end;
if (aName='') or (aName='edCon.Artikelgruppe') then begin
  Erx # RecLink(826,950,4,0);     // AGr holen
  if (Erx<=_rLocked) then
    $Lb.Artikelgruppe->wpcaption # Agr.Bezeichnung.L1
  else
    $Lb.Artikelgruppe->wpcaption # '';
end;

if (aName='') or (aName='edCon.Auftragsart') then begin
  Erx # RecLink(835,950,7,0);     // Auftragsart holen
  if (Erx<=_rLocked) then
    $lb.AuftragsArt->wpcaption # AAr.Bezeichnung
  else
    $lb.AuftragsArt->wpcaption # '';
end;



if (aName='') or (aName='edCon.Artikelnummer') then begin
  if (Con.Dateinr=250) then begin
```

```
      Erx # RecLink(250,950,5,0);   // Artikel holen
      if (Erx<=_rLocked) then
        $Lb.Artikel->wpcaption # Art.Stichwort
      else
        $Lb.Artikel->wpcaption # '';
      if (aChanged) or ($edCon.Artikelnummer->wpchanged) then
        Con.MEH # Art.MEH;
      end
    else begin
      $Lb.Artikel->wpcaption # '';
      Con.MEH # 't';
    end;
//    $lb.MEH1->winupdate(_WinUpdFld2Obj);
  end;
  if (aName='') then begin
    if (Con.DateiNr=200) then begin
      $cb.Material->wpCheckState # _WinStateChkchecked;
      $cb.Artikel->wpCheckState # _WinStateChkUnchecked;
      $lbCon.Artikelgruppe->wpvisible # false;
      $edCon.Artikelgruppe->wpvisible # false;
      $bt.AGr->wpvisible              # false;
      $lb.Artikelgruppe->wpvisible    # false;
      $lbCon.Artikelnummer->wpcaption # Translate('Güte');
    end
    else begin
      $cb.Material->wpCheckState # _WinStateChkUnchecked;
```

```
    $cb.Artikel->wpCheckState # _WinStateChkchecked;
    $lbCon.Artikelgruppe->wpvisible # true;
    $edCon.Artikelgruppe->wpvisible # true;
    $bt.AGr->wpvisible              # true;
    $lb.Artikelgruppe->wpvisible    # true;
    $lbCon.Artikelnummer->wpcaption # Translate('Artikelnummer');
  end;
end;

vHdl # $dl.Con.Tabelle;
if (vHdl->wpcustom='Umsatz') then begin
  vName # Translate('Umsatz');
  vDS # y;
end
else if (vHdl->wpcustom='DB') then begin
  vName # Translate('DB');
  vDS # y;
end
else begin
  vName # Translate('Menge');
  vDS # n;
end;

// 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:PflichtfelderEinfaerben();
end;



//=========================================================================
//  RecInit
//          Init für Änderung und Neuanlage
//=========================================================================
sub RecInit()
local begin
  vDat  : date;
  vTmp  : int;
end;
begin

  // Felder Disablen durch:
```

```
//Lib_GuiCom:Disable($...);
// Focus setzen auf Feld:
if (Mode=c_modeNew) then begin
  Con.Dateinr # 250;
  Con.MEH     # 't';
  Con.Typ     # '_';

  RefreshDataList();
  vDat # today;
  Con.Jahr # vDat->vpyear;
  $edCon.Jahr->WinFocusSet(true);

  vTmp # WinDialog('Con.Typ.Auswahl',_WinDialogCenter,gMDI);
  IF (vTmp= _WinIdClose) or (gSelected=0) then begin
    App_Main_Sub:StopModeNew();   // Neuerfassung abbrechen
    RETURN;
  end;

  gMDI->winfocusset(true);
  vTmp # gSelected;
  gSelected # 0;
  case vTmp of
    450 :   Con.Typ # '';
    401 :   Con.Typ # 'A';
    501 :   Con.Typ # 'B';
    1401 :  Con.Typ # 'G';
```

```
    end;
  end;


end;



//=========================================================================
//  RecSave
//          vor Speicherung
//=========================================================================
sub RecSave() : logic;
local begin
  Erx   : int;
  vM    : int;
  vI    : int;
  vX    : float;
end;
begin
  // dynamische Pflichtfelder überprüfen
  if ( !Lib_Pflichtfelder:PflichtfelderPruefenVorSpeichern() ) then
    RETURN false;

  // logische Prüfung
  // Nummernvergabe
  // Satz zurückspeichern & protokolieren
  if (Mode=c_ModeEdit) then begin
```

```
    Erx # RekReplace(gFile,_recUnlock,'MAN');
    if (Erx<>_rOk) then begin
      Msg(001000+Erx,gTitle,0,0,0);
      RETURN False;
    end;
    PtD_Main:Compare(gFile);
  end
  else begin
    Con.Refreshdatum  # today;
    Con.Refreshzeit   # now;
    Erx # RekInsert(gFile,0,'MAN');
    if (Erx<>_rOk) then begin
      Msg(001000+Erx,gTitle,0,0,0);
      RETURN False;
    end;
  end;

  RETURN true;  // Speichern erfolgreich
end;


//=========================================================================
//  RecCleanup
//              Aufräumen bei "Cancel"
//=========================================================================
sub RecCleanup() : logic
```

```
begin
  $dl.Con.Tabelle->wpdisabled # n;
  RETURN true;
end;



//=========================================================================
//  RecDel
//          Satz soll gelöscht werden
//=========================================================================
sub RecDel()
begin
  // Diesen Eintrag wirklich löschen?
  if (Msg(000001,'',_WinIcoQuestion,_WinDialogYesNo,2)=_WinIdYes) then begin
    RekDelete(gFile,0,'MAN');
    RefreshDataList();
  end;
end;



//=========================================================================
//  EvtFocusInit
//          Fokus auf Objekt neu gesetzt
//=========================================================================
sub EvtFocusInit (
  aEvt                  : event;        // Ereignis
```

```
  aFocusObject          : int          // vorheriges Objekt
) : logic
begin

  if (gMDI<>w_Mdi) then  gMDI # w_MDI;  // MDIBUGFIX 03.06.2014

  if (aEvt:Obj->wpname='Edit') then begin
    if (gZLList->wpcustom='->DATALIST') then begin
      $dl.Con.Tabelle->winfocusset(true);
      gZLList->wpcustom # '2DATALIST';
      RETURN true;
    end;
    if (gZLList->wpcustom='2DATALIST') then begin
      $dl.Con.Tabelle->winfocusset(true);
      gZLList->wpcustom # '3DATALIST';
      RETURN true;
    end;
    if (gZLList->wpcustom='3DATALIST') then begin
      $dl.Con.Tabelle->winfocusset(true);
      gZLList->wpcustom # '';
    end;
    RETURN App_Main:EvtFocusInit(aEvt, aFocusObject);
  end;

/***
  if (aEvt:Obj->wpname='jump') then begin
```

```
    case (aEvt:Obj->wpcustom) of
      'Page1Start' : begin
        if (aFocusObject<>0) then aFocusObject->winfocusset(false);
        $NB.Main->wpcurrent # 'NB.Page1';
        $...->winfocusset(false)
        end;
      'Page1E' : begin
        if (aFocusObject<>0) then aFocusObject->winfocusset(false);
        $NB.Main->wpcurrent # 'NB.Page1';
        $...->winfocusset(false);
        end;
    end;
    RETURN true;
  end;
***/

  // Auswahlfelder aktivieren
  if (Lib_Pflichtfelder:TypAuswahlFeld(aEvt:Obj)<>'') then
    Lib_GuiCom:AuswahlEnable(aEvt:Obj);
  else
    Lib_GuiCom:AuswahlDisable(aEvt:Obj);

end;


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

```
//  EvtFocusTerm
//          Fokus vom Objekt wegbewegen
//=========================================================================
sub EvtFocusTerm (
  aEvt                : event;        // Ereignis
  aFocusObject        : int           // neu zu fokusierendes Objekt
) : logic
local begin
  vM    : int;
  vX    : float;
end;
begin

  // logische Prüfung von Verknüpfungen
  RefreshIfm(aEvt:Obj->wpName);

  RETURN true;
end;



//=========================================================================
//  Auswahl
//          Auswahliste öffnen
//=========================================================================
sub Auswahl(
  aBereich : alpha;
```

```
)
local begin
  vA    : alpha;
end;

begin

  case aBereich of
    'Adresse' : begin
      RecBufClear(100);         // ZIELBUFFER LEEREN
      gMDI # Lib_GuiCom:AddChildWindow(gMDI, 'Adr.Verwaltung',here+':AusAdresse');
      Lib_GuiCom:RunChildWindow(gMDI);
    end;


    'Vertreter' : begin
      RecBufClear(110);         // ZIELBUFFER LEEREN
      gMDI # Lib_GuiCom:AddChildWindow(gMDI, 'Ver.Verwaltung',here+':AusVertreter');
      Lib_GuiCom:RunChildWindow(gMDI);
    end;


    'WGr' : begin
      RecBufClear(819);         // ZIELBUFFER LEEREN
      gMDI # Lib_GuiCom:AddChildWindow(gMDI, 'WGr.Verwaltung',here+':AusWGr');
      Lib_GuiCom:RunChildWindow(gMDI);
```

```
end;



'AGr' : begin
  RecBufClear(826);         // ZIELBUFFER LEEREN
  gMDI # Lib_GuiCom:AddChildWindow(gMDI, 'AGr.Verwaltung',here+':AusAGr');
  Lib_GuiCom:RunChildWindow(gMDI);
end;

'Auftragsart' : begin
  RecBufClear(835);         // ZIELBUFFER LEEREN
  gMDI # Lib_GuiCom:AddChildWindow(gMDI, 'AAr.Verwaltung',here+':AusAuftragsart');
  Lib_GuiCom:RunChildWindow(gMDI);
end;

'Artikel' : begin
  if (Con.Dateinr=250) then begin
    RecBufClear(250);         // ZIELBUFFER LEEREN
    gMDI # Lib_GuiCom:AddChildWindow(gMDI, 'Art.Verwaltung',here+':AusArtikel');
    end
  else begin
    RecBufClear(832);         // ZIELBUFFER LEEREN
    gMDI # Lib_GuiCom:AddChildWindow(gMDI, 'MQu.Verwaltung',here+':AusGuete');
  end;
  Lib_GuiCom:RunChildWindow(gMDI);
end;
```

```
/*
    'Kostenstelle' : begin
      RecBufClear(846);         // ZIELBUFFER LEEREN
      gMDI # Lib_GuiCom:AddChildWindow(gMDI, 'KSt.Verwaltung',here+':AusKSt');
      Lib_GuiCom:RunChildWindow(gMDI);
    end;
*/
  end;  // ...case

end;



//=========================================================================
//  AusAdresse
//
//=========================================================================
sub AusAdresse()
local begin
  vTmp  : int;
end;
begin
  if (gSelected<>0) then begin
    RecRead(100,0,_RecId,gSelected);
    gSelected # 0;
    // Feldübernahme
```

```
    Con.Adressnummer # Adr.Nummer;


    vTmp # WinFocusget();   // LastFocus-Feld refreshen
    if (vTmp<>0) then vTmp->Winupdate(_WinUpdFld2Obj);
  end;
  // Focus auf Editfeld setzen:
  $edCon.Adressnummer->Winfocusset(false);
  // ggf. Labels refreshen
  // RefreshIfm('edxxx.xxxxxxx',y);
end;



//=========================================================================
//  AusVertreter
//
//=========================================================================
sub AusVertreter();
local begin
  vTmp  : int;
end;
begin
  if (gSelected<>0) then begin
    RecRead(110,0,_RecId,gSelected);
    gSelected # 0;
    // Feldübernahme
    Con.Vertreternr # Ver.Nummer;
```

```
    vTmp # WinFocusget();   // LastFocus-Feld refreshen
    if (vTmp<>0) then vTmp->Winupdate(_WinUpdFld2Obj);
  end;
  // Focus auf Editfeld setzen:
  $edCon.Vertreternr->Winfocusset(false);
  // ggf. Labels refreshen
  // RefreshIfm('edxxx.xxxxxxx',y);
end;



//=========================================================================
//  AusAuftragsart
//
//=========================================================================
sub AusAuftragsart()
local begin
  vTmp  : int;
end;
begin
  if (gSelected<>0) then begin
    RecRead(835,0,_RecId,gSelected);
    gSelected # 0;
    // Feldübernahme
    Con.Auftragsart # AAr.Nummer;
```

```
    vTmp # WinFocusget();   // LastFocus-Feld refreshen
    if (vTmp<>0) then vTmp->Winupdate(_WinUpdFld2Obj);
  end;
  // Focus auf Editfeld setzen:
  $edCon.Auftragsart->Winfocusset(false);
  // ggf. Labels refreshen
  // RefreshIfm('edxxx.xxxxxxx',y);
end;



//=========================================================================
//  AusWGr
//
//=========================================================================
sub AusWGr()
local begin
  vTmp  : int;
end;
begin
  if (gSelected<>0) then begin
    RecRead(819,0,_RecId,gSelected);
    gSelected # 0;
    // Feldübernahme
    Con.Warengruppe # WGr.Nummer;

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

```
    if (vTmp<>0) then vTmp->Winupdate(_WinUpdFld2Obj);
  end;
  // Focus auf Editfeld setzen:
  $edCon.Warengruppe->Winfocusset(false);
  // ggf. Labels refreshen
  // RefreshIfm('edxxx.xxxxxxx',y);
end;



//=========================================================================
//  AusAGr
//
//=========================================================================
sub AusAGr()
local begin
  vTmp  : int;
end;
begin
  if (gSelected<>0) then begin
    RecRead(826,0,_RecId,gSelected);
    gSelected # 0;
    // Feldübernahme
    Con.Artikelgruppe # Agr.Nummer;

    vTmp # WinFocusget();   // LastFocus-Feld refreshen
    if (vTmp<>0) then vTmp->Winupdate(_WinUpdFld2Obj);
```

```
  end;
  // Focus auf Editfeld setzen:
  $edCon.Artikelgruppe->Winfocusset(false);
  // ggf. Labels refreshen
  // RefreshIfm('edxxx.xxxxxxx',y);
end;




//=========================================================================
//  AusArtikel
//
//=========================================================================
sub AusArtikel()
local begin
  vTmp  : int;
end;
begin
  if (gSelected<>0) then begin
    RecRead(250,0,_RecId,gSelected);
    gSelected # 0;
    // Feldübernahme
    Con.Artikelnummer # Art.Nummer;
    Con.MEH           # Art.MEH;
    vTmp # WinFocusget();   // LastFocus-Feld refreshen
    if (vTmp<>0) then vTmp->Winupdate(_WinUpdFld2Obj);
```

```
  end;
  // Focus auf Editfeld setzen:
  $edCon.Artikelnummer->Winfocusset(false);
  // ggf. Labels refreshen
  RefreshIfm('edCon.Artikelnummer',y);
end;



//=========================================================================
//  AusGuete
//
//=========================================================================
sub AusGuete()
local begin
  vTmp  : int;
end;
begin
  if (gSelected<>0) then begin
   RecRead(832,0,_RecId,gSelected);
    gSelected # 0;
    // Feldübernahme
    if (MQu.ErsetzenDurch<>'') then
      Con.Artikelnummer # MQu.ErsetzenDurch
    else if ("MQu.Güte1"<>'') then
      Con.Artikelnummer # "MQu.Güte1"
    else
```

```
      Con.Artikelnummer # "MQu.Güte2";


    vTmp # WinFocusget();   // LastFocus-Feld refreshen
    if (vTmp<>0) then vTmp->Winupdate(_WinUpdFld2Obj);
  end;
  // Focus auf Editfeld setzen:
  $edCon.Artikelnummer->Winfocusset(false);
  // ggf. Labels refreshen
  // RefreshIfm('edxxx.xxxxxxx',y);
end;



//=========================================================================
//  AusKSt
//
//=========================================================================
sub AusKSt()
local begin
  vTmp  : int;
end;
begin
  if (gSelected<>0) then begin
    RecRead(846,0,_RecId,gSelected);
    gSelected # 0;
    // Feldübernahme
    Con.Kostenstelle # KSt.Nummer;
```

```
    vTmp # WinFocusget();   // LastFocus-Feld refreshen
    if (vTmp<>0) then vTmp->Winupdate(_WinUpdFld2Obj);
  end;
  // Focus auf Editfeld setzen:
  $edCon.Kostenstelle->Winfocusset(false);
  // ggf. Labels refreshen
  // RefreshIfm('edxxx.xxxxxxx',y);
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);

  // Button & Menßs sperren
  vHdl # gMenu->WinSearch('Mnu.Mark.Sel');
  if (vHdl <> 0) then
```

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


vHdl # gMdi->WinSearch('New');
if (vHdl <> 0) then
  vHdl->wpDisabled # (vHdl->wpDisabled) or (Rechte[Rgt_CON_Anlegen]=n);
vHdl # gMenu->WinSearch('Mnu.New');
if (vHdl <> 0) then
  vHdl->wpDisabled # (vHdl->wpDisabled) or (Rechte[Rgt_CON_Anlegen]=n);


vHdl # gMdi->WinSearch('Edit');
if (vHdl <> 0) then
  vHdl->wpDisabled # (vHdl->wpDisabled) or (Rechte[Rgt_CON_Aendern]=n);
vHdl # gMenu->WinSearch('Mnu.Edit');
if (vHdl <> 0) then
  vHdl->wpDisabled # (vHdl->wpDisabled) or (Rechte[Rgt_CON_Aendern]=n);


vHdl # gMdi->WinSearch('Delete');
if (vHdl <> 0) then
  vHdl->wpDisabled # (vHdl->wpDisabled) or (Rechte[Rgt_CON_Loeschen]=n);
vHdl # gMenu->WinSearch('Mnu.Delete');
if (vHdl <> 0) then
  vHdl->wpDisabled # (vHdl->wpDisabled) or (Rechte[Rgt_CON_Loeschen]=n);


vHdl # gMenu->WinSearch('Mnu.Recalc');
if (vHdl <> 0) then
  vHdl->wpDisabled # (mode<>c_ModeList) or (Rechte[Rgt_CON_Aendern]=n);
```

```
  vHdl # gMenu->WinSearch('Mnu.Generieren');
  if (vHdl <> 0) then
    vHdl->wpDisabled # (mode<>c_ModeList) or (Rechte[Rgt_CON_Aendern]=n);


  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.Recalc' : begin
    Con_Data:Recalc(true);
  end;

  'Mnu.Generieren' : begin
    Con_Data:Generieren();
  end;

  'Mnu.GenerierenVorgaben' : begin
    Con_Data:GenerierenVorgaben();
  end;


  'Mnu.Protokoll' : begin
    PtD_Main:View(gFile);
  end;


  'Mnu.Mark.SetField' : begin
    Lib_Mark:SetField(gFile);
  end;
```

```
    'Mnu.Mark.Sel' : begin
      Con_Mark_Sel();  // Aufruf für Selektionsmaske
    end;




  end; // ...case



end;



//=========================================================================
//  EvtClicked
//              Button gedrückt
//=========================================================================
sub EvtClicked (
  aEvt                  : event;        // Ereignis
) : logic
begin

  case (aEvt:Obj->wpName) of
    'btAuf'   : gSelected # 401;
    'btBest'  : gSelected # 501;
    'btAng'   : gSelected # 1401;
    'btErl'   : gSelected # 450;
```

```
    'bt.Graph'  : Mdi_Con_Graph:SpawnGraph(RecBufDefault(950),'MENGE');


    'bt.Refresh'     : begin
      Con_Data:Recalc(false);
      REfreshifm();
    end;
  end;

  if (Mode=c_ModeView) then RETURN true;

  case (aEvt:Obj->wpName) of
    'bt.Adresse'      : Auswahl('Adresse');
    'bt.Vertreter'    : Auswahl('Vertreter');
    'bt.WGr'          : Auswahl('WGr');
    'bt.AGr'          : Auswahl('AGr');
    'bt.Artikel'      : Auswahl('Artikel');
    'bt.Auftragsart'  : Auswahl('Auftragsart');
  end;  // ...case

  if (gSelected<>0) then begin
    $Con.Typ.Auswahl->Winclose();
  end;

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.Artikel') and (aEvt:OBj->wpCheckState=_WinStateChkChecked) then begin
//    $cb.Material->wpCheckState # _WinStateChkUnchecked;
    Con.Dateinr # 250;
  end;
  if (aEvt:Obj->wpname='cb.Material') and (aEvt:OBj->wpCheckState=_WinStateChkChecked) then begin
//    $cb.Artikel->wpCheckState # _WinStateChkUnchecked;
    Con.Dateinr # 200;
  end;

  if (aEvt:Obj->wpname='cb.Menge') or (aEvt:Obj->wpname='cb.Erloes') or (aEvt:Obj->wpname='cb.DB') or
    (aEvt:Obj->wpname='cb.Ist') or (aEvt:Obj->wpname='cb.Soll') or (aEvt:Obj->wpname='cb.Sim') or (aEvt:
    Filter(true);
    RETURN true;
  end;
```

```
  RefreshIFM('etwas');


  RETURN (true);
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;
end;
begin

  Erx # 100;
  if (Con.Adressnummer<>0) then
    Erx # RecLink(100,950,1,0);     // Adresse holen
  if (Erx>_rLocked) then RecBufClear(100);

  Erx # 100;
  if (Con.VertreterNr<>0) then
    Erx # RecLink(110,950,2,0);     // Vertreter holen
  if (Erx>_rLocked) then RecBufClear(110);

  Erx # 100;
  if (Con.Auftragsart<>0) then
    Erx # RecLink(835,950,7,0);     // Auftragsart holen
  if (Erx>_rLocked) then RecBufClear(835);

  Erx # 100;
  if (Con.Warengruppe<>0) then
```

```
    Erx # RecLink(819,950,3,0);     // WGr holen
  if (Erx>_rLocked) then RecBufClear(819);

  Erx # 100;
  if (Con.Artikelgruppe<>0) then
    Erx # RecLink(826,950,4,0);     // AGr holen
  if (Erx>_rLocked) then RecBufClear(826);

  Erx # 100;
  if (Con.Artikelnummer<>'') and (Con.Dateinr=250) then
      Erx # RecLink(250,950,5,0);   // Artikel holen
  if (Erx>_rLocked) then RecBufClear(250);

  GV.Alpha.01 # '';
  case (Con.Typ) of
    ''  : Gv.Alpha.01 # Translate('Erlöse');
    'A' : Gv.Alpha.01 # Translate('Auftragserfassung');
    'B' : Gv.Alpha.01 # Translate('Bestellerfassung');
    'G' : Gv.Alpha.01 # Translate('Angebotserfassung');
  end;

end;


//=========================================================================
//  EvtLstSelect
```

```
//              Zeilenauswahl von RecList/DataList
//=========================================================================
sub EvtLstSelect(
  aEvt                : event;       // Ereignis
  aRecID               : int;
) : logic
local begin
  vM  : int;
end;
begin
  RETURN true;
end;



//=========================================================================
// EvtClose
//         Schliessen eines Fensters
//=========================================================================
sub EvtClose(
  aEvt                : event;       // Ereignis
): logic
begin
  Lib_GuiCom:RememberList($dl.Con.Tabelle);

  Lib_GuiCom2:CloseAllChilds(aEvt:obj);
```

```
  RETURN true;
end;



//=========================================================================
//  EvtMouseItem
//
//=========================================================================
sub EvtMouseItem(
  aEvt                : event;    // Ereignis
  aButton             : int;      // Maustaste
  aHitTest            : int;      // Hittest-Code
  aItem               : int;      // Spalte oder Gantt-Intervall
  aID                 : int;      // RecID bei RecList / Zelle bei GanttGraph
) : logic;
begin

  if ((aButton & _WinMouseDouble)>0) and
    ((aButton & _WinMouseLeft)>0) then begin
    Edit($dl.Con.Tabelle->wpcurrentint);
  end;

  RETURN(true);
end;
```

```
//=========================================================================
//  EvtKeyItem
//
//=========================================================================
sub EvtKeyItem(
  aEvt                : event;    // Ereignis
  aKey                : int;      // Taste
  aID                 : int;      // RecID bei RecList, Node-Deskriptor bei TreeView
) : logic;
begin
  if (aKey=_WinKeyReturn) then begin
    Edit($dl.Con.Tabelle->wpcurrentint);
  end;

  RETURN(true);
end;



//=========================================================================
//=========================================================================
sub EvtPosChanged(
 aEvt         : event;    // Ereignis
 aRect        : rect;     // Größe des Fensters
 aClientSize  : point;    // Größe des Client-Bereichs
 aFlags       : int       // Aktion
) : logic
```

```
local begin
  vRect     : rect;
  vHdl      : int;
end
begin

  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;

  if (aFlags & _WinPosSized != 0) then begin
    if (gZLList<>0) then vHdl # gZLList;
    else if (gDataList<>0) then vHdl # gDataList
    else RETURN true;

    vRect           # vHdl->wpArea;
    vRect:right     # aRect:right-aRect:left-4;
    vRect:bottom    # aRect:bottom-aRect:Top-28 - w_QBHeight;
    vHdl->wparea # vRect;
```

```
  end;

  vHdl # Winsearch(gMDI,'dl.Con.Tabelle');
  if (vHdl<>0) then begin
    vRect           # vHdl->wpArea;
    vRect:right     # aRect:right-aRect:left-2;
    vRect:bottom    # aRect:bottom-aRect:Top-28;
    vHdl->wparea    # vRect;
  end;

 RETURN (true);
end;


sub JumpTo(
  aName : alpha;
  aBuf  : int);
begin

  if ((aName =^ 'edCon.Adressnummer') AND (aBuf->Con.Adressnummer<>0)) then begin
    RekLink(100,950,1,0);   // Adresse holen
    Lib_Guicom2:JumpToWindow('Adr.Verwaltung');
    RETURN;
  end;

  if ((aName =^ 'edCon.Auftragsart') AND (aBuf->Con.Auftragsart<>0)) then begin
```

```
  RekLink(110,950,7,0);   // Auftragsart holen
  Lib_Guicom2:JumpToWindow('Adr.Verwaltung');
  RETURN;
end;

if ((aName =^ 'edCon.Vertreternr') AND (aBuf->Con.Vertreternr<>0)) then begin
  RekLink(100,950,2,0);   // Vertreter holen
  Lib_Guicom2:JumpToWindow('Ver.Verwaltung');
  RETURN;
end;

if ((aName =^ 'edCon.Warengruppe') AND (aBuf->Con.Warengruppe<>0)) then begin
  RekLink(819,950,3,0);   // Warengruppe holen
  Lib_Guicom2:JumpToWindow('WGr.Verwaltung');
  RETURN;
end;

if ((aName =^ 'edCon.Artikelnummer') AND (aBuf->Con.Artikelnummer<>'')) then begin
  RekLink(250,950,5,0);   // Artieklnummer holen
  Lib_Guicom2:JumpToWindow('Art.Verwaltung');
  RETURN;
end;

if ((aName =^ 'edCon.Artikelgruppe') AND (aBuf->Con.Artikelgruppe<>0)) then begin
  RekLink(826,950,4,0);   // Artikelgruppe holen
  Lib_Guicom2:JumpToWindow('AGr.Verwaltung');
```

```
    RETURN;
  end;


end;


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