

@A+

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

//

// Prozedur Auf_P_Main

// OHNE E_R_G

// Info

//

//

// 19.01.2004 AI Erstellung der Prozedur

// 13.03.2009 TM Intrastat-Auswahl wahlweise selektiert oder komplett

// 27.07.2009 ST Fehlerkorrektur: Positionsanlage sperrte RID und RAD

// 06.08.2009 AI Felderlogik Abruf/Re.Nr.

// 03.09.2009 MS Beim Anhängen einer Abrufpos ist das Feld Abruf-Best.Nr. nun verfuegbar

// 10.09.2009 MS Bedingung fuer Artikel-Tausch korrigiert

// 24.11.2009 MS Nach der Auswahl der Lieferadresse poppt bei Anschriften > 1 das Anschriftsfenster au

// 25.01.2010 AI Kopfbestellnummer nicht mehr in Positionen schreiben

// 13.04.2010 MS Validierung Abruf-Best Nr.

// 21.06.2010 ST Druck des BAs aus Lohnauftrag

// 18.10.2011 AI Zeile 2920: Auf.P.Menge setzen aus kg bzw. Stk bei Material

// 22.03.2012 AI NEU: Adr-VPG mit Erzeuger

// 18.04.2012 MS Feld Rechnungsnummer fuer GUT/BEL beim Pos. ANHAENGEN aktivieren

// 25.04.2012 MS Protokollrecht hinzugefuegt

// 30.05.2012 AI NEU: Rechnungsanschrift

// 25.06.2012 AI Warengruppe aus Material übernehmen

// 05.07.2012 TM Materialbereich Mengeneinheit als Pflichtfeld

// 19.07.2012 ST Druck: DMS-Deckblatt hinzugefügt

// 30.08.2012 ST Anbieterstellung auf direktes Material trägt bei 209er auch Artikelnr in Aktion ein (132

// 25.09.2012 TM AuftragsAnfrage eingesetzt

// 28.11.2012 ST Kein Automatischer AB Druck bei Setting = ja und Lohnaufträgen

// 17.01.2013 ST EvtPosChanged: Höhe für Infoleiste in Verbindung mit Quickbuttons angepasst Prj. 14

// 25.02.2013 AI "AusLFS" neues Setting "Set.LFS.SofortDLFAYN"

// 27.02.2013 ST "Vorgang Fahrauftrag" für markierte Auftragspositionen Prj 1449/5

// 06.03.2013 ST Formular "Gelangensbestätigung" hinzugefügt

// 21.03.2013 ST Avisdruck in Auf_P_Subst:DruckAvis() verschoben

// 13.05.2013 AI Vertretersperre eingebaut

// 07.03.2013 ST Customfelder hinzugefügt

// 21.08.2013 AH Bugfix: Preisfinden holt vorher Kundenadresse

// 11.02.2014 AH SumEKGesamtpreis auch bei "RecSave"

// 12.02.2014 ST Artikelnummer auch "einfügbar" gemacht Prj. 1304/237

// 11.04.2014 AH Angebote können immer uu Rahmen bzw. Vertrag verändert werden (Prj.1488/43)

// 31.07.2014 AH Feldsperre Abruf, Liefervertrag, Gültigkeit korrigiert

// 25.08.2014 TM EvtMenuCommand temporär erweitert um "Mnu.Mark.SetField" zur Serienänderung Re

// 12.01.2015 AH DFakt.ArtC kann auch Artikel ohne Bestandsführung zuordnen

// 13.01.2015 AH MatZ.Mat filtert bei MatMix

// 22.01.2015 AH Set.Auf.ArtFilter wieder gängig gemacht

// 26.01.2015 AH Buffix: Label für Aufpreise

// 07.02.2015 AH Neu: VorlageBAG + EinsatzVPG

// 11.02.2015 AH Edit: std. Lieferanschriften aus Adresse

// 01.03.2015 AH Neu: AufNummer-Filter

// 10.04.2015 AH Edit: Stücklistensummen nicht nur bei c_Art_Cut, sondern IMMER bei Stückliste

// 21.05.2015 AH ZL.Erfassung nur refreshen, wenn mind. ein Pos.Satz existiert

// 14.10.2015 ST "Sub start" hinzugefügt

// 05.11.2015 AH Schnellzugriff auf aus Vorlage erzeugten BAs

// 19.01.2015 ST Anker "Auf.P.AusMaterial.Post" hinzugefügt

// 23.02.2016 AH Erweiterung Artikelmaske

// 17.03.2016 ST Fixierte Zahlungsbedingung hinzugefügt

// 24.03.2016 AH Einsatzmenge bei 250 wird anderes berechnet und ist gesperrt

// 17.05.2016 AH Sprungreihenfolge Artikelmaske

// 08.06.2016 AH neues Feld "Güte" bei Artikel

// 14.07.2016 AH Kopfrabatte werden mit in Position gerechnet

// 08.11.2016 AH PAbufr

// 22.12.2016 AH MATZ.Mat nimmt Markierte

// 31.07.2017 AH MATZ.Mat prüft Markierte auf "passend"

// 26.01.2018 AH AnalyseErweitert

// 21.03.2018 AH Kopie von Lohnauftrag kann BA kopieren

// 20.06.2018 AH AFX: "Auf.P.AusText"

// 25.07.2018 AH Fix: F9 auf Reschnungspos (AbrufPos)

// 13.02.2019 AH Neu: LFS-Verbuchen-Setting kann auch OHNE Druck nur "B"uchen sagen

// 26.07.2019 AH Fix: Statistikverbuchung

// 25.11.2019 AH AFX: "Auf.P.RefreshIfm.Post"

// 07.09.2020 AH Kurzinfo

// 20.01.2021 AH VorlageAuf

// 10.11.2021 AH Änderungen der Zieladresse refreshen den Versandpool

// 04.04.2022 AH ERX

// 19.05.2022 AH Fix für Abrufmengenedit (HOW)

// 23.05.2022 TM Sicherheitsabfrage vor Wandlung in Liefervertrag

// 13.07.2022 TM Sub EvtTimer: 'LFS'- Timer speziell für Brockhaus deaktiviert

// 14.07.2022 HA Quick Jump

```
// 13.09.2022 ST Neu: Mnu.ChangePosNr: Angebotspositionen ändern

// 2023-01-17 AH Filter auf User

// 2023-03-03 ST Fix: Gütenauswahl per F9 korrigiert

// 2023-08-14 AH "LiB.SperreNeuYN"

//

// Subprozeduren

// SUB EvtInit(aEvt : event) : logic

// SUB Etikettendaten();

// SUB Skizzendaten();

// SUB RefreshIlfm(optaName : alpha; optaChanged : logic)

// SUB Reclnit(opt aBehalten : logic; opt aMitAufp : logic; opt aMitKalk : logic)

// SUB RecSave() : logic;

// SUB RecCleanup() : logic;

// SUB RecDel()

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

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

// SUB Auswahl(aBereich : alpha)

// SUB AusEinzelObfOben()

// SUB AusEinzelObfUnten()

// SUB AusAktion()

// SUB AusText1()

// SUB AusLFS()

// SUB AusMatzMat()

// SUB AusMatzMatGuBe()

// SUB AusMaterial()

// SUB AusDFaktMat()
```

```
// SUB AusMatzArtC()

// SUB AusDFaktArtC()

// SUB AusKundenArtNr()

// SUB AusArtikelnummer()

// SUB AusArtikelnummer_Mat()

// SUB AusStruktur()

// SUB AusKundenMatArtNr()

// SUB AusStueckliste()

// SUB AusKunde()

// SUB AusLieferadresse()

// SUB AusLieferanschrift()

// SUB AusVerbraucher()

// SUB AusRechnungsempf()

// SUB AusRechnungsanschr()

// SUB AusAnsprechpartner()

// SUB AusLand()

// SUB AusBDS()

// SUB AusWaehrung()

// SUB AusLieferbed()

// SUB AusZahlungsbed()

// SUB AusSteuerschluessel()

// SUB AusVersandart()

// SUB AusSachbearbeiter()

// SUB AusVertreter1()

// SUB AusVertreter2()

// SUB AusAuftragsArt()
```

```
// SUB AusSkizze();  
  
// SUB AusVerpackung();  
  
// SUB AusKopftext();  
  
// SUB AusFusstext();  
  
// SUB AusKopftextAdd();  
  
// SUB AusFusstextAdd();  
  
// SUB AusTextAdd();  
  
// SUB AusWarengruppe()  
  
// SUB AusProjekt()  
  
// SUB AusIntrastat()  
  
// SUB AusProjektSL()  
  
// SUB AusGuete()  
  
// SUB AusGuetenstufe()  
  
// SUB AusAFOben()  
  
// SUB AusAFUnten()  
  
// SUB AusZeugnis()  
  
// SUB AusErzeuger()  
  
// SUB AusVerwiegungsart()  
  
// SUB AusZwischenlage()  
  
// SUB AusUnterlage()  
  
// SUB AusUmverpackung()  
  
// SUB AusEtikettentyp()  
  
// SUB AusEtikettentyp2()  
  
// SUB AusKopfaufpreise()  
  
// SUB AusAufpreise()  
  
// SUB AusKalkulation()
```

```

// SUB AusAbruf()

// SUB AusErloes()

// SUB AusErloesKonto()

// SUB AusPreis()

// SUB RefreshMode(optaNoRefresh : logic);

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

// SUB EvtClicked(aEvt : event) : logic

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

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

// SUB EvtChanged(aEvt : event) : logic

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

// SUB EvtLstRecControl(aEvt : event; aRecId : int) : logic

// SUB EvtClose(aEvt : event) : logic

// SUB AufPTextSave()

// SUB AufPTextLoad()

// SUB AFObenRecCtrl(aEvt : event; aRecId : int) : logic;

// SUB AFUntenRecCtrl(aEvt : event; aRecId : int) : logic;

// SUB Pflichtfelder();

// SUB EvtTimer(aEvt : event; aTimerId : int) : logic

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

//

//=====

@I:Def_Global

@I:Def_Rights

@I:Def_Aktionen

@I:Def_BAG

```

define begin

cTitle : 'Auftragspositionen'

cFile : 401

cMenuName : 'Auf.P.Bearbeiten'

cPrefix : 'Auf_P'

// cZList : \$ZL.AufPositionen

cZList : 'ZL.AufPositionen'

cKey : 1

cDialog2 : \$Auf.Verwaltung

cZList2 : \$ZL.Auftraege

cZLName2 : 'ZL.Auftraege'

cDialog : 'Auf.P.Verwaltung'

cRecht : Rgt_Auftrag

cMdiVar : gMDIAuf

cPZeile : '('+__PROC__+':'+aint(__LINE__)+')

end;

declare AufPTextLoad();

declare AufPTextSave();

declare Pflichtfelder();

declare RefreshMode(opt aNoRefresh : logic);

declare Auswahl(aBereich : alpha)

declare AusMaterial()

//=====

// Start

// Startet die Verwaltung

//=====

sub Start(

opt aRecId : int;

opt aAufNr : int;

opt aAufPos : int;

opt aView : logic) : logic;

local begin

Erx : int;

end;

begin

if (aRecId=0) AND (aAufNr<>0) AND (aAufPos <> 0) then begin

Auf.P.Nummer # aAufNr;

Auf.P.Position # aAufPos;

Erx # RecRead(401,1,0);

if (Erx>_rLocked) then RETURN false;

aRecId # RecInfo(401,_recId);

end;

App_Main_Sub:StartVerwaltung(cDialog, cRecht, var cMDIvar, aRecId, aView);

if (aView) then begin // WORKAROUND

```
if ($NB.Main->wpcurrent<>'NB.Page1') then

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

end;

end;
```

```
//=====

// EvtInit

//      Initialisieren der Applikation

//=====

sub EvtInit(

    aEvt : event;      // Ereignis

): logic

local begin

    vPar : int;

    vHdl : int;

end;

begin

    WinSearchPath(aEvt:Obj);

    gTitle  # Translate(cTitle);

    gFile   # cFile;

    gMenuName # cMenuName;

    gPrefix # cPrefix;

    gZLList # winsearch(aEvt:Obj,cZList);
```

```
gKey    # cKey;
```

```
vHdl # Winsearch(aEvt:Obj, 'ZL.Erfassung');
```

```
Lib_GuiCom:RecallList(vHdl,'AUF.');
```

```
// Chemietitel ggf. setzen
```

```
if (Set.Chemie.Titel.C<>") then begin
```

```
    $lbAuf.P.Chemie.C1->wpcaption # Set.Chemie.Titel.C;
```

```
end;
```

```
if (Set.Chemie.Titel.Si<>") then begin
```

```
    $lbAuf.P.Chemie.Si1->wpcaption # Set.Chemie.Titel.Si;
```

```
end;
```

```
if (Set.Chemie.Titel.Mn<>") then begin
```

```
    $lbAuf.P.Chemie.Mn1->wpcaption # Set.Chemie.Titel.Mn;
```

```
end;
```

```
if (Set.Chemie.Titel.P<>") then begin
```

```
    $lbAuf.P.Chemie.P1->wpcaption # Set.Chemie.Titel.P;
```

```
end;
```

```
if (Set.Chemie.Titel.S<>") then begin
```

```
    $lbAuf.P.Chemie.S1->wpcaption # Set.Chemie.Titel.S;
```

```
end;
```

```
if (Set.Chemie.Titel.Al<>") then begin
```

```
    $lbAuf.P.Chemie.Al1->wpcaption # Set.Chemie.Titel.Al;
```

```
end;
```

```
if (Set.Chemie.Titel.Cr<>") then begin
```

```
    $lbAuf.P.Chemie.Cr1->wpcaption # Set.Chemie.Titel.Cr;
```

end;

if (Set.Chemie.Titel.V<>") then begin

 \$lbAuf.P.Chemie.V1->wpcaption # Set.Chemie.Titel.V;

end;

if (Set.Chemie.Titel.Nb<>") then begin

 \$lbAuf.P.Chemie.Nb1->wpcaption # Set.Chemie.Titel.Nb;

end;

if (Set.Chemie.Titel.Ti<>") then begin

 \$lbAuf.P.Chemie.Ti1->wpcaption # Set.Chemie.Titel.Ti;

end;

if (Set.Chemie.Titel.N<>") then begin

 \$lbAuf.P.Chemie.N1->wpcaption # Set.Chemie.Titel.N;

end;

if (Set.Chemie.Titel.Cu<>") then begin

 \$lbAuf.P.Chemie.Cu1->wpcaption # Set.Chemie.Titel.Cu;

end;

if (Set.Chemie.Titel.Ni<>") then begin

 \$lbAuf.P.Chemie.Ni1->wpcaption # Set.Chemie.Titel.Ni;

end;

if (Set.Chemie.Titel.Mo<>") then begin

 \$lbAuf.P.Chemie.Mo1->wpcaption # Set.Chemie.Titel.Mo;

end;

if (Set.Chemie.Titel.B<>") then begin

 \$lbAuf.P.Chemie.B1->wpcaption # Set.Chemie.Titel.B;

end;

if (Set.Chemie.Titel.1<>") then begin

```

$lbAuf.P.Chemie.Frei1.1->wpcaption # Set.Chemie.Titel.1;

end;

if ("Set.Mech.Titel.Härte"<>") then begin

    $lbAuf.P.Haerte1->wpcaption # "Set.Mech.Titel.Härte";

end;

if ("Set.Mech.Titel.Körn"<>") then begin

    $lbAuf.P.Koernung1->wpcaption # "Set.Mech.Titel.Körn";

end;

if ("Set.Mech.Titel.Sonst"<>") then begin

    $lbAuf.P.Mech.Sonstig1->wpcaption # "Set.Mech.Titel.Sonst";

end;

if ("Set.Mech.Titel.Rau1"<>") then begin

    $lbAuf.P.RauigkeitA1->wpcaption # "Set.Mech.Titel.Rau1";

end;

if ("Set.Mech.Titel.Rau2"<>") then begin

    $lbAuf.P.RauigkeitB1->wpcaption # "Set.Mech.Titel.Rau2";

end;


// Verpackungstitel setzen

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

    $lbAuf.P.VpgText1 -> wpcaption # Set.Vpg1.Titel;

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

    $lbAuf.P.VpgText2 -> wpcaption # Set.Vpg2.Titel;

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

    $lbAuf.P.VpgText3 -> wpcaption # Set.Vpg3.Titel;

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

```

```

$lbAuf.P.VpgText4 -> wpcaption # Set.Vpg4.Titel;

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

    $lbAuf.P.VpgText5 -> wpcaption # Set.Vpg5.Titel;

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

    $lbAuf.P.VpgText6 -> wpcaption # Set.Vpg6.Titel;


if (Set.Mech.Dehnung.Wie=1) then

    $edAuf.P.DehnungA2->wpcustom # '_N';

if (Set.Mech.Dehnung.Wie=2) then

    $edAuf.P.DehnungB2->wpcustom # '_N';


// Feldberechtigungen...

if (Rechte[Rgt_Auf_Preise]) then begin

    $clmGV.Num.01->wpvisible      # true; // Restwert

    $clmAuf.P.Grundpreis->wpvisible  # true;

    //$edAuf.P.Kalkuliert->wpvisible # true;

    $edAuf.P.Grundpreis->wpvisible  # true;

    $edRabatt1->wpvisible          # true;

    $lb.Poswert->wpvisible          # true;

    $lb.Kalkuliert->wpvisible       # true;

    $edAuf.P.Kalkuliert_Mat->wpvisible # true;

    $edAuf.P.Grundpreis_Mat->wpvisible # true;

    $lb.Rohgewinn->wpvisible        # true;

    $lb.Aufpreise_Mat->wpvisible    # true;

    $lb.Aufpreise->wpvisible        # true;

    $lb.P.Einzelpreis_Mat->wpvisible  # true;

```

```

$lb.Poswert_Mat->wpvisible      # true;

$clmAuf.P.Grundpreis_ERF->wpvisible # true;

$clmGV.Num.01_ERF->wpvisible      # true;

$clmGV.Num.02_ERF->wpvisible      # true;

end;

if (Rechte[Rgt_Auf_P_PEH_Edit]=false) then begin

    $edAuf.P.PEH->wpcustom # '_N';

    $edAuf.P.MEH.Preis->wpcustom # '_N';

    $bt.PreisMEH->wpcustom # '_N';

    $edAuf.P.PEH_Mat->wpcustom # '_N';

    $edAuf.P.MEH.Preis_Mat->wpcustom # '_N';

    $bt.PreisMEH_Mat->wpcustom # '_N';

end;

```

```

Lib_Guicom2:Underline($edAuf.Kundennr);

Lib_Guicom2:Underline($edAuf.Lieferadresse);

Lib_Guicom2:Underline($edAuf.Lieferanschrift);

Lib_Guicom2:Underline($edAuf.Verbraucher);

Lib_Guicom2:Underline($edAuf.Rechnungsempf);

Lib_Guicom2:Underline($edAuf.Rechnungsanschr);

Lib_Guicom2:Underline($edAuf.Best.Bearbeiter);

Lib_Guicom2:Underline($edAuf.BDSNummer);

Lib_Guicom2:Underline($edAuf.Waehrung);

Lib_Guicom2:Underline($edAuf.Lieferbed);

Lib_Guicom2:Underline($edAuf.Zahlungsbed);

Lib_Guicom2:Underline($edAuf.Versandart);

```

Lib_Guicom2:Underline(\$SedAuf.Vertreter1);

Lib_Guicom2:Underline(\$SedAuf.Land);

Lib_Guicom2:Underline(\$SedAuf.Steuerschluessel);

Lib_Guicom2:Underline(\$SedAuf.Sachbearbeiter);

Lib_Guicom2:Underline(\$SedAuf.Vertreter2);

Lib_Guicom2:Underline(\$SedAuf.P.Auftragsart);

Lib_Guicom2:Underline(\$SedAuf.P.Warengruppe);

Lib_Guicom2:Underline(\$SedAuf.P.Projektnummer);

Lib_Guicom2:Underline(\$SedAuf.P.Guete);

Lib_Guicom2:Underline(\$SedAuf.P.ArtikelNr);

Lib_Guicom2:Underline(\$SedAuf.P.Auftragsart_Mat);

Lib_Guicom2:Underline(\$SedAuf.P.Warengruppe_Mat);

Lib_Guicom2:Underline(\$SedAuf.P.Projektnummer_Mat);

Lib_Guicom2:Underline(\$SedAuf.P.AbrufAufPos_Mat);

Lib_Guicom2:Underline(\$SedAuf.P.Erzeuger_Mat);

Lib_Guicom2:Underline(\$SedAuf.P.Kalkuliert_Mat);

Lib_Guicom2:Underline(\$SedAuf.P.Grundpreis_Mat);

Lib_Guicom2:Underline(\$SedAuf.P.TextNr2b);

Lib_Guicom2:Underline(\$SedAuf.P.Verpacknr);

Lib_Guicom2:Underline(\$SedAuf.P.Verwiegungsart);

Lib_Guicom2:Underline(\$SedAuf.P.Etikettentyp);

Lib_Guicom2:Underline(\$SedAuf.P.Zwischenlage);

Lib_Guicom2:Underline(\$edAuf.P.Unterlage);

Lib_Guicom2:Underline(\$edAuf.P.Umverpackung);

Lib_Guicom2:Underline(\$edAuf.P.Etikettentyp2);

Lib_Guicom2:Underline(\$edAuf.P.Skizzennummer);

Lib_Guicom2:Underline(\$edAuf.P.EinsatzVPG.Nr);

Lib_Guicom2:Underline(\$edAuf.P.VorlageBAG);

// Auswahlfelder...

SetStdAusFeld('edAuf.Vorgangstyp' , 'Vorgangstyp');

SetStdAusFeld('edAuf.Kundennr' , 'Kunde');

SetStdAusFeld('edAuf.Lieferadresse' , 'Lieferadresse');

SetStdAusFeld('edAuf.Lieferanschrift' , 'Lieferanschrift');

SetStdAusFeld('edAuf.Verbraucher' , 'Verbraucher');

SetStdAusFeld('edAuf.Rechnungsempf' , 'Rechnungsempf');

SetStdAusFeld('edAuf.Rechnungsanschr' , 'Rechnungsanschr');

SetStdAusFeld('edAuf.Best.Bearbeiter' , 'Ansprechpartner');

SetStdAusFeld('edAuf.BDSNummer' , 'BDSNummer');

SetStdAusFeld('edAuf.Land' , 'Land');

SetStdAusFeld('edAuf.Waehrung' , 'Waehrung');

SetStdAusFeld('edAuf.Lieferbed' , 'Lieferbed');

SetStdAusFeld('edAuf.Zahlungsbed' , 'Zahlungsbed');

SetStdAusFeld('edAuf.Steuerschluessel' , 'Steuerschluessel');

SetStdAusFeld('edAuf.Versandart' , 'Versandart');

SetStdAusFeld('edAuf.Sprache' , 'Sprache');

```
SetStdAusFeld('edAuf.AbmessungenEH'      , 'AbmessungenEH');
SetStdAusFeld('edAuf.GewichtsEH'          , 'GewichtsEH');
SetStdAusFeld('edAuf.Sachbearbeiter'      , 'Sachbearbeiter');
SetStdAusFeld('edAuf.Vertreter1'          , 'Vertreter1');
SetStdAusFeld('edAuf.Vertreter2'          , 'Vertreter2');
SetStdAusFeld('edAuf.P.Auftragsart'        , 'Auftragsart');
SetStdAusFeld('edAuf.P.Warengruppe'        , 'Warengruppe');
SetStdAusFeld('edAuf.P.AbrufAufNr'        , 'Abruf');
SetSpeziAusFeld('edAuf.P.AbrufAufPos'     , 'AbrufPos');
SetStdAusFeld('edAuf.P.ArtikelNr'         , 'Artikelnummer');
// SetStdAusFeld('edAuf.P.KundenArtNr_Mat' , 'KundenArtNr');
SetStdAusFeld('edAuf.P.KundenArtNr'       , 'KundenArtNr');
SetStdAusFeld('edAuf.P.Projektnummer'     , 'Projekt');
SetStdAusFeld('edAuf.P.MEH.Wunsch'        , 'MEH');
SetStdAusFeld('edAuf.P.Termin1W.Art'      , 'Terminart');
SetStdAusFeld('edAuf.P.MEH.Preis'         , 'PreisMEH');
SetSpeziAusFeld('edAuf.P.Grundpreis'      , 'Preis');
SetStdAusFeld('edAuf.P.Auftragsart_Mat'   , 'Auftragsart');
SetStdAusFeld('edAuf.P.Warengruppe_Mat'   , 'Warengruppe');
SetStdAusFeld('edAuf.P.AbrufAufNr_Mat'    , 'Abruf');
SetSpeziAusFeld('edAuf.P.AbrufAufPos_Mat' , 'AbrufPos');

SetStdAusFeld('edAuf.P.ArtikelNr_Mat'     , 'Artikelnummer_Mat');
SetStdAusFeld('edAuf.P.Guete_Mat'         , 'Guete');
SetStdAusFeld('edAuf.P.Guete'             , 'Guete');
SetStdAusFeld('edAuf.P.Guetenstufe_Mat'   , 'Guetenstufe');
```

SetStdAusFeld('edAuf.P.AusfOben_Mat' , 'AusfOben');
SetStdAusFeld('edAuf.P.AusfUnten_Mat' , 'AusfUnten');
SetSpeziAusFeld('edAuf.P.KundenMatArtNr_Mat' , 'KundenMatArtNr');
SetStdAusFeld('edAuf.P.Termin1W.Art_Mat' , 'Terminart');
SetStdAusFeld('edAuf.P.Zeugnisart_Mat' , 'Zeugnis');
SetStdAusFeld('edAuf.P.Zeugnisart' , 'Zeugnis');
SetStdAusFeld('edAuf.P.Projektnummer_Mat' , 'Projekt');
SetStdAusFeld('edAuf.P.Erzeuger_Mat' , 'Erzeuger');
SetStdAusFeld('edAuf.P.Intrastatnr_Mat' , 'Intrastat');
SetStdAusFeld('edAuf.P.Intrastatnr' , 'Intrastat');
SetStdAusFeld('edAuf.P.MEH.Preis_Mat' , 'PreisMEH');
SetStdAusFeld('edAuf.P.Kalkuliert_Mat' , 'Kalkulation');
SetStdAusFeld('edAuf.P.Grundpreis_Mat' , 'Preis');
SetStdAusFeld('edAuf.P.Skizzennummer' , 'Skizze');

SetStdAusFeld('edAuf.P.VorlageBAG' , 'VorlageBAG');
SetStdAusFeld('edAuf.P.EinsatzVPG.Adr' , 'EinsatzVPG');
SetStdAusFeld('edAuf.P.EinsatzVPG.Nr' , 'EinsatzVPG2');

SetStdAusFeld('edAuf.P.Verpacknr' , 'Verpackung');
SetStdAusFeld('edAuf.P.TextNr2' , 'Text');
SetStdAusFeld('edAuf.P.TextNr2b' , 'Text2');
SetStdAusFeld('edAuf.P.Zwischenlage' , 'Zwischenlage');
SetStdAusFeld('edAuf.P.Unterlage' , 'Unterlage');
SetStdAusFeld('edAuf.P.Umverpackung' , 'Umverpackung');
SetStdAusFeld('edAuf.P.Verwieguungsart' , 'Verwieguungsart');

```
SetStdAusFeld('edAuf.P.Etikettentyp'      , 'Etikett');  
SetStdAusFeld('edAuf.P.Etikettentyp2'    , 'Etikett2');
```

```
if (Set.LyseErweitertYN) then begin
```

```
  vHdl # Winsearch(aEvt:Obj, 'lbAuf.P.SbeligkeitMax');  
  if (vHdl<>0) then vHdl->wpVisible # false;  
  vHdl # Winsearch(aEvt:Obj, 'edAuf.P.SbeligkeitMax');  
  if (vHdl<>0) then vHdl->wpVisible # false;  
  vHdl # Winsearch(aEvt:Obj, 'lbAuf.P.SaebelProM');  
  if (vHdl<>0) then vHdl->wpVisible # false;  
  vHdl # Winsearch(aEvt:Obj, 'edAuf.P.SaebelProM');  
  if (vHdl<>0) then vHdl->wpVisible # false;  
  vHdl # Winsearch(aEvt:Obj, 'lbSaebel');  
  if (vHdl<>0) then vHdl->wpVisible # false;
```

```
  vPar # Winsearch(aEvt:Obj, 'NB.Page4');
```

```
  vHdl # Winsearch(aEvt:Obj, 'bt.InternerText');
```

```
  vHdl # Lib_GuiCom2:CreateObjFrom(vHdl, _WinTypeButton, vPar, 'bt.AnalyseErweitert', 'erweiterte Ana
```

```
  vHdl->wpCustom # '_I';
```

```
  Lib_GuiCom2:Hide(vPar, 'lbAnalyseStart', 'lbAnalyseEnde');
```

```
  Lib_MoreBufs:Init(401);
```

```
end;
```

```
// Ankerfunktion?
```

```
// RunAFX('Auf.P.Init',aint(aevt:obj));

// RETURN App_Main:EvtInit(aEvt);

RunAFX('Auf.P.Init.Pre',aint(aevt:obj));

App_Main:EvtInit(aEvt);

RunAFX('Auf.P.Init',aint(aEvt:Obj));

RETURN true;
```

```
end;
```

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

```
// Etikettendaten
```

```
//
```

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

```
sub Etikettendaten();
```

```
local begin
```

```
    vA1,vA2 : alpha;
```

```
end;
```

```
begin
```

```
    // SPEZI
```

```
    gMDI->wpdisabled # y;
```

```
    Dlg_Standard:Standard('Artikelnummer',var vA1);
```

```
    Dlg_Standard:Standard('Abmessung',var vA2);
```

```
    gMDI->wpdisabled # n;
```

```
    $edAuf.P.Etikettentyp2->Winfocusset(false);
```

```
    Auf.P.VpgText1 # vA1;
```

```

Auf.P.VpgText2 # vA2;

$edAuf.P.VpgText12->winupdate(_WinUpdFld2Obj);

$edAuf.P.VpgText22->winupdate(_WinUpdFld2Obj);

end;


//=====

// Skizzendaten

//

//=====

sub Skizzendaten();

local begin

    vZ      : int;

    vA      : alpha;

    vWert   : alpha;

    vX      : int;

    vA3,vA4,vA5 : alpha(1000);

end;

begin

    // SPEZI

    gMDI->wpdisabled # y;

    FOR vX # 1 loop inc(vX) WHILE (vX<=Skz.Anzahl.Variablen) do begin

        vA # StrChar(64+vX);

        if (false) then begin

```

```
Dlg_Standard:Anzahl('Variable '+vA,var vZ,0,300,400);
```

```
case (vX%3) of
```

```
0 : if (vA5=") then
```

```
    vA5 # vA + '='+cnvai(vZ)
```

```
else
```

```
    vA5 # vA5 +', '+ vA + '='+cnvai(vZ);
```

```
1 : if (vA3=") then
```

```
    vA3 # vA + '='+cnvai(vZ)
```

```
else
```

```
    vA3 # vA3 +', '+ vA + '='+cnvai(vZ);
```

```
2 : if (vA4=") then
```

```
    vA4 # vA + '='+cnvai(vZ)
```

```
else
```

```
    vA4 # vA4 +', '+ vA + '='+cnvai(vZ);
```

```
end; // case
```

```
end;
```

```
if (true) then begin
```

```
Dlg_Standard:Standard('Variable '+vA,var vWert);
```

```
case (vX%3) of
```

```
0 : if (vA5=") then
```

```
    vA5 # vA + '='+vWert;
```

```
else
```

```
    vA5 # vA5 +', '+ vA + '='+vWert;
```

```
1 : if (vA3=") then
```

```
    vA3 # vA + '='+vWert;
```

else

vA3 # vA3 +'; '+ vA + '='+vWert;

2 : if (vA4="") then

vA4 # vA + '='+vWert;

else

vA4 # vA4 +'; '+ vA + '='+vWert;

end; // case

end;

END;

gMDI->wpdisabled # n;

\$edAuf.P.Skizzennummer->Winfocusset(false);

Auf.P.VpgText4 # StrCut(vA3,1,64);

Auf.P.VpgText5 # StrCut(vA4,1,64)

Auf.P.VpgText6 # StrCut(vA5,1,64)

\$edAuf.P.VpgText42->winupdate(_WinUpdFld2Obj);

\$edAuf.P.VpgText52->winupdate(_WinUpdFld2Obj);

\$edAuf.P.VpgText62->winupdate(_WinUpdFld2Obj);

end;

//=====

// RefreshIfm

// "Infomasken" refreshen

//=====


```

sub RefreshIfm(

    opt aName    : alpha;

    opt aChanged : logic;

)

local begin

    vHdl      : inT;

    vTxtHdl   : int;

    vArtChanged : logic;

    iTemp     : int;

    vListHdl   : int;

    vPageName  : alpha;

    vA         : alpha;

end;

begin

    if (Mode=c_ModeList) then RETURN;

    if (mode=c_ModeList2) then begin

        if (RecLinkInfo(401,400,9,_RecCount)>0) then // 21.05.2015

            $ZL.Erfassung->WinUpdate(_WinUpdOn, _WinLstRecFromRecId | _WinLstRecDoSelect);

        end;

        // Ankerfunktion?

        if (aChanged) then vA # '1'+aName

        else vA # '0' + aName;

        if (RunAFX('Auf.P.RefreshIfm',vA)<0) then RETURN;
    end;
end;

```

```
RecBufClear(402);
```

```
$RL.AFOben->WinUpdate(_WinUpdOn, _WinLstFromFirst | _WinLstRecDoSelect);
```

```
$RL.AFUnten->WinUpdate(_WinUpdOn, _WinLstFromFirst | _WinLstRecDoSelect);
```

```
vTxtHdl # $Auf.P.TextEditPos->wpdbTextBuf; // Textpuffer ggf. anlegen
```

```
if (vTxtHdl=0) then begin
```

```
    vTxtHdl # TextOpen(32);
```

```
    $Auf.P.TextEditPos->wpdbTextBuf # vTxtHdl;
```

```
end;
```

```
vTxtHdl # $Auf.P.TextStammdaten->wpdbTextBuf; // Textpuffer ggf. anlegen
```

```
if (vTxtHdl=0) then begin
```

```
    vTxtHdl # TextOpen(32);
```

```
    $Auf.P.TextStammdaten->wpdbTextBuf # vTxtHdl;
```

```
end;
```

```
vTxtHdl # $Auf.P.TextEditKopf->wpdbTextBuf; // Textpuffer ggf. anlegen
```

```
if (vTxtHdl=0) then begin
```

```
    vTxtHdl # TextOpen(32);
```

```
    $Auf.P.TextEditKopf->wpdbTextBuf # vTxtHdl;
```

```
end;
```

```
vTxtHdl # $Auf.P.TextEditFuss->wpdbTextBuf; // Textpuffer ggf. anlegen
```

```
if (vTxtHdl=0) then begin
```

```
    vTxtHdl # TextOpen(32);
```

```
    $Auf.P.TextEditFuss->wpdbTextBuf # vTxtHdl;
```

end;

// ganze Seite refreshen?

vPageName # \$NB.Main->wpcurrent;

if (StrLen(aName)>3) then

if (StrCut(aName,1,3)='NB.') then begin

vPageName # aName;

aName # "";

end;

//debugx('page:'+vpagename);

if (vPageName='NB.Kopf') then

Auf_P_SMain:RefreshIfm_Kopf(aName, aChanged)

else if (vPageName='NB.Page1') and (\$NB.Page1->wpcustom='NB.Page1_Art') then

Auf_P_SMain:RefreshIfm_Page1_Art(aName, aChanged)

else if (vPageName='NB.Page1') and (\$NB.Page1->wpcustom='NB.Page1_Mat') then

Auf_P_SMain:RefreshIfm_Page1_Mat(aName, aChanged)

else if (vPageName='NB.Page2') then

Auf_P_SMain:RefreshIfm_Page2(aName, aChanged)

else

Auf_P_SMain:RefreshIfm_Page3(aName, aChanged);

// veränderte Felder in Objekte schreiben

if (aName<>") then begin

vHdl # gMdi->winsearch(aName);

if (vHdl<>0) then

```

    vHdl->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:PflichtfelderEinfärben();

// Ankerfunktion?

if (aChanged) then vA # '1'+aName
else vA # '0' + aName;

RunAFX('Auf.P.RefreshIfm.Post',vA);

end;

//=====

// Reclnit

//      Init für Znderung und Neuanlage

//=====

sub Reclnit(

    opt aBehalten    : logic;

    opt aMitAufp     : logic;

```

```

    opt aMitKalk    : logic;

)

local begin

    Erx            : int;

    vHdl           : int;

    vPos           : int;

    vRab1,vRab2 : float;

    vTerm1W        : date;

    vTerm1WArt     : alpha;

    vTerm1WZahl    : int;

    vTerm1WJahr    : int;

    vTxt           : int;

    vOK            : logic;

end;

begin

    // Felder Disablen durch:

    if ("Auf.WährungFixYN") then begin

        Lib_GuiCom:Enable($edAuf.Waehrungskurs);

    end

    else begin

        Lib_GuiCom:Disable($edAuf.Waehrungskurs);

    end;

    if (Auf.P.MEH.Wunsch=Auf.P.MEH.Einsatz) then begin

//    Auf.P.Menge # Auf.P.Menge.Wunsch;

```

```
// 24.03.2016   Lib_GuiCom:Disable($edAuf.P.Menge);
```

```
    $edAuf.P.Menge->WinUpdate(_WinUpdFld2Obj);
```

```
end
```

```
else begin
```

```
// 24.03.2016   Lib_GuiCom:Enable($edAuf.P.Menge);
```

```
end;
```

```
$edRabatt1->wpcaptionfloat # 0.0;
```

```
vRab1 # 0.0;
```

```
vRab2 # 0.0;
```

```
Erx # RecLink(403,401,6,_RecFirst);
```

```
WHILE (Erx<=_rLocked) and ((vRab1=0.0) or (vRab2=0.0)) do begin
```

```
    if ("Auf.Z.Schlüssel"='*RAB1') then vRab1 # (-1.0) * Auf.Z.Menge;
```

```
    if ("Auf.Z.Schlüssel"='*RAB2') then vRab2 # (-1.0) * Auf.Z.Menge;
```

```
    Erx # RecLink(403,401,6,_RecNext);
```

```
END;
```

```
// Felder Abruf/Rechnungsnr. aktivieren...
```

```
begin // 06.08.2009
```

```
    vOK # n;
```

```
    if (Mode=c_ModeNew) or (Mode=c_ModeNew2) or (mode=c_modeEdit2) then begin
```

```
        if (Auf.AbrufYN) or (Auf.PAbrufYN) or
```

```
            ((Auf.Vorgangstyp<>c_VorlageAuf) and (Auf.Vorgangstyp<>c_Auf) and (Auf.Vorgangstyp<>c_Ang) a
```

end;

// 30.10.2015

if (Mode=c_ModeEdit) and (Auf.Vorgangstyp=C_REKOR) and (Auf.P.Abrufaufnr=0) then vOK # y;

if (vOK) then begin

Lib_GuiCom:Enable(\$edAuf.P.AbrufAufNr);

Lib_GuiCom:Enable(\$edAuf.P.AbrufAufPos);

Lib_GuiCom:Enable(\$bt.Abruf);

Lib_GuiCom:Enable(\$edAuf.P.AbrufAufNr_Mat);

Lib_GuiCom:Enable(\$edAuf.P.AbrufAufPos_Mat);

Lib_GuiCom:Enable(\$bt.Abruf_Mat);

end

else begin

Lib_GuiCom:Disable(\$edAuf.P.AbrufAufNr);

Lib_GuiCom:Disable(\$edAuf.P.AbrufAufPos);

Lib_GuiCom:Disable(\$bt.Abruf);

Lib_GuiCom:Disable(\$edAuf.P.AbrufAufNr_Mat);

Lib_GuiCom:Disable(\$edAuf.P.AbrufAufPos_Mat);

Lib_GuiCom:Disable(\$bt.Abruf_Mat);

end;

end; // ... Felder Abruf/ReNr.

if (Mode=c_ModeEdit) then begin // Edit?

\$edRabatt1->wpcaptionfloat # vRab1;

```
// echter Auftrag?

if (Auf.P.Nummer<>0) and (Auf.P.Nummer<1000000000) then begin

    Lib_GuiCom:Disable($edAuf.Datum);

    // Angebote können immer verändert werden (Prj.1488/43)

    if (Auf.Vorgangstyp<>C_Ang) then begin

        Lib_GuiCom:Disable($cbAuf.LiefervertragYN);

        Lib_GuiCom:Disable($cbAuf.AbrufYN);

        Lib_GuiCom:Disable($cbAuf.PAbrufYN);

    end;

    Lib_GuiCom:Disable($edAuf.Vorgangstyp);

    Lib_GuiCom:Disable($bt.Vorgangstyp);

    Lib_GuiCom:Disable($edAuf.Kundennr);

    Lib_GuiCom:Disable($bt.Kunde);

    Lib_GuiCom:Disable($edAuf.Rechnungsempf);

    Lib_GuiCom:Disable($edAuf.Rechnungsanschr);

    Lib_GuiCom:Disable($bt.RechEmpf);

    Lib_GuiCom:Disable($bt.RechAnschr);

    Lib_GuiCom:Disable($edAuf.P.ArtikelNr);

    Lib_GuiCom:Disable($bt.Artikelnummer);

    Lib_GuiCom:Disable($edAuf.P.KundenArtNr);

    Lib_GuiCom:Disable($bt.KundenArtNr);

    if (Wgr_Data:IstMix(Auf.P.Wgr.Dateinr)) then begin

        Lib_GuiCom:Disable($edAuf.P.ArtikelNr_Mat);

        Lib_GuiCom:Disable($bt.Artikelnummer_Mat);
```



```
Lib_GuiCom:Enable($bt.Preis_Mat);
```

```
end
```

```
else begin
```

```
Lib_GuiCom:Enable($edAuf.P.Artikelnr_Mat);
```

```
Lib_GuiCom:Enable($bt.Artikelnummer_Mat);
```

```
Lib_GuiCom:Disable($bt.Preis_Mat);
```

```
end;
```

```
//Lib_GuiCom:Disable($edAuf.P.KundenMatArtNr_Mato);
```

```
Lib_GuiCom:Disable($bt.KundenMatArtNr_Mat);
```

```
// Lib_GuiCom:Disable($edAuf.P.Termin1W.Art);
```

```
Lib_GuiCom:Disable($edAuf.P.MEH.Wunsch);
```

```
Lib_GuiCom:Disable($bt.MEH);
```

```
if (Auf.Zahlungsbed <> 0) then begin
```

```
Exr # RekLink(816,400,6,0);
```

```
if ("ZaB.FixImAuftragYN") AND (Rechte[Rgt_Auf_Zbd_Aendern] = false ) then begin
```

```
Lib_GuiCom:Disable($edAuf.Zahlungsbed);
```

```
Lib_GuiCom:Disable($bt.Zahlungsbed);
```

```
end else begin
```

```
Lib_GuiCom:Enable($edAuf.Zahlungsbed);
```

```
Lib_GuiCom:Enable($bt.Zahlungsbed);
```

```
end;
```

```
end;
```

end;

end;

// Focus setzen auf Feld:

if (Mode=c_ModeEdit) or (Mode=c_ModeEdit2) then begin // Edit?

Lib_MoreBufs:Reclnit(401, false);

if (Auf.P.Nummer<>0) and (Auf.P.Nummer<1000000000) then begin

Erx # RecLink(400,401,3,_RecLock); // Kopf holen

PtD_Main:Memorize(400);

end;

if (\$NB.Page1->wpcustom='NB.Page1_Art') then

\$sedAuf.P.Auftragsart->WinFocusSet(true)

else

\$sedAuf.P.Auftragsart_Mat->WinFocusSet(true);

if (Auf.Vorgangstyp=c_ANG) or (Auf.LiefervertragYN) or (Auf.Vorgangstyp=c_VorlageAuf) then begin

Lib_guiCom:Enable(\$sedAuf.GltigkeitVom);

Lib_guiCom:Enable(\$sedAuf.GltigkeitBis);

end

else begin

Lib_guiCom:Disable(\$sedAuf.GltigkeitVom);

Lib_guiCom:Disable(\$sedAuf.GltigkeitBis);

end;

```
// if ("Auf.P.Länge"<>0.0) then begin  
//   Lib_GuiCom:Disable($edAuf.P.RID_Mat);  
//   Lib_GuiCom:Disable($edAuf.P.RIDMax_Mat);  
//   Lib_GuiCom:Disable($edAuf.P.RAD_Mat);  
//   Lib_GuiCom:Disable($edAuf.P.RADMax_Mat);  
// end;  
  
end;
```

```
if (Mode=c_ModeNew2) then begin // neue Position anlegen
```

```
Lib_GuiCom:Disable($edAuf.Kundennr);  
Lib_GuiCom:Disable($bt.Kunde);
```

```
$NB.Page1->wpdisabled # false;  
  
if ($NB.Page1->wpcustom='NB.Page1_Art') then begin  
  vHdl # gMdi->Winsearch('NB.Main');  
  
  vHdl->wpcurrent # 'NB.Page1'; // doppelt als Workaround  
  
  vHdl->wpcurrent # 'NB.Page1';  
  
  $edAuf.P.Auftragsart->WinFocusSet(true)  
  
end  
  
else begin  
  
  vHdl # gMdi->Winsearch('NB.Main');  
  
  vHdl->wpcurrent # 'NB.Page1';
```

```

$edAuf.P.Auftragsart_Mat->WinFocusSet(true);

end;

$NB.Erfassung->wpvisible # false;


if (aBehalten) then begin // Position mit altem Inhalt -----
    if (w_AppendNr<>0) then begin // 2023-01-26 AH
        Erx # RecLink(401,400,9,_RecLast);

        vPos # Auf.P.Position + 1;

        Erx # RecRead(401,0,0,w_AppendNr);
// 2023-04-20 AHProj. 2465/81/1      w_AppendNr # 0;

    end

    else begin

        Erx # RecLink(401,400,9,_RecLast);

        if (Erx<=_rLocked) then

            vPos # Auf.P.Position + 1;

        end;

        if (vPos>0) then begin

            Lib_MoreBufs:Reclnit(401, y, y);


            w_BinKopieVonDatei # gFile;

            w_BinKopieVonRecID # RecInfo(gFile, _recid);

            // internen Text kopieren 29.10.2021 AH

            TxtCopy(myTmpText+'.401.'+CnvAI(Auf.P.Position,_FmtNumLeadZero | _FmtNumNoGroup,0,3)+'.01

            // Aufpreise kopieren?

```

```
if (aMitAufp) then begin
    Erx # RecLink(403,401,6,_recFirst);
    WHILE (Erx<=_rLocked) do begin
        Auf.Z.Position # vPos;
        Erx # RekInsert(403,0,'MAN');
        Auf.Z.Position # Auf.P.Position;
        Recread(403,1,0);
        Erx # RecLink(403,401,6,_recNext);
    END;
END;
```

end

else begin

```
Auf.P.Aufpreis # 0.0;
```

end;

// Kalkulation kopieren?

if (aMitKalk) then begin

```
Erx # RecLink(405,401,7,_recFirst);
```

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

```
Auf.K.Position # vPos;
```

```
Erx # RekInsert(405,0,'MAN');
```

```
Auf.K.Position # Auf.P.Position;
```

```
Recread(405,1,0);
```

```
Erx # RecLink(405,401,7,_recNext);
```

```
END;
```

end

else begin

```
Auf.P.Kalkuliert # 0.0;
```

end;

Auf.P.Einzelpreis # Auf.P.Grundpreis + Auf.P.Aufpreis;

// Ausführungen kopieren...

Erx # RecLink(402,401,11,_recFirst);

WHILE (Erx<=_rLocked) do begin

 Auf.AF.Position # vPos;

 Erx # RekInsert(402,0,'MAN');

 Auf.AF.Position # Auf.P.Position;

 Recread(402,1,0);

 Erx # RecLink(402,401,11,_recNext);

END;

end

else begin

 vPos # 1;

end;

\$edRabatt1->wpcaptionfloat # vRab1;

end

else begin // leere Position erzeugen -----

 Erx # RecLink(401,400,9,_RecLast);

 if (Erx<>_rOk) then begin

 vPos # 1;

 end

else begin

vPos # Auf.P.Position + 1;

end;

vTerm1W # Auf.P.Termin1Wunsch;

vTerm1WArt # Auf.P.Termin1W.Art;

vTerm1WZahl # Auf.P.Termin1W.Zahl;

vTerm1WJahr # Auf.P.Termin1W.Jahr;

Lib_MoreBufs:RecInit(401, y);

RecBufClear(401);

Auf.P.Kundennr # Auf.Kundennr;

Auf.P.KundenSW # Auf.KundenStichwort;

Auf.P.MEH.Preis # Set.Auf.MEH.PEH;

Auf.P.PEH # Set.Auf.PEH;

Auf.P.MEH.Wunsch # Auf.P.MEH.Preis;

Auf.P.Warengruppe # Set.Auf.Warengruppe;

Auf.P.Wgr.Dateinr # Set.Auf.Dateinr; // 2023-04-25 AH ff.

Erx # RecLink(819,401,1,0); // Warengruppe holen

if (Erx>_rLocked) then RecBufClear(819)

else Auf.P.Wgr.DateiNr # Wgr.Dateinummer;

Auf_Data:SetWgrDateinr(Auf.P.Wgr.Dateinr);

RecLink(100,400,1,0); // Kunde holen

Auf.P.Verwiegungsart # Adr.VK.Verwiegeart;

Auf.P.Auftragsart # Set.Auf.Auftragsart;

```
if (vTerm1WArt="") then  
    vTerm1WArt # Set.Auf.TerminArt;  
    Auf.P.Best.Nummer # Auf.Best.Nummer; // RICHTER Logik AUS  
    Auf.P.Termin1Wunsch # vTerm1W;  
    Auf.P.Termin1W.Art # vTerm1WArt;  
    Auf.P.Termin1W.Zahl # vTerm1WZahl;  
    Auf.P.Termin1W.Jahr # vTerm1WJahr;
```

```
$edAuf.P.ArtikelNr_Mat->wpcaption # ";
```

```
if (Set.Auf.PosText=999) then begin  
    $cb.Text1->wpcheckstate # _WinStateChkUnchecked;  
    $cb.Text2->wpcheckstate # _WinStateChkUnchecked;  
    $cb.Text3->wpcheckstate # _WinStateChkchecked;  
    Auf.P.TextNr1 # 401;  
    Auf.P.TextNr2 # 0;  
    $edAuf.P.TextNr2->wpCaptionInt # 0;  
    $edAuf.P.TextNr2b->wpCaptionInt # 0;  
end
```

```
else begin  
    $cb.Text1->wpcheckstate # _WinStateChkUnchecked;  
    $cb.Text2->wpcheckstate # _WinStateChkchecked;  
    $cb.Text3->wpcheckstate # _WinStateChkUnchecked;  
    Auf.P.TextNr1 # 0;  
    Auf.P.TextNr2 # 0;  
    $edAuf.P.TextNr2->wpCaptionInt # 0;
```



```
$edAuf.P.TextNr2b->wpCaptionInt # Set.Auf.Posttext;
```

```
end;
```

```
end;
```

```
Auf.P.Nummer      # Auf.Nummer;
```

```
Auf.P.Position    # vPos;
```

```
if (Auf.P.TextNr1=401) then
```

```
  $Auf.P.TextEditPos->wpcustom # myTmpText+'.401.'+CnvAI(Auf.P.Position,_FmtNumLeadZero | _Fm
```

```
Erx # RekLink(814,400,8,_RecFirst);
```

```
$lb.WAE1->wpcaption # "Wae.Kürzel";
```

```
$lb.WAE2->wpcaption # "Wae.Kürzel";
```

```
$lb.WAE3->wpcaption # "Wae.Kürzel";
```

```
$lb.WAE4->wpcaption # "Wae.Kürzel";
```

```
$lb.WAE5->wpcaption # "Wae.Kürzel";
```

```
// ST 2009-07-27
```

```
// Bei neuen Positionen RID und RAD Angaben wieder freischalten *****
```

```
// begin
```

```
// Lib_GuiCom:Enable($edAuf.P.RID_Mat);
```

```
// Lib_GuiCom:Enable($edAuf.P.RIDMax_Mat);
```

```
// Lib_GuiCom:Enable($edAuf.P.RAD_Mat);
```

```
// Lib_GuiCom:Enable($edAuf.P.RADMax_Mat);
```

```
// end; // ST 2009-07-27
```

```

// Auto BA-Auftrag?

if (w_Command='NimmMatFuerBA') then begin
    gSelected # cnvia(w_Cmd_para);
//    w_Cmd_para # "";
    AusMaterial();
end;

// Material per Drag&Drop eingefügt??

if (w_Command='AusMaterial') then begin
    w_Command # "";
    AusMaterial();
end;

RunAFX('Auf.P.Reclnit.Post',"");

gMDI->winupdate();

RETURN;

end; // neue Pos anlegen


// neuen Kopf&Pos. anlegen *****
// neuen Kopf&Pos. anlegen *****
// neuen Kopf&Pos. anlegen *****

if (Mode=c_ModeNew) then begin

```

// 21.03.2018 AH:

Lib_RmtData:UserReset('400');

// Anhängen???

if (w_AppendNr<>0) then begin

Auf.Nummer # w_AppendNr;

RecRead(400,1,0);

Erx # RecLink(401,400,9,_RecLast);

if (Erx<>_rOk) then vPos # 1

else vPos # Auf.P.Position + 1;

RecBufClear(401);

RecLink(100,400,1,0); // Kunde holen

Auf.P.Kundennr # Auf.Kundennr;

Auf.P.KundenSW # Auf.KundenStichwort;

Mode # c_modeNew2;

if ((Auf.AbrufYN) or (Auf.PAbrufYN) or ((Auf.Vorgangstyp != c_Auf) and (Auf.Vorgangstyp<>c_Vorlage/

Lib_GuiCom:Enable(\$edAuf.P.AbrufAufNr);

Lib_GuiCom:Enable(\$edAuf.P.AbrufAufPos);

Lib_GuiCom:Enable(\$bt.Abruf);

Lib_GuiCom:Enable(\$edAuf.P.AbrufAufNr_Mat);

Lib_GuiCom:Enable(\$edAuf.P.AbrufAufPos_Mat);

Lib_GuiCom:Enable(\$bt.Abruf_Mat);

end

else begin

Lib_GuiCom:Disable(\$edAuf.P.AbrufAufNr);

Lib_GuiCom:Disable(\$edAuf.P.AbrufAufPos);

```
Lib_GuiCom:Disable($bt.Abruf);  
  
Lib_GuiCom:Disable($SedAuf.P.AbrufAufNr_Mat);  
  
Lib_GuiCom:Disable($SedAuf.P.AbrufAufPos_Mat);  
  
Lib_GuiCom:Disable($bt.Abruf_Mat);
```

```
end;
```

```
Lib_MoreBufs:Reclnit(401, y, y);
```

```
end
```

```
else begin
```

```
    RecBufClear(400);
```

```
    vPos # 1;
```

```
end;
```

```
Lib_guiCom:Disable($SedAuf.GltigkeitVom);
```

```
Lib_guiCom:Disable($SedAuf.GltigkeitBis);
```

```
Lib_GuiCom:Enable($cbAuf.LiefervertragYN);
```

```
Lib_GuiCom:Enable($cbAuf.AbrufYN);
```

```
Lib_GuiCom:Enable($cbAuf.PAbrufYN);
```

```
Lib_GuiCom:Enable($SedAuf.Kundennr);
```

```
Lib_GuiCom:Enable($bt.Kunde);
```

```
Auf.Nummer      # myTmpNummer;
```

```
if (w_AppendNr=0) then begin
```

```
    Auf.Datum      # today;
```

```
    Auf.Sachbearbeiter # gUserName;
```

Auf.Vorgangstyp # Set.Auf.Vorgangstyp;

end;

\$edAuf.Kundennr->wpcaptionint # -1; // für 1. Eingabe - WinKrampf

if (w_Appendnr=0) then begin

vHdl # gMdi->Winsearch('NB.Main');

vHdl->wpcurrent # 'NB.Kopf';

end;

if (Auf.Vorgangstyp="") then

\$edAuf.Vorgangstyp->WinFocusSet(false)

else

\$edAuf.Kundennr->WinFocusSet(false);

\$NB.Erfassung->wpvisible # false;

Auf.P.Nummer # Auf.Nummer;

Auf.P.Position # vPos;

// Auf.P.Best.Nummer # cnvai(Auf.P.Position,_FmtNumLeadZero,0,2); RICHTER Logik

Auf.P.MEH.Preis # Set.Auf.MEH.PEH;

Auf.P.PEH # Set.Auf.PEH;

Auf.P.MEH.Wunsch # Auf.P.MEH.Preis;

Auf.P.Warengruppe # Set.Auf.Warengruppe;

Auf.P.Wgr.Dateinr # Set.Auf.Dateinr; // 2023-04-25 AH ff.

Erx # RecLink(819,401,1,0); // Warengruppe holen

if (Erx>_rLockeD) then RecBufClear(819)

else Auf.P.Wgr.Dateinr # Wgr.Dateinummer;

Auf_Data:SetWgrDateinr(Auf.P.Wgr.Dateinr);

Auf.P.Auftragsart # Set.Auf.Auftragsart;

Auf.P.Termin1W.Art # Set.Auf.TerminArt;

\$edAuf.P.Artikelnr_Mat->wpcaption # ";

if (Set.Auf.PosText=999) then begin

\$cb.Text1->wpcheckstate # _WinStateChkUnchecked;

\$cb.Text2->wpcheckstate # _WinStateChkUnchecked;

\$cb.Text3->wpcheckstate # _WinStateChkchecked;

Auf.P.TextNr1 # 401;

Auf.P.TextNr2 # 0;

\$edAuf.P.TextNr2->wpCaptionInt # 0;

\$edAuf.P.TextNr2b->wpCaptionInt # 0;

end

else begin

\$cb.Text1->wpcheckstate # _WinStateChkUnchecked;

\$cb.Text2->wpcheckstate # _WinStateChkchecked;

\$cb.Text3->wpcheckstate # _WinStateChkUnchecked;

Auf.P.TextNr1 # 0;

Auf.P.TextNr2 # 0;

\$edAuf.P.TextNr2->wpCaptionInt # 0;

\$edAuf.P.TextNr2b->wpCaptionInt # Set.Auf.Posttext;

end;

vTxt # \$Auf.P.TextEditKopf->wpdbTextBuf;

```
if (vTxt<>0) then TextClear(vTxt);  
  
$Auf.P.TextEditKopf->WinUpdate(_WinUpdBuf2Obj);  
  
vTxt # $Auf.P.TextEditFuss->wpdbTextBuf;  
  
if (vTxt<>0) then TextClear(vTxt);  
  
$Auf.P.TextEditFuss->WinUpdate(_WinUpdBuf2Obj);
```

```
Erx # RekLink(814,400,8,_RecFirst);  
  
$lb.WAE1->wpcaption # "Wae.Kürzel";  
  
$lb.WAE2->wpcaption # "Wae.Kürzel";  
  
$lb.WAE3->wpcaption # "Wae.Kürzel";  
  
$lb.WAE4->wpcaption # "Wae.Kürzel";  
  
$lb.WAE5->wpcaption # "Wae.Kürzel";
```

```
if (w_AppendNr=0) then begin  
  
    $NB.Page1->wpdisabled # y;  
  
    $NB.Page2->wpdisabled # y;  
  
    $NB.Page3->wpdisabled # y;  
  
    $NB.Page4->wpdisabled # y;  
  
    $NB.Page5->wpdisabled # y;  
  
    $NB.Page5->wpdisabled # y;  
  
end
```

```
else begin  
  
    Mode # c_ModeNew2;  
  
    $NB.Page1->wpdisabled # n;  
  
    $NB.Page2->wpdisabled # n;
```

\$NB.Page3->wpdisabled # n;

\$NB.Page4->wpdisabled # n;

\$NB.Page5->wpdisabled # n;

// 17.05.2016 AH: damit bei Append die richtige Maske kommt

Auf.P.Warengruppe # Set.Auf.Warengruppe;

Auf.P.Wgr.Dateinr # Set.Auf.Dateinr; // 2023-04-25 AH ff.

Erx # RekLink(819,401,1,0); // Warengruppe holen

if (Erx<=_rLocked) then Auf.P.Wgr.Dateinr # Wgr.Dateinummer;

Auf_Data:SetWgrDateinr(Auf.P.Wgr.Dateinr);

Auf_P_SMain:Switchmask(false);

vHdl # gMdi->Winsearch('NB.Main');

vHdl->wpcurrent # 'NB.Page1';

end;

if (w_Command='NimmMatFuerBA') then begin

Erx # RecRead(200,0,_RecId,cnvia(w_Cmd_Para));

if (Erx<=_rLocked) then begin

Erx # RecLink(100,200,4,_recFirst); // Lieferant holen

if (Erx<=_rLocked) then begin

Auf.Kundennr # Adr.Kundennr;

RefreshIfm('edAuf.Kundennr',y);

end;

end


```

else begin

    w_Command # ";

    w_Cmd_Para # "

end;

end;

end; // neue Kopf+Pos


// Sonderfunktion:

RunAFX('Auf.P.Reclnit.Post','');


RETURN;

end;


//=====

// RecSave

//      vor Speicherung

//=====

sub RecSave() : logic;

local begin

    Erx      : int;

    vNummer  : int;

    vPos     : int;

    vOk      : logic;

```

```

vAlteMenge : float;

vAlteOffen : float;

vAltArtikel : alpha;

vHdl      : int;

vTxt      : int;

vKLim     : float;

vKreis    : alpha;

vBuf      : int;

vBuf2     : int;

vPage     : int;

vI        : int;

vBuf401   : int;

vBuf100   : int;

vRmtData  : alpha;

end;

begin

    vPage # gMdi->Winsearch('NB.Main');

    // logische Prüfung

    if(Lib_Pflichtfelder:PflichtfelderPruefenVorSpeichern() = false) then

        RETURN false;

    if ((Mode=c_ModeEdit) or (Mode=c_ModeEdit2) or (Mode=c_ModeNew2)) and

        (Set.LyseErweitertYN) then begin

        if(Lib_Pflichtfelder:PflichtfelderPruefenVorSpeichern('Lys.Maske2') = false) then

            RETURN false;

```

end;

if (Auf.Vorgangstyp='') then begin

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

vPage->wpcurrent # 'NB.Kopf';

\$edAuf.Vorgangstyp->WinFocusSet(true);

RETURN false;

end;

if (Auf.Vorgangstyp!=c_BOGUT and Auf.Vorgangstyp<>c_VorlageAuf and

Auf.Vorgangstyp != c_Ang and Auf.Vorgangstyp != c_Auf and Auf.Vorgangstyp != c_REKOR and

Auf.Vorgangstyp != c_Bel_KD and Auf.Vorgangstyp != c_GUT and Auf.Vorgangstyp != c_Bel_LF) then

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

vPage->wpcurrent # 'NB.Kopf';

\$edAuf.Vorgangstyp->WinFocusSet(true);

RETURN false;

end;

if(Auf.LiefervertragYN = true) then begin // Liefervertrag

if("Auf.GültigkeitVom" = 00.00.0000) then begin

Msg(001200,Translate('Gültigkeit Vom'),0,0,0);

vPage->wpcurrent # 'NB.Kopf';

\$edAuf.GltigkeitVom->WinFocusSet(true);

RETURN false;

end;

```
if("Auf.GültigkeitBis" = 00.00.0000) then begin  
    Msg(001200,Translate('Gültigkeit Bis'),0,0,0);  
    vPage->wpcurrent # 'NB.Kopf';  
    $edAuf.GltigkeitBis->WinFocusSet(true);  
    RETURN false;  
end;  
end;
```

```
If (Auf.Kundennr=0) then begin  
    Msg(001200,Translate('Kunde'),0,0,0);  
    vPage->wpcurrent # 'NB.Kopf';  
    $edAuf.Kundennr->WinFocusSet(true);  
    RETURN false;  
end;
```

```
Erx # RecLink(100,400,1,_RecFirst);    // Kunde holen
```

```
If (Erx>_rLocked) then begin  
    Msg(001201,Translate('Kunde'),0,0,0);  
    vPage->wpcurrent # 'NB.Kopf';  
    $edAuf.Kundennr->WinFocusSet(true);  
    RETURN false;  
end;
```

```
if ((Auf.Vorgangstyp=c_BOGUT) or (Auf.Vorgangstyp=c_AUF) or (Auf.Vorgangstyp=c_REKOR)) and  
    (Adr.SperrKundeYN) then begin  
    Msg(100005,Adr.Stichwort,0,0,0);
```

```
vPage->wpcurrent # 'NB.Kopf';

$edAuf.Kundennr->WinFocusSet(true);

RETURN false;

end;

if (Auf.Vorgangstyp=c_GUT) and (Adr.SperrLieferantYN) then begin

    Msg(100006,Adr.Stichwort,0,0,0);

    vPage->wpcurrent # 'NB.Kopf';

    $edAuf.Kundennr->WinFocusSet(true);

    RETURN false;

end;


If (Auf.Lieferadresse=0) then begin

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

    vPage->wpcurrent # 'NB.Kopf';

    $edAuf.Lieferadresse->WinFocusSet(true);

    RETURN false;

end;

If (Auf.Lieferanschrift=0) then begin

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

    vPage->wpcurrent # 'NB.Kopf';

    $edAuf.Lieferanschrift->WinFocusSet(true);

    RETURN false;

end;

Erx # RecLink(101,400,2,_recTest);

If (Erx>_rLocked) then begin

    Msg(001201,Translate('Lieferanschrift'),0,0,0);
```

```

vPage->wpcurrent # 'NB.Kopf';

$edAuf.Lieferadresse->WinFocusSet(true);

RETURN false;

end;

If (Auf.Verbraucher<>0) then begin

    Erx # RecLink(100,400,3,_recTest);    // Verbraucher holen

    If (Erx>_rLocked) then begin

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

        vPAge->wpcurrent # 'NB.Kopf';

        $edAuf.Verbraucher->WinFocusSet(true);

        RETURN false;

    end;

end;

If (Auf.Rechnungsempf=0) then begin

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

    vPAge->wpcurrent # 'NB.Kopf';

    $edAuf.Rechnungsempf->WinFocusSet(true);

    RETURN false;

end;

Erx # RecLink(100,400,4,_recFirst);    // Rechnungsempfänger holen

If (Erx>_rLocked) then begin

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

    vPage->wpcurrent # 'NB.Kopf';

    $edAuf.Rechnungsempf->WinFocusSet(true);

    RETURN false;

```

end;

If (Auf.Rechnungsanschr=0) then begin

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

vPage->wpcurrent # 'NB.Kopf';

\$edAuf.Rechnungsanschr->WinFocusSet(true);

RETURN false;

end;

vBuf100 # Adr_Data:HoleBufferAdrOderAnschrift(Auf.Rechnungsempf, Auf.Rechnungsanschr);

If (vBuf100=0) then begin

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

vPage->wpcurrent # 'NB.Kopf';

\$edAuf.Rechnungsanschr->WinFocusSet(true);

RETURN false;

end;

RecBufDestroy(vBuf100);

if (Auf.Vorgangstyp=C_GUT) or (Auf.Vorgangstyp=C_BEL_LF) then begin

If (Adr.Lieferantennr=0) then begin

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

vPage->wpcurrent # 'NB.Kopf';

\$edAuf.Rechnungsempf->WinFocusSet(true);

RETURN false;

end;

end;

if ((Auf.Vorgangstyp=c_BOGUT) or (Auf.Vorgangstyp=c_AUF) or (Auf.Vorgangstyp=c_REKOR)) and

(Adr.SperrKundeYN) then begin

```

Msg(100005,Adr.Stichwort,0,0,0);

vPage->wpcurrent # 'NB.Kopf';

$edAuf.Rechnungsempf->WinFocusSet(true);

RETURN false;

end;

if (Auf.Vorgangstyp=c_GUT) and (Adr.SperrLieferantYN) then begin

Msg(100006,Adr.Stichwort,0,0,0);

vPage->wpcurrent # 'NB.Kopf';

$edAuf.Rechnungsempf->WinFocusSet(true);

RETURN false;

end;


If ("Auf.Währung "=0) then begin

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

vPage->wpcurrent # 'NB.Kopf';

$edAuf.Waehrung->WinFocusSet(true);

RETURN false;

end;

Erx # RecLink(814,400,8,_recTest);

If (Erx>_rLocked) then begin

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

vPAge->wpcurrent # 'NB.Kopf';

$edAuf.Waehrung->WinFocusSet(true);

RETURN false;

end;

```


If (Auf.Lieferbed=0) then begin

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

vPage->wpcurrent # 'NB.Kopf';

\$edAuf.Lieferbed->WinFocusSet(true);

RETURN false;

end;

Erx # RecLink(815,400,5,_recTest);

If (Erx>_rLocked) then begin

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

vPAge->wpcurrent # 'NB.Kopf';

\$edAuf.Lieferbed->WinFocusSet(true);

RETURN false;

end;

if (Mode=c_ModeNew) or (Mode=c_ModeNew2) then begin // 2023-08-14 AH

If (Lib.SperreNeuYN) then begin

Msg(815001,"",0,0,0);

vPage->wpcurrent # 'NB.Kopf';

\$edAuf.Lieferbed->WinFocusSet(true);

RETURN false;

end;

end;

If (Auf.Zahlungsbed=0) then begin

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

vPage->wpcurrent # 'NB.Kopf';

```

$edAuf.Zahlungsbed->WinFocusSet(true);

RETURN false;

end;

Erx # RecLink(816,400,6,_recTest);

If (Erx>_rLocked) then begin

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

    vPage->wpcurrent # 'NB.Kopf';

    $edAuf.Zahlungsbed->WinFocusSet(true);

    RETURN false;

end;

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

    If (ZaB.SperreNeuYN) then begin

        Msg(816003,"",0,0,0);

        vPage->wpcurrent # 'NB.Kopf';

        $edAuf.Zahlungsbed->WinFocusSet(true);

        RETURN false;

    end;

end;

If ("Auf.Steuerschlüssel"=0) then begin

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

    vPage->wpcurrent # 'NB.Kopf';

    $edAuf.Steuerschluessel->WinFocusSet(true);

    RETURN false;

end;

Erx # RecLink(813,400,19,_recTest);

```

If (Erx>_rLocked) or ("Auf.Steuerschlüssel">999) then begin

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

vPage->wpcurrent # 'NB.Kopf';

\$edAuf.Steuerschluessel->WinFocusSet(true);

RETURN false;

end;

if (StS.UstIDPflichtYN) and (Adr.USIdentNr=") then begin

Msg(400023,",_WinIcoWarning,_WinDialogOk,1);

end;

If (Auf.Versandart=0) then begin

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

vPage->wpcurrent # 'NB.Kopf';

\$edAuf.Versandart->WinFocusSet(true);

RETURN false;

end;

Erx # RecLink(817,400,7,_recTest);

If (Erx>_rLocked) then begin

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

vPage->wpcurrent # 'NB.Kopf';

\$edAuf.Versandart->WinFocusSet(true);

RETURN false;

end;

If (Auf.Datum=0.0.0) then begin

```
Msg(001200,Translate('Bestelldatum'),0,0,0);  
  
vPage->wpcurrent # 'NB.Kopf';  
  
$edAuf.Datum->WinFocusSet(true);  
  
RETURN false;  
  
end;
```

```
If (Auf.Sprache='') then begin  
  
Msg(001200,Translate('Sprache'),0,0,0);  
  
vPage->wpcurrent # 'NB.Kopf';  
  
$edAuf.Sprache->WinFocusSet(true);  
  
RETURN false;  
  
end;
```

```
If (Auf.Sachbearbeiter='') then begin  
  
Msg(001200,Translate('Sachbearbeiter'),0,0,0);  
  
vPage->wpcurrent # 'NB.Kopf';  
  
$edAuf.Sachbearbeiter->WinFocusSet(true);  
  
RETURN false;  
  
end;
```

```
// Vertreter 1 prüfen...
```

```
if (Auf.Vertreter<>0) then begin  
  
Erx # RekLink(110,400,20,_RecFirst);  
  
if (Erx>_rLocked) then begin  
  
Msg(001201,Translate('Vertreter'),0,0,0);
```

```

vPage->wpcurrent # 'NB.Kopf';

$edAuf.Vertreter1->WinFocusSet(true);

RETURN false;

end;

if (Ver.SperreYN) then begin

    Msg(110000,Ver.Stichwort,0,0,0);

    vPage->wpcurrent # 'NB.Kopf';

    $edAuf.Vertreter1->WinFocusSet(true);

    RETURN false;

end;

end;

// Vertreter 2 prüfen...

if (Auf.Vertreter2<>0) then begin

    Erx # RekLink(110,400,21,_RecFirst);

    if (Erx>_rLocked) then begin

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

        vPage->wpcurrent # 'NB.Kopf';

        $edAuf.Vertreter2->WinFocusSet(true);

        RETURN false;

    end;

    if (Ver.SperreYN) then begin

        Msg(110000,Ver.Stichwort,0,0,0);

        vPage->wpcurrent # 'NB.Kopf';

        $edAuf.Vertreter2->WinFocusSet(true);

        RETURN false;

    end;

end;

```

end;

if (Wgr_Data:IstMixMat(Auf.P.Wgr.Dateinr)) then begin

Auf.P.Artikelnr # \$edAuf.P.Artikelnr_Mat->wpcaption;

end

else begin

Auf.P.Strukturnr # StrCut(\$edAuf.P.Artikelnr_Mat->wpcaption,1,20);

end;

// Positions Logik

if (Mode=c_ModeEdit) or (Mode=c_ModeEdit2) or (Mode=c_ModeNew2) then begin

if (Auf.Vorgangstyp=c_BOGUT) or (Auf.Vorgangstyp=c_REKOR) or (Auf.Vorgangstyp=c_GUT) then begin

if (Auf.P.Menge>0.0) then Auf.P.Menge # (-1.0) * Auf.P.Menge;

if (Auf.P.Menge.Wunsch>0.0) then Auf.P.Menge.Wunsch # (-1.0) * Auf.P.Menge.Wunsch;

if (Auf.P.Gewicht>0.0) then Auf.P.Gewicht # (-1.0) * Auf.P.Gewicht;

if ("Auf.P.Stückzahl">0) then "Auf.P.Stückzahl" # (-1) * "Auf.P.Stückzahl";

\$edAuf.P.Menge.Wunsch->winupdate(_WinUpdFld2Obj);

\$edAuf.P.Menge->winupdate(_WinUpdFld2Obj);

\$edAuf.P.Stueckzahl_Mat->winupdate(_WinUpdFld2Obj);

\$edAuf.P.Gewicht_Mat->winupdate(_WinUpdFld2Obj);

end;

if (Auf.Vorgangstyp=c_BEL_KD) or (Auf.Vorgangstyp=c_BEL_LF) then begin

if (Auf.P.Menge<0.0) then Auf.P.Menge # (-1.0) * Auf.P.Menge;

if (Auf.P.Menge.Wunsch<0.0) then Auf.P.Menge.Wunsch # (-1.0) * Auf.P.Menge.Wunsch;

if (Auf.P.Gewicht<0.0) then Auf.P.Gewicht # (-1.0) * Auf.P.Gewicht;

```

if ("Auf.P.Stückzahl"<0) then "Auf.P.Stückzahl" # (-1) * "Auf.P.Stückzahl";

$edAuf.P.Menge.Wunsch->winupdate(_WinUpdFld2Obj);

$edAuf.P.Menge->winupdate(_WinUpdFld2Obj);

$edAuf.P.Stueckzahl_Mat->winupdate(_WinUpdFld2Obj);

$edAuf.P.Gewicht_Mat->winupdate(_WinUpdFld2Obj);

end;

```

```

Auf.P.Einzelpreis # Auf.P.Grundpreis + Auf.P.Aufpreis;

Auf.P.Gesamtpreis # Auf_data:SumGesamtpreis(Auf.P.Menge, "Auf.P.Stückzahl" , Auf.P.Gewicht);

If (Auf.P.Auftragsart=0) then begin

    Lib_Guicom2:InhaltFehlt('Vorgangsart', 'NB.Page1', 'edAuf.P.Auftragsart');

    RETURN false;

end;

Erx # RecLink(835,401,5,_RecTest);

If (Erx>_rLocked) then begin

    Lib_Guicom2:InhaltFalsch('Vorgangsart', 'NB.Page1', 'edAuf.P.Auftragsart');

    RETURN false;

end;

```

```

If (Auf.P.Warengruppe=0) then begin

    Lib_Guicom2:InhaltFehlt('Warengruppe', 'NB.Page1', 'edAuf.P.Wrengruppe');

    RETURN false;

end;

Erx # RecLink(819,401,1,_RecFirst);

If (Erx>_rLocked) then begin

    Lib_Guicom2:InhaltFalsch('Warengruppe', 'NB.Page1', 'edAuf.P.Wrengruppe');

```

```

RETURN false;

end;

// If (Auf.P.Verwiegun=0) then begin
//   Lib_Guicom2:InhaltFehlt('Warengruppe', 'NB.Page1', 'edAuf.P.Wrengruppe');
//   RETURN false;
// end;

// 11.02.2014 AH:
Auf_Data:SumEKGesamtPreis();

// Rechnungsnr. prüfen...
if (Auf.Vorgangstyp=c_REKOR) and (Auf.P.AbrufAufNr=0) then begin
  //if (Dbalicense(_Dbalicense)<>'TA152658MN') then begin    // nicht bei Holzrichter VFP
//   if (Set.Installname='RSW') or (Set.Installname='HOWVFP') then begin    // nicht bei Holzrichter VFP u
// 12.10.2020 AH: für alle Pflichtfeld
  Lib_Guicom2:InhaltFehlt('Rechnungsnr.', 'NB.Page1', 'edAuf.P.AbrufAufNr');
  RETURN false;
//   end;
end;

if ((Auf.Vorgangstyp=c_REKOR) or (Auf.Vorgangstyp=c_GUT)) and
(Auf.P.AbrufAufNr<>0) then begin
  Erl.Rechnungsnr # Auf.P.AbrufAufNr;
  Erx # RecRead(450,1,0); // Erlöse holen

```



```

if (Erx>_rLocked) then begin

    Lib_Guicom2:InhaltFalsch('Rechnungsnr.', 'NB.Page1', 'edAuf.P.AbrufAufNr');

    RETURN false;

end;

if (Auf.P.AbrufAufPos<>0) then begin

    RecBufClear(451);

    Erl.K.Rechnungsnr  # Auf.P.AbrufAufNr;

// 17.05.2022 AH      Erl.K.Rechnungspos  # Auf.P.AbrufAufPos;

//      Erx # RecRead(451,4,0); // Erlösekonto holen

    Erl.K.lfdnr # Auf.P.AbrufAufPos;

    Erx # RecRead(451,1,0); // Erlösekonto holen

    if (Erx>_rMultikey) then begin

        Lib_Guicom2:InhaltFalsch('Rechnungsnr.', 'NB.Page1', 'edAuf.P.AbrufAufPos');

        RETURN false;

    end;

end;

end;

```

```

if (Auf.AbrufYN) or (Auf.PAbrufYN) then begin

    if ((Auf.P.AbrufAufNr <> 0) or (Auf.P.AbrufAufPos <> 0)) then begin // wenn Rahmen angegeben PRUEFEN

        vOK # true;

        vBuf # RekSave(401);

        vBuf -> Auf.P.Nummer  # Auf.P.AbrufAufNr;

        vBuf -> Auf.P.Position # Auf.P.AbrufAufPos;

        Erx # RecRead(vBuf, 1, 0); // gibt es den Auftrag?
    end;
end;

```

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

```
  vBuf2 # RekSave(400);
```

```
  Erx # RecLink(vBuf2, vBuf, 3, _recFirst); // Kopf lesen
```

```
  if(vBuf2 -> Auf.LiefervertragYN = false) then // angegebender Auftrag Liefervertrag?
```

```
    vOK # false;
```

```
  RecBufDestroy(vBuf2);
```

```
  if(vBuf -> "Auf.P.Löschmarker" = '*') and (Mode = c_ModeNew2) then // Position geloescht?
```

```
    vOK # false;
```

```
end
```

```
else
```

```
  vOK # false;
```

```
  RecBufDestroy(vBuf);
```

```
end
```

```
else // Abruf OHNE RAHMENNR???
```

```
  vOK # false;
```

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

```
  Lib_Guicom2:InhaltFehlt('Abruf', 'NB.Page1', 'edAuf.P.AbrufAufNr');
```

```
  RETURN false;
```

```
end;
```

```
end;
```

// Artikeldatei?

if (Wgr_Data:IstArt(Auf.P.Wgr.Dateinr)) or (Wgr_Data:IstMixArt(Auf.P.Wgr.Dateinr)) then begin

 If (Auf.P.Menge.Wunsch=0.0) then begin

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

 vPage->wpcurrent # 'NB.Page1';

 \$edAuf.P.Menge.Wunsch->WinFocusSet(true);

 RETURN false;

 end;

 If (Auf.P.Artikelnr='') then begin

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

 vPage->wpcurrent # 'NB.Page1';

 \$edAuf.P.Artikelnr->WinFocusSet(true);

 RETURN false;

 end;

 Erx # RecLink(250,401,2,_RecTest);

 If (Erx>_rLocked) then begin

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

 vPage->wpcurrent # 'NB.Page1';

 \$edAuf.P.Artikelnr->WinFocusSet(true);

 RETURN false;

 end;

if (Wgr_Data:IstMixArt(Auf.P.Wgr.Dateinr)) and ("Auf.P.Güte=") then begin

 Msg(001200,Translate('Güte'),0,0,0);

```
vPage->wpcurrent # 'NB.Page1';  
  
$edAuf.P.Guete->WinFocusSet(true);  
  
RETURN false;  
  
end;
```

```
If (Lib_Einheiten:CheckMEH(var Auf.P.MEH.Wunsch)=false) then begin  
  
    Lib_Guicom2:InhaltFalsch('Mengeneinheit', 'NB.Page1', 'edAuf.P.MEH.Wunsch');  
  
    RETURN false;  
  
end;
```

```
If (Lib_Einheiten:CheckMEH(var Auf.P.MEH.Preis)=false) then begin  
  
    Lib_Guicom2:InhaltFalsch('Mengeneinheit', 'NB.Page1', 'edAuf.P.MEH.Preis');  
  
    RETURN false;  
  
end;
```

```
If (Auf.P.Termin1Wunsch=0.0.0) then begin  
  
    Msg(001200,Translate('Wunschtermin'),0,0,0);  
  
    vPage->wpcurrent # 'NB.Page1';  
  
    $edAuf.P.Termin1Wunsch->WinFocusSet(true);  
  
    RETURN false;  
  
end;
```

```
If (Auf.P.Termin1W.Art='') then begin  
  
    Msg(001200,Translate('Wunschtermin'),0,0,0);  
  
    vPage->wpcurrent # 'NB.Page1';  
  
    $edAuf.P.Termin1W.Art->WinFocusSet(true);  
  
    RETURN false;
```

end;

If (Auf.P.Warengruppe=0) then begin

Lib_GuiCom:Pflichtfeld(\$edAuf.P.Warengruppe);

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

vPage->wpcurrent # 'NB.Page2';

\$edAuf.P.Warengruppe->WinFocusSet(true);

RETURN false;

end;

end; // Artikeldatei

// Materialdatei?

if (Wgr_Data:IstMat(Auf.P.Wgr.Dateinr)) or (Wgr_Data:IstMixMat(Auf.P.Wgr.Dateinr)) then begin

if (Auf.P.MEH.Einsatz="") then Auf.P.MEH.Einsatz # 'kg';

if (Auf.P.MEH.Wunsch="") then Auf.P.MEH.Wunsch # 'kg';

if (Auf.P.MEH.Wunsch='kg') then Auf.P.Menge.Wunsch # Rnd(Auf.P.Gewicht, Set.Stellen.Menge)

else if (Auf.P.MEH.Wunsch='t') then Auf.P.Menge.Wunsch # Rnd(Auf.P.Gewicht / 1000.0, Set.Stellen

else if (Auf.P.MEH.Wunsch='Stk') then Auf.P.Menge.Wunsch # cnvfi("Auf.P.Stückzahl")

else if (Auf.P.MEH.Wunsch=Auf.P.MEH.Einsatz) then Auf.P.Menge.Wunsch # Auf.P.Menge

else Auf.P.Menge.Wunsch # Lib_Einheiten:WandleMEH(401, "Auf.P.Stückzahl", Auf.P.Gewicht, Auf.P.

If ("Auf.P.MEH.Preis=") then begin

Lib_GuiCom:Pflichtfeld(\$edAuf.P.MEH.Preis_Mat);

```
Msg(001200,Translate('Preiseinheit'),0,0,0);  
vPage->wpcurrent # 'NB.Page1';  
$edAuf.P.MEH.Preis_Mat->WinFocusSet(true);  
RETURN false;  
end;
```

```
If ("Auf.P.Güte=") then begin  
Msg(001200,Translate('Güte'),0,0,0);  
vPage->wpcurrent # 'NB.Page1';  
$edAuf.P.Guete_Mat->WinFocusSet(true);  
RETURN false;  
end;
```

```
If (Auf.P.Erzeuger<>0) then begin  
Erx # RecLink(100,401,10,_recTest);  
If (Erx>_rLocked) then begin  
Msg(001201,Translate('Erzeuger'),0,0,0);  
vPage->wpcurrent # 'NB.Page1';  
$edAuf.P.Erzeuger_Mat->WinFocusSet(true);  
RETURN false;  
end;  
end;
```

```
If (Auf.P.Gewicht=0.0) then begin  
Msg(001200,Translate('Gewicht'),0,0,0);  
vPage->wpcurrent # 'NB.Page1';
```

```
$edAuf.P.Gewicht_Mat->WinFocusSet(true);
```

```
RETURN false;
```

```
end;
```

```
// 07.04.2017 aH    If ($edAuf.P.Menge_Mat->wpvisible) and
```

```
if (Auf.P.Menge=0.0) then begin
```

```
Msg(001200,Translate('Menge'),0,0,0);
```

```
vPage->wpcurrent # 'NB.Page1';
```

```
$edAuf.P.Menge_Mat->WinFocusSet(true);
```

```
RETURN false;
```

```
end;
```

```
If (Auf.P.PEH=0) then begin
```

```
Msg(001200,Translate('Preiseinheit'),0,0,0);
```

```
vPage->wpcurrent # 'NB.Page1';
```

```
$edAuf.P.PEH_Mat->WinFocusSet(true);
```

```
RETURN false;
```

```
end;
```

```
If (Lib_Einheiten:CheckMEH(var Auf.P.MEH.Preis)=false) then begin
```

```
Lib_Guicom2:InhaltFalsch('Mengeneinheit', 'NB.Page1', 'edAuf.P.MEH.Preis_Mat');
```

```
RETURN false;
```

```
end;
```

```
If (Auf.P.Termin1Wunsch=0.0.0) then begin
```

```
Msg(001200,Translate('Wunschtermin'),0,0,0);
```

```
vPage->wpcurrent # 'NB.Page1';  
  
$edAuf.P.Termin1Wunsch_Mat->WinFocusSet(true);  
  
RETURN false;  
  
end;
```

```
If (Auf.P.Termin1W.Art='') then begin  
  
Msg(001200,Translate('Wunschtermin'),0,0,0);  
  
vPage->wpcurrent # 'NB.Page1';  
  
$edAuf.P.Termin1W.Art_Mat->WinFocusSet(true);  
  
RETURN false;  
  
end;
```

```
If ("Auf.P.MEH.Preis=") then begin  
  
Lib_GuiCom:Pflichtfeld($edAuf.P.MEH.Preis_Mat);  
  
Msg(001200,Translate('Preiseinheit'),0,0,0);  
  
vPage->wpcurrent # 'NB.Page1';  
  
$edAuf.P.MEH.Preis_Mat->WinFocusSet(true);  
  
RETURN false;  
  
end;
```

```
If (Auf.P.Warengruppe=0) then begin  
  
Lib_GuiCom:Pflichtfeld($edAuf.P.Warengruppe);  
  
Msg(001200,Translate('Warengruppe'),0,0,0);  
  
vPage->wpcurrent # 'NB.Page2';  
  
$edAuf.P.Warengruppe->WinFocusSet(true);
```


RETURN false;

end;

If (Auf.P.Verwiegungsart<>0) then begin

 Erx # RecLink(818,401,9,_recTest);

 If (Erx>_rLocked) then begin

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

 vPage->wpcurrent # 'NB.Page3';

 \$edAuf.P.Verwiegungsart->WinFocusSet(true);

 RETURN false;

 end;

end;

If (Auf.P.Etikettentyp<>0) then begin

 Erx # RecLink(840,401,8,_recTest);

 If (Erx>_rLocked) then begin

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

 if (\$NB.Page3->wpvisible) then begin

 vPage->wpcurrent # 'NB.Page3';

 \$edAuf.P.Etikettentyp->WinFocusSet(true);

 end

 else begin

 vPage->wpcurrent # 'NB.Page5';

 \$edAuf.P.Etikettentyp2->WinFocusSet(true);

 end;

```
RETURN false;
```

```
end;
```

```
end;
```

```
end; // Material
```

```
if (Auf.P.MaterialNr<>0) and (Mode=c_ModeNew2) then begin
```

```
  Erx # Auf_Data:PasstAuf2Mat(Abs(Auf.P.MaterialNr),n,true);
```

```
  if (erx<0) then RETURN false;
```

```
  if (erx=0) then begin
```

```
    if (Rechte[Rgt_Auf_MATZ_Konf_Abm]=false) then begin
```

```
      Msg(401016,",_WinlcoError,_WinDialogOk,1);
```

```
      RETURN false;
```

```
    end;
```

```
  end;
```

```
end;
```

```
end; // Poslogik
```

```
// Hier erweiterte Meldungen bei falschen Daten
```

```
if (RunAFX('Auf.P.RecSave.Pre',")<0) then
```

```
  RETURN false;
```

// neuer Kopf? -> merken und in Position springen *****

// neuer Kopf? -> merken und in Position springen *****

// neuer Kopf? -> merken und in Position springen *****

if (Mode=c_ModeNew) then begin

if (vPage->wpcurrent='NB.Kopf') then begin

// Kreditlimit prüfen...

if (Auf.Vorgangstyp=c_AUF) and ("Set.KLP.Auf-Anlage"<>") then begin

if (Adr_K_Data:Kreditlimit(Auf.Rechnungsempf,"Set.KLP.Auf-anlage",y, var vKLim)=false) then begin

vPage->wpcurrent # 'NB.Kopf';

\$edAuf.Rechnungsempf->WinFocusSet(true);

RETURN false;

end;

end;

Mode # c_ModeNew2;

// Text vorbelegen?

if (Set.Auf.Kopftext<>0) then begin

Txt.Nummer # Set.Auf.Kopftext;

Erx # RecRead(837,1,0);

if (Erx<_rLocked) then begin

vHdl # \$Auf.P.TextEditKopf->wpdbTextBuf;

Lib_Texte:TxtLoadLangBuf('~837.'+CnvAl(Txt.Nummer,_FmtNumLeadZero | _FmtNumNoGroup,0,8

RunAFX('Auf.P.AusText', 'Kopf|'+aint(vHdl));

\$Auf.P.TextEditKopf->WinUpdate(_WinUpdBuf2Obj);

```

$Auf.P.TextEditKopf->wpcustom # myTmpText+'.401.K';

end;

end;

vTxt # 0;

RecLink(100,400,1,0); // Kunde holen

vTxt # Adr.VK.Fusstext;

if (Auf.Vorgangstyp=c_GUT) or (Auf.Vorgangstyp=c_Bel_LF) then

    vTxt # Adr.EK.Fusstext;

if (vTxt=0) then vTxt # Set.Auf.Fusstext;

if (vTxt<>0) then begin

    Txt.Nummer # vTxt;

    Erx # RecRead(837,1,0);

    if (Erx<_rLocked) then begin

        vHdl # $Auf.P.TextEditFuss->wpdbTextBuf;

        Lib_Texte:TxtLoadLangBuf('~837.'+CnvAI(Txt.Nummer,_FmtNumLeadZero | _FmtNumNoGroup,0,8

        RunAFX( 'Auf.P.AusText', 'Fuss|'+aint(vHdl));

        $Auf.P.TextEditFuss->WinUpdate(_WinUpdBuf2Obj);

        $Auf.P.TextEditFuss->wpcustom # myTmpText+'.401.F';

    end;

end;

```

```

$NB.Page1->wpdisabled # n;

$NB.Page2->wpdisabled # n;

$NB.Page3->wpdisabled # n;

```

```
$NB.Page4->wpdisabled # n;  
$NB.Page5->wpdisabled # n;  
vPage->wpcurrent # 'NB.Page1';
```

```
Auf.P.Best.Nummer # Auf.Best.Nummer; // RICHTER Logic AUS
```

```
$edAuf.P.Best.nummer->winupdate(_WinUpdFld2Obj);
```

```
$edAuf.P.Best.nummer_mat->winupdate(_WinUpdFld2Obj);
```

```
Refreshfm();
```

```
Refreshmode();
```

```
if ($NB.Page1->wpcustom='NB.Page1_Art') then
```

```
    $edAuf.P.Auftragsart->WinFocusSet(true)
```

```
else
```

```
    $edAuf.P.Auftragsart_Mat->WinFocusSet(true);
```

```
Reclnit(n); // neue Position belegen
```

```
RETURN false;
```

```
end;
```

```
end;
```

```
// neue Position? -> temporär sichern *****
```

```
// neue Position? -> temporär sichern *****
```

```
// neue Position? -> temporär sichern *****
```

```
if (Mode=c_ModeNew2) then begin
```

```
    Auf_P_Sub:CalcMengen();
```

```
Auf_data:SaveRabatt('*RAB1',$edRabatt1->wpcaptionfloat);
```

```
Auf_Data:SumAufpreise();
```

```
Auf_K_Data:SumKalkulation();
```

```
Erx # Auf_Data:PosInsert(0,'MAN');
```

```
if (Erx<>_rOk) then begin
```

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

```
    RETURN False;
```

```
end;
```

```
AufPTextSave();
```

```
Erx # Lib_MoreBufs:SaveAll(401)
```

```
if (Erx<>_rOK) then begin
```

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

```
    RETURN False;
```

```
end;
```

```
// Materialdatei und Kundenartikelnr?
```

```
if (Set.Auf.KdArtAnlegen='A') and (Auf.P.KundenArtNr<>") and
```

```
((Wgr_Data:IstMix(Auf.P.Wgr.Dateinr)) or (Wgr_Data:IstMat(Auf.P.Wgr.Dateinr))) then begin
```

```
Erx # RecLink(100,400,1,_recFirst); // Kunde holen
```

```
Adr.V.Adressnr    # Adr.Nummer;
```

```
Adr.V.KundenArtNr # Auf.P.KundenArtNr;
```

```
Erx # RecRead(105,2,0); // Verpackung testen
```

```

if (Erx>=_rNoKey) then begin
    if (Msg(401020, ", _WinIcoQuestion, _WinDialogYesNo, 2) =_WinIdYes) then
        Auf_Data:Auf2Verpackung(y);
    end;
end;

```

```

// auf 1. Seite 1. Feld positionieren

```

```

$NB.Main->wpcurrent # 'NB.Page1';
if ($NB.Page1->wpcustom='NB.Page1_Art') then
    $edAuf.P.Auftragsart->winfocusset(false)
else
    $edAuf.P.Auftragsart_Mat->winfocusset(false);

```

```

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

```

```

    vl # Dlg_Standard:PosErfassung(Set.Auf.Copy.AufprYN, Set.Auf.Copy.KalkYN);

```

```

    case (vl) of

```

```

        0    : Reclnit(n);
        1    : Reclnit(y);
        1+2  : Reclnit(y,y,n);
        1+4  : Reclnit(y,n,y);
        1+2+4 : Reclnit(y,y,y);

```

```

    end;

```

```

$RL.Aufpreise->winupdate(_winupdon,_WinLstFromFirst);

```

```

$RL.Aufpreise_Mat->winupdate(_winupdon,_WinLstFromFirst);

$lb.Position1->wpcaption # cnvai(Auf.P.Position,_fmtnumnogroup);

$lb.Position1_Mat->wpcaption # cnvai(Auf.P.Position,_fmtnumnogroup);

RETURN false;

end

else begin

    RETURN true;

end;

RETURN false;

end; // ModeNew2

// ganzen NEUEN Auftrag sichern *****

// ganzen NEUEN Auftrag sichern *****

// ganzen NEUEN Auftrag sichern *****

if (Mode=c_ModeList2) then begin

    vPos # Auf.P.Position;

    vOk # y;

    Erx # RecLink(401,400,9,_RecFirst);

    WHILE (Erx<=_rLocked) and (vOK) do begin

        if (Auf.P.Termin1Wunsch=0.0.0) then begin

            vok # n;

            BREAK;

        end;

        Erx # RecLink(401,400,9,_RecNext);

```


END;

Auf.P.Position # vPos;

Recread(401,1,0);

if (vOK=n) then begin

if (Msg(401003,",_WinIcoWarning,_WinDialogYesNo,2)=_WinIdNo) then RETURN false;

end

TRANSON; // Transaktionsstart

// Anhängen??

if (w_AppendNr<>0) then begin

Auf.P.Nummer # w_AppendNr+1;

Auf.P.Position # 1;

Recread(401,1,0);

RecRead(401,1,_recPrev);

if (Auf.P.Nummer<>w_AppendNr) then begin

TRANSBRK;

RETURN false;

end;

vNummer # w_AppendNr;

vPos # Auf.P.Position + 1;

end

else begin

// Nummernvergabe

vPos # 1;

vKreis # 'Auftrag';

```

if (Auf.Vorgangstyp=c_ANG) then vKreis # 'Angebot';

if (Auf.Vorgangstyp=c_VorlageAuf) then vKreis # 'Vorlageauftrag';

if ("Set.Auf.GutBel#SepYN") and

  ((Auf.Vorgangstyp=c_BOGUT) or (Auf.Vorgangstyp=c_REKOR) or (Auf.Vorgangstyp=c_GUT) or

  (Auf.Vorgangstyp=c_BEL_KD) or (Auf.Vorgangstyp=c_BEL_LF)) then

  vKreis # 'Auftrag-Gutschrift/Belastung';

vNummer # Lib_Nummern:ReadNummer(vKreis);

if (vNummer<>0) then Lib_Nummern:SaveNummer()

else begin

  TRANSBRK;

  RETURN false;

end;

end;

// Kopftexte kopieren

TxtRename(myTmpText+'.401.K','~401.'+CnvAI(vNummer,_FmtNumLeadZero | _FmtNumNoGroup,0,8))

TxtRename(myTmpText+'.401.F','~401.'+CnvAI(vNummer,_FmtNumLeadZero | _FmtNumNoGroup,0,8))


Erx # RecLink(401,400,9,_RecFirst | _Reclock);

WHILE (Erx=_rOk) do begin // Positionen umnummerieren

  // Texte ggf. umbenennen

  if (Auf.P.TextNr1=401) then begin // Individuell

    TxtRename(myTmpText+'.401.'+CnvAI(Auf.P.Position,_FmtNumLeadZero | _FmtNumNoGroup,0,3),'-

  end

```

else begin

 TxtDelete('~401.'+CnvAl(vNummer,_FmtNumLeadZero | _FmtNumNoGroup,0,8)+'.'+CnvAl(vPos,_Fm
end;

// Internen Text umbenennen

TxtRename(myTmpText+ '.401.'+CnvAl(Auf.P.Position,_FmtNumLeadZero | _FmtNumNoGroup,0,3)+'.

// Aufpreise kopieren

Erx # RecLink(403,401,6,_RecFirst);

WHILE (Erx=_rOK) do begin

 RecRead(403,1,_RecLock);

 Auf.Z.Nummer # vNummer;

 Auf.Z.Position # vPos;

 Erx # RekReplace(403,_recUnlock,'MAN');

 if (Erx<>_rOk) then begin

 TRANSBRK;

 Msg(401403,gTitle,0,0,0);

 RETURN False;

 end;

 Erx # RecLink(403,401,6,_RecFirst);

END;

 // Kalkulation ggf. umnummerieren

WHILE (RecLink(405,401,7,_recFirst)=_rOk) do begin

 RecRead(405,1,_Reclock);

 Auf.K.Nummer # vNummer;

```

Auf.K.Position # vPos;

Erx # Rekreplace(405,_recUnlock,'MAN');

if (Erx<>_rOk) then begin

    TRANSBRK;

    Msg(401405,gTitle,0,0,0);

    RETURN False;

end;

END;

```

// Ausführung ggf. umnummerieren

```

WHILE (RecLink(402,401,11,_recFirst)=_rOk) do begin

    RecRead(402,1,_Reclock);

    Auf.AF.Nummer # vNummer;

    Auf.AF.Position # vPos;

    Erx # Rekreplace(402,_recUnlock,'MAN');

    if (Erx<>_rOk) then begin

        TRANSBRK;

        Msg(401402,gTitle,0,0,0);

        RETURN False;

    end;

END;

```

// Feinabrufe ggf. umnummerieren

```

WHILE (RecLink(408,400,17,_recFirst)=_rOk) do begin

    RecRead(408,1,_Reclock);

    Auf.FA.Nummer # vNummer;

```

```

Auf.FA.Position # vPos;

Erx # Rekreplace(408,_recUnlock,'MAN');

if (Erx<>_rOk) then begin

    TRANSBRK;

    Msg(401408,gTitle,0,0,0);

    RETURN False;

end;

END;

```

// Stückliste ggf. umnummerieren

```

WHILE (RecLink(409,401,15,_recFirst)=_rOk) do begin

    RecRead(409,1,_Reclock);

    Auf.SL.Nummer # vNummer;

    Auf.SL.Position # vPos;

    Erx # Rekreplace(409,_recUnlock,'MAN');

    if (Erx<>_rOk) then begin

        TRANSBRK;

        Msg(401409,gTitle,0,0,0);

        RETURN False;

    end;

END;

```

```

Auf_P_Sub:CalcMengen();

Auf.P.Anlage.Datum # Today;

Auf.P.Anlage.Zeit # now;

```

```
Auf.P.Anlage.User # gUsername;
```

```
Lib_MoreBufs:ReadAll(401);
```

```
Lib_MoreBufs:Lock();
```

```
// 21.03.2018 AH:
```

```
vRmtData # Lib_RmtData:UserRead('400|'+aint(Auf.P.Position), y);
```

```
if (vRmtData<>") then
```

```
    Lib_RmtData:UserWrite('400|'+aint(vPos), vRmtData);
```

```
Auf.P.Nummer # vNummer; // Positionen umnummerieren
```

```
Auf.P.Position # vPos;
```

```
vPos # vPos + 1;
```

```
Auf.P.Aktionsmarker # 'N';
```

```
Erx # Auf_Data:PosReplace(_recUnlock,'MAN');
```

```
if (Erx<>_rOk) then begin
```

```
    Lib_MoreBufs:Unlock();
```

```
    TRANSBRK;
```

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

```
    RETURN False;
```

```
end;
```

```
Erx # Lib_MoreBufs:SaveAll(401);
```

```
if (erx<>_rOK) then begin
```

```
    TRANSBRK;
```

```
Msg(001000+Erx,gTitle+cPZeile,0,0,0);  
  
RETURN False;  
  
end;  
  
Auf.Nummer # myTmpNummer;  
  
Erx # RecLink(401,400,9,_RecFirst | _RecLock);  
  
END; // Positionen umbenennen
```

```
// Kopfaufpreise kopieren  
  
Erx # RecLink(403,400,13,_RecFirst);  
  
WHILE (Erx=_rOK) do begin  
  
    RecRead(403,1,_RecLock);  
  
    Auf.Z.Nummer # vNummer;  
  
    Erx # RekReplace(403,_recUnlock,'MAN');  
  
    if (Erx<>_rOk) then begin  
  
        TRANSBRK;  
  
        Msg(401403,gTitle,0,0,0);  
  
        RETURN False;  
  
    end;
```

```
Erx # RecLink(403,400,13,_RecFirst);  
  
END;
```

```
// nicht anhängen? dann Kopf anlegen
```

```

if (w_AppendNr=0) then begin

  Auf.Anlage.Datum # Today;

  Auf.Anlage.Zeit # now;

  Auf.Anlage.User # gUsername;

  Auf.Nummer # vNummer;      // Kopf sichern

  Erx # RekInsert(400,0,'MAN');

  if (Erx<>_rOk) then begin

    TRANSBRK;

    Auf.Nummer # myTmpNummer;

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

    RETURN False;

  end;

  if (Auf.Vorgangstyp=c_AUF) then begin

    Erx # RecLink(100,400,1,_recFirst); // Kunde holen

    RecRead(100,1,_recLock);

    Adr.Fin.letzterAufam # today;

    Erx # RekReplace(100,_recUnlock,'AUTO');

  end;

end

else begin

  RecLink(400,401,3,_recFirst); // Kopf holen

  $NB.Kopf->wpdisabled # n;

end;

w_AppendNr # 0;

```



```

FOR Erx # RecLink(401,400,9,_RecFirst)

LOOP Erx # RecLink(401,400,9,_RecNext)

WHILE (Erx<=_rLocked) do begin

    // nur NEUE Positionen

    if (Auf.P.Aktionsmarker<>'N') then CYCLE;

    // nötige Verbuchungen im Artikel druchführen...

    Auf_Data:VerbucheArt(",0.0,"Auf.P.Löschmarker");

    // 01.07.2020 AH: ggf. vorherige Userantwort holen

    vl # cnvia(Lib_RmtData:UserRead('400|'+aint(Auf.P.Position)+'|Mat', y));

    if (vl=0) then vl #Set.Auf.MatNr.Ablauf;

    // Dirket mit Material verbunden?

    if (Auf.P.MaterialNr<>0) then begin

        // Reservierung ist NEGATIV

        if (Auf.P.MaterialNr<0) then Mat.Nummer # 0 - Auf.P.MaterialNr

        else Mat.Nummer # Auf.P.MaterialNr;

    Erx # RecRead(200,1,0);

    if (Erx<=_rLocked) then begin

        // Mataktion ud Auftragsaktion für Angebot anlegen...

```

if (Auf.Vorgangstyp=c_ANG) then begin

if (Auf.P.MaterialNr>0) then begin

RecBufClear(404);

Auf.A.Aktionstyp # c_Akt_Angebot;

Auf.A.Bemerkung # c_AktBem_Angebot;

Auf.A.Aktionsnr # Auf.Nummer;

Auf.A.Aktionspos # Auf.P.Position;

Auf.A.Aktionsdatum # Today;

Auf.A.TerminStart # Today;

Auf.A.TerminEnde # Today;

Auf.A.MaterialNr # Mat.Nummer;

if (Wgr_Data:IstMix(Auf.P.Wgr.Dateinr)) or (Wgr_Data:IstArt(Auf.P.Wgr.Dateinr)) then

Auf.A.ArtikelNr # Mat.StrukturNr;

Auf_A_Data:NeuAnlegen();

// Auf.P.Aktionsmarker # '!';

RecBufClear(204);

Mat.A.Aktionsmat # Mat.Nummer;

Mat.A.Aktionstyp # c_Akt_Angebot;

Mat.A.Aktionsnr # Auf.Nummer;

Mat.A.Aktionspos # Auf.P.Position;

Mat.A.Bemerkung # c_AktBem_Angebot;

Mat.A.Aktionsdatum # today;

Mat.A.Terminstart # "Auf.GültigkeitVom";

```

Mat.A.Terminende  # "Auf.GültigkeitBis";

Mat.A.Adressnr    # 0;

Mat_A_Data:Insert(0,'AUTO')

end

// Reservieren...

else begin

    if (Auf_Data:ReservMat())=false) then begin

        TRANSBRK;

        Error(010021,AInt(Auf.P.Position)+'|'+AInt(Mat.Nummer));

        Auf.Nummer # myTmpNummer;

        ErrorOutput;

        RETURN False;

    end;

end;

end

else if (Auf.Vorgangstyp=c_AUF) then begin

    // Material reservieren...

    if (vl=2) then begin

        if (Auf_Data:ReservMat())=false) then begin

            TRANSBRK;

            Error(010021,AInt(Auf.P.Position)+'|'+AInt(Mat.Nummer));

            Auf.Nummer # myTmpNummer;

            ErrorOutput;

            RETURN False;

        end;

    end

end

```

```
// Material direkt VSB setzen...
```

```
else if (vl=3) then begin
```

```
  if (Auf_Data:MatzMat(Y)=false) then begin
```

```
    TRANSBRK;
```

```
    Auf.Nummer # myTmpNummer;
```

```
    ErrorOutput;
```

```
    RETURN False;
```

```
  end;
```

```
end;
```

```
// Material direkt fakturieren ...
```

```
else if (vl=4) then begin
```

```
  if (Auf_Data:DFaktMat(Mat.Nummer,n, today, now)=false) then begin
```

```
    TRANSBRK;
```

```
    Auf.Nummer # myTmpNummer;
```

```
    ErrorOutput;
```

```
    RETURN False;
```

```
  end;
```

```
end;
```

```
end;
```

```
end;
```

```
end;
```

```
RecRead(401,1,_recLock);
```

```
Auf.P.MaterialNr  # 0;
```

```
if (Auf.P.Aktionsmarker='N') then Auf.P.Aktionsmarker # '';
```

```
Erx # Auf_Data:PosReplace(_Recunlock,'AUTO');
```

```
if (Erx<>_rOk) then begin  
    TRANSBRK;  
    Auf.Nummer # myTmpNummer;  
    if (Erx<999) then Msg(001000+Erx,gTitle+cPZeile,0,0,0);  
    ErrorOutput;  
    RETURN False;  
end;
```

```
// ST 2012-11-28: Auftragsart erneut lesen, um später  
//           bei Entscheidung ob direkter AB Druck  
//           durchgeführt werden soll, zu prüfen  
Erx # RekLink(835,401,5,0);
```

```
// Sonderfunktion:
```

```
if (RunAFX('Auf.P.RecSave','')<>0) then begin  
    if (AfxRes<>_rOk) then begin  
        TRANSBRK;  
        Auf.Nummer # myTmpNummer;  
        Msg(001000+Erx,gTitle+cPZeile,0,0,0);  
        RETURN False;  
    end;  
end;
```

```
Lib_Workflow:Trigger(401, 401, _WOF_KTX_NEU);
```

```
// Abruf??
```

```
if ((Auf.AbrufYN) or (Auf.PAbrufYN)) and (Auf.P.AbrufAufNr<>0) then begin
```

```
  vOk # Auf_Data:VerbucheAbruf(y);
```

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

```
    TRANSBRK;
```

```
    ErrorOutput;
```

```
    Msg(401401,CnvAI(Auf.P.Position),0,0,0);
```

```
    RETURN false;
```

```
  end;
```

```
end; // Abruf
```

```
// 21.03.2018 AH:
```

```
vRmtData # Lib_RmtData:UserRead('400|'+aint(Auf.P.Position), y);
```

```
if (vRmtData<>") then begin
```

```
  Erx # RekLink(835,401,5,_RecFirst); // Auftragsart holen
```

```
  if (AAr.Berechnungsart >= 700) and (AAr.Berechnungsart <= 799) then begin
```

```
    Auf_Subs:CopyBAGanAuf(var Erx, cnvia(StrCut(vRmtData,1,10)), Auf.P.Nummer, Auf.P.Position);
```

```
  end;
```

```
end;
```

```
END; // posloop
```

```
TRANSOFF; // Transaktionsende
```

```
// Kreditlimit prüfen...
```

```
if (Auf.Vorgangstyp=c_AUF) then begin
```

```
  Auf_Data:SperrPruefung(0);
```

```
end;
```

```
// Sonderfunktion:
```

```
RunAFX('Auf.P.RecSave.Post','');
```

```
Lib_Workflow:Trigger(400, 400, _WOF_KTX_NEU);
```

```
// direkter Druck?
```

```
if (Set.Auf.SofortDABYN) and (Auf.Vorgangstyp=c_AUF) then begin
```

```
  // ST 2012-11-28 Projekt 1406/31:
```

```
  //   Bei Lohnaufträgen keine Auftragsbestätigung
```

```
  //   automatisch drucken, da noch nicht alle Daten
```

```
  //   eingegeben worden sein können
```

```
  if (AAr.Berechnungsart < 700) OR (AAr.Berechnungsart >= 800) then
```

```
    Auf_Sub:DruckAB();
```

```
end;
```

```
// Kopfmode??
```

```
if (gMDI=cDialog2) then begin
```

```

cZList2->WinUpdate(_WinUpdOn, _WinLstRecFromBuffer | _WinLstRecDoSelect);

Auf.Nummer # Auf.P.Nummer;

RecRead(400,1,0);

end;


if (w_Command='NimmMatFuerBA') then begin

    w_Command # '';

    w_TimerVar # 'NimmMatFuerBA|'+w_Cmd_Para;

    gTimer2 # SysTimerCreate(1000,1,gMdiAuf);

end;


RETURN true;

end; // kompletten Auftrag speichern

```

```

// temp.Position editieren ? -> temp. zurückspeichern *****
// temp.Position editieren ? -> temp. zurückspeichern *****
// temp.Position editieren ? -> temp. zurückspeichern *****

if (Mode=c_ModeEdit2) then begin

    Auf_data:SaveRabatt('*RAB1',$edRabatt1->wpcaptionfloat);

    Auf_Data:SumAufpreise();

    Auf_K_Data:SumKalkulation();

    Erx # Auf_Data:PosReplace(_recUnlock,'MAN');

```



```
if (Erx<>_rOk) then begin  
    Msg(001000+Erx,gTitle+cPZeile,0,0,0);  
    RETURN False;  
end;
```

```
Erx # Lib_MoreBufs:SaveAll(401);  
if (Erx<>_rOK) then begin  
    Msg(001000+Erx,gTitle+cPZeile,0,0,0);  
    RETURN False;  
end;
```

```
AufPTextSave();
```

```
RETURN true;  
end; // temp. Pos.
```

```
// Position editiert? -> zurückspeichern & protokollieren *****
```

```
// Position editiert? -> zurückspeichern & protokollieren *****
```

```
// Position editiert? -> zurückspeichern & protokollieren *****
```

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

```
//   if (Auf.P.MEH.Wunsch=$lb.Art.MEH5->wpcaption) then begin
```

```
//     Auf.P.Menge # Auf.P.Menge.Wunsch;
```

```
//   end;
```

// 2023-02-10 AH, Proj. 2465/58

if (Auf_P_Subst:CheckMehWechsel(Protokollbuffer[401]->Auf.P.MEH.Preis, Auf.P.MEH.Preis, true)=false

RETURN false;

end;

Auf_P_Subst:CalcMengen();

TRANSON;

Auf_data:SaveRabatt('*RAB1',\$edRabatt1->wpcaptionfloat);

Auf_Data:SumAufpreise(c_ModeSave); // 15.07.2020 AH: c_ModeSave!

Auf_K_Data:SumKalkulation();

Erx # Auf_Data:PosReplace(_recUnlock,'MAN');

if (Erx<>_rOk) then begin

TRANSBRK;

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

RETURN False;

end;

Erx # Lib_MoreBufs:SaveAll(401, true);

if (erx<>_rOK) then begin

TRANSBRK;

Msg(001000+erx,gTitle+cPZeile,0,0,0);

RETURN False;

end;

```
vBuf401 # RecBufCreate(401);
```

```
RecBufCopy(ProtokollBuffer[401],vBuf401);
```

```
// Mengenänderung?
```

```
vAlteMenge # ProtokollBuffer[401]->Auf.P.Menge;
```

```
vAlteOffen # ProtokollBuffer[401]->Auf.P.Menge - ProtokollBuffer[401]->Auf.P.Prd.Plan - ProtokollBuffer[401]->Auf.P.Lsg.Plan;
```

```
vAltArtikel # ProtokollBuffer[401]->Auf.P.ArtikelNr;
```

```
Erx # RekReplace(400,_recUnlock,'MAN');
```

```
if (Erx<>_rOk) then begin
```

```
    TRANSBRK;
```

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

```
    RETURN False;
```

```
end;
```

```
Auf_Data:VererbeKopfReferenzInPos(ProtokollBuffer[400]->Auf.Best.Nummer, Auf.Best.Nummer);
```

```
// 10.11.2021 AH
```

```
if (Auf.Lieferadresse<>ProtokollBuffer[400]->Auf.Lieferadresse) or (Auf.Lieferanschrift<>ProtokollBuffer[400]->Auf.Lieferanschrift) then begin
```

```
    if (Auf_Data:VeraenderteLieferadresse())=false) then begin
```

```
        TRANSBRK;
```

```
        Msg(400004,gTitle,0,0,0);
```

```
        Erroroutput;
```

```
        RETURN False;
```

```
    end;
```

```
end;
```

PtD_Main:Compare(401);

PtD_Main:Compare(400);

// nötige Verbuchungen im Artikel durchführen...

if (vAlteMenge<>Auf.P.Menge) or (vAltArtikel<>Auf.P.ArtikelNr) then begin

// 19.05.2022 AH Auf_Data:VerbucheArt(vAltArtikel, vAlteMenge, "Auf.P.Löschmarker");

 Auf_Data:VerbucheArt(vAltArtikel, vAlteOffen, "Auf.P.Löschmarker");

end;

// Text speichern

if (Auf.P.TextNr1<>401) then begin // NICHT Individuell ?

 TxtDelete('~401.'+CnvAI(vNummer,_FmtNumLeadZero | _FmtNumNoGroup,0,8)+'.'+CnvAI(Auf.P.Positi

end;

AufPTextSave();

vPos # Auf.P.Position;

Auf.P.Position # vPos;

RecRead(401,1,0);

// Sonderfunktion:

if (RunAFX('Auf.P.RecSave','')<>0) then begin

 if (AfxRes<>_rOk) then begin

 TRANSBRK;

 Msg(001000+AfxRes,gTitle+cPZeile,0,0,0);

 RETURN False;

end;

end;

// Abruf??

if ((Auf.AbrufYN) or (Auf.PAbrufYN)) and (Auf.P.AbrufAufNr<>0) then begin

 vOk # Auf_Data:VerbucheAbruf(n);

 if (vOK=false) then begin

 TRANSBRK;

 ErrorOutput;

 Msg(401401,CnvAI(Auf.P.Position),0,0,0);

 RETURN false;

 end;

end; // Abruf

TRANSOFF;

// Kreditlimit prüfen...

if (Auf.Vorgangstyp=c_AUF) then begin

 Auf_Data:SperrPruefung(vBuf401);

end;

RecBufDestroy(vBuf401);

// Sonderfunktion:

RunAFX('Auf.P.RecSave.Post','');

end; // Pos. editiert

```
if (gMDI=cDialog2) then begin
```

```
  cZList2->WinUpdate(_WinUpdOn, _WinLstRecFromBuffer | _WinLstRecDoSelect);
```

```
  Auf.Nummer # vNummer;
```

```
  RecRead(400,1,0);
```

```
end;
```

```
  RETURN true; // Speichern erfolgreich
```

```
end;
```

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

```
// RecCleanup
```

```
//      Aufräumen bei Abbruch der Erfassung
```

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

```
sub RecCleanup() : logic;
```

```
local begin
```

```
  Erx : int;
```

```
  vHdl : int;
```

```
end;
```

```
begin
```

```
if (Mode=c_ModeEdit) or (Mode=c_ModeEdit2) then begin
```

```
  Lib_MoreBufs:Unlock();
```

```
// diese veränderte Position verwerfen
```

```
PtD_Main:Forget(400);
```

```
RecRead(400,1,_RecUnlock);
```

```
if (Mode=c_modeEdit2) then begin
```

```
    $NB.Erfassung->wpdisabled # n;
```

```
    $NB.Erfassung->wpvisible # true;
```

```
    vHdl # gMdi->Winsearch('NB:Main');
```

```
    vHdl->wpcurrent      # 'NB.Erfassung';
```

```
    Refreshfm();
```

```
end;
```

```
end
```

```
else if (Auf.Nummer>1000000000) and (mode=c_ModeList2) then begin
```

```
    // kompletten Auftrag verwerfen
```

```
    w_Command # ";
```

```
    w_Cmd_Para # "
```

```
TRANSON;
```

```
Erx # RecLink(401,400,9,_RecFirst);
```

```
WHILE (Erx=_rOk) do begin // Positionen entfernen
```

```
    TxtDelete(myTmpText+'.401.'+CnvAI(Auf.P.Position,_FmtNumLeadZero | _FmtNumNoGroup,0,3),0);
```

```
    TxtDelete(myTmpText+'.401.'+CnvAI(Auf.P.Position,_FmtNumLeadZero | _FmtNumNoGroup,0,3)+'.01
```

```
    Erx # RekDelete(401,0,'MAN');
```

```

if (Lib_MoreBufs>DeleteAll(401)<>_rOK) then begin

    TRANSBRK;

    RETURN false;

end;


Erx # RecLink(401,400,9,_RecFirst);

END;

TxtDelete(myTmpText+'.401.K',0);

TxtDelete(myTmpText+'.401.F',0);

        // Ausführungen löschen

WHILE (RecLink(402,400,16,_recFirst)=_rOk) do begin

    Erx # RekDelete(402,0,'MAN');

END;

        // Aufpreise löschen

WHILE (RecLink(403,400,13,_recFirst)=_rOk) do begin

    Erx # RekDelete(403,0,'MAN');

END;

        // Aktionen löschen

WHILE (RecLink(404,400,15,_recFirst)=_rOk) do begin

    if (Auf.A.Rechnungsnr<>0) then begin

        TRANSBRK;

        RETURN false;

    end;

    Erx # RekDelete(404,0,'MAN');

END;

        // Kalkulation löschen

```



```
WHILE (RecLink(405,400,14,_recFirst)=_rOk) do begin
```

```
    Erx # RekDelete(405,0,'MAN');
```

```
END;
```

```
    // Feinabrufe löschen
```

```
WHILE (RecLink(408,400,17,_recFirst)=_rOk) do begin
```

```
    Erx # RekDelete(408,0,'MAN');
```

```
END;
```

```
    // Stückliste löschen
```

```
WHILE (RecLink(409,400,18,_recFirst)=_rOk) do begin
```

```
    Erx # RekDelete(409,0,'MAN');
```

```
END;
```

```
TRANSOFF;
```

```
// 21.03.2018 AH:
```

```
Lib_RmtData:UserReset('400');
```

```
$NB.Page1->wpdisabled # n;
```

```
$NB.Page2->wpdisabled # n;
```

```
$NB.Page3->wpdisabled # n;
```

```
$NB.Page4->wpdisabled # n;
```

```
$NB.Page5->wpdisabled # n;
```

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

```
    $NB.Kopf->wpdisabled # n;
```

```

w_AppendNr # 0;

end;

end

else if (Auf.Nummer>1000000000) and (mode=c_ModeNew2) then begin

TRANSON;

TxtDelete(myTmpText+'.401.'+CnvAI(Auf.P.Position,_FmtNumLeadZero | _FmtNumNoGroup,0,3),0);
TxtDelete(myTmpText+'.401.'+CnvAI(Auf.P.Position,_FmtNumLeadZero | _FmtNumNoGroup,0,3)+'.01',

// Ausführungen löschen

WHILE (RecLink(402,401,11,_recFirst)=_rOk) do begin

Erx # RekDelete(402,0,'MAN');

END;

// Aufpreise löschen

WHILE (RecLink(403,401,6,_recFirst)=_rOk) do begin

Erx # RekDelete(403,0,'MAN');

END;

// Aktionen löschen

WHILE (RecLink(404,401,12,_recFirst)=_rOk) do begin

if (Auf.A.Rechnungsnr<>0) then begin

TRANSEBRK;

RETURN false;

end;

Erx # RekDelete(404,0,'MAN');

END;

// Kalkulation löschen

```

```
WHILE (RecLink(405,401,7,_recFirst)=_rOk) do begin
```

```
    Erx # RekDelete(405,0,'MAN');
```

```
END;
```

```
    // Feinabrufe löschen
```

```
WHILE (RecLink(408,401,13,_recFirst)=_rOk) do begin
```

```
    Erx # RekDelete(408,0,'MAN');
```

```
END;
```

```
    // Stückliste löschen
```

```
WHILE (RecLink(409,401,15,_recFirst)=_rOk) do begin
```

```
    Erx # RekDelete(409,0,'MAN');
```

```
END;
```

```
TRANSOFF;
```

```
// 21.03.2018 AH:
```

```
Lib_RmtData:UserReset('400|'+aint(Auf.P.Position));
```

```
end;
```

```
$NB.Kopf->wpdisabled # n;
```

```
RETURN true;
```

```
end;
```

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

```
// RecDel
```

```
//      Satz soll gelöscht werden
```

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

```
sub RecDel()
```

```
local begin
```

```
    Erx  : int;
```

```
    vNr  : int;
```

```
    vHdl : int;
```

```
end;
```

```
begin
```

```
    vHdl # gMdi->Winsearch('NB.Main');    // während Erfassung löschen?
```

```
    if (vHdl->wpcurrent='NB.Erfassung') and (Auf.P.Nummer>1000000000) then begin    // 05.10.2020 AH : z
```

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

```
            TRANSON;
```

```
                // Ausführungen löschen
```

```
            WHILE (RecLink(402,401,11,_recFirst)=_rOk) do begin
```

```
                Erx # RekDelete(402,0,'MAN');
```

```
            END;
```

```
                // Aufpreise löschen
```

```
            WHILE (RecLink(403,401,6,_recFirst)=_rOk) do begin
```

```
                Erx # RekDelete(403,0,'MAN');
```

```
            END;
```

```
                // Aktionen löschen
```

```
            WHILE (RecLink(404,401,12,_recFirst)=_rOk) do begin
```

```
                if (Auf.A.Rechnungsnr<>0) then begin
```

```

    TRANSBRK;

    RETURN;

end;

Erx # RekDelete(404,0,'MAN');

END;

        // Kalkulation löschen

WHILE (RecLink(405,401,7,_recFirst)=_rOk) do begin

    Erx # RekDelete(405,0,'MAN');

END;

        // Feinabrufe löschen

WHILE (RecLink(408,401,13,_recFirst)=_rOk) do begin

    Erx # RekDelete(408,0,'MAN');

END;

        // Stückliste löschen

WHILE (RecLink(409,401,15,_recFirst)=_rOk) do begin

    Erx # RekDelete(409,0,'MAN');

END;


if (RekDelete(401,0,'MAN')<>_rOk) then begin // Position löschen

    TRANSBRK;

    RETURN;

end;


if (Lib_MoreBufs>DeleteAll(401)<>_rOk) then begin

    TRANSBRK;

    RETURN;

```

end;

TRANSOFF;

\$ZL.Erfassung->WinUpdate(_WinUpdOn, _WinLstRecFromRecId | _WinLstRecDoSelect);

end;

RETURN;

end;

// LÖSCHEN:

if (Auf_P_Subst:ToggleLoeschmarker(y)=n) then begin

if (AfxRes<999) then Msg(401000,cnvai(Erg),0,0,0);

ErrorOutput;

RETURN;

end;

if (gMDI=cDialog2) then begin

vNr # Auf.Nummer;

cZList2->WinUpdate(_WinUpdOn, _WinLstRecFromBuffer | _WinLstRecDoSelect);

Auf.Nummer # vNr;

RecRead(400,1,0);

end;

end;

```

//=====

// EvtFocusInit

//      Fokus auf Objekt neu gesetzt

//=====

sub EvtFocusInit (
    aEvt          : event;    // Ereignis
    aFocusObject   : int      // vorheriges Objekt
) : logic
local begin
    itemp : int;
    vFocus : alpha;
    vHdl   : int;
end
begin
    if (gMDI<>w_Mdi) then  gMDI # w_MDI; // MDIBUGFIX 03.06.2014

    vFocus # aEvt:Obj->wpname;

    if (vFocus='jump') then begin

        case (aEvt:Obj->wpcustom) of
            'MainStart' : begin
                if ($NB.Page1->wpcustom='NB.Page1_Art') then
                    $edAuf.P.Auftragsart->winfocusset(false)

```

```

else

    $edAuf.P.Auftragsart_Mat->winfocusset(false);

end;

'vorMain' : begin

    if ($NB.Page1->wpcustom='NB.Page1_Art') then

        $edAuf.P.Auftragsart->winfocusset(false)

    else

        $edAuf.P.Auftragsart_Mat->winfocusset(false);

        $NB.Main->wpcurrent # 'NB.Page5';

        $edAuf.P.VorlageBAG->winfocusset(false);

    end;

'nachMain' : begin

    if ($NB.Page1->wpcustom='NB.Page1_Art') then

        $edAuf.P.Termin.Zusatz->winfocusset(false)

    else

        $edAuf.P.Termin.Zusatz_Mat->winfocusset(false);

        $NB.Main->wpcurrent # 'NB.Page2';

        $cb.Text1->winfocusset(true);

    end;

```

```

'TextStart' : begin

    $cb.Text1->winfocusset(false);

end;

'vorText' : begin

    $cb.Text1->winfocusset(false);

```



```

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

if ($NB.Page1->wpcustom='NB.Page1_Art') then

    $edAuf.P.Termin.Zusatz->winfocusset(false)

else

    $edAuf.P.Termin.Zusatz_Mat->winfocusset(false);

end;

'nachText' : begin

    $cb.Text3->winfocusset(false);

    if ($NB.Page1->wpcustom='NB.Page1_Art') then begin

        $NB.Main->wpcurrent # 'NB.Page5';

        $edAuf.P.Etikettentyp2->winfocusset();

    end

    else begin

        $NB.Main->wpcurrent # 'NB.Page3';

        $edAuf.P.Verpacknr->winfocusset();

    end;

end;

'VerpackungStart' : begin

    $edAuf.P.Verpacknr->winfocusset(false);

end;

'vorVerpackung' : begin

    $edAuf.P.Verpacknr->winfocusset(false);

    $NB.Main->wpcurrent # 'NB.Page2';

    $cb.Text3->winfocusset(true);

    Refreshfm();

```

end;

'nachVerpackung' : begin

\$edAuf.P.VpgText6->winfocusset(false);

\$NB.Main->wpcurrent # 'NB.Page4';

\$edAuf.P.Streckgrenze1->winfocusset(false);

Refreshfm();

end;

'AnalyseStart' : begin

\$edAuf.P.Streckgrenze1->winfocusset(false);

end;

'vorAnalyse' : begin

\$edAuf.P.Streckgrenze1->winfocusset(false);

\$NB.Main->wpcurrent # 'NB.Page3';

\$edAuf.P.VpgText6->winfocusset(false);

Refreshfm();

end;

'nachAnalyse' : begin

\$edAuf.P.Chemie.Frei1.2->winfocusset(false);

\$NB.Main->wpcurrent # 'NB.Page5';

\$edAuf.P.Etikettentyp2->winfocusset(false);

Refreshfm();

end;

'EtikettierungStart' : begin

\$edAuf.P.Etikettentyp2->winfocusset(false);

```

end;

'vorEtikettierung' : begin

    $edAuf.P.Etikettentyp2->winfocusset(false);

    if ($NB.Page1->wpcustom='NB.Page1_Art') then begin

        $NB.Main->wpcurrent # 'NB.Page2';

        $cb.Text3->winfocusset(true);

    end

    else begin

        $NB.Main->wpcurrent # 'NB.Page4';

        $edAuf.P.Chemie.Frei1.2->winfocusset(false);

    end;

    Refreshfm();

end;

'nachEtikettierung' : begin

    $edAuf.P.VpgText62->winfocusset(false);

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

    if ($NB.Page1->wpcustom='NB.Page1_Art') then

        $edAuf.P.Auftragsart->winfocusset(false)

    else

        $edAuf.P.Auftragsart_Mat->winfocusset(false);

        Refreshfm();

    end;

end;

'Termin' : begin

    $edAuf.P.Termin1W.Zahl->winfocusset(false);

```

end;

end;

RETURN true;

end;

if (vFocus='jump.Detail') then begin

if (Wgr_Data:IstMat(Auf.P.Wgr.Dateinr)) or (Wgr_Data:IstMixMat(Auf.P.Wgr.Dateinr)) then

\$edAuf.P.Guete->winfocusset(false)

else

\$edAuf.P.ArtikelNr->winfocusset(false);

RETURN true;

end;

// Auswahlfelder aktivieren

if ((vFocus='edAuf.P.AbrufAufPos') and (Auf.AbrufYN=n) and (Auf.PAbrufYN=n)) or

((vFocus='edAuf.P.Grundpreis') and (Auf.P.ArtikelNr<>"")) or

((vFocus='edAuf.P.AbrufAufPos_Mat') and (Auf.AbrufYN=n) and (Auf.PAbrufYN)) or

((vFocus='edAuf.P.AbrufAufPos_Mat') and (Auf.Vorgangstyp=C_REKOR)) or

((vFocus='edAuf.P.KundenMatArtNr_Mat') and (Mode=c_ModeNew2)) or

(Lib_Pflichtfelder:TypAuswahlFeld(aEvt:Obj,y)<>"") 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

    Erx    : int;

    vFocus  : alpha;

    vX      : float;

    vGew    : float;

    vStk    : int;

    vBuf401 : int;

    vOK     : logic;

end;

begin

    vFocus # aEvt:Obj->wpname;

    // Ankerfunktion

    RunAFX('Auf.P.EvtFocusTerm',vFocus);

```

if (aEvt:obj->wpname='edAuf.P.AusfOben_Mat') and (aEvt:Obj->wpchanged) then

RunAFX('Obf.Changed','401|1|'+aint(aEvt:obj));

if (aEvt:obj->wpname='edAuf.P.AusfUnten_Mat') and (aEvt:Obj->wpchanged) then

RunAFX('Obf.Changed','401|2|'+aint(aEvt:obj));

if

((aEvt:obj->wpname='^cbAuf.AbrufYN') or

(aEvt:obj->wpname='^cbAuf.PAbrufYN')) and

(aEvt:Obj->wpchanged) then begin

vOK # aEvt:Obj->wpCheckState=_WinStateChkChecked;

if (vOK=false) then begin

Auf.P.AbrufAufNr # 0;

Auf.P.AbrufAufPos # 0;

end;

Lib_GuiCom:Able(\$edAuf.P.AbrufAufNr, vOK);

Lib_GuiCom:Able(\$edAuf.P.AbrufAufPos, vOK);

Lib_GuiCom:Able(\$bt.Abruf, vOK);

Lib_GuiCom:Able(\$edAuf.P.AbrufAufNr_Mat, vOK);

Lib_GuiCom:Able(\$edAuf.P.AbrufAufPos_Mat, vOK);

Lib_GuiCom:Able(\$bt.Abruf_Mat, vOK);

end;

if (vFocus='edAuf.P.TextNr2') or (vFocus='edAuf.P.TextNr2b') then begin

if (Auf.P.TextNr1=400) then

```

Auf.P.TextNr2 # $edAuf.P.TextNr2->wpcaptionint;

if (Auf.P.TextNr1=0) then

    Auf.P.TextNr2 # $edAuf.P.TextNr2b->wpcaptionint;

end;


// logische Prüfung von Verknüpfungen

RefreshIfm(vFocus);


if (vFocus='cbAuf.AbrufYN') or (vFocus='cbAuf.PAbrufYN') then begin

    if (Auf.AbrufYN=n) and (Auf.PAbrufYN) then begin

        Lib_GuiCom:Enable($edAuf.P.ArtikelNr);

        Lib_GuiCom:Enable($edAuf.P.MEH.Wunsch);

        Lib_GuiCom:Enable($edAuf.P.Warengruppe);

        Lib_GuiCom:Enable($edAuf.P.KundenArtNr);

        Lib_GuiCom:Enable($bt.Artikelnnummer);

        Lib_GuiCom:Enable($bt.KundenMatArtNr_Mat);

        Lib_GuiCom:Enable($bt.KundenArtNr);

//    Lib_GuiCom:Disable($edAuf.P.AbrufAufNr);

//    Lib_GuiCom:Disable($edAuf.P.AbrufAufPos);

//    Lib_GuiCom:Disable($bt.Abruf);

//    Lib_GuiCom:Disable($edAuf.P.AbrufAufNr_Mat);

//    Lib_GuiCom:Disable($edAuf.P.AbrufAufPos_Mat);

//    Lib_GuiCom:Disable($bt.Abruf_Mat);

end

else begin

    Lib_GuiCom:Disable($edAuf.P.ArtikelNr);

```

```

Lib_GuiCom:Disable($edAuf.P.MEH.Wunsch);

Lib_GuiCom:Disable($edAuf.P.Warengruppe);

Lib_GuiCom:Disable($edAuf.P.KundenArtNr);

Lib_GuiCom:Disable($bt.Artikelnummer);

Lib_GuiCom:Disable($bt.KundenMatArtNr_Mat);

Lib_GuiCom:Disable($bt.KundenArtNr);

//  Lib_GuiCom:Enable($edAuf.P.AbrufAufNr);

//  Lib_GuiCom:Enable($edAuf.P.AbrufAufPos);

//  Lib_GuiCom:Enable($bt.Abruf);

//  Lib_GuiCom:Enable($edAuf.P.AbrufAufNr_Mat);

//  Lib_GuiCom:Enable($edAuf.P.AbrufAufPos_Mat);

//  Lib_GuiCom:Enable($bt.Abruf_Mat);

end;

end;

if (vFocus='cbAuf.WaehrungFixYN') then begin

  if ("Auf.WährungFixYN"=n) then begin

    "Auf.Währungskurs" # 0.0;

    Refreshm('edAuf.Waehrungskurs')

  end;

end;

if (vFocus='edAuf.Best.Nummer') and (Auf.Best.Nummer<>") and

($edAuf.Best.Nummer->wpchanged) and (Auf.Best.Datum=0.0.0) then begin

  Auf.Best.Datum # today;

end;

```



```
if ((vFocus='edAuf.P.Skizzennummer') and ($edAuf.P.Skizzennummer->wpchanged)) then begin  
    Skizzendaten();  
    RETURN false;  
end;
```

```
if (( (Wgr_Data:IstArt(Auf.P.Wgr.Dateinr)) or (Wgr_Data:IstMixArt(Auf.P.Wgr.Dateinr)) ) and  
    (vFocus='edAuf.P.Etikettentyp2') and ($edAuf.P.Etikettentyp2->wpchanged)) then begin  
    Etikettendaten();  
    RETURN false;  
end;
```

```
if (vFocus='edAuf.P.Warengruppe') then begin//and ($edAuf.P.Warengruppe->wpchanged) then begin  
    if (Auf_P_SMain:Switchmask(y)) then begin  
        $edAuf.P.Warengruppe_Mat->winfocusset(true);  
        RETURN false;  
    end;  
end;
```

```
if (vFocus='edAuf.P.Warengruppe_Mat') then begin //and ($edAuf.P.warengruppe_Mat->wpchanged) then begin  
    if (Auf_P_SMain:Switchmask(y)) then begin  
        $edAuf.P.Warengruppe->winfocusset(true);  
        RETURN false;  
    end;  
end;
```

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

```
if (vFocus='edAuf.P.Warengruppe_Mat') and (Auf.P.Warengruppe=0) and  
  (aFocusObject->wpname='edAuf.P.Best.Nummer') then begin  
  Auswahl('Warengruppe');  
end;
```

```
if (vFocus='edAuf.P.Termin1Wunsch') and (Auf.P.Termin1Wunsch=0.0.0) and  
  (aFocusObject->wpname='edAuf.P.Termin2W.Zahl') then begin  
  $edAuf.P.PEH->winfocusset(true);  
  RETURN true;  
end;  
end;
```

```
if (Mode=c_ModeEdit) then begin  
  if (vFocus='edAuf.P.Warengruppe') and (aEvt:obj->wpchanged) then begin  
    RecLink(819,401,1,0); // Warengruppe holen  
    if (Auf.P.Wgr.Dateinr<>Wgr.Dateinummer) then begin  
      Auf.P.Warengruppe # Fldword(ProtokollBuffer[cFile],1,9);  
      RefreshIfm(vFocus);  
      RETURN false;  
    end;  
  end;  
end;  
end;
```

```
if ((vFocus='edAuf.P.Termin1W.Zahl') or (vFocus='edAuf.P.Termin1W.Jahr')) and  
  (($edAuf.P.Termin1W.Zahl->wpchanged) or ($edAuf.P.Termin1W.Jahr->wpchanged)) then begin
```

```
//if ($edAuf.P.Termin1W.Zahl->wpchanged) then debugx('zahl changed');
```

```
//if ($edAuf.P.Termin1W.Jahr->wpchanged) then debugx('Jahr changed');
```

```
Lib_Berechnungen:Datum_aus_ZahlJahr(Auf.P.Termin1W.Art, var Auf.P.Termin1W.Zahl, var Auf.P.Termin1W.Jahr)
```

```
if (Auf.P.Termin1W.Art='KW') then
```

```
    Auf.P.Termin1Wunsch->vmdaymodify(Set.Auf.Term.AutoTag);
```

```
    $edAuf.P.Termin1W.Jahr->winupdate(_WinUpdFld2Obj);
```

```
    $edAuf.P.Termin1Wunsch->winupdate(_WinUpdFld2Obj);
```

```
end;
```

```
if (vFocus='edAuf.P.Termin1Wunsch') and
```

```
    ($edAuf.P.Termin1Wunsch->wpchanged) then begin
```

```
    Lib_Berechnungen:ZahlJahr_aus_Datum(Auf.P.Termin1Wunsch, Auf.P.Termin1W.Art, var Auf.P.Termin1W.Zahl, var Auf.P.Termin1W.Jahr)
```

```
    $edAuf.P.Termin1W.Zahl->winupdate(_WinUpdFld2Obj);
```

```
    $edAuf.P.Termin1W.Jahr->winupdate(_WinUpdFld2Obj);
```

```
end;
```

```
if ((vFocus='edAuf.P.Termin2W.Zahl') or (vFocus='edAuf.P.Termin2W.Jahr')) and
```

```
    (($edAuf.P.Termin2W.Zahl->wpchanged) or ($edAuf.P.Termin2W.Jahr->wpchanged)) then begin
```

```
    Lib_Berechnungen:Datum_aus_ZahlJahr(Auf.P.Termin1W.Art, var Auf.P.Termin2W.Zahl, var Auf.P.Termin2W.Jahr)
```

```
    if (Auf.P.Termin1W.Art='KW') then
```

```
        Auf.P.Termin2Wunsch->vmdaymodify(Set.Auf.Term.AutoTag);
```

```
        $edAuf.P.Termin2W.Jahr->winupdate(_WinUpdFld2Obj);
```

```
        $edAuf.P.Termin2Wunsch->winupdate(_WinUpdFld2Obj);
```

```
end;
```

```
if (vFocus='edAuf.P.Termin2Wunsch') and
```

(\$edAuf.P.Termin2Wunsch->wpchanged) then begin

Lib_Berechnungen:ZahlJahr_aus_Datum(Auf.P.Termin2Wunsch, Auf.P.Termin1W.Art, var Auf.P.Termin

\$edAuf.P.Termin2W.Zahl->winupdate(_WinUpdFld2Obj);

\$edAuf.P.Termin2W.Jahr->winupdate(_WinUpdFld2Obj);

end;

if ((vFocus='edAuf.P.TerminZ.Zahl') or (vFocus='edAuf.P.TerminZ.Jahr')) and

((\$edAuf.P.TerminZ.Zahl->wpchanged) or (\$edAuf.P.TerminZ.Jahr->wpchanged)) then begin

Lib_Berechnungen:Datum_aus_ZahlJahr(Auf.P.Termin1W.Art, var Auf.P.TerminZ.Zahl, var Auf.P.Termin

if (Auf.P.Termin1W.Art='KW') then

Auf.P.TerminZusage->vmdaymodify(Set.Auf.Term.AutoTag);

\$edAuf.P.TerminZ.Jahr->winupdate(_WinUpdFld2Obj);

\$edAuf.P.TerminZusage->winupdate(_WinUpdFld2Obj);

end;

if (vFocus='edAuf.P.TerminZusage') and

(\$edAuf.P.TerminZusage->wpchanged) then begin

Lib_Berechnungen:ZahlJahr_aus_Datum(Auf.P.TerminZusage, Auf.P.Termin1W.Art, var Auf.P.TerminZ

\$edAuf.P.TerminZ.Zahl->winupdate(_WinUpdFld2Obj);

\$edAuf.P.TerminZ.Jahr->winupdate(_WinUpdFld2Obj);

end;

if ((vFocus='edAuf.P.Termin1W.Zahl_Mat') or (vFocus='edAuf.P.Termin1W.Jahr_Mat')) and

((\$edAuf.P.Termin1W.Zahl_Mat->wpchanged) or (\$edAuf.P.Termin1W.Jahr_Mat->wpchanged)) then be

Lib_Berechnungen:Datum_aus_ZahlJahr(Auf.P.Termin1W.Art, var Auf.P.Termin1W.Zahl, var Auf.P.Termin

if (Auf.P.Termin1W.Art='KW') then

Auf.P.Termin1Wunsch->vmdaymodify(Set.Auf.Term.AutoTag);

\$edAuf.P.Termin1W.Jahr_Mat->winupdate(_WinUpdFld2Obj);

```

$edAuf.P.Termin1Wunsch_Mat->winupdate(_WinUpdFld2Obj);

end;

if (vFocus='edAuf.P.Termin1Wunsch_Mat') and

($edAuf.P.Termin1Wunsch_Mat->wpchanged) then begin

Lib_Berechnungen:ZahlJahr_aus_Datum( Auf.P.Termin1Wunsch, Auf.P.Termin1W.Art, var Auf.P.Termin1W.Zahl_Mat->winupdate(_WinUpdFld2Obj);

$edAuf.P.Termin1W.Zahl_Mat->winupdate(_WinUpdFld2Obj);

$edAuf.P.Termin1W.Jahr_Mat->winupdate(_WinUpdFld2Obj);

end;

if ((vFocus='edAuf.P.Termin2W.Zahl_Mat') or (vFocus='edAuf.P.Termin2W.Jahr_Mat')) and

(($edAuf.P.Termin2W.Zahl_Mat->wpchanged) or ($edAuf.P.Termin2W.Jahr_Mat->wpchanged)) then begin

Lib_Berechnungen:Datum_aus_ZahlJahr(Auf.P.Termin1W.Art, var Auf.P.Termin2W.Zahl, var Auf.P.Termin2W.Jahr_Mat->winupdate(_WinUpdFld2Obj);

if (Auf.P.Termin1W.Art='KW') then

Auf.P.Termin2Wunsch->vmdaymodify(Set.Auf.Term.AutoTag);

$edAuf.P.Termin2W.Jahr_Mat->winupdate(_WinUpdFld2Obj);

$edAuf.P.Termin2Wunsch_Mat->winupdate(_WinUpdFld2Obj);

end;

if (vFocus='edAuf.P.Termin2Wunsch_Mat') and

($edAuf.P.Termin2Wunsch_Mat->wpchanged) then begin

Lib_Berechnungen:ZahlJahr_aus_Datum( Auf.P.Termin2Wunsch, Auf.P.Termin1W.Art, var Auf.P.Termin2W.Zahl_Mat->winupdate(_WinUpdFld2Obj);

$edAuf.P.Termin2W.Zahl_Mat->winupdate(_WinUpdFld2Obj);

$edAuf.P.Termin2W.Jahr_Mat->winupdate(_WinUpdFld2Obj);

end;

if ((vFocus='edAuf.P.TerminZ.Zahl_Mat') or (vFocus='edAuf.P.TerminZ.Jahr_Mat')) and

(($edAuf.P.TerminZ.Zahl_Mat->wpchanged) or ($edAuf.P.TerminZ.Jahr_Mat->wpchanged)) then begin

```

```

Lib_Berechnungen:Datum_aus_ZahlJahr(Auf.P.Termin1W.Art, var Auf.P.TerminZ.Zahl, var Auf.P.TerminZ.Zahl);

if (Auf.P.Termin1W.Art='KW') then

    Auf.P.TerminZusage->vmdaymodify(Set.Auf.Term.AutoTag);

    $edAuf.P.TerminZ.Jahr_Mat->winupdate(_WinUpdFld2Obj);

    $edAuf.P.TerminZusage_Mat->winupdate(_WinUpdFld2Obj);

end;

if (vFocus='edAuf.P.TerminZusage_Mat') and

    ($edAuf.P.TerminZusage_Mat->wpchanged) then begin

    Lib_Berechnungen:ZahlJahr_aus_Datum( Auf.P.TerminZusage, Auf.P.Termin1W.Art, var Auf.P.TerminZ.Zahl);

    $edAuf.P.TerminZ.Zahl_Mat->winupdate(_WinUpdFld2Obj);

    $edAuf.P.TerminZ.Jahr_Mat->winupdate(_WinUpdFld2Obj);

end;

if (vFocus='edAuf.P.Laenge_Mat') and ($edAuf.P.Laenge_Mat->wpchanged) then begin

    if ("Auf.P.Länge"<>0.0) then begin

//    Auf.P.RID    # 0.0;

//    Auf.P.RIDmax # 0.0;

//    Auf.P.RAD    # 0.0;

//    Auf.P.RADmax # 0.0;

//    $edAuf.P.RID_Mat->winupdate(_WinUpdFld2Obj);

//    $edAuf.P.RIDmax_Mat->winupdate(_WinUpdFld2Obj);

//    $edAuf.P.RAD_Mat->winupdate(_WinUpdFld2Obj);

//    $edAuf.P.RADmax_Mat->winupdate(_WinUpdFld2Obj);

//    Lib_GuiCom:Disable($edAuf.P.RID_Mat);

```

```

//  Lib_GuiCom:Disable($edAuf.P.RIDMax_Mat);

//  Lib_GuiCom:Disable($edAuf.P.RAD_Mat);

//  Lib_GuiCom:Disable($edAuf.P.RADMax_Mat);

    end

    else begin

//  Lib_GuiCom:Enable($edAuf.P.RID_Mat);

//  Lib_GuiCom:Enable($edAuf.P.RIDMax_Mat);

//  Lib_GuiCom:Enable($edAuf.P.RAD_Mat);

//  Lib_GuiCom:ENable($edAuf.P.RADMax_Mat);

        end;

    end;

if (vFocus='edAuf.P.Stueckzahl_Mat') and ($edAuf.P.Stueckzahl_Mat->wpchanged) and
    ((Auf.Vorgangstyp = c_Auf) or (Auf.Vorgangstyp = c_Ang) or (Auf.Vorgangstyp=c_VorlageAuf)) then begin
    if (StrFind(Set.Auf.Calc.Menge,'K',1)>0) or
        (("Auf.P.Stückzahl"<>0) and (Auf.P.Gewicht=0.0) and (StrFind(Set.Auf.Calc.Menge,'G',1)>0)) then begin
//  vGew # Lib_Berechnungen:KG_aus_StkDBLWgrArt("Auf.P.Stückzahl", Auf.P.Dicke, Auf.P.Breite, "Au
vGew # Auf_P_Subst:CalcGewicht(); // 22.06.2022 AH

        if (vGew<>0.0) then begin

            Auf.P.Gewicht # vGew;

            $edAuf.P.Gewicht_Mat->winupdate(_WinUpdFld2Obj);

        end;

    end;

end;

end;

```

```

if (vFocus='edAuf.P.Gewicht_Mat') and ($edAuf.P.Gewicht_Mat->wpchanged) and
((Auf.Vorgangstyp = c_Auf) or (Auf.Vorgangstyp = c_Ang) or (Auf.Vorgangstyp=c_VorlageAuf)) then begin
if (StrFind(Set.Auf.Calc.Menge,'K',1)>0) or
(("Auf.P.Stückzahl"=0) and (Auf.P.Gewicht<>0.0) and (StrFind(Set.Auf.Calc.Menge,'S',1)>0)) then begin
vStk # Lib_Berechnungen:STK_aus_KgDBLWgrArt(Auf.P.Gewicht, Auf.P.Dicke, Auf.P.Breite, "Auf.P.lä
if (vStk<>0) then begin
"Auf.P.Stückzahl" # vStk;
$edAuf.P.Stueckzahl_Mat->winupdate(_WinUpdFld2Obj);
end;
end;
end;

```

```

if (vFocus='edAuf.P.MEH.Wunsch') and ($edAuf.P.MEH.Wunsch->wpchanged) then begin
Lib_Einheiten:CheckMEH(var Auf.P.MEH.Wunsch);
$edAuf.P.MEH.Wunsch->winupdate(_WinUpdFld2Obj);
end;

```

```

if (vFocus='edAuf.P.MEH.Preis') and ($edAuf.P.MEH.Preis->wpchanged) then begin
Lib_Einheiten:CheckMEH(var Auf.P.MEH.Preis);
$edAuf.P.MEH.Preis->winupdate(_WinUpdFld2Obj);
end;

```

```

if (vFocus='edAuf.P.MEH.Preis_Mat') and ($edAuf.P.MEH.Preis_Mat->wpchanged) then begin
Lib_Einheiten:CheckMEH(var Auf.P.MEH.Preis);
$edAuf.P.MEH.Preis_Mat->winupdate(_WinUpdFld2Obj);
end;

```



```

if (vFocus='edAuf.P.Menge.Wunsch') or
(vFocus='edAuf.P.MEH.Wunsch') or
(vFocus='edAuf.P.Menge') or
(vFocus='edAuf.P.PEH') or
(vFocus='edAuf.P.MEH.Preis') or
(vFocus='edAuf.P.Grundpreis') or
(vFocus='edRabatt1') then begin
Auf_K_Data:SumKalkulation();
Auf_Data:SumEKGesamtPreis();
Auf.P.Einzelpreis # Auf.P.Grundpreis + Auf.P.Aufpreis;
Auf.P.Gesamtpreis # Auf_data:SumGesamtpreis(Auf.P.Menge, "Auf.P.Stückzahl" , Auf.P.Gewicht)
$lb.Kalkuliert->wpcaption # ANum(Auf.P.Kalkuliert,2);
$lb.Poswert->wpcaption # ANum(Auf.P.Gesamtpreis,2);
$lb.Rohgewinn->wpcaption # ANum(Auf.P.Gesamtpreis - "Auf.P.GesamtwertEKW1",2);
end;

```

```

if (vFocus='edAuf.P.Menge') then begin
Erx # RecLink(250,401,2,_RecFirst); // Artikel holen
if (Erx>_rLocked) then RecBufClear(250);
if (RecLinkInfo(409,401,15,_RecCount)<>0) then begin
// "Auf.P.Stückzahl" # 0;
// Auf.P.Gewicht # 0.0;
// Art_Data:BerechneFelder(var "Auf.P.Stückzahl", var Auf.P.Gewicht, var Auf.P.Menge, Auf.P.MEH.Ein

```

```

//Auf_Data:SumEKGesamtPreis();

end

else begin

    if (Auf.P.MEH.Einsatz='Stk') then "Auf.P.Stückzahl" # Cnvif(Auf.P.Menge);

    if (Auf.P.MEH.Einsatz='kg') then "Auf.P.Gewicht" # rnd(Auf.P.Menge, Set.Stellen.Gewicht);

    if (Auf.P.MEH.Einsatz='t') then "Auf.P.Gewicht" # rnd(Auf.P.Menge / 1000.0, Set.Stellen.Gewicht);

end;

$lb.Auf.P.Gewicht->winupdate(_WinUpdFld2Obj);

$lb.Auf.P.Stueck->winupdate(_WinUpdFld2Obj);

end;

```

```

// 24.03.2016 AH besser so laut Sonne/KuZ    Art_Data:BerechneFelder(var "Auf.P.Stückzahl", var Auf.P.

```

```

if ((vFocus='edAuf.P.Dicke') or
    (vFocus='edAuf.P.Breite') or
    (vFocus='edAuf.P.Laenge') or
    (vFocus='edAuf.P.Menge.Wunsch')) and (aEvt:obj->wpchanged) then begin
Auf_P_Sub:CalcEinsatzMenge250();
Auf_P_Sub:CalcMengen();
Auf_P_Sub:CalcGesamtPreis();

end;

```

```

/***

```

```

if (vFocus='edAuf.P.Menge.Wunsch') then begin

    if (RecLinkInfo(409,401,15,_RecCount)=0) then begin

        "Auf.P.Stückzahl" # 0;
    end;
end;

```

```

Auf.P.Gewicht    # 0.0;

// 24.03.2016 AH besser so laut Sonne/KuZ    Art_Data:BerechneFelder(var "Auf.P.Stückzahl", var Auf.P.

Auf.P.Menge      # Rnd(Lib_Einheiten:WandleMEH(401, 0, 0.0, Auf.P.Menge.Wunsch, Auf.P.MEH.Wun

debugx(anum(auf.p.menge,0)+' '+anum(Auf.P.Menge.Wunsch,0)+' auf '+auf.p.meh.einsatz);

"Auf.P.Stückzahl" # cnvif(Lib_Einheiten:WandleMEH(401, 0, 0.0, Auf.P.Menge.Wunsch, Auf.P.MEH.W

Auf.P.Gewicht    # Rnd(Lib_Einheiten:WandleMEH(401, 0, 0.0, Auf.P.Menge.Wunsch, Auf.P.MEH.Wun

end;

$edAuf.P.Menge->winupdate(_WinUpdFld2Obj);

$lb.Auf.P.Gewicht->winupdate(_WinUpdFld2Obj);

$lb.Auf.P.Stueck->winupdate(_WinUpdFld2Obj);

if (Auf.P.MEH.Wunsch=$lb.Art.MEH5->wpcaption) then begin

Auf.P.Menge # Auf.P.Menge.Wunsch;

Auf_P_Sub:CalcMengen();

$edAuf.P.Menge->winupdate(_WinUpdFld2Obj);

$lb.Auf.P.Prd.Rest->wpcaption # ANum(Auf.P.Prd.Rest,Set.Stellen.Menge);

end

else begin

if ($edAuf.P.Menge.Wunsch->wpchanged) then begin

Auf.P.Menge # Rnd(Lib_Einheiten:WandleMEH(250,0,0.0, Auf.P.Menge.Wunsch, Auf.P.MEH.Wunsch

$edAuf.P.Menge->winupdate(_WinUpdFld2Obj);

end;

end;

end;

***/

```

```

if (vFocus='edAuf.P.MEH.Wunsch') then begin

    if (Auf.P.MEH.Wunsch=Auf.P.MEH.Einsatz) then begin

        Auf.P.Menge # Auf.P.Menge.Wunsch;

// 24.03.2016    Lib_GuiCom:Disable($edAuf.P.Menge);

        Refreshfm('edAuf.P.Menge');

    end

    else begin

// 24.03.2016    Lib_GuiCom:Enable($edAuf.P.Menge);

        if ($edAuf.P.MEH.Wunsch->wpchanged) then begin

            Auf.P.Menge # Lib_Einheiten:WandleMEH(250,0,0.0, Auf.P.Menge.Wunsch, Auf.P.MEH.Wunsch, Auf.P.MEH.Einsatz);

            $edAuf.P.Menge->winupdate(_WinUpdFld2Obj);

        end;

    end;

end;

end;

if (vFocus='edAuf.P.Gewicht_Mat') or (vFocus='edAuf.P.Stueckzahl_Mat') then begin//and (Auf.P.Wgr.Da

    if (Auf.P.MEH.Einsatz='kg') then Auf.P.Menge # Rnd(Auf.P.Gewicht,Set.Stellen.Menge);

    if (Auf.P.MEH.Einsatz='Stk') then Auf.P.Menge # cnvfi("Auf.P.Stückzahl");

//    if (Auf.P.MEH.Wunsch=Auf.P.MEH.Einsatz) then begin

//c    Auf.P.Menge # Rnd(Auf.P.Gewicht, Set.Stellen.Menge);

//    Auf.P.Menge.Wunsch # Rnd(Auf.P.Gewicht, Set.Stellen.Menge);

//    end;

end;

if (vFocus='edAuf.P.Gewicht_Mat') or (vFocus='edAuf.P.Stueckzahl_Mat') or

```

```

(vFocus='edAuf.P.Grundpreis_Mat') or
(vFocus='edAuf.P.PEH_Mat') or (vFocus='edAuf.P.MEH.Preis_Mat') then begin
Auf_P_Subst:CalcGesamtPreis();
end;

// Toleranzfelder zurücksetzen
if ((Auf.AbrufYN) or (Auf.PAbrufYN)) and (Auf.P.AbrufaufNr<>0) then begin
vBuf401 # RecBufCreate(401);
vBuf401->Auf.P.Nummer # Auf.P.AbrufAufNr;
vBuf401->Auf.P.Position # Auf.P.AbrufAufPos;
Erx # RecRead(vBuf401,1,0);
if (Erx>_rLocked) then begin
RecbufDestroy(vBuf401);
vBuf401 # 0;
end;
end;

if (vFocus = 'edAuf.P.Dicke_Mat') then begin
vOK # (aEvt:Obj->wpChanged);
if (vBuf401<>0) then
vOK # vOK and (vBuf401->Auf.P.Dicke<>0.0);
if (vOK) and (Set.Wie.ClrTolBeiEdt) then begin
"Auf.P.Dickentol" # "";
$edAuf.P.Dickentol_Mat->WinUpdate(_winUpdFld2Obj);
end;
end;
end;

```

```
if (vFocus = 'edAuf.P.Breite_Mat') then begin
    vOK # (aEvt:Obj->wpChanged);
    if (vBuf401<>0) then
        vOK # vOK and (vBuf401->Auf.P.Breite<>0.0);
    if (vOK) and (Set.Wie.ClrTolBeiEdt) then begin
        "Auf.P.Breitentol" # "";
        $edAuf.P.Breitentol_Mat->WinUpdate(_winUpdFld2Obj);
    end;
end;
```

```
if (vFocus = 'edAuf.P.Laenge_Mat') then begin
    vOK # (aEvt:Obj->wpChanged);
    if (vBuf401<>0) then
        vOK # vOK and (vBuf401->"Auf.P.Länge"<>0.0);
    if (vOK) and (Set.Wie.ClrTolBeiEdt) then begin
        "Auf.P.Längentol" # "";
        $edAuf.P.Laengentol_Mat->WinUpdate(_winUpdFld2Obj);
    end;
end;

if (vBuf401<>0) then RecBufDestroy(vBuf401);
```

```
RETURN true;
```

```
end;
```

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

```

// Auswahl

//      Auswahlliste öffnen

//=====

sub Auswahl(

    aBereich : alpha;

)

local begin

    Erx      : int;

    vA       : alpha;

    vHdl     : int;

    vHdl2    : int;

    vFilter  : int;

    vSel     : alpha;

    vSelName : alpha;

    vSel2    : int;

    vQ       : alpha(4000);

    vQ2,vQ3  : alpha(4000);

    vI       : int;

    tErg     : int;

    vTmp     : int;

    vM       : float;

end;

begin

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

```

case aBereich of

'AnalyseErweitert' : begin

if (Mode=c_ModeNew) or (Mode=c_ModeNew2) or (Mode=c_ModeEdit2) then

Lys_Msk_Main:Start(", y, 'neue Auftragsposition '+Auf.P.KundenSW, "Auf.P.Güte", "Auf.P.Gütenstufe

else

Lys_Msk_Main:Start(", Mode=c_ModeEdit, 'Auftrag '+aint(Auf.P.Nummer)+'/'+aint(Auf.P.Position)+' '+

RETURN;

end;

'Reservierungen' : begin

// NICHT bei Artiklegeschäft

if (Wgr_Data:IstArt(Auf.P.Wgr.Dateinr)) then RETURN;

if (RunAFX('Auf.P.Mnu.Reservierungen',"<0) then RETURN;

RecBufClear(200);

RecBufClear(203);

gMDI # Lib_GuiCom:AddChildWindow(gMDI,'Mat.Rsv.Verwaltung',here+'AusRes',y);

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

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

vHdl->wpcustom # 'AUF';

// <<< MUSTER


```

// Selektion aufbauen...

vQ # "";

Lib_Sel:QInt(var vQ, 'Mat.R.Auftragsnr' , '=', Auf.P.Nummer);

Lib_Sel:QInt(var vQ, 'Mat.R.Auftragspos' , '=', Auf.P.Position);

// 06.05.2019 Proj. 1884/115

//   vQ # vQ + ' AND LinkCount(Material) > 0 ';

//   vQ2 # ' "Mat.Löschmarker" = ""';

vHdl # SelCreate(203, gkey);

//   vHdl->SelAddLink(",200, 203, 1, 'Material');

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

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

//   tErg # vHdl->SelDefQuery('Material', vQ2);

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

// speichern, starten und Name merken...

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

// Liste selektieren...

gZLList->wpDbSelection # vHdl;

// ENDE MUSTER >>>


Lib_GuiCom:RunChildWindow(gMDI);

end;


'Grobplanung' : begin

RecBufClear(600);

gMDI # Lib_GuiCom:AddChildWindow(gMDI,'GPI.Verwaltung',",y);

```

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

```
// ehemals Selektion 600 'ZU AUFTRAG'
```

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

```
Lib_Sel:QInt(var vQ2, 'GPI.P.Typ', '=', 401);
```

```
Lib_Sel:QInt(var vQ2, 'GPI.P.ID1', '=', Auf.P.Nummer);
```

```
Lib_Sel:QInt(var vQ2, 'GPI.P.ID2', '=', Auf.P.Position);
```

```
// Selektion aufbauen...
```

```
vHdl # SelCreate(600, gKey);
```

```
vHdl->SelAddLink(",601, 600, 1, 'Pos');
```

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

```
tErg # vHdl->SelDefQuery('Pos', vQ2);
```

```
// speichern, starten und Name merken...
```

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

```
// Liste selektieren...
```

```
gZLList->wpDbSelection # vHdl;
```

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

```
end;
```

```
'Skizze' : begin
```

```
RecBufClear(829);      // ZIELBUFFER LEEREN
```

```
gMDI # Lib_GuiCom:AddChildWindow(gMDI,'Skz.Verwaltung',here+':AusSkizze');
```

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

```
end;
```

```
'EinsatzVPG' : begin
```

```
gMDI # Lib_GuiCom:AddChildWindow(gMDI,'Adr.Verwaltung',here+':AusEinsatzVPG');
```

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

```
end;
```

```
'EinsatzVPG2' : begin
```

```
Adr.Nummer # Adr.V.EinsatzVPG.Adr;
```

```
RecRead(100,1,0);
```

```
gMDI # Lib_GuiCom:AddChildWindow(gMDI,'Adr.V.Verwaltung',here+':AusEinsatzVPG2');
```

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

```
Lib_Sel:QInt(var vQ, 'Adr.V.AdressNr', '=', Auf.P.EinsatzVPG.Adr);
```

```
vQ # vQ + ' AND Adr.V.VerkaufYN'; // 21.07.2015
```

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

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

```
end;
```

```
'VorlageBAG' : begin
```

```
RecBufClear(700);      // ZIELBUFFER LEEREN
```

```
gMDI # Lib_GuiCom:AddChildWindow(gMDI,'BA1.Verwaltung',here+':AusvorlageBAG');
```

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

```
vHdl # gZLList;
```

```
Lib_Sel:QRecList(vHdl,'BAG.VorlageYN=true AND BAG.VorlageSperrYN=false');
```

```

Lib_GuiCom:RunChildWindow(gMDI);

end;

'Verpackung' : begin

    Erx # RecLink(100,400,1,_RecFirst);      // Kunde holen

    if (Erx>_rLocked) then RecBufClear(100);

    RecBufClear(105);

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


    // Selektion

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

    if ("Set.Auf.!EigeneVPG") then begin

        Lib_Sel:QInt(var vQ, 'Adr.V.AdressNr', '=', Adr.Nummer);

    end

    else begin

        Lib_Sel:QInt(var vQ, 'Adr.V.AdressNr', '=', Adr.Nummer);

        Lib_Sel:QInt(var vQ, 'Adr.V.AdressNr', '=', Set.eigeneAdressnr, 'OR');

    end;

    if (Adr.Nummer<>Auf.Lieferadresse) then begin

        vQ # vQ + ' OR Adr.V.Adressnr = ' + CnvAI(Auf.Lieferadresse, _fmtNumNoGroup);

    end;

    vQ # 'Adr.V.VerkaufYN AND ('+vQ+)'; // 21.07.2015

    Lib_Sel:QRecList(0, vQ);


    Lib_GuiCom:RunChildWindow(gMDI);

```

end;

'Matz.Mat.GuBe' : begin

RecLink(400,401,3,_recFirst); // Kopf holen

if ("Auf.P.Löschmarker"='*') or ((Auf.Vorgangstyp != c_GUT) and (Auf.Vorgangstyp != c_Bel_LF)) or
((Wgr_Data:IstMat(Auf.P.Wgr.Dateinr)=false) and (Wgr_Data:IstMix(Auf.P.Wgr.Dateinr)=false)) then b

Msg(200400,"0,0,0,0);

RETURN;

end;

gMDI # Lib_GuiCom:AddChildWindow(gMDI,'Mat.Verwaltung',here+':AusMatzMatGuBe' ,n,n, '401')

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

Sel.Mat.von.Status # 0;

Sel.Mat.bis.Status # 999;

vQ # ";

// 13.01.2015

if (Wgr_data:IstMix(Auf.P.Wgr.Dateinr)) then begin

Lib_Sel:QAlpha(var vQ, 'Mat.Strukturnr', '=', Auf.P.Artikelnr);

end

else begin

end;

Lib_Sel:QRecList(0,vQ);

Lib_GuiCom:RunChildWindow(gMDI);

end;

'Matz.Mat' : begin

if (RecLinkInfo(409,401,15,_RecCount)>0) then begin

if (msg(401026,",_WinlcoWarning,_WinDialogYesNo,1)<>_winidyces) then RETURN;

end;

RecLink(400,401,3,_recFirst); // Kopf holen

if ("Auf.P.Löschmarker"='*') or (Auf.Vorgangstyp<>c_AUF) or

((Wgr_Data:IstMat(Auf.P.Wgr.Dateinr)=false) and (Wgr_Data:IstMix(Auf.P.Wgr.Dateinr)=false)) then b

Msg(200400,"0,0,0,0);

RETURN;

end;

gMDI # Lib_GuiCom:AddChildWindow(gMDI,'Mat.Verwaltung',here+':AusMatzMat' ,n,n, '401');

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

Sel.Mat.von.Status # c_Status_Frei;

Sel.Mat.bis.Status # c_Status_bisFrei;

vQ # ";

Lib_Sel:QAlpha(var vQ, "Mat.Löschmarker", '=', "');

// 13.01.2015

if (Wgr_data:IstMix(Auf.P.Wgr.Dateinr)) then begin

Lib_Sel:QAlpha(var vQ, 'Mat.Strukturnr', '=', Auf.P.ArtikelNr);

Lib_Sel:QVonBisI(var vQ, 'Mat.Status', Sel.Mat.von.Status, Sel.Mat.bis.Status);

```

Lib_Sel:QInt(var vQ, 'Mat.Status', '=', c_status_VsbPuffer, 'OR');

Lib_Sel:QInt(var vQ, 'Mat.Status', '=', c_status_VsbRahmen, 'OR');

Lib_Sel:QInt(var vQ, 'Mat.Status', '=', c_status_VsbKonsiRahmen, 'OR');

vQ # '(' + vQ +') OR ((Mat.Auftragsnr=Auf.P.Nummer) AND (Mat.Auftragspos=Auf.P.Position));

end

else begin

    Lib_Sel:QVonBisl(var vQ, 'Mat.Status', Sel.Mat.von.Status, Sel.Mat.bis.Status);

    Lib_Sel:QInt(var vQ, 'Mat.Status', '=', c_status_VsbPuffer, 'OR');

    Lib_Sel:QInt(var vQ, 'Mat.Status', '=', c_status_VsbRahmen, 'OR');

    Lib_Sel:QInt(var vQ, 'Mat.Status', '=', c_status_VsbKonsiRahmen, 'OR');

    vQ # '(' + vQ +') OR (Mat.Auftragsnr=Auf.P.Nummer)';

end;

Lib_Sel:QRecList(0,vQ);

Lib_GuiCom:RunChildWindow(gMDI);

end;

'Material' : begin

if ("Auf.P.Löschmarker"='*') or (Auf.Vorgangstyp<>c_AUF) or

    ((Wgr_Data:IstMix(Auf.P.Wgr.Dateinr)=false) and (Wgr_Data:IstMat(Auf.P.Wgr.Dateinr)=false)) then b

    Msg(200400,"0,0,0,0);

    RETURN;

end;

RecBufClear(200);      // ZIELBUFFER LEEREN

gMDI # Lib_GuiCom:AddChildWindow(gMDI,'Mat.Verwaltung',here+':AusMaterial');

```

```

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

Sel.Mat.von.Status # c_Status_Frei;

Sel.Mat.bis.Status # c_Status_bisFrei;

vQ # "";

Lib_Sel:QAlpha(var vQ, ""Mat.Löschmarker", '=' , "");

Lib_Sel:QVonBisI(var vQ, 'Mat.Status', Sel.Mat.von.Status, Sel.Mat.bis.Status);

Lib_Sel:QInt(var vQ, 'Mat.Status', '=', c_status_VsbPuffer, 'OR');

Lib_Sel:QInt(var vQ, 'Mat.Status', '=', c_status_VsbRahmen, 'OR');

Lib_Sel:QInt(var vQ, 'Mat.Status', '=', c_status_VsbKonsiRahmen, 'OR');

Lib_Sel:QRecList(0,vQ);

Lib_GuiCom:RunChildWindow(gMDI);

end;

'Matz.ArtC' : begin

if (Auf.P.ArtikelNr=") then RETURN;

if (RecLinkInfo(409,401,15,_RecCount)>0) then begin

    if (msg(401026,",_WinIcoWarning,_WinDialogYesNo,1)<>_winidyes) then RETURN;

end;

Erx # RecLink(250,401,2,_RecFirst); // Artikel holen

if (Erx>_rlockeD) then RETURN;

RecBufClear(252);      // ZIELBUFFER LEEREN

gMDI # Lib_GuiCom:AddChildWindow(gMDI,'Art.C.Verwaltung',here+':AusMatzArtC');

```



```

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

vQ # "";

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

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

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

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

vHdl # SelCreate(252, gKey);

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

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

// speichern, starten und Name merken...

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

// Liste selektieren...

gZLList->wpDbSelection # vHdl;


Lib_GuiCom:RunChildWindow(gMDI);

end;


'DFakt.Mat' : begin

if (Auf.PAbrufYN) then RETURN;

Lib_Mark:Reset(200);

gMDI # Lib_GuiCom:AddChildWindow(gMDI,'Mat.Verwaltung',here+':AusDFaktMat');

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


Erx # RecLink(400,401,3,_RecFirst); // Kopf holen

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

```

```

// Konsi?...

if (AAr.KonsiYN) then begin

    // 2023-03-06 AH

    if (Set.Installname='BSP') then begin

        vQ # "";

        Lib_Sel:QInt(var vQ, 'Mat.Lageradresse', '=', Auf.Lieferadresse);

        Lib_Sel:QInt(var vQ, 'Mat.VK.Rechnr', '=', 0);

        Lib_Sel:QInt(var vQ, 'Mat.KommKundennr', '=', Auf.P.Kundennr);

        Lib_Sel:QAlpha(var vQ, 'Mat.Löschmarker', '=', "");

        Lib_Sel:QFloat(var vQ, 'Mat.Bestand.Gew'      , '>', 0.0);

        vQ # vQ + ' AND LinkCount(AufPos) > 0';

        Lib_Sel:QVonBisI(var vQ, 'Mat.Status', 400, 409);

        Lib_sel:Qlogic(var vQ3, 'AAr.KonsiYN', true);

        vQ2 # 'LinkCount(AAr) > 0';

        vHdl # SelCreate(200, gkey);

        vHdl->SelAddLink("",401, 200, 16, 'AufPos');

        vHdl->SelAddLink('AufPos',835, 401, 5, 'AAr');

        tErg # vHdl->SelDefQuery('AufPos', vQ2);

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

        tErg # vHdl->SelDefQuery('AAr', vQ3);

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

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

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

        // speichern, starten und Name merken...

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

        // Liste selektieren...

```

```

    gZLList->wpDbSelection # vHdl;

end

else begin

    vQ # vQ + ' (( "Mat.Auftragsnr" = ' + Aint(Auf.P.Nummer) + ' AND "Mat.AuftragsPos" = ' + Aint(Auf.
    if (Auf.P.AbrufaufNr<>0) then
        vQ # vQ + ' OR ( "Mat.Auftragsnr" = ' + Aint(Auf.P.AbrufAufNr) + ' AND "Mat.AuftragsPos" = ' + Aint
        vQ # vQ + ');
    Lib_Sel:QRecList(0, vQ);

end;

end

// MatMix?....

else if (Wgr_Data:IstMix(Auf.P.Wgr.Dateinr)) then begin

    Lib_Sel:QAlpha(var vQ, 'Mat.StrukturNr'      , '=', Auf.P.ArtikelNr);

    Lib_Sel:QFloat(var vQ, 'Mat.Bestand.Gew'      , '>', 0.0);

    Lib_Sel:QRecList(0, vQ);

end;

Lib_GuiCom:RunChildWindow(gMDI);

end;

'DFakt.ArtC' : begin

    if (Auf.P.AbrufYN) then RETURN;

    if (Auf.P.ArtikelNr="") then RETURN;

    Erx # RecLink(250,401,2,_RecFirst); // Artikel holen

    if (Erx>_rlockeD) then RETURN;

```

```

// 12.01.2015

Erx # RekLink(819,250,10,0); // Warengruppe holen

if (Wgr.OhneBestandYN) then begin

//   Auf_Data_Buchen:DFaktArtC(Art.C.ArtikelNr, Art.C.Adressnr, Art.C.Anschriftnr, Art.C.Charge.Intern, y

   if (Auf.P.MEH.Preis='Stk') then vM # cnvfi(Auf.P.Prd.Rest.Stk)

   else if (Auf.P.MEH.Preis=Auf.P.MEH.Einsatz) then vM # Auf.P.Prd.Rest;

   Auf_Data_Buchen:DFaktArtC(Art.C.ArtikelNr, Art.C.Adressnr, Art.C.Anschriftnr, Art.C.Charge.Intern, y

   RETURN;

end;


RecBufClear(252);      // ZIELBUFFER LEEREN

gMDI # Lib_GuiCom:AddChildWindow(gMDI,'Art.C.Verwaltung',here+':AusDFaktArtC');

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


vQ # "";

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

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

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

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


vHdl # SelCreate(252, gKey);

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

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

// speichern, starten und Name merken...

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

```

```
// Liste selektieren...
```

```
gZLList->wpDbSelection # vHdl;
```

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

```
end;
```

```
'ProjektSL' : begin
```

```
RecBufClear(120);      // ZIELBUFFER LEEREN
```

```
gMDI # Lib_GuiCom:AddChildWindow(gMDI,'Prj.Verwaltung',here+':AusProjektSL');
```

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

```
gZLList->wpDbFileNo    # 100; // Ausgangsdatei: Adressen
```

```
gZLList->wpDbLinkFileNo # 120; // Zielfile: Projekte
```

```
gZLList->wpDbKeyNo      # 30; // Verknüpfung Adr->Projekte
```

```
// Flag setzen, da vorher KEINE Verknüpfung aktiv war!
```

```
gZLList->wpLstFlags # gZLList->wpLstFlags | _WinLstRecFocusTermReset;
```

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

```
end;
```

```
'MEH' : begin
```

```
Lib_Einheiten:Popup('MEH',$edAuf.P.MEH.Wunsch,401,1,39)
```

```
end;
```

```
'KundenArtNr' : begin
```

```
if (Wgr_Data:IstMix(Auf.P.Wgr.Dateinr)) then begin // 02.03.2020 besser nicht, Proj. 2101/14
```

```
  Auswahl('KundenMatArtNr');
```

```
  RETURN;
```

```
end;
```

```
RecLink(100,401,4,0); // Kunde holen
```

```
RecBufClear(254);
```

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

```
Art_P_Main:Selektieren(gMDI, "", Adr.Nummer);
```

```
/**
```

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

```
gZLList->wpDbFileNo # 100; // Ausgangsdatei: Adressen
```

```
gZLList->wpDbLinkFileNo # 254; // Zielfile: Preise
```

```
gZLList->wpDbKeyNo # 20; // Verknüpfung Adr->Preise (Nummer 20)
```

```
// Flag setzen, da vorher KEINE Verknüpfung aktiv war!
```

```
gZLList->wpLstFlags # gZLList->wpLstFlags | _WinLstRecFocusTermReset;
```

```
vFilter # RecFilterCreate(254,3);
```

```
vFilter->RecFilterAdd(1,_FltAND,_FltEq,Adr.Nummer); // nach AdressNr Filtern
```

```
gZLList->wpDbFilter # vFilter;
```

```
***/
```

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

```
end;
```

```
'Artikelnummer_Mat' : begin
```

```
  if (Wgr_Data:IstMix(Auf.P.Wgr.Dateinr)) then begin
```

```

RecBufClear(250);      // ZIELBUFFER LEEREN

gMDI # Lib_GuiCom:AddChildWindow(gMDI,'Art.Verwaltung',here+':AusArtikelnummer_Mat');

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

gKey # 1;


// Selektion aufbauen...

if (Set.Auf.Artfilter=819) and (Auf.P.Wgr.Dateinr<>0) then begin
    vl # Auf.P.Wgr.Dateinr;

// 22.01.2015      vl # Wgr_Data:WennArtDannCharge(vl);

    vQ # 'LinkCount(WGR) > 0 AND NOT(Art.GesperrtYN)';
    Lib_Sel:QInt(var vQ2, 'Wgr.Dateinummer' , '=', vl);
    vHdl # SelCreate(250, gKey);
    vHdl->SelAddLink(",819, 250, 10, 'WGR');
    tErg # vHdl->SelDefQuery("", vQ);
    if (tErg != 0) then Lib_Sel:QError(vHdl);
    tErg # vHdl->SelDefQuery('WGR', vQ2);
    if (tErg != 0) then Lib_Sel:QError(vHdl);

    // speichern, starten und Name merken...

    w_SelName # Lib_Sel:SaveRun(var vHdl,1,n);

    // Liste selektieren...

    gZLList->wpDbSelection # vHdl;

end

else if (Set.Auf.Artfilter=250) and (Auf.P.Warengruppe<>0) then begin
    vHdl # Winsearch(gMDI,'ZL.Artikel');

    Lib_Sel:QRecList(vHdl,'Art.Warengruppe='+aint(Auf.P.Warengruppe)+' AND NOT(Art.GesperrtYN)')
end

```

else begin

vHdl # Winsearch(gMDI,'ZL.Artikel');

Lib_Sel:QRecList(vHdl,'Art.Nummer>"" AND NOT(Art.GesperrtYN)');

end;

Lib_GuiCom:RunChildWindow(gMDI);

end

else begin

RecBufClear(220); // ZIELBUFFER LEEREN

gMDI # Lib_GuiCom:AddChildWindow(gMDI,'MSL.Verwaltung',here+':AusStruktur');

Lib_GuiCom:RunChildWindow(gMDI);

end;

end;

'Artikelnummer' : begin

RecBufClear(250); // ZIELBUFFER LEEREN

gMDI # Lib_GuiCom:AddChildWindow(gMDI,'Art.Verwaltung',here+':AusArtikelnummer');

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

// gKey # 1;

vHdl # Winsearch(gMDI,'ZL.Artikel');

if (Auf.P.Warengruppe<>0) and (Set.Auf.Artfilter=819) then begin

vQ # 'LinkCount(WGR) > 0 AND NOT(Art.GesperrtYN)';

vl # Auf.P.Wgr.Dateinr;

// 22.01.2015 vl # Wgr_Data:WennArtDannCharge(vl);

Lib_Sel:QInt(var vQ2, 'Wgr.Dateinummer' , '=', vl);

vHdl # SelCreate(250, gKey);


```

vHdl->SelAddLink("",819, 250, 10, 'WGR');

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

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

tErg # vHdl->SelDefQuery('WGR', vQ2);

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

// speichern, starten und Name merken...

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

// Liste selektieren...

gZLList->wpDbSelection # vHdl;

end

else if (Set.Auf.Artfilter=250) and (Auf.P.Warengruppe<>0) then begin

    Lib_Sel:QRecList(vHdl,'Art.Warengruppe='+aint(Auf.P.Warengruppe)+' AND NOT(Art.GesperrtYN)');

end

else begin

    Lib_Sel:QRecList(vHdl,'Art.Nummer>"" AND NOT(Art.GesperrtYN)');

end;

Lib_GuiCom:RunChildWindow(gMDI);

end;

'Vorgangstyp' : begin

    Lib_Einheiten:Popup('Vorgangstyp',$edAuf.Vorgangstyp,400,1,3);

    Refreshfm('edAuf.Vorgangstyp', y);

end;

```

'Vertreter1' : begin

RecBufClear(110); // ZIELBUFFER LEEREN

gMDI # Lib_GuiCom:AddChildWindow(gMDI,'Ver.Verwaltung',here+':AusVertreter1');

Lib_GuiCom:RunChildWindow(gMDI);

end;

'Vertreter2' : begin

RecBufClear(110); // ZIELBUFFER LEEREN

gMDI # Lib_GuiCom:AddChildWindow(gMDI,'Ver.Verwaltung',here+':AusVertreter2');

Lib_GuiCom:RunChildWindow(gMDI);

end;

'Kunde' : begin

RecBufClear(100); // ZIELBUFFER LEEREN

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

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

Lib_Sel:QRecList(0,'Adr.KundenNr > 0');

Lib_GuiCom:RunChildWindow(gMDI);

end;

'Lieferadresse' : begin

RecBufClear(100); // ZIELBUFFER LEEREN

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

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

```
end;
```

```
'Lieferanschrift' : begin
```

```
    RecLink(100,400,12,0);    // Lieferadresse holen
```

```
    RecBufClear(101);        // ZIELBUFFER LEEREN
```

```
    gMDI # Lib_GuiCom:AddChildWindow(gMDI,'Adr.A.Verwaltung',here+':AusLieferanschrift');
```

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

```
    vQ # ";
```

```
    Lib_Sel:QInt(var vQ, 'Adr.A.Adressnr', '=', Adr.Nummer);
```

```
    vHdl # SelCreate(101, 1);
```

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

```
    if (Erx <> 0) then
```

```
        Lib_Sel:QError(vHdl);
```

```
    // speichern, starten und Name merken...
```

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

```
    // Liste selektieren...
```

```
    gZLList->wpDbSelection # vHdl;
```

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

```
end;
```

'Verbraucher' : begin

RecBufClear(100); // ZIELBUFFER LEEREN

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

Lib_GuiCom:RunChildWindow(gMDI);

end;

'Rechnungsempf' : begin

RecBufClear(100); // ZIELBUFFER LEEREN

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

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

Lib_Sel:QRecList(0,'Adr.KundenNr > 0');

Lib_GuiCom:RunChildWindow(gMDI);

end;

'Rechnungsanschr' : begin

RecLink(100,400,4,0); // Rechnungsempfänger holen

RecBufClear(101); // ZIELBUFFER LEEREN

gMDI # Lib_GuiCom:AddChildWindow(gMDI,'Adr.A.Verwaltung',here+':AusRechnungsanschr');

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

vQ # "";

Lib_Sel:QInt(var vQ, 'Adr.A.Adressnr', '=', Adr.Nummer);

vHdl # SelCreate(101, 1);

```

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

if (Erx <> 0) then

    Lib_Sel:QError(vHdl);

    // speichern, starten und Name merken...

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

    // Liste selektieren...

    gZLList->wpDbSelection # vHdl;

    Lib_GuiCom:RunChildWindow(gMDI);

end;

```

```

'Ansprechpartner' : begin

    RecLink(100,400,1,0);    // Kunde holen

    RecBufClear(102);

    gMDI # Lib_GuiCom:AddChildWindow(gMDI,'Adr.P.Verwaltung',here+':AusAnsprechpartner');

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

```

/**

```

gZLList->wpdbfileno    # 100;

gZLList->wpdbkeyno     # 13;

gZLList->wpdbLinkFileNo # 102;

// Flag setzen, da vorher KEINE Verknüpfung aktiv war!

gZLList->wpLstFlags # gZLList->wpLstFlags | _WinLstRecFocusTermReset;

```

***/

```

// Selektion aufbauen...

vQ # "";

```

```

Lib_Sel:QInt(var vQ, 'Adr.P.Adressnr' , '=', Adr.Nummer);

vHdl # SelCreate(102, gKey);

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

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


// speichern, starten und Name merken...

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

// Liste selektieren...

gZLList->wpDbSelection # vHdl;


Lib_GuiCom:RunChildWindow(gMDI);

end;


'BDSNummer' : begin

    RecBufClear(836);      // ZIELBUFFER LEEREN

    gMDI # Lib_GuiCom:AddChildWindow(gMDI,'BDS.Verwaltung',here+':AusBDS');

    Lib_GuiCom:RunChildWindow(gMDI);

end;


'Land' : begin

    RecBufClear(812);      // ZIELBUFFER LEEREN

    gMDI # Lib_GuiCom:AddChildWindow(gMDI,'Lnd.Verwaltung',here+':AusLand');

    Lib_GuiCom:RunChildWindow(gMDI);

end;

```

'Waehrung' : begin

RecBufClear(814);

gMDI # Lib_GuiCom:AddChildWindow(gMDI,'Wae.Verwaltung',here+':AusWaehrung');

Lib_GuiCom:RunChildWindow(gMDI);

end;

'Lieferbed' : begin

RecBufClear(815);

gMDI # Lib_GuiCom:AddChildWindow(gMDI,'LiB.Verwaltung',here+':AusLieferbed');

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

vQ # 'Lib.SperreNeuYN=false';

Lib_Sel:QRecList(0, vQ);

Lib_GuiCom:RunChildWindow(gMDI);

end;

'Zahlungsbed' : begin

RecBufClear(816);

gMDI # Lib_GuiCom:AddChildWindow(gMDI,'Zab.Verwaltung',here+':AusZahlungsbed');

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

vQ # 'ZaB.SperreNeuYN=false';

Lib_Sel:QRecList(0, vQ);

Lib_GuiCom:RunChildWindow(gMDI);

end;

'Steuerschluessel' : begin

RecBufClear(813);

gMDI # Lib_GuiCom:AddChildWindow(gMDI,'StS.Verwaltung',here+':AusSteuerschluessel');

Lib_GuiCom:RunChildWindow(gMDI);

end;

'Versandart' : begin

RecBufClear(817);

gMDI # Lib_GuiCom:AddChildWindow(gMDI,'VsA.Verwaltung',here+':AusVersandart');

Lib_GuiCom:RunChildWindow(gMDI);

end;

'Sachbearbeiter' : begin

RecBufClear(800);

gMDI # Lib_GuiCom:AddChildWindow(gMDI,'Usr.Verwaltung',here+':AusSachbearbeiter');

Lib_GuiCom:RunChildWindow(gMDI);

end;

'Sprache' : begin

Lib_Einheiten:Popup('Sprache',\$edAuf.Sprache,400,1,24);

end;

'AbmessungenEH' : begin

Lib_Einheiten:Popup('AbmessungenEH',\$edAuf.AbmessungenEH,400,1,22);

end;

'GewichtsEH' : begin

Lib_Einheiten:Popup('GewichtsEH',\$edAuf.GewichtsEH,400,1,23);

end;

'Auftragsart' : begin

RecBufClear(835);

gMDI # Lib_GuiCom:AddChildWindow(gMDI,'AAr.Verwaltung',here+':AusAuftragsart');

Lib_GuiCom:RunChildWindow(gMDI);

end;

'Abruf' : begin

if (Auf.Vorgangstyp=c_REKOR) or (Auf.Vorgangstyp=c_GUT) or

(Auf.Vorgangstyp=c_BEL_KD) or (Auf.Vorgangstyp=c_BEL_Lf) then begin

RecBufClear(450);

gMDI # Lib_GuiCom:AddChildWindow(gMDI,'Erl.Verwaltung',here+':AusErloes');

Lib_GuiCom:RunChildWindow(gMDI);

```

RETURN;

end;

if (mode<>c_ModeEdit) then begin

    ProtokollBuffer[400] # RecBufCreate(400);

    RecBufCopy(400,ProtokollBuffer[400]);

    ProtokollBuffer[401] # RecBufCreate(401);

    RecBufCopy(401,ProtokollBuffer[401]);

end;


// Kunde holen

Erx # RecLink(100, 401, 4, _recFirst);

if(Erx > _rLocked) then

    RecBufClear(100);


RecBufClear(401);

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

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


// Selektion 'ABRUF' in 401

vQ # "";

Lib_Sel:QInt(var vQ, 'Auf.P.Nummer', '<', 1000000000);

if (Set.Auf.AbrufEgalWo=false) then

    Lib_Sel:QInt(var vQ, 'Auf.P.Kundennr', '=', Auf.Kundennr);

vQ # vQ + ' AND ("Auf.P.Löschmarker"="")';

vQ # vQ + ' AND LinkCount(AufKopf) > 0 ';

```

```

vQ2 # ' Auf.LiefervertragYN ';

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

// Selektion aufbauen...

vHdl # SelCreate(401, gKey);

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

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

if (tErg <> 0) then

    Lib_Sel:QError(vHdl);

tErg # vHdl->SelDefQuery('AufKopf', vQ2);

if (tErg <> 0) then

    Lib_Sel:QError(vHdl);


// speichern, starten und Name merken...

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

// Liste selektieren...

gZLList->wpDbSelection # vHdl;


Lib_GuiCom:RunChildWindow(gMDI);

end;


'AbrufPos' : begin

    Erl.Rechnungsnr # Auf.P.AbrufAufNr;

    Erx # RecRead(450,1,0); // Erlöse holen

    if (Erx<=_rLocked) then begin

```

```

RecBufClear(451);

gMDI # Lib_GuiCom:AddChildWindow(gMDI,'Erl.K.Verwaltung',here+':AusErloesKonto');

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

gZLList->wpdbFileNo    # 450;

gZLList->wpdbKeyno      # 1;

gZLList->wpdbLinkFileNo # 451;

// Flag setzen, da vorher KEINE Verknüpfung aktiv war!

gZLList->wpLstFlags # gZLList->wpLstFlags | _WinLstRecFocusTermReset;

Lib_GuiCom:RunChildWindow(gMDI);

end;

RETURN;

end;

```

'Kopftext' : begin

```

RecBufClear(837);      // ZIELBUFFER LEEREN

gMDI # Lib_GuiCom:AddChildWindow(gMDI,'Txt.Verwaltung',here+':AusKopftext');

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

if (Auf.Vorgangstyp=c_ANG) then

    Gv.Alpha.01 # 'A'

else

    Gv.Alpha.01 # 'V'

vQ # "";

Lib_Sel:QenthaeltA(var vQ, 'Txt.Bereichstring', Gv.Alpha.01);

Lib_Sel:QRecList(0, vQ);

Lib_GuiCom:RunChildWindow(gMDI);

```

end;

'Fusstext' : begin

RecBufClear(837); // ZIELBUFFER LEEREN

gMDI # Lib_GuiCom:AddChildWindow(gMDI,'Txt.Verwaltung',here+':AusFussText');

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

if (Auf.Vorgangstyp=c_ANG) then

 Gv.Alpha.01 # 'A'

else

 Gv.Alpha.01 # 'V'

vQ # ";

Lib_Sel:QenthaeltA(var vQ, 'Txt.Bereichstring', Gv.Alpha.01);

Lib_Sel:QRecList(0, vQ);

Lib_GuiCom:RunChildWindow(gMDI);

end;

'Warengruppe' : begin

RecBufClear(819);

gMDI # Lib_GuiCom:AddChildWindow(gMDI,'Wgr.Verwaltung',here+':AusWarengruppe');

Lib_GuiCom:RunChildWindow(gMDI);

end;

'Guete' : begin

```
RecBufClear(832);
```

```
gMDI # Lib_GuiCom:AddChildWindow(gMDI,'MQu.Verwaltung',here+':AusGuete');
```

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

```
RecBufClear(848);
```

```
MQu.S.Stufe # "Auf.P.Gütenstufe";
```

```
if (MQu.S.Stufe<>") then begin
```

```
  vQ # ' MQu.NurStufe = '"+MQu.S.Stufe+"' OR MQu.NurStufe = "" ';
```

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

```
end;
```

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

```
end;
```

```
'Guetenstufe' : begin
```

```
  RecBufClear(848);      // ZIELBUFFER LEEREN
```

```
  gMDI # Lib_GuiCom:AddChildWindow(gMDI,'MQu.S.Verwaltung',here+':AusGuetenstufe');
```

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

```
end;
```

```
'AusfOben' : begin
```

```
  vFilter # RecFilterCreate(402, 1);
```

```
  vFilter->RecFilterAdd(1,_FltAND,_FltEq, Auf.P.Nummer);
```

```
  vFilter->RecFilterAdd(2,_FltAND,_FltEq, Auf.P.Position);
```

```
  vFilter->RecFilterAdd(3,_FltAND,_FltEq, '1');
```

```

vTmp # RecLinkInfo(402,401,11,_Reccount,vFilter);

RecFilterDestroy(vFilter);

if (vTmp=0) and ("Set.Wie.Obj.!autoF9"=false) then begin

    RecBufClear(841);      // ZIELBUFFER LEEREN

    gMDI # Lib_GuiCom:AddChildWindow(gMDI,'Obf.Verwaltung',here+':AusEinzelObfOben');

    RunAFX('Auf.P.Obf.Filter',aint(gMDI)+'1');

    Lib_GuiCom:RunChildWindow(gMDI);

    RETURN;

end

RecBufClear(402);

gMDI # Lib_GuiCom:AddChildWindow(gMDI,'Auf.AF.Verwaltung',here+':AusAFOben');

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

vFilter # RecFilterCreate(402, 1);

vFilter->RecFilterAdd(1,_FltAND,_FltEq, Auf.P.Nummer);

vFilter->RecFilterAdd(2,_FltAND,_FltEq, Auf.P.Position);

vFilter->RecFilterAdd(3,_FltAND,_FltEq, '1');

gZLList->wpDbFilter # vFilter;

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

vHdl->wpcustom # '1';

Lib_GuiCom:RunChildWindow(gMDI);

end;

```

'AusfUnten' : begin

```

vFilter # RecFilterCreate(402, 1);

vFilter->RecFilterAdd(1,_FltAND,_FltEq, Auf.P.Nummer);

```

```

vFilter->RecFilterAdd(2,_FltAND,_FltEq, Auf.P.Position);

vFilter->RecFilterAdd(3,_FltAND,_FltEq, '2');

vTmp # RecLinkInfo(402,401,11,_Reccount,vFilter);

RecFilterDestroy(vFilter);

if (vTmp=0) and ("Set.Wie.Obj.!autoF9"=false) then begin

    RecBufClear(841);      // ZIELBUFFER LEEREN

    gMDI # Lib_GuiCom:AddChildWindow(gMDI,'Obf.Verwaltung',here+':AusEinzelObfUnten');

    RunAFX('Auf.P.Obf.Filter',aint(gMDI)+'|2');

    Lib_GuiCom:RunChildWindow(gMDI);

    RETURN;

end

RecBufClear(402);

gMDI # Lib_GuiCom:AddChildWindow(gMDI,'Auf.AF.Verwaltung',here+':AusAFUnten');

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

vFilter # RecFilterCreate(402,1);

vFilter->RecFilterAdd(1,_FltAND,_FltEq, Auf.P.Nummer);

vFilter->RecFilterAdd(2,_FltAND,_FltEq, Auf.P.Position);

vFilter->RecFilterAdd(3,_FltAND,_FltEq, '2');

gZLList->wpDbFilter # vFilter;

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

vHdl->wpcustom # '2';

Lib_GuiCom:RunChildWindow(gMDI);

end;

'KundenMatArtNr' : begin

```



```

Erx # RecLink(100,400,1,_RecFirst);      // Kunde holen

if (Erx<>_rOK) then RecBufClear(100);

RecBufClear(105);

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

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

vQ # "";

if ("Set.Auf.!EigeneVPG") then begin

    Lib_Sel:QInt(var vQ, 'Adr.V.AdressNr', '=', Adr.Nummer);

end

else begin

    Lib_Sel:QInt(var vQ, 'Adr.V.AdressNr', '=', Adr.Nummer);

    Lib_Sel:QInt(var vQ, 'Adr.V.AdressNr', '=', Set.eigeneAdressnr, 'OR');

end;

vQ # 'Adr.V.VerkaufYN AND ('+vQ+)'; // 21.07.2015

Lib_Sel:QRecList(0, vQ);

Lib_GuiCom:RunChildWindow(gMDI);

end;


'Zeugnis' : begin

    RecBufClear(839);

    gMDI # Lib_GuiCom:AddChildWindow(gMDI,'Zeu.Verwaltung',here+':AusZeugnis');

    Lib_GuiCom:RunChildWindow(gMDI);

end;

```

'Erzeuger' : begin

RecBufClear(100); // ZIELBUFFER LEEREN

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

Lib_GuiCom:RunChildWindow(gMDI);

end;

'Intrastat' : begin

if (Wgr_Data:IstArt(Auf.P.Wgr.Dateinr)=false) then begin

if (Msg(220001,"0,_WinDialogYesNo,1)=_WinIdYes) then begin

 // Selektion

 vQ # ";

 Lib_Sel:QAlpha(var vQ, 'MSL.Strukturtyp', '=', 'INTRA');

 Lib_Sel:QAlpha(var vQ, 'MSL.Intrastatnr', '>', "');

 Lib_Sel:QInt(var vQ, 'MSL.von.Warengruppe', '<=', Auf.P.Warengruppe);

 Lib_Sel:QInt(var vQ, 'MSL.bis.Warengruppe', '>=', Auf.P.Warengruppe);

 vQ # vQ + ' AND ("MSL.Güte" = "Auf.P.Güte" OR "MSL.Güte" = "" OR "Auf.P.Güte" = "") ';

 vQ # vQ + ' AND ("MSL.Gütenstufe" = "Auf.P.Gütenstufe" OR "MSL.Gütenstufe" = "" OR "Auf.P.Güte

 vQ # vQ + ' AND (Auf.P.Dicke = 0.0 OR (MSL.von.Dicke <= Auf.P.Dicke AND MSL.bis.Dicke >= Auf.

 vQ # vQ + ' AND (Auf.P.Breite = 0.0 OR (MSL.von.Breite <= Auf.P.Breite AND MSL.bis.Breite >= Au

 vQ # vQ + ' AND ("Auf.P.Länge" = 0.0 OR ("MSL.von.Länge" <= "Auf.P.Länge" AND "MSL.bis.Länge

 vSel2 # SelCreate(220, 2);

 tErg # vSel2->SelDefQuery("", vQ);

 if (tErg != 0) then Lib_Sel:QError(vSel2);

 vSelName # Lib_Sel:SaveRun(var vSel2, 0);

```
RecBufClear(220);
```

```
gMDI # Lib_GuiCom:AddChildWindow(gMDI,'MSL.Verwaltung',here+':AusIntrastat');
```

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

```
gZLList->wpDbSelection # vSel2;
```

```
w_SelName # vSelName;
```

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

```
RETURN;
```

```
end;
```

```
end;
```

```
RecBufClear(220);
```

```
gMDI # Lib_GuiCom:AddChildWindow(gMDI,'MSL.Verwaltung',here+':AusIntrastat');
```

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

```
end;
```

```
'PreisMEH' : begin
```

```
if ($NB.Page1->wpcustom='NB.Page1_Art') then
```

```
Lib_Einheiten:Popup('MEH',$edAuf.P.MEH.Preis,401,1,41)
```

```
else
```

```
Lib_Einheiten:Popup('MEH',$edAuf.P.MEH.Preis_Mat,401,1,41)
```

```
end;
```

```
'Terminart' : begin
```

```

if ($NB.Page1->wpcustom='NB.Page1_Art') then

    Lib_Einheiten:Popup('Datumstyp',$edAuf.P.Termin1W.Art,401,1,48)

else

    Lib_Einheiten:Popup('Datumstyp',$edAuf.P.Termin1W.Art_Mat,401,1,48);

end;

```

```

'KopfAufpreise' : begin

```

```

    if (RunAFX('Auf.P.Auswahl.Aufpreise','Kopf')<>0) then RETURN;

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

    vHdl->wpcustom # cnvai(Auf.P.Position,_FmtNumNoGroup,0,5)+CnvAI(Winfocusget(),_FmtNumNogro

    RecBufClear(403);

    // MUSTER

    gMDI # Lib_GuiCom:AddChildWindow(gMDI,'Auf.Z.Verwaltung',here+':AusKopfAufpreise',y);

    Lib_GuiCom:RunChildWindow(gMDI);

    Auf.P.Position # 0;

end;

```

```

'Aufpreise' : begin

```

```

    if (RunAFX('Auf.P.Auswahl.Aufpreise','Pos')<>0) then RETURN;

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

        if (Msg(842000,gTitle,_WinIcoQuestion,_WinDialogYesNo,1)=_Winidyes) then begin

            ApL_Data:AutoGenerieren(401);

        end;

    end;

end;

```

```

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

vHdl->wpcustom # cnvai(Auf.P.Position,_FmtNumNoGroup,0,5)+CnvAI(Winfocusget(),_FmtNumNogro

RecBufClear(403);

gMDI # Lib_GuiCom:AddChildWindow(gMDI,'Auf.Z.Verwaltung',here+':AusAufpreise',y);

Lib_GuiCom:RunChildWindow(gMDI);

end;

```

'Preis' : begin

```

if (Auf.P.ArtikelNr='') then RETURN;

```

```

if (Wgr_Data:IstMat(Auf.P.Wgr.Dateinr)) then RETURN;

```

```

Erx # RekLink(250,401,2,_RecFirst); // Artikel holen

```

```

RecBufClear(254);

```

```

gMDI # Lib_GuiCom:AddChildWindow(gMDI,'Art.P.Verwaltung',here+':AusPreis');

```

```

Art_P_Main:Selektieren(gMDI, Art.Nummer, 0);

```

```

Lib_GuiCom:RunChildWindow(gMDI);

```

```

end;

```

'Stueckliste' : begin

```

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

```

```

vHdl->wpcustom # cnvai(Auf.P.Position,_FmtNumNoGroup,0,5)+CnvAI(Winfocusget(),_FmtNumNogro

```

```

RecBufClear(409);

```

```

gMDI # Lib_GuiCom:AddChildWindow(gMDI,'Auf.SL.Verwaltung',here+':AusStueckliste',y);
vHdl # gMDI->winsearch('NB.Main');
Lib_GuiCom:RunChildWindow(gMDI);
end;

```

'Kalkulation' : begin

```

vHdl # winsearch(gMDI, 'NB.Main');
vHdl->wpcustom # cnvai(Auf.P.Position,_FmtNumNoGroup,0,5)+CnvAI(Winfocusget(),_FmtNumNogro
RecBufClear(405);
gMDI # Lib_GuiCom:AddChildWindow(gMDI,'Auf.K.Verwaltung',here+':AusKalkulation',y);
Lib_GuiCom:RunChildWindow(gMDI);
end;

```

'Aktionen' : begin

```

RecBufClear(404);
gMDI # Lib_GuiCom:AddChildWindow(gMDI,'Auf.A.Verwaltung',here+':AusAktion',y);
VarInstance(WindowBonus,cnvIA(gMDI->wpcustom));
Lib_Sel:QInt(var vQ, 'Auf.A.Nummer' , '=', Auf.P.Nummer);
vQ # vQ +' AND (';
Lib_Sel:QInt(var vQ, 'Auf.A.Position' , '=', 0, ' ');
Lib_Sel:QInt(var vQ, 'Auf.A.Position' , '=', Auf.P.Position, 'OR');
vQ # vQ + ')';
Lib_Sel:QRecList(0, vQ);
Lib_GuiCom:RunChildWindow(gMDI);

```

end;

'Text2' : begin

RecBufClear(837); // ZIELBUFFER LEEREN

gMDI # Lib_GuiCom:AddChildWindow(gMDI,'Txt.Verwaltung',here+':AusText1');

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

if (Auf.Vorgangstyp=c_ANG) then

 Gv.Alpha.01 # 'A'

else

 Gv.Alpha.01 # 'V'

vQ # ";

Lib_Sel:QenthaeltA(var vQ, 'Txt.Bereichstring', Gv.Alpha.01);

Lib_Sel:QRecList(0, vQ);

Lib_GuiCom:RunChildWindow(gMDI);

end;

'Projekt' : begin

RecBufClear(120); // ZIELBUFFER LEEREN

gMDI # Lib_GuiCom:AddChildWindow(gMDI,'Prj.Verwaltung',here+':AusProjekt');

Lib_GuiCom:RunChildWindow(gMDI);

end;

'Zwischenlage' : begin

```
RecBufClear(838);

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

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

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

Lib_Sel:QRecList(0, vQ);

Lib_GuiCom:RunChildWindow(gMDI);

end;
```

'Unterlage' : begin

```
RecBufClear(838);

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

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

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

Lib_Sel:QRecList(0, vQ);

Lib_GuiCom:RunChildWindow(gMDI);

end;
```

'Umverpackung' : begin

```
RecBufClear(838);

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

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

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

Lib_Sel:QRecList(0, vQ);

Lib_GuiCom:RunChildWindow(gMDI);
```


end;

'Verwiegungsart' : begin

RecBufClear(818); // ZIELBUFFER LEEREN

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

Lib_GuiCom:RunChildWindow(gMDI);

end;

'Etikett' : begin

RecBufClear(840);

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

Lib_GuiCom:RunChildWindow(gMDI);

end;

'Etikett2' : begin

RecBufClear(840);

gMDI # Lib_GuiCom:AddChildWindow(gMDI,'Eti.Verwaltung',here+':AusEtikettentyp2');

Lib_GuiCom:RunChildWindow(gMDI);

end

otherwise

RunAFX('Auf.P.Auswahl',aBereich);

end;

end;

//=====

// AusEinzelObfOben

//

//=====

sub AusEinzelObfOben()

local begin

 vFilter : int;

 vTmp : int;

end;

begin

 if (gSelected<>0) then begin

 RecBufClear(402);

 gMDI # Lib_GuiCom:AddChildWindow(gMDI,'Auf.AF.Verwaltung',here+':AusAFOBen');

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

 vFilter # RecFilterCreate(402,1);

 vFilter->RecFilterAdd(1,_FltAND,_FltEq, Auf.P.Nummer);

 vFilter->RecFilterAdd(2,_FltAND,_FltEq, Auf.P.Position);

 vFilter->RecFilterAdd(3,_FltAND,_FltEq, '1');

 gZLList->wpDbFilter # vFilter;

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

 vTmp->wpcustom # '1';

```

Mode # c_modeBald + c_modeNew;

w_Command # 'SETOBF:.';

w_cmd_para # aint(gSelected);

gSelected # 0;


Lib_GuiCom:RunChildWindow(gMDI);


RETURN;

end;

end;


//=====

// AusEinzelObfUnten

//

//=====

sub AusEinzelObfUnten()

local begin

    vFilter : int;

    vTmp    : int;

end;

begin

    if (gSelected<>0) then begin

        RecBufClear(402);

        gMDI # Lib_GuiCom:AddChildWindow(gMDI,'Auf.AF.Verwaltung',here+':AusAFUnten');

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

```

```
vFilter # RecFilterCreate(402,1);  
  
vFilter->RecFilterAdd(1,_FltAND,_FltEq, Auf.P.Nummer);  
  
vFilter->RecFilterAdd(2,_FltAND,_FltEq, Auf.P.Position);  
  
vFilter->RecFilterAdd(3,_FltAND,_FltEq, '2');  
  
gZLList->wpDbFilter # vFilter;  
  
vTmp # winsearch(gMDI, 'NB.Main');  
  
vTmp->wpcustom # '2';
```

```
  
Mode # c_modeBald + c_modeNew;  
  
w_Command # 'SETOBF:';  
  
w_cmd_para # aint(gSelected);  
  
gSelected # 0;
```

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

```
  
RETURN;
```

```
end;
```

```
end;
```

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

```
// AusAktion
```

```
//
```

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

```
sub AusAktion()
```

```
begin
```

```

gSelected # 0;

RefreshList(gZllist, _WinLstRecFromRecid | _WinLstRecDoSelect);

end;


//=====

// AusRes

//

//=====

sub AusRes()

local begin

    vM : float;

    vGew : float;

    vStk : int;

end;

begin

    gSelected # 0;


    // Reservierungen loopen... 07.02.2020 AH: wird dirket bei Anlage/Edit der Reserv. gemacht bzw. nur noch

    RefreshList(gZllist, _WinLstRecFromRecid | _WinLstRecDoSelect);

end;


//=====

// AusText1

//

```

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

```
sub AusText1()
```

```
local begin
```

```
    vTxtHdl    : int;
```

```
    vHdl       : int;
```

```
end;
```

```
begin
```

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

```
        RecRead(837,0,_RecId,gSelected);
```

```
        gSelected # 0;
```

```
        vTxtHdl # $Auf.P.TextEditPos->wpdbTextBuf;
```

```
        Lib_Texte:TxtLoadLangBuf('~837.'+CnvAI(Txt.Nummer,_FmtNumLeadZero | _FmtNumNoGroup,0,8), v
```

```
        RunAFX( 'Auf.P.AusText', 'Pos|'+aint(vTxtHdl));
```

```
        $Auf.P.TextEditPos->WinUpdate(_WinUpdBuf2Obj);
```

```
        // Ausgewählten Text in das Feld eintragen
```

```
        Auf.P.TextNr1 # 0;
```

```
        Auf.P.TextNr2 # 0;
```

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

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

```
        $edAuf.P.TextNr2->wpCaptionint # 0;
```

```
        $edAuf.P.TextNr2->Winupdate(_WinUpdFld2Obj);
```

```

Lib_GuiCOM:Disable($edAuf.P.TextNr2);

Lib_GuiCOM:Enable($edAuf.P.TextNr2b);

Auf.P.TextNr2 # Txt.Nummer;

$edAuf.P.TextNr2b->wpCaptionint # Txt.Nummer;

$edAuf.P.TextNr2b->Winupdate(_WinUpdFld2Obj);


// ggf. Labels refreshen

$cb.Text1->wpCheckState # _WinStateChkUnchecked;

$cb.Text2->wpCheckState # _WinStateChkChecked;

$cb.Text3->wpCheckState # _WinStateChkUnchecked;


// Focus auf Editfeld setzen:

$cb.Text2->Winfocusset(true);

end;

gSelected # 0;


end;


//=====

// AusLFS

//

//=====

sub AusLFS()

local begin

```

```

vTmp : int;

end;

begin

RefreshList(gZLList, _WinLstRecFromRecid | _WinLstRecDoSelect);


// ggf. Auftragskopfdaten liste aktualisieren

vTmp # winsearch(gMDI,'ZL.Auftraege');

if (vTmp <> 0) then

vTmp->WinUpdate(_WinUpdOn, _WinLstRecFromRecID | _WinLstRecDoSelect);


// direkt drucken

if (gSelected<>0) then begin

gSelected # 0;

// LFS.Nummer # Auf.A.Aktionsnr;

// 25.08.2020 AH: unterscheiden, ob LFA oder LFS

Lfs.Nummer # Lfs.P.Nummer;

RecRead(440,1,0);

w_TimerVar # '';

if (Set.LFS.Verbuchen='B') and (Lfs.zuBA.Nummer=0) then

w_TimerVar # w_TimerVar + '|LFSVERBUCHEN';

if (Set.LFS.SofortDVLDYN) then

w_TimerVar # w_TimerVar + '|VLDAW';

if (Set.LFS.SofortDLFAYN) and (Lfs.zuBA.Nummer<>0) then

w_TimerVar # w_TimerVar + '|LFA';

if (Set.LFS.SofortDLFSYN) then

w_TimerVar # w_TimerVar + '|LFS';

```



```

    if (w_TimerVar<>"") then

        gTimer2 # SysTimerCreate(500,1,gMdiAuf);

    end;

    gSelected # 0;

end;


//=====

// AusMatzMat

//

//=====

sub AusMatzMat()

local begin

    Erx      : int;

    vHdl     : int;

    vItem    : int;

    vMFile   : int;

    vMID     : int;

    vAnz     : int;

    vOK      : logic;

    vA       : alpha(4000);

end;

begin

    if (gSelected=0) then RETURN;

    RecRead(200,0,_RecId,gSelected);

```

```
// Feldübernahme
```

```
gSelected # 0;
```

```
vltem # gMarkList->CteRead(_CteFirst);
```

```
WHILE (vltem > 0) do begin
```

```
  Lib_Mark:TokenMark(vltem,var vMFile,var vMID);
```

```
  if (vMFile=200) then inc(vAnz);
```

```
  vltem # gMarkList->CteRead(_CteNext, vltem);
```

```
END;
```

```
// markierte Sätze übernehmen?
```

```
if (vAnz>0) then begin
```

```
  if (Msg(401212,cnvai(vAnz),_WinlcoQuestion,_WinDialogYesNo,2)=_WinIdYes) then begin
```

```
    // 31.07.2017 AH: Prüflauf...
```

```
    vltem # gMarkList->CteRead(_CteFirst);
```

```
    WHILE (vltem > 0) do begin
```

```
      Lib_Mark:TokenMark(vltem,var vMFile,var vMID);
```

```
      if (vMFile=200) then begin
```

```
        RecRead(200,0,_RecId,vMID);
```

```
        Erx # Auf_Data:PasstAuf2Mat(0,false, true);
```

```
        if (erx<0) then RETURN;
```

```
        if (erx=0) then begin
```

```
          if (Rechte[Rgt_Auf_MATZ_Konf_Abm]) then begin
```

```
            if (StrLen(vA)<1000) then begin
```

```
if (vA<>") then
```

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

```
    vA # vA + aint(Mat.Nummer);
```

```
end;
```

```
end
```

```
else begin
```

```
    Msg(401016,aint(Mat.Nummer),_WinIcoError,_WinDialogOk,1);
```

```
    RETURN;
```

```
end;
```

```
end;
```

```
end; // 200?
```

```
vltem # gMarkList->CteRead(_CteNext,vltem);
```

```
END;
```

```
if (vA<>") then begin
```

```
    if (StrLen(vA)>=1000) then vA # 'SEHR VIELE';
```

```
    if (Msg(401015,vA,_WinIcoQuestion,_WinDialogYesNo,2)<>_WinIdYes) then
```

```
        RETURN;
```

```
end;
```

```
// Buchlauf...
```

```
vltem # gMarkList->CteRead(_CteFirst);
```

```
WHILE (vltem > 0) do begin
```

```
    Lib_Mark:TokenMark(vltem,var vMFile,var vMID);
```

```
if (vMFile=200) then begin
```

```
    RecRead(200,0,_RecId,vMID);
```

```
    vOK # Auf_Data:MatzMat(y);
```

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

```
        ErrorOutput;
```

```
    end;
```

```
end;
```

```
vltem # gMarkList->CteRead(_CteNext,vltem);
```

```
END;
```

```
// Markierungen löschen...
```

```
Lib_Mark:Reset(200);
```

```
Msg(999998,"0,0,0,0); // ERFOLG
```

```
end;
```

```
end
```

```
else begin
```

```
    vOK # Auf_Data:MatzMat(n);
```

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

```
        ErrorOutput;
```

```
    end;
```

```
    vHdl # winsearch(gMDI,cZList);
```

```
    vHdl->WinUpdate(_WinUpdOn, _WinLstRecFromRecid | _WinLstRecDoSelect);
```

```
    Auswahl('Matz.Mat');
```

```

end;

/**
if (gSelected<>0) then begin
    RecRead(200,0,_RecId,gSelected);
    // Feldübernahme
    gSelected # 0;

    if (Auf_Data:MatzMat(n)=false) then begin
        ErrorOutput;
    end;

    vHdl # winsearch(gMDI,cZList);
    vHdl->WinUpdate(_WinUpdOn, _WinLstRecFromRecid | _WinLstRecDoSelect);
    Auswahl('Matz.Mat');
end;
**/

// Focus auf Editfeld setzen:

// ggf. Labels refreshen

end;


//=====

// AusMatzMatGuBe

//

//=====

sub AusMatzMatGuBe()

```

local begin

 vHdl : int;

 vItem : int;

 vMFile : int;

 vMID : int;

 vAnz : int;

end;

begin

 if (gSelected=0) then RETURN;

 RecRead(200,0,_RecId,gSelected);

 // Feldübernahme

 gSelected # 0;

 vItem # gMarkList->CteRead(_CteFirst);

 WHILE (vItem > 0) do begin

 Lib_Mark:TokenMark(vItem,var vMFile,var vMID);

 if (vMFile=200) then inc(vAnz);

 vItem # gMarkList->CteRead(_CteNext,vItem);

 END;

 // markierte Sätze übernehmen?

 if (vAnz>0) then begin

 if (Msg(401212,cnvai(vAnz),_WinIcoQuestion,_WinDialogYesNo,2)=_WinIdYes) then begin

```
vltem # gMarkList->CteRead(_CteFirst);  
  
WHILE (vltem > 0) do begin  
  
    Lib_Mark:TokenMark(vltem,var vMFile,var vMID);  
  
    if (vMFile=200) then begin  
  
        RecRead(200,0,_RecId,vMID);  
  
        Auf_Data:MatzMatGuBe(Mat.Nummer);  
  
    end;  
  
    vltem # gMarkList->CteRead(_CteNext,vltem);  
  
END;
```

```
// Markierungen löschen...  
  
Lib_Mark:Reset(200);  
  
Msg(999998,"0,0,0,0); // ERFOLG  
  
end;
```

end

else begin

```
Auf_Data:MatzMatGuBe(Mat.Nummer);
```

```
vHdl # winsearch(gMDI,cZList);  
  
vHdl->WinUpdate(_WinUpdOn, _WinLstRecFromRecid | _WinLstRecDoSelect);  
  
Auswahl('Matz.Mat.GuBe');  
  
end;
```

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

```
// ggf. Labels refreshen
```

```
end;
```

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

```
// AusMaterial
```

```
//
```

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

```
sub AusMaterial()
```

```
local begin
```

```
    Erx      : int;
```

```
    vStk     : int;
```

```
    vMenge   : float;
```

```
    vHdl     : int;
```

```
    vl       : int;
```

```
end;
```

```
begin
```

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

```
        RecRead(200,0,_RecId,gSelected);
```

```
        // Feldübernahme
```

```
        gSelected # 0;
```

```
        Auf.P.KundenArtNr    # ";
```

```
        Auf.P.VpgText1       # ";
```

```
        Auf.P.VpgText2       # ";
```

```
        Auf.P.VpgText3       # ";
```


Auf.P.VpgText4 # " ";
Auf.P.VpgText5 # " ";
Auf.P.VpgText6 # " ";
"Auf.P.Güte" # "Mat.Güte";
"Auf.P.Gütenstufe" # "Mat.Gütenstufe";

Auf.P.Werkstoffnr # MQu_data:GetWerkstoffnr("Auf.P.Güte");

// 25.06.2012 AI

Auf.P.Warengruppe # Mat.Warengruppe;
Erx # RecLink(819,401,1,_RecFirst); // Warengruppe holen
Auf_Data:SetWgrDateinr(Wgr.Dateinummer);

Auf.P.Artikelnr # Mat.Strukturnr;
Auf.P.AusfOben # "Mat.AusführungOben";
Auf.P.AusfUnten # "Mat.AusführungUnten";
Auf.P.Dicke # Mat.Dicke;
Auf.P.DickenTol # Mat.DickenTol;
Auf.P.Breite # Mat.Breite;
Auf.P.BreitenTol # Mat.BreitenTol;
"Auf.P.Länge" # "Mat.Länge";
"Auf.P.LängenTol" # "Mat.LängenTol";
Auf.P.RID # Mat.RID;
Auf.P.RIDmax # Mat.RID;
Auf.P.RAD # Mat.RAD;
Auf.P.RADmax # Mat.RAD;

Auf.P.Zeugnisart # Mat.Zeugnisart;
Auf.P.AbbindungL # Mat.AbbindungL;
Auf.P.AbbindungQ # Mat.AbbindungQ;
Auf.P.Zwischenlage # Mat.Zwischenlage;
Auf.P.Unterlage # Mat.Unterlage;
Auf.P.Umverpackung # Mat.Umverpackung;
Auf.P.Wicklung # Mat.Wicklung;
// Auf.P.MitLfEYN # Mat.MitLfEYN;
Auf.P.StehendYN # Mat.StehendYN;
Auf.P.LiegendYN # Mat.LiegendYN;
Auf.P.Nettoabzug # Mat.Nettoabzug;
"Auf.P.Stapelhöhe" # "Mat.Stapelhöhe";
Auf.P.StapelhAbzug # "Mat.StapelhöhenAbzug";
Auf.P.RingKgVon # 0.0;
Auf.P.RingKgBis # 0.0;
Auf.P.KgmmVon # Mat.Kgmm;
Auf.P.KgmmBis # Mat.Kgmm;
"Auf.P.StückProVE" # 0;
Auf.P.VEkgMax # 0.0;
Auf.P.RechtwinkMax # Mat.Rechtwinkligkeit;
Auf.P.EbenheitMax # Mat.Ebenheit;
"Auf.P.SäbeligkeitMax" # "Mat.Säbeligkeit";
"Auf.P.SäbelProM" # "Mat.SäbelProM";
Auf.P.Etikettentyp # 0;
Auf.P.Verwiegungsart # Mat.Verwiegungsart;
Auf.P.Intrastatnr # Mat.Intrastatnr;

Auf.P.Gewicht # Mat.Bestand.Gew + Mat.Bestellt.Gew;

"Auf.P.Stückzahl" # Mat.Bestand.Stk + Mat.Bestellt.Stk;

Auf.P.MEH.Einsatz # 'kg';

Auf.P.MEH.Wunsch # 'kg';

Auf.P.Menge # Auf.P.Gewicht;

RunAFX('Auf.P.Drop.Mat',''); // 18.03.2019

Auf.P.Materialnr # 0;

if (Auf.Vorgangstyp = c_Auf) then begin // falls Auftrag

 vl # Set.Auf.MatNr.Ablauf;

if (vl=99) then begin

 if (Msg(401207,"",0,0,0)=_Winidyes) then vl # 2

 else if (Msg(401209,"",0,0,0)=_Winidyes) then vl # 3;

end;

// 01.07.2020 AH: Antwort merken!

Lib_RmtData:UserWrite('400|'+aint(Auf.P.Position)+'|Mat', aint(vl));

// Reservieren?

if (vl=2) then begin

 Auf.P.Materialnr # 0 - Mat.Nummer;

end

// Kommissionieren?

else if (vl=3) then begin

Auf.P.MaterialNr # Mat.Nummer;

end

// Dfakt ?

else if (vl=4) then begin

Auf.P.MaterialNr # Mat.Nummer;

end;

end

else if (Auf.Vorgangstyp = c_Ang) then begin // falls Angebot

if (Msg(401207, ", _WinIcoQuestion, _WinDialogYesNo, 2) = _WinIdyes) then // Fragen ob Reservierung

Auf.P.MaterialNr # 0 - Mat.Nummer // NEGIEREN = auch Reservieren

else

Auf.P.MaterialNr # Mat.Nummer;

end;

// Erzeuger [09.03.2010/PW]

Auf.P.erzeuger # Mat.Erzeuger;

// Ausführungen löschen & kopieren

WHILE (RecLink(402,401,11,_recFirst)=_rOK) do

Erx # RekDelete(402,0,'MAN');

Erx # RecLink(201,200,11,_recFirst);

WHILE (Erx<=_rLocked) do begin

```
Auf.AF.Nummer      # Auf.P.Nummer;

Auf.AF.Position    # Auf.P.Position;

Auf.AF.Seite       # Mat.AF.Seite;

Auf.AF.IfzNr       # Mat.AF.IfzNr;

Auf.AF.ObjNr       # Mat.AF.ObjNr;

Auf.AF.Bezeichnung # Mat.AF.Bezeichnung;

Auf.AF.Zusatz      # Mat.AF.Zusatz;

Auf.AF.Bemerkung   # Mat.AF.Bemerkung;

"Auf.AF.Kürzel"    # "Mat.AF.Kürzel";

Erx # RekInsert(402,0,'MAN');

Erx # RecLink(201,200,11,_recNext);

END;
```

```
Auf.P.Auftragsart # Set.Auf.Auftragsart;
```

```
RunAFX('Auf.P.AusMaterial.Post','');
```

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

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

```
end;
```

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

```
$edAuf.P.Guete_Mat->Winfocusset(true);
```

```

Refreshfm();

// ggf. Labels refreshen

gMdi -> WinUpdate(_WinUpdFld2Obj);

end;


//=====

// AusDFaktMat

//

//=====

sub AusDFaktMat()

local begin

    vStk      : int;

    vMenge    : float;

    vHdl      : int;

    vItem     : int;

    vMFile    : int;

    vMID      : int;

    vAnz      : int;

    vDatum    : date;

    vZeit     : time;

end;

begin

    if (gSelected=0) then RETURN;

```

```
RecRead(200,0,_RecId,gSelected);
```

```
// Feldübernahme
```

```
gSelected # 0;
```

```
vltem # gMarkList->CteRead(_CteFirst);
```

```
WHILE (vltem > 0) do begin
```

```
    Lib_Mark:TokenMark(vltem,var vMFile,var vMID);
```

```
    if (vMFile=200) then inc(vAnz);
```

```
    vltem # gMarkList->CteRead(_CteNext,vltem);
```

```
END;
```

```
// markierte Sätze übernehmen?
```

```
if (vAnz>0) then begin
```

```
    if (Msg(401208,cnvai(vAnz),_WinIcoQuestion,_WinDialogYesNo,2)=_WinIdYes) then begin
```

```
        if (Dlg_Standard:Datum(Translate('Verbuchungsdatum'), var vDatum, today)=false) then RETURN;
```

```
        if (vDatum=today) then vZeit # now;
```

```
        vltem # gMarkList->CteRead(_CteFirst);
```

```
        WHILE (vltem > 0) do begin
```

```
            Lib_Mark:TokenMark(vltem,var vMFile,var vMID);
```

```
            if (vMFile=200) then begin
```

```
                RecRead(200,0,_RecId,vMID);
```

```
                Auf_Data:DFaktMat(Mat.Nummer,n, vDatum, vZeit);
```

```
            end;
```

```
            vltem # gMarkList->CteRead(_CteNext,vltem);
```

END;

// Markierungen löschen...

Lib_Mark:Reset(200);

Msg(999998,"0,0,0,0); // ERFOLG

end;

end

else begin

Auf_Data:DFaktMat(Mat.Nummer,y, today, now);

vHdl # winsearch(gMDI,cZList);

vHdl->WinUpdate(_WinUpdOn, _WinLstRecFromRecid | _WinLstRecDoSelect);

Auswahl('DFakt.Mat');

end;

// Focus auf Editfeld setzen:

// ggf. Labels refreshen

end;

//=====

// AusMatzArtC

//


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

```
sub AusMatzArtC()
```

```
begin
```

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

```
    RecRead(252,0,_RecId,gSelected);
```

```
    // Feldübernahme
```

```
    gSelected # 0;
```

```
    Auf_Data_Buchen:MatzArt(Art.C.Artikelnr, Art.C.Adressnr,Art.C.Anschriftnr,Art.C.Charge.Intern,y,n,0.0,0.0,
```

```
end;
```

```
// ggf. Labels refreshen
```

```
end;
```

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

```
// AusDFaktArtC
```

```
//
```

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

```
sub AusDFaktArtC()
```

```
local begin
```

```
    vStk      : int;
```

```
    vMenge    : float;
```

```
end;
```

```
begin
```

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

```
    RecRead(252,0,_RecId,gSelected);
```

```
    // Feldübernahme
```

```

    gSelected # 0;

//  Auf_Data:DFaktArtC();

    Auf_Data_Buchen:DFaktArtC(Art.C.Artikelnr, Art.C.Adressnr, Art.C.Anschriftnr, Art.C.Charge.Intern, y,0.

    RefreshList(gZLList, _WinLstRecFromRecid | _WinLstRecDoSelect);

end;

// Focus auf Editfeld setzen:

// ggf. Labels refreshen

end;

```

```

//=====

```

```

// AusKundenArtNr

```

```

//

```

```

//=====

```

```

sub AusKundenArtNr()

```

```

local begin

```

```

    Erx  : int;

```

```

    vOk  : logic;

```

```

    vHdl : int;

```

```

end;

```

```

begin

```

```

    if (gSelected<>0) then begin

```

```

        RecRead(254,0,_RecId,gSelected);

```

```

        gSelected # 0;

```

```

        // Feldübernahme

```

```

Erx # RecLink(250,254,2,0);

if (Erx <= _rLocked) then begin

    Auf.P.ArtikelNr  # Art.Nummer;

    $edAuf.P.ArtikelNr->winupdate(_WinUpdFld2Obj);

    RefreshIfm('edAuf.P.ArtikelNr',y);

    $edAuf.P.GrundPreis->Winupdate(_WinUpdFld2Obj);

    vOk # y;

end;

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

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

end;


// Focus auf Editfeld setzen:

if (vOK) then begin

    $edAuf.P.KundenArtNr->winupdate(_WinUpdFld2Obj);

    $edAuf.P.Menge.Wunsch->Winfocusset(true);

end

else begin

    $edAuf.P.KundenArtNr->Winfocusset(false);

end;


gMDI->Winupdate();

// ggf. Labels refreshen

RefreshIfm();

end;

```

```

//=====

// AusArtikelnummer

//

//=====

sub AusArtikelnummer()

local begin

    vOk : logic;

    vHdl : int;

end;

begin

    if (gSelected<>0) then begin

        RecRead(250,0,_RecId,gSelected);

        gSelected # 0;

        // Feldübernahme

        Auf.P.ArtikelNr # Art.Nummer ;

        Auf_P_SMain:GetArtikel();

    /**

        $edAuf.P.ArtikelNr->winupdate(_WinUpdFld2Obj);

        RefreshIfm('edAuf.P.ArtikelNr',y);

        $edAuf.P.GrundPreis->Winupdate(_WinUpdFld2Obj);

    **/

```

```

vOk # y;

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

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

end;

// Focus auf Editfeld setzen:

if (vOK) then begin

    $edAuf.P.ArtikelNr->winupdate(_WinUpdFld2Obj);

    $edAuf.P.MEH.Wunsch->Winfocusset(true);

    // 17.05.2016 AH:

    $edAuf.P.Bemerkung2->Winfocusset(true);

//    $edAuf.P.MEH.Wunsch->Winfocusset(true);

end

else begin

    $edAuf.P.ArtikelNr->Winfocusset(true);

end;

gMDI->Winupdate();

// ggf. Labels refreshen

Refreshfm();

end;

//=====

// AusArtikelnummer_Mat

```

```
//
```

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

```
sub AusArtikelnummer_Mat()
```

```
local begin
```

```
    vHdl    : int;
```

```
end;
```

```
begin
```

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

```
        RecRead(250,0,_RecId,gSelected);
```

```
        gSelected # 0;
```

```
        // Feldübernahme...
```

```
        Auf.P.ArtikelNr    # Art.Nummer;
```

```
        Auf_P_SMain:GetArtikel_Mat();
```

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

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

```
    end;
```

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

```
    $edAuf.P.ArtikelNr_Mat->Winfocusset(true);
```

```
    gMDI->Winupdate();
```

```
    // ggf. Labels refreshen
```

```
Refreshfm();
```

```
end;
```

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

```
// AusStruktur
```

```
//
```

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

```
sub AusStruktur()
```

```
local begin
```

```
    vHdl : int;
```

```
end;
```

```
begin
```

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

```
        RecRead(220,0,_RecId,gSelected);
```

```
        gSelected # 0;
```

```
        // Feldübernahme
```

```
        Auf.P.StrukturNr # MSL.StrukturNR;
```

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

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

```
        $edAuf.P.ArtikelNr_Mat->wpcaption # Auf.P.StrukturNr;
```

```
end;
```

```

// Focus auf Editfeld setzen:

$edAuf.P.ArtikelNr_Mat->Winfocusset(true);

// ggf. Labels refreshen

Refreshfm('edAuf.P.ArtikelNr_Mat',y);


gMDI->Winupdate();

end;


//=====

// AusKundenMatArtNr

//

//=====

sub AusKundenMatArtNr()

local begin

    Erx      : int;

    vHdl     : int;

    vHatAP   : logic;

end

begin

    if (gSelected<>0) then begin

        RecRead(105,0,_RecId,gSelected);

        gSelected # 0;
    end
end

```



```
// schon APs vorhanden?
```

```
FOR Erx # RecLink(403,400,13,_recFirst)
```

```
LOOP Erx # RecLink(403,400,13,_recNext)
```

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

```
  if (Auf.Z.Position=0) or (Auf.Z.Position=Auf.P.Position) then begin
```

```
    vHatAP # true;
```

```
    BREAK;
```

```
  end;
```

```
END;
```

```
// 10.03.2020 AH
```

```
if (vHatAP) then
```

```
  vHatAP # (Msg(401032,",_WinIcoQuestion, _WinDialogYesNo, 2)=_winIdNo);
```

```
Auf_Data:Verpackung2Auf(true, vHatAP=false);
```

```
RefreshIfm('edAuf.P.Erzeuger_Mat',y);
```

```
RefreshIfm('Text');
```

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

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

```
end;
```

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

```
$edAuf.P.KundenMatArtNr_Mat->Winfocusset(true);
```

```
gMDI->Winupdate(_WinUpdFld2Obj);
```

```
// ggf. Labels refreshen
```

```
// Refreshfm();
```

```
end;
```

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

```
// AusStueckliste
```

```
//
```

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

```
sub AusStueckliste()
```

```
local begin
```

```
    Erx      : int;
```

```
    vMenge   : float;
```

```
    vStk     : int;
```

```
    vGew     : float;
```

```
    vExist   : logic;
```

```
    vAlteMenge : float;
```

```
    vHdl     : int;
```

```
    vI       : int;
```

```
end
```

```
begin
```

```
    // Summe errechnen
```

```
    Erx # RecLink(409,401,15,_Recfirst);
```

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

```

Erx # RecLink(250,409,3,_recFirst); // Artikel holen

if (Erx>_rLocked) then RecBufClear(250);


// if (Auf.P.MEH.Wunsch=Art.MEH) then vMenge # vMenge + Auf.SL.Menge;

vMenge # vMenge + Lib_Einheiten:WandleMEH(409, "Auf.SL.Stückzahl", Auf.Sl.Gewicht, Auf.SL.Menge);


vStk   # vStk + "Auf.SL.Stückzahl";

vGew   # vGew + Auf.SL.Gewicht;

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

END;

```

```

Erx # RekLink(250,401,2,_RecFirst); // Positions-Artikel holen


// nur EK-Preis summieren...

// if (Art.Typ=c_art_cut) then begin+

if (1=1) then begin // 10.04.2015

    Erx # RecRead(401,1,_recTest);

    if (Erx=_rOK) then begin // Position existiert bereits?

        vAlteMenge # Auf.P.Menge;

        RecRead(401,1,_recLock);

        vExist # y;

    end;

    Auf_Data:SumEKGesamtPreis();

```

```

Auf_Data:SumAufpreise();

if (vExist) then begin
    Auf_Data:PosReplace(_recUnlock,'AUTO');
end;

if (Mode<>c_modeList) then begin
    $lb.P.Einzelpreis_Mat->wpcaption # ANum(Auf.P.Einzelpreis,2);
    $lb.Kalkuliert->wpcaption      # ANum(Auf.P.Kalkuliert,2);
    $lb.Poswert->wpcaption        # ANum(Auf.P.Gesamtpreis,2);
    $lb.Poswert_Mat->wpcaption    # ANum(Auf.P.Gesamtpreis,2);
    $lb.Rohgewinn->wpcaption      # ANum(Auf.P.Gesamtpreis - "Auf.P.GesamtwertEKW1",2);
    $lb.Aufpreise->wpcaption      # ANum(Auf.P.Aufpreis,2);
    $lb.Aufpreise_Mat->wpcaption  # ANum(Auf.P.Aufpreis,2);
end;

end;

// Mengensumme auch anderes?

if (vMenge<>0.0) and (Abs(vMenge-Auf.P.Menge.Wunsch)>0.01) then begin // 10.04.2015 and (Art.Type=
    // Summe übernehmen?

    if (Msg(409003,ANum(vMenge,-1)+' '+Auf.P.MEH.Wunsch,_WinIcoQuestion,_WinDialogYesNo,1)=_WinI
        _WinIcoQuestion) then begin
        Erx # RecRead(401,1,_recTest);

        if (Erx=_rOK) then begin // Position existiert bereits?

            vAlteMenge # Auf.P.Menge;

            PtD_Main:Memorize(401);
        end;
    end;
end;

```

```
RecRead(401,1,_recLock);
```

```
vExist # y;
```

```
end;
```

```
Auf.P.Menge.Wunsch # vMenge;
```

```
"Auf.P.Stückzahl" # vStk;
```

```
Auf.P.Gewicht # vGew;
```

```
if (Auf.P.MEH.einsatz=Auf.P.MEH.Wunsch) then
```

```
    Auf.P.Menge # vMenge
```

```
else if (Auf.P.MEH.Einsatz='kg') then
```

```
    Auf.P.Menge # vGew
```

```
else if (Auf.P.MEH.Einsatz='Stk') then
```

```
    Auf.P.Menge # cnvfi(vStk)
```

```
else if (Auf.P.MEH.Einsatz='t') then
```

```
    Auf.P.Menge # Rnd(vGew / 1000.0,0)
```

```
else
```

```
    Auf.P.Menge # Rnd(Lib_Einheiten:WandleMEH(401, "Auf.P.Stückzahl", Auf.P.Gewicht, Auf.P.Mer
```

```
Auf.P.Prd.Rest # Auf.P.Menge - Auf.P.Prd.LFS;
```

```
Auf.P.Prd.Rest.Stk # "Auf.P.Stückzahl" - Auf.P.Prd.LFS.Stk;
```

```
Auf.P.Prd.Rest.Gew # Auf.P.Gewicht - Auf.P.Prd.LFS.Gew;
```

```
Auf_Data:SumEKGesamtPreis();
```

```
Auf_Data:SumAufpreise(c_modeEdit);
```

```
Auf.P.Gesamtpreis # Auf_data:SumGesamtpreis(Auf.P.Menge, "Auf.P.Stückzahl" , Auf.P.Gewicht);
```

```
if (vExist) then begin
```

```
Auf_Data:PosReplace(_recUnlock,'AUTO');  
PtD_Main:Compare(401);  
  
// nötige Verbuchungen im Artikel druchführen...  
  
//Auf_data:VerbucheArt(Art.P.ArtikelNr, vAlteMenge, "Auf.P.Löschmarker"); 22.11.2021  
Auf_data:VerbucheArt(Art.P.ArtikelNr, vAlteMenge, "Auf.P.Löschmarker", Auf.P.Menge-vAlteMenge);  
end;
```

```
if (Mode<>c_modeList) then begin
```

```
    $edAuf.P.Menge.Wunsch->winupdate(_WinUpdFld2Obj);
```

```
    $edAuf.P.Menge->winupdate(_WinUpdFld2Obj);
```

```
    $edAuf.P.Menge_Mat->winupdate(_WinUpdFld2Obj);
```

```
    $edAuf.P.Stueckzahl_Mat->winupdate(_WinUpdFld2Obj);
```

```
    $edAuf.P.Gewicht_Mat->winupdate(_WinUpdFld2Obj);
```

```
    $lb.Auf.P.gewicht->winupdate(_WinUpdFld2Obj);
```

```
    $lb.Auf.P.Stueck->winupdate(_WinUpdFld2Obj);
```

```
    $lb.Auf.P.Prd.Rest->wpcaption # ANum(Auf.P.Prd.Rest,Set.Stellen.Menge);
```

```
    $RL.Aufpreise->winupdate(_winupdon,_WinLstFromFirst);
```

```
    $RL.Aufpreise_Mat->winupdate(_winupdon,_WinLstFromFirst);
```

```
    $lb.P.Einzelpreis_Mat->wpcaption      # ANum(Auf.P.Einzelpreis,2);
```

```
    $lb.Kalkuliert->wpcaption            # ANum(Auf.P.Kalkuliert,2);
```

```
    $lb.Poswert->wpcaption                # ANum(Auf.P.Gesamtpreis,2);
```

```
    $lb.Poswert_Mat->wpcaption            # ANum(Auf.P.Gesamtpreis,2);
```

```

$lb.Rohgewinn->wpcaption      # ANum(Auf.P.Gesamtpreis - "Auf.P.GesamtwertEKW1",2);

$lb.Aufpreise->wpcaption      # ANum(Auf.P.Aufpreis,2);

$lb.Aufpreise_Mat->wpcaption   # ANum(Auf.P.Aufpreis,2);

end;

end;

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

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

end;

// gesamtes Fenster aktivieren

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

Auf.P.Position # cnvia(StrCut(vHdl->wpcustom,1,5));

vl # CnvIA(StrCut(vHdl->wpcustom,6,10));

// springt in Feld zurück, dass bei verlassen den Focus besaß

if (vl<>0) then begin

    if (HdlInfo(vl,_HdlExists)>0) then vl->winfocusset(false);

end

else if (Mode=c_ModeView) then begin

    vl # gMDI->WinSearch('Edit');

    vl->wpdisabled # false;

    vl->WinFocusSet(false);

end;

end;

vHdl->wpcustom # "";

```

```

if (Mode=c_modeList) then RefreshList(gZLList, _WinLstRecFromRecid | _WinLstRecDoSelect);
end;

//=====

// AusKunde

//

//=====

sub AusKunde()
local begin

    vHdl : int;

end;

begin

    if (gSelected<>0) then begin

        RecRead(100,0,_RecId,gSelected);

        gSelected # 0;

        // Feldübernahme

        Auf.Kundennr # Adr.Kundennr;

        if(Mode = c_ModeNew) and (Adr.Sachbearbeiter<>") then

            Auf.Sachbearbeiter # Adr.Sachbearbeiter;

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

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

        end;

        // Focus auf Editfeld setzen:

        $edAuf.Kundennr->Winfocusset(false);

```



```

// ggf. Labels refreshen

RefreshIfm('edAuf.Kundennr',y);

RefreshIfm('edAuf.Sachbearbeiter');

end;


//=====

// AusLieferadresse

//

//=====

sub AusLieferadresse()

local begin

    Erx  : int;

    vHdl : int;

end;

begin

    if (gSelected<>0) then begin

        RecRead(100,0,_RecId,gSelected);

        // Feldübernahme

        Auf.Lieferadresse # Adr.Nummer;


        if(Set.Auf.LiefAnLeerYN = true) then

            Auf.Lieferanschrift # 0;

        else begin

            Auf.Lieferanschrift # 1;

            if (Adr.VK.Std.Lieferadr=Adr.Nummer) then begin

```

```
Erx # RecLink(101,100,76,_recFirst); // Lieferanschrift holen
```

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

```
    Auf.Lieferanschrift # Adr.A.Nummer;
```

```
end;
```

```
Erx # RecLink(101, 400, 2, _recFirst); // Lieferanschrift testen
```

```
if (Erx>_rLocked) then
```

```
    Auf.Lieferanschrift # 1;
```

```
end;
```

```
gSelected # 0;
```

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

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

```
    vHdl->Winupdate(_WinUpdFld2Obj);
```

```
end;
```

```
// Focus setzen:
```

```
$edAuf.Lieferadresse->Winfocusset(false);
```

```
// RefreshIfm('edAuf.Lieferadresse'); 10.12.2021 AH : etwas später
```

```
if (Auf.Lieferanschrift = 1) then begin
```

```
    Erx # RecLinkInfo(101,100,12,_recCount); // Mehr als eine Anschrift vorhanden?
```

```
    if (Erx > 1) then begin
```

```
        Auswahl('Lieferanschrift');
```

```
    end
```

```
else begin
```

```
    Erx # RecLink(101,100,12,_recFirst); // Wenn nur 1, diese holen
```

```

    if(Erx > _rLocked) then

        RecBufClear(101);

        Auf.Lieferanschrift # Adr.A.Nummer;

    end;

end;

RefreshIfm('edAuf.Lieferadresse',y);

$edAuf.Lieferanschrift->Winupdate(_WinUpdFld2Obj);

end;

```

```

//=====

```

```

// AusLieferanschrift

```

```

//

```

```

//=====

```

```

sub AusLieferanschrift()

```

```

local begin

```

```

    vHdl : int;

```

```

end;

```

```

begin

```

```

    if (gSelected<>0) then begin

```

```

        RecRead(101,0,_RecId,gSelected);

```

```

        // Feldübernahme

```

```

        Auf.Lieferanschrift # Adr.A.Nummer;

```

```

        gSelected # 0;

```

```

        vHdl # WinFocusget(); // LastFocus-Feld refreshen
    end;
end;

```

```

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

end;

// Focus setzen:

$edAuf.Lieferanschrift->Winfocusset(false);

RefreshIfm('edAuf.Lieferanschrift',y);

end;


//=====

// AusVerbraucher

//

//=====

sub AusVerbraucher()

local begin

    vHdl : int;

end;

begin

    if (gSelected<>0) then begin

        RecRead(100,0,_RecId,gSelected);

        // Feldübernahme

        Auf.Verbraucher # Adr.Nummer;

        gSelected # 0;

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

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

    end;

    // Focus auf Editfeld setzen:

```

```

$edAuf.Verbraucher->Winfocusset(false);

// ggf. Labels refreshen

RefreshIfm('edAuf.Verbraucher');

end;

//=====

// AusRechnungsempf

//

//=====

sub AusRechnungsempf()

local begin

    vHdl : int;

end;

begin

    if (gSelected<>0) then begin

        RecRead(100,0,_RecId,gSelected);

        gSelected # 0;

        // Feldübernahme

        Auf.RechnungsEmpf # Adr.KundenNr;

        Auf.RechnungsAnschr # 1;

        if (Set.Auf.STSvomKdYN=n) then

            "Auf.Steuerschlüssel" # "Adr.Steuerschlüssel";

        Auf.Zahlungsbed # Adr.VK.Zahlungsbed;

        "Auf.Währung" # "Adr.VK.Währung";

        if (Auf.Vorgangstyp=c_GUT) or (Auf.Vorgangstyp=c_Bel_LF) then begin

```

```

Auf.Zahlungsbed      # Adr.EK.ZAhlungsbed;

"Auf.Währung"        # "Adr.EK.Währung";

end;


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

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

end;

// Focus auf Editfeld setzen:

$edAuf.RechnungsEmpf->Winfocusset(false);


// ggf. Labels refreshen

RefreshIfm('edAuf.Rechnungsempf');

end;


//=====

// AusRechnungsanschr

//

//=====

sub AusRechnungsanschr()

local begin

    vHdl : int;

end;

begin

    if (gSelected<>0) then begin

        RecRead(101,0,_RecId,gSelected);

```

```

// Feldübernahme

Auf.Rechnungsanschr # Adr.A.Nummer;

gSelected # 0;

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

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

end;

// Focus setzen:

$edAuf.Rechnungsanschr->Winfocusset(false);

RefreshIfm('edAuf.Rechnungsanschr');

end;


//=====

// AusAnsprechpartner

//

//=====

sub AusAnsprechpartner()

local begin

    vHdl : int;

end;

begin

    if (gSelected<>0) then begin

        RecRead(102,0,_RecId,gSelected);

        // Feldübernahme

        if(StrLen('#'+cnvAl(Adr.P.Nummer)+'.'+StrAdj(Adr.P.Titel+' '+Adr.P.Vorname+' '+Adr.P.Name,_Strbegin),

```

```

    Auf.Best.Bearbeiter # '#' + cnvAl(Adr.P.Nummer) + ':' + StrAdj(Adr.P.Titel + ' ' + Adr.P.Vorname + ' ' + Adr.P.Nachname, _Strbegin);
else if (StrLen('#' + cnvAl(Adr.P.Nummer) + ':' + StrAdj(Adr.P.Vorname + ' ' + Adr.P.Name, _Strbegin)) < 32) then
    Auf.Best.Bearbeiter # '#' + cnvAl(Adr.P.Nummer) + ':' + StrAdj(Adr.P.Vorname + ' ' + Adr.P.Name, _Strbegin);
else if (StrLen('#' + cnvAl(Adr.P.Nummer) + ':' + StrAdj(Adr.P.Name, _Strbegin)) < 32) then
    Auf.Best.Bearbeiter # '#' + cnvAl(Adr.P.Nummer) + ':' + StrAdj(Adr.P.Name, _Strbegin);
else
    Auf.Best.Bearbeiter # 'Name zu lang';

gSelected # 0;

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

// Focus auf Editfeld setzen:
$edAuf.Best.Bearbeiter -> Winfocusset(false);

// ggf. Labels refreshen
RefreshIfm('edAuf.Best.Bearbeiter');
end;

//=====

// AusLand

//

//=====

sub AusLand()
local begin

```



```

vHdl : int;

end;

begin

if (gSelected<>0) then begin

    RecRead(812,0,_RecId,gSelected);

    // Feldübernahme

    Auf.Land # "Lnd.Kürzel";

    gSelected # 0;

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

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

end;

// Focus setzen:

$edAuf.Land->Winfocusset(false);

RefreshIfm('edAuf.Land');

end;

```

```

//=====

```

```

// AusBDS

```

```

//

```

```

//=====

```

```

sub AusBDS()

```

```

local begin

```

```

    vHdl : int;

```

```

end;

```

```

begin

```

```

if (gSelected<>0) then begin

    RecRead(836,0,_RecId,gSelected);

    // Feldübernahme

    Auf.BDSNummer # BDS.Nummer;

    gSelected # 0;

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

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

end;

// Focus setzen:

$edAuf.BDSNummer->Winfocusset(false);

// RefreshIfm('edAuf.BDSNummer');

end;


//=====

// AusWaehrung

//

//=====

sub AusWaehrung()

local begin

    vHdl : int;

end;

begin

    if (gSelected<>0) then begin

        RecRead(814,0,_RecId,gSelected);

        // Feldübernahme
    
```

```

"Auf.Währung" # Wae.Nummer;

gSelected # 0;

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

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

end;

// Focus auf Editfeld setzen:

$edAuf.Waehrung->Winfocusset(true);

// ggf. Labels refreshen

RefreshIfm('edAuf.Waehrung');

end;

```

```

//=====

```

```

// AusLieferbed

```

```

//

```

```

//=====

```

```

sub AusLieferbed()

```

```

local begin

```

```

    vHdl : int;

```

```

end;

```

```

begin

```

```

    if (gSelected<>0) then begin

```

```

        RecRead(815,0,_RecId,gSelected);

```

```

        // Feldübernahme

```

```

        Auf.Lieferbed # LiB.Nummer;

```

```

        gSelected # 0;

```

```

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

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

end;

// Focus auf Editfeld setzen:

$edAuf.Lieferbed->Winfocusset(false);

// ggf. Labels refreshen

RefreshIfm('edAuf.Lieferbed', true);

end;

```

```

//=====

```

```

// AusZahlungsbed

```

```

//

```

```

//=====

```

```

sub AusZahlungsbed()

```

```

local begin

```

```

    vHdl : int;

```

```

end;

```

```

begin

```

```

    if (gSelected<>0) then begin

```

```

        RecRead(816,0,_RecId,gSelected);

```

```

        // Feldübernahme

```

```

        Auf.Zahlungsbed # ZaB.Nummer;

```

```

        gSelected # 0;

```

```

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

```

```

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

```

```

end;

// Focus auf Editfeld setzen:

$edAuf.Zahlungsbed->Winfocusset(false);

// ggf. Labels refreshen

RefreshIfm('edAuf.Zahlungsbed');

end;


//=====

// AusSteuerschluessel

//

//=====

sub AusSteuerschluessel()

local begin

    vHdl : int;

end;

begin

    if (gSelected<>0) then begin

        RecRead(813,0,_RecId,gSelected);

        // Feldübernahme

        "Auf.Steuerschlüssel" # StS.Nummer;

        gSelected # 0;

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

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

    end;

    // Focus auf Editfeld setzen:

```

```

$edAuf.Steuerschluessel->Winfocusset(false);

// ggf. Labels refreshen

RefreshIfm('edAuf.steuerschluessel');

end;


//=====

// AusVersandart

//

//=====

sub AusVersandart()

local begin

    vHdl : int;

end;

begin

    if (gSelected<>0) then begin

        RecRead(817,0,_RecId,gSelected);

        // Feldübernahme

        Auf.Versandart # Vsa.Nummer;

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

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

    end;

    // Focus auf Editfeld setzen:

    $edAuf.Versandart->Winfocusset(false);

    // ggf. Labels refreshen

    if (gSelected<>0) then begin

```

```

gSelected # 0;

RefreshIfm('edAuf.Versandart',y);

end;

end;

//=====

// AusSachbearbeiter

//

//=====

sub AusSachbearbeiter()

local begin

    vHdl : int;

end;

begin

    if (gSelected<>0) then begin

        RecRead(800,0,_RecId,gSelected);

        // Feldübernahme

        Auf.Sachbearbeiter # Usr.Username;

        gSelected # 0;

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

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

    end;

    Usr_data:RecReadThisUser();

    // Focus auf Editfeld setzen:

    $edAuf.Sachbearbeiter->Winfocusset(false);

```

```
// ggf. Labels refreshen
```

```
RefreshIfm('edAuf.Sachbearbeiter');
```

```
end;
```

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

```
// AusVertreter1
```

```
//
```

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

```
sub AusVertreter1()
```

```
local begin
```

```
    vHdl : int;
```

```
end;
```

```
begin
```

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

```
        RecRead(110,0,_RecId,gSelected);
```

```
        // Feldübernahme
```

```
        Auf.Vertreter # Ver.Nummer;
```

```
        if (Ver.ProvisionProTJN) then begin
```

```
            Auf.Vertreter.ProT # Ver.ProvisionProz;
```

```
            Auf.Vertreter.Prov # 0.0;
```

```
        end
```

```
    else begin
```

```
        Auf.Vertreter.ProT # 0.0;
```

```
        Auf.Vertreter.Prov # Ver.ProvisionProz;
```

```
    end;
```



```

gSelected # 0;

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

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

end;

// Focus setzen:

$edAuf.Vertreter1->Winfocusset(false);

RefreshIfm('edAuf.Vertreter1');

end;

```

```

//=====

```

```

// AusVertreter2

```

```

//

```

```

//=====

```

```

sub AusVertreter2()

```

```

local begin

```

```

    vHdl : int;

```

```

end;

```

```

begin

```

```

    if (gSelected<>0) then begin

```

```

        RecRead(110,0,_RecId,gSelected);

```

```

        // Feldübernahme

```

```

        Auf.Vertreter2 # Ver.Nummer;

```

```

        if (Ver.ProvisionProTJN) then begin

```

```

            Auf.Vertreter2.ProT # Ver.ProvisionProz;

```

```

            Auf.Vertreter2.Prov # 0.0;

```

```

end

else begin

    Auf.Vertreter2.ProT # 0.0;

    Auf.Vertreter2.Prov # Ver.ProvisionProz;

end;

gSelected # 0;

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

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

end;

// Focus setzen:

$edAuf.Vertreter2->Winfocusset(false);

RefreshIfm('edAuf.Vertreter2');

end;

```

```

//=====

```

```

// AusAuftragsart

```

```

//

```

```

//=====

```

```

sub AusAuftragsArt()

```

```

local begin

```

```

    vHdl : int;

```

```

end;

```

```

begin

```

```

    if (gSelected<>0) then begin

```

```

        RecRead(835,0,_RecId,gSelected);

```

```

// Feldübernahme

Auf.P.Auftragsart # AAr.Nummer;

gSelected # 0;

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

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

end;

// Focus auf Editfeld setzen:

if ($NB.Page1->wpcustom='NB.Page1_Art') then begin

    $edAuf.P.Auftragsart->Winfocusset(false);

    RefreshIfm('edAuf.P.Auftragsart',y);

end

else begin

    $edAuf.P.Auftragsart_Mat->Winfocusset(false);

    RefreshIfm('edAuf.P.Auftragsart_Mat',y);

end;

end;

end;

//=====

// AusSkizze

//

//=====

sub AusSkizze();

local begin

    vTxtHdl : int;

    vDolt  : logic;

```

```

vHdl  : int;

end;

begin

if (gSelected<>0) then begin

    RecRead(829,0,_RecId,gSelected);

    Auf.P.Skizzennummer # Skz.Nummer;

    $Picture2->wpcaption # '*' + Skz.Dateiname;

    vDolt # y;

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

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

end;

gSelected # 0;

// Focus auf Editfeld setzen:

$edAuf.P.Skizzennummer->Winfocusset(false);

// ggf. Labels refreshen

if (vDolt) then Skizzendaten();

end;

```

```

//=====

```

```

// AusVorlageBAG

```

```

//

```

```

//=====

```

```

sub AusVorlageBAG()

```

```

local begin

```

```

    vTmp : int;

```

```

end;

begin
    if (gSelected<>0) then begin
        RecRead(700,0,_RecId,gSelected);

        gSelected # 0;

        // Feldübernahme

        Auf.P.VorlageBAG # BAG.Nummer;

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

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

    end;

    // Focus auf Editfeld setzen:

    $edAuf.P.VorlageBAG->Winfocusset(false);

end;

```

```

//=====

```

```

// AusEinsatzVPG

```

```

//

```

```

//=====

```

```

sub AusEinsatzVPG()

```

```

local begin

```

```

    vTmp    : int;

```

```

end;

```

```

begin

```

```

    if (gSelected<>0) then begin

```

```

        RecRead(100,0,_RecId,gSelected);

```

```

gSelected # 0;

// Feldübernahme

Auf.P.EinsatzVPG.Adr # Adr.Nummer;

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

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

end;

// Focus setzen:

$edAuf.P.EinsatzVPG.Adr->Winfocusset(false);

Auswahl('EinsatzVPG2');

end;

```

```

//=====

```

```

// AusEinsatzVPG2

```

```

//

```

```

//=====

```

```

sub AusEinsatzVPG2()

```

```

local begin

```

```

    vTmp    : int;

```

```

end;

```

```

begin

```

```

    if (gSelected<>0) then begin

```

```

        RecRead(105,0,_RecId,gSelected);

```

```

        gSelected # 0;

```

```

        // Feldübernahme

```

```

        Auf.P.EinsatzVPG.Nr # Adr.v.lfdNr;

```

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

```
// Focus setzen:  
$edAuf.P.EinsatzVPG.Nr->Winfocusset(false);  
end;
```

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

```
// AusVerpackung
```

```
//
```

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

```
sub AusVerpackung();
```

```
local begin
```

```
    vHdl : int;
```

```
end;
```

```
begin
```

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

```
    RecRead(105,0,_RecId,gSelected);
```

```
    Auf.P.Verpacknr      # Adr.V.lfdNr;
```

```
    Auf.P.VerpackAdrNr   # Adr.V.Adressnr;
```

```
    Auf.P.AbbindungL     # Adr.V.AbbindungL;
```

```
    Auf.P.AbbindungQ     # Adr.V.AbbindungQ;
```

```
    Auf.P.Zwischenlage   # Adr.V.Zwischenlage;
```

Auf.P.Unterlage # Adr.V.Unterlage;
Auf.P.Umverpackung # Adr.V.Umverpackung;
Auf.P.Wicklung # Adr.V.Wicklung;
Auf.P.MitLfEYN # Adr.V.MitLfEYN;
Auf.P.StehendYN # Adr.V.StehendYN;
Auf.P.LiegendYN # Adr.V.LiegendYN;
Auf.P.Nettoabzug # Adr.V.Nettoabzug;
"Auf.P.Stapelhöhe" # "Adr.V.Stapelhöhe";
Auf.P.StapelhAbzug # Adr.V.StapelhAbzug;
Auf.P.RingKgVon # Adr.V.RingKgVon;
Auf.P.RingKgBis # Adr.V.RingKgBis;
Auf.P.KgmmVon # Adr.V.KgmmVon;
Auf.P.KgmmBis # Adr.V.KgmmBis;
"Auf.P.StückProVE" # "Adr.V.StückProVE";
Auf.P.VEkgMax # Adr.V.VEkgMax;
Auf.P.RechtwinkMax # Adr.V.RechtwinkMax;
Auf.P.EbenheitMax # Adr.V.EbenheitMax;
"Auf.P.SäbeligkeitMax" # "Adr.V.SäbeligkeitMax";
"Auf.P.SäbelProM" # "Adr.V.SäbelProM";
Auf.P.Etikettentyp # Adr.V.Etikettentyp;
Auf.P.Verwiegungsart # Adr.V.Verwiegungsart;
Auf.P.VpgText1 # Adr.V.VpgText1;
Auf.P.VpgText2 # Adr.V.VpgText2;
Auf.P.VpgText3 # Adr.V.VpgText3;
Auf.P.VpgText4 # Adr.V.VpgText4;
Auf.P.VpgText5 # Adr.V.VpgText5;


```

Auf.P.VpgText6      # Adr.V.VpgText6;

Auf.P.Skizzennummer # Adr.V.Skizzennummer;

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

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

$NB.Page3->WinUpdate(_winUpdFld2Obj);

end;

gSelected # 0;

// Focus auf Editfeld setzen:

$edAuf.P.Verpacknr->Winfocusset(false);

// ggf. Labels refreshen

end;


//=====

// AusKopftext

//

//=====

sub AusKopftext();

local begin

    vTxtHdl : int;

end;

begin

    if (gSelected<>0) then begin

        RecRead(837,0,_RecId,gSelected);

        gSelected # 0;

        vTxtHdl # $Auf.P.TextEditKopf->wpdbTextBuf;

```

```

Lib_Texte:TxtLoadLangBuf('~837.'+CnvAl(Txt.Nummer,_FmtNumLeadZero | _FmtNumNoGroup,0,8), v7
RunAFX( 'Auf.P.AusText', 'Kopf|'+aint(vTxtHdl));

$Auf.P.TextEditKopf->WinUpdate(_WinUpdBuf2Obj);

end;

// Focus auf Editfeld setzen:

$Auf.P.TextEditKopf->Winfocusset(false);

// ggf. Labels refreshen

end;

```

```

//=====

// AusFusstext

//

//=====

```

```

sub AusFusstext();

local begin

    vTxtHdl : int;

end;

begin

    if (gSelected<>0) then begin

        RecRead(837,0,_RecId,gSelected);

        gSelected # 0;

        vTxtHdl # $Auf.P.TextEditFuss->wpdbTextBuf;

        Lib_Texte:TxtLoadLangBuf('~837.'+CnvAl(Txt.Nummer,_FmtNumLeadZero | _FmtNumNoGroup,0,8), v7
        RunAFX( 'Auf.P.AusText', 'Fuss|'+aint(vTxtHdl));

        $Auf.P.TextEditFuss->WinUpdate(_WinUpdBuf2Obj);
    end;
end;

```

```
end;
```

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

```
$Auf.P.TextEditFuss->Winfocusset(false);
```

```
// ggf. Labels refreshen
```

```
end;
```

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

```
// AusKopftextAdd
```

```
//
```

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

```
sub AusKopftextAdd();
```

```
local begin
```

```
  vTxtHdl  : int;
```

```
  vTxtHdl2 : int;
```

```
  vI       : int;
```

```
  vA       : alpha(250);
```

```
end;
```

```
begin
```

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

```
    RecRead(837,0,_RecId,gSelected);
```

```
    gSelected # 0;
```

```
    $Auf.P.TextEditKopf->WinUpdate(_WinUpdObj2Buf);
```

```
    vTxtHdl # $Auf.P.TextEditKopf->wpdbTextBuf;
```

```
    vTxtHdl2 # TextOpen(16);
```

```
    Lib_Texte:TxtLoadLangBuf('~837.'+CnvAI(Txt.Nummer,_FmtNumLeadZero | _FmtNumNoGroup,0,8), v
```

```

FOR vl # 1 loop inc(vl) WHILE (vl<=Textinfo(vTxtHdl2,_TextLines)) do begin

    TextLineWrite(vTxtHdl, TextInfo(vTxtHdl,_textLines)+1, TextLineRead(vTxtHdl2,vl,0), _TextLineInsert);

END;

TextClose(vTxtHdl2);

RunAFX( 'Auf.P.AusText', 'Kopf|'+aint(vTxtHdl));

$Auf.P.TextEditKopf->WinUpdate(_WinUpdBuf2Obj);

end;

// Focus auf Editfeld setzen:

$Auf.P.TextEditKopf->Winfocusset(false);

// ggf. Labels refreshen

end;

```

```

//=====

// AusFusstextAdd

//

//=====

```

```

sub AusFusstextAdd();

```

```

local begin

```

```

    vTxtHdl  : int;

```

```

    vTxtHdl2 : int;

```

```

    vl      : int;

```

```

    vA      : alpha(250);

```

```

end;

```

```

begin

```

```

    if (gSelected<>0) then begin

```

```

RecRead(837,0,_RecId,gSelected);

gSelected # 0;

$Auf.P.TextEditFuss->WinUpdate(_WinUpdObj2Buf);

vTxtHdl # $Auf.P.TextEditFuss->wpdbTextBuf;

vTxtHdl2 # TextOpen(16);

Lib_Texte:TxtLoadLangBuf('~837.'+CnvAI(Txt.Nummer,_FmtNumLeadZero | _FmtNumNoGroup,0,8), v7

FOR vl # 1 loop inc(vl) WHILE (vl<=Textinfo(vTxtHdl2,_TextLines)) do begin

    TextLineWrite(vTxtHdl, TextInfo(vTxtHdl,_textLines)+1, TextLineRead(vTxtHdl2,vl,0), _TextLineInsert),

END;

TextClose(vTxtHdl2);

RunAFX( 'Auf.P.AusText', 'Fuss|'+aint(vTxtHdl));

$Auf.P.TextEditFuss->WinUpdate(_WinUpdBuf2Obj);

end;

// Focus auf Editfeld setzen:

$Auf.P.TextEditFuss->Winfocusset(false);

// ggf. Labels refreshen

end;

//=====

// AusTextAdd

//

//=====

sub AusTextAdd();

local begin

    vTxtHdl : int;

```

```

vTxtHdl2 : int;

vI      : int;

vA      : alpha(250);

vName   : alpha;

end;

begin

```

```

if ( RunAFX( 'Auf.P.TextAdd', '' ) <> 0 ) then RETURN;

```

```

if (gSelected<>0) then begin

  RecRead(837,0,_RecId,gSelected);

  gSelected # 0;

```

```

$cb.Text1->wpcheckstate(_WinFlagNoFocusSet) # _WinStateChkUnchecked;

$cb.Text1->wpColBkg # _WinColParent;

$cb.Text2->wpcheckstate(_WinFlagNoFocusSet) # _WinStateChkUnchecked;

$cb.Text2->wpColBkg # _WinColParent;

$cb.Text3->wpcheckstate(_WinFlagNoFocusSet) # _WinStateChkchecked;

$cb.Text3->wpColBkg # _WinColParent;

```

```

if (Auf.P.TextNr1<>401) then begin

  vName # '~837.'+CnvAl(Auf.P.TextNr2,_FmtNumLeadZero | _FmtNumNoGroup,0,8);


  Auf.P.TextNr1 # 401;

  Auf.P.TextNr2 # 0;

```

```
$edAuf.P.TextNr2b->wpCaptionInt # 0;
```

```
vTxtHdl # $Auf.P.TextEditPos->wpdbTextBuf;
```

```
Lib_Texte:TxtLoadLangBuf(vName,vTxtHdl, Auf.Sprache);
```

```
if (Auf.P.Nummer=0) or (Auf.P.Nummer>1000000000) then
```

```
  vName # myTmpText+'.401.'+CnvAI(Auf.P.Position,_FmtNumLeadZero | _FmtNumNoGroup,0,3)
```

```
else
```

```
  vName # '~401.'+CnvAI(Auf.P.Nummer,_FmtNumLeadZero | _FmtNumNoGroup,0,8)+'.'+CnvAI(Auf.P.
```

```
RunAFX( 'Auf.P.AusText', 'Pos|'+aint(vTxtHdl));
```

```
$Auf.P.TextEditPos->wpcustom # vName;
```

```
$Auf.P.TextEditPos->WinUpdate(_WinUpdBuf2Obj);
```

```
RefreshIfm('Text');
```

```
end;
```

```
$Auf.P.TextEditPos->WinUpdate(_WinUpdObj2Buf);
```

```
vTxtHdl # $Auf.P.TextEditPos->wpdbTextBuf;
```

```
vTxtHdl2 # TextOpen(16);
```

```
Lib_Texte:TxtLoadLangBuf('~837.'+CnvAI(Txt.Nummer,_FmtNumLeadZero | _FmtNumNoGroup,0,8), v
```

```
FOR vl # 1 loop inc(vl) WHILE (vl<=Textinfo(vTxtHdl2,_TextLines)) do begin
```

```
  TextLineWrite(vTxtHdl, TextInfo(vTxtHdl,_textLines)+1, TextLineRead(vTxtHdl2,vl,0), _TextLineInsert);
```

```
END;
```

```
TextClose(vTxtHdl2);
```

```
RunAFX( 'Auf.P.AusText', 'Pos|'+aint(vTxtHdl));
```

```
$Auf.P.TextEditPos->WinUpdate(_WinUpdBuf2Obj);
```

```
end;
```

```

// Focus auf Editfeld setzen:

$Auf.P.TextEditPos->Winfocusset(false);

// ggf. Labels refreshen

end;


//=====

// AusWarengruppe

//

//=====

sub AusWarengruppe()

begin

if (gSelected<>0) then begin

    RecRead(819,0,_RecId,gSelected);

    gSelected # 0;

    // Feldübernahme

    Auf.P.Warengruppe # Wgr.Nummer;

    Auf_Data:SetWgrDateinr(Wgr.Dateinummer);

end;

// Focus auf Editfeld setzen:


$edAuf.P.Warengruppe->Winupdate(_WinUpdFld2Obj);

$edAuf.P.Warengruppe_Mat->Winupdate(_WinUpdFld2Obj);


Auf_P_SMain:SwitchMask(y);

if ($NB.Page1->wpcustom='NB.Page1_Art') then

```



```

    $edAuf.P.Warengruppe->Winfocusset(true)

else

    $edAuf.P.Warengruppe_Mat->Winfocusset(true);

end;


//=====

// AusProjekt

//

//=====

sub AusProjekt()

local begin

    vHdl : int;

end;

begin

    if (gSelected<>0) then begin

        RecRead(120,0,_RecId,gSelected);

        // Feldübernahme

        Auf.P.Projektnummer # Prj.Nummer;

        gSelected # 0;

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

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

    end;

    // Focus auf Editfeld setzen:

    if ($NB.Page1->wpcustom='NB.Page1_Art') then

        $edAuf.P.Projektnummer->Winfocusset(false);

```

```

else

    $edAuf.P.Projektnummer_Mat->Winfocusset(false);

end;

//=====

// AusIntrastat

//

//=====

sub AusIntrastat()

local begin

    vHdl : int;

end;

begin

    if (gSelected<>0) then begin

        RecRead(220,0,_RecId,gSelected);

        gSelected # 0;

        // Feldübernahme

        Auf.P.Intrastatnr # MSL.Intrastatnr;

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

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

    end;

    // Focus auf Editfeld setzen:

    if ($NB.Page1->wpcustom='NB.Page1_Art') then

        $edAuf.P.Intrastatnr->Winfocusset(false)

```

else

\$edAuf.P.Intrastatnr_Mat->Winfocusset(false);

end;

//=====

// AusProjektSL

//

//=====

sub AusProjektSL()

local begin

 vHdl : int;

end;

begin

 if (gSelected<>0) then begin

 RecRead(120,0,_RecId,gSelected);

 gSelected # 0;

 if (Auf_SL_Data.ProjektSLImport(Prj.Nummer)=y) then begin

 vHdl # gMdi->winsearch('NB.Erfassung');

 vHdl->wpdisabled # false;

 vHdl->wpvisible # true;

 vHdl # gMdi->winsearch('NB.Main');

 vHdl->wpCurrent # 'NB.Erfassung';

 Refreshfm();

```

    $ZL.Erfassung->WinUpdate(_WinUpdOn, _WinLstRecFromRecId | _WinLstRecDoSelect);

    Mode # c_ModeList2;

end;

end;

// Focus auf Editfeld setzen:


end;


//=====

// AusGuete

//

//=====

sub AusGuete()

local begin

    vHdl : int;

end;

begin

    if (gSelected<>0) then begin

        RecRead(832,0,_RecId,gSelected);

        // Feldübernahme

        if (MQu.ErsetzenDurch<>") then

            "Auf.P.Güte" # MQu.ErsetzenDurch

        else if ("MQu.Güte1"<>") then

            "Auf.P.Güte" # "MQu.Güte1"

        else

```

```

    "Auf.P.Güte" # "MQu.Güte2";

Auf.P.Werkstoffnr # MQU.Werkstoffnr;

gSelected # 0;

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

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

end;

// Focus auf Editfeld setzen:

// Focus auf Editfeld setzen:

if ($NB.Page1->wpcustom='NB.Page1_Art') then

    $edAuf.P.Guete->Winfocusset(true)

else

    $edAuf.P.Guete_Mat->Winfocusset(true);

end;


//=====

// AusGuetenstufe

//

//=====

sub AusGuetenstufe()

local begin

    vHdl : int;

end;

begin

    if (gSelected<>0) then begin

        RecRead(848,0,_RecId,gSelected);

```

```

gSelected # 0;

// Feldübernahme

"Auf.P.Gütenstufe" # MQu.S.Stufe;

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

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

end;

// Focus auf Editfeld setzen:

$edAuf.P.Guetenstufe_Mat->Winfocusset(false);

// ggf. Labels refreshen

end;


//=====

// AusAFOben

//

//=====

sub AusAFOben()

local begin

vA : alpha;

end;

begin

gSelected # 0;


vA # Obf_Data:BildeAFString(401,'1');

if (vA<>"Auf.P.AusfOben") then RunAFX('Obf.Changed','401|1');

Auf.P.AusfOben # vA;

```

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

```
$edAuf.P.AusfOben_Mat->Winfocusset(true);
```

```
end;
```

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

```
// AusAFUnten
```

```
//
```

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

```
sub AusAFUnten()
```

```
local begin
```

```
    vA : alpha;
```

```
end;
```

```
begin
```

```
    gSelected # 0;
```

```
    vA # Obf_Data:BildeAFString(401,'2');
```

```
    if (vA<>"Auf.P.AusfUnten") then RunAFX('Obf.Changed','401|2');
```

```
    Auf.P.AusfUnten # vA;
```

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

```
$edAuf.P.AusfUnten_Mat->Winfocusset(true);
```

```
end;
```

```

//=====

// AusZeugnis

//

//=====

sub AusZeugnis()

local begin

    vHdl : int;

end;

begin

    if (gSelected<>0) then begin

        RecRead(839,0,_RecId,gSelected);

        // Feldübernahme

        Auf.P.Zeugnisart # Zeu.Bezeichnung;

        gSelected # 0;

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

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

    end;

    // Focus auf Editfeld setzen:

    if ($NB.Page1->wpcustom='NB.Page1_Art') then

        $edAuf.P.Zeugnisart->Winfocusset(true)

    else

        $edAuf.P.Zeugnisart_Mat->Winfocusset(true);

    end;

end;

//=====

```



```
// AusErzeuger
```

```
//
```

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

```
sub AusErzeuger()
```

```
local begin
```

```
    vHdl : int;
```

```
end;
```

```
begin
```

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

```
        RecRead(100,0,_RecId,gSelected);
```

```
        // Feldübernahme
```

```
        Auf.P.Erzeuger # Adr.Nummer;
```

```
        gSelected # 0;
```

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

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

```
    end;
```

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

```
    $edAuf.P.Erzeuger_Mat->Winfocusset(false);
```

```
    // ggf. Labels refreshen
```

```
end;
```

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

```
// AusVerwiegungsart
```

```
//
```

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

```

sub AusVerwiegungsart()

local begin

    vHdl : int;

end;

begin

    if (gSelected<>0) then begin

        RecRead(818,0,_RecId,gSelected);

        // Feldübernahme

        Auf.P.Verwiegungsart # VwA.Nummer;

        gSelected # 0;

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

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

    end;

    // Focus auf Editfeld setzen:

    $edAuf.P.Verwiegungsart->Winfocusset(false);

    // ggf. Labels refreshen

    RefreshIfm('edAuf.P.Verwieungsart');

end;

```

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

```
// AusZwischenlage
```

```
//
```

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

```
sub AusZwischenlage()
```

```
local begin
```

```

vHdl : int;

end;

begin

if (gSelected<>0) then begin

    RecRead(838,0,_RecId,gSelected);

    // Feldübernahme

    Auf.P.Zwischenlage # ULa.Bezeichnung;

    gSelected # 0;

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

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

end;

// Focus auf Editfeld setzen:

$edAuf.P.Zwischenlage->Winfocusset(false);

// ggf. Labels refreshen

RefreshIfm('edAuf.P.Zwischenlage');

end;

```

```

//=====

```

```

// AusUnterlage

```

```

//

```

```

//=====

```

```

sub AusUnterlage()

```

```

local begin

```

```

    vHdl : int;

```

```

end;

```

begin

if (gSelected<>0) then begin

RecRead(838,0,_RecId,gSelected);

// Feldübernahme

Auf.P.Unterlage # ULa.Bezeichnung;

Auf.P.StapelhAbzug # "ULa.Höhenabzug";

gSelected # 0;

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

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

end;

// Focus auf Editfeld setzen:

\$edAuf.P.Unterlage->Winfocusset(false);

// ggf. Labels refreshen

RefreshIfm('edAuf.P.Unterlage');

end;

//=====

// AusUmverpackung

//

//=====

sub AusUmverpackung()

local begin

vHdl : int;

end;

```
begin
```

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

```
    RecRead(838,0,_RecId,gSelected);
```

```
    // Feldübernahme
```

```
    Auf.P.Umverpackung # ULa.Bezeichnung;
```

```
    gSelected # 0;
```

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

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

```
end;
```

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

```
$edAuf.P.Umverpackung->Winfocusset(false);
```

```
// ggf. Labels refreshen
```

```
RefreshIfm('edAuf.P.Umverpackung');
```

```
end;
```

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

```
// AusEtikettentyp
```

```
//
```

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

```
sub AusEtikettentyp()
```

```
local begin
```

```
    vHdl : int;
```

```
end;
```

```
begin
```

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

```

RecRead(840,0,_RecId,gSelected);

// Feldübernahme

Auf.P.Etikettentyp # Eti.Nummer;

gSelected # 0;

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

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

end;

// Focus auf Editfeld setzen:

$edAuf.P.Etikettentyp->Winfocusset(false);

// ggf. Labels refreshen

RefreshIfm('edAuf.P.Etikettentyp');

end;


//=====

// AusEtikettentyp2

//

//=====

sub AusEtikettentyp2()

local begin

    vDolt : logic;

    vHdl : int;

end;

begin

    if (gSelected<>0) then begin

        RecRead(840,0,_RecId,gSelected);

```

```

// Feldübernahme

Auf.P.Etikettentyp # Eti.Nummer;

gSelected # 0;

vDoIT # y;

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

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

end;

// Focus auf Editfeld setzen:

$edAuf.P.Etikettentyp2->Winfocusset(false);

// ggf. Labels refreshen

RefreshIfm('edAuf.P.Etikettentyp2');

if (Wgr_Data:IstArt(Auf.P.Wgr.Dateinr)) and (vDolt) then Etikettendaten();

end;


//=====

// AusKopfAufpreise

//

//=====

sub AusKopfaufpreise()

local begin

Erx : int;

vHdl : int;

vHdl2 : int;

vBuf401 : int;

end;

```

begin

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

Auf.P.Position # cnvia(StrCut(vHdl->wpcustom,1,5));

// if (mode=c_modeView) then Refreshfm();

vHdl2 # CnvIA(StrCut(vHdl->wpcustom,6,10));

if (vHdl2<>0) then begin

vHdl2->WinFocusSet(false);

end

else if (Mode=c_ModeView) then begin

vHdl2 # gMDI->WinSearch('Edit');

vHdl2->wpdisabled # false;

vHdl2->WinFocusSet(false);

end;

vHdl->wpcustom # ";

// ALLE Positionen refreshen?

// if (Auf.P.Nummer<1000000000) then begin

// if (Mode = c_ModeList) then

// Erx # RecRead(401,0,_RecID | _RecLock,gZLList->wpDbRecID);

APPOFF();

vBuf401 # RekSave(401);

FOR Erx # RecLink(401,400,9,_recFirst)

LOOP Erx # RecLink(401,400,9,_recNext)


```

WHILE (Erx<=_rLocked) do begin

  if (Mode=c_ModeList) then begin

    Auf_Data:SumAufpreise(c_ModeView);

  end

  else begin

    RecRead(401,1,_recLock);

    Auf_Data:SumAufpreise();

    Erx # RekReplace(401);

  end;

  if (vBuf401->Auf.P.Position=Auf.P.Position) then begin

    vBuf401->Auf.P.Aufpreis  # Auf.P.Aufpreis;

    vBuf401->Auf.P.Einzelpreis # Auf.P.Einzelpreis;

    vBuf401->Auf.P.Gesamtpreis # Auf.P.Gesamtpreis;

  end;

END;

RekRestore(vBuf401);

if (Mode=c_ModeEdit) then

  Erx # RecRead(401,1,_recLock | _recNoLoad);

if (Mode<>c_modeList) then begin

  $RL.Aufpreise->winupdate(_winupdon,_WinLstFromFirst);

  $RL.Aufpreise_Mat->winupdate(_winupdon,_WinLstFromFirst);

  $lb.P.Einzelpreis_Mat->wpcaption # ANum(Auf.P.Einzelpreis,2);

```

```

$lb.Kalkuliert->wpcaption      # ANum(Auf.P.Kalkuliert,2);

$lb.Poswert->wpcaption         # ANum(Auf.P.Gesamtpreis,2);

$lb.Poswert_Mat->wpcaption     # ANum(Auf.P.Gesamtpreis,2);

$lb.Rohgewinn->wpcaption       # ANum(Auf.P.Gesamtpreis - "Auf.P.GesamtwertEKW1",2);

$lb.Aufpreise->wpcaption       # ANum(Auf.P.Aufpreis,2);

$lb.Aufpreise_Mat->wpcaption   # ANum(Auf.P.Aufpreis,2);

end;


APPON();


if (Mode=c_ModeList) then
    RefreshList(gZLList, _WinLstRecFromRecID | _WinLstRecDoSelect);
// end;


end;


//=====

// AusAufpreise

//

//=====

sub AusAufpreise()

local begin

    vHdl : int;

    vHdl2 : int;

end;

```

begin

// gesamtes Fenster aktivieren

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

Auf.P.Position # CnvIA(StrCut(vHDL->wpcustom,1,5));

vHdl2 # CnvIA(StrCut(vHDL->wpcustom,6,10));

// springt in Feld zurück, dass bei verlassen den Focus besaá

if (vHdl2<>0) then begin

vHdl2->WinFocusSet(false);

end

else if (Mode=c_ModeView) then begin

vHdl2 # gMDI->WinSearch('Edit');

vHdl2->wpdisabled # false;

vHdl2->WinFocusSet(false);

end;

Auf_Data:SumAufpreise();

// Auf.P.Gesamtpreis # Auf_data:SumGesamtpreis(Auf.P.Menge, "Auf.P.Stückzahl" , Auf.P.Gewicht);

if (Mode<>c_modeList) then begin

\$RL.Aufpreise->winupdate(_winupdon,_WinLstFromFirst);

\$RL.Aufpreise_Mat->winupdate(_winupdon,_WinLstFromFirst);

\$lb.P.Einzelpreis_Mat->wpcaption # ANum(Auf.P.Einzelpreis,2);

\$lb.Kalkuliert->wpcaption # ANum(Auf.P.Kalkuliert,2);

\$lb.Poswert->wpcaption # ANum(Auf.P.Gesamtpreis,2);

\$lb.Poswert_Mat->wpcaption # ANum(Auf.P.Gesamtpreis,2);

```

$lb.Rohgewinn->wpcaption      # ANum(Auf.P.Gesamtpreis - "Auf.P.GesamtwertEKW1",2);

$lb.Aufpreise->wpcaption      # ANum(Auf.P.Aufpreis,2);

$lb.Aufpreise_Mat->wpcaption   # ANum(Auf.P.Aufpreis,2);

end;


vHDL->wpcustom # ";

end;


//=====

// AusKalkulation

//

//=====

sub AusKalkulation()

local begin

    vHDL : int;

    vHdl2 : int;

end;

begin

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

    vHdl2 # CnvIA(StrCut(vHDL->wpcustom,6,10));

    // springt in Feld zurück, dass bei verlassen den Focus besaá

    if (vHdl2<>0) then begin

        vHdl2->WinFocusSet(false);

    end

    else if (Mode=c_ModeView) then begin

```

```

vHdl2 # gMDI->WinSearch('Edit');

vHdl2->wpdisabled # false;

vHdl2->WinFocusSet(false);

end;

vHDL->wpcustom # ";

Auf_K_Data:SumKalkulation();

Auf_Data:SumEKGesamtPReis();

$lb.Rohgewinn->wpcaption # ANum(Auf.P.Gesamtpreis - "Auf.P.GesamtwertEKW1",2);

$lb.Kalkuliert->wpcaption # ANum(Auf.P.Kalkuliert,2);

$edAuf.P.Kalkuliert_Mat->winupdate(_WinUpdFld2Obj);

if (RecLinkInfo(405,401,7,_RecCount)=0) then begin

    $cb.Kalkulation_Art->wpCheckState # _WinStateChkUnChecked;

    $cb.Kalkulation_Mat->wpCheckState # _WinStateChkUnChecked;

end

else begin

    $cb.Kalkulation_Art->wpCheckState # _WinStateChkChecked;

    $cb.Kalkulation_Mat->wpCheckState # _WinStateChkChecked;

end;

end;

//=====

// AusAbruf

//

```

//=====

sub AusAbruf()

local begin

 vAuf : int;

 vPos : int;

 vTxtHdl : int;

 vName : alpha;

 vHdl2 : int;

end;

begin

 if (gSelected<>0) then begin

 RecRead(401,0,_RecId,gSelected);

 // Feldübernahme

 vAuf # Auf.P.Nummer;

 vPos # Auf.P.Position;

 gSelected # 0;

 end;

 if (mode<>c_ModeEdit) then begin

 RecBufCopy(ProtokollBuffer[400],400);

 RecBufDestroy(ProtokollBuffer[400]);

 ProtokollBuffer[400] # 0;

 RecBufCopy(ProtokollBuffer[401],401);

 RecBufDestroy(ProtokollBuffer[401]);

 ProtokollBuffer[401] # 0;

end;

if (vAuf<>0) then

Auf_P_SMain:CopyRahmen2Abruf(vAuf,vPos);

vHdl2 # WinFocusget(); // LastFocus-Feld refreshen

if (vHdl2<>0) then vHdl2->Winupdate(_WinUpdFld2Obj);

// Focus auf Editfeld setzen:

if (\$NB.Page1->wpcustom='NB.Page1_Art') then

\$edAuf.P.AbrufAufNr->Winfocusset(true)

else

\$edAuf.P.AbrufAufNr_Mat->Winfocusset(true);

// RefreshIfm('edAuf.P.AbrufAufNr',y);

RefreshIfm("");

gMDI->WinUpdate(_WinUpdFld2Obj);

end;

//=====

// AusErloes

//

//=====

sub AusErloes()

local begin

 vAuf : int;

 vPos : int;

 vHdl : int;

end;

begin

 if (gSelected<>0) then begin

 RecRead(450,0,_RecId,gSelected);

 // Feldübernahme

 Auf.P.AbrufAufNr # Erl.Rechnungsnr;

 Auf.P.AbrufAufPos # 0;

 gSelected # 0;

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

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

 end;

 // Focus auf Editfeld setzen:

 if (\$NB.Page1->wpcustom='NB.Page1_Art') then

 \$edAuf.P.AbrufAufNr->Winfocusset(true)

 else

 \$edAuf.P.AbrufAufNr_Mat->Winfocusset(true);

end;

//=====

// AusErloesKonto


```
//
```

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

```
sub AusErloesKonto()
```

```
local begin
```

```
    vAuf : int;
```

```
    vPos : int;
```

```
    vHdl : int;
```

```
end;
```

```
begin
```

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

```
        RecRead(451,0,_RecId,gSelected);
```

```
        // Feldübernahme
```

```
        Auf.P.AbrufAufNr      # Erl.K.Rechnungsnr;
```

```
        Auf.P.AbrufAufPos     # Erl.K.Rechnungspos;
```

```
        gSelected # 0;
```

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

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

```
    end;
```

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

```
if ($NB.Page1->wpcustom='NB.Page1_Art') then begin
```

```
    RefreshIfm('edAuf.P.AbrufAufPos',y);
```

```
    $edAuf.P.AbrufAufPos->Winfocusset(true);
```

```
end
```

```
else begin
```

```
    RefreshIfm('edAuf.P.AbrufAufPos_Mat',y);
```

```

    $edAuf.P.AbrufAufPos_Mat->Winfocusset(true);

end;

end;

//=====

// AusPreis

//

//=====

sub AusPreis()

local begin

    vHdl : int;

end;

begin

    if (gSelected<>0) then begin

        RecRead(254,0,_RecId,gSelected);

        gSelected # 0;

        if (Art.P.Preistyp<>'EK') then begin

            // Feldübernahme

            Auf.P.MEH.Preis # Art.P.MEH;

            Auf.P.PEH      # Art.P.PEH;

            Wae_Umrechnen(Art.P.Preis,"Art.P.Währung",var Auf.P.Grundpreis, "Auf.Währung");

        end;

        $edAuf.P.MEH.Preis->Winupdate(_WinUpdFld2Obj);

        $edAuf.P.PEH->Winupdate(_WinUpdFld2Obj);

        $edAuf.P.MEH.Preis_Mat->Winupdate(_WinUpdFld2Obj);

```

```

$edAuf.P.PEH_Mat->Winupdate(_WinUpdFld2Obj);

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

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

end;

// Focus auf Editfeld setzen:

if ($NB.Page1->wpcustom='NB.Page1_Art') then

    $edAuf.P.Grundpreis->Winfocusset(false)

else

    $edAuf.P.Grundpreis_Mat->Winfocusset(false);

// ggf. Labels refreshen

end;


//=====

// RefreshMode

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

//=====

sub RefreshMode(opt aNoRefresh : logic);

local begin

    Erx      : int;

    d_MenuItem : int;

    vHdl      : int;

    vKopfMode  : logic;

    vNurKopf   : logic;

    vTmp       : int;

end

```

begin

gMenu # gFrmMain->WinInfo(_WinMenu);

if (Auf.P.Nummer<>0) and (auf.P.Nummer<1000000000) and

(mode<>c_ModeList2) and (Mode<>c_ModeNew) and (mode<>c_ModeNew2) and (mode<>c_ModeEd

RecLink(400,401,3,_RecFirst);

end;

Erx # RekLink(835,401,5,_RecFirst); // Auftragsart holen

if (mode<>c_ModeList2) then

Auf_P_SMain:SwitchMask(\$NB.Main->wpcurrent='NB.Page1')

else if (\$ZL.Erfassung->wpdbrecid<>0) then

Auf_P_SMain:SwitchMask(\$NB.Main->wpcurrent='NB.Page1');

if (gFile=400) then RETURN;

vKopfMode # gZLList->wpDbLinkFileNo=401;

// Button & Menüs sperren

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

if (vHdl <> 0) then

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

```
d_MenuItem # gMenu->WinSearch('Mnu.KundenNr');
```

```
if (d_MenuItem != 0) then
```

```
    d_MenuItem->wpDisabled # ((Mode = c_ModeEdit) or (Mode = c_ModeNew) or (Rechte[Rgt_Auf_Change] = n));
```

```
d_MenuItem # gMenu->WinSearch('Mnu.KdArtNr');
```

```
if (d_MenuItem != 0) then
```

```
    d_MenuItem->wpDisabled # ((Mode = c_ModeEdit) or (Mode = c_ModeNew) or (Rechte[Rgt_Auf_P_Aenderung] = n));
```

```
d_MenuItem # gMenu->WinSearch('Mnu.Rechnungsempf');
```

```
if (d_MenuItem != 0) then
```

```
    d_MenuItem->wpDisabled # ((Mode = c_ModeEdit) or (Mode = c_ModeNew) or (Rechte[Rgt_Auf_Change] = n));
```

```
d_MenuItem # gMenu->WinSearch('Mnu.Liefervertrag');
```

```
if (d_MenuItem != 0) then
```

```
    d_MenuItem->wpDisabled # ((Mode = c_ModeEdit) or (Mode = c_ModeNew) or (Rechte[Rgt_Auf_P_Aenderung] = n));
```

```
// ---- AuftragsAnfrage ----
```

```
vHdl # gMenu->WinSearch('Mnu.Druck.Anf');
```

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

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

```
    or ((Auf.Vorgangstyp<>c_AUF) and (Auf.Vorgangstyp<>c_ANG)) or
```

```
    (w_Auswahlmode) or (Rechte[Rgt_Auf_Druck_AB]=n);
```

```
end;
```

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

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

```

vHdl->wpDisabled # (vKopfMode) or (vHdl->wpDisabled) or (w_AuswahlMode) or (Rechte[Rgt_Auf_P_A
vHdl # gMenu->WinSearch('Mnu.New2');

if (vHdl <> 0) then

    vHdl->wpDisabled # (vKopfMode) or (vHdl->wpDisabled) or (w_AuswahlMode) or (Rechte[Rgt_Auf_P_A

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

//  (vKopfMode) or (w_Auswahlmode) or (Rechte[Rgt_Auf_P_Anlegen]=n);


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

if (vHdl <> 0) then

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

        ("Auf.P.Löschmarker"='*');

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

if (vHdl <> 0) then

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

        ("Auf.P.Löschmarker"='*');


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

if (vHdl <> 0) then

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

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

if (vHdl <> 0) then

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


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

if (vHdl <> 0) then

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

```

```
((Mode<>c_ModeView) and (Mode<>c_ModeList));
```

```
vHdl # gMenu->WinSearch('Mnu.Druck.AB');
```

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

```
  vHdl->wpDisabled # ((Mode<>c_ModeView) and (Mode<>c_ModeLisT))
```

```
  or (Auf.Vorgangstyp<>c_AUF) or
```

```
  (w_Auswahlmode) or (Rechte[Rgt_Auf_Druck_AB]=n);
```

```
end;
```

```
vHdl # gMenu->WinSearch('Mnu.Druck.Angebot');
```

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

```
  vHdl->wpDisabled # ((Mode<>c_ModeView) and (Mode<>c_ModeLisT))
```

```
  or (Auf.Vorgangstyp<>c_ANG) or
```

```
  (w_Auswahlmode) or (Rechte[Rgt_Auf_Druck_Angebot]=n);
```

```
end;
```

```
vHdl # gMenu->WinSearch('Mnu.Druck.Avis');
```

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

```
  vHdl->wpDisabled # ((Mode<>c_ModeView) and (Mode<>c_ModeLisT))
```

```
  or (Auf.Vorgangstyp<>c_AUF) or
```

```
  (w_Auswahlmode) or (Rechte[Rgt_Auf_Druck_AB]=n);
```

```
end;
```

```
vHdl # gMenu->WinSearch('Mnu.Druck.Gut');
```

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

```
  vHdl->wpDisabled # ((Mode<>c_ModeView) and (Mode<>c_ModeLisT))
```

or ((Auf.Vorgangstyp<>c_BOGUT) and (Auf.Vorgangstyp<>c_REKOR) and (Auf.Vorgangstyp<>c_GUT))
(w_Auswahlmode) or (Rechte[Rgt_Auf_Druck_Gut]=n);

vHdl # gMenu->WinSearch('Mnu.Druck.Gut.VS');

if (vHdl <> 0) then

vHdl->wpDisabled # ((Mode<>c_ModeView) and (Mode<>c_ModeLisT))

or ((Auf.Vorgangstyp<>c_BOGUT) and (Auf.Vorgangstyp<>c_REKOR) and (Auf.Vorgangstyp<>c_GUT))
(w_Auswahlmode) or (Rechte[Rgt_Auf_Druck_Gut]=n);

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

if (vHdl <> 0) then

vHdl->wpDisabled # ((Mode<>c_ModeView) and (Mode<>c_ModeLisT))

or ((Auf.Vorgangstyp<>c_BEL_KD) and (Auf.Vorgangstyp<>c_BEL_LF)) or
(w_Auswahlmode) or (Rechte[Rgt_Auf_Druck_Bel]=n);

vHdl # gMenu->WinSearch('Mnu.Druck.Bel.VS');

if (vHdl <> 0) then

vHdl->wpDisabled # ((Mode<>c_ModeView) and (Mode<>c_ModeLisT))

or ((Auf.Vorgangstyp<>c_BEL_KD) and (Auf.Vorgangstyp<>c_BEL_LF)) or
(w_Auswahlmode) or (Rechte[Rgt_Auf_Druck_Bel]=n);

vHdl # gMenu->WinSearch('Mnu.Druck.RE.Proforma');

if (vHdl <> 0) then

vHdl->wpDisabled # (Auf.P.Aktionsmarker='\$') or

(Auf.Vorgangstyp<>c_AUF) or

(w_Auswahlmode) or (Rechte[Rgt_Auf_Druck_RE]=n);

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

if (vHdl <> 0) then

vHdl->wpDisabled # (Auf.P.Aktionsmarker<>'\$') or
(Auf.Vorgangstyp<>c_AUF) or
(w_Auswahlmode) or (Rechte[Rgt_Auf_Druck_RE]=n);

vHdl # gMenu->WinSearch('Mnu.Druck.RE.VS');

if (vHdl <> 0) then

vHdl->wpDisabled # (Auf.P.Aktionsmarker<>'\$') or
(Auf.Vorgangstyp<>c_AUF) or
(w_Auswahlmode) or (Rechte[Rgt_Auf_Druck_RE]=n);

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

if (vHdl <> 0) then begin

vHdl->wpDisabled # ((Mode<>c_ModeView) and (Mode<>c_ModeLisT))
or (Auf.Vorgangstyp<>c_AUF) or
(w_Auswahlmode) or (Rechte[Rgt_Auf_Druck_FM]=n);

end;

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

if (vHdl <> 0) then begin

vHdl->wpDisabled # (AAr.Berechnungsart<700) or (AAr.Berechnungsart>799) or ((Mode<>c_modeList)

end;

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

if (vHdl <> 0) then

vHdl->wpDisabled #

((Wgr_Data:IstArt(Auf.P.Wgr.Dateinr))) or

(Rechte[Rgt_Mat_Reservierung]=n) or

((Mode<>c_ModeView) and (Mode<>c_ModeList));

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

if (vHdl <> 0) then

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

((Mode<>c_ModeView) and (Mode<>c_ModeList));

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

if (vHdl <> 0) then

vHdl->wpDisabled # (Rechte[Rgt_Auf_Aktion]=n) or (Auf.Vorgangstyp=c_VorlageAuf) or

((Mode<>c_ModeView) and (Mode<>c_ModeList));

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

if (vHdl <> 0) then begin

vHdl->wpDisabled # ((Mode<>c_ModeNew2)) or (Auf.Vorgangstyp=c_VorlageAuf) or

(w_Auswahlmode) or (Rechte[Rgt_Auf_ProjektSL]=n);

end;

vHdl # gMenu->WinSearch('Mnu.Ang2Auf');

if (vHdl <> 0) then

vHdl->wpDisabled # (Rechte[Rgt_Auf_Ang2Auf]=n) or

(Auf.Vorgangstyp<>c_ANG) or

((Mode<>c_ModeView) and (Mode<>c_ModeList));

vHdl # gMenu->WinSearch('Mnu.Auf2And');

if (vHdl <> 0) then

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

(Auf.Vorgangstyp=c_Auf) or

((Mode<>c_ModeView) and (Mode<>c_ModeList));

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

if (vHdl <> 0) then

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

(Auf.Vorgangstyp<>c_ANG) or

((Mode<>c_ModeView) and (Mode<>c_ModeList));

vHdl # gMenu->WinSearch('Mnu.Auf2Verpackung');

if (vHdl <> 0) then

vHdl->wpDisabled # (Rechte[Rgt_Auf_Auf2Verpackung] = false) or

(Auf.Vorgangstyp<>c_Auf) or

((Mode<>c_ModeView) and (Mode<>c_ModeList));

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

if (vHdl <> 0) then

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

(Auf.Vorgangstyp<>c_AUF) or

(Auf.LiefervertragYN = true) or

((Mode<>c_ModeView) and (Mode<>c_ModeList));

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

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

```
  vHdl->wpDisabled # (Rechte[Rgt_Auf_MATZ]=n) or
```

```
    //(Auf.Vorgangstyp<>c_AUF) or 02.11.2016 GuBe
```

```
    ((Auf.Vorgangstyp != c_GUT) and (Auf.Vorgangstyp != c_Bel_LF) and (Auf.Vorgangstyp != c_
```

```
    (Auf.LiefervertragYN = true) or
```

```
    ((Mode<>c_ModeView) and (Mode<>c_ModeList) and (Mode<>c_ModeNew2)));
```

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

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

```
  vHdl->wpDisabled # (Rechte[Rgt_Auf_Mat_DFakt]=n) or
```

```
    (Auf.PAbrufYN) or
```

```
    (Auf.Vorgangstyp<>c_AUF) or
```

```
    (Auf.LiefervertragYN = true) or
```

```
    ((Mode<>c_ModeView) and (Mode<>c_ModeList)));
```

```
vHdl # gMenu->WinSearch('Mnu.DFakt.Gut');
```

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

```
  vHdl->wpDisabled # (Rechte[Rgt_Auf_Gut_DFakt]=n) or
```

```
    ((Auf.Vorgangstyp<>c_BOGUT) and (Auf.Vorgangstyp<>c_REKOR) and (Auf.Vorgangstyp<>
```

```
    ((Mode<>c_ModeView) and (Mode<>c_ModeList)));
```

```
vHdl # gMenu->WinSearch('Mnu.DFakt.Bel');
```

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

```
  vHdl->wpDisabled # (Rechte[Rgt_Auf_Bel_DFakt]=n) or
```

((Auf.Vorgangstyp<>c_BEL_KD) and (Auf.Vorgangstyp<>c_BEL_LF)) or
((Mode<>c_ModeView) and (Mode<>c_ModeList));

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

if (vHdl <> 0) then

vHdl->wpDisabled # (Rechte[Rgt_Auf_Fahrauftrag]=n) or
(Auf.PAbrufYN) or
(Auf.Vorgangstyp<>c_AUF) or
(Auf.LiefervertragYN = true) or
((Mode<>c_ModeView) and (Mode<>c_ModeList));

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

if (vHdl <> 0) then

vHdl->wpDisabled # (Rechte[Rgt_Auf_Liefersch]=n) or
(Auf.PAbrufYN) or
(Auf.Vorgangstyp<>c_AUF) or
(Auf.LiefervertragYN = true) or
((Mode<>c_ModeView) and (Mode<>c_ModeList));

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

if (vHdl <> 0) then

vHdl->wpDisabled # (Rechte[Rgt_Auf_Versandpool]=n) or
(Auf.Vorgangstyp<>c_AUF) or
(Set.LFS.mitVersandYN=false) or
(Auf.LiefervertragYN = true) or
((Mode<>c_ModeView) and (Mode<>c_ModeList));

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

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

```
  vHdl->wpDisabled # (mode=c_modeEdit) or (mode=c_modeNew) or  
    (Rechte[Rgt_Auf_Stueckliste]=n) or  
    ((Wgr_Data:IstMat(Auf.P.Wgr.Dateinr)));
```

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

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

```
  vHdl->wpDisabled # ((mode<>c_modeView) and (mode<>c_modeList)) or  
    (Wgr_Data:IstMat(Auf.P.Wgr.Dateinr));
```

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

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

```
  vHdl->wpDisabled # (Auf.Vorgangstyp<>c_AUF) or ((Mode<>c_modeList) and (Mode<>c_modeView));  
  //(AAr.Berechnungsart<700) or (AAr.Berechnungsart>799) or ((Mode<>c_modeList) and (Mode<>c_modeView));
```

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

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

```
  vHdl->wpDisabled # (AAr.Berechnungsart<700) or (AAr.Berechnungsart>799) or ((Mode<>c_modeList) and (Mode<>c_modeView));
```

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

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

```
  vHdl->wpDisabled # (AAr.Berechnungsart<700) or (AAr.Berechnungsart>799) or ((Mode<>c_modeList) and (Mode<>c_modeView));
```

```

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

if (vHdl <> 0) then

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

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

if (vHdl <> 0) then

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

vHdl # gMdi->WinSearch('bt.Aufpreise');

if (vHdl <> 0) then

    vHdl->wpDisabled # (Rechte[Rgt_Auf_Aufpreise]=n) or ((Mode<>c_ModeNew) and (Mode<>c_ModeNe

vHdl # gMdi->WinSearch('bt.Aufpreise_Mat');

if (vHdl <> 0) then

    vHdl->wpDisabled # (Rechte[Rgt_Auf_Aufpreise]=n) or ((Mode<>c_ModeNew) and (Mode<>c_ModeNe

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

if (vHdl <> 0) then

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

vHdl # gMdi->WinSearch('bt.Kalkulation');

if (vHdl <> 0) then

    vHdl->wpDisabled # (Rechte[Rgt_Auf_Kalkulation]=n) or ((Mode<>c_ModeNew) and (Mode<>c_ModeNe

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

if (vHdl <> 0) then

    vHdl->wpdisabled # (Rechte[Rgt_Auf_P_Change_Artikel]=n) or

// ST 2014-02-12: Artikelnummer auch "einfügbar" machen Prj. 1304/237

// (Auf.p.ArtikelNr=") or

    ((Mode<>c_ModeList) and (Mode<>c_ModeView));

```

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

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

```
    vHdl->wpDisabled # (Rechte[Rgt_Auf_P_Anlegen]=n) or  
        ((Mode<>c_ModeView) and (Mode<>c_ModeList));
```

```
vTmp # gMdi->Winsearch('NB.Main');
```

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

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

```
    vHdl->wpDisabled # (Rechte[Rgt_Auf_Korr_Termin]=n) or  
        (Auf.Vorgangstyp<>c_AUF) or  
        (  
            (Mode<>c_ModeList2));
```

```
// vHdl # gMenu->WinSearch('Mnu.Text.Add');
```

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

```
//    vHdl->wpDisabled # (Mode<>c_ModeEdit) and (Mode<>c_ModeNew) and (Mode<>c_ModeNew2);
```

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

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

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

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

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

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

```
vHdl # gMenu->WinSearch('Mnu.Protokoll.Kopf');
```

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


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

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

if (vHdl <> 0) then

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

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

if (vHdl <> 0) then

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

if (Mode=c_ModeList2) then begin

vNurKopf # RecLinkInfo(401,400,9,_recCount)=0;

if (vNurKopf) then begin

\$NB.Page1->wpdisabled # y;

\$NB.Page2->wpdisabled # y;

\$NB.Page3->wpdisabled # y;

\$NB.Page4->wpdisabled # y;

\$NB.Page5->wpdisabled # y;

end;

end

else if (Mode<>c_ModeNew) then begin

\$NB.Page1->wpdisabled # n;

\$NB.Page2->wpdisabled # n;

\$NB.Page3->wpdisabled # n;

\$NB.Page4->wpdisabled # n;

```

$NB.Page5->wpdisabled # n;

end;

$NB.KopfText->wpdisabled # (Mode=c_ModeNew) or (vNurKopf);

$NB.Fusstext->wpdisabled # (Mode=c_ModeNew) or (vNurKopf);


// Position anhängen?

if (w_AppendNr<>0) then $NB.Kopf->wpdisabled # y;


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


RETURN;

end;


//=====

// EvtMenuCommand

//      Menüpunkt aufgerufen

//=====

sub EvtMenuCommand (

    aEvt          : event;    // Ereignis

    aMenuItem     : int      // Menüeintrag

) : logic

local begin

    Erx    : int;

    vHdl   : handle;

```

```

vNr    : int;

vKLim  : float;

vNumNeu : int;

vStk   : int;

vGew   : float;

vQ     : alpha(4000);

vBuf401 : int;

vTmp   : int;

vM     : float;

end;

begin

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

    case (aMenuItem->wpName) of

        'Mnu.AlsKurzinfo' :

            Lib_Workbench:OpenKurzInfo(401, RecInfo(401,_recID));

        // temporär aktiviert für Umstellung Reverse Charge /2014-08-25 TM

        'Mnu.Mark.SetField' : begin

            Lib_Mark:SetField(gFile);

        end;

        'Mnu.Filter.Start' : begin

```

```
Auf_P_Mark_Sel('401.xml');
```

```
RETURN true;
```

```
end;
```

```
'Mnu.Filter.AufNr' : begin
```

```
  Auf_P_subs:SelAufNr();
```

```
  RETURN true;
```

```
end;
```

```
'Mnu.Filter.User' : begin  // 2023-01-17 AH
```

```
  Auf_P_subs:SelUser();
```

```
  RETURN true;
```

```
end;
```

```
'Mnu.Filter.Abruf' : begin
```

```
  Auf_P_subs:SelAbruf();
```

```
  RETURN true;
```

```
end;
```

```
'Mnu.Aktivitaeten' : begin
```

```
  TeM_Sub:Start(401);
```

```
end;
```

'Mnu.Mark.Sel' : begin

 Auf_P_Mark_Sel();

end;

'Mnu.Neu.AusAuftrag' : begin

 Auf_Subst:CopyAuftragAuswahl();

end;

'Mnu.Neu.AusAuftragAblage' : begin

 Auf_Subst:CopyAuftragAuswahl('ABLAGE');

end;

'Mnu.Neu.AusAufNr' : begin

 Auf_Subst:CopyAuftrag(0,0);

end;

'Mnu.Neu.AusBestellung' : begin

 Auf_Subst:CopyBestellungAuswahl();

end;

```
'Mnu.Neu.AusBestellnr' : begin
```

```
  Auf_Subst:CopyBestellung(0,0);
```

```
end;
```

```
'Mnu.Ktx.Errechnen' : begin
```

```
  if (aEvt:Obj->wpname='edAuf.P.Grundpreis') then begin
```

```
    RecLink(100,401,4,0);      // Kunde holen
```

```
    if (Art_P_Data:FindePreis('VK', Adr.Nummer, Auf.P.Menge, Auf.P.MEH.Preis, 1)) then begin
```

```
      Auf.P.Grundpreis  # Art.P.PreisW1;
```

```
      $edAuf.P.GrundPreis->Winupdate(_WinUpdFld2Obj);
```

```
    end;
```

```
  end;
```

```
  if (aEvt:Obj->wpname='edAuf.P.Grundpreis_Mat') and (Wgr_Data:IstMix(Auf.P.Wgr.Dateinr)) then begin
```

```
    RecLink(100,401,4,0);      // Kunde holen
```

```
    if (Art_P_Data:FindePreis('VK', Adr.Nummer, Auf.P.Menge, Auf.P.MEH.Preis, 1)) then begin
```

```
      Auf.P.Grundpreis  # Art.P.PreisW1;
```

```
      $edAuf.P.Grundpreis_Mat->Winupdate(_WinUpdFld2Obj);
```

```
    end;
```

```
  end;
```

```
  if (aEvt:Obj->wpname='edAuf.P.Gewicht_Mat') then begin
```

```
    Auf.P.Gewicht # Auf_P_Subst:CalcGewicht(); // 22.06.2022 AH
```

```
    $edAuf.P.Gewicht_Mat->winupdate(_WinUpdFld2Obj);
```

```
  end;
```

```
  if (aEvt:Obj->wpname='edAuf.P.Stueckzahl_Mat') then begin
```

```

"Auf.P.Stückzahl" # Lib_Berechnungen:STK_aus_KgDBLWgrArt(Auf.P.Gewicht, Auf.P.Dicke, Auf.P.B
$edAuf.P.Stueckzahl_Mat->winupdate(_WinUpdFld2Obj);

end;

if (aEvt:Obj->wpname='edAuf.P.Menge_Mat') then begin

    Auf.P.Menge # Lib_Einheiten:WandleMEH(401, "Auf.P.Stückzahl", Auf.P.Gewicht, 0.0, ", Auf.P.MEH.

    $edAuf.P.Menge_Mat->winupdate(_WinUpdFld2Obj);

end;

if (aEvt:Obj->wpname='edAuf.P.Dickentol_Mat') then MTo_Data:BildeVorgabe(401,'Dicke');

if (aEvt:Obj->wpname='edAuf.P.Breitentol_Mat') then MTo_Data:BildeVorgabe(401,'Breite');

if (aEvt:Obj->wpname='edAuf.P.Laengentol_Mat') then MTo_Data:BildeVorgabe(401,'Länge');

if (aEvt:Obj->wpname='edAuf.P.TextNr2b') then begin

    RecLink(100,400,1,_RecFirst); // Kunde holen

    Auf.P.TextNr2 # Txt_Data:Automatisch(401, "Auf.P.Gütenstufe", "Auf.P.Güte", Auf.P.Warengruppe, A

    $edAuf.P.TextNr2b->wpcaptionint # Auf.P.TextNr2;

    vHDL # $Auf.P.TextEditPos->wpdbTextBuf;

    Lib_Texte:TxtLoadLangBuf('~837.'+CnvAI(Auf.P.TextNr2,_FmtNumLeadZero | _FmtNumNoGroup,0,8

    RunAFX( 'Auf.P.AusText', 'Pos|'+aint(vHdl));

    $Auf.P.TextEditPos->WinUpdate(_WinUpdBuf2Obj);

end;

end;

```

```

'Mnu.New2' : begin

```

```

if (StrFind(gMDI->wpname,'Auf.P.Verwaltung',0,_StrCaseIgnore)<>0) or (Mode=c_ModeList2) then beg

    w_AppendNr # 0;

    App_Main:Action(c_ModeNew);

```

```

end

else begin          // neue Positionen?

    w_AppendNr # Auf.P.Nummer;

    App_Main:Action(c_ModeNew);

end;

end;

'Mnu.Restore' : begin

    Auf_Abl_Data:RestoreAusAblage();

    RecRead(401,1,0);

    RefreshList(gZLList, _WinLstRecFromRecID | _WinLstRecDoSelect);

end;


'Mnu.ChangePosNr' : begin

    Auf_P_Sub:ChangePosNr();

    if (ErrList <> 0) then begin

        ErrorOutput;

        RETURN false;

    end;

    RecRead(401,1,0);

    RefreshList(gZLList, _WinLstRecFromRecID | _WinLstRecDoSelect);

end;

```


'Mnu.KundenNr' : begin

 Auf_Sub:ChangeKundenNr();

end;

'Mnu.KdArtNr' : begin

 Auf_Sub:ChangeKundenArtNr();

end;

'Mnu.Rechnungsempf' : begin

 Auf_Sub:ChangeRechnungsempf();

end;

'Mnu.Liefervertrag' : begin // [13.04.2010/PW]

 if (Auf.LiefervertragYN) or (Auf.AbrufYN) then

 RETURN true;

 // Aktionen loopen...

 FOR Erx # RecLink(404,400,15,_recFirst)

 LOOP Erx # RecLink(404,400,15,_recNext)

 WHILE (Erx<=_rLocked) do begin

 if (Auf.A.Aktionstyp=c_Akt_Angebot) then CYCLE;

```
Msg( 400021, ", 0, 0, 0 );
```

```
RETURN false;
```

```
END;
```

```
if (msg(400101,",_WinIcoQuestion,_WinDialogYesNo,2)=_WinIdYes) then begin
```

```
  //if ( RecLinkInfo( 404, 400, 15, _recCount ) > 0 ) then begin
```

```
    // Msg( 400021, ", 0, 0, 0 );
```

```
    RecRead( 400, 1, _recLock );
```

```
    Auf.LiefervertragYN # true;
```

```
    Auf.AbrufYN  # false;
```

```
    Auf.PAbrufYN # false;
```

```
    if ( RekReplace( 400, _recUnlock,'AUTO') != _rOk ) then
```

```
      Msg( 999999, 'Änderungen können nicht vorgenommen werden.', 0, 0, 0 );
```

```
    else
```

```
      Msg( 999998, ", 0, 0, 0 );
```

```
    end;
```

```
end;
```

```
'Mnu.Protokoll' : begin
```

```
  PtD_Main:View(401,Auf.P.Anlage.Datum, Auf.P.Anlage.Zeit, Auf.P.Anlage.User, "Auf.P.Lösch.Datum",
```

```
end;
```

```
'Mnu.Protokoll.Kopf' : begin
```

```
  if (Auf.Freigabe.Datum<>0.0.0) then
```

```
    PtD_Main:View(400,Auf.Anlage.Datum, Auf.Anlage.Zeit, Auf.Anlage.User,0.0.0,0:0,"", Translate('Frei'))
```

```
  else
```

```
    PtD_Main:View(400,Auf.Anlage.Datum, Auf.Anlage.Zeit, Auf.Anlage.User);
```

```
end;
```

```
'Mnu.Artikeldispo' : begin
```

```
  Erx # RecLink(250,401,2,_RecFirst); // Artikel holen
```

```
  // Sonderfunktion:
```

```
  if (RunAFX('Art.Dispoliste','250_401_-RES_409_501_701')<>0) then begin
```

```
    RETURN true;
```

```
  end;
```

```
  gMDI # Lib_GuiCom:AddChildWindow(gMDI,'Art.Dispo.Verwaltung','');
```

```
  Art_Disposition2:Show('Dispoliste','250_401_-RES_409_501_701',y,n, gMDI);
```

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

```
end;
```

```
'Mnu.Artikel' : begin
```

```
  // ST 2014-02-12: Artikelnummer auch "einfügbar" machen Prj. 1304/237
```

```
  //if (Auf.P.Artikelnr="") then RETURN true;
```

```
  if (Rechte[Rgt_Auf_P_Change_Artikel]) then
```

```
    Auf_Subs:ChangeArtikel();
```

```
end;
```

```
'Mnu.TerminCopy' : begin
```

```
  Auf_Data:TerminCopy();
```

```
  $ZL.Erfassung->WinUpdate(_WinUpdOn, _WinLstRecFromRecID | _WinLstRecDoSelect);
```

```
end;
```

```
'Mnu.Append' : begin
```

```
  if (Auf.P.Nummer<>0) then begin
```

```
    w_AppendNr # Auf.P.Nummer;
```

```
    App_Main:Action(c_ModeNew);
```

```
  end;
```

```
end;
```

```
'Mnu.DMS' : begin
```

```
  RecLink(100,400,1,_RecFirst); // Kunde holen
```

```
  if (Auf.Vorgangstyp=c_ANG) then
```

```
    DMS_ArcFlow:ShowAbm('ANG',Auf.Nummer, Adr.Nummer)
```

```
  else
```

```
    DMS_ArcFlow:ShowAbm('AUF',Auf.Nummer, Adr.Nummer);
```

```
end;
```

```
'Mnu.CUS.Felder' : begin
```

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

```
end;
```

```
'Mnu.ProjektSL' : begin
```

```
  Auswahl('ProjektSL');
```

```
end;
```

```
'Mnu.Einsatzmaterial' : begin
```

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

```
  if (Erx>_rlocked) then RecBufClear(819);
```

```
  if (AAr.Berechnungsart>=700) and (AAr.Berechnungsart<=799) then
```

```
    BA1_Lohn_Subst:VerwalteEinsatz(Auf.P.Nummer, Auf.P.Position);
```

```
end;
```

```
'Mnu.Fertigungen' : begin
```

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

```
  if (Erx>_rlocked) then
```

```
    RecBufClear(819);
```

```
  if (AAr.Berechnungsart >= 700) and (AAr.Berechnungsart <= 799) then
```

```
    BA1_Lohn_Subst:VerwalteFertigung(Auf.P.Nummer, Auf.P.Position);
```

```
end;
```

```
'Mnu.Betriebsauftrag' : begin
```

```
if (Auf.Vorgangstyp<>c_AUF) then RETURN false;
```

```
Exr # RecLink(835,401,5,_RecFirst);    // Auftragsart holen
```

```
if (Exr>_rlocked) then RecBufClear(819);
```

```
if (Auf.P.VorlageBAG<>0) then
```

```
    Auf_Subs:BAG2Auf()
```

```
else if (AAr.Berechnungsart>=700) and (AAr.Berechnungsart<=799) then
```

```
    BA1_Lohn_Subs:VerwalteBetriebsauftrag(Auf.P.Nummer, Auf.P.Position)
```

```
else
```

```
    Auf_Subs:BAG2Auf();
```

```
end;
```

```
'Mnu.Versandpool' : begin
```

```
    if (Set.LFS.mitVersandYN) then Auf_P_Subs:Versand();
```

```
end;
```

```
'Mnu.Lieferschein' : begin
```

```
// 2022-09-20 AH   Proj. 2346/19   if (gMdiLfs<>0) then RETURN true;
```

```
// Kreditlimit prüfen...
```

```
if ("Set.KLP.LFS-Druck"<>") then
```

```
    if (Adr_K_Data:Kreditlimit(Auf.Rechnungsempf,"Set.KLP.LFS-Druck",n, var vKLim)=false) then RETU
```

```
RecBufClear(440);
```

Lfs.Nummer # myTmpNummer;

Lfs.Anlage.Datum # today;

Lfs.Kundennummer # Auf.P.Kundennr;

Lfs.Kundenstichwort # Auf.P.KundenSW;

Lfs.Zieladresse # Auf.Lieferadresse;

Lfs.Zielanschrift # Auf.Lieferanschrift;

if (Set.LFS.SpediLeerYN=false) then begin

 Lfs.SpediteurNr # Set.eigeneAdressnr;

 RecLink(100,440,6,_recFirst);

 Lfs.Spediteur # Adr.Stichwort;

end;

Lfs.Kosten.PEH # 1000;

Lfs.Kosten.MEH # 'kg';

Lfs.Lieferdatum # today;

RecLink(400,401,3,_recFirst); // Kopf holen

// MATERIAL -----

if (Wgr_Data:IstMat(Auf.P.Wgr.Dateinr)) or (Wgr_Data:IstMix(Auf.P.Wgr.Dateinr)) then begin

 // tmp. Lieferschein aufbauen

 Auf_Subs:Lieferschein(n,440);

end

else

 // ARTIKEL -----

 if (Wgr_Data:IstArt(Auf.P.Wgr.Dateinr)) then begin

```
// gibts schon offene VLDAW ??
```

```
if (Auf_Data:UpdateVLDAW())=true) then RETURN true;
```

```
// sonst neuen LFS generieren:
```

```
// tmp. Lieferschein aufbauen
```

```
Auf_Subst:Lieferschein(n,440);
```

```
if (RecLinkInfo(441,440,4,_recCount)=0) then begin
```

```
    Msg(440105,"",0,0,0);
```

```
    RETURN true;
```

```
end;
```

```
end; // Artikel
```

```
Erx # RecLink(441,440,4,_recFirst); // 1. Pos holen
```

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

```
    if (Lfs.P.Materialtyp=c_IO_Art) then begin
```

```
        Erx # RecLink(250,441,3,0); // Artikel holen
```

```
        Lfs.Kosten.PEH    # Art.PEH;
```

```
        Lfs.Kosten.MEH    # Art.MEH;
```

```
    end;
```

```
end;
```

```
gMDI # Lib_GuiCom:AddChildWindow(gMDI,'Lfs.Maske',here+':AusLFS');
```

```
// gleich in Neuanlage....
```

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

```
Mode # c_ModeNew;
```

```
w_Command # '->POS';
```



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

```
RETURN true;
```

```
end; // LFS
```

```
'Mnu.Fahrauftrag' : begin
```

```
// 2022-09-20 AH Proj. 2346/19 if (gMdiLfs<>0) then RETURN true;
```

```
// ST 2013-02-17 1449/5 : neue Variante für ggf. mehrere Auftragspositionen
```

```
if (Auf_Subst:ZuFahrauftrag() = false) then begin
```

```
    ErrorOutput;
```

```
    RETURN false;
```

```
end;
```

```
Lib_Mark:Reset(401);
```

```
gMDI # Lib_GuiCom:AddChildWindow(gMDI,'BA1.P.LFA.Maske',here+':AusLFS');
```

```
// gleich in Neuanlage....
```

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

```
Mode # c_ModeNew;
```

```
w_Command # '->POS';
```

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

```
RETURN true;
```

```
end;
```

```
// ST 2013-02-17: Alte Variante nur für aktuellen Auftrag
```

```
'Mnu.Fahrauftrag_BAK' : begin
```

```
if (gMdiLfs<>0) then RETURN true;
```

```
// Kreditlimit prüfen...
```

```
if ("Set.KLP.LFA-Druck"<>"") then
```

```
  if (Adr_K_Data:Kreditlimit(Auf.Rechnungsempf,"Set.KLP.LFA-Druck",n, var vKLim)=false) then RETU
```

```
RecBufClear(440);
```

```
Lfs.Nummer      # myTmpNummer;
```

```
Lfs.Anlage.Datum # today;
```

```
Lfs.Kundennummer # Auf.P.Kundennr;
```

```
Lfs.Kundenstichwort # Auf.P.KundenSW;
```

```
Lfs.Zieladresse  # Auf.Lieferadresse;
```

```
Lfs.Zielanschrift # Auf.Lieferanschrift;
```

```
RecLink(400,401,3,_recFirst); // Kopf holen
```

```
// MATERIAL -----
```

```
if (Wgr_Data:IstMat(Auf.P.Wgr.Dateinr)) or (Wgr_Data:IstMix(Auf.P.Wgr.Dateinr)) then begin
```

```
  // tmp. Lieferschein aufbauen
```

```
  Auf_Subs:Lieferschein(y,700);
```

```
  Erx # RecLink(441,440,4,_recFirst); // temp. Lieferpositionen holen
```

```
  if (Erx<=_rlocked) then begin
```

```
    RecLink(401,441,5,_recFirst); // Auftragspos holen
```

```
  end;
```

```
end
```

```
else
```

```
// ARTIKEL -----
```

```
if (Wgr_Data:IstArt(Auf.P.Wgr.Dateinr)) then begin
```

```
// gibts schon offene VLDAW ??
```

```
if (Auf_Data:UpdateVLDAW())=true) then RETURN true;
```

```
// sonst neuen LFS generieren:
```

```
// tmp. Lieferschein aufbauen
```

```
Auf_Sub:Lieferschein(y,700);
```

```
if (RecLinkInfo(441,440,4,_recCount)=0) then begin
```

```
    Msg(440105,"0,0,0,0);
```

```
    RETURN true;
```

```
end;
```

```
end; // Artikel
```

```
gMDI # Lib_GuiCom:AddChildWindow(gMDI,'BA1.P.LFA.Maske',here+':AusLFS');
```

```
// gleich in Neuanlage....
```

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

```
Mode # c_ModeNew;
```

```
w_Command # '->POS';
```

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

```
RETURN true;
```

```
end;
```

'Mnu.Matz' : begin

if (Auf.Vorgangstyp=c_VorlageAuf) then RETURN true;

// 02.11.2016 GuBE

if (Auf.Vorgangstyp=c_BOGUT) or (Auf.Vorgangstyp=c_REKOR) or (Auf.Vorgangstyp=c_GUT) or
(Auf.Vorgangstyp=c_BEL_LF) or (Auf.Vorgangstyp=c_BEL_KD) then begin

if ((Wgr_Data:IstMat(Auf.P.Wgr.Dateinr)) or (Wgr_Data:IstMix(Auf.P.Wgr.Dateinr))) then begin

Auswahl('Matz.Mat.GuBe')

end;

RETURN true;

end;

Erx # RecLink(819,401,1,0); // Warengruppe holen

// direkter Materialverkaufs-Auftrag?

if (Mode=c_ModeNew2) then begin

if (Wgr_Data:IstMat(Auf.P.Wgr.Dateinr)) or (Wgr_Data:IstMix(Auf.P.Wgr.Dateinr)) then begin

Auswahl('Material')

RETURN true;

end

else begin

RETURN true;

end;

end;

if ((Wgr_Data:IstMat(Auf.P.Wgr.Dateinr)) or (Wgr_Data:IstMix(Auf.P.Wgr.Dateinr))) then begin

Auswahl('Matz.Mat')

end;

if (Wgr_Data:IstArt(Auf.P.Wgr.Dateinr)) then begin

 Erx # RecLink(250,401,2,_RecFirst); // Artikel holen

 if (Erx<=_rLocked) then begin

 RecBufClear(252);

 Art.C.ArtikelNr # Auf.P.ArtikelNr;

 Art_Data:ReadCharge();

 end

else begin

 RETURN false;

end;

Auswahl('Matz.ArtC');

end;

end;

'Mnu.DFakt.Gut' : begin

 // Ankerfunktion

 if (RunAFX('Auf.P.DFakt', aMenuItem->wpName) < 0) then RETURN true;

 RecBufClear(404);

 gMDI # Lib_GuiCom:AddChildWindow(gMDI, 'Auf.A.Verwaltung', here+':AusAktion',y);

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

```
Lib_Sel:QInt(var vQ, 'Auf.A.Nummer' , '=', Auf.P.Nummer);
```

```
vQ # vQ +' AND (';
```

```
Lib_Sel:QInt(var vQ, 'Auf.A.Position' , '=', 0, ' ');
```

```
Lib_Sel:QInt(var vQ, 'Auf.A.Position' , '=', Auf.P.Position, 'OR');
```

```
vQ # vQ + ')';
```

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

```
Mode # c_modeBald + c_modeNew;
```

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

```
end;
```

```
'Mnu.DFakt.Bel' : begin
```

```
// Ankerfunktion
```

```
if ( RunAFX( 'Auf.P.DFakt', aMenuItem->wpName) < 0 ) then RETURN true;
```

```
// Auf_Data:DFaktBel();
```

```
// gZLList->WinUpdate(_WinUpdOn, _WinLstRecFromRecid | _WinLstRecDoSelect);
```

```
RecBufClear(404);
```

```
gMDI # Lib_GuiCom:AddChildWindow(gMDI, 'Auf.A.Verwaltung', here+':AusAktion',y);
```

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

```
Lib_Sel:QInt(var vQ, 'Auf.A.Nummer' , '=', Auf.P.Nummer);
```

```
vQ # vQ +' AND (';
```

```

Lib_Sel:QInt(var vQ, 'Auf.A.Position' , '=', 0, ' ');

Lib_Sel:QInt(var vQ, 'Auf.A.Position' , '=', Auf.P.Position, 'OR');

vQ # vQ + ' ';

Lib_Sel:QRecList(0, vQ);

```

```

Mode # c_modeBald + c_modeNew;

Lib_GuiCom:RunChildWindow(gMDI);

```

```

end;

```

```

'Mnu.DFakt' : begin

```

```

    // Ankerfunktion

```

```

    if ( RunAFX( 'Auf.P.DFakt', aMenuItem->wpName) < 0 ) then RETURN true;

```

```

    Erx # RecLink(819,401,1,0); // Warengruppe holen

```

```

    if (Wgr_Data:IstMat(Auf.P.Wgr.Dateinr)) or (Wgr_Data:IstMix(Auf.P.Wgr.Dateinr)) then begin

```

```

        Auswahl('DFakt.Mat')

```

```

    end;

```

```

    if (Wgr_Data:IstArt(Auf.P.Wgr.Dateinr)) then begin

```

```

        Erx # RecLink(250,401,2,_RecFirst); // Artikel holen

```

```

        if (Erx<=_rLocked) then begin

```

```

            RecBufClear(252);

```

```

            Art.C.ArtikelNr    # Auf.P.ArtikelNr;

```

```

            Art_Data:ReadCharge();

```

```

        end

```

else begin

RETURN false;

end;

// mehrere Chargen vorhanden?

if (RecLinkInfo(252,250,4,_recCount)>1) then begin

Auswahl('DFakt.ArtC');

end

else begin

if (Auf.P.MEH.Preis='Stk') then vM # cnvfi(Auf.P.Prd.Rest.Stk)

else if (Auf.P.MEH.Preis=Auf.P.MEH.Einsatz) then vM # Auf.P.Prd.Rest;

if (RecLinkInfo(409,401,15,_RecCount)>0) then begin

Auf_Data_Buchen:DFaktSL(true);

RETURN true;

end;

Auf_Data_Buchen:DFaktArtC(Art.Nummer,0,0,"y, Auf.P.Prd.Rest, Auf.P.Prd.Rest.Stk , vM, today);

// Auf_Data_Buchen:DFaktArtC(Art.Nummer,0,0,"y,0.0,0,0.0, today);

RefreshList(gZLList, _WinLstRecFromRecid | _WinLstRecDoSelect);

end;

end;

end;

'Mnu.Auswahl' : begin


```

vHdl # WinFocusGet();

if (vHdl<>0) then begin

  case (vHdl->wpname) of

    'edAuf.P.AbrufAufPos'      : Auswahl('AbrufPos');

    'edAuf.P.KundenMatArtNr_Mat' : Auswahl('KundenMatArtNr');

    'edAuf.P.Grundpreis'       : Auswahl('Preis');

    'edAuf.P.AbrufAufPos_Mat'  : Auswahl('AbrufPos');

  end;

end;

end;

```

```

'Mnu.Auf2Verpackung' : begin

  Auf_Data:Auf2Verpackung();

end;

```

```

'Mnu.Ang2Auf' : begin

  Auf_Sub:Ang2Auf();

  RefreshList(gZLList, _WinLstRecFromRecID | _WinLstRecDoSelect);

  if (gMDI=cDialog2) then begin

    vNr # Auf.Nummer;

    cZList2->WinUpdate(_WinUpdOn, _WinLstRecFromBuffer | _WinLstRecDoSelect);

    Auf.Nummer # vNr;

    RecRead(400,1,0);

  end;

```

end;

'Mnu.Auf2Anf' : begin

Ein_Subst:Auf2Anf();

RefreshList(gZLList, _WinLstRecFromRecID | _WinLstRecDoSelect);

if (gMDI=cDialog2) then begin

vNr # Auf.Nummer;

cZList2->WinUpdate(_WinUpdOn, _WinLstRecFromBuffer | _WinLstRecDoSelect);

Auf.Nummer # vNr;

RecRead(400,1,0);

end;

end;

'Mnu.BAG.Abschluss' : begin

Auf_SL_Data:BAGAbschluss_allePos();

RefreshList(gZLList, _WinLstRecFromRecID | _WinLstRecDoSelect);

end;

'Mnu.Feinabrufe' : begin

Auswahl('Feinabrufe');

end;

'Mnu.Stueckliste' : begin

 Auswahl('Stueckliste');

end;

'Mnu.Kalkulation' : begin

 Auswahl('Kalkulation');

end;

'Mnu.PosAufpreise' : begin

 Auswahl('Aufpreise');

end;

'Mnu.KopfAufpreise' : begin

 Auswahl('KopfAufpreise');

end;

'Mnu.Reserv' : begin

 Auswahl('Reservierungen');

end;

'Mnu.Grobplanung' : begin

```
Auswahl('Grobplanung');
```

```
end;
```

```
'Mnu.Aktionen' : begin
```

```
  Auswahl('Aktionen');
```

```
end;
```

```
'Mnu.Druck.Status' : begin
```

```
  vBuf401 # RekSave(401);
```

```
  Lib_Dokumente:PrintForm(400,'Statusblatt',false);
```

```
  RekRestore(vBuf401);
```

```
end;
```

```
'Mnu.Druck.Angebot' : begin
```

```
  RecLink(400,401,3,_RecFirst); // Kopf holen
```

```
  vBuf401 # RekSave(401);
```

```
  if (Lib_Dokumente:Printform(400,'Angebot',true)) and
```

```
    (Set.Auf.DruckInAktYN) then begin
```

```
    RecLink(401,400,9,_RecFirst); // 1.Position holen
```

```
    // Druck-Aktion anlegen
```

```
    RecBufClear(404);
```

```
    Auf.A.Aktionstyp  # c_Akt_Druck;
```

```
    Auf.A.Bemerkung   # c_AktBem_Angebot;
```

```
Auf.A.Aktionsnr    # Auf.Nummer;

Auf.A.Aktionspos   # 0;

Auf.A.Aktionsdatum # Today;

Auf.A.TerminStart  # Today;

Auf.A.TerminEnde   # Today;

Auf_A_Data:NeuAmKopfAnlegen();

Auf_Data:BerechneMarker();

vTmp # winsearch(gMDI,cZList);

if (vTmp<>0) then vTmp->WinUpdate(_WinUpdOn, _WinLstRecFromRecID | _WinLstRecDoSelect);

end;

RekRestore(vBuf401);

end;
```

```
'Mnu.Druck.AB' : begin
```

```
  Auf_Sub:DruckAB();
```

```
end;
```

```
'Mnu.Druck.Anf' : begin
```

```
  Auf_P_Sub:DruckAnfrage_Init();
```

```
end;
```

```
'Mnu.Druck.Avis' : begin
```

```
  Auf_P_Sub:DruckAvis();
```

```
end;
```

```

'Mnu.Druck.RE', 'Mnu.Druck.Gut', 'Mnu.Druck.Bel' : begin

  vBuf401 # RekSave(401);

  Lib_Faktura:Rechnungsdruck(n);

  RekRestore(vBuf401);

  gMDI->WinfocusSet(false);

  vTmp # winsearch(gMDI,cZList);

  vTmp->WinUpdate(_WinUpdOn, _WinLstRecFromRecID | _WinLstRecDoSelect);


  // ggf. Auftragskopfdaten liste aktualisieren

  vTmp # winsearch(gMDI,'ZL.Auftraege');

  if (vTmp <> 0) then

    vTmp->WinUpdate(_WinUpdOn, _WinLstRecFromRecID | _WinLstRecDoSelect);

  end;

```

```

'Mnu.Druck.Gelangensbest' : begin

  Auf_Sub:DruckGelangensbest();

end;

```

```

'Mnu.Druck.FM' : begin

  Auf_Sub:DruckFM();

end;

```

```

'Mnu.Druck.FMI' : begin

  RecLink(400,401,3,_RecFirst); // Kopf holen

```

```
vBuf401 # RekSave(401);  
  
Lib_Dokumente:Printform(400,'Fertigmeldung Intern',false);  
  
RekRestore(vBuf401);  
  
end;
```

```
'Mnu.Druck.RE.VS','Mnu.Druck.Gut.VS','Mnu.Druck.Bel.VS' : begin  
  
vBuf401 # RekSave(401);  
  
Lib_Faktura:Rechnungsdruck(y);  
  
RekRestore(vBuf401);  
  
end;
```

```
'Mnu.Druck.RE.Proforma' : begin  
  
vBuf401 # RekSave(401);  
  
if (Lib_Dokumente:PrintForm(450,'Proformarechnung',false)) then begin  
  
    // 31.03.2021 AH: Neu:  
  
    RecLink(401,400,9,_RecFirst); // 1.Position holen  
  
    // Druck-Aktion anlegen  
  
    RecBufClear(404);  
  
    Auf.A.Aktionstyp # c_Akt_Druck;  
  
    Auf.A.Bemerkung # 'Proformarechnung';  
  
    Auf.A.Aktionsnr # Auf.Nummer;  
  
    Auf.A.Aktionspos # 0;  
  
    Auf.A.Aktionsdatum # Today;  
  
    Auf.A.TerminStart # Today;
```

```

Auf.A.TerminEnde # Today;

Auf_A_Data:NeuAmKopfAnlegen();

Auf_Data:BerechneMarker();

end;

RekRestore(vBuf401);

end;


'Mnu.Druck.BA' : begin

// Betriebsauftrag zum Auftrag lesen

Bag.P.Nummer # 0;

Bag.P.Position # 0;

Erx # RecLink(404,401,12,_RecFirst); // Aktionen loopen

WHILE (Erx <= _rLocked) DO BEGIN

if (Auf.A.Aktionstyp = c_Akt_BA) then begin

    Bag.P.Nummer # Auf.A.Aktionsnr;

    Bag.P.Position # Auf.A.Aktionspos;

    break;

end;

Erx # RecLink(404,401,12,_RecNext); // Aktionen loopen

END;


// Kein BA gefunden

Erx # RecRead(702,1,0);

if (Erx > _rLocked) then begin

    RETURN false;

```



```
end;  
  
Bag.Nummer # Bag.P.Nummer;  
  
RecRead(700,1,0);
```

```
// Formular drucken  
  
vBuf401 # RekSave(401);  
  
Lib_Dokumente:Printform(700,'Betriebsauftrag',true);  
  
RekRestore(vBuf401);  
  
end;
```

```
'Mnu.Druck.DmsDeckblatt' : begin  
  
    Lib_Dokumente:Printform(400,'DMS Deckblatt',false);  
  
end;
```

```
end; // case
```

```
end;
```

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

```
// IsPageActive
```

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

```
Sub IsPageActive(aName : alpha) : logic;  
  
begin  
  
    RETURN aName<>'NB.Kopftext' and aName<>'NB.Fusstext';
```

end

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

```
// EvtClicked
```

```
//      Button gedrückt
```

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

```
sub EvtClicked (
```

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

```
) : logic
```

```
local begin
```

```
    vName  : alpha;
```

```
    vQ     : Alpha(4096);
```

```
end;
```

```
begin
```

```
if (aEvt:Obj->wpname='bt.AnalyseErweitert') then Auswahl('AnalyseErweitert');
```

```
if (aEvt:Obj->wpname='bt.showpic') then begin
```

```
    Lib_Picture:ShowPic($Picture1->wpcaption);
```

```
    RETURN true;
```

```
end;
```

```
if (aEvt:Obj->wpName='bt.InternerText') then begin
```

```
    if (Auf.P.Nummer=0) or (Auf.P.Nummer>1000000000) then
```

```
        vName # myTmpText+'.401.'+CnvAI(Auf.P.Position,_FmtNumLeadZero | _FmtNumNoGroup,0,3)+'.01'
```

else

vName # '~401.'+CnvAl(Auf.P.Nummer,_FmtNumLeadZero | _FmtNumNoGroup,0,8)+'.'+CnvAl(Auf.P.

Mdi_RtfEditor_Main:Start(vName, Rechte[Rgt_Auf_P_Aendern], Translate('interner Text'));

end;

if (Mode=c_ModeView) then RETURN true;

case (aEvt:Obj->wpName) of

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

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

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

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

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

'bt.RechEmpf' : Auswahl('Rechnungsempf');

'bt.RechAnschr' : Auswahl('Rechnungsanschr');

'bt.BestBearbeiter' : Auswahl('Ansprechpartner');

'bt.BDS' : Auswahl('BDSNummer');

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

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

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

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

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

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

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

'bt.AbmEH' : Auswahl('AbmessungsEH');

```

'bt.GewEH'      : Auswahl('GewichtsEH');

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

'bt.Vertreter1'    : Auswahl('Vertreter1');

'bt.Vertreter2'    : Auswahl('Vertreter2');


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

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


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

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

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

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

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

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

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

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

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

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

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

'bt.Autoaufpreis'  : begin

    ApL_Data:AutoGenerieren(401);

    Auf_Data:SumAufpreise();

    Auf.P.Gesamtpreis # Auf_data:SumGesamtpreis(Auf.P.Menge, "Auf.P.Stückzahl" , Auf.P.Gewicht);


    $lb.Aufpreise_Mat->wpcaption    # ANum(Auf.P.Aufpreis,2);

    $lb.Aufpreise->wpcaption        # ANum(Auf.P.Aufpreis,2);

```

```
$lb.Kalkuliert->wpcaption      # ANum(Auf.P.Kalkuliert,2);  
$lb.Poswert->wpcaption        # ANum(Auf.P.Gesamtpreis,2);  
$lb.Poswert_Mat->wpcaption    # ANum(Auf.P.Gesamtpreis,2);  
$lb.Rohgewinn->wpcaption      # ANum(Auf.P.Gesamtpreis - "Auf.P.GesamtwertEKW1",2);  
$lb.P.Einzelpreis_Mat->wpcaption # ANum(Auf.P.Einzelpreis,2);  
  
$RL.Aufpreise->winupdate(_winupdon,_WinLstFromFirst);  
end;
```

```
'bt.Auftragsart_Mat' : Auswahl('Auftragsart');  
'bt.Abruf_Mat'      : Auswahl('Abruf');  
'bt.Warengruppe_Mat' : Auswahl('Warengruppe');  
'bt.Projekt_Mat'     : Auswahl('Projekt');  
'bt.Artikelnummer_Mat' : Auswahl('Artikelnummer_Mat');  
'bt.Guete_Mat'       : Auswahl('Guete');  
'bt.Guete'           : Auswahl('Guete');  
'bt.Guetenstufe_Mat' : Auswahl('Guetenstufe');  
'bt.AusfOben_Mat'    : Auswahl('AusfOben');  
'bt.AusfUnten_Mat'   : Auswahl('AusfUnten');  
'bt.KundenArtNr_Mat' : Auswahl('KundenArtNr');  
'bt.KundenMatArtNr_Mat' : Auswahl('KundenMatArtNr');  
'bt.Zeugnis_Mat'     : Auswahl('Zeugnis');  
'bt.Zeugnis'         : Auswahl('Zeugnis');  
'bt.Erzeuger_Mat'    : Auswahl('Erzeuger');  
'bt.Intrastat_Mat'   : Auswahl('Intrastat');  
'bt.Intrastat'       : Auswahl('Intrastat');
```

```

'bt.PreisMEH_Mat' : Auswahl('PreisMEH');

'bt.Kalkulation_Mat' : Auswahl('Kalkulation');

'bt.Preis_Mat' : Auswahl('Preis');

'bt.Aufpreise_Mat' : Auswahl('Aufpreise');

'bt.Autoaufpreis_Mat' : begin

    ApL_Data:AutoGenerieren(401);

    Auf_Data:SumAufpreise();

    Auf.P.Gesamtpreis # Auf_data:SumGesamtpreis(Auf.P.Menge, "Auf.P.Stückzahl" , Auf.P.Gewicht);


    $lb.Aufpreise_Mat->wpcaption # ANum(Auf.P.Aufpreis,2);

    $lb.Aufpreise->wpcaption # ANum(Auf.P.Aufpreis,2);

    $lb.Kalkuliert->wpcaption # ANum(Auf.P.Kalkuliert,2);

    $lb.Poswert->wpcaption # ANum(Auf.P.Gesamtpreis,2);

    $lb.Poswert_Mat->wpcaption # ANum(Auf.P.Gesamtpreis,2);

    $lb.Rohgewinn->wpcaption # ANum(Auf.P.Gesamtpreis - "Auf.P.GesamtwertEKW1",2);

    $lb.P.Einzelpreis_Mat->wpcaption # ANum(Auf.P.Einzelpreis,2);


    $RL.Aufpreise_Mat->winupdate(_winupdon,_WinLstFromFirst);

end;


'bt.Standardtext2' : Auswahl('Text2');


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

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

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

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

```

```
'bt.Etikett'      : Auswahl('Etikett');  
'bt.Etikett2'    : Auswahl('Etikett2');  
'bt.Skizze'      : Auswahl('Skizze');  
'bt.EinsatzVPG'  : Auswahl('EinsatzVPG');  
'bt.VorlageBAG'  : Auswahl('VorlageBAG');  
'bt.Verpackung'  : Auswahl('Verpackung');
```

```
'bt.Standardtext2.Add' : begin
```

```
    RecBufClear(837);      // ZIELBUFFER LEEREN
```

```
    gMDI # Lib_GuiCom:AddChildWindow(gMDI,'Txt.Verwaltung',here+':AusTextAdd');
```

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

```
    if (Auf.Vorgangstyp=c_ANG) then
```

```
        Gv.Alpha.01 # 'A'
```

```
    else
```

```
        Gv.Alpha.01 # 'V'
```

```
    vQ # ";
```

```
    Lib_Sel:QenthaeltA(var vQ, 'Txt.Bereichstring', Gv.Alpha.01);
```

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

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

```
end;
```

```
'bt.Kopftext.Add' : begin
```

```
    RecBufClear(837);      // ZIELBUFFER LEEREN
```

```
    gMDI # Lib_GuiCom:AddChildWindow(gMDI,'Txt.Verwaltung',here+':AusKopfTextAdd');
```

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

```
    if (Auf.Vorgangstyp=c_ANG) then
```

```

    Gv.Alpha.01 # 'A'

else

    Gv.Alpha.01 # 'V'

vQ # "";

Lib_Sel:QenthaeltA(var vQ, 'Txt.Bereichstring', Gv.Alpha.01);

Lib_Sel:QRecList(0, vQ);

Lib_GuiCom:RunChildWindow(gMDI);

end;

'bt.Fusstext.Add' : begin

    RecBufClear(837);      // ZIELBUFFER LEEREN

    gMDI # Lib_GuiCom:AddChildWindow(gMDI,'Txt.Verwaltung',here+':AusFusstextAdd');

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

    if (Auf.Vorgangstyp=c_ANG) then

        Gv.Alpha.01 # 'A'

    else

        Gv.Alpha.01 # 'V'

    vQ # "";

    Lib_Sel:QenthaeltA(var vQ, 'Txt.Bereichstring', Gv.Alpha.01);

    Lib_Sel:QRecList(0, vQ);

    Lib_GuiCom:RunChildWindow(gMDI);

end;

end;

```


end;

//=====

// EvtPageSelect

// Seitenauswahl von Notebooks

//=====

sub EvtPageSelect(

 aEvt : event; // Ereignis

 aPage : int;

 aSelecting : logic;

) : logic

begin

 if (aPage<>0) then begin

 RefreshIfm(aPage->wpname);

 end;

 // AnalyseErweitert

 if (aSelecting) and (aPage->wpName='NB.Page4') and (Set.LyseErweitertYN) then begin

 w_TimerVar # 'AnalyseErweitert';

 gTimer2 # SysTimerCreate(300,1,gMdi);

 end;

 RETURN true;

end;

```

//=====

// EvtLstDataInit

//

//=====

sub EvtLstDataInit(
    aEvt    : Event;
    aRecId  : int;
    Opt aMark : logic;
);
local begin
    Erx    : int;
    vCol    : int;
    vStellen : int;
    vW      : Word;
end;
begin

    Lib_GuiCom:ZLQuickJumpInfo($clmAuf.P.Nummer);

    if (aMark) then begin
        if (RunAFX('Auf.P.EvtLstDataInit','y' + aEvt:obj->wpName)<0) then RETURN;
    end

    else if (RunAFX('Auf.P.EvtLstDataInit','n' + aEvt:obj->wpName)<0) then RETURN;

```

```
vCol # RGB(255,255,255);
```

```
// in Erfassung???
```

```
if (aEvt:obj->wpname='ZL.Erfassung') then begin
```

```
    Gv.Num.01 # 0.0;
```

```
    Gv.Num.02 # 0.0;
```

```
    Erx # RecLink(403,401,6,_RecFirst);
```

```
    WHILE (Erx<=_rLocked) and ((Gv.num.01=0.0) or (Gv.Num.02=0.0)) do begin
```

```
        if ("Auf.Z.Schlüssel"='*RAB1') then Gv.Num.01 # (-1.0) * Auf.Z.Menge;
```

```
        if ("Auf.Z.Schlüssel"='*RAB2') then Gv.Num.02 # (-1.0) * Auf.Z.Menge;
```

```
        Erx # RecLink(403,401,6,_RecNext);
```

```
    END;
```

```
    RETURN;
```

```
end; // in Erfassung
```

```
// Zeilenfarbe anpassen
```

```
if (Mode=c_ModeList) then RecLink(400,401,3,_RecFirst); // Kopf holen
```

```
if (Auf.LiefervertragYN) then vCol # Set.Auf.Col.LVertrag;
```

```
if ("Set.KLP.Auf-Anlage"='A') and (Auf.Freigabe.Datum=0.0.0) and (Auf.Vorgangstyp=c_Auf) and (Auf.Lie
```

```
if (Auf.Vorgangstyp=c_ANG) then vCol # Set.Auf.Col.Ang;
```

```
if (Auf.Vorgangstyp=c_BOGUT) or (Auf.Vorgangstyp=c_REKOR) or (Auf.Vorgangstyp=c_GUT) then vCol
```

```
if ("Auf.P.Löschmarker"='*') then vCol # Set.Col.RList.Deletd;
```

```
if (aMark=n) then begin
```

```
//debugx('KEY401 Farbe '+aint(vCol));
```

```
//if (vCol=16777215) then
```

```
//    Lib_GuiCom:ZLColorLine(gZLList,vCol,true)
```

//else

Lib_GuiCom:ZLColorLine(gZLList,vCol);

end;

Gv.Alpha.20 # Auf.P.Aktionsmarker;

if (Gv.Alpha.20='') then

if (RecLinkInfo(404,401,12,_RecCOunt)>0) then Gv.Alpha.20 # '!';

Gv.Alpha.20 # "Auf.P.Löschmarker" + Gv.alpha.20;

if (RecLinkInfo(203,401,18,_RecCOunt)>0) then Gv.Alpha.20 # Gv.Alpha.20 + 'R';

GV.Alpha.20 # GV.Alpha.20 + Auf.P.Flags;

vStellen # Set.Stellen.Menge;

case Strcnv(Auf.P.MEH.Einsatz,_strupper) of

'KG','T' : vStellen # Set.Stellen.Gewicht;

'STK' : vStellen # 0;

end;

Gv.alpha.01 # ANum(Auf.P.Prd.Plan,vStellen)+' '+Auf.P.MEH.Einsatz;

Gv.Alpha.02 # ANum(Auf.P.Prd.VSB,vStellen)+' '+Auf.P.MEH.Einsatz;

Gv.Alpha.03 # ANum(Auf.P.Prd.VSAuf,vStellen)+' '+Auf.P.MEH.Einsatz;

Gv.Alpha.04 # ANum(Auf.P.Prd.LFS,vStellen)+' '+Auf.P.MEH.Einsatz;

Gv.Alpha.05 # ANum(Auf.P.Prd.Rech,vStellen)+' '+Auf.P.MEH.Preis;//Auf.P.MEH.Einsatz;

Gv.Alpha.06 # ANum(Auf.P.Prd.Rest,vStellen)+' '+Auf.P.MEH.Einsatz;

Gv.Alpha.07 # ANum(Auf.P.Prd.EkBest,vStellen)+' '+Auf.P.MEH.Einsatz;

Gv.Alpha.10 # ANum(Auf.P.Menge.Wunsch,vStellen)+' '+Auf.P.MEH.Wunsch;

Gv.Alpha.11 # ANum(Auf.P.Prd.Reserv,vStellen)+' '+Auf.P.MEH.Einsatz;

RecLink(101, 400, 2, _recFirst); // Lieferanschrift

GV.Alpha.30 # Adr.A.Stichwort;

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

if(Erx > _rLocked) then

RecBufClear(835);

GV.Alpha.31 # AAr.Bezeichnung;

if (Auf.P.Termin1W.Art='KW') then begin

GV.Ints.01 # Auf.P.Termin1W.Zahl;

GV.Ints.02 # Auf.P.TerminZ.Zahl;

end

else begin

Lib_Berechnungen:KW_aus_Datum(Auf.P.Termin1Wunsch, VAR GV.Ints.01, VAR vW);

Lib_Berechnungen:KW_aus_Datum(Auf.P.TerminZusage, VAR GV.Ints.02, VAR vW);

end;

// Gv.Num.01 # 0.0;

Gv.Num.01 # Lib_Berechnungen:Prozent(Auf.P.Prd.LFS, Auf.P.Menge.Wunsch);

Gv.Num.01 # 100.0 - Gv.Num.01;

if (Gv.num.01<0.0) then GV.Num.01 # 0.0;

Gv.Num.01 # Auf.P.Gesamtpreis * Gv.Num.01 / 100.0;

```

// if (Auf.P.MEH.Preis=Auf.P.MEH.Einsatz) then

// Gv.Num.01  # Auf.P.Gesamtpreis / Auf.P.Menge * Auf.P.Prd.Rest


// else

/*

if (auf.P.Menge<>0.0) then begin

    if (Auf.P.MEH.Preis=Auf.P.MEH.Einsatz) then

        Gv.Num.01  # Auf.P.Gesamtpreis / Auf.P.Menge * Auf.P.Prd.Rest

    else

        if (Auf.P.MEH.Preis='Stk') then

            Gv.Num.01  # Auf.P.Gesamtpreis / Auf.P.Stückzahl * Au.P.Prd.Rest

end;

*/

```

```

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

end;

```

```

//=====

```

```

// EvtChanged

```

```

//

```

```

//=====

```

```

sub EvtChanged(

```

```

    aEvt          : event;    // Ereignis

```

```

): logic

```

```

local begin

```

```

Erx      : int;

vName    : alpha;

vTxtHdl  : int;

end;

begin

    // 25.06.2014

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

    if (aEvt:Obj->wpname='cbAuf.LiefervertragYN') then begin

        if (Auf.LiefervertragYN) then begin

            Lib_guiCom:Enable($edAuf.GltigkeitVom);

            Lib_guiCom:Enable($edAuf.GltigkeitBis);

            Auf.AbrufYN  # false;

            Auf.PAbrufYN # false;

            $cbAuf.AbrufYN->winupdate(_WinUpdFld2Obj);

            $cbAuf.PAbrufYN->winupdate(_WinUpdFld2Obj);

        end

        else if (Auf.Vorgangstyp=c_AUF) then begin

            Lib_guiCom:Disable($edAuf.GltigkeitVom);

            Lib_guiCom:Disable($edAuf.GltigkeitBis);

        end;

    end;

    if (aEvt:Obj->wpname='cbAuf.AbrufYN') and (Auf.AbrufYN) then begin

        Lib_guiCom:Disable($edAuf.GltigkeitVom);

        Lib_guiCom:Disable($edAuf.GltigkeitBis);

        Auf.LiefervertragYN # false;

```

```

Auf.PAbrufYN      # false;

$cbAuf.LiefervertragYN->winupdate(_WinUpdFld2Obj);

$cbAuf.PAbrufYN->winupdate(_WinUpdFld2Obj);

end;

if (aEvt:Obj->wpname='cbAuf.PAbrufYN') and (Auf.PAbrufYN) then begin

    Lib_guiCom:Disable($edAuf.GltigkeitVom);

    Lib_guiCom:Disable($edAuf.GltigkeitBis);

    Auf.LiefervertragYN # false;

    Auf.AbrufYN      # false;

    $cbAuf.LiefervertragYN->winupdate(_WinUpdFld2Obj);

    $cbAuf.AbrufYN->winupdate(_WinUpdFld2Obj);

end;


if (aEvt:Obj->wpname='cbAuf.P.StehendYN') and (Auf.P.StehendYN) then begin

    Auf.P.LiegendYN # n;

    $cbAuf.P.LiegendYN->winupdate(_WinUpdFld2Obj);

end;

if (aEvt:Obj->wpname='cbAuf.P.LiegendYN') and (Auf.P.LiegendYN) then begin

    Auf.P.StehendYN # n;

    $cbAuf.P.StehendYN->winupdate(_WinUpdFld2Obj);

end;


if (aEvt:Obj->wpname='cbAuf.WaehrungFixYN') then begin

    if ("Auf.WährungFixYN") then begin

        Lib_GuiCom:Enable($edAuf.Waehrungskurs);

```



```

end

else begin

  Lib_GuiCom:Disable($edAuf.Waehrungskurs);

end;

Erx # RecLink(814,400,8,_RecFirst); // Währung holen

if (Erx<=_rLocked) and ("Auf.WährungFixYN"=y) then

  "Auf.Währungskurs" # Wae.VK.Kurs;

else

  "Auf.Währungskurs" # 0.0;

  $edAuf.Waehrungskurs->winupdate(_WinUpdFld2Obj);

end;


if (aEvt:Obj->wpName='cb.Text1') then begin

  if ($cb.Text1->wpCheckState=_WinStateChkChecked) then begin

    $cb.Text2->wpcheckstate # _WinStateChkUnchecked;

    $cb.Text3->wpcheckstate # _WinStateChkUnchecked;

    Auf.P.TextNr1 # 400;

    Auf.P.TextNr2 # 0;

    $edAuf.P.TextNr2b->wpCaptionInt # 0;

    AufPTextLoad();

    RefreshIfm('Text');

  end;

end;

if (aEvt:Obj->wpName='cb.Text2') then begin

  if ($cb.Text2->wpCheckState=_WinStateChkChecked) then begin

    $cb.Text1->wpcheckstate # _WinStateChkUnchecked;

  end;

end;

```

```

$cb.Text3->wpcheckstate # _WinStateChkUnchecked;

Auf.P.TextNr1 # 0;

Auf.P.TextNr2 # 0;

$edAuf.P.TextNr2b->wpCaptionInt # 0;

AufPTextLoad();

RefreshIfm('Text');

end;

end;

if (aEvt:Obj->wpName='cb.Text3') then begin

if ($cb.Text3->wpCheckState=_WinStateChkChecked) then begin

$cb.Text1->wpcheckstate # _WinStateChkUnchecked;

$cb.Text2->wpcheckstate # _WinStateChkUnchecked;


vName # '~837.'+CnvAI(Auf.P.TextNr2,_FmtNumLeadZero | _FmtNumNoGroup,0,8);


Auf.P.TextNr1 # 401;

Auf.P.TextNr2 # 0;

$edAuf.P.TextNr2b->wpCaptionInt # 0;


vTxtHdl # $Auf.P.TextEditPos->wpdbTextBuf;

Lib_Texte:TxtLoadLangBuf(vName,vTxtHdl, Auf.Sprache);


if (Auf.P.Nummer=0) or (Auf.P.Nummer>1000000000) then

vName # myTmpText+'.401.'+CnvAI(Auf.P.Position,_FmtNumLeadZero | _FmtNumNoGroup,0,3)

else

vName # '~401.'+CnvAI(Auf.P.Nummer,_FmtNumLeadZero | _FmtNumNoGroup,0,8)+'.'+CnvAI(Auf.P

```

```
RunAFX( 'Auf.P.AusText', 'Pos|'+aint(vTxtHdl));  
  
$Auf.P.TextEditPos->wpcustom # vName;  
  
$Auf.P.TextEditPos->WinUpdate(_WinUpdBuf2Obj);
```

```
  
RefreshIfm('Text');  
  
end;  
  
end;
```

```
  
RETURN true;  
  
end;
```

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

```
// EvtLstSelect
```

```
//          Zeilenauswahl von RecList/DataList
```

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

```
sub EvtLstSelect ( aEvt : event; aRecID : int; ) : logic
```

```
begin
```

```
  
// Auf-Kopf Selected??
```

```
if (aEvt:obj=cZList2) then begin
```

```
    RETURN true;
```

```
end;
```

```
  
if ( aRecId = 0 ) then
```

```
    RETURN true;
```

```

RecRead( 401, 0, _recId, aRecID );

if ( RunAFX( 'Auf.P.EvtLstSelect', aEvt:obj->wpName ) < 0 ) then

    RETURN true;

if ( aEvt:obj->wpName = 'ZL.AufPositionen' ) then begin

    Auf_P_SMain:SwitchMask( false );

end;

//$lbAuf.P.Info1->wpcaption # aint(auf.p.nummer)+': '+anum(auf.p.gewicht,0)+'kg '+anum(auf.p.menge,0)-

RefreshMode(true);

end;

//=====

// EvtLstRecControl

//      Anzeigekontrolle der Zugriffsliste

//=====

sub EvtLstRecControl(

    aEvt : event;

    aRecId : int;

) : logic

begin

    RETURN true;

// RETURN (Auf.P.Nummer<1000000000);

```

end;

//=====

// EvtClose

// Schliessen Aufes Fensters

//=====

sub EvtClose

(

 aEvt : event; // Ereignis

): logic

local begin

 vTxtHdl : int;

end;

begin

 vTxtHdl # \$Auf.P.TextEditPos->wpdbTextBuf;

 if (vTxtHdl<>0) then begin

 TextClose(vTxtHdl);

 end;

 vTxtHdl # \$Auf.P.TextEditKopf->wpdbTextBuf;

 if (vTxtHdl<>0) then begin

 TextClose(vTxtHdl);

 end;

 vTxtHdl # \$Auf.P.TextEditFuss->wpdbTextBuf;

 if (vTxtHdl<>0) then begin

 TextClose(vTxtHdl);

```

end;

vTxtHdl # $Auf.P.TextStammdaten->wpdbTextBuf;

if (vTxtHdl<>0) then begin
    TextClose(vTxtHdl);
end;

if ((w_AuswahlMode=n) or (w_Context<>"")) then Lib_GuiCom:RememberList($ZL.Erfassung,'AUF.');
```

RETURN true;

```

end;
```

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

```
// AufPTextSave
```

```
//      Text abspeichern
```

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

```
sub AufPTextSave()
```

```
local begin
```

```
    vTxtHdl  : int;      // Handle des Textes
```

```
    vName    : alpha;
```

```
end;
```

```
begin
```

```
    // PosTextbuffer holen
```

```
    vTxtHdl # $Auf.P.TextEditPos->wpdbTextBuf;
```

```
    $Auf.P.TextEditPos->WinUpdate(_WinUpdObj2Buf);
```

```
if (Auf.P.Nummer=0) or (Auf.P.Nummer>1000000000) then begin
```

```
    vName # myTmpText+'.401.'+CnvAI(Auf.P.Position,_FmtNumLeadZero | _FmtNumNoGroup,0,3);
```

```
end
```

```
else begin
```

```
    if (Auf.P.TextNr1=401) then begin // Idividuell
```

```
        vName # '~401.'+CnvAI(Auf.P.Nummer,_FmtNumLeadZero | _FmtNumNoGroup,0,8)+'.'+CnvAI(Auf.P.
```

```
    end
```

```
else begin
```

```
    vName # ";
```

```
    TxtDelete(myTmpText+'.401.'+CnvAI(Auf.P.Position,_FmtNumLeadZero | _FmtNumNoGroup,0,3),0)
```

```
end;
```

```
end;
```

```
// PosText speichern
```

```
if ($Auf.P.TextEditPos->wpcustom=vName) and (vName<>") then begin
```

```
    if ((TextInfo(vTxtHdl,_TextSize)+TextInfo(vTxtHdl,_TextLines))=0) then begin
```

```
        TxtDelete(vName,0);
```

```
    end
```

```
else begin
```

```
    TxtWrite(vTxtHdl,vName, _TextUnlock);
```

```
end;
```

```
end;
```

```
// KopfTextBuffer holen
```

```
vTxtHdl # $Auf.P.TextEditKopf->wpdbTextBuf;
```

```

$Auf.P.TextEditKopf->WinUpdate(_WinUpdObj2Buf);

if (Auf.P.Nummer=0) or (Auf.P.Nummer>1000000000) then

    vName # myTmpText+'.401.K'

else

    vName # '~401.'+CnvAI(Auf.P.Nummer,_FmtNumLeadZero | _FmtNumNoGroup,0,8)+'.K';

// Kopftext speichern

if ($Auf.P.TextEditKopf->wpcustom=vName) then begin

    if ((TextInfo(vTxtHdl,_TextSize)+TextInfo(vTxtHdl,_TextLines))=0) then

        TxtDelete(vName,0)

    else

        TxtWrite(vTxtHdl,vName, _TextUnlock);

end;

```

```

// FussTextBuffer holen

vTxtHdl # $Auf.P.TextEditFuss->wpdbTextBuf;

$Auf.P.TextEditFuss->WinUpdate(_WinUpdObj2Buf);

if (Auf.P.Nummer=0) or (Auf.P.Nummer>1000000000) then

    vName # myTmpText+'.401.F'

else

    vName # '~401.'+CnvAI(Auf.P.Nummer,_FmtNumLeadZero | _FmtNumNoGroup,0,8)+'.F';

// Kopftext speichern

if ($Auf.P.TextEditFuss->wpcustom=vName) then begin

    if ((TextInfo(vTxtHdl,_TextSize)+TextInfo(vTxtHdl,_TextLines))=0) then

        TxtDelete(vName,0)

    else

```



```

    TxtWrite(vTxtHdl,vName, _TextUnlock);

end;

end;

//=====

// AufPTextLoad

//      Text Auflsesen

//=====

sub AufPTextLoad()

local begin

    vTxtHdl    : int;      // Handle des Textes

    vName      : alpha;

end

begin

    if (Auf.P.TextNr2 = 0) then

        RecBufClear(837);

    // PosText laden

    vTxtHdl # $Auf.P.TextEditPos->wpdbTextBuf;

    if (Auf.P.Nummer=0) or (Auf.P.Nummer>1000000000) then begin

        vName # myTmpText+'.401.'+CnvAl(Auf.P.Position,_FmtNumLeadZero | _FmtNumNoGroup,0,3);

        if (Auf.P.TextNr1=400) then      // anderer Psoitionstext

```

```

vName # myTmpText+'.401.'+CnvAl(Auf.P.Textnr2,_FmtNumLeadZero | _FmtNumNoGroup,0,3);

if (Auf.P.TextNr1=0) then      // Standardtext

    vName # '~837.'+CnvAl(Auf.P.TextNr2,_FmtNumLeadZero | _FmtNumNoGroup,0,8);

end

else begin // vorhandener Auftrag....

if ($cb.Text1->wpCheckState = _WinStateChkChecked) then begin

    if (Auf.P.TextNr1=400) then begin // anderer Psoitionstext

        vName # '~401.'+CnvAl(Auf.P.Nummer,_FmtNumLeadZero | _FmtNumNoGroup,0,8)+'.'+CnvAl(Auf.P.Nummer,_FmtNumLeadZero | _FmtNumNoGroup,0,8);

        if (Auf.P.Nummer=0) or (Auf.P.Nummer>1000000000) then begin

            vName # myTmpText+'.401.'+CnvAl(Auf.P.Textnr2,_FmtNumLeadZero | _FmtNumNoGroup,0,3);

            end;

        end;

    end;

end;

if ($cb.Text2->wpCheckState = _WinStateChkChecked) then begin

    if (Auf.P.TextNr1=0) then begin // Standardtext

        vName # '~837.'+CnvAl(Auf.P.TextNr2,_FmtNumLeadZero | _FmtNumNoGroup,0,8);

        end;

    end;

end;

if ($cb.Text3->wpCheckState = _WinStateChkChecked) then begin

    if (Auf.P.TextNr1=401) then begin // ldividuell

        vName # '~401.'+CnvAl(Auf.P.Nummer,_FmtNumLeadZero | _FmtNumNoGroup,0,8)+'.'+CnvAl(Auf.P.Nummer,_FmtNumLeadZero | _FmtNumNoGroup,0,8);

        end;

    end;

end;

```

end;

end;

if (\$Auf.P.TextEditPos->wpcustom<>vName) or (Mode=c_ModeView) then begin

if (StrFind(vName,'~837',0)<>0) then begin

Lib_Texte:TxtLoadLangBuf(vName,vTxtHdl, Auf.Sprache);

end

else begin

if (TextRead(vTxtHdl,vName, _TextUnlock)>_rLocked) then begin

TextClear(vTxtHdl);

end;

end;

RunAFX('Auf.P.AusText', 'Pos|'+aint(vTxtHdl));

\$Auf.P.TextEditPos->wpcustom # vName;

\$Auf.P.TextEditPos->WinUpdate(_WinUpdBuf2Obj);

end;

// KopfText laden

vTxtHdl # \$Auf.P.TextEditKopf->wpdbTextBuf;

if (Auf.P.Nummer=0) or (Auf.P.Nummer>1000000000) then

vName # myTmpText+'.401.K';

else

vName # '~401.'+CnvAI(Auf.P.Nummer,_FmtNumLeadZero | _FmtNumNoGroup,0,8)+'.K';

if (\$Auf.P.TextEditKopf->wpcustom<>vName) or (Mode=c_ModeView) then begin

```

if (TextRead(vTxtHdl,vName, _TextUnlock)>_rLocked) then

    TextClear(vTxtHdl);

    $Auf.P.TextEditKopf->wpcustom # vName;

    $Auf.P.TextEditKopf->WinUpdate(_WinUpdBuf2Obj);

end;


// FussText laden

vTxtHdl # $Auf.P.TextEditFuss->wpdbTextBuf;

if (Auf.P.Nummer=0) or (Auf.P.Nummer>1000000000) then

    vName # myTmpText+'.401.F'

else

    vName # '~401.'+CnvAI(Auf.P.Nummer,_FmtNumLeadZero | _FmtNumNoGroup,0,8)+'.F';

if ($Auf.P.TextEditFuss->wpcustom<>vName) or (Mode=c_ModeView) then begin

    if (TextRead(vTxtHdl,vName, _TextUnlock)>_rLocked) then

        TextClear(vTxtHdl);

        $Auf.P.TextEditFuss->wpcustom # vName;

        $Auf.P.TextEditFuss->WinUpdate(_WinUpdBuf2Obj);

    end;

end;


//=====

// AFObenRecCtrl

//      Populist Auflage Oben Filter

//=====

```

```
sub AFObenRecCtrl(
```

```
    aEvt : event;
```

```
    aRecId : int;
```

```
) : logic;
```

```
begin
```

```
    Auf.AF.Bezeichnung # StrCut(Auf.AF.Bezeichnung + ':' + Auf.AF.Zusatz, 1, 32);
```

```
    RETURN Auf.AF.Seite='1';
```

```
end;
```

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

```
// AFUntenRecCtrl
```

```
//      Populist Auflage Unten Filter
```

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

```
sub AFUntenRecCtrl(
```

```
    aEvt : event;
```

```
    aRecId : int;
```

```
) : logic;
```

```
begin
```

```
    Auf.AF.Bezeichnung # StrCut(Auf.AF.Bezeichnung + ':' + Auf.AF.Zusatz, 1, 32);
```

```
    RETURN Auf.AF.Seite='2';
```

```
end;
```

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

```
// Pflichtfelder
```

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

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

```
sub Pflichtfelder();
```

```
begin
```

```
    RunAFX('Auf.P.Pflichtfelder','');
```

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

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

```
    if(Auf.LiefervertragYN = true) then begin
```

```
        Lib_GuiCom:Pflichtfeld($edAuf.GltigkeitVom);
```

```
        Lib_GuiCom:Pflichtfeld($edAuf.GltigkeitBis);
```

```
    end;
```

```
    Lib_GuiCom:Pflichtfeld($edAuf.Vorgangstyp);
```

```
    Lib_GuiCom:Pflichtfeld($edAuf.Kundennr);
```

```
    Lib_GuiCom:Pflichtfeld($edAuf.Lieferadresse);
```

```
    Lib_GuiCom:Pflichtfeld($edAuf.lieferanschrift);
```

```
    Lib_GuiCom:Pflichtfeld($edAuf.Rechnungsempf);
```

```
    Lib_GuiCom:Pflichtfeld($edAuf.Rechnungsanschr);
```

```
    Lib_GuiCom:Pflichtfeld($edAuf.Waehrung);
```

```
    Lib_GuiCom:Pflichtfeld($edAuf.Lieferbed);
```

```
    Lib_GuiCom:Pflichtfeld($edAuf.Zahlungsbed);
```

```
    Lib_GuiCom:Pflichtfeld($edAuf.steuerschluessel);
```

```
    Lib_GuiCom:Pflichtfeld($edAuf.Versandart);
```

Lib_GuiCom:Pflichtfeld(\$edAuf.Sprache);

Lib_GuiCom:Pflichtfeld(\$edAuf.AbmessungenEH);

Lib_GuiCom:Pflichtfeld(\$edAuf.GewichtsEH);

Lib_GuiCom:Pflichtfeld(\$edAuf.Sachbearbeiter);

Lib_GuiCom:Pflichtfeld(\$edAuf.P.Auftragsart);

Lib_GuiCom:Pflichtfeld(\$edAuf.P.Warengruppe);

// Artikellogik

if (Wgr_Data:IstArt(Auf.P.Wgr.Dateinr)) or (Wgr_Data:IstMixArt(Auf.P.Wgr.Dateinr)) then begin

Lib_GuiCom:Pflichtfeld(\$edAuf.P.Artikelnr);

Lib_GuiCom:Pflichtfeld(\$edAuf.P.Menge.Wunsch);

Lib_GuiCom:Pflichtfeld(\$edAuf.P.MEH.Wunsch);

Lib_GuiCom:Pflichtfeld(\$edAuf.P.Menge.Wunsch);

Lib_GuiCom:Pflichtfeld(\$edAuf.P.Menge);

Lib_GuiCom:Pflichtfeld(\$edAuf.P.PEH);

Lib_GuiCom:Pflichtfeld(\$edAuf.P.MEH.PEH);

Lib_GuiCom:Pflichtfeld(\$edAuf.P.MEH.Preis);

if (Wgr_Data:IstMixArt(Auf.P.Wgr.Dateinr)) then

Lib_GuiCom:Pflichtfeld(\$edAuf.P.Guete);

Lib_GuiCom:Pflichtfeld(\$edAuf.P.Termin1Wunsch);

Lib_GuiCom:Pflichtfeld(\$edAuf.P.Termin1W.Art);

if (Auf.AbrufYN) or (Auf.PAbrufYN) or (Auf.Vorgangstyp=c_REKOR) then begin

Lib_GuiCom:Pflichtfeld(\$edAuf.P.AbrufAufNr);

Lib_GuiCom:Pflichtfeld(\$edAuf.P.AbrufAufPos);

end;

```

end;

// Materiallogik

if (Wgr_Data:IstMat(Auf.P.Wgr.Dateinr)) or (Wgr_Data:IstMixMat(Auf.P.Wgr.Dateinr)) then begin

    Lib_GuiCom:Pflichtfeld($edAuf.P.Guete_Mat);

    Lib_GuiCom:Pflichtfeld($edAuf.P.Gewicht_Mat);

    Lib_GuiCom:Pflichtfeld($edAuf.P.MEH.PEH_Mat);

    Lib_GuiCom:Pflichtfeld($edAuf.P.MEH.Preis_Mat);

    Lib_GuiCom:Pflichtfeld($edAuf.P.Termin1Wunsch_Mat);

    Lib_GuiCom:Pflichtfeld($edAuf.P.Termin1W.Art_Mat);

    if ($edAuf.P.Menge_Mat->wpvisible) then begin

        Lib_GuiCom:Pflichtfeld($edAuf.P.Menge_Mat);

    end;

    if (Auf.AbrufYN) or (Auf.PAbrufYN) or (Auf.Vorgangstyp=c_REKOR) then begin

        Lib_GuiCom:Pflichtfeld($edAuf.P.AbrufAufNr_Mat);

        Lib_GuiCom:Pflichtfeld($edAuf.P.AbrufAufPos_Mat);

    end;

end;

end;

end;

```

```

//=====

```

```

// EvtTimer

```

```

//

```

```

//=====

```

```

sub EvtTimer(

```



```

aEvt          : event;    // Ereignis

aTimerId      : int;

): logic

local begin

    vParent : int;

    vA      : alpha;

    vMode : alpha;

end;

begin

    if (gTimer2=aTimerId) then begin

        gTimer2->SysTimerClose();

        gTimer2 # 0;

    if (StrFind(w_TimerVar,'NimmMatFuerBA',0)>0) then begin

        vA # Str_Token(w_TimerVar,'|',2);

        BA1_Lohn_Sub:ErzeugeBAMitMat(cnvia(vA));

        RETURN true;

    end

    else if (w_TimerVar='->MATZ') then begin

        w_TimerVar # "";

        Auswahl('Matz.Mat');

    end

    else if (w_TimerVar='AnalyseErweitert') then begin

        w_TimerVar # "";

```

```

Auswahl('AnalyseErweitert');

end

else begin

// 13.02.2019 AH

if (StrFind(w_TimerVar,'LFSVERBUCHEN',0)>0) then begin

    if (Lfs.Datum.Verbucht=0.0.0) and

        (Rechte[Rgt_Lfs_Verbuchen]) then begin

            if (Msg(440007,",_WinIcoQuestion,_WinDialogYesNo,1)=_winIdyes) then begin

//      if (Dlg_Standard:Datum(Translate('Verbuchungsdatum'), var vDat, today)=false) then

//      RETURN

            Lfs_Data:Verbuchen(Lfs.Nummer, today, now);

            ErrorOutput;

            end;

            end;

        end;

end;

if (StrFind(w_TimerVar,'VLDAW',0)>0) then

    Lfs_VLDAW_Data:Druck_VLDAW();

if (StrFind(w_TimerVar,'LFA',0)>0) then

    Lfs_VLDAW_Data:Druck_LFA();

// 'LFS'- Timer speziell für Brockhaus deaktiviert

if (StrFind(w_TimerVar,'LFS',0)>0) and (Set.Installname != 'BSP') then

    Lfs_data:Druck_Auto();

```

```

if (w_TimerVar<>"") then

    RunAFX('Auf.P.EvtTimer',aint(aTimerID));

    w_TimerVar # "";

end;

end

else begin

    App_Main:EvtTimer(aEvt,aTimerId);

end;

RETURN true;

end;

//=====

// EvtPosChanged [15.03.2010/PW]

//

//=====

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

    vRect : rect;

    vHdl : int;

```

end

begin

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

// WORKAROUND

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

// Quickbar

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

if (vHdl<>0) then begin

vRect # vHdl->wpArea;

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

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

vHdl->wparea # vRect;

end;

if (aFlags & _winPosSized != 0) then begin

vRect # gZLList->wpArea;

vRect:right # aRect:right - aRect:left - 4;

vRect:bottom # aRect:bottom - aRect:top - 28 - 60 - w_QBHeight;

gZLList->wpArea # vRect;

Lib_GuiCom:ObjSetPos(\$lbAuf.P.Info1, 0, vRect:bottom + 8);

Lib_GuiCom:ObjSetPos(\$lbAuf.P.Info2, 0, vRect:bottom + 8 + 28);

```

// RecList:

vHdl # Winsearch(aEvt:Obj, 'ZL.Erfassung');

vRect      # vHdl->wpArea;

vRect:right  # aRect:right-aRect:left-4;

vRect:bottom # aRect:bottom-aRect:Top-28-w_QBHeight;

vHdl->wparea # vRect;


end;


RETURN true;

end


//=====

// JumpTo

//   Schnellsprung in andere Verwaltung duch Mittlere-Maustaste auf RecList-Spalte

//=====

sub JumpTo(

  aName : alpha;

  aBuf  : int);

local begin

  vQ   : alpha(1000);

end

begin

  if (aName = StrCnv('clmAuf.P.Nummer',_StrUpper) AND (aBuf->Auf.P.Nummer<>0)) then begin

```

```
Lib_Workbench:OpenKurzInfo(401, RecInfo(aBuf,_recID));
```

```
end;
```

```
if ((aName =^ 'edAuf.Kundennr') AND (Auf.Kundennr<>0)) then begin
```

```
    RekLink(100,400,1,0); // Kunde holen
```

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

```
    RETURN;
```

```
end;
```

```
if ((aName =^ 'edAuf.Lieferadresse') AND (Auf.Lieferadresse<>0)) then begin
```

```
    RekLink(100,400,12,0); // Lieferadresse holen
```

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

```
    RETURN;
```

```
end;
```

```
if ((aName =^ 'edAuf.Lieferanschrift') AND (Auf.Lieferanschrift<>0)) then begin
```

```
    RecLink(100,400,12,0); // Lieferadresse holen
```

```
    RekLink(101,400,2,0); // Anschrift holen
```

```
    Lib_Sel:QInt(var vQ, 'Adr.A.Adressnr', '=', Adr.Nummer);
```

```
    Lib_Guicom2:JumpToWindow('Adr.A.Verwaltung',vQ);
```

```
    RETURN;
```

```
end;
```

```
if ((aName =^ 'edAuf.Verbraucher') AND (Auf.Verbraucher<>0)) then begin
```

```
RekLink(100,400,3,0); // Verbraucher holen
```

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

```
RETURN;
```

```
end;
```

```
if ((aName =^ 'edAuf.Rechnungsempf') AND (Auf.Rechnungsempf<>0)) then begin
```

```
RekLink(100,400,4,0); // Rechn.Empf. holen
```

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

```
RETURN;
```

```
end;
```

```
if ((aName =^ 'edAuf.Rechnungsanschr') AND (Auf.Rechnungsanschr<>0)) then begin
```

```
RecLink(100,400,4,0); // Anschrift holen
```

```
Adr.A.Adressnr # Adr.Nummer;
```

```
Adr.A.Nummer # Auf.Rechnungsanschr;
```

```
RecRead(101,1,0);
```

```
Lib_Sel:QInt(var vQ, 'Adr.A.Adressnr', '=', Adr.Nummer);
```

```
Lib_Guicom2:JumpToWindow('Adr.A.Verwaltung',vQ);
```

```
RETURN;
```

```
end;
```

```
if ((aName =^ 'edAuf.Best.Bearbeiter') AND (Auf.Best.Bearbeiter<>'')) then begin
```

```
// RekLink(819,200,1,0); // Ansprechpartner holen
```

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

```
RETURN;
```

end;

if ((aName =^ 'edAuf.BDSNummer') AND (Auf.BDSNummer<>0)) then begin

 RekLink(836,400,11,0); // Anschrift holen

 Lib_Guicom2:JumpToWindow('BDS.Verwaltung');

 RETURN;

end;

if ((aName =^ 'edAuf.Waehrung') AND ("Auf.Währung"<>0)) then begin

 RekLink(814,400,8,0); // Währung holen

 Lib_Guicom2:JumpToWindow('Wae.Verwaltung');

 RETURN;

end;

if ((aName =^ 'edAuf.Lieferbed') AND (Auf.Lieferbed<>0)) then begin

 RekLink(815,400,5,0); // Lieferbed holen

 Lib_Guicom2:JumpToWindow('LiB.Verwaltung');

 RETURN;

end;

if ((aName =^ 'edAuf.Zahlungsbed') AND (Auf.Zahlungsbed<>0)) then begin

 RekLink(816,400,6,0); // Zahlungsbed holen

 Lib_Guicom2:JumpToWindow('Zab.Verwaltung');

 RETURN;

end;


```
if ((aName =^ 'edAuf.Versandart') AND (Auf.Versandart<>0)) then begin
```

```
    RekLink(817,400,7,0); // Versandart holen
```

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

```
    RETURN;
```

```
end;
```

```
if ((aName =^ 'edAuf.Vertreter1') AND (Auf.Vertreter<>0)) then begin
```

```
    RekLink(110,400,20,0); // Vertreter holen
```

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

```
    RETURN;
```

```
end;
```

```
if ((aName =^ 'edAuf.Land') AND (Auf.Land<>")) then begin
```

```
    RekLink(812,400,10,0); // Eintrittsland holen
```

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

```
    RETURN;
```

```
end;
```

```
if ((aName =^ 'edAuf.Steuerschluessel') AND ("Auf.Steuerschlüssel"<>0)) then begin
```

```
    RekLink(813,400,19,0); // Steuerschlüssel holen
```

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

```
    RETURN;
```

```
end;
```

```
if ((aName =^ 'edAuf.Sachbearbeiter') AND (Auf.Sachbearbeiter<>")) then begin
```

```
    Usr.Username # Auf.Sachbearbeiter;
```

```
RecRead(800,1,0)

Lib_Guicom2:JumpToWindow('Usr.Verwaltung');

RETURN;

end;
```

```
if ((aName =^ 'edAuf.Vertreter2') AND (Auf.Vertreter2<>0)) then begin

    RekLink(110,400,21,0); // Verband holen

    Lib_Guicom2:JumpToWindow('Ver.Verwaltung');

    RETURN;

end;
```

```
if ((aName =^ 'edAuf.P.Auftragsart') AND (Auf.P.Auftragsart<>0)) then begin

    RekLink(835,401,5,0); // vorgangsart holen

    Lib_Guicom2:JumpToWindow('AAr.Verwaltung');

    RETURN;

end;
```

```
if ((aName =^ 'edAuf.P.Warengruppe') AND (Auf.P.Warengruppe<>0)) then begin

    RekLink(819,401,1,0); // warengruppe holen

    Lib_Guicom2:JumpToWindow('Wgr.Verwaltung');

    RETURN;

end;
```

```
if ((aName =^ 'edAuf.P.Projektnummer') AND (Auf.P.Projektnummer<>0)) then begin

    RekLink(120,401,14,0); // Projektnummer holen

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

RETURN;

end;

if ((aName =^ 'edAuf.P.Guete') AND ("Auf.P.Güte"<>")) then begin

"MQu.Güte1" # "Auf.P.Güte";

RecRead(832,2,0)

Lib_Guicom2:JumpToWindow('MQu.Verwaltung');

RETURN;

end;

if ((aName =^ 'edAuf.P.ArtikelNr') AND (Auf.P.ArtikelNr<>')) then begin

RekLink(250,401,2,0); // ArtikelNummer holen

Lib_Guicom2:JumpToWindow('Art.Verwaltung');

RETURN;

end;

if ((aName =^ 'edAuf.P.Auftragsart_Mat') AND (Auf.P.Auftragsart<>0)) then begin

RekLink(835,401,5,0); // vorgangsart holen

Lib_Guicom2:JumpToWindow('AAr.Verwaltung');

RETURN;

end;

if ((aName =^ 'edAuf.P.Warengruppe_Mat') AND (Auf.P.Warengruppe<>0)) then begin

RekLink(819,401,1,0); // warengruppe holen

Lib_Guicom2:JumpToWindow('Wgr.Verwaltung');

RETURN;

end;

if ((aName =^ 'edAuf.P.Projektnummer_Mat') AND (Auf.P.Projektnummer<>0)) then begin

RekLink(120,401,14,0); // Projektnummer holen

Lib_Guicom2:JumpToWindow('Prj.Verwaltung');

RETURN;

end;

if ((aName =^ 'edAuf.P.AbrufAufPos_Mat') AND (Auf.P.AbrufAufPos<>0)) then begin

RekLink(401,400,23,0); // Abruf-best.Nr. holen

Lib_Guicom2:JumpToWindow('Erl.K.Verwaltung');

RETURN;

end;

if ((aName =^ 'edAuf.P.Erzeuger_Mat') AND (Auf.P.Erzeuger<>0)) then begin

RekLink(100,401,10,0); // Erzeuger Nr. holen

Lib_Guicom2:JumpToWindow('Adr.Verwaltung');

RETURN;

end;

if ((aName =^ 'edAuf.P.Kalkuliert_Mat') AND (Auf.P.Kalkuliert<>0.00)) then begin

RekLink(405,401,7,0); // Kalkulation holen

Lib_Guicom2:JumpToWindow('Auf.K.Verwaltung');

RETURN;

end;

```
if ((aName =^ 'edAuf.P.Grundpreis_Mat') AND (Auf.P.Grundpreis<>0.00)) then begin  
  todo('Preis')  
  
  //RekLink(100,401,4,0); // Grundpreis holen  
  
  Lib_Guicom2:JumpToWindow('Art.P.Verwaltung');  
  
  RETURN;  
  
end;
```

```
if ((aName =^ 'edAuf.P.TextNr2b') AND (aBuf<>0)) then begin  
  todo('Text2')  
  
  //RekLink(100,401,4,0); // Standardtext holen  
  
  Lib_Guicom2:JumpToWindow('Txt.Verwaltung');  
  
  RETURN;  
  
end;
```

```
if ((aName =^ 'edAuf.P.Verpacknr') AND (Auf.P.Verpacknr<>0)) then begin  
  RekLink(105,401,22,0); // Verpackungsnummer. holen  
  
  Lib_Guicom2:JumpToWindow('Adr.V.Verwaltung');  
  
  RETURN;  
  
end;
```

```
if ((aName =^ 'edAuf.P.Verwiegungsart') AND (Auf.P.Verwiegungsart<>0)) then begin  
  RekLink(818,401,9,0); // Verweigungsart holen  
  
  Lib_Guicom2:JumpToWindow('VwA.Verwaltung');  
  
  RETURN;  
  
end;
```

```
if ((aName =^ 'edAuf.P.Etikettentyp') AND (Auf.P.Etikettentyp<>0)) then begin
```

```
    RekLink(840,401,8,0); // Etikettentyp holen
```

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

```
    RETURN;
```

```
end;
```

```
if ((aName =^ 'edAuf.P.Zwischenlage') AND (Auf.P.Zwischenlage<>'')) then begin
```

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

```
    //RekLink(818,401,9,0); // Zwischenlage holen
```

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

```
    RETURN;
```

```
end;
```

```
if ((aName =^ 'edAuf.P.Unterlage') AND (Auf.P.Unterlage<>'')) then begin
```

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

```
    // RekLink(840,401,8,0); // Unterlage holen
```

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

```
    RETURN;
```

```
end;
```

```
if ((aName =^ 'edAuf.P.Umverpackung') AND (Auf.P.Umverpackung<>'')) then begin
```

```
    todo('Umverpackung')
```

```
    // RekLink(840,401,8,0); // Umverpackung holen
```

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

```
    RETURN;
```

```
end;
```

```
if ((aName =^ 'edAuf.P.Etikettentyp2') AND (Auf.P.Etikettentyp<>0)) then begin
```

```
    RekLink(840,401,8,0); // Etikettentyp holen
```

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

```
    RETURN;
```

```
end;
```

```
if ((aName =^ 'edAuf.P.Skizzennummer') AND (Auf.P.Skizzennummer<>0)) then begin
```

```
    RekLink(829,401,16,0); // Etikettentyp holen
```

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

```
    RETURN;
```

```
end;
```

```
if ((aName =^ 'edAuf.P.EinsatzVPG.Nr') AND (Auf.P.EinsatzVPG.Nr<>0)) then begin
```

```
    todo('EinsatzVPG2')
```

```
    // RekLink(829,401,16,0); // Einsatz holen holen
```

```
    Lib_Guicom2:JumpToWindow('Adr.V.Verwaltung');
```

```
    RETURN;
```

```
end;
```

```
if ((aName =^ 'edAuf.P.VorlageBAG') AND (Auf.P.VorlageBAG<>0)) then begin
```

```
    todo('VorlageBAG')
```

```
    // RekLink(829,401,16,0); // Vorlage-BAG holen
```

```
    Lib_Guicom2:JumpToWindow('BA1.Verwaltung');
```

```
    RETURN;
```

```
end;
```

end;

//=====

//=====

//=====