

@A+

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

//

// Prozedur App_Main

// OHNE E_R_G

// Info Startpunkt von Stahl-Control

//

//

// 05.02.2003 AI Erstellung der Prozedur

// 20.08.2009 AI EvtFocusInit erweitert für DUMMYNEW

// 19.04.2010 AI Maskenwechsel mit Maus bei gelinkten RecLists

// 28.07.2010 MS Lib_GuiCom:GetAlternativeName fuer alle Mdi's gesetzt

// 15.06.2011 ST BDE Benutzer hinzugefügt

// 22.02.2012 AI EvtInit entfernt das Flag aus RecLists

// 20.03.2012 ST ExterneAnhänge hinzugefügt

// 11.07.2012 AI Versuch die Refreshes zu minimieren

// 03.08.2012 AI Versuch die Refreshes zu minimieren

// 28.08.2012 AI "EvtFocusInit" setzt nun auch Hauptmenü

// 12.09.2012 ST "EvtMdiActivate": 'Mnu.SelAuswahl' hinzugefügt damit bei

// Selektionsdialogen auch direkt nach EvtActivate aktiv ist (Projekt 1326/292)

// 28.05.2013 ST Betriebsmenüaufruf für BagZeiten hinzugefügt

// 29.05.2013 ST Hintergrundbild "Testsystem" sucht nach "Testsystem" im Datenbanknamen

// 27.05.2014 AH Neu: "BugFix"

// 25.06.2014 AH Fix: EvtChanged terminiert, wenn doch keine Änderung passiert ist (seit neuestem wird

// ist in ALLEN Prozeduren erweitert

// 26.06.2014 AH Umsortieren in Zugriffsliste nur noch per Doppelklick

// 26.08.2014 AH Neu: "SucheEvtFocusTerm"

// 23.01.2015 AH Neue Modi für EdLists, die per Menü/Maus gespeichert werden

// 18.03.2015 AH BlueMode als SFX

// 26.03.2015 AH Initialisierung Prüft SQL-Link ab

// 30.06.2016 AH CTX-Installation (Office) deaktiviert

// 13.07.2016 AH AFX für Anhänge

// 13.09.2016 ST EvtTimer Integration Snom Tapi

// 19.01.2017 AH MoreBufs

// 04.07.2019 AH andere EvtClose-Prüfungen

// 20.09.2019 AH neue Flag für "Buffer nicht refreshen" bei App-Wechsel

// 29.10.2019 ST Bugfix Anhanganzeige

// 31.10.2019 AH Neu: Fenster Skalieren ruft Subprozedur "_Main:ScaleObject" auf

// 22.05.2020 AH Prev/NextPage über zentrale Unterfunktion

// 22.02.2021 AH "BugFix" wegen/und "EvtTerm" angepasst

// 17.03.2021 AH EvtClose von Child fokussiert nicht, sondern erst EvtTerm

// 27.04.2021 ST Betriebsmenü "Nettoverwiegung" hinzugefügt

// 27.07.2021 AH ERX

// 12.11.2021 ST Anzahl der selektierten Datensätze im Fenstertitel

// 23.02.2022 AH Doppelklick auf gesperrten Felder kopiert in die Zwischenablage

// 18.07.2023 DS RunAFX('Anh.Show'...) erst nach RecRead, damit es sicher auf dem fokussierten (und

//

// Subprozeduren

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

// SUB RefreshMode(opt aReentry : logic);

// SUB StartMdi(aName : alpha)

// SUB Action(aNewMode : Alpha; opt aPageName : alpha)

```
// SUB Mark()

// SUB AttachmentsShow()

// SUB Refresh()

// SUB EvtMdiActivate(aEvt : event) : logic

// SUB EvtClicked(aEvt : event) : logic

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

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

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

// SUB EvtMouse(aevt : event; aButton : int) : logic;

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

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

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

// SUB EvtMenuInitPopup ( aEvt : event; aMenuItem : handle ) : logic

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

// SUB EvtTapi(aEvt : event; aTapiDevice : int; aCallID : int; aCallState : int; aCallTime : caltime; aCallerID : int) : logic

// sub Suche(aSuch : alpha);

// SUB SucheEvtFocusTerm...

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

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

// SUB EvtChanged(aEvt : event) : logic

// SUB EvtInit(aEvt : event) : logic

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

// SUB EvtClose(aEvt : event) : logic

// SUB EvtTerm(aEvt : event) : logic

// SUB EvtDragInit(aEvt : event; aDataObject : int; aEffect : int; aMouseBtn : int) : logic

// SUB EvtDragTerm(aEvt : event; aDataObject : int; aEffect : int) : logic
```

//

//=====

@I:Def_Global

@I:Def_Rights

//@define Debugmode

@undef Debugmode

@ifdef Debugmode

define begin

REIN(a) : Lib_Debug:DebugModeEnter(a)

RAUS(a) : Lib_Debug:DebugModeLeave(a)

//cSchnellsuche : true

end;

@else

define begin

REIN(a) : begin end;

RAUS(a) : begin end;

//cSchnellsuche : true

end;

@Endif

define begin

clmgKeinAnhang : 185//63

clmgHatAnhang : 185+1//63+4

clmgKeineAktivitaet : 63

```

clmgHatAktivitaet : 63+4

// //   vButton->wplImageTileUser # 185 + 1

end;

//=====

//=====

declare EvtMenuCommand (aEvt : event; aMenuItem : int) : logic

//=====

//=====

/*

sub EvtFsiMonitor(

    aEvt      : event;  // Ereignis

    aAction    : int;    // Dateioperation

    aFileName   : alpha;  // Dateiname

    aFileAttrib : int;    // Dateiattribute

    aFileSize   : bigint; // Dateigröße

    aFileCT     : caltime; // Datum-/Uhrzeit der Änderung

    aFileNameOld : alpha;  // Alter Dateiname beim Umbenennen

)

: logic;

local begin

    vA : alpha;

end;

begin

    case (aAction) of

```

```

_FsiMonActionCreate : vA # aFileName + ' erzeugt';

_FsiMonActionModify : vA # aFileName + ' geändert';

_FsiMonActionDelete : vA # aFileName + ' gelöscht';

_FsiMonActionRename : vA # aFileNameOld + ' umbenannt nach ' + aFileName;

end;


$Hauptmenue->wpcaption # vA;

debug(aEvT:obj->wpname);


return(true);

end;

*/

//=====

//=====

sub BugFix(

aText      : alpha;

aObj       : int;

opt aKeinBug :logic) : int

local begin

vA  : alpha(500);

vHdl : int;

vAkt : int;

vDA  : logic;

end;

begin

```

```
if (aObj=0) then RETURN 0;
```

```
vHdl # Wininfo(aObj, _wintype);
```

```
case (vHdl) of
```

```
  _WinTypeAppframe : begin
```

```
  end;
```

```
  _WinTypeMdiFrame : begin
```

```
  end;
```

```
  _WinTypeframe    : begin // schlecht?!
```

```
    WinSearchPath(aObj); // 11.03.2021 AH laut VBS
```

```
    RETURN 0;
```

```
  end;
```

```
  _WinTypeDialog    : begin // schlecht?!
```

```
    WinSearchPath(aObj); // 11.03.2021 AH laut VBS
```

```
    RETURN 0;
```

```
  end;
```

```
otherwise begin          // Unterobjekt? -> Höher gucken...
```

```
  RETURN BugFix(aText, WinInfo(aObj, _Winframe));
```

```
end;
```

```
end;
```

```

vAkt # VarInfo(WindowBonus);

if ((aObj->wpcustom="") or (aObj->wpcustom=cnvai(vAkt))) then begin

    WinSearchPath(aObj); // 11.03.2021 AH laut VBS

    RETURN 0;

end;


vDA # HdllInfo(vAkt,_HdlExists)>0;

// vA # aText+' : realloc von '+w_name;

if (vDA) then begin

    vA # 'IST: '+w_Name+'';

    if (HdllInfo(W_Mdi, _HdlExists)>0) then vA # vA + w_Mdi->wpname;

end;

//vA # vA + aObj->wpcustom+' zu '+cnvai(VarInfo(WindowBonus));

// REPAIR:

if (HdllInfo(cnvIA(aObj->wpcustom),_HdlExists)>0) then begin

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

    vA # vA + ' SOLL: '+w_Name+'';

    if (HdllInfo(W_Mdi, _HdlExists)>0) then vA # vA + w_Mdi->wpname;

    vA # vA + ' U:'+Gusername;

//Dbalog(_LogWarning, N, 'WINDOWBONUS abweichend bei '+aObj->wpName+'! Wurde per BugFix repa

if (aKeinBug=false) then begin

    Dbalog(_LogWarning, N, 'WINDOWBONUS Prob:'+aText+'/'+aObj->wpName+' : '+vA);

end;

//debug('WINDOWBONUS Prob:'+aText+'/'+aObj->wpName+' : '+vA);

//if (aObj->wpDBRecBuf(401)<>0) then begin

```



```

// RecBufCopy(aObj->wpDbRecBuf(401),401);

//debugx('KEY401');

//end;


//debug('WINDOWBONUS Prob:'+aText+ '/' +aObj->wpName+ ' : '+vA);

//  vA # vA + ' auf '+w_name;

//debugx(vA);

end;


// if (gMdi<>0) then WinSearchPath(gMDI);

WinSearchPath(aObj);


RETURN vAkt;

end;


//=====

//=====

sub Entwicklerversion() : logic

begin

// interne Lizenzen BCS

if (dbalicense(_DbaSrvLicense)<>'CE101448MU') and

(dbalicense(_DbaSrvLicense)<>'CE101446MU') and

(dbalicense(_DbaSrvLicense)<>'CD152667MN/H') and

(dbalicense(_DbaSrvLicense)<>'CD152667MN/H') then RETURN false;

RETURN true;

```

end;

//=====

// EvtLstDataInit

//

//=====

sub EvtLstDataInit(

 aEvt : event;

 aReclD : int;

) : logic

local begin

 vMark : logic;

 vHdl : handle;

 vTmp : int;

 vFirst : logic;

end;

begin

 REIN('EvtLstDataInit');

 BugFix(__PROCFUNC__, aEvt:Obj);

//BUGFIX if (gMdi<>0) then WinSearchPath(gMDI); // 21.11.2011

// wenn die RecList durch ein anderes Fenster refrehed wurde, müssen die globlaen

// Variablen für diese Prozedur richtig gesetzt werden...

/*

```

vHdl # WinInfo(aEvt:Obj, _Winframe);

if (vHdl<>0) then begin

  if (vHdl->wpcustom<>"") and

    (vHdl->wpcustom<>cnvai(VarInfo(WindowBonus))) then begin

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

      end;

    end;

  */

vTmp # gMdi->winSearch('ed.Suche');

if (vTmp<>0) then

  if (vTmp->wpcustom='AUSWAHL') then begin

    vTmp->wpcustom # "";

    vTmp->winfocusset();

    end;

  // 06.04.2022 AH, Problem: 1. spalte fixiert, etwas markiert und Reopen

  // 08.04.2022 AH, man darf den 1. Aufruf nicht pauschal ignorieren, weil dann z.B. tempvars nicht gefüllt

  if (gZLList<>0) and ( StrCut(gZLList->wpCustom, 1, 6 ) = '_FIXED' ) then begin

    if (CnvIA( StrCut( gZLList->wpCustom, 7, 100 ) )>0) then begin

      if (gZLList->wphdrheight=0) then begin

        gZLList->wphdrheight # 1;

        vFirst # true;

        end;

      end;

    end;

  end;

```

```

////debugx('KEY401 '+aint(gFile)+'/'+aint(aRecId)+'/'+aint(aEvT:obj));

if (gFile<>0) then begin

    // Markiert?

    if (vFirst=false) and ( Lib_Mark:IstMarkiert( gFile, RecInfo( gFile, _recId ) ) ) then begin

        vMark # !Lib_GuiCom:ZLColorLine( gZLList, Set.Col.RList.Marked, true );

//    if (vMark) then begin

//        Lib_GuiCom:ZLColorLine( gZLList, Set.Col.RList.Marked, true );

//    end;

//    vMark # true;

//    Lib_GuiCom:ZLColorLine( gZLList, Set.Col.RList.Marked, true );

//debugx('KEY401 MARK '+abool(vMark));

//vMark # true;

    end

    else begin

        vMark # false;

    end;

// end;

try begin

    ErrTryIgnore(_rlocked,_rNoRec);

    ErrTryCatch(_ErrNoProcInfo,y);

    ErrTryCatch(_ErrNoSub,y);

    if (gPrefix<>") then Call(gPrefix+'_Main:EvtLstDataInit',aEvt,aRecId,vMark);

end;

end;

```

```
RAUS('EvtLstDataInit');  
  
RETURN true;  
  
end;
```

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

```
// RefreshMode
```

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

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

```
sub RefreshMode(  
    opt aReentry : logic;  
);
```

```
local begin
```

```
    Erx      : int;
```

```
    vListHdl : int;
```

```
    vOK      : logic;
```

```
    vMenuitem : int;
```

```
    vButton   : int;
```

```
    vList2Empty : logic;
```

```
    vTmp      : int;
```

```
    vA        : alpha;
```

```
    vHdl      : int;
```

```
    vDataList : logic;
```

```
// vFilter    : int;
```

```
    vSelCount : int;
```

end;

begin

 REIN('Refreshmode')

if (gMDI=0) then begin

 RAUS('RefreshMode');

 RETURN;

end;

if (gMdi<>0) then WinSearchPath(gMDI);

if (Mode='') then RETURN; // 15.09.2021 AH, wenn man im Organigram eine Aktivität startet, refreshed da

if (Mode<>c_ModeNew) and (Mode<>c_ModeNew2) and

 (Mode<>c_ModeOther) and (Mode<>c_Modeview) and

 (Mode<>c_ModeEdit) and (Mode<>c_ModeEdit2) then begin

if (HdlInfo(gZLList,_HdlExists)=0) then gZLList # 0; // 03.03.2021 AH

 // aktuellen Satz nochmals lesen

 if (gZLList<>0) and (gFile<>0) then begin

// if (gZLList->wpDbLinkFileNo<>0) then begin

// Erx # RecLink(gZLList->wpDbLinkFileNo,gZLList->wpDbFileNo,gZLList->wpdbkeyno,0);

// end;

// else begin

 if (gZLList->wpDbSelection<>0) and

 (mode<>c_ModeList2) then begin // 2022-12-07 AH wenn Dauerselektion in AufErfassung (z.B: HV

 vTmp # RecInfo(gFile, _recID);

```

    if (vTmp<>0) then begin
        Erx # RecRead(gFile,0,_recId,vTmp);
    end
    else begin
        Erx # RecRead(gFile,gZLList->wpDbSelection,0);
    end;
end
else begin
    Erx # RecRead(gFile,1,0);
end;

//    end;

    if (Erx>_multikey) then RecBufClear(gFile);
end;

end;

gMenu # gFrmMain->WinInfo(_WinMenu);

if (mode=c_ModeEdList) and (gZLList=0) then vDataList # y;

if (Mode=c_ModeList2) then
    if (RecLinkInfo($ZL.Erfassung->wpDbLinkFileNo,$ZL.Erfassung->wpDbFileNo,$ZL.Erfassung->wpdbkey)
//if (vList2Empty) then debug('Leeeeeeeeeeeeeer')

// Zugriffsliste setzen

vMenuItem # gMdi->WinSearch('NB.List');

if (vMenuItem <> 0) then

```

```

vMenuItem->wpDisabled # (
    ((Mode=c_ModeList2) or (Mode=c_ModeEdit) or (Mode=c_ModeEdit2) or (Mode=c_ModeNew) or (Mod

// Buttons & Menüs setzen

vOK # ((Mode=c_modeList2) or (Mode=c_ModeEdit) or (Mode=c_ModeEdit2) or (Mode=c_ModeNew) or
    (Mode=c_ModeEdListEdit) or
    (Mode=c_ModeEdListNew));

if (vOK) and (vList2Empty) then vOK # n;

vButton # gMdi->WinSearch('Save');

if (vButton <> 0) then vButton->wpDisabled # !(vOK)

vMenuItem # 0;

if (gMenu<>0) then begin
    vMenuItem # gMenu->WinSearch('Mnu.Save');

    if (vMenuItem <> 0) then vMenuItem->wpDisabled # !(vOK);

    vMenuItem # gMenu->WinSearch('Mnu.DL.Save');

    if (vMenuItem <> 0) then vMenuItem->wpDisabled # !(vOK);

end;

vOK # ((Mode=c_modeList2) or (Mode=c_ModeEdit) or (Mode=c_ModeEdit2) or
    (Mode=c_ModeNew) or (Mode=c_ModeView) or (Mode=c_ModeNew2) or
    (Mode=c_ModeEdListnew) or (Mode=c_ModeEdListEdit));

vButton # gMdi->WinSearch('Cancel');

if (vButton <> 0) then vButton->wpDisabled # !(vOK);

vOK # ((Mode=c_ModeEdList) or (Mode=c_modeList2) or (Mode=c_ModeEdit) or (Mode=c_ModeEdit2) or
    (Mode=c_ModeEdListnew) or (Mode=c_ModeEdListEdit));

if (gMenu<>0) then begin

```



```
vMenuItem # gMenu->WinSearch('Mnu.Cancel');  
  
if (vMenuItem <> 0) then vMenuItem->wpDisabled # !(vOK);  
  
end;
```

```
vOK ## (vDataList=false) and  
  
(w_Context<>' -INFO') and  
  
((Mode=c_ModeEdList) or (Mode=c_modeList2) or (Mode=c_ModeList) or (Mode=c_ModeView));  
  
vButton # gMdi->WinSearch('New');  
  
if (vButton <> 0) then vButton->wpDisabled # !(vOK);  
  
if (gMenu<>0) then begin  
  
    vMenuItem # gMenu->WinSearch('Mnu.New');  
  
    if (vMenuItem <> 0) then vMenuItem->wpDisabled # !(vOK);  
  
    vMenuItem # gMenu->WinSearch('Mnu.New2');  
  
    if (vMenuItem <> 0) then vMenuItem->wpDisabled # !(vOK);  
  
    vMenuItem # gMenu->WinSearch('Mnu.DL.New');  
  
    if (vMenuItem <> 0) then vMenuItem->wpDisabled # !(vOK);  
  
end;
```

```
vOK # (Mode=c_ModeList) or (Mode=c_ModeEdList);  
  
vButton # gMdi->WinSearch('Refresh');  
  
if (vButton <> 0) then vButton->wpDisabled # !(vOK);  
  
if (gMenu<>0) then begin  
  
    vMenuItem # gMenu->WinSearch('Refresh');
```

```
if (vMenuItem <> 0) then vMenuItem->wpDisabled # !(vOK);  
end;
```

```
vOK # (vDataList=false) and ((Mode=c_ModeList) or (Mode=c_ModeEdList));
```

```
vButton # gMdi->WinSearch('Mark');
```

```
if (vButton <> 0) then vButton->wpDisabled # !(vOK);
```

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

```
  vMenuItem # gMenu->WinSearch('Markierungen');
```

```
  if (vMenuItem <> 0) then vMenuItem->wpDisabled # !(vOK);
```

```
  vMenuItem # gMenu->WinSearch('Mnu.Markierungen');
```

```
  if (vMenuItem <> 0) then vMenuItem->wpDisabled # !(vOK);
```

```
  vMenuItem # gMenu->WinSearch('Mnu.Mark');
```

```
  if (vMenuItem <> 0) then vMenuItem->wpDisabled # !(vOK);
```

```
end;
```

```
vOK # (vDataList=false) and ((Mode=c_ModeEdList) or (Mode=c_ModeList) or (Mode=c_ModeView));
```

```
// if (Set.Installname='DEX') then // 02.02.2022 AH 2022-06-27 AH
```

```
// vOK # (vDataList=false) and ((Mode=c_ModeEdList) or (Mode=c_ModeList));
```

```
vButton # gMdi->WinSearch('RecPrev');
```

```
if (vButton <> 0) then vButton->wpDisabled # !(vOK);
```

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

```
  vMenuItem # gMenu->WinSearch('Mnu.RecPrev');
```

```
  if (vMenuItem <> 0) then vMenuItem->wpdisabled # !(vOK);
```

```
end;
```

```
vButton # gMdi->WinSearch('RecNext');
```

```
if (vButton <> 0) then vButton->wpDisabled # !(vOK);
```

if (gMenu<>0) then begin

 vMenuItem # gMenu->WinSearch('Mnu.RecNext');

 if (vMenuItem <> 0) then vMenuItem->wpDisabled # !(vOK);

end;

if (gMenu<>0) then begin

 vMenuItem # gMenu->WinSearch('Mnu.RecLast');

 if (vMenuItem <> 0) then vMenuItem->wpDisabled # !(vOK);

end;

if (gMenu<>0) then begin

 vMenuItem # gMenu->WinSearch('Mnu.RecFirst');

 if (vMenuItem <> 0) then vMenuItem->wpDisabled # !(vOK);

end;

vOK # y;

//if (gZLList<>0) then vOK # (gZLList->wpDbRecId<>0); // bei gefüllte RecLists ist beim Start das leider I

vOK # vOK and //(vDataList=false) and

 (w_Auswahlmode=n) and

 (w_Context<>'-INFO') and (

 (Mode=c_modeList2) or (Mode=c_ModeList) or (Mode=c_ModeView) or (Mode=c_ModeEdList));

if (vOK) and (vList2Empty) then vOK # n;

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

if (vButton <> 0) then vButton->wpDisabled # !(vOK);

if (gMenu<>0) then begin

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

 if (vMenuItem <> 0) then vMenuItem->wpDisabled # !(vOK);

```
vMenuItem # gMenu->WinSearch('Mnu.DL.Edit');  
  
if (vMenuItem <> 0) then vMenuItem->wpDisabled # !(Mode=c_ModeEdList);  
  
end;
```

```
vOK # (vDataList=false) and ((Mode=c_ModeList) or (Mode=c_ModeEdList));  
  
vButton # gMdi->WinSearch('Search');  
  
if (vButton <> 0) then vButton->wpDisabled # !(vOK);  
  
if (gMenu<>0) then begin  
  
    vMenuItem # gMenu->WinSearch('Mnu.Search');  
  
    if (vMenuItem <> 0) then vMenuItem->wpDisabled # !(vOK);  
  
end;
```

```
// Quickbar
```

```
vButton # gMdi->WinSearch('bt.Quick1');  
  
if (vButton<>0) then begin  
  
    vButton->wpdisabled # (Mode<>C_modeList) and (Mode<>c_ModeView);  
  
    vButton # gMdi->WinSearch('bt.Quick2');  
  
    vButton->wpdisabled # (Mode<>C_modeList) and (Mode<>c_ModeView);  
  
    vButton # gMdi->WinSearch('bt.Quick3');  
  
    vButton->wpdisabled # (Mode<>C_modeList) and (Mode<>c_ModeView);  
  
    vButton # gMdi->WinSearch('bt.Quick4');  
  
    vButton->wpdisabled # (Mode<>C_modeList) and (Mode<>c_ModeView);  
  
    vButton # gMdi->WinSearch('bt.Quick5');  
  
    vButton->wpdisabled # (Mode<>C_modeList) and (Mode<>c_ModeView);  
  
    vButton # gMdi->WinSearch('bt.Quick6');  
  
    vButton->wpdisabled # (Mode<>C_modeList) and (Mode<>c_ModeView);  
  
end;
```

```

vButton # gMdi->WinSearch('bt.Quick7');

vButton->wpdisabled # (Mode<>C_modeList) and (Mode<>c_ModeView);

vButton # gMdi->WinSearch('bt.Quick8');

vButton->wpdisabled # (Mode<>C_modeList) and (Mode<>c_ModeView);

vButton # gMdi->WinSearch('bt.Quick9');

vButton->wpdisabled # (Mode<>C_modeList) and (Mode<>c_ModeView);

vButton # gMdi->WinSearch('bt.Quick10');

vButton->wpdisabled # (Mode<>C_modeList) and (Mode<>c_ModeView);

end;

```

```

// vOK # (vDataList=false) and

```

```

vOK # (((Mode=c_ModeList) and (w_Auswahlmode=n)) or (Mode=c_ModeList2) or (Mode=c_ModeEdList

```

```

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

```

```

if (vButton <> 0) then vButton->wpDisabled # !(vOK);

```

```

if (gMenu<>0) then begin

```

```

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

```

```

    if (vMenuItem <> 0) then vMenuItem->wpDisabled # !(vOK);

```

```

end;

```

```

// ST 2012-03-20 START Projekt: 1326/192

```

```

vOK # (vDataList=false) and ((Mode=c_ModeEdList) or (Mode=c_ModeList) or (Mode=c_ModeView));

```

```

vButton # gMdi->WinSearch('Attachment');

```

```

if (vButton <> 0) then vButton->wpDisabled # !(vOK);

```

```

if (vButton <> 0) AND (vButton->wpDisabled = false) then begin

```

```

// 21.07.2016:

if (RunAFX('Anh.Check',aint(gFile))>=0) then begin

    Erx # Anh_Data:Check(gFile);

end

else begin

    Erx # AfxRes;

end;

if (Erx < _rNoKey) then

    vButton->wplImageTileUser # clmgHatAnhang; // 185+1

else

    vButton->wplImageTileUser # clmgKeinAnhang; // 185;

// vFilter->RecFilterDestroy();

end;

// ST 2012-03-20 ENDE Projekt: 1326/192


// 29.03.2020 AH:

vButton # gMdi->WinSearch('Aktivitaeten');

if (vButton <> 0) then vButton->wpDisabled # !(vOK);

if (vButton <> 0) AND (vButton->wpDisabled = false) then begin

    Erx # Tem_Data:Check(gFile);

    if (Erx=-100) then vButton->wpVisible # false;

    if (Erx < _rNoKey) then

        vButton->wplImageTileUser # clmgHatAktivitaet

    else

        vButton->wplImageTileUser # clmgKeineAktivitaet;

```

end;

// Menüleiste setzen

if (gMenu<>0) then begin

 vMenuItem # gMenu->WinSearch('Ansicht');

 if (vMenuItem <> 0) then vMenuItem->wpDisabled # vList2Empty;

 vMenuItem # gMenu->WinSearch('Mnu.Erfassung');

 if (vMenuItem <> 0) then vMenuItem->wpDisabled # (Mode<>c_ModeList2);

end;

vOk # ((Mode=c_ModeEdit) or (Mode=c_ModeEdit2) or (Mode=c_ModeNew) or (Mode=c_ModeNew2));

if (gMenu<>0) then begin

 vMenuItem # gMenu->WinSearch('Mnu.Auswahl');

 if (vOK=false) and (vMenuItem <> 0) then vMenuItem->wpDisabled # !(vOK);

end;

vOk # ((Mode=c_ModeEdit) or (Mode=c_ModeEdit2) or (Mode=c_ModeNew) or (Mode=c_ModeNew2) or

if (gMenu<>0) then begin

 vMenuItem # gMenu->WinSearch('Mnu.NextPage');

 if (vMenuItem <> 0) then vMenuItem->wpDisabled # !(vOK);

end;

if (gMenu<>0) then begin

 vMenuItem # gMenu->WinSearch('Mnu.PrevPage');

 if (vMenuItem <> 0) then vMenuItem->wpDisabled # !(vOK);

end;

```
vOK # ((Mode=c_ModeList) or (Mode=c_ModeView) or (Mode=c_ModeEdList));
```

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

```
  vMenuItem # gMenu->WinSearch('Info');
```

```
  if (vMenuItem <> 0) then vMenuItem->wpDisabled # !(vOK);
```

```
end;
```

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

```
  vMenuItem # gMenu->WinSearch('Mnu.Info');
```

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

```
    vMenuItem->wpDisabled # !(vOK);
```

```
  end;
```

```
end;
```

```
// 22.08.2012 AI : Alles im Menü Ansicht ausblenden, was NICHT Standard ist
```

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

```
  vMenuItem # gMenu->WinSearch('Ansicht');
```

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

```
    FOR vHdl # vMenuItem->WinInfo( _winFirst,0 ,_WinTypeMenuItem)
```

```
    LOOP vHdl # vHdl->WinInfo( _winNext,0 ,_WinTypeMenuItem)
```

```
    WHILE ( vHdl > 0 ) DO BEGIN
```

```
      // 12.09.2012 ST:  'Mnu.SelAuswahl' hinzugefügt damit bei Selektionsdialogen auch direkt nach EvtA
```

```
      // Projekt 1326/292  die Selektion möglich ist
```

```
      case vHdl->wpname of
```

```
        'separator','Mnu.Auswahl','Mnu.NextPage','Mnu.PrevPage','Mnu.SelAuswahl' : begin end;
```

```
        otherwise vHdl->wpDisabled # !(vOK);
```

```
      end;
```


END;

end;

end;

if (gMenu<>0) then begin

 vMenuItem # gMenu->WinSearch('Mnu.Filter');

 if (vMenuItem <> 0) then vMenuItem->wpDisabled # !(vOK);

 vMenuItem # gMenu->WinSearch('Filter');

 if (vMenuItem <> 0) then vMenuItem->wpDisabled # !(vOK);

end;

if (gMenu<>0) then begin

 vMenuItem # gMenu->WinSearch('Listen');

 if (vMenuItem <> 0) then vMenuItem->wpDisabled # !(vOK);

end;

if (gMenu<>0) then begin

 vMenuItem # gMenu->WinSearch('Druck');

 if (vMenuItem <> 0) then vMenuItem->wpDisabled # !(vOK);

end;

if (gMenu<>0) then begin

 vMenuItem # gMenu->WinSearch('Extras');

 if (vMenuItem <> 0) then vMenuItem->wpDisabled # !(vOK);

end;

if (gMenu<>0) then begin

 vMenuItem # gMenu->WinSearch('Mnu.Extras');

 if (vMenuItem <> 0) then vMenuItem->wpDisabled # !(vOK);

end;

if (gMenu<>0) then begin

 vMenuItem # gMenu->WinSearch('Mnu.Markierungen');

 if (vMenuItem <> 0) then vMenuItem->wpDisabled # !(vOK);

end;

if (gMenu<>0) then begin

 vMenuItem # gMenu->WinSearch('Mnu.Mark.SetField');

 if (vMenuItem <> 0) then vMenuItem->wpDisabled # (Rechte[Rgt_SerienEdit] = false);

end;

if (gMenu<>0) then begin

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

 if (vMenuItem <> 0) then vMenuItem->wpDisabled # y;

end;

if (gMenu<>0) then begin

 vMenuItem # gMenu->WinSearch('Mnu.Excel.Export');

 if (vMenuItem <> 0) then vMenuItem->wpDisabled # y;

end;

if (gMenu<>0) then begin

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

 if (vMenuItem <> 0) then vMenuItem->wpDisabled # y;

end;

if (gMenu<>0) then begin

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

 if (vMenuItem <> 0) then vMenuItem->wpDisabled # y;

end;

```
if (gMenu<>0) then begin
    vMenuItem # gMenu->WinSearch('SFX');
    if (vMenuItem <> 0) then vMenuItem->wpDisabled # !(vOK);
end;
```

```
if (gZLList<>0) and (gMenu<>0) then begin
    if (gZLList->wpdbselection<>0) then begin
        if (StrFind(w_selName, '.SEL', 0) <> 0) then vA # 'SEL'
        else if (StrFind(w_selName, '.MARK', 0) <> 0) then vA # 'MARK';
    end;
    vHdl # gMenu->WinSearch('Mnu.Filter.Stop');
    if (vHdl <> 0) then vHdl->wpDisabled # (vA='MARK');
    vHdl # gMenu->WinSearch('Mnu.Filter.Start');
    if (vHdl <> 0) then vHdl->wpMenucheck # (vA='SEL');
    if (vHdl <> 0) then vHdl->wpDisabled # (vA='MARK');
    vHdl # gMenu->WinSearch('Mnu.Mark.Filter');
    if (vHdl <> 0) then vHdl->wpDisabled # (vA='SEL');
end;
```

```
if (Mode=c_ModeOther) then begin
    if ($NB.Main<>0) then
        $NB.Main->wpDisabled # True;
    if (gZLList<>0) then begin
        gZLList->wpDisabled # True;
```

```

end;

end

else begin

    if ($NB.Main<>0) then

        $NB.Main->wpDisabled # false;

    if (gZLList<>0) then begin

        gZLList->wpDisabled # false;

    end;

end;

end;


// Überschriften setzen

vTmp # gMDI->winsearch('NB.Main');

if (vTmp<>0) and (StrCut(gMDI->wpcaption,1,1)<>' ') then begin

    case Mode of

        c_ModeList, c_ModeEdList: begin

            if (w_Auswahlmode) then

                gMdi->wpcaption # gTitle+' '+Translate('Auswahl')

            else

                gMdi->wpcaption # gTitle+' '+Translate('Übersicht');

            if (gZLList<>0) then begin

                if (gZLList->wpDbSelection<>0) then begin

//          gMdi->wpcaption # gMdi->wpcaption + ' ('+ Translate('selektiert')+');

// ST 2021-11-11: Anzeige der Selektierten Datensätze

                if (gFile > 0) then

                    vSelCount # RecInfo(gFile,_RecCount,gZLList->wpDbSelection);

                    gMdi->wpcaption # gMdi->wpcaption + ' ('+ Translate('selektiert')+ ': '+Aint(vSelCount)+)';

```

```

    end;

end;

end;

c_ModeView: gMdi->wpcaption # gTitle+' '+Translate('Ansicht');
c_ModeEdit: gMdi->wpcaption # gTitle+' '+Translate('bearbeiten');
c_ModeNew: gMdi->wpcaption # gTitle+' '+Translate('neu anlegen');

end;

gMdi->Winupdate(_Winupdon);

if (gPrefix<>") then Call(gPrefix+'_Main:RefreshMode',aReEntry);

end;


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

if (vButton<>0) then begin

    if (vButton->wpdisabled) and (Mode=c_ModeView) then begin
//    vButton->wplImageTile # _WinImgNone;

//21.07.2016    vButton->wpvisible # false;


vButton # gMdi->WinSearch('EditErsatz');

if (vButton<>0) then begin

    vButton->wpdisabled # false;

    vButton->wpReturnKeyClick # true;

end;

end

else begin

//    vButton->wplImageTile # _WinImgEdit;

```

```
//21.07.2016    vButton->wpvisible    # true;
```

```
    vButton # gMdi->WinSearch('EditErsatz');
```

```
    if (vButton<>0) then vButton->wpdisabled # true;
```

```
end;
```

```
end;
```

```
// gesamte Fenster neu zeichnen
```

```
if (gprefix<>'Sel') then gMdi->winUpdate();
```

```
RAUS('RefreshMode');
```

```
end;
```

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

```
// StartMDI
```

```
//      MDI-Fenster starten bzw. reaktivieren
```

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

```
sub StartMdi(
```

```
    aName : alpha;
```

```
)
```

```
local begin
```

```
    vHdlMenu : int;
```

```
    vHdl      : int;
```

```
    vHdlNode : int;
```

```
    vRecFlag : int;
```

```

vSelName : alpha;

vSelData : int;

vSelErg : int;

defaultnode : alpha;

ok      : logic;

aTemp   : alpha;

vQ      : alpha(4000);

vQ2     : alpha(4000);

tErg    : int;

vSel     : alpha;

vName    : alpha;

vFont    : font;

end;

begin

```

```

// Organigramm

```

```

Org_Data:KeepAlive();

```

```

case aName of

```

```

  'AppFrame' : begin

```

```

    // wo ist bisher der Focus?

```

```

    vHdl # WinfocusGet();

```

```

    if (vHdl<>0) then begin

```

```

        vHdl # WinInfo(vHdl, _WinFrame);

```

```

    end;

```

```

StartMdi( 'Hauptmenue' );

StartMdi( 'Workbench' );

if ( Set.TimerYN ) then

    StartMdi( 'Notifier' );


// ggf. Fenster wieder aktivieren

if (vhdl<>0) then vHdl->WinUpdate(_WinUpdActivate);

end;

```

```

'Hauptmenue': begin

```

```

    if ( gMdiMenu = 0 ) then begin

```

```

        gMdiMenu # WinAddByName( gFrmMain, Lib_GuiCom:GetAlternativeName( 'Mdi.Hauptmenue' ), _w

```

```

        gMdiMenu->WinUpdate( _winUpdOn );

```

```

        App_Main_Sub:RebuildFavoriten();

```

```

        // Usersettings holen

```

```

        Lib_GuiCom:RecallWindow(gMdiMenu);

```

```

    if (Set.Org.IdleInterval=0) then begin

```

```

        vHDL # Winsearch(gMdiMenu,'gt.Organigramm');

```

```

        vHDL->wpvisible # false;

```

```

        vHDL # Winsearch(gMdiMenu,'gt.Mainmenue');

```

```

        vHDL->wpFlagsTitlebar # 0;

```

```

    end

```

```

    else begin

```



```
vHDL # Winsearch(gMdiMenu,'tv.Organigramm');  
vHDL->wpColFocusBkg # Set.Col.RList.Cursor;  
vHDL->wpColFocusOffBkg # "Set.Col.RList.CurOff";
```

```
if (Usr.Font.Size<>0) then begin
```

```
    vFont # vHDL->wpfont;
```

```
    vFont:Size # Usr.Font.Size * 10;
```

```
    vHDL->wpfont # vFont;
```

```
end;
```

```
end;
```

```
end
```

```
else
```

```
    WinUpdate( gMDIMenu, _winUpdActivate);
```

```
end;
```

```
'Workbench': begin
```

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

```
        gMdiWorkbench # WinAddByName(gFrmMain, Lib_GuiCom:GetAlternativeName('Mdi.Workbench'), _
```

```
        gMdiWorkbench->WinUpdate(_WinUpdOn);
```

```
        // Usersettings holen
```

```
        Lib_GuiCom:RecallWindow(gMdiWorkbench);
```

```
    end
```

```
    else begin
```

```
//        gMdiWorkbench->WinFocusSet(true);
```

```
        WinUpdate(gMDIWorkbench, _winupdactivate);
```

end;

end;

'Notifier': begin

Lib_Notifier:Start();

end;

'Mdi.Dashboard' : begin

if (gMdiDashboard=0) then begin

gMdiDashboard # WinAddByName(gFrmMain, Lib_GuiCom:GetAlternativeName('Mdi.Dashboard2'),

gMdiDashboard->WinUpdate(_WinUpdOn);

end

else begin

WinUpdate(gMDIDashboard, _winupdactivate);

end;

end;

'Translate.Verwaltung' : begin

if (gMdiPara = 0) then begin

//gFrmMain->wpDisabled # true;

gMdiPara # Lib_GuiCom:OpenMdi(gFrmMain, 'Prg.Trn.Verwaltung', _WinAddHidden);

gMdiPara->WinUpdate(_WinUpdOn);

//gFrmMain->wpDisabled # false;

```

$NB.Main->WinFocusSet(true);

end

else begin

    Lib_guiCom:ReOpenMDI(gMdiPara);

end;

end;

'Auf.P.Verwaltung','Auf.P.Ablage' :

if Rechte[Rgt_Auftrag] then begin

    if (gMdiAuf = 0) then begin

        //gFrmMain->wpDisabled # true;

        gMdiAuf # Lib_GuiCom:OpenMdi(gFrmMain, aName, _WinAddHidden);

        gMdiAuf->WinUpdate(_WinUpdOn);

        //gFrmMain->wpDisabled # false;

        $NB.Main->WinFocusSet(true);

    end

    else begin

        Lib_guiCom:ReOpenMDI(gMDIAuf);

    end;

end;

'Auf.Verwaltung' :

if Rechte[Rgt_Auftrag] then begin

    if (gMdiAuf = 0) then begin

```

```

//gFrmMain->wpDisabled # true;

gMdiAuf # Lib_GuiCom:OpenMdi(gFrmMain, aName, _WinAddHidden);

gMdiAuf->WinUpdate(_WinUpdOn);

//gFrmMain->wpDisabled # false;

$NB.Main->WinFocusSet(true);

end

else begin

    Lib_guiCom:ReOpenMDI(gMDIAuf);

end;

end;

```

```

'GPI.Verwaltung' : begin

if (Rechte[Rgt_Grobplanung]) then begin

    if (gMdiGantt = 0) then begin

        //gFrmMain->wpDisabled # true;

        gMdiGantt # Lib_GuiCom:OpenMdi(gFrmMain, aName, _WinAddHidden);

        gMdiGantt->WinUpdate(_WinUpdOn);

        //gFrmMain->wpDisabled # false;

        //$NB.Main->WinFocusSet(true);

        SetFocus($NB.Main, true);

    end

    else begin

        Lib_guiCom:ReOpenMDI(gMDIGantt);

    end;

end;

end;

```

end;

'BAG.Planung' : begin

if (Rechte[Rgt_BAG_Planung]) then begin

if (gMdiBAG = 0) then begin

BA1_P_Plan_Main:Start();

end

else begin

Lib_guiCom:ReOpenMDI(gMDIBAG);

end;

end;

end;

'RSO.Planung' : begin

if (Rechte[Rgt_BAG_Planung]) then begin

if (gMdiRso = 0) then begin

//gFrmMain->wpDisabled # true;

gMdiRSO # Lib_GuiCom:OpenMdi(gFrmMain, 'Rso.Bel.Verwaltung_Neu', _WinAddHidden);

gMdiRSO->WinUpdate(_WinUpdOn);

//gFrmMain->wpDisabled # false;

end

else begin

Lib_guiCom:ReOpenMDI(gMDIRso);

end;

end;

end;

'Pak.Verwaltung' :

if (Rechte[Rgt_Pakete]) then begin

if (gMdiPrj = 0) then begin

//gFrmMain->wpDisabled # true;

gMdiPrj # Lib_GuiCom:OpenMdi(gFrmMain, aName, _WinAddHidden);

gMdiPrj->WinUpdate(_WinUpdOn);

//gFrmMain->wpDisabled # false;

\$NB.Main->WinFocusSet(true);

end

else begin

Lib_guiCom:ReOpenMDI(gMDIPrj);

end;

end;

'BAG.Verwaltung' : begin

if (Rechte[Rgt_BAG]) then begin

if (gMdiBAG = 0) then begin

gMdiBAG # Lib_GuiCom:OpenMdi(gFrmMain, 'BA1.Verwaltung', _WinAddHidden);

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

Filter_BAG # y;

vQ # ";

```

Lib_Sel:QAlpha( var vQ, '"BAG.Löschmarker"', '=', '');
Lib_Sel:QRecList(0,vQ);
gMdiBAG->WinUpdate(_WinUpdOn);
$NB.Main->WinFocusSet(true);

end

else begin

    Lib_guiCom:ReOpenMDI(gMDIBAG);

end;

end;

end;

```

```

'BAG.Output' : begin

if (Rechte[Rgt_BAG]) then begin

    if (gMdiBAG = 0) then begin

        //gFrmMain->wpDisabled # true;

        gMdiBAG # Lib_GuiCom:OpenMdi(gFrmMain, 'BA1.Output', _WinAddHidden);
//      gMdiBAG # Lib_GuiCom:OpenMdi(gFrmMain, 'DiB.Verwaltung', _WinAddHidden);

        gMdiBAG->WinUpdate(_WinUpdOn);

        //gFrmMain->wpDisabled # false;

        $NB.Main->WinFocusSet(true);

    end

    else begin

        Lib_guiCom:ReOpenMDI(gMDIBAG);

    end;

end;

end;

```

end;

'Adr.Verwaltung' :

Adr_Main:Start();

/*

if (Rechte[Rgt_Adressen]) then begin

if (gMdiAdr = 0) then begin

//gFrmMain->wpDisabled # true;

gMdiAdr # Lib_GuiCom:OpenMdi(gFrmMain, 'Adr.Verwaltung', _WinAddHidden);

gMdiAdr->WinUpdate(_WinUpdOn);

//gFrmMain->wpDisabled # false;

\$NB.Main->WinFocusSet(true);

end

else begin

Lib_guiCom:ReOpenMDI(gMDIAdr);

end;

end;

*/

'Adr.P.Verwaltung' :

Adr_P_Main:Start();

/**

if (Rechte[Rgt_Adr_Ansprechpartner]) then begin

if (gMdiAdr = 0) then begin


```

//gFrmMain->wpDisabled # true;

gMdiAdr # Lib_GuiCom:OpenMdi(gFrmMain, aName, _WinAddHidden);

gMdiAdr->WinUpdate(_WinUpdOn);

//gFrmMain->wpDisabled # false;

$NB.Main->WinFocusSet(true);

end

else begin

    Lib_guiCom:ReOpenMDI(gMDIAdr);

end;

end;

**/

```

'Prj.Verwaltung' :

```

if (Rechte[Rgt_Projekte]) then begin

    if (gMDIPrj = 0) then begin

        //gFrmMain->wpDisabled # true;

        gMDIPrj # Lib_GuiCom:OpenMdi(gFrmMain, aName, _WinAddHidden);

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

        vQ # "";

        Lib_Sel:QAlpha( var vQ, "Prj.Löschmarker", '=', "");

        Lib_Sel:QRecList(0,vQ);

        gMDIPrj->WinUpdate(_WinUpdOn);

        //gFrmMain->wpDisabled # false;

        $NB.Main->WinFocusSet(true);

    end

    else begin

```

```
Lib_guiCom:ReOpenMDI(gMDIPrj);
```

```
end;
```

```
end;
```

```
'Mat.Verwaltung', 'Mat.Ablage' :
```

```
if (Rechte[Rgt_Material]) then begin
```

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

```
//gFrmMain->wpDisabled # true;
```

```
gMdiMat # Lib_GuiCom:OpenMdi(gFrmMain, aName, _WinAddHidden);
```

```
gMdiMat->WinUpdate(_WinUpdOn);
```

```
//gFrmMain->wpDisabled # false;
```

```
//      "GV.Fil.Mat.gelöscht" # y;
```

```
$NB.Main->WinFocusSet(true);
```

```
end
```

```
else begin
```

```
Lib_guiCom:ReOpenMDI(gMDIMat);
```

```
end;
```

```
end;
```

```
'Msl.Verwaltung' :
```

```
if (Rechte[Rgt_Materialstruktur]) then begin
```

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

```
//gFrmMain->wpDisabled # true;
```

```
gMdiMsl # Lib_GuiCom:OpenMdi(gFrmMain, aName, _WinAddHidden);
```

```

gMdiMsl->WinUpdate(_WinUpdOn);

//gFrmMain->wpDisabled # false;

$NB.Main->WinFocusSet(true);

end

else begin

    Lib_guiCom:ReOpenMDI(gMDIMsl);

end;

end;

```

'Lys.K.Verwaltung' :

```

if (Rechte[Rgt_Materialanalyse]) then begin

    if (Set.LyseErweitertYN) then

        aName # 'Lys.K.Verwaltung2';

    if (gMdiMsl = 0) then begin

        //gFrmMain->wpDisabled # true;

        RecBufClear(832);

        RecBufClear(833);

        gMdiMsl # Lib_GuiCom:OpenMdi(gFrmMain, aName, _WinAddHidden);

        gMdiMsl->WinUpdate(_WinUpdOn);

        //gFrmMain->wpDisabled # false;

        $NB.Main->WinFocusSet(true);

    end

    else begin

        Lib_guiCom:ReOpenMDI(gMdiMsl);

    end;

end;

```

end;

'Art.Verwaltung' :

if (gMdiArt = 0) then begin

gMdiArt # Lib_GuiCom:OpenMdi(gFrmMain, aName, _WinAddHidden);

gMdiArt->WinUpdate(_WinUpdOn);

// \$NB.Main->WinFocusSet(true);

end

else begin

Lib_guiCom:ReOpenMDI(gMdiArt);

end;

/*

if (Rechte[Rgt_Artikel]) then begin

if (gMdiArt = 0) then begin

gMdiArt # Lib_GuiCom:OpenMdi(gFrmMain, aName, _WinAddHidden);

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

Filter_Art # y;

gMdiArt->WinUpdate(_WinUpdOn);

\$NB.Main->WinFocusSet(true);

end

else begin

Lib_guiCom:ReOpenMDI(gMdiArt);

end;

end;

*/

'Art.SLK.Verwaltung' :

if (Rechte[Rgt_Art_SL]) then begin

if (gMdiMsl = 0) then begin

//gFrmMain->wpDisabled # true;

gMdiMsl # Lib_GuiCom:OpenMdi(gFrmMain, aName, _WinAddHidden);

gMdiMsl->WinUpdate(_WinUpdOn);

//gFrmMain->wpDisabled # false;

\$NB.Main->WinFocusSet(true);

end

else begin

Lib_guiCom:ReOpenMDI(gMDIMsl);

end;

end;

'Art.P.Verwaltung' :

if (Rechte[Rgt_Art_Preise]) then begin

if (gMdiArt = 0) then begin

gMdiArt # Lib_GuiCom:OpenMdi(gFrmMain, aName, _WinAddHidden);

// VarInstance(WindowBonus,cnvIA(gMDIArt->wpcustom));

// vQ # ";

// Lib_Sel:QAlpha(var vQ, "SWe.Löschmarker", '=', "');

// Lib_Sel:QRecList(0,vQ);

```

    gMdiArt->WinUpdate(_WinUpdOn);

end

else begin

    Lib_guiCom:ReOpenMDI(gMdiArt);

end;

end;

'Gantt.TeM.Woche' :

if (Rechte[Rgt_Termine]) then begin

    if (gMdiGantt = 0) then begin

        //gFrmMain->wpDisabled # true;

        gMdiGantt # Lib_GuiCom:OpenMdi(gFrmMain, aName, _WinAddHidden);

        gMdiGantt->WinUpdate(_WinUpdOn);

        //gFrmMain->wpDisabled # false;

//    $NB.Main->WinFocusSet(true);

    end

    else begin

        Lib_guiCom:ReOpenMDI(gMdiGantt);

    end;

end;

// --

'MathCalculator' : begin // gMdiMathCalculator

    if (Rechte[Rgt_Formel]) then begin

        if (gMdiMathCalculator = 0) then begin

```

```

//gFrmMain->wpDisabled # true;

gMdiMathCalculator # Lib_GuiCom:OpenMdi(gFrmMain, aName, _WinAddHidden);

gMdiMathCalculator->WinUpdate(_WinUpdOn);

//gFrmMain->wpDisabled # false;

$NB.Main->WinFocusSet(true);

end

else begin

    Lib_guiCom:ReOpenMDI(gMdiMathCalculator);

end;

end;

end

```

'Math.Verwaltung' :

```

if (Rechte[Rgt_Formel]) then begin

    if (gMdimath = 0) then begin

        //gFrmMain->wpDisabled # true;

        gMdiMath # Lib_GuiCom:OpenMdi(gFrmMain, aName, _WinAddHidden);

        gMdiMath->WinUpdate(_WinUpdOn);

        //gFrmMain->wpDisabled # false;

        $NB.Main->WinFocusSet(true);

    end

    else begin

        Lib_guiCom:ReOpenMDI(gMdiMath);

    end;

end;

end;

```

'MathVar.Verwaltung' :

if (Rechte[Rgt_Fvariable]) then begin

if (gMdimathVar = 0) then begin

//gFrmMain->wpDisabled # true;

gMdiMathVar # Lib_GuiCom:OpenMdi(gFrmMain, aName, _WinAddHidden);

gMdiMathVar->WinUpdate(_WinUpdOn);

//gFrmMain->wpDisabled # false;

\$NB.Main->WinFocusSet(true);

end

else begin

Lib_guiCom:ReOpenMDI(gMdiMathVar);

end;

end;

'Math.Alphabet.Verwaltung' : begin

if (Rechte[Rgt_MathAlpabet]) then begin

if (gMdiMathAlphabet = 0) then begin

//gFrmMain->wpDisabled # true;

gMdiMathAlphabet # Lib_GuiCom:OpenMdi(gFrmMain, aName, _WinAddHidden);

gMdiMathAlphabet->WinUpdate(_WinUpdOn);

//gFrmMain->wpDisabled # false;

\$NB.Main->WinFocusSet(true);

end


```

else begin

    Lib_guiCom:ReOpenMDI(gMdiMathAlphabet);

end;

end;

end;

// Miniprogramme für Variablen im Mathematikssystem

'Math.Alphabet.MiniPrg' : begin

    if ( gMdiMathVarMiniPrg = 0) then begin

        //gFrmMain->wpDisabled # true;

        gMdiMathVarMiniPrg # Lib_GuiCom:OpenMdi(gFrmMain, aName, _WinAddHidden);

        gMdiMathVarMiniPrg->WinUpdate(_WinUpdOn);

        //gFrmMain->wpDisabled # false;

        $NB.Main->WinFocusSet(true);

    end

    else begin

        Lib_guiCom:ReOpenMDI(gMdiMathVarMiniPrg);

    end;

end;

```

```

'TeM.Verwaltung.Termine' :

    if Rechte[Rgt_Termine_Berichte] then begin

        if (gMdiPara = 0) then begin

            //gFrmMain->wpDisabled # true;

```

```

gMdiPara # Lib_GuiCom:OpenMdi(gFrmMain, 'TeM.Verwaltung', _WinAddHidden);

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

//    Usr.Username # gUserName;


// ehemals Selektion 980 FOR_USER_SELF

vQ # ' (LinkCount(Anker) > 0 OR TeM.Anlage.User = '"+gUserName+"')';

Lib_Sel:QDate( var vQ, 'TeM.Erledigt.Datum', '=', 0.0.0);

vQ2 # ";

Lib_Sel:QInt( var vQ2, 'TeM.A.Datei', '=', 800);

Lib_Sel:QEnthaelteA( var vQ2, 'TeM.A.Code', gUserName );


// Selektion aufbauen...

vHdl # SelCreate(980, gZLList->wpdbkeyno);

vHdl->SelAddLink("", 981, 980, 1, 'Anker');

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

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

tErg # vHdl->SelDefQuery('Anker', 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;


gMdiPara->WinUpdate(_WinUpdOn);

//gFrmMain->wpDisabled # false;

```

```

$NB.Main->WinFocusSet(true);

end

else begin

    Lib_guiCom:ReOpenMDI(gMdiPara);

end;

end;

'Ein.P.Verwaltung','Ein.P.Ablage' :

if Rechte[Rgt_Einkauf] then begin

    if (gMdiEin = 0) then begin

        //gFrmMain->wpDisabled # true;

        gMdiEin # Lib_GuiCom:OpenMdi(gFrmMain, aName, _WinAddHidden);

        gMdiEin->WinUpdate(_WinUpdOn);

        //gFrmMain->wpDisabled # false;

        $NB.Main->WinFocusSet(true);

    end

    else begin

        Lib_guiCom:ReOpenMDI(gMdiEin);

    end;

end;

'SWe.Verwaltung' :

if Rechte[Rgt_SammelWE] then begin

    if (gMdiEin = 0) then begin

```

```

gMdiEin # Lib_GuiCom:OpenMdi(gFrmMain, aName, _WinAddHidden);

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

Filter_SWE # y;

vQ # "";

Lib_Sel:QAlpha( var vQ, "SWe.Löschmarker", '=', '');

Lib_Sel:QRecList(0,vQ);

gMdiEin->WinUpdate(_WinUpdOn);

$NB.Main->WinFocusSet(true);

end

else begin

    Lib_guiCom:ReOpenMDI(gMdiEin);

end;

end;

```

'Rso.Verwaltung' :

```

if (Rechte[Rgt_Ressourcen]) then begin

    if (gMdiRso = 0) then begin

        //gFrmMain->wpDisabled # true;

        gMdiRso # Lib_GuiCom:OpenMdi(gFrmMain, aName, _WinAddHidden);

        gMdiRso->WinUpdate(_WinUpdOn);

        //gFrmMain->wpDisabled # false;

        $NB.Main->WinFocusSet(true);

    end

    else begin

        Lib_guiCom:ReOpenMDI(gMdiRso);
    end
end

```

end;

end;

'HuB.EK' :

if Rechte[Rgt_HuB_Einkauf] then begin

if (gMdiEin = 0) then begin

//gFrmMain->wpDisabled # true;

gMdiEin # Lib_GuiCom:OpenMdi(gFrmMain, aName, _WinAddHidden);

gMdiEin->WinUpdate(_WinUpdOn);

//gFrmMain->wpDisabled # false;

\$NB.Main->WinFocusSet(true);

end

else begin

Lib_guiCom:ReOpenMDI(gMdiEin);

end;

end;

'Bdf.Verwaltung' :

if Rechte[Rgt_Bedarf] then begin

if (gMdiBdf = 0) then begin

//gFrmMain->wpDisabled # true;

gMdiBdf # Lib_GuiCom:OpenMdi(gFrmMain, aName, _WinAddHidden);

gMdiBdf->WinUpdate(_WinUpdOn);

//gFrmMain->wpDisabled # false;

```
$NB.Main->WinFocusSet(true);
```

```
end
```

```
else begin
```

```
Lib_guiCom:ReOpenMDI(gMdiBdf);
```

```
end;
```

```
end;
```

```
'Bdf.Ablage' : // 2023-05-16 AH
```

```
if Rechte[Rgt_Bedarf] then begin
```

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

```
gMdiBdf # Lib_GuiCom:OpenMdi(gFrmMain, aName, _WinAddHidden);
```

```
gMdiBdf->WinUpdate(_WinUpdOn);
```

```
$NB.Main->WinFocusSet(true);
```

```
end
```

```
else begin
```

```
Lib_guiCom:ReOpenMDI(gMdiBdf);
```

```
end;
```

```
end;
```

```
'Lfs.Verwaltung' :
```

```
if Rechte[Rgt_Lieferschein] then begin
```

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

```
gMdiLFS # Lib_GuiCom:OpenMdi(gFrmMain, aName, _WinAddHidden);
```

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

Filter_VSD # y;

vQ # ";

Lib_Sel:QDate(var vQ, 'Lfs.Datum.Verbucht', '=', 0.0.0);

Lib_Sel:QAlpha(var vQ, '"Lfs.Löschmarker"', '=', "');

Lib_Sel:QRecList(0,vQ);

gMdiLFS->WinUpdate(_WinUpdOn);

\$NB.Main->WinFocusSet(true);

end

else begin

Lib_guiCom:ReOpenMDI(gMdiLFS);

end;

end;

'Vsd.Verwaltung' :

if Rechte[Rgt_Versand] then begin

if (gMdiLFS = 0) then begin

gMdiLFS # Lib_GuiCom:OpenMdi(gFrmMain, aName, _WinAddHidden);

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

Filter_VSD # y;

vQ # ";

Lib_Sel:QAlpha(var vQ, '"Vsd.Löschmarker"', '=', "');

Lib_Sel:QRecList(0,vQ);

gMdiLFS->WinUpdate(_WinUpdOn);

\$NB.Main->WinFocusSet(true);

end

else begin

Lib_guiCom:ReOpenMDI(gMdiLFS);

end;

end;

'VsP.Verwaltung' :

if Rechte[Rgt_Versandpool] then begin

if (gMdiVSP = 0) then begin

//gFrmMain->wpDisabled # true;

gMdiVSP # Lib_GuiCom:OpenMdi(gFrmMain, aName, _WinAddHidden);

gMdiVSP->WinUpdate(_WinUpdOn);

//gFrmMain->wpDisabled # false;

\$NB.Main->WinFocusSet(true);

end

else begin

Lib_guiCom:ReOpenMDI(gMdiVSP);

end;

end;

'Rek.P.Verwaltung' :

if Rechte[Rgt_Rek_Positionen] then begin

if (gMdiQS = 0) then begin

gMdiQS # Lib_GuiCom:OpenMdi(gFrmMain, aName, _WinAddHidden);

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


```

Filter_REK # y;

vQ # "";

Lib_Sel:QAlpha( var vQ, "Rek.P.Löschmarker", '=', "");

Lib_Sel:QRecList(0,vQ);

gMdiQS->WinUpdate(_WinUpdOn);

$NB.Main->WinFocusSet(true);

end

else begin

    Lib_guiCom:ReOpenMDI(gMdiQS);

end;

end;

```

'LfE.Verwaltung' :

```

if (StrFind(Set.Module,'L',0)<>0) and
    (Rechte[Rgt_LfErklaerungen]) then begin

    LfE_Main:Start();

end;

```

'OSt.Verwaltung' :

```

if (Rechte[Rgt_OSt]) then begin

    if (gMdiEKK = 0) then begin

        //gFrmMain->wpDisabled # true;

//if (gUsername='AH') then aName # 'OSt.E.Verwaltung';

        gMdiEKK # Lib_GuiCom:OpenMdi(gFrmMain, aName, _WinAddHidden);

```

```

gMdiEKK->WinUpdate(_WinUpdOn);

//gFrmMain->wpDisabled # false;

$NB.Main->WinFocusSet(true);

end

else begin

    Lib_guiCom:ReOpenMDI(gMdiEKK);

end;

end;

```

'Con.Verwaltung' :

```

if (StrFind(Set.Module,'C',0)<>0) and
    (Rechte[Rgt_Controlling]) then begin
    if (gMdiEKK = 0) then begin
        //gFrmMain->wpDisabled # true;

        gMdiEKK # Lib_GuiCom:OpenMdi(gFrmMain, 'Con.Verwaltung2', _WinAddHidden)
//      gMdiEKK # Lib_GuiCom:OpenMdi(gFrmMain, aName, _WinAddHidden);

        gMdiEKK->WinUpdate(_WinUpdOn);

        //gFrmMain->wpDisabled # false;

        $NB.Main->WinFocusSet(true);

    end

    else begin

        Lib_guiCom:ReOpenMDI(gMdiEKK);

    end;

end;

```

'Erl.Verwaltung' :

if (Rechte[Rgt_Erloese]) then begin

if (gMdiErl = 0) then begin

//gFrmMain->wpDisabled # true;

gMdiErl # Lib_GuiCom:OpenMdi(gFrmMain, aName, _WinAddHidden);

gMdiErl->WinUpdate(_WinUpdOn);

//gFrmMain->wpDisabled # false;

\$NB.Main->WinFocusSet(true);

end

else begin

Lib_guiCom:ReOpenMDI(gMdiERI);

end;

end;

'OfP.Verwaltung' :

if (Rechte[Rgt_OffenePosten]) then begin

if (gMdiOfp = 0) then begin

//gFrmMain->wpDisabled # true;

gMdiOfP # Lib_GuiCom:OpenMdi(gFrmMain, aName, _WinAddHidden);

gMdiOfP->WinUpdate(_WinUpdOn);

//gFrmMain->wpDisabled # false;

\$NB.Main->WinFocusSet(true);

end

else begin

Lib_guiCom:ReOpenMDI(gMdiOfP);

end;

end;

'OfP.Ablage' :

if (Rechte[Rgt_OffenePosten]) then begin

if (gMdiOfp = 0) then begin

//gFrmMain->wpDisabled # true;

gMdiOfP # Lib_GuiCom:OpenMdi(gFrmMain, aName, _WinAddHidden);

gMdiOfP->WinUpdate(_WinUpdOn);

//gFrmMain->wpDisabled # false;

\$NB.Main->WinFocusSet(true);

end

else begin

Lib_guiCom:ReOpenMDI(gMdiOfP);

end;

end;

'EKK.Verwaltung' :

if (Rechte[Rgt_EKK]) then begin

if (gMdiEKK = 0) then begin

//gFrmMain->wpDisabled # true;

gMdiEKK # Lib_GuiCom:OpenMdi(gFrmMain, aName, _WinAddHidden);

/***/

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

Filter_EKK # y;

vQ # '';

Lib_Sel:QInt(var vQ, 'EKK.EingangsreNr', '=', 0);

Lib_Sel:QRecList(0,vQ);

/***/

gMdiEKK->WinUpdate(_WinUpdOn);

//gFrmMain->wpDisabled # false;

\$NB.Main->WinFocusSet(true);

end

else begin

Lib_guiCom:ReOpenMDI(gMdiEKK);

end;

end;

'Vbk.Verwaltung' :

if (Rechte[Rgt_Vbk]) then begin

if (gMdiERe = 0) then begin

//gFrmMain->wpDisabled # true;

gMdiERe # Lib_GuiCom:OpenMdi(gFrmMain, aName, _WinAddHidden);

gMdiERe->WinUpdate(_WinUpdOn);

//gFrmMain->wpDisabled # false;

\$NB.Main->WinFocusSet(true);

end

else begin

Lib_guiCom:ReOpenMDI(gMdiEre);

end;

end;

'ERe.Verwaltung' :

if (Rechte[Rgt_ERe]) then begin

if (gMdiERe = 0) then begin

//gFrmMain->wpDisabled # true;

gMdiERe # Lib_GuiCom:OpenMdi(gFrmMain, aName, _WinAddHidden);

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

Filter_ERe # y;

Lib_Sel:QRecList(0, "ERe.Löschmarker" = "");

gMdiERe->WinUpdate(_WinUpdOn);

//gFrmMain->wpDisabled # false;

\$NB.Main->WinFocusSet(true);

end

else begin

Lib_guiCom:ReOpenMDI(gMdiERe);

end;

end;

'Kas.Verwaltung' :

if (Rechte[Rgt_Kasse]) then begin

```

if (gMdiERe = 0) then begin
    gMdiERe # Lib_GuiCom:OpenMdi(gFrmMain, aName, _WinAddHidden);
    VarInstance(WindowBonus,cnvIA(gMDIERe->wpcustom));
    gMdiERe->WinUpdate(_WinUpdOn);
    $NB.Main->WinFocusSet(true);
end
else begin
    Lib_guiCom:ReOpenMDI(gMdiERe);
end;
end;

```

'Kos.K.Verwaltung' :

```

if (Rechte[Rgt_Kostenbuchung]) then begin
    if (gMdiERe = 0) then begin
        gMdiERe # Lib_GuiCom:OpenMdi(gFrmMain, aName, _WinAddHidden);
        VarInstance(WindowBonus,cnvIA(gMDIERe->wpcustom));
        gMdiERe->WinUpdate(_WinUpdOn);
        $NB.Main->WinFocusSet(true);
    end
    else begin
        Lib_guiCom:ReOpenMDI(gMdiERe);
    end;
end;
end;

```

'ZEi.Verwaltung' :

if (Rechte[Rgt_ZAu]) then begin

if (gMdiOfp = 0) then begin

//gFrmMain->wpDisabled # true;

gMdiOfp # Lib_GuiCom:OpenMdi(gFrmMain, aName, _winAddHidden);

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

Filter_ZEi # y;

Lib_Sel:QRecList(0, "ZEi.Zugeordnet" < "ZEi.Betrag" OR ("ZEi.Zahldatum"=0.0.0)');

gMdiOfp->WinUpdate(_winUpdOn);

//gFrmMain->wpDisabled # false;

\$NB.Main->WinFocusSet(true);

end

else begin

Lib_guiCom:ReOpenMDI(gMdiOfp);

end;

end;

'ZAu.Verwaltung' :

if (Rechte[Rgt_ZAu]) then begin

if (gMdiERe = 0) then begin

//gFrmMain->wpDisabled # true;

gMdiEre # Lib_GuiCom:OpenMdi(gFrmMain, aName, _winAddHidden);

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

Filter_ZAu # y;

Lib_Sel:QRecList(0, "ZAu.Zahldatum" = 0.0.0');


```

gMdiEre->WinUpdate( _winUpdOn );

//gFrmMain->wpDisabled # false;

$NB.Main->WinFocusSet( true );

end

else begin

    Lib_guiCom:ReOpenMDI( gMdiEre );

end;

end;

```

'Rso.Kal.Verwaltung' :

```

if (Rechte[Rgt_Ressourcengruppen]) then begin;

    if (gMdiRsoKalender = 0) then begin

        //gFrmMain->wpDisabled # true;

        gMdiRsoKalender # Lib_GuiCom:OpenMdi(gFrmMain, aName, _WinAddHidden);

        gMdiRsoKalender->WinUpdate(_WinUpdOn);

        //gFrmMain->wpDisabled # false;

        $NB.Main->WinFocusSet(true);

    end

    else begin

        Lib_guiCom:ReOpenMDI(gMdiRsoKalender);

    end;

end;

```

'Job.Verwaltung' :

```
if (Rechte[Rgt_Jobs]) then begin
  if (gMdiPara = 0) then begin
    //gFrmMain->wpDisabled # true;

    gMdiPara # Lib_GuiCom:OpenMdi(gFrmMain, aName, _WinAddHidden);

    gMdiPara->WinUpdate(_WinUpdOn);

    //gFrmMain->wpDisabled # false;

    $NB.Main->WinFocusSet(true);

  end

  else begin

    Lib_guiCom:ReOpenMDI(gMdiPara);

  end;

end;
```

'Job.Err.Verwaltung' :

```
if (Rechte[Rgt_Jobs]) then begin
  if (gMdiPara = 0) then begin

    gMdiPara # Lib_GuiCom:OpenMdi(gFrmMain, aName, _WinAddHidden);

    gMdiPara->WinUpdate(_winUpdOn);

    $NB.Main->WinFocusSet(true);

  end

  else begin

    Lib_guiCom:ReOpenMDI(gMdiPara);

  end;

end;
```

'FSP.Verwaltung' :

if (Rechte[Rgt_Filescanpfade]) then begin

if (gMdiPara = 0) then begin

//gFrmMain->wpDisabled # true;

gMdiPara # Lib_GuiCom:OpenMdi(gFrmMain, aName, _WinAddHidden);

gMdiPara->WinUpdate(_WinUpdOn);

//gFrmMain->wpDisabled # false;

\$NB.Main->WinFocusSet(true);

end

else begin

Lib_guiCom:ReOpenMDI(gMdiPara);

end;

end;

'Kal.Verwaltung' :

if (Rechte[Rgt_Kalkulationen]) then begin

if (gMdiPara = 0) then begin

//gFrmMain->wpDisabled # true;

gMdiPara # Lib_GuiCom:OpenMdi(gFrmMain, aName, _WinAddHidden);

gMdiPara->WinUpdate(_WinUpdOn);

//gFrmMain->wpDisabled # false;

\$NB.Main->WinFocusSet(true);

end

else begin

Lib_guiCom:ReOpenMDI(gMdiPara);

end;

end;

'Tol.D.Verwaltung' :

if (Rechte[Rgt_Dickentoleranz]) then begin

if (gMdiPara = 0) then begin

//gFrmMain->wpDisabled # true;

gMdiPara # Lib_GuiCom:OpenMdi(gFrmMain, aName, _WinAddHidden);

gMdiPara->WinUpdate(_WinUpdOn);

//gFrmMain->wpDisabled # false;

\$NB.Main->WinFocusSet(true);

end

else begin

Lib_guiCom:ReOpenMDI(gMdiPara);

end;

end;

'WoF.Sch.Verwaltung' :

WoF_Sch_Main:Start();

'Betrieb': begin

// gMdiMenu # WinAddByName(gFrmMain, 'Mdi.Betrieb', _WinAddHidden);

// ggf. anderes Objekt benutzen

vName # Lib_GuiCom:GetAlternativeName('MDI.Betrieb');

```

gMdiPara # WinAddByName(gFrmMain, vName, _WinaddHidden);

if (gMDIPara<=0) then begin

    vName # 'MDI.Betrieb';

    gMdiPara # WinAddByName(gFrmMain, vName, _WinaddHidden);

    if (gMDIPara<=0) then begin

        Org_Data:Killme();

        TODO('DIALOG '+vName+' NICHT GEFUNDEN!');

        WinHalt();

    end;

end;

```

```

Lib_GuiCom:RecallWindow(gMdiPara);

```

```

//vName # UserInfo(_UserGroup, cnvia(UserInfo(_UserCurrent)));

//if (vName="") then vName # gUsername;

//if (vName = 'MC9090') or (vName= 'BETRIEB_TS') OR

// (vName = 'BETRIEB') then begin

if (gUserGroup='MC9090') or (gUserGroup= 'BETRIEB_TS') OR

    (gUserGroup = 'BETRIEB') then begin

    gMDIPara->wpStyleCloseBox # false;

end;

gMdiPara->WinUpdate(_WinUpdOn);

end;

```

```

'BDE': begin

```

```

vName # Lib_GuiCom:GetAlternativeName('MDI.BDE');

gMdiPara # WinAddByName(gFrmMain, vName, _WinaddHidden);

if (gMDIPara<=0) then begin
    Org_Data:Killme();

    TODO('DIALOG '+vName+' NICHT GEFUNDEN!');

    WinHalt();

end;

gMdiPara->WinUpdate(_WinUpdOn);

end;

'Coilculator': begin

if (gMDIMathCalculator=0) then begin

    //gFrmMain->wpDisabled # true;

    gMdiMathCalculator # Lib_GuiCom:OpenMdi(gFrmMain, Lib_GuiCom:GetAlternativeName('MDI.Coilculator'));

    gMdiMathCalculator->WinUpdate(_WinUpdOn);

    //gFrmMain->wpDisabled # false;

//    gMdiMathCalculator # WinAddByName(gFrmMain, 'MDI.Coilculator', 0);

end

else begin

//    gMDIMathCalculator->winfocusset(true);

    WinUpdate(gMDIMathCalculator, _winupdactivate);

//    Lib_guiCom:ReOpenMDI(gMdiMathCalculator);

end;

//    vHdl # WinAddByName(gFrmMain, 'MDI.Calculator', 0);

```

end;

//MC9090': begin

//gMdiMenu # WinAddByName(gFrmMain, 'Mdi.MC9090', _WinAddHidden);

//gMdiMenu->WinUpdate(_WinUpdOn);

//end;

otherwise begin

if (aName='Lnd.Verwaltung') then

if (Rechte[Rgt_Laender]=false) then RETURN;

if (aName='MTo.Verwaltung') then

if (Rechte[Rgt_Abmessungstol]=false) then RETURN;

if (aName='Abt.Verwaltung') then

if (Rechte[Rgt_Abteilungen]=false) then RETURN;

if (aName='Anr.Verwaltung') then

if (Rechte[Rgt_Anreden]=false) then RETURN;

if (aName='ArG.Verwaltung') then

if (Rechte[Rgt_Arbeitsgaenge]=false) then RETURN;

if (aName='Agr.Verwaltung') then

if (Rechte[Rgt_Artikelgruppen]=false) then RETURN;

if (aName='Apl.Verwaltung') then

if (Rechte[Rgt_Aufpreise]=false) then RETURN;

if (aName='ARr.Verwaltung') then

if (Rechte[Rgt_Auftragsarten]=false) then RETURN;

if (aName='BDS.Verwaltung') then

if (Rechte[Rgt_BDS]=false) then RETURN;

if (aName='Tol.D.Verwaltung') then

if (Rechte[Rgt_Dickentoleranz]=false) then RETURN;

if (aName='Eti.Verwaltung') then

if (Rechte[Rgt_Etiketten]=false) then RETURN;

if (aName='GKo.Verwaltung') then

if (Rechte[Rgt_Gegenkonten]=false) then RETURN;

if (aName='Grp.Verwaltung') then

if (Rechte[Rgt_Gruppen]=false) then RETURN;

if (aName='IHA.Mas.Verwaltung') then

if (Rechte[Rgt_IHA_Massnahmen]=false) then RETURN;

if (aName='IHA.Mld.Verwaltung') then

if (Rechte[Rgt_IHA_Meldungen]=false) then RETURN;

if (aName='IHA.Urs.Verwaltung') then

if (Rechte[Rgt_IHA_Ursachen]=false) then RETURN;

if (aName='Rso.KalTage.Verwaltung') then

if (Rechte[Rgt_Ressourcengruppen]=false) then RETURN;

if (aName='Kal.Verwaltung') then

if (Rechte[Rgt_Kalkulationen]=false) then RETURN;

if (aName='KSt.Verwaltung') then

if (Rechte[Rgt_KSt]=false) then RETURN;

// TODO('ST Lagerplätze als Stammdaten');

if (aName='LPI.Verwaltung') then

if (Rechte[Rgt_Lagerplaetze]=false) then RETURN;

if (aName='MSt.Verwaltung') then

if (Rechte[Rgt_Materialstatus]=false) then RETURN;

if (aName='Obf.Verwaltung') then

if (Rechte[Rgt_Oberflaechen]=false) then RETURN;

if (aName='Ort.Verwaltung') then

if (Rechte[Rgt_Orte]=false) then RETURN;

if (aName='MQu.Verwaltung') then

if (Rechte[Rgt_Qualitaeten]=false) then RETURN;

if (aName='MQu.S.Verwaltung') then

if (Rechte[Rgt_Qualitaeten]=false) then RETURN;

if (aName='RTy.Verwaltung') then

if (Rechte[Rgt_Rechnungstypen]=false) then RETURN;

if (aName='StS.Verwaltung') then

if (Rechte[Rgt_Steuerschluessel]=false) then RETURN;

if (aName='Skz.Verwaltung') then

if (Rechte[Rgt_Skizzen]=false) then RETURN;

if (aName='TTy.Verwaltung') then

if (Rechte[Rgt_Termintypen]=false) then RETURN;

if (aName='ULa.Verwaltung') then

if (Rechte[Rgt_Unterlagen]=false) then RETURN;

if (aName='VsA.Verwaltung') then

if (Rechte[Rgt_Versandarten]=false) then RETURN;

if (aName='VwA.Verwaltung') then

if (Rechte[Rgt_Verwiegungsarten]=false) then RETURN;

if (aName='Wae.Verwaltung') then

if (Rechte[Rgt_Waehrungen]=false) then RETURN;

if (aName='Wgr.Verwaltung') then

if (Rechte[Rgt_Warengruppen]=false) then RETURN;

if (aName='Art.Zst.Verwaltung') then

if (Rechte[Rgt_Artikel_Zustaende]=false) then RETURN;

if (aName='ZaB.Verwaltung') then

if (Rechte[Rgt_Zahlungsbed]=false) then RETURN;

if (aName='ZhA.Verwaltung') then

if (Rechte[Rgt_Zahlungsarten]=false) then RETURN;

if (aName='Zeu.Verwaltung') then

if (Rechte[Rgt_Zeugnisse]=false) then RETURN;

if (aName='Blb.Verwaltung') then begin

if (RunAFX('XLINK.CONNECT.DOKCA1','')<=0) then RETURN;

gDBAConnect # 3;

if (gMdiPara = 0) then begin

gMdiPara # Lib_GuiCom:OpenMdi(gFrmMain, aName, _WinAddHidden);

gMdiPara->WinUpdate(_WinUpdOn);

```

end

else begin

    Lib_guiCom:ReOpenMDI(gMdiPara);

end;

RETURN;

end;


if (gMdiPara = 0) then begin

    //gFrmMain->wpDisabled # true;

    gMdiPara # Lib_GuiCom:OpenMdi(gFrmMain, aName, _WinAddHidden);

    gMdiPara->WinUpdate(_WinUpdOn);

    //gFrmMain->wpDisabled # false;

    $NB.Main->WinFocusSet(true);

end

else begin

    Lib_guiCom:ReOpenMDI(gMdiPara);

end;

end;


end;

end;


//=====

// Action

//      Actionabwicklung

```

//=====

```
sub Action(  
    aNewMode      : Alpha;  
    opt aPageName  : alpha;  
)  
  
local begin  
  
    vNoRefresh : logic;  
  
    vHdl,vHdl2 : int;  
  
    vOk      : logic;  
  
    vListHdl  : int;  
  
    vMDI      : int;  
  
    vObj      : int;  
  
    vEvt      : event;  
  
    vSwitch   : alpha;  
  
    vTmp      : int;  
  
end;  
  
begin  
  
    // BUGFIX: gMDI zeigt ins Nirvana?  
  
    if (HdlInfo(gMDI, _HdlExists)=0) then begin  
  
        vHdl # TextOpen(10);  
  
        TextAddLine(vHdl,'gMDI not exists:'+aint(gMDI));  
  
        TextAddLine(vHdl,'w_Name = '+w_Name);  
  
        TextWrite(vHdl, StrCut('!!!RTE:'+gUsername,1,20),0);  
  
        TextClose(vHdl);  
  
        RETURN;
```

end;

REIN('Action')

if (aPageName='') then aPageName # 'NB.Page1';

//DEBUGX('Action : '+aNewMode);

// Organigramm

Org_Data:KeepAlive();

// TIMER DEAKTIVEREN...

WinEvtProcessSet(_WinEvtTimer, false);

Crit_Prozedur:Manage();

if (gMdi<>0) then WinSearchPath(gMDI);

// Initialize

if (Mode='') and ((aNewMode=c_Modelist) or (aNewMode=c_ModeedList)) then begin

Mode # aNewMode;

aNewMode # '';

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

if (vHdl<>0) then begin

vHdl->wpCurrent # 'NB.List';

/** Test

```

    FOR vHdl # WinInfo(vHdl, _winFirst, 1, _WintypeNotebookPage)

    LOOP vHdl # WinInfo(vHdl, _winNext, 1, _WintypeNotebookPage)

    WHILE (vHdl<>0) do begin

vHdl->wpTabOrder # _WintabOrderPageChild;

        END;

    ***/

    end;

    // beim Neueinstieg immer auf Suche fokussieren

    vTmp # WinSearch(gMDI,'ed.Suche');

    if (vTmp<>0) then vTmp->winfocusset(n);

    // TIMER AKTIVEREN...

    WinEvtProcessSet(_WinEvtTimer, true);

    RAUS('Action')

    RETURN;

end;

// direkt auf Ansicht springen???

if (Mode="") and (aNewMode=c_ModeView) then begin

    Mode    # aNewMode;

    aNewMode # "";

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

    if (vHdl<>0) then begin

        vHdl->wpCurrent # aPageName;

    end;

end;

```

```
// TIMER AKTIVEREN...
```

```
WinEvtProcessSet(_WinEvtTimer, true);
```

```
RAUS('Action')
```

```
RETURN;
```

```
end;
```

```
// Suchenmodus *****
```

```
// Suchenmodus *****
```

```
// Suchenmodus *****
```

```
if (aNewMode=c_ModeSearch) then begin // auf "Suchen" focusieren
```

```
//   Mode # c_ModeList;
```

```
  aNewMode # ";
```

```
  vHdl # WinFocusGet()
```

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

```
    if (vHdl->wpname = 'ed.Suche') then begin
```

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

```
      vHdl->wpCurrent # 'NB.List';
```

```
      $NB.List->WinFocusSet(false);
```

```
      gZLList->winFocusset(n);
```

```
      vHdl # gMdi->winsearch('ed.Suche');
```

```
      vHdl->wpCaption # ";
```

```
      vHdl # gMdi->winsearch('ed.Sort');
```

```
      vHdl->winFocusset(n);
```

```
      vHdl->wpPopupopen # true;
```



```

    vHdl # 1;

end

else begin

    vHdl # 0;

end;

end;


if (vHdl=0) then begin

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

    vHdl->wpCurrent # 'NB.List';

    $NB.List->WinFocusSet(false);

    gZLList->winFocusset(n);

    vHdl # gMdi->winsearch('ed.Suche');

    if (vHdl<>0) then begin

        vHdl->wpCaption # "";

        vHdl->winFocusset(n);

    end;

end;


// TIMER AKTIVEREN...

WinEvtProcessSet(_WinEvtTimer, true);

RAUS('Action')

RETURN;

end;

```

```
//debugx('Moduswechsel : '+Mode+' nach '+aNewMode);
```

```
case Mode of
```

```
// PROGRAMM BEENDEN?
```

```
'CLOSE', " : begin
```

```
if (aNewMode=c_ModeCancel) or (aNewMode=c_ModeCancel+'X') then begin
```

```
    Mode # c_ModeCancel;
```

```
    gMdi->Winclose();
```

```
    // TIMER AKTIVEREN...
```

```
    WinEvtProcessSet(_WinEvtTimer, true);
```

```
    RAUS('Action')
```

```
    RETURN;
```

```
end;
```

```
// 01.12.2015:
```

```
if (Mode='CLOSE') then begin
```

```
    Mode # aNewMode;
```

```
    // TIMER AKTIVEREN...
```

```
    WinEvtProcessSet(_WinEvtTimer, true);
```

```
    RAUS('Action')
```

```
    RETURN
```

```
end;
```

```
end;
```

```
// Listmodus *****
```

```
c_Modelist : begin
```

```
  if (aNewMode=c_ModelCancel) then    vSwitch # 'List-Cancel';
```

```
  if (aNewMode=c_ModelCancel+'X') then vSwitch # 'List-CancelX';
```

```
  if (aNewMode=c_ModelRecNext) then  vSwitch # 'List-RecNext';
```

```
  if (aNewMode=c_ModelRecPrev) then  vSwitch # 'List-RecPrev';
```

```
  if (aNewMode=c_ModelRecFirst) then vSwitch # 'List-RecFirst';
```

```
  if (aNewMode=c_ModelRecLast) then  vSwitch # 'List-RecLast';
```

```
  if (aNewMode=c_ModelView) then     vSwitch # 'List-View';
```

```
  if (aNewMode=c_ModelEdit) then     vSwitch # 'List-Edit';
```

```
  if (aNewMode=c_ModelNew) then      vSwitch # 'List-New';
```

```
  if (aNewMode=c_ModelDelete) then   vSwitch # 'List-Del';
```

```
end;
```

```
// Liste 2 (II) Modus *****
```

```
c_Modelist2 : begin
```

```
  if (aNewMode=c_ModelCancel) then    vSwitch # 'List2-Cancel';
```

```
  if (aNewMode=c_ModelCancel+'X') then vSwitch # 'List2-CancelX';
```

```
  if (aNewMode=c_ModelSave) then      vSwitch # 'List2-Save';
```

```
  if (aNewMode=c_ModelRecNext) then  vSwitch # 'List-RecNext';
```

```
  if (aNewMode=c_ModelRecPrev) then  vSwitch # 'List-RecPrev';
```

```
  if (aNewMode=c_ModelRecFirst) then vSwitch # 'List-RecFirst';
```

```
  if (aNewMode=c_ModelRecLast) then  vSwitch # 'List-RecLast';
```

```
  if (aNewMode=c_ModelNew) then      vSwitch # 'List2-New';
```

```

if (aNewMode=c_ModeEdit) then    vSwitch # 'List2-Edit';

if (aNewMode=c_ModeView) then    vSwitch # 'List2-Edit';

if (aNewMode=c_ModeDelete) then  vSwitch # 'List2-Del';

end;


// EdListe Modus *****

/***

c_ModeEdListedit : begin

if (aNewMode=c_ModeSave) then begin

    vEvt:Obj # Winfocusget();

    Lib_RecList:Save(vEvt);

    Winfocusset(gZLList, true);

//Mode # c_ModeEdList;

vSwitch # 'ok';

//vSwitch # 'List-Cancel';

    end;

end;

**/

c_ModeEdList : begin

if (aNewMode=c_ModeCancel) then    vSwitch # 'List-Cancel';

if (aNewMode=c_ModeCancel+'X') then vSwitch # 'List-CancelX';


if (aNewMode=c_ModeRecNext) then  vSwitch # 'List-RecNext';

if (aNewMode=c_ModeRecPrev) then  vSwitch # 'List-RecPrev';

if (aNewMode=c_ModeRecFirst) then vSwitch # 'List-RecFirst';

if (aNewMode=c_ModeRecLast) then  vSwitch # 'List-RecLast';

```

```
if (aNewMode=c_ModeView) then    vSwitch # 'List-Edit';
```

```
if (aNewMode=c_ModeDelete) then begin
```

```
//    vSwitch # 'List-Del';
```

```
if (gFile=0) then
```

```
    Lib_DataList:RemoveDLRow()
```

```
else
```

```
    Lib_RecList:RecDel();
```

```
// TIMER AKTIVEREN...
```

```
WinEvtProcessSet(_WinEvtTimer, true);
```

```
RAUS('Action')
```

```
RETURN;
```

```
end;
```

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

```
if (w_ListQuickEdit) then begin
```

```
    vSwitch # 'List-New';
```

```
end
```

```
else begin
```

```
if (gFile=0) then
```

```
    Lib_DataList:NewDLRow()
```

```
else
```

```
    Lib_RecList:StartListEdit(gZLList,y,0,_WinLstEditClearChanged);
```

```
// TIMER AKTIVEREN...
```

```
WinEvtProcessSet(_WinEvtTimer, true);
```

```
RAUS('Action')
```

RETURN;

end;

end;

if (aNewMode=c_ModeEdit) then begin

if (gZLList<>0) then

Lib_RecList:StartListEdit(gZLList,n,0,_WinLstEditClearChanged)

else

Call(gPrefix+'_Main:StartEdit')

// TIMER AKTIVEREN...

WinEvtProcessSet(_WinEvtTimer, true);

RAUS('Action')

RETURN;

end;

end;

// EdListe Modus *****

c_ModeEdListNew : begin

if (aNewMode=c_ModeSave) then begin

if (gZLList<>0) then begin

mode # c_ModeEdListNew2Save;

\$Save->Winfocusset(y);

RETURN;

end;

end;

// vSwitch # 'List-Save';

```
//vEvt:Obj # gZLList;
```

```
//Lib_RecList:Save(vEvt);
```

```
//RETURN;
```

```
end;
```

```
c_ModeEdListedit : begin
```

```
if (aNewMode=c_ModeSave) then begin
```

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

```
mode # c_ModeEdListEdit2Save;
```

```
$Save->Winfocusset(y);
```

```
RETURN;
```

```
end;
```

```
end;
```

```
end;
```

```
/**
```

```
if (aNewMode=c_ModeSave) then begin
```

```
if (gZLList<>0) and ((mode=c_ModeEdlistedit) or (mode=c_ModeEdlistnew)) then begin
```

```
// RecList?
```

```
debugx('simulate save');
```

```
mode # mode+'2SAVE';
```

```
$Save->Winfocusset(y);
```

```
RETURN;
```

```
end;
```

```
end;
```

```
***/  
*/
```

```
// Ansichtsmodus *****
```

```
c_ModeView : begin
```

```
  if (aNewMode=c_ModeRecNext) then  vSwitch # 'List-RecNext';
```

```
  if (aNewMode=c_ModeRecPrev) then  vSwitch # 'List-RecPrev';
```

```
  if (aNewMode=c_ModeRecFirst) then vSwitch # 'List-RecFirst';
```

```
  if (aNewMode=c_ModeRecLast) then  vSwitch # 'List-RecLast';
```

```
  if (aNewMode=c_ModeList) then    vSwitch # 'View-List';
```

```
  if (aNewMode=c_ModeEdit) then    vSwitch # 'View-Edit';
```

```
  if (aNewMode=c_ModeNew) then     vSwitch # 'View-New';
```

```
  if (aNewMode=c_ModeCancel) then  vSwitch # 'View-Cancel';
```

```
  if (aNewMode=c_ModeCancel+'X') then vSwitch # 'View-CancelX';
```

```
  if (aNewMode=c_ModeDelete) then  vSwitch # 'View-Del';
```

```
end;
```

```
// Neuanlagemodus *****
```

```
c_ModeNew : begin
```

```
  if (aNewMode=c_ModeCancel) then  vSwitch # 'New-Cancel';
```

```
  if (aNewMode=c_ModeCancel+'X') then vSwitch # 'New-CancelX';
```

```
  if (aNewMode=c_ModeSave) then begin
```

```
    // 20.07.2012 - beim Editieren von Datalist, den Inhalt vom Editobjekt übernehmen
```

```
    vHdl # Winfocusget();
```

```
    if (vHdl->wpname='edEditList') then begin
```

```
      vHdl2 # Wininfo(vHdl,_winparent); // Datalist holen
```

```
      vTmp # cnvia(vHdl->wpcustom);    // Column
```



```

    vHdl2->WinLstCellSet(vHdl->wpcaption, vTmp);

end;

vSwitch # 'New-Save';

end;

end;

// Neueingabe 2 (II) Modus *****
c_ModeNew2 : begin

    if (aNewMode=c_ModeCancel) then    vSwitch # 'New2-Cancel';

    if (aNewMode=c_ModeCancel+'X') then vSwitch # 'New2-CancelX';

    if (aNewMode=c_ModeSave) then      vSwitch # 'New2-Save';

    if (aNewMode=c_ModeEdit) then      vSwitch # 'List2-Edit';

    if (aNewMode=c_ModeView) then      vSwitch # 'List2-Edit';

    if (aNewMode=c_ModeNew) then       vSwitch # 'List2-New';

    if (aNewMode=c_ModeDelete) then    vSwitch # 'List2-Del';

end;

// Editiermodus *****
c_ModeEdit : begin

    if (aNewMode=c_ModeCancel) then    vSwitch # 'Edit-Cancel';

    if (aNewMode=c_ModeCancel+'X') then vSwitch # 'Edit-CancelX';

    if (aNewMode=c_ModeSave) then begin

        // 20.07.2012 - beim Editieren von Datalist, den Inhalt vom Editobjekt übernehmen

        vHdl # Winfocusget();

```

```

if (vHdl<>0) then

  if (vHdl->wpname='edEditList') then begin

    vHdl2 # Wininfo(vHdl,_winparent); // Datalist holen

    vTmp # cnvia(vHdl->wpcustom);    // Column

    vHdl2->WinLstCellSet(vHdl->wpcaption, vTmp);

  end;

  vSwitch # 'Edit-Save';

end;

end;

// Editier 2 (II) Modus *****

c_modeEdit2 : begin

  if (aNewMode=c_ModeCancel) then    vSwitch # 'Edit2-Cancel';

  if (aNewMode=c_ModeCancel+'X') then vSwitch # 'Edit2-CancelX';

  if (aNewMode=c_ModeSave) then      vSwitch # 'Edit2-Save';

end;

end; // Mode

if (vSwitch='') then begin

  TODO('Moduswechsel : '+Mode+' nach '+aNewMode);

  // TIMER AKTIVEREN...

  WinEvtProcessSet(_WinEvtTimer, true);

  RAUS('Action');

  RETURN;

end;

```

```

aNewMode # ";

if (vSwitch<>'ok') then

    App_Main_Sub:ModeCommand(vSwitch, aPageName);


// TIMER AKTIVEREN...

WinEvtProcessSet(_WinEvtTimer, true);

RAUS('Action')

RETURN;

end;


//=====

// Mark

//

//=====

sub Mark(

    opt aDataList : int)

local begin

    vl : int;

end;

begin

    if (aDataList<>0) then begin

        vl # aDataList->wpCurrentInt;

        if (vl>0) then begin

            Lib_Mark:MarkAdd(aDataList, n, y, vl);

```

```

aDataList->WinUpdate(_WinUpdOn, _WinLstFromSelected);

aDataList->WinFocusSet(false);

end;

RETURN;

end;


Lib_Mark:MarkAdd(gFile);

end;


//=====

// AttachmentsShow

//  Selektiert die externen Anhänge eines Datensatzes und zeigt diese

//  in der Verwaltung an

//=====

sub AttachmentsShow()

local begin

    Erx      : int;

    vFile    : int;

    vA       : alpha;

    vQ,vQ2   : alpha(4000);

    vHdlSel  : int;

    vHdlKey1 : handle;

    vHdlKey2 : handle;

end;

begin

```

```
if (gFile=0) then RETURN;
```

```
vFile # gFile;
```

```
if (Mode=c_ModeList) then
```

```
    RecRead(gFile,0,0,gZLList->wpdbrecid);
```

```
// 13.07.2016:
```

```
// 18.07.2023 DS RunAFX('Anh.Show'...) erst nach RecRead, damit es sicher auf dem fokussierten (und
```

```
if (RunAFX('Anh.Show',aint(gFile))<0) then RETURN;
```

```
RecBufClear(916);
```

```
gMDI # Lib_GuiCom:AddChildWindow(gMDI, 'Anh.Verwaltung', '', true);
```

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

```
if (gZLList -> wpDbSelection = 0) then begin
```

```
    vQ # Anh_Data:SelQuery(vFile);
```

```
    vHdlSel # SelCreate(916, 4); // 26.03.2020
```

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

```
    if (Erx != 0) then
```

```
        Lib_Sel:QError(vHdlSel);
```

```

// speichern, starten und Name merken...

w_SelName # Lib_Sel:SaveRun(var vHdlSel, 0, false);

// Liste selektieren...

gZLList->wpDbSelection # vHdlSel;

end

// in den custom Feldern der GUI Elemente werden Daten hinterlegt, die mitunter
// ZUM SPEICHERN von Anhängen genutzt werden, daher wird Anh_Data:Remap(..., TRUE) verwendet

vHdlKey1 # gMDI->winsearch('lb.key1');

vHdlKey1->wpcustom # cnvai(Anh_Data:Remap(vFile, true));

vHdlKey2 # gMDI->winsearch('lb.key2');

vHdlKey2->wpcustom # Anh_Data:MakeKey(CnvIA(vHdlKey1->wpcustom));

Lib_GuiCom:RunChildWindow(gMDI)

end;

//=====

// Refresh

//

//=====

sub Refresh(opt aMaches : logic);

local begin

Erx : int;

vHdl : handle;

vList : handle;

```

vOK : logic;

end;

begin

if (gMdi<>0) then WinSearchPath(gMDI); // 21.11.2011

if (Mode=c_ModeView) then begin // 2023-08-24 AH

if (gPrefix<>") then Call(gPrefix+'_Main:RefreshIfm');

end;

// 07.05.2018 AH

if (gDataList<>0) and (gPrefix<>") then begin

try begin

ErrTryIgnore(_rlocked,_rNoRec);

ErrTryCatch(_ErrNoProcInfo,y);

Call(gPrefix+'_Main:Refresh');

end;

end;

vList # gZLList;

if (vList<>0) and ((Mode=c_ModeList) or (mode=c_ModeEdList) or (Mode=c_ModeList2) or (aMaches)) then

// 25.05.2021 AH: NEU, z.B. BA-FM-Anzeige auf Input

try begin

ErrTryIgnore(_rlocked,_rNoRec);

ErrTryCatch(_ErrNoProcInfo,y);

```

vOK # Call(gPrefix+'_Main:Refresh');

if (vOK=false) then RETURN;

end;


vHdl # vList->wpDbSelection;

// bei Selektionslisten, erstmal Selektion neu starten

if (vHdl<>0) and (w_SelName<>") then begin

    Erx # SelRun(vHdl,_SelDisplay | _SelServer | _SelServerAutoFld);

end;

vList->WinUpdate(_WinUpdOn, _WinLstRecFromBuffer|_WinLstRecDoSelect);

end;


end;


//=====

// _SetMenu

//

//=====

sub _SetMenu(aName : alpha);

local begin

    vHdl : int;

end;

begin

    if (gFrmMain->wpmenuname<>aName) then begin

```



```

gFrmMain->wpMenuname # aName; // Menü setzen

gMenu # gFrmMain->WinInfo(_WinMenu);

if (gMenu<>0) then begin

    Lib_GuiCom:TranslateObject(gMenu);

    if (gPrefix<>"") then begin

        vHdl # gFrmMain->WinInfo(_WinMenu);

        Lib_SFX:CreateMenu(vHdl, gPrefix);

        Help_Main:AddMenu(vHdl);

    end;

end;

end;

end;

// 25.03.2020 AH:

if (gMenu<>0) then begin

    RunAFX('HM.Init',aint(gMenu)); // 12.01.2022 AH

    if ((gMenu->wpname='*Main*')=false) then begin

        vHdl # Winsearch(gMenu, 'Mnu.Comment');

        if (vHdl=0) then begin

            vHdl # Winsearch(gMenu, 'Extras');

            if (vHdl<>0) then begin

                vHdl # vHdl->WinMenuItemAdd('Mnu.Comment','&Kommentar',2);

                if (vHdl<>0) then

                    vHdl->wpMenuKey # _WinKeyf1 | _WinkeyCtrl;

            end;

        end;

    end;

end;

```

```

    end;

end;

// vHdl # Winsearch(gMenu, 'Extras');

// if (vHdl<>0) then begin

//   vHdl # vHdl->WinMenuItemAdd('Mnu.SetUserStatus','Userstatus setzen',2);

//   vHdl->wpMenuKey # _winkeyShift | _WinKeyCtrl | _WinkeyY;

// end;

end;

end;

//=====

// EvtMdiActivate

//           MDI-Fenster erhält Focus

//=====

sub EvtMdiActivate(

    aEvt           : event;    // Ereignis

) : logic

local begin

    vA           : alpha;

    vChild       : int;

    vParent      : int;

    vHdl,vHdl2   : int;

    vHdl3        : int;

    vName        : alpha;

```

```

vAusChild  : logic;

vChildMode : int;

vFilter    : int;

vX         : int;

vVorgabeKey : int;

vDLSort    : int;

vFrame     : int;

vOK        : logic;

vTmp       : int;

vFocus     : int;

end;

begin

    REIN('MdiActivate :'+aEvt:obj->wpname)

BugFix(__PROCFUNC__, aEvt:Obj);

// Organigramm

Org_Data:KeepAlive();

/*

// Test 26.1.2010 AI

vHdl # aEvt:Obj;

if (vHdl<>0) then begin

    if (vHdl->wpcustom<>") and

        (vHdl->wpcustom<>cnvai(VarInfo(WindowBonus))) then

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

```

```
end;
```

```
*/
```

```
// aEvt:Obj->wpChangedTrigger # aEvt:Obj->wpChangedTrigger | _WinChgTriggerFldBuf | _WinChgTrigg
```

```
vFrame # aEvt:Obj;
```

```
gMdi # vFrame;
```

```
gFrmMain->wpautoupdate # n;
```

```
// Hauptmenü setzen
```

```
_SetMenu(gMenuName);
```

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

```
    todo('Aktiv trotz Kinder!!!');
```

```
    RETURN true;
```

```
end;
```

```
// gMDI # aEvt:obj;
```

```
//BUGFIX Winsearchpath(gMDI);
```

```
/* WORKAROUND für RunChildWIndow
```

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

```
    if (HdllInfo(w_Lastfocus,_HdlExists)>0) then
```

```

    WinFocusSet(w_LastFocus);

w_LastFocus # 0;


RAUS('MdiActivate')

RETURN false;

end;

*/


if (vFrame->wpname=Lib_GuiCom:GetAlternativeName('Mdi.Hauptmenue')) or
(vFrame->wpname=Lib_GuiCom:GetAlternativeName('Mdi.Notifier')) or
(vFrame->wpname=Lib_GuiCom:GetAlternativeName('Mdi.Cockpit')) or
(vFrame->wpname=Lib_GuiCom:GetAlternativeName('Mdi.Workbench')) then begin
vTmp # $MDI.Hauptmenue;
if (vTmp<>0) then begin
    // HAUPTMENÜ SETZEN...

    Rgt_Menudata(wininfo($AppFrame,_WinMenu),Winsearch($Mdi.Hauptmenue,'TV.Hauptmenue'));
end;

if (ErrList<>0) then begin
    Msg(019999,'ERRORLISTE GEFÜLLT!!!',_winicoerror, _WinDialogOk,0);
    ErrorOutput;
end;


vX # 0;

//@ifdef Prog_ReadOnly

//  if (cReadOnlyCondition) then vX # 1;

//@endif

```

```

if (TransCount<>vX) then begin
    Error(019999,'TRANSAKTIONSCOUNTER = '+AInt(TransCount));
    ErrorOutput;
end;

```

```

RAUS('MdiActivate')

RETURN true;

end;

```

```

if (vFrame->wpname=Lib_GuiCom:GetAlternativeName('Mdi.Betrieb')) then begin
    App_Betrieb:RightCheck();
    RAUS('MdiActivate')
    RETURN true;
end;

```

```

// Problem bei Neustart des Fensters, dass REPOS hier nicht klappt. Dann muss das über EVTCREATED
if (w_Command='REPOS') or (w_Command='NEWREPOS') then begin
    // 2022-09-05 AH "NEWREPOS" auch hier Proj. 2228/82/10
    RecRead(gFile,0,_recId, cnvia(w_Cmd_Para));
//debugx(Mode+' repos KEY401 '+aint(RecInfo(401,_RecID))+ ' soll '+w_Cmd_para);
    if (Mode<>'') and (w_Command='REPOS') then w_Command # '';
/*
vTMP # gMDi->wpDBRecBuf(gFile);
if (vTMP<>0) then begin

```

```

debugx('copy KEY401  '+w_name);

    RecBufcopy(gFile, vTMP);

end;

*/

// gMDI->winupdate(_winupdon|_WinUpdFld2Buf);

if (gZLList<>0) and

    ((mode=c_ModeList) or (Mode=c_ModeBald+c_ModeView)) then begin    // 12.01.2021 AH : NEU das

        gZLList->winupdate(_Winupdon, _WinIstRecDoSelect); // 12.11.2014

    end;

end;

/****/

// a) 17.05.2022 AH: Fix z.B. BA1.Verwaltung, wenn Selektion einen Satz ausgrenzt, der aber andere GU

// b) 2022-07-01 AH: DEAKTIVIERT, weil dann Artikel->Mat->(mehrdeutiger Key)->Kommisionieren versp

// if (HdlInfo(gZLList, _HdlExists)<>0) and (gZLList->wpDbSelection<>0) and (gFile<>0) and (Mode=c_Mo

// RecRead(gFile,gZLList->wpDbSelection,0);

// end;


// gelinkter Datensatz? -> dann Sort/Suche abschalten

if (HdlInfo(gZLList, _HdlExists)<>0) and // 28.01.2021 AHGWS

    (gZLList<>0) then begin

    vOK # y;

    if (gZLList->wpDbLinkFileNo<>0) or (gZLList->wpdbfilter<>0) and (gMDI<>0) then vOK # n;

    if (gZLList->wpDbSelection<>0) then begin

        if (SelInfo( gZLList->wpDbSelection, _selSort)=0) then vOK # n;

    end;

    // Ausblenden...

```

```

if (vOK=n) then begin
    vTmp # gMDI->WinSearch('lb.Sort');
    if (vTmp<>0) then vTmp->wpvisible # false;
    vTmp # gMDI->WinSearch('lb.Suche');
    if (vTmp<>0) then vTmp->wpvisible # false;
    vTmp # gMDI->WinSearch('ed.Sort');
    if (vTmp<>0) then vTmp->wpvisible # false;
    vTmp # gMDI->WinSearch('ed.Suche');
    if (vTmp<>0) then vTmp->wpvisible # false;
end;

end;

if (mode=c_modeedList) and (gZLList<>0) then begin // wegen ededlist
    Lib_GuiCom:SetMaskState(false);
end;

if (Mode="") and (gZLList<>0) then begin // Neueinstieg?
    Lib_GuiCom:SetMaskState(false); // 16.01.06
    Action(c_ModeList);           // dann in Übersichtsliste gehen
end

else if (Mode=c_modeBald+c_ModeEdit) then begin
    Lib_GuiCom:SetMaskState(false);
    Mode # "";
    Mode # c_modeList;
//  Action(c_ModeList);           // dann erst in Übersichtsliste gehen
//  Action(c_modeView);
    Action(c_ModeEdit);

```



```

end

else if (Mode=c_modeBald+c_ModeNew) then begin

    Lib_GuiCom:SetMaskState(false);

    Mode # "";

    Action(c_ModeList);          // dann erst in Übersichtsliste gehen

    Action(c_ModeNew);

    vTmp # RekSave(gFile);


    // alles markieren...

    gTMP # winfocusget();

    if (gTMP<>0) and (WinInfo(gTMP,_WinType)<>_WinTypeCheckbox) then

        gTMP->wprange # Rangemake(0,100);


// ?? 16.10.2012 AI  w_Command # 'BUF';

//  w_Cmd_para # aint(vTmp);

end

else if (Mode=c_modeBald+c_ModeView) then begin

    Lib_GuiCom:SetMaskState(false);

    Mode # "";

    Action(c_ModeView, w_BaldPage);

    w_BaldPage # "";

/*

dxdebugx(gmdi->wpname);

vTmp # gMdi->winsearch('Edit');

if (vTmp->wpdisabled) then begin

    if (gMdi->winsearch('EditErsatz')<>0) then begin

```

```

vTmp # gMdi->winsearch('EditErsatz');

vTmp->wpdisabled # false;

end;

end;

vTmp->winfocusset(true);

dxdebugx('set:'+vtmp->wpname);

*/

//vTmp # Winfocusget();

// RefreshMode();           // Menü & Buttons setzen

//vTmp->winfocusset(true);

//RETURN true;

// früher so - ging aber nicht wegen RecID=0 in der ZL

//  Action(c_ModeList);      // dann erst in Übersichtsliste gehen

//  Action(c_ModeView);

end;

```

```

vFocus # Winfocusget();

RefreshMode();           // Menü & Buttons setzen

gFrmMain->wpautoupdate # y;

```

```

// 06.10.2016 AH:

```

```

if (Mode=c_ModeView) then begin

    vFocus # gMdi->winsearch('Edit');

    if (vFocus<>0) then

        if (vFocus->wpdisabled) then vFocus # 0;

    if (vFocus=0) then begin

```

```

vFocus # gMdi->winsearch('EditErsatz');

if (vFocus<>0) then vFocus->wpdisabled # false;

end;

end;

if (vFocus<>0) then vFocus->winfocusset(true);

RAUS('MdiActivate')

RETURN true;

end;

//=====

// EvtClicked

//      Mausclick auf Buttons

//=====

sub EvtClicked(

    aEvt      : event;    // Ereignis

) : logic

local begin

    Erx  : int;

    vTmp : int;

    vAfx : int;

end;

begin

    REIN('EvtClicked')

```

```

BugFix(__PROCFUNC__, aEvt:Obj);

// if (gMdi<>0) then WinSearchPath(gMDI); // 21.11.2011

// quickbar

/**

if (gPrefix<>"") and

  ((Mode=c_Modeedit) or

  (Mode=c_ModeEdit2) or

  (Mode=c_ModeNew) or

  (Mode=c_ModeNew2)) then begin

  vTmp # WinFocusGet();

  if (vTmp<>0) then begin

    end;

  end;

end;

***/

if (w_QBButton>0) then begin

  if (StrCut(aEvt:Obj->wpname,1,8)='bt.Quick') then Lib_GuiCom:SetQBButton(aEvt:obj);

  RETURN true;

end

else begin

  if (StrCut(aEvt:Obj->wpname,1,8)='bt.Quick') then begin

    if (aEvt:Obj->wpcustom<>"") then begin

      gMenu # gFrmMain->WinInfo(_WinMenu);

      if (gMenu<>0) then begin

        // 11.10.2019

```

```

if (StrCut(aEvt:obj->wpcustom,1,4)='LFM;') then begin

    Lfm.Kuerzel # Str_Token(aEvt:obj->wpcustom,',';2);

    Lfm.Nummer # cnvia(Str_Token(aEvt:obj->wpcustom,',';3));

    Erx # RecRead(910,1,0);

    if (Erx<=_rLocked) then Lfm_Ausgabe:Starten(Lfm.Kuerzel, Lfm.Nummer);

    RETURN true;

end;

vTmp # gMenu->WinSearch(aEvt:obj->wpcustom);

if (vTmp <> 0) then begin

    if (vTmp->wpdisabled=false) then begin

        EvtMenuCommand(aEvt, vTmp);

    end;

end;

end;

end;

RETURN true;

end;

end;

```

```

case (aEvt:Obj->wpname) of

```

```

'Refresh' : begin

```

```

    Refresh();

```

```

end;

```

```
'Mark' : begin  
  
  if (Mode=c_ModeList) then begin  
  
    if (gDataList<>0) then begin  
  
      Mark(gDataList);  
  
    end  
  
    else if (gZLList->wpdbrecid<>0) then begin  
  
      RecRead(gFile,0,0,gZLList->wpdbrecid);  
  
      Mark();  
  
    end;  
  
  end;  
  
end;
```

```
'Attachment' : begin  
  
  AttachmentsShow();  
  
end;
```

```
'Aktivitaeten' : begin  
  
  if (gFile<>0) then TeM_Sub:Start(gFile);  
  
end;
```

```
'Delete': begin  
  
  Action(c_ModeDelete);
```

end;

'Edit': begin

 Action(c_ModeEdit);

end;

'Cancel': begin

 Action(c_ModeCancel);

end;

'New': begin

 Action(c_ModeNew);

end;

'Save': begin

 if (Mode=c_ModeList) and (gPrefix<>"") then

 Call(gPrefix+'_Main:RecSave')

// else if (Mode=c_ModeEdListEdit) then

// Lib_RecList:Save(aEvt)

 else

 Action(c_ModeSave);

end;

'Search' : begin

vAfx # RunAFX('DMS.Search', "");

// ggf. Fallback auf STD Funktionalität:

if vAfx >= 0 then

begin

//DebugM('STD');

Action(c_ModeSearch);

end

end;

end; // ...case

RAUS('EvtClicked')

end;

//=====

// EvtPageSelect

// Seitenauswahl von Notebooks

//=====

sub EvtPageSelect(

 aEvt : event; // Ereignis

 aPage : int;


```

aSelecting      : logic;

opt aPageNew     : int;

) : logic

local begin

    Erx          : int;

    vOk          : logic;

    vHdl         : int;

    vTmp         : int;

    vNB          : int;

end;

begin

    REIN('PageSelect')


    BugFix(__PROCFUNC__, aEvt:Obj);

    // gMdi # aEvt:Obj->WinInfo(_WinFrame);

    // if (gMdi<>0) then WinSearchPath(gMDI); // 21.11.2011


    if (aSelecting=n) then begin

/*

        vOk # y;

        try begin

            ErrTryIgnore(_rlocked,_rNoRec);

            ErrTryCatch(_ErrNoSub,y);

            if (gPrefix<>") then vOk # Call(gPrefix+'_Main:EvtPageDeSelect',aEvt, aPage, aSelecting, aPageNew);

```

```

end;

if (vOk=false) then begin
    RAUS('PageSelect2')
    RETURN false;
end;

*/

RAUS('PageSelect')
RETURN true;

end;

// 18.01.2018 AH : von "unten"

vOk # y;

try begin
    ErrTryIgnore(_rlocked,_rNoRec);
    ErrTryCatch(_ErrNoSub,y);
    if (gPrefix<>") then vOk # Call(gPrefix+'_Main:EvtPageSelect',aEvt, aPage, aSelecting);
end;

// if (vOk=false) then begin
//   RAUS('PageSelect2')
//   RETURN false;
// end;

if ((Mode=c_ModeView)) and // or (mode=c_ModeList)) and
(aPage->wpname<>'NB.List') then begin // 13.5.2004
    vNB # gMdi->winsearch('NB.Main');

```

```
// 03.08.2012
```

```
vNB->wpautoupdate # false;
```

```
vNB->wpCurrent(_WinFlagNoFocusSet) # aPage->wpname;
```

```
vTmp # gMdi->winsearch('Edit');
```

```
if (vTmp<>0) and (vTmp->wpdisabled) then begin
```

```
  if (gMdi->winsearch('EditErsatz')<>0) then begin
```

```
    vTmp # gMdi->winsearch('EditErsatz');
```

```
    vTmp->wpdisabled # false;
```

```
  end;
```

```
end;
```

```
if (vTmp<>0) then vTmp->winfocusset(true);
```

```
// 11.07.2012 AI
```

```
RefreshMode();
```

```
vNB->wpautoupdate # true;
```

```
RAUS('PageSelect1')
```

```
RETURN false;
```

```
end;
```

```
/**
```

```
vOk # y;
```

```
try begin
```

```
  ErrTryIgnore(_rlocked,_rNoRec);
```

```
  ErrTryCatch(_ErrNoSub,y);
```

```

    if (gPrefix<>"") then vOk # Call(gPrefix+'_Main:EvtPageSelect',aEvt, aPage, aSelecting);
end;

***/

if (vOk=false) then begin

    RAUS('PageSelect2')

    RETURN false;

end;


// test AI 04.03.2011 war vor dem TRY block oben

if ((mode=c_ModeList)) and

    (aPage->wpname<>'NB.List') then begin

    vNB # gMdi->winsearch('NB.Main');

    vNB->wpautoupdate # false;

    Action(c_ModeView);

    if (Mode<>c_ModeList) then begin    // 2022-06-23 AH

        vNB->wpCurrent(_WinFlagNoFocusSet) # aPage->wpname;

//    vNB->wpautoupdate # true;

        RefreshMode();

    end;

    vNB->wpautoupdate # true;

    RAUS('PageSelect1')

    RETURN false;

end;


if ((Mode=c_ModeList) or (Mode=c_ModeEdList)) and (aPage->wpname<>'NB.List') then begin

```

```
// 19.04.2010 AI : Maskenwechsel mit Maus bei gelinkten RecLists

// if (gZLList->wpDbLinkFileNo<>0) then begin

//   Erx # RecLink(gZLList->wpDbLinkFileNo,gZLList->wpDbFileNo,gZLList->wpdbkeyno,_Rectest);

// end

// else begin

  Erx # RecRead(gFile,0,_recid,gZLList->wpDbRecId);

// end;

if (Erx>_rLocked) then begin

  RAUS('PageSelect3')

  RETURN false;

end;

end;
```

```
if (Mode=c_ModeList2) and (aPage->wpname<>'NB.Erfassung') then begin

  Action(c_ModeEdit)

  RAUS('PageSelect4')

  RETURN true;

end;
```

```
if (Mode=c_ModeView) or (Mode=c_ModeEdList) or (Mode=c_ModeList) then begin

  if (aPage->wpname='NB.List') then begin

    Mode # c_ModeList;
```

//21.06.2007 28.06.2007 muss so sein, wegen "View->List" per Maus

```
gZLList->WinUpdate(_WinUpdOn, _WinLstRecFromRecId|_WinLstRecDoSelect);
```

```
// gZLList->WinUpdate(_WinUpdOn, _WinLstRecFromBuffer | _WinLstRecDoSelect);
```

```
$NB.List->WinFocusSet(false);
```

```
RefreshMode();
```

```
end
```

```
else begin
```

```
Mode # c_ModeView;
```

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

```
RefreshMode();
```

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

```
// 03.08.2012
```

```
vHdl->wpCurrent(_WinFlagNoFocusSet) # aPage->wpname
```

```
if (gPrefix<>") then Call(gPrefix+'_Main:RefreshIfm');
```

```
vTmp # gMdi->winsearch('Edit');
```

```
if (vTmp->wpdisabled) then
```

```
if (gMdi->winsearch('EditErsatz')<>0) then
```

```
vTmp # gMdi->winsearch('EditErsatz');
```

```
vTmp->winfocusset(false);
```

```
RAUS('PageSelect5')
```

```
RETURN false;
```

```

    end;

end;

RAUS('PageSelect')

RETURN true;

end;


//=====

// EvtLstSelect

//      Zeilenauswahl von RecList/DataList

//=====

sub EvtLstSelect(

    aEvt      : event;    // Ereignis

    aRecID    : int;

) : logic

local begin

    Erx      : int;

    vParent : int;

    vA       : alpha;

    vButton  : int;

    // vFilter : int;

end;

begin

// WinEvtProcessSet(_WinEvtMdiActivate, false);

    REIN('EvtLstSelect')

```

```
/**
```

```
//gMDIBAG # 1;
```

```
for vparent # 1 loop inc (vParent) while vParent<100 do begin
```

```
    Mat.Nummer# vParent * cnvif( random() * 1000.0);
```

```
    RecRead(200,1,0);
```

```
end;
```

```
//winsleep(1);
```

```
//RecRead(903,1,0);
```

```
//end;
```

```
//gMDIBag # 0;
```

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

```
    gZLList->wpColFocusBkg  # Set.Col.RList.Cursor;
```

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

```
// Lib_GuiCom:ZLColorLine(gZLList, RGB(100,100,100));
```

```
// Lib_GuiCom:ZLColorLine(gZLList, RGB(150,150,150));
```

```
// Lib_GuiCom:ZLColorLine(gZLList, RGB(200,200,200));
```

```
end;
```

```
RAUS('EvtLstSelect')
```

```
RETURN true;
```

```
**/
```

```
BugFix(__PROCFUNC__, aEvt:Obj);
```

```
// if (gMdi<>0) then WinSearchPath(gMDI); // 21.11.2011
```

```
if (aEvt:obj->wpname='DL.Sort') then begin
```



```

WinLstCellGet($DL.Sort,vA,2,aRecId);

gKeyID # 0;

// LfdNr , C16Key

if (Str_Count(vA,'|')=2) then begin

    gKeyID  # CnvIA(Strcut(vA,1,3));

    gKey    # CnvIA(Strcut(vA,5,3));

    gSuchProc # Strcut(vA,9,50);

end

else begin

    // alt

    gKey    # CnvIA(Strcut(vA,1,3));

    gSuchProc # Strcut(vA,5,50);

end;

if (gKeyID=0) then gKeyID # gKey;

Lib_GuiCom:ZLSetSort(0, gKeyID);

$ed.Suche->WinFocusSet(false);

end;

```

```

vParent # aEvt:obj->WinInfo(_WinParent);

If (vParent->wpname ='gt.List') or

    (vParent->wpName = 'NB.List') or (vParent->wpName = 'NB.Erfassung') then begin

    if (gZLList<>0) then begin

/*      gZLList->wpColFocusBkg  # rgb(10,0,0);

        gZLList->wpColFocusOffBkg # rgb(20,0,0);

```

```

gZLList->wpColFocusBkg # rgb(30,0,0);
gZLList->wpColFocusOffBkg # rgb(40,0,0);
gZLList->wpColFocusBkg # rgb(50,0,0);
gZLList->wpColFocusOffBkg # rgb(60,0,0);
gZLList->wpColFocusBkg # rgb(70,0,0);
gZLList->wpColFocusOffBkg # rgb(80,0,0);
gZLList->wpColFocusBkg # rgb(100,0,0);
gZLList->wpColFocusOffBkg # rgb(120,0,0);
gZLList->wpColFocusBkg # rgb(140,0,0);
gZLList->wpColFocusOffBkg # rgb(160,0,0);
gZLList->wpColFocusBkg # rgb(170,0,0);
gZLList->wpColFocusOffBkg # rgb(190,0,0);

*/

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

// gZLList->wpAutoUpdate # true;


// ST 2012-03-20 START Projekt: 1326/192

vButton # gMdi->WinSearch('Attachment');

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


// 13.07.2016:

if (RunAFX('Anh.Check',aint(gFile))>=0) then begin

```

```

    Erx # Anh_Data:Check(gFile);

end

else begin

    Erx # AfxRes;

end;

if (Erx < _rNoKey) then

    vButton->wplImageTileUser # clmgHatAnhang; //185+1

else

    vButton->wplImageTileuser # clmgKeinAnhang; //185;

//    vFilter->RecFilterDestroy();

```

// 29.03.2020 AH:

```

vButton # gMdi->WinSearch('Aktivitaeten');

if (vButton <> 0) then begin

    Erx # Tem_Data:Check(gFile);

    if (Erx=-100) then vButton->wpVisible # false;

    if (Erx < _rNoKey) then

        vButton->wplImageTileUser # clmgHatAktivitaet

    else

        vButton->wplImageTileUser # clmgKeineAktivitaet;

    end;

end;

// ST 2012-03-20 ENDE Projekt: 1326/192

end;

```

```

if (gPrefix<>") then begin

```

```

try begin
    ErrTryIgnore(_rlocked,_rNoRec);
    ErrTryCatch(_ErrNoProcInfo,y);
    Call(gPrefix+'_Main:EvtLstSelect',aEvt,aRecId)
end;

if (errget() = _errnoproinfo) then begin
    Erx # RecRead(gFile,0,_RecId,aRecID);
end;

end;

end;

RAUS('EvtLstSelect')

// WinEvtProcessSet(_winEvtAll, true);

RETURN true;

end;

//=====

// EvtMenuInitKontext

//      EvtMenuInitKontext für Favoritenmenü

//=====

sub EvtMenuInitKontext (
    aEvt      : event;
    aMenuItem : handle ) : logic
local begin

```

```
vMenuItem : handle;  
  
vMenuParent : handle;  
  
vCtxMenu : handle;  
  
vCtxItem : handle;
```

```
end
```

```
begin
```

```
BugFix(__PROCFUNC__, aEvt:Obj);
```

```
vMenuItem # $TV.Hauptmenue->wpCurrentInt;
```

```
vCtxMenu # WinInfo( aEvt:obj, _winContextMenu );
```

```
// Leermenüpunkte nicht anzeigen
```

```
WinMenuItemRemove( vCtxMenu->WinSearch( 'temp' ) );
```

```
WinMenuItemRemove( vCtxMenu->WinSearch( 'Default_set' ) );
```

```
WinMenuItemRemove( vCtxMenu->WinSearch( 'miMenuItem' ) );
```

```
// Kontextmenü für Knoten anpassen
```

```
if ( vMenuItem != 0 ) and ( vMenuItem->wpNodeStyle != _winNodeFolder ) then begin
```

```
    vMenuParent # vMenuItem->WinInfo( _winParent );
```

```
// Favorit hinzufügen / entfernen
```

```
if ( vMenuParent->wpName = 'Favoriten' ) then begin
```

```
    vCtxItem # vCtxMenu->WinMenuItemAdd( 'Fav_Del', 'aus Favoriten entfernen' )
```

```
    vCtxItem->wpImageTile # _winImgDelete;
```

```

if ( StrFind( vMenuItem->wpCaption, '*', 1 ) > 0 ) then begin
    vCtxItem->wpDisabled # true;
end;

end

else begin

    vCtxItem # vCtxMenu->WinMenuItemAdd( 'Fav_Add', 'zu Favoriten hinzufügen' );

    vCtxItem->wplImageTile # _winImgImport;

end;


// Trennlinie hinzufügen

vCtxItem # vCtxMenu->WinMenuItemAdd( 'miMenuItem', " );

vCtxItem->wpMenuSeparator # true;


// Standarddefinition

if ( StrFind( vMenuItem->wpCaption, '*', 1 ) = 0 ) then begin

    vCtxItem # vCtxMenu->WinMenuItemAdd( 'Default_set', 'Als Standard definieren' );

    vCtxItem->wplImageTile # _winImgOk;

end

else begin

    vCtxItem # vCtxMenu->WinMenuItemAdd( 'Default_del', 'Standarddefinition aufheben' );

    vCtxItem->wplImageTile # _winImgMarkOneClear;

end;

end

else if ( vMenuItem->wpName = 'Favoriten' ) then begin

    vCtxItem # vCtxMenu->WinMenuItemAdd( 'Fav_Add_SFX', 'Sonderfunktion als Favorit hinzufügen' );

    vCtxItem->wplImageTile # _winImgImport;

```

end;

end;

//=====

// EvtMouse

//

//=====

sub EvtMouse(

 aEvt : event;

 aButton : int) : logic;

local begin

 vHdl : int;

 vOK : logic;

 vA : alpha(4096);

end;

begin

/**

debug('mouse:'+aEvt:Obj->wpname);

 vHdl # aEvt:obj;

//debug('clik auf '+vHdl->wpname);

 if (vHdl<>0) then begin

 if (WinInfo(vHdl,_Wintype)=_Wintypeedit) then begin

 if (vHDL->wpPopupType=_WinPopupList) then begin

 if (vHDL->wppopupopen) then begin

 xdebug('OPEN');

```

        mode # 'POPUP'+aint(vHdl);

        RETURN false;

    end

    else begin

        mode # c_modeview;

debug('go edit');

        RETURN false;

        end;

    end;

//    Erx # Wininfo(vHdl,_WinType);

//debug(vHdl->wpname+' '+aint(Erx));

        end;

    end;

***/

BugFix(__PROCFUNC__, aEvt:Obj);

vOK # true;

if (aEvt:obj->wpdisabled) then vOK # false;

if (aEvt:obj->wpColBkg=c_Collnactive) then vOK # false;


// if (gMdi<>0) then WinSearchPath(gMDI); // 21.11.2011


if (aButton = _winMousemiddle ) then begin // 31.05.2022 AH : in MASKE

    if (gPrefix<>") and (Mode=c_modeView) and (aEvt:Obj<>0) then begin

        if (gFile<>0) then begin

            try begin

                ErrTryIgnore(_rlocked,_rNoRec);

```



```

    ErrTryCatch(_ErrNoProcInfo,y);

    Call(gPrefix+'_Main:JumpTo', StrCnv(aEvt:Obj->wpname, _strUpper), RecBufDefault(gFile));

end;

end;

end;

end;

```

```

if (aButton=_Winmouseleft|_WinMouseDouble) then begin

    if (vOK) then begin

//    vRange # aEvt:obj->wprange;

//    if (vRange:min<>0) then todo('X');

        if (gMenu<>0) then begin

            gTmp # gMenu->WinSearch('Mnu.Auswahl');

            if (gTmp=0) then gTmp # gMenu->WinSearch('Mnu.SelAuswahl');

end;

            if (gTmp <> 0) then // ST 2011-05-23: nur machen wenn was gefunden wurde

                EvtMenuCommand(aEvt, gTMP);

end

else begin // 23.02.2022 AH

    case (WinInfo(aEvt:obj,_Wintype)) of

        _Wintypeedit :    vA # StrCut(aEvt:Obj->wpcaption,1,4096);

        _WintypeIntedit :    vA # aint(aEvt:Obj->wpcaptionint);

        _WintypeBigIntedit : vA # cnvab(aEvt:Obj->wpCaptionBigInt);

        _WintypeFloatedit : vA # cnvaf(aEvt:Obj->wpcaptionfloat);

        _WinTypeDecimalEdit : vA # cnvam(aEvt:Obj->wpCaptionDecimal);

```

```

_WintypeTimeedit :   vA # cnvat(aEvt:Obj->wpcaptionTime);

_WintypeDateedit :   vA # cnvad(aEvt:Obj->wpcaptionDate);

_WintypeTextedit :   vA # StrCut(aEvt:Obj->wpcaption,1,4096);

end;

if (vA<>") then ClipboardWrite(vA);

end;


end;


end;


//=====

// EvtMouseItem

//      Mausclicks in Listen

//=====

sub EvtMouseItem(

aEvt      : event;    // Ereignis

aButton   : int;      // Button

aHit      : int;      // Hitcode

altem     : int;      // Item

aID       : int;      // ID

) : logic

local begin

Erx      : int;

vFilter   : int;

```

```

vX      : int;

vChildmode : int;

vA      : alpha(500);

vBuf     : int;

end;

begin

if ( altem = 0 ) then RETURN false;

REIN('EvtMouseItem')

BugFix(__PROCFUNC__, aEvt:Obj);

// if (gMdi<>0) then WinSearchPath(gMDI); // 21.11.2011

// 19.02.2014 AH: JumpTo

if (aButton = _winMousemiddle ) and ( aHit = _winHitLstView ) and

((aEvt:obj = gZLList) or (gZLList=0)) then begin

if (aID=0) then RETURN true;

if (gPrefix<>") and (Mode=c_modeList) then begin

if (gFile<>0) then begin // RECLIST

vBuf # RecBufCreate(gFile);

Erx # RecRead(vBuf,0,_reclD, aID);

if (Erx<=_rLocked) then begin

try begin

ErrTryIgnore(_rlocked,_rNoRec);

ErrTryCatch(_ErrNoProcInfo,y);

```

```

    Call(gPrefix+'_Main:JumpTo', StrCnv(altem->wpname, _strUpper), vBuf);

end;

end;

RecBufDestroy(vBuf);

end

else begin  // DATALIST

    try begin

        ErrTryIgnore(_rlocked,_rNoRec);

        ErrTryCatch(_ErrNoProcInfo,y);

        Call(gPrefix+'_Main:JumpTo', StrCnv(altem->wpname, _strUpper), aID);

    end;

end;

RETURN true;

end;

end;

// Falls mit der linken Taste auf den Spaltenkopf geklickt wurde,

// Sortierung ändern (Aufruf der entsprechenden Unterfunktion)

if (aEvt:obj = gZLList) and (altem->wpDbKeyNo != 0) and (aHit = _winHitLstHeader) and

    ( aButton = _winMouseLeft | _winMouseDouble) or

    ((Dbalicense(_DbasrvLicense)='TE150086MN') and (aButton = _winMouseLeft))) then begin

// 21.07.2014: Knappstein will SINGLECLICK

if (gZLList->wpDbLinkFileNo = 0 ) /*and (gZLList->wpDbSelection=0)*/ then begin

    gKey # altem->wpDbKeyNo;

```

```

Lib_GuiCom:ZLSetSort(0, gKey);

// Sortiermenü setzen

vX # 0;

WHILE (vX<WinLstDatLineInfo($DL.Sort, _WinLstDatInfoCount)) do begin

    inc(vX);

    WinLstCellGet($DL.Sort, vA, 2, vX);

    if (StrCut(vA,1,8)=CnvAl(gKey,_fmtNumLeadZero,0,3)+'|'+CnvAl(gKey,_fmtNumLeadZero,0,3)+'|') then

        $DL.Sort->wpcurrentint # vX;

        gSuchProc # Str_Token(vA,'|',3);

        BREAK;

    end;

END;

/****

$DL.Sort->WinLstDatLineRemove(_WinLstDatLineAll);

vFilter # RecFilterCreate(901,2);

RecFilterAdd(vFilter,1,_FltAND, _FltEq, gFile);

FOR Erx # RecRead( 901, 2, _recFirst, vFilter );

LOOP Erx # RecRead( 901, 2, _recNext, vFilter );

WHILE ( Erx = _rOk ) DO BEGIN

    vX # vX + 1;

    $DL.Sort->WinLstDatLineAdd(Prg.Key.Name);

// 10.02.2020    if (Prg.Key.EchteKeyNr=0) then

//        vA # CnvAl(Prg.Key.Key,_fmtNumLeadZero,0,3)+'|'+Prg.Key.SuchProzedur

//    else

//        vA # CnvAl(Prg.Key.Key,_fmtNumLeadZero,0,3)+'|'+Prg.Key.SuchProzedur;

    if (Prg.Key.EchteKeyNr=0) then Prg.Key.EchteKeyNr # Prg.Key.Key;

```

```

vA # CnvAl(Prg.Key.Key,_fmtNumLeadZero,0,3)+'|'+CnvAl(Prg.Key.EchteKeyNr,_fmtNumLeadZero,0,3);
$DL.Sort->WinLstCellSet(vA,2,_WinLstDatLineLast);
if ((gKeyID<>0) and (gKeyID=Prg.Key.EchteKeyNr)) or
  ((gKeyID=0) and (gKey=Prg.Key.Key)) then begin
  $DL.Sort->wpcurrentint # vX;
  gSuchProc # Prg.Key.SuchProzedur;
end;
END;
RecFilterDestroy(vFilter);
***/

RAUS('EvtMouseItem')
RETURN true;

end;

end;

if (aButton = _WinMouseLeft | _winMouseCtrl) and
  (Mode=c_ModeList) then begin
  if (aID<>0) then begin
    if (gDataList<>0) then begin
      Mark(gDataList);
    end
  else begin
    Erx # RecRead(gFile,0,_RecId,aID);
    Mark();
  end
end

```

```

end;

RAUS('EvtMouseItem')

RETURN true;

end;

end;

if (aButton & _winmouseright > 0 ) then begin

    if (Mode=c_ModeList) then begin

        /**

            if (altem<>0) and (aHit = _WinHitLstHeader) then begin

                if (altem->wpFmtPostComma<5) then

                    altem->wpFmtPostComma # altem->wpFmtPostComma + 1

                else

                    altem->wpFmtPostComma # 0;

            end;

        ***/

        if (gPrefix<>") then begin

            try begin

                ErrTryIgnore(_rlocked,_rNoRec);

                ErrTryCatch(_ErrNoProcInfo,y);

                Call(gPrefix+'_Main:EvtMouseItem',aEvt,aButton,aHit,altem, aID);

            end;

            //    ErrTryCatch(_ErrNoProcInfo | _ErrNoArgument,false);

        end;

        RAUS('EvtMouseItem')

```

```

    RETURN(true);

end;

end;

/* AB HIER NUR NOCH DOPPELCLICK */

if ((aButton & _WinMouseDouble)=0) then begin

    RAUS('EvtMouseItem')

    RETURN false;

end;


// Datensatz aus Liste gewählt -> ENTER

if (aEvt:Obj=gZLList) or (aEvt:Obj->wpname='ZL.Erfassung') then begin

    // wäre auch über Mode prüfbar...

    if (aEvt:Obj=gZLList) then

        Erx # gZLList->wpdbRecId

    else begin

        Erx # $ZL.Erfassung->wpdbRecId;

        if (Erx<>0) and (gFile<>0) then RecRead(gFile,0,_recId, aID); // 2022-12-19 AH

    end;

    if (Erx=0) then begin

        RAUS('EvtMouseItem')

        RETURN true;

    end;

    // SELECT in Auswahl -> Satz übernehmen

```



```
if (w_AuswahlMode) then begin
```

```
  gSelected # aID;
```

```
  gMdi->Winclose();
```

```
  RAUS('EvtMouseItem')
```

```
  RETURN true;
```

```
end;
```

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

```
  // SELECT in Liste -> Ansichtsmodus
```

```
  Action(c_ModeView);
```

```
end;
```

```
end;
```

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

```
'TV.HauptmenueFM' : begin
```

```
  case (altem -> wpname) of
```

```
    'tnfertmeld' : begin
```

```
      // msg(700900,altem -> wpname,0,0,0);
```

```
      StartMdi('BAG.FertMeld.User');
```

```
    end;
```

```
  end;
```

```
end;
```

```
'TV.Hauptmenue': begin
```

case (altem->wpName) of

'AdV':

StartMdi('ADV.Verwaltung');

'MVA':

StartMdi('MVA.Verwaltung');

'VZN':

StartMdi('VZN.Verwaltung');

'VBP':

StartMdi('VBP.Verwaltung');

'Pakete':

StartMdi('Pak.Verwaltung');

'Grobplanung' :

StartMdi('GPI.Verwaltung');

'BAG.Planung' :

StartMdi('BAG.Planung');

'RSO.Planung' :

StartMdi('RSO.Planung');

'BAG' :

```
StartMdi('BAG.Verwaltung');
```

```
'BAG.Output' :
```

```
StartMdi('BAG.Output');
```

```
'tnrsobelegung' : begin
```

```
    //sartMdi("");
```

```
//    gMDI # WinOpen('Rso.Bel.Verwaltung');
```

```
    gMDI # WinOpen('Rso.Bel.Verwaltung_neu');
```

```
//    Lib_GuiCom:AddChildWindow(gFrmMain, gMDI,"vMode);
```

```
    gFrmMain -> WinAdd(gMDI, _WinAddHidden);
```

```
    gMdi->WinUpdate(_WinUpdOn);
```

```
end;
```

```
'Arbeitsgaenge':
```

```
StartMdi('ArG.Verwaltung');
```

```
'Hauptmenue':
```

```
StartMdi('Hauptmenue');
```

```
'Adressen':
```

```
StartMdi('Adr.Verwaltung');
```

```
'Ansprechpartner':
```

```
StartMdi('Adr.P.Verwaltung');
```

'Projekte':

StartMdi('Prj.Verwaltung');

'Auf_P':

StartMdi('Auf.P.Verwaltung');

'Ablage_Auf_P':

StartMdi('Auf.P.Ablage');

'Auf':

StartMdi('Auf.Verwaltung');

'LFS':

StartMdi('Lfs.Verwaltung');

'Versand':

StartMdi('Vsd.Verwaltung');

'Versandpool':

StartMdi('VsP.Verwaltung');

'VertreterVerbaende' :

StartMdi('Ver.Verwaltung');

'Artikel':

StartMdi('Art.Verwaltung');

'Artikelstruktur' :

StartMdi('Art.SLK.Verwaltung');

'Artikelpreise':

StartMdi('Art.P.Verwaltung');

'Material':

StartMdi('Mat.Verwaltung');

'Ablage_Material':

StartMdi('Mat.Ablage');

'Materialstruktur':

StartMdi('Msl.Verwaltung');

'Materialanalyse':

StartMdi('Lys.K.Verwaltung');

'Ressourcen':

StartMdi('Rso.Verwaltung');

'HuB':

StartMdi('HuB.Verwaltung');

'EK_P':

StartMdi('Ein.P.Verwaltung');

'Ablage_EK_P':

StartMdi('Ein.P.Ablage');

'HuB.EK':

StartMdi('HuB.EK.Verwaltung');

'SWE':

StartMdi('SWe.Verwaltung');

'EK_Bedarf':

StartMdi('Bdf.Verwaltung');

'Ablage_EK_Bedarf' :

StartMdi('Bdf.Ablage');

'Gantt.Termine' :

StartMdi('Gantt.TeM.Woche');

'calculator' :

StartMdi('MathCalculator');

'Mathematik':

StartMdi('Math.Verwaltung');

'Mathematikalphabet' :

startmdi('Math.Alphabet.Verwaltung');

'Mathematikvarminiprg' :

startmdi('Math.Alphabet.MiniPrg');

'DB.Info' :

C16_Info();

'DB.Schluesselreorg' :

if (Msg(998011,",_WinlcoWarning,_WinDialogYesNo,2)=_WinIdYes) then begin

App_Extras:KeyReorg();

end;

'DB.Diagnose' : begin

if (DbalInfo(_DbaUserCount)>1) then begin

Msg(998008,"0,0,0,0);

RAUS('EvtMouseItem')

RETURN true;

end;

if (Msg(998009,",_WinlcoWarning,_WinDialogYesNo,2)=_WinIdYes) then begin

_app->wpvisible # true;

CallOld('old_Diagnostic',0+4+8+64); // Std + Recover + Key (64=ext.)

_app->wpvisible # false;

end;

end;

'DB.Optimierung' : begin

```
if (DbInfo(_DbidUserCount)>1) then begin
```

```
    Msg(998008,"",0,0,0);
```

```
    RAUS('EvtMouseItem')
```

```
    RETURN true;
```

```
end;
```

```
if (Msg(998010,"",_WinIcoWarning,_WinDialogYesNo,2)=_WinIdYes) then begin
```

```
    _app->wpvisible # true;
```

```
    CallOld('old_Diagnostic',16);  // Opt1/ 32=Opt2
```

```
    _app->wpvisible # false;
```

```
end;
```

```
end;
```

```
'Abteilungen':
```

```
    StartMdi('Abt.Verwaltung');
```

```
'Anreden':
```

```
    StartMdi('Anr.Verwaltung');
```

```
'Artikelgruppen':
```

```
    StartMdi('Agr.Verwaltung');
```

```
'Aufpreise':
```

```
    StartMdi('Apl.Verwaltung');
```

```
'Auftragsarten':
```

```
    StartMdi('AAr.Verwaltung');
```


'bds':

StartMdi('BDS.Verwaltung');

'Dickentoleranzen':

StartMdi('Tol.D.Verwaltung');

'Fehlercodes':

StartMdi('FhC.Verwaltung');

'Gegenkonten':

StartMdi('GKo.Verwaltung');

'Gruppen':

StartMdi('Grp.Verwaltung');

'Kalkulationen':

StartMdi('Kal.Verwaltung');

'Kostenstellen':

StartMdi('KSt.Verwaltung');

'Laender':

StartMdi('Lnd.Verwaltung');

'Orte':

StartMdi('Ort.Verwaltung');

'Lagerplaetze':

StartMdi('LPl.Verwaltung');

'Lieferbedingungen':

StartMdi('LiB.Verwaltung');

'Materialstatus':

StartMdi('MSt.Verwaltung');

'Oberflaechen':

StartMdi('Obf.Verwaltung');

'Rabatte':

StartMdi('Rab.Verwaltung');

'Rechnungstypen':

StartMdi('RTy.Verwaltung');

'Reklamationsarten':

StartMdi('Rek.Art.Verwaltung');

'Reklamationen':

StartMdi('Rek.P.Verwaltung');

'LfE':

StartMdi('LfE.Verwaltung');

'WoF.Schema':

StartMdi('WoF.Sch.Verwaltung');

'Reklamationstexte':

StartMdi('Rek.8.Verwaltung');

'Steuerschluessel':

StartMdi('StS.Verwaltung');

'Skizzen':

StartMdi('Skz.Verwaltung');

'Vorgangsstatus':

StartMdi('VgSt.Verwaltung');

'Zahlungsbedingungen':

StartMdi('ZaB.Verwaltung');

'Zahlungsarten':

StartMdi('ZhA.Verwaltung');

'Versandarten':

StartMdi('VsA.Verwaltung');

'Verwiegungsarten':

StartMdi('VwA.Verwaltung');

'Ressourcengruppen':

StartMdi('Rso.Grp.Verwaltung');

'IHA.Massnahmen':

StartMdi('IHA.Mas.Verwaltung');

'IHA.Meldungen':

StartMdi('IHA.Mld.Verwaltung');

'IHA.Ursachen':

StartMdi('IHA.Urs.Verwaltung');

'Kalendertage' :

startMdi('Rso.KalTage.Verwaltung');

'Termintypen':

StartMdi('TTy.Verwaltung');

'Waehrungen':

StartMdi('Wae.Verwaltung');

'Warengruppen':

StartMdi('Wgr.Verwaltung');

'Artikelzustaende':

StartMdi('Art.Zst.Verwaltung');

'Userliste':

StartMdi('Usr.Verwaltung');

'Usergruppen':

StartMdi('Usr.G.Verwaltung');

'Schluessel':

StartMdi('Prg.Key.Verwaltung');

'Hilfedateien':

StartMdi('Prg.Help.Verwaltung');

'Nummernkreise' :

StartMdi('Prg.Nr.Verwaltung');

'Texte' :

StartMdi('Txt.Verwaltung');

'Adr.Dok.Struktur' :

StartMdi('Bib.Verwaltung');

'Ost' :

StartMdi('OSt.Verwaltung');

'Controlling':

StartMdi('Con.Verwaltung');

'Erloese' :

StartMdi('Erl.Verwaltung');

'OffenePosten' :

StartMdi('OfP.Verwaltung');

'OffenePosten.Ablage' :

StartMdi('OfP.Ablage');

'Zahlungsein' :

StartMdi('ZEi.Verwaltung');

'Zahlungsaus' :

StartMdi('ZAu.Verwaltung');

'Fixkosten' :

StartMdi('FxK.Verwaltung');

'Kasse' :

StartMdi('Kas.Verwaltung');

'Kostenbuchung' :

StartMdi('Kos.K.Verwaltung');

'EKK' :

StartMdi('EKK.Verwaltung');

'Mat.Lagergeld.Kunde' :

if (Rechte[Rgt_Fin_Mat_LagerKunde]) then Mat_Subs:LagergeldKunde();

'Mat.Lagergeld.Fremd' :

if (Rechte[Rgt_Fin_Mat_LagerFremd]) then Mat_Subs:LagergeldFremd();

'Mat.Zinsen' :

if (Rechte[Rgt_Fin_Mat_Zinsen]) then Mat_Subs:Zinsen();

'Verbindlichkeiten' :

StartMDI('Vbk.Verwaltung');

'Eingangsrechnung' :

StartMDI('ERe.Verwaltung');

'Arbeitsplanvorlagen' :

startMdi('Arb.Plan.Vorlagen.Verwalt');

'Arbeitsvorgaben' :

startMdi('Arb.Plan.Aktionenvorgabe');

'Unterlagen' :

startMdi('ULa.Verwaltung');

'Etiketten' :

startMdi('Eti.Verwaltung');

'Zeugnisse' :

startMdi('Zeu.Verwaltung');

'Reorg.EK' : begin

if (Rechte[Rgt_Abl_Ein_Reorg]) then begin

Ein_Abl_Data:Reorganisation(n);

end;

end;

'Reorg.VK' : begin

if (Rechte[Rgt_Abl_Auf_Reorg]) then begin

Auf_Abl_Data:Reorganisation(n);

end;

end;

'Reorg.Mat' : begin

if (Rechte[Rgt_Abl_Mat_Reorg]) then begin

Mat_Abl_Data:Reorganisation(n);

end;

end;

'Reorg.OfP' : begin

if (Rechte[Rgt_Abl_OfP_Reorg]) then begin

OfP_Abl_Data:Reorganisation(n);

end;

end;

'Protokoll' :

PtD_Main:ManageStatus();

'Settings':

Set_Main:Verwaltung();

'Uebersetzungen' :

StartMdi('Translate.Verwaltung');

'Listen':

StartMdi('Lfm.Verwaltung');

'Druckzonen':

StartMdi('Dzo.Verwaltung');

'XML':

if (Rechte[Rgt_Ex_Importe]) then begin

XML_Main();

end;

'Customfelder':

StartMdi('CUS.FP.Verwaltung');

'Customrechte':

StartMdi('Usr.CR.Verwaltung');

'Customauswahlfelder':

//StartMdi('CUS.AF.Verwaltung');

StartMdi('CUS.AF.Verwaltung2');

'Dialoge':

StartMdi('Dia.Verwaltung');

'Formulare':

StartMdi('Frm.Verwaltung');

'Scripte':

StartMdi('Scr.Verwaltung');

'Ankerfunktionen':

StartMdi('AFX.Verwaltung');

'Serviceinventar' :

StartMdi('SVi.Verwaltung');

'Sonderfunktionen':

```
StartMdi('SFX.Verwaltung');
```

```
'Aktivitaeten':
```

```
StartMdi('TeM.Verwaltung');
```

```
'Inventur':
```

```
if (Rechte[Rgt_Art_Inventur]) then begin
```

```
Art_Inv_Main:Start(true);
```

```
end;
```

```
'Dashboard':
```

```
StartMdi('Mdi.Dashboard');
```

```
'Termine' :
```

```
StartMdi('TeM.Verwaltung.Termine');
```

```
'DokumentAblage' : begin
```

```
RecBufClear(915);
```

```
gDokTyp # altem->wpCustom;
```

```
StartMdi('Dok.Verwaltung');
```

```
end;
```

```
'Passwortaendern': begin
```

```
Call('C16_UserPassw:Init');
```

```
end;
```

'Benutzerwechsel': begin

if (gUsergroup='USER') then begin

 vA # NetInfo(_NtiAddressServer);

 vA # vA + ' ' + DbName(_DbAreaAlias)+''';

 sysexecute(Set.Client.Pfad+'c16_Winc.exe',vA,0);//_ExecMaximized);

 gFrmMain->winclose();

end;

RETURN true;

end;

'Beenden': begin

 gFrmMain->winclose();

end;

'Kommandozeile' : begin

 App_Extras:Kommandozeile();

end;

'Helpdesk' :

 App_Extras:Helpdesk();

'Jobs' : begin

 StartMdi('Job.Verwaltung');

end;

'JobError' : begin

```
    StartMdi('Job.Err.Verwaltung');  
end;
```

```
'Filescanpfade': begin  
    StartMdi('FSP.Verwaltung');  
end;
```

```
'Abmessungstoleranzen' : begin  
    StartMdi('MTo.Verwaltung');  
end;
```

```
'Qualitaeten' : begin  
    StartMdi('MQu.Verwaltung');  
end;
```

```
'Qualitaetsstufen' : begin  
    StartMdi('MQu.S.Verwaltung');  
end;
```

```
'Betrieb' : begin  
    StartMdi('Betrieb');  
end;
```

```
'Zeitentypen' : begin  
    StartMdi('ZTy.Verwaltung');  
end;
```

```

otherwise begin

    if (StrCnv(StrCut(altem->wpName, 1, 3),_strupper) = 'SFX' ) then begin

        Lib_SFX:Run( CnvIA( altem->wpName ) );

    end;

end;

end;

end;

end;

otherwise begin

end;

end;

RAUS('EvtMouseItem')

RETURN true;

end;

//=====

// EvtKeyItem

//      Keyboard in RecList/DataList

//=====

sub EvtKeyItem(

    aEvt          : event;    // Ereignis

    aKey          : int;      // Taste

    aRecID        : int;      // RecID

```

) : logic

local begin

 erx : int;

 vChildMode : int;

 vHdl : handle;

 vTmp : int;

end;

begin

 REIN('EvtKeyItem')

 BugFix(__PROCFUNC__, aEvt:Obj);

// if (gMdi<>0) then WinSearchPath(gMDI); // 21.11.2011

 // Quickbar

 if (w_QBButton<>0) then begin

 RAUS('EvtKeyItem');

 RETURN true;

 end;

 // ENTER im Menübaum -> Doppelclick simulieren

 if (aKey=_WinKeyReturn) and (aEvt:obj->wpname='TV.Hauptmenue') then begin

 vTmp # \$TV.Hauptmenue->wpCurrentInt;

 EvtMouseItem(aEvt,_WinMouseDouble,0,vTmp,0);

 RAUS('EvTKeyItem')

 RETURN true;

 end;

```
if (aKey=_WinKeyReturn) and ((Mode=c_ModeList) or (Mode=c_ModeEdList) or (Mode=c_ModeList2)) then
```

```
  if (Mode=c_ModeList) or (Mode=c_modeEdList) then
```

```
    Erx # gZLList->wpdbRecId
```

```
  else
```

```
    Erx # $ZL.Erfassung->wpdbRecId;
```

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

```
    RAUS('EvTKeyItem')
```

```
    RETURN true;
```

```
  end;
```

```
  // RETURN-Taste in Auswahl -> Satz übernehmen
```

```
  vTmp # gMDI->winsearch('NB.Main');
```

```
  if (w_AuswahlMode) then begin
```

```
    gSelected # aRecID;
```

```
    gMDI->Winclose();
```

```
    RAUS('EvTKeyItem')
```

```
    RETURN true;
```

```
  end;
```

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

```
    // RETURN-Taste in Liste -> Ansichtsmodus
```

```
    Action(c_ModeView);
```

```
  end;
```

```
end
```

```
else if (akey=_WinKeyInsert) and (Mode=c_ModeList) then begin
```

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

```
    Mark(gDataList);
```

```
  end
```



```

else if (gFile<>0) and (gZLList->wpdbrecid<>0) then begin

    RecRead(gFile,0,0,gZLList->wpdbrecid);

    Mark();

end;

end

else if (akey=_WinKeyDelete) and (Mode=c_ModeList) then begin

    if (gMenu<>0) then begin

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

        if (vHdl<>0) then begin

            if (vHdl->wpdisabled=false) and

                (gFile<>0) and (gZLList->wpdbrecid<>0) then begin

                    RecRead(gFile,0,0,gZLList->wpdbrecid);

                    Action(c_ModeDelete);

                end;

            end;

        end;

    end;

end

else begin

    if (gPrefix<>") then begin

        try begin

            ErrTryIgnore(_rlocked,_rNoRec);

            ErrTryCatch(_ErrNoProcInfo,y);

            ErrTryCatch(_ErrNoSub,y);

            Call(gPrefix+'_Main:EvtKeyItem',aEvt,akey,aRecId);

        end;

        // ErrTryCatch(_ErrNoProcInfo | _ErrNoArgument,false);

```

```

    end;

end;

RAUS('EvTKeyItem')

RETURN true;

end;


//=====

// EvtMenuCommand

//      Menüpunkt aufgerufen

//=====

sub EvtMenuCommand (

    aEvt      : event;    // Ereignis

    aMenuItem : int      // Menüeintrag

) : logic

local begin

    erx      : int;

    vName    : alpha(250);

    vFilter   : int;

    vHdl      : int;

    vHdlNode  : int;

    vRecFlag  : int;

    vl       : int;

    vA       : alpha;

    vFile    : int;

```

```

vQ      : alpha(4000);

vPref   : alpha;

vOK      : logic;

vCMD     : alpha;

end;

begin

    REIN('EvtMenuCommand')

// BugFix(__PROCFUNC__, aEvt:Obj);

    Crit_Prozedur:Manage();

    if (aMenuItem=0) then begin

        RAUS('EvtMenuCommand')

        RETURN true;

    end;

// 17.07.2018 AH:

    if (VarInfo(windowbonus)>0) then begin

        if (gMenuEvtProc<>") then begin

            if (Call(gMenuEvtProc,aEvt, aMenuItem)) then begin

                RAUS('EvtMenuCommand')

                RETURN true;

            end;

        end;

    end;

```

```

end;

// if (gMdi<>0) then WinSearchPath(gMDI); // 21.11.2011

// BUGFIX WinSearchPath(aEvt:Obj); // 12.05.2014


// quickbar

if (aMenuItem->wpname='Mnu.Quickbar.Config') then begin

  if (mode=c_modeList) and (w_QBButton=0) then begin

    //Lib_GuiCom:SetQBButton($bt.Quick1);

    Lib_GuiCom:SetQBButton(1);

  end

  else if (w_QBButton<>0) then begin

    Lib_GuiCom:SetQBButton(0);

    // Resize durchführen:

    vHdl # Winsearch(gMDI, 'fc.Main');

    if (vHdl<>0) then begin

//      App_Main:EvtPosChanged(aEvt, vHdl->wparea, PointMake(0,0), _WinPosSized);

      vA # WinEvtProcNameGet(vHdl, _WinEvtPosChanged);

      if (vA<>") then call(vA, aEvt, vHdl->wparea, PointMake(0,0), _WinPosSized);

    end;

  end;

  RETURN true;

end;

if (w_QBButton=1) then begin

```

```

RAUS('EvtMenuCommand')

RETURN true;

end;

if (w_QBButton>1) then begin

    vHdl # Wininfo(aMenuItem,_Winparent);

    if (vHdl<>0) then

        if (vHdl->wpname='TB.User') then begin

            RAUS('EvtMenuCommand')

            RETURN true;

        end;

    vA  # aMenuItem->wpcaption;

    vCMD # aMenuItem->wpname;

// 11.10.2019

    if (aMenuItem->wpName='Listendruck') or

        (aMenuItem->wpName='Listen') or

        (aMenuItem->wpName='Mnu.Listen') then begin

        if (aMenuItem->wpCustom<>'') then

            vOK # Lfm_Ausgabe:Auswahl(aMenuItem->wpCustom, true)

        else

            vOK # Lfm_Ausgabe:Auswahl(w_Listen, true);

        if (vOK=false) then RETURN true;

        vA  # 'List '+Lfm.Name;

        vCMD # 'LFM;'+Lfm.Kuerzel+';'+aint(Lfm.Nummer);

    end;

```

```
vA # Str_ReplaceAll(vA, '&', '');  
  
Dlg_Standard:Standard(Translate('Beschriftung'), var vA);  
  
w_QBButton->wpcaption # vA;  
  
w_QBButton->wpcustom # vCMD;  
  
RAUS('EvtMenuCommand')  
  
RETURN true;  
  
end;
```

```
// Sonderfunktion?
```

```
if (StrFind(aMenuItem->wpName,'SFX.',1)=1) then begin  
  
    if (Mode=c_ModeList) and (gFile<>0) and (gZLList<>0) then  
  
        RecRead(gFile,0,0,gZLList->wpdbrecid);  
  
    Lib_SFX:Run(cnvia(aMenuItem->wpname));  
  
    RAUS('EvtMenuCommand')  
  
    RETURN true;  
  
end;
```

```
case (aMenuItem->wpName) of
```

```
    'Mnu.Mdi.Reset' : Lib_GuiCom2:Mdis.Reset();
```

```
// 22.05.2020
```

```
'Mnu.NextPage' : App_Main_Sub:NextPage(gMdi->Winsearch('NB.Main'));
```

```
'Mnu.PrevPage' : App_Main_Sub:PrevPage(gMdi->Winsearch('NB.Main'));
```

```
'Mnu.Help' : Help_Main:Show(gMDI->wpname);
```

```
'Userstatus' : begin
```

```
    Org_Main:SetUserStatus();
```

```
end;
```

```
'Mnu.Comment' : begin
```

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

```
        if (Mode=c_ModeList) and (gZLList<>0) then begin
```

```
            RecRead(gFile,0,0,gZLList->wpdbrecid);
```

```
        end;
```

```
        Anh_Data:Comment(gFile, (gZLList->wpdbrecid)=0);
```

```
    if (gFile=916) then begin
```

```
        SelRecInsert(gZLList->wpDbSelection, 916);
```

```
        RefreshList(gZllist, _WinLstRecFromRecid | _WinLstRecDoSelect);
```

```
    end;
```

```
end;
```

```
end;
```

```
'Mnu.Auswahl' : begin
```

```
    vHdl # WinFocusGet();
```

```
    vA # ";
```

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

```

vA # Lib_Pflichtfelder:TypAuswahlFeld(vHdl);

if (StrCut(vA,1,1)='#') then begin
    vA # StrCut(vA,2,250);
    CUS_AF_Main:Popup(vA,vHdl);
    vA # "";
end

else if (vA<>") then begin
    try begin
        ErrTryIgnore(_rlocked,_rNoRec);
        ErrTryCatch(_ErrNoProcInfo,y);
        ErrTryCatch(_ErrNoSub,y);
        if (gPrefix<>") then Call(gPrefix+'_Main:Auswahl',vA);
    end
end;

end;

if (vA="") then begin
    try begin
        ErrTryIgnore(_rlocked,_rNoRec);
        ErrTryCatch(_ErrNoProcInfo,y);
        ErrTryCatch(_ErrNoSub,y);
        if (gPrefix<>") then Call(gPrefix+'_Main:EvtMenuCommand',aEvt, aMenuItem);
    end
end;

end;

```


'Mnu.StartWOF' : begin

 Erx # Msg(99,'Soll ein dynamischer Workflow gestartet werden?','_WinIcoQuestion','_WinDialogYesNoC

 if (Erx=_WinIdCancel) then RETURN true;

 if (Erx=_WinIdyes) then

 Lib_WorkFlow:DynamischerWof(cnvia(aMenuItem->wpcustom), "", "")

 else

 Lib_WorkFlow:AuswahlWof(cnvia(aMenuItem->wpcustom));

end;

'DEBUG' :

 Lib_debug:Button();

'DEBUG2' :

 Lib_debug:Button2();

'AdV':

 StartMdi('ADV.Verwaltung');

'MVA':

 StartMdi('MVA.Verwaltung');

'VZN':

 StartMdi('VZN.Verwaltung');

'VBP':

 StartMdi('VBP.Verwaltung');

'Hauptmenue':

```
// gFrmMain->wpMenuname # 'Main';//gMenuName; // Menü setzen
```

```
StartMdi('Hauptmenue');
```

'Pakete':

```
StartMdi('Pak.Verwaltung');
```

'Grobplanung' :

```
StartMdi('GPI.Verwaltung');
```

'BAG.Planung' :

```
StartMdi('BAG.Planung');
```

'RSO.Planung' :

```
StartMdi('RSO.Planung');
```

'BAG' :

```
StartMdi('BAG.Verwaltung');
```

'BAG.Output' :

```
StartMdi('BAG.Output');
```

'Adressen':

```
StartMdi('Adr.Verwaltung');
```

'Ansprechpartner':

StartMdi('Adr.P.Verwaltung');

'Projekte':

StartMdi('Prj.Verwaltung');

'Auf_P':

StartMdi('Auf.P.Verwaltung');

'Ablage_Auf_P':

StartMdi('Auf.P.Ablage');

'Auf':

StartMdi('Auf.Verwaltung');

'LFS':

StartMdi('Lfs.Verwaltung');

'Versand':

StartMdi('Vsd.Verwaltung');

'Versandpool':

StartMdi('VsP.Verwaltung');

'VertreterVerbaende' :

StartMdi('Ver.Verwaltung');

'Artikel':

StartMdi('Art.Verwaltung');

'Artikelstruktur':

StartMdi('Art.SLK.Verwaltung');

'Artikelpreise':

StartMdi('Art.P.Verwaltung');

'Material':

StartMdi('Mat.Verwaltung');

'Ablage_Material':

StartMdi('Mat.Ablage');

'Materialstruktur':

StartMdi('Msl.Verwaltung');

'Materialanalyse':

StartMdi('Lys.K.Verwaltung');

'Ressourcen':

StartMdi('Rso.Verwaltung');

'Arbeitsgaenge':

StartMdi('ArG.Verwaltung');

'HuB':

StartMdi('HuB.Verwaltung');

'Gantt.Termine' :

StartMdi('Gantt.TeM.Woche');

'Notifier' :

if (Set.TimerYN) then

StartMdi('Notifier');

'calculator' :

StartMdi('MathCalculator');

'Mathematik':

StartMdi('Math.Verwaltung');

'Mathematikalphabet' :

Startmdi('Math.Alphabet.Verwaltung');

'Mathematikvarminiprg' :

startmdi('Math.Alphabet.MiniPrg');

'EK_P':

StartMdi('Ein.P.Verwaltung');

'Ablage_EK_P':

StartMdi('Ein.P.Ablage');

'SWE':

StartMdi('SWe.Verwaltung');

'EK_Bedarf':

StartMdi('Bdf.Verwaltung');

'Ablage_EK_Bedarf' :

StartMdi('Bdf.Ablage');

'HuB_EK':

StartMdi('HuB.EK.Verwaltung');

'DB.Info' :

C16_Info();

'DB.Schluesselreorg' :

if (Msg(998011,","_WinlcoWarning,_WinDialogYesNo,2)=_WinIdYes) then begin

App_Extras:KeyReorg();

end;

'DB.Diagnose' : begin

if (DbalInfo(_DbidUserCount)>1) then begin

Msg(998008,"0,0,0,0);

RAUS('EvtMenuCommand')

```

RETURN true;

end;

if (Msg(998009,",_WinIcoWarning,_WinDialogYesNo,2)=_WinIdYes) then begin
    _app->wpvisible # true;

    CallOld('old_Diagnostic',0+4+8+64);  // Std + Recover + Key (64=ext.)

    _app->wpvisible # false;

end;

end;

```

```

'DB.Optimierung' : begin

if (DbalInfo(_DbidUserCount)>1) then begin

    Msg(998008," ,0,0,0);

    RAUS('EvtMenuCommand')

    RETURN true;

end;

if (Msg(998010,",_WinIcoWarning,_WinDialogYesNo,2)=_WinIdYes) then begin

    _app->wpvisible # true;

    CallOld('old_Diagnostic',16);  // Opt1/ 32=Opt2

    _app->wpvisible # false;

end;

end;

```

```

'Abteilungen':

StartMdi('Abt.Verwaltung');

```

```

'Anreden':

```

StartMdi('Anr.Verwaltung');

'Artikelgruppen':

StartMdi('Agr.Verwaltung');

'Aufpreise':

StartMdi('Apl.Verwaltung');

'Auftragsarten':

StartMdi('AAr.Verwaltung');

'bds':

StartMdi('BDS.Verwaltung');

'Dickentoleranzen':

StartMdi('Tol.D.Verwaltung');

'Fehlercodes':

StartMdi('FhC.Verwaltung');

'Gegenkonten':

StartMdi('GKo.Verwaltung');

'Gruppen':

StartMdi('Grp.Verwaltung');

'Kalkulationen':

StartMdi('Kal.Verwaltung');

'Kostenstellen':

StartMdi('KSt.Verwaltung');

'Laender':

StartMdi('Lnd.Verwaltung');

'Orte':

StartMdi('Ort.Verwaltung');

'Lagerplaetze':

StartMdi('LPl.Verwaltung');

'Lieferbedingungen':

StartMdi('LiB.Verwaltung');

'Materialstatus':

StartMdi('MSt.Verwaltung');

'Oberflaechen':

StartMdi('Obf.Verwaltung');

'Rabatte':

StartMdi('Rab.Verwaltung');

'Rechnungstypen':

StartMdi('RTy.Verwaltung');

'Reklamationsarten':

StartMdi('Rek.Art.Verwaltung');

'Reklamationen':

StartMdi('Rek.P.Verwaltung');

'LfE':

StartMdi('LfE.Verwaltung');

'WoF.Schema':

StartMdi('WoF.Sch.Verwaltung');

'Reklamationstexte':

StartMdi('Rek.8.Verwaltung');

'Steuerschluessel':

StartMdi('StS.Verwaltung');

'Skizzen':

StartMdi('Skz.Verwaltung');

'Vorgangsstatus':

StartMdi('VgSt.Verwaltung');

'Zahlungsbedingungen':

StartMdi('ZaB.Verwaltung');

'Zahlungsarten':

StartMdi('ZhA.Verwaltung');

'Versandarten':

StartMdi('VsA.Verwaltung');

'Verwiegungsarten':

StartMdi('VwA.Verwaltung');

'Ressourcengruppen':

StartMdi('Rso.Grp.Verwaltung');

'IHA.Massnahmen':

StartMdi('IHA.Mas.Verwaltung');

'IHA.Meldungen':

StartMdi('IHA.Mld.Verwaltung');

'IHA.Ursachen':

StartMdi('IHA.Urs.Verwaltung');

'Kalendertage' :

startMdi('Rso.KalTage.Verwaltung');

'Termintypen':

StartMdi('TTy.Verwaltung');

'Waehrungen':

StartMdi('Wae.Verwaltung');

'Warengruppen':

StartMdi('Wgr.Verwaltung');

'Artikelzustaende':

StartMdi('Art.Zst.Verwaltung');

'Userliste':

StartMdi('Usr.Verwaltung');

'Usergruppen':

StartMdi('Usr.G.Verwaltung');

'Schluessel':

StartMdi('Prg.Key.Verwaltung');

'Hilfdateien':

StartMdi('Prg.Help.Verwaltung');

'Nummernkreise' :

StartMdi('Prg.Nr.Verwaltung');

'Texte' :

StartMdi('Txt.Verwaltung');

'Adr.Dok.Struktur' :

StartMdi('Blb.Verwaltung');

'OSt' :

StartMdi('OSt.Verwaltung');

'Controlling':

StartMdi('Con.Verwaltung');

'Erloese' :

StartMdi('Erl.Verwaltung');

'OffenePosten' :

StartMdi('OfP.Verwaltung');

'OffenePosten.Ablage' :

StartMdi('OfP.Ablage');

'Zahlungsein' :

StartMdi('ZEi.Verwaltung');

'Zahlungsaus' :

StartMdi('ZAu.Verwaltung');

'Fixkosten' :

StartMdi('FxK.Verwaltung');

'Kasse' :

StartMdi('Kas.Verwaltung');

'Kostenbuchung' :

StartMdi('Kos.K.Verwaltung');

'EKK' :

StartMdi('EKK.Verwaltung');

'Mat.Lagergeld.Kunde' :

if (Rechte[Rgt_Fin_Mat_LagerKunde]) then Mat_Subst:LagergeldKunde();

'Mat.Lagergeld.Fremd' :

if (Rechte[Rgt_Fin_Mat_LagerFremd]) then Mat_Subst:LagergeldFremd();

'Mat.Zinsen' :

if (Rechte[Rgt_Fin_Mat_Zinsen]) then Mat_Subst:Zinsen();

'Verbindlichkeiten' :

StartMDI('Vbk.Verwaltung');

'Eingangsrechnung' :

StartMDI('ERe.Verwaltung');

'Arbeitsplanvorlagen' :

StartMdi('Arb.Plan.Vorlagen.Verwalt');

'Arbeitsvorgaben' :

StartMdi('Arb.Plan.Aktionenvorgabe');

'Unterlagen' :

startMdi('ULa.Verwaltung');

'Etiketten' :

startMdi('Eti.Verwaltung');

'Zeugnisse' :

startMdi('Zeu.Verwaltung');

'Eti.DMS.Neu' : App_extras:EtiDMSNeu();

'Eti.DMS.Import' : App_extras:EtiDMSImport();

'Reorg.EK' : begin

if (Rechte[Rgt_Abl_Ein_Reorg]) then begin

```

Ein_Abl_Data:Reorganisation(n);

end;

end;

'Reorg.VK' : begin

    if (Rechte[Rgt_Abl_Auf_Reorg]) then begin

        Auf_Abl_Data:Reorganisation(n);

    end;

end;

'Reorg.Mat' : begin

    if (Rechte[Rgt_Abl_Mat_Reorg]) then begin

        Mat_Abl_Data:Reorganisation(n);

    end;

end;

'Reorg.OfP' : begin

    if (Rechte[Rgt_Abl_OfP_Reorg]) then begin

        Ofp_Abl_Data:Reorganisation(n);

    end;

end;

'Protokoll' :

    PtD_Main:ManageStatus();

'Settings':

    Set_Main:Verwaltung();

```


'Uebersetzungen' :

StartMdi('Translate.Verwaltung');

'Listenverwaltung':

StartMdi('Lfm.Verwaltung');

'Druckzonen':

StartMdi('Dzo.Verwaltung');

'Listendruck', 'Listen', 'Mnu.Listen' : begin

if (aMenuItem->wpCustom<>") then

Lfm_Ausgabe:Auswahl(aMenuItem->wpCustom)

else

Lfm_Ausgabe:Auswahl(w_Listen);

end;

'Mnu.Sonderfunktionen' : begin

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

//Scr_Main:Auswahlliste(aMenuItem->wpCustom);

Scr_Main:Auswahlliste(gPrefix);

if (Mode=c_ModeList) and (gZLList<>0) then

gZLList->WinUpdate(_WinUpdOn, _WinLstRecFromRecid | _WinLstRecDoSelect);

end;

'XML': XML_Main();

'Customfelder':

StartMdi('CUS.FP.Verwaltung');

'Customrechte':

StartMdi('Usr.CR.Verwaltung');

'Customauswahlfelder':

// StartMdi('CUS.AF.Verwaltung');

StartMdi('CUS.AF.Verwaltung2');

'Dialoge':

StartMdi('Dia.Verwaltung');

'Formulare':

StartMdi('Frm.Verwaltung');

'Scripte':

StartMdi('Scr.Verwaltung');

'Ankerfunktionen':

StartMdi('AFX.Verwaltung');

'Sonderfunktionen':

StartMdi('SFX.Verwaltung');

'Serviceinventar' : // ST 2010-09-10: hinzugefügt

StartMdi('SVI.Verwaltung');

'Aktivitaeten':

StartMdi('TeM.Verwaltung');

'Inventur':

if (Rechte[Rgt_Art_Inventur]) then begin

Art_Inv_Main:Start(true);

end;

'Dashboard':

StartMdi('Mdi.Dashboard');

'Termine' :

StartMdi('TeM.Verwaltung.Termine');

'DokumentAblage' : begin

RecBufClear(915);

gDokTyp # aMenuItem->wpCustom;

StartMdi('Dok.Verwaltung');

end;

'Passwortaendern': begin

Call('C16_UserPassw:Init');

end;

```
'Benutzerwechsel': begin  
  
  if (gUsergroup='USER') then begin  
  
    vA # NetInfo(_NtiAddressServer);  
  
    vA # vA + ' "' + DbName(_DbAreaAlias)+'";  
  
    sysexecute(Set.Client.Pfad+'\\c16_Winc.exe',vA,0);//_ExecMaximized);  
  
    gFrmMain->winclose();  
  
  end;  
  
  RETURN true;  
  
end;
```

```
'Mnu.Helpdesk' :  
  
  App_Extras:Helpdesk();
```

```
'Mnu.Update' :  
  
  App_Update_Data:CheckUpdate();
```

```
'Mnu.Changelog' :  
  
  StartMdi('Log.Verwaltung');
```

```
'Kommandozeile' : begin  
  
  App_Extras:Kommandozeile();  
  
end;
```

```
'Jobs' : begin  
  
  StartMdi('Job.Verwaltung');
```

end;

'JobError' : begin

 StartMdi('Job.Err.Verwaltung');

end;

'Filescanpfade': begin

 StartMdi('FSP.Verwaltung');

end;

'Abmessungstoleranzen' : begin

 StartMdi('MTo.Verwaltung');

end;

'Qualitaeten' : begin

 StartMdi('MQu.Verwaltung');

end;

'Qualitaetsstufen' : begin

 StartMdi('MQu.S.Verwaltung');

end;

'Betrieb': begin

```
StartMdi('Betrieb');
```

```
end;
```

```
'Zeitentypen' : begin
```

```
    StartMdi('ZTy.Verwaltung');
```

```
end;
```

```
/* Betriebsmenü */
```

```
'Mnu.BM.Wareneingang'    : App_Betrieb:Auswahl('Wareneingang');
```

```
'Mnu.BM.WE.Mat'         : App_Betrieb:Auswahl('WE.Mat');
```

```
'Mnu.BM.Etikettendruck'  : App_Betrieb:Auswahl('Etikettendruck');
```

```
'Mnu.BM.Lageruebersicht' : App_Betrieb:Auswahl('Lageruebersicht');
```

```
'Mnu.BM.Umlagern'       : App_Betrieb:Auswahl('Umlagern');
```

```
'Mnu.BM.Lieferschein'    : App_Betrieb:Auswahl('Lieferschein');
```

```
'Mnu.BM.LFS.Erfassung'   : App_Betrieb:Auswahl('LFS.Erfassung');
```

```
'Mnu.BM.LFS.ErfassungVLDAW' : App_Betrieb:Auswahl('LFS.ErfassungVLDAW');
```

```
'Mnu.BM.BagFm'          : App_Betrieb:Auswahl('BagFertigmeldung');
```

```
'Mnu.BM.BagTheoFm'      : App_Betrieb:Auswahl('BagTheoFertigmeldung');
```

```
'Mnu.BM.BagVWSperre'    : App_Betrieb:Auswahl('BagVWSperre');
```

```
'Mnu.BM.BagVWStorno'    : App_Betrieb:Auswahl('BagVWStorno');
```

```
'Mnu.BM.BagAbschluss'   : App_Betrieb:Auswahl('BagAbschluss');
```

```
'Mnu.BM.RsoPlan'        : App_Betrieb_RsoPlan:Start();
```

```
'Beenden'              : App_Betrieb:Auswahl('Beenden');
```

```
'Mnu.BM.BA_Einsatz_Raus' : App_Betrieb:Auswahl('BA_Einsatz_Raus');
```

```
'Mnu.BM.BA_Einsatz_Rein' : App_Betrieb:Auswahl('BA_Einsatz_Rein');
```

```
'Mnu.BM.Pak.Material' : App_Betrieb:Auswahl('Pak.Material');
```

```
'Mnu.BM.BagZeiten' : App_Betrieb:BagZVerwaltung();
```

```
'Mnu.BM.WaageNetto' : App_Betrieb:Auswahl('Waage.Netto');
```

```
'Mnu.Ktx.RecListSave' : begin
```

```
    Lib_GuiCom:RememberList(gZLList);
```

```
end;
```

```
// Datensatz Befehle
```

```
'Mnu.Refresh' : begin
```

```
    Refresh();
```

```
end;
```

```
'Mnu.Mark','Mnu.Ktx.Mark' : begin
```

```
    if ((Mode=c_ModeList) or (mode=c_modeView)) and
```

```
        (gZLList<>0) then begin
```

```
        if (gZLList->wpdbrecid<>0) then begin
```

```
            RecRead(gFile,0,0,gZLList->wpdbrecid);
```

```
            Mark();
```

```
        end;
```

```
    end;
```

```
end;
```

```

'Mnu.Ktx.Workbench' : begin

  if ((Mode=c_ModeList) or (mode=c_modeView)) and

    (gZLList<>0) then begin

    RecRead(gFile,0,0,gZLList->wpdbrecid);

    Lib_Workbench:CreateName(gFile, var vPref, var vA);

    if (vA="") then begin

      RAUS('EvtMenuCommand')

      RETURN false;

    end;

    Lib_Workbench:Insert(vA, gFile, gZLList->wpdbrecid);

  end;

end;

```

```

'Mnu.Ktx.Workbench.AllMark' : begin

  if ((Mode=c_ModeList) or (mode=c_modeView)) and

    (gFile<>0) and (gZLList<>0) then begin

    Lib_Workbench:InsertAllMarked(gFile);

  end;

end;

```

```

'Mnu.Ktx.Notifier' : begin

```

```

/*

```

```

  if ((Mode=c_ModeList) or (mode=c_modeView)) and

```



```

(gZLList<>0) then begin
  RecRead(gFile,0,0,gZLList->wpdbrecid);
  vA # Lib_Workbench:CreateName(gFile);
  if (vA="") then begin
    RAUS('EvtMenuCommand')
    RETURN false;
  end;
  Lib_Workbench:Insert(vA, gFile, gZLList->wpdbrecid);
end;
*/
end;

'Mnu.Ktx.Resize.Reset' : begin
  if (Mode=c_ModeNew) or (Mode=c_ModeNew2) or
    (Mode=c_ModeEdit) or (Mode=c_ModeEdit2) then RETURN true;
  w_ZoomX # 1.0;
  w_ZoomY # 1.0;
  if (w_Obj2Scale<>0) then begin
    Lib_GuiCom:ScaleObjects(w_Obj2Scale,n);
    try begin
      ErrTryIgnore(_rlocked,_rNoRec);
      ErrTryCatch(_ErrNoProcInfo,y);
      ErrTryCatch(_ErrNoSub,y);
      if (gPrefix<>") then Call(gPrefix+'_Main:ScaleObjects', true);
    end;
  end;
end;

```

```

if (gMDI<>0) then begin // Fenster refreshn: WORKAROUND

    gMDI->wpStyleframe # gMDI->wpStyleframe;

end;

end;

end;

'Mnu.Ktx.Resize' : begin

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

    (Mode=c_ModeEdit) or (Mode=c_ModeEdit2) then RETURN true;

if (w_Obj2Scale<>0) then begin

    Lib_GuiCom:ScaleObjects(w_Obj2Scale,y);

    try begin

        ErrTryIgnore(_rlocked,_rNoRec);

        ErrTryCatch(_ErrNoProcInfo,y);

        ErrTryCatch(_ErrNoSub,y);

        if (gPrefix<>'') then Call(gPrefix+'_Main:ScaleObjects', false);

    end;

if (gMDI<>0) then begin // Fenster refreshn: WORKAROUND

    gMDI->wpStyleframe # gMDI->wpStyleframe;

end;

end;

end;

// 'Mnu.Ktx.Columns.Reset' : begin

// sub ResetColumnWidth(aObj : int; opt aPrefix: alpha);

// end;

```

// Markierungen

'Mnu.Mark.Reset' : Lib_Mark:Reset(gFile);

'Mnu.Mark.Filter' : Lib_Mark:Filter();

'Mnu.Mark.Invert' : Lib_Mark:MarkInvert(gFile);

'Mnu.Mark.Save' : Lib_Mark:MarkSave();

'Mnu.Mark.Load' : Lib_Mark:MarkLoad();

'Mnu.Filter.Stop' : begin

 Sel_Main:Filter_Stop();

 RETURN true;

end;

'Mnu.Anhang' : begin

 if (gFile=916) then RETURN true;

 AttachmentsShow();

 RAUS('EvtMenuCommand')

 RETURN true;

end;

'Usersettings.Import' : begin

 vName # Lib_FileIO:FileIO(_WinComFileOpen,gMDI, ", 'TXT-Dateien|*.txt');

 if (vName='') then begin

 RAUS('EvtMenuCommand')

 RETURN false;

 end;

```
Usr_data:ImportINI(vName);
```

```
Msg(999998,"0,0,0,0);
```

```
end;
```

```
'Usersettings.Export' : begin
```

```
  vName # Lib_FileIO:FileIO(_WinComFileSave,gMDI, ", 'TXT-Dateien|'+gUsername+'.txt');
```

```
  if (vName="") then begin
```

```
    RAUS('EvtMenuCommand')
```

```
    RETURN false;
```

```
  end;
```

```
  if (StrCnv(StrCut(vName,strlen(vName)-3,4),_StrUpper) <>' .TXT') then vName # vName + ' .txt';
```

```
  Usr_data:ExportINI(vName);
```

```
  Msg(999998,"0,0,0,0);
```

```
end;
```

```
'Mnu.Daten.Export' : begin
```

```
  Lib_EDI:Export();
```

```
  RETURN true;
```

```
end;
```

```
'Mnu.Excel.Export' : begin
```

```
  vName # Lib_FileIO:FileIO(_WinComFileSave,gMDI, ", 'CSV-Dateien|*.csv');
```

```
  if (vName="") then begin
```

```

RAUS('EvtMenuCommand')

RETURN false;

end;

if (StrCnv(StrCut(vName,strlen(vName)-3,4),_StrUpper) <>'CSV') then vName # vName + '.csv';

Erx # Msg(998003,vName,_WinlcoQuestion,_WinDialogYesNoCancel,2);

if (Erx=_WinldCancel) then RETURN true;

if (Erx=_winldYes) then begin

    if (Lib_Excel:SchreibeDatei(gFile,vName, false)=true) then

        Msg(998004,vName+'|'+cnvai(Gv.int.01),_WinlcoInformation,0,0)

    else

        Msg(999999,gv.alpha.01,_WinlcoError,0,0);

    RAUS('EvtMenuCommand')

    RETURN true;

end;

// nur markierte?

Erx # Msg(998012,vName,_WinlcoQuestion,_WinDialogYesNoCancel,2);

if (Erx=_WinldCancel) then RETURN true;

if (Erx=_winldYes) then begin

    if (Lib_Excel:SchreibeDatei(gFile,vName, true)=true) then

        Msg(998004,vName+'|'+cnvai(Gv.int.01),_WinlcoInformation,0,0)

    else

        Msg(999999,gv.alpha.01,_WinlcoError,0,0);

    RAUS('EvtMenuCommand')

    RETURN true;

end;

end;

```

'Mnu.Excel.Import' : begin

vName # Lib_FileIO:FileIO(_WinComFileOpen, gMdi, ", 'CSV-Dateien|*.csv');

if (vName='') then begin

RAUS('EvtMenuCommand')

RETURN false;

end;

If (Msg(998005,vName,_WinIcoQuestion,_WinDialogYesNo,2)=_WinidYes) then begin

if (Lib_Excel:LiesDatei(gFile,vName,n)=true) then

Msg(998006,vName+'|'+cnvai(Gv.int.01),_WinIcoInformation,0,0);

else

Msg(999999,gv.alpha.01,_WinIcoError,0,0);

RAUS('EvtMenuCommand')

RETURN true;

end;

end;

'Mnu.Tobit.Export' : begin

If (Msg(998003,"Globaler Adressordner",_WinIcoQuestion,_WinDialogYesNo,2)=_WinidYes) then begin

if (Lib_AdrExportTobit:Export())=true) then

Msg(998004,"Globaler Adressordner"+'|'+cnvai(Gv.int.01),_WinIcoInformation,0,0)

else

Msg(999999,gv.alpha.01,_WinIcoError,0,0);

RAUS('EvtMenuCommand')

```
RETURN true;
```

```
end;
```

```
end;
```

```
'Mnu.RecNext':
```

```
  Action(c_ModeRecNext);
```

```
'Mnu.RecPrev':
```

```
  Action(c_ModeRecPrev);
```

```
'Mnu.RecFirst':
```

```
  Action(c_ModeRecFirst);
```

```
'Mnu.RecLast':
```

```
  Action(c_ModeRecLast);
```

```
'Mnu.Edit':
```

```
  Action(c_ModeEdit);
```

```
'Mnu.Cancel': begin
```

```
Action(c_ModeCancel);
```

```
end;
```

```
'Mnu.Save':
```

```
if (Mode=c_ModeList) and (gPrefix<>"") then begin
```

```
    Call(gPrefix+'_Main:RecSave')
```

```
end
```

```
else begin
```

```
    Action(c_ModeSave);
```

```
end;
```

```
'Mnu.New' :
```

```
    Action(c_ModeNew);
```

```
'Mnu.Search' :
```

```
    Action(c_ModeSearch);
```

```
'Mnu.Delete' :
```

```
    Action(c_ModeDelete);
```

```
// Favoriten
```



```

'Fav_Add' : begin

vHdl          # $TV.Hauptmenue->wpCurrentInt;

Usr.Fav.Username # gUsername;

Usr.Fav.Caption  # vHdl->wpCaption;

Usr.Fav.Name     # vHdl->wpName;

Usr.Fav.Custom   # vHdl->wpCustom;

Erx # RekInsert( 803, 0, 'MAN' );

if ( Erx != _rOk ) then

    Msg( 001000 + Erx, Translate( 'Favoriten'), 0, 0, 0 );


App_Main_Sub:RebuildFavoriten();

end;

```

```

'Fav_Add_SFX' : begin

vHdl # WinOpen( 'SFX.Auswahl', _winOpenDialog );


if ( vHdl > 0 ) then begin

    Lib_Sel:QRecList( vHdl->WinSearch( 'DL.SFX.Auswahl' ), 'SFX.Bereich = "Favoriten"' );

    vHdl->WinDialogRun();


if ( vHdl->WinDialogResult() = _winIdOk ) and ( gSelected != 0 ) then begin

    RecRead( 922, 0, _recId, gSelected );


if (SFX.EinzelrechtYN) then begin

    SFX.Usr.Nummer  # SFX.Nummer;

```

```
SFX.Usr.Username # gUsername;
```

```
Erx # RecRead(924,1,0);
```

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

```
    gSelected # 0;
```

```
    vHdl->WinClose();
```

```
    Msg(921002,"",0,0,0);
```

```
    RETURN true;
```

```
end;
```

```
end;
```

```
Usr.Fav.Username # gUsername;
```

```
Usr.Fav.Caption # SFX.Name;
```

```
Usr.Fav.Name    # 'sfx' + CnvAl( SFX.Nummer, _fmtNumNoGroup | _fmtInternal );
```

```
Usr.Fav.Custom  # SFX.Hauptmenuname;
```

```
Erx # RekInsert( 803, 0, 'MAN' );
```

```
if ( Erx != _rOk ) then
```

```
    Msg( 001000 + Erx, Translate( 'Favoriten'), 0, 0, 0 );
```

```
App_Main_Sub:RebuildFavoriten();
```

```
end;
```

```
gSelected # 0;
```

```
vHdl->WinClose();
```

```
end;
```

```
end;
```

```
'Fav_Del' : begin  
  
    vHdl          # $TV.Hauptmenue->wpCurrentInt;  
  
    Usr.Fav.Username # gUsername;  
  
    Usr.Fav.Caption  # vHdl->wpCaption;  
  
    Erx # RekDelete( 803, 0, 'MAN' );  
  
    if ( Erx != _rOk ) then  
  
        Msg( 001000 + Erx, ", 0, 0, 0 );  
  
        App_Main_Sub:RebuildFavoriten();  
  
    end;
```

```
// Default setzen  
  
'Default_set' : begin  
  
    Lib_GuiCom:node_set_default($TV.Hauptmenue);  
  
end;
```

```
// Default löschen  
  
'Default_del' : begin  
  
    Lib_GuiCom:node_del_default($TV.Hauptmenue);  
  
end;
```

```
'Coilculator' : begin
```

```

StartMDI('Coilculator');

end;

// Spaltenfixierung [06.10.2010/PW]

'Mnu.FixCols' : begin

    if ( gSelectedColumn = 0 ) then

        RETURN true;

    if ( gSelectedColumn->wpClmFixed = _winClmFixedNone ) then begin

        gSelectedColumn->wpClmFixed # _winClmFixedLeft;

        vFilter # 1;

    end

    else begin

        gSelectedColumn->wpClmFixed # _winClmFixedNone;

        vFilter # -1;

    end;

    if ( gZLList != 0 ) then begin

        if ( StrCut( gZLList->wpCustom, 1, 6 ) = '_FIXED' ) then

            vFilter # vFilter + CnvIA( StrCut( gZLList->wpCustom, 7, 100 ) );

            gZLList->wpCustom # '_FIXED' + CnvAI( vFilter );

        gZLList->WinUpdate( _winUpdOn, _winLstFromTop | _winLstPosSelected );

    end;

end;

'Mnu.ColFirst' : begin

```

```
if ( gSelectedColumn = 0 ) then
```

```
    RETURN true;
```

```
if ( gZLList != 0 ) then begin
```

```
    gSelectedColumn->wpCImOrder # 1;
```

```
    gZLList->WinUpdate( _winUpdOn, _winLstFromTop | _winLstPosSelected );
```

```
end;
```

```
end;
```

```
'Mnu.ColLast' : begin
```

```
    if ( gSelectedColumn = 0 ) then
```

```
        RETURN true;
```

```
if ( gZLList != 0 ) then begin
```

```
    gSelectedColumn->wpCImOrder # 100;
```

```
    gZLList->WinUpdate( _winUpdOn, _winLstFromTop | _winLstPosSelected );
```

```
end;
```

```
end;
```

```
otherwise begin
```

```
    try begin
```

```
        ErrTryIgnore(_rlocked,_rNoRec);
```

```
        ErrTryCatch(_ErrNoProcInfo,y);
```

```
        ErrTryCatch(_ErrNoSub,y);
```

```
        if (gPrefix<>"") then Call(gPrefix+'_Main:EvtMenuCommand',aEvt, aMenuItem);
```

```
    end
```

```

    end;

end;

gSelectedColumn # 0;

gSelectedRowID # 0;

RAUS('EvtMenuCommand')

RETURN true;

end;

//=====

// EvtMenuInitPopup

//    Initialisierung eines Menüfensters

//=====

sub EvtMenuInitPopup (

    aEvt    : event;

    aMenuItem : handle ) : logic

local begin

    vHdl : handle;

end;

begin

    vHdl # aEvt.obj->WinInfo( _winContextMenu );

    if ( vHdl = 0 ) then RETURN true;

```

```

BugFix(__PROCFUNC__, aEvt:Obj);

// if (gMdi<>0) then WinSearchPath(gMDI); // 21.11.2011

if (Wininfo(aEvt:Obj,_WinType)<>_WinTypeREcList) then RETURN true;

// Spaltenfixierung [30.09.2010/PW]

if ( gSelectedColumn <> 0 ) then begin

  if ( gSelectedColumn->wpCImFixed = _winCImFixedLeft ) then

    vHdl->WinMenuItemAdd( 'Mnu.FixCols', 'Fixierung aufheben', 1 );

  else

    vHdl->WinMenuItemAdd( 'Mnu.FixCols', 'Spalte fixieren', 1 );

  vHdl->WinMenuItemAdd( 'Mnu.ColFirst', 'Spalte an den Anfang schieben', 2 );

  vHdl->WinMenuItemAdd( 'Mnu.ColLast', 'Spalte an das Ende schieben', 3);

end;

RETURN true;

end;

//=====

// EvtTimer

//

//=====

sub EvtTimer (

  aEvt    : event;

  aTimerId : int; ) : logic

```

local begin

vl : int;

xvBuf800 : handle;

vErx : int;

vPrio : int;

v989 : handle;

vNeue : int;

vDel : int;

vWB : int;

vZoneList : int;

vltem : int;

vFocus : int;

vHdl : int;

end;

begin

if (gFrmMain = 0 or gMDINotifier = 0) then

RETURN false;

if (Lib_Tapi_Snom:isDev()) then

Lib_Tapi_Snom:ClientPollTapiActions();

/* Hinweiston bei neuen Nachrichten */

GV.Sys.Username # gUsername;

try begin


```

ErrTryCatch(_ErrNoFile, true);

vl # RecLinkInfo( 989, 999, 6, _recCount );

end;

if (Errget() <> _ErrOK) then begin

    ErrSet(_ErrOK);

    RETURN false;

end;


// ALTER TIMER

if (gMdiNotifier <> 0) then

if (StrFind(gMdiNotifier->wpname, 'Cockpit', 0) = 0) then begin


    if ( gNotifierCounter != -1 ) then begin


        v989 # recBufCreate(989);

        vErx # RecLink(v989, 999, 8, _recFirst);

        WHILE (vErx <= _rLocked) and (v989->"TeM.E.LöschenYN") do begin

            dec(vl);

            vErx # RekDelete(v989, 0, 'AUTO');

            vErx # RecLink(v989, 999, 8, _recFirst);

        END;


        if ( vl > gNotifierCounter ) then begin


            vErx # RecLink(v989, 999, 7, _recFirst);

```

```
WHILE (vErx<=_rLocked) and (v989->TeM.E.NotifiedYN=false) do begin
```

```
  if (v989->"TeM.E.Priorität">vPrio) then vPrio # v989->"TeM.E.Priorität";
```

```
  vErx # RecRead(v989,1,_recLock);
```

```
  v989->TeM.E.NotifiedYN # y;
```

```
  vErx # RecReplacE(v989,_recUnlock);
```

```
  vErx # RecLink(v989, 999, 7, _recFirst);
```

```
END;
```

```
if (vPrio>0) then begin
```

```
  if (gUsername<>'TJ') then Lib_Sound:Play( 'Hupe LKW.wav' )
```

```
end
```

```
else begin
```

```
  Lib_Sound:Play( 'notice.wav' );
```

```
end;
```

```
//  $rl.Messages->WinUpdate( _winUpdOn, _winLstFromSelected | _winLstRecDoSelect| _winLstRecE
```

```
  $rl.Messages->WinUpdate( _winUpdOn, _WinLstFromLast | _winLstRecDoSelect| _winLstRecEvtSkip
```

```
end
```

```
else if ( vl < gNotifierCounter ) then begin
```

```
//  $rl.Messages->WinUpdate( _winUpdOn, _winLstFromSelected | _winLstRecDoSelect| _winLstRecE
```

```
//  $rl.Messages->WinUpdate( _winUpdOn, _winLstFromLast | _winLstRecDoSelect| _winLstRecEvtSkip
```

```
  $rl.Messages->WinUpdate( _winUpdOn, _WinLstFromTop | _winLstPosSelected | _winLstRecEvtSkip
```

```
end;
```

```
RecBufDestroy(v989);
```

```
end;
```

```
gNotifierCounter # vl;
```

```
RETURN true;
```

end;

// REST PASSIERT IM NEUEN COCKPIT !!!

RETURN true;

/**

// COCKPIT -----

vWB # VarInfo(WindowBonus);

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

vZoneList # gZoneList;

VarInstance(WindowBonus,vWB);

// NEUE?

v989 # recBufCreate(989);

FOR vErx # RecLink(v989, 999, 7, _recFirst)

LOOP vErx # RecLink(v989, 999, 7, _recFirst)

WHILE (vErx<=_rLocked) and (v989->TeM.E.NotifiedYN=false) do begin

if (v989->"TeM.E.Priorität">vPrio) then vPrio # v989->"TeM.E.Priorität";

vErx # RecRead(v989,1,_recLock);

v989->TeM.E.NotifiedYN # y;

vErx # Recreplace(v989,_recUnlock);

inc(vNeue);

Mdi_Cockpit:AddEventToZoneList(vZonelist, v989, y);

END;

// zu Löschen?

FOR vErx # RecLink(v989, 999, 8, _recFirst)

LOOP vErx # RecLink(v989, 999, 8, _recFirst)

WHILE (vErx<=_rLocked) and (v989->"TeM.E.LöschenYN") do begin

inc(vDel);

vItem # Mdi_cockpit:FindEventInZoneList(vZonelist, v989);

if (vItem>0) then

Mdi_Cockpit:DeleteItem(vZoneList, vItem, 0);

END;

RecBufDestroy(v989);

if (gNotifierCounter != -1) then begin

if (vNeue>0) then begin

if (vPrio=999) then begin

end

else if (vPrio>0) then

Lib_Sound:Play('Hupe LKW.wav')

else if (vPrio=0) then

Lib_Sound:Play('notice.wav');

end;

if (vDel>0) then

Lib_Sound:Play('remove.wav')

end;

if (vNeue>0) or (vDel>0) then begin

 Mdi_Cockpit:ReBuildZones(gMDINotifier, vZonelist);

end;

gNotifierCounter # vl;

RETURN true;

***/

end;

//=====

// EvtTapi

// Tapi meldet sich

//=====

sub EvtTapi (

 aEvt : event; // Ereignis

 aTapiDevice : int; // Tapi Device

 aCallID : int; // Call-ID

 aCallState : int; // Anrufstatus

 aCallTime : caltime; // Datum und Uhrzeit

 aCallerID : alpha; // Rufnummer

```

aCalledID : alpha // Rufnummer des Anschlusses auf dem angerufen wird

) : logic

begin

if (Lib_SFX:Check_AFX('App.EvtTapi')) then begin

    Call(AFX.Prozedur, aEvt, aTapiDevice, aCallID, aCallState, aCallTime, aCallerID, aCalledID);

    RETURN true;

end;

//debug('outer:'+aCallerID+' ' +aint(aCallState));

Lib_Tapi:TapiIncoming(aCallState, aCallID, aCallerID, aCalledID, aCallTime);

RETURN true;

end;

```

```

//=====

```

```

// Suche

```

```

//     RESULT: Refreshen?

```

```

//=====

```

```

sub Suche(aSuch : alpha) : logic

```

```

local begin

```

```

    vA : alpha;

```

```

    vB : alpha;

```

```

    vI1,vI2,vI3 : int;

```

```

    vAnz : int;

```

```

    vX : int;

```

```

    vY : int;

```

```

vTds : int;

vFld : int;

vTyp : int;

end;

begin

  aSuch # $ed.Suche->wpCaption;


  vAnz # keyinfo(gFile, gKey, _KeyFldCount);

  if (vAnz>1) then begin // müssen TOKENIZEN...

    if (StrFind(aSuch,'|',1)>0) then begin

      end

    else if (StrFind(aSuch,'*',1)>0) then begin // ggf. "*" ?

      aSuch # Str_ReplaceAll(aSuch,'*','|');

      end

    else if (StrFind(aSuch,' ',1)>0) then begin // ggf. " " ?

      aSuch # Str_ReplaceAll(aSuch,' ','|');

      end

    else if (StrFind(aSuch,'/',1)>0) then begin // ggf. "/" ?

      aSuch # Str_ReplaceAll(aSuch,'/','|');

      end;

    end;

  end;


  if (gSuchProc<>") then begin

    vI1 # Call(gSuchProc, gFile, gKeyID, aSuch);

    if (vI1<0) then RETURN false;

    if (vI1>0) then RETURN true;

```

end;

vI1 # 0;

RecBufClear(gFile);

FOR vX # 1; loop inc(vX) while (vX<=vAnz) do begin

 vTyp # KeyFldInfo(gFile, gKey, vX, _KeyFldType);

 vTds # KeyFldInfo(gFile, gKey, vX, _KeyFldSbrNumber);

 vFld # KeyFldInfo(gFile, gKey, vX, _KeyFldNumber);

 vY # StrFind(aSuch, '|', 0);

 if (vY<>0) then begin

 vA # StrCut(aSuch, 1, vY-1);

 aSuch # StrCut(aSuch, vY+1, 99);

 end

 else begin

 vA # aSuch;

 aSuch # ";

 end;

case vTyp of

 _TypeAlpha : begin

 vA # StrCut(vA, 1, FldInfo(gFile, vTds, vFld, _FldLen));

 FldDef(gFile, vTds, vFld, StrCnv(vA, _StrUpper));

 end;

 _TypeDate : begin


```

vB # Str_Token(vA,',',1);

vI1 # cnvia(vB);

vB # Str_Token(vA,',',2);

vI2 # cnvia(vB);

vB # Str_Token(vA,',',3);

vI3 # cnvia(vB);

if (vI1<1) or (vI1>31) then vI1 # 0;

if (vI2<1) or (vI2>12) then vI2 # 0;

if (vI3=0) then vI3 # DateYear(today);

if (vI3<100) then vI3 # vI3 + 100;

if (vI3>1900) then vI3 # vI3 - 1900;

if (vI3<1) or (vI3>150) then vI3 # 0;

if (vI1<>0) and (vI2<>0) and (vI3<>0) then begin

    vB # Cnvai(vI1)+','+Cnvai(vI2)+','+Cnvai(vI3);

    Try begin

        ErrTryCatch(_Errcnv,y);

        FldDef(gFile, vTds, vFld, CnvDA(vB));

    end;

end;

end;

_TypeFloat : begin

    try begin

        ErrTryCatch(_Errcnv,y);

        FldDef(gFile, vTds, vFld, CnvFA(vA));

    end;

```

end;

_Typeint : begin

try begin

ErrTryCatch(_Errcnv,y);

FldDef(gFile, vTds, vFld, CnvIA(vA));

end;

end;

_TypeWord : begin

try begin

ErrTryCatch(_Errcnv,y);

FldDef(gFile, vTds, vFld, CnvIA(vA));

end;

end;

_TypeTime : begin

try begin

ErrTryCatch(_Errcnv,y);

FldDef(gFile, vTds, vFld, CnvTA(vA));

end;

end;

end;

END;

// Satz holen

RecRead(gFile, gKey, 0);

```

RETURN true;

end;

//=====

// SucheEvtFocusTerm

//=====

sub SucheEvtFocusTerm (
    aEvt          : event;    // Ereignis
    aFocusObj      : int      // nächstes Objekt
) : logic
begin

// if (cSchnellsuche) then RETURN true;

if ("Set.!SchnellSuche"=false) then RETURN true;

if (Mode<>c_ModeList) and (Mode<>c_ModeEdList) then RETURN true;

if ($ed.Suche->wpCaption=") then RETURN true;

if (Suche($ed.Suche->wpCaption)) then begin
    if (gZLList<>0) then
        gZLList->WinUpdate(_WinUpdOn, _WinLstRecFromBuffer | _WinLstRecDoSelect);
end;

RETURN true;

end;

```

```

//=====

// FocusInit

//      Fokus auf Objekt neu gesetzt

//=====

sub EvtFocusInit (

    aEvt          : event;    // Ereignis

    aFocusObject   : int      // vorheriges Objekt

) : logic

local begin

    Erx    : int;

    vX     : int;

    vFrame : int;

    vHdl   : int;

    vTmp   : int;

end;

begin

    REIN('EvtFocusInit '+aEvt:Obj->wpname)

    BugFix(__PROCFUNC__, aEvt:Obj);

    if (w_Mdi<>0) and (gMDI<>w_Mdi) then  gMDI # w_MDI; // MDIBUGFIX 03.06.2014; 31.03.2022 AH : Te

    if (w_Command='BUF2') then begin

        w_Command # "";

        vTmp # cnvia(w_cmd_para);

        RekRestore(vTmp);

```

```

end;

if (w_Command='xxBUF') then w_Command # 'BUF2';

//debug('FocusInit :'+aEvt:obj->wpname+' Var:'+w_name+' Mdi:'+gMDI->wpname+' '+mode);

// falsche Reihenfolge der Events: Term->Init->MdiActivate

// also hier prüfen, ob gMDI noch stimmt:

// vHDL # Wininfo(aEVT:Obj, _Winframe);

// if (vHDL<>gMDI) then begin

//   debug('init gMDI<>vHDL auf '+vHDL->wpname);

//   gMDI # vHDL;

// end;

// if (w_Name<>gMDI->wpname) then begin

//   if (gMDI->wpcustom<>") then

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

//   if (w_Name<>gMDI->wpname) then

//     debug('AAAAAAARRRRRRRRRRRGGGGGGGGGGGGGGGGG');

// end;

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

// BUGFIX if (gMdi<>0) then WinSearchPath(gMDI); // 21.11.2011

// Hauptmenü setzen 28.08.2012 AI:

_SetMenu(gMenuName);

```

```

// Unsinniges Aufruf VOR MDIActivate??

// if (w_Name<>gMDI->wpname) then begin

//   RAUS('EvtFocusInit (vor MdiActivate)');

//   RETURN true;

// end

// else begin

//debug('Fl:'+aEvt:Obj->wpname);

//if (aFocusobject<>0) then debug('Fl fo:'+aFocusobject->wpname);


if (aEvt:Obj->wpname='DUMMYNEW') then begin

  Erx # cnvia(aEvt:Obj->wpcustom);

  if (Erx<>0) and (Erx<>aEvt:Obj) then begin

    Erx->winfocusset(false);

    RAUS('EvtFocusInit')

    RETURN false;

  end

  else if (w_lastFocus<>0) then begin

    w_lastFocus->winfocusset(false);

//    w_LastFocus # 0; --- AI 20.08.2009

    RAUS('EvtFocusInit')

    RETURN true;

  end

  else if (aFocusObject=0) then begin

    if (Mode=c_ModeView) or (Mode=c_ModeBald+c_ModeView) then begin // Wieder auf "Edit" focusieren

```

```

vTmp # 0;

vTmp # gMdi->winsearch('Edit');

if (vTmp<>0) and (vTmp->wpdisabled) then

    if (gMdi->winsearch('EditErsatz')<>0) then

        vTmp # gMdi->winsearch('EditErsatz');

        if (vTmp<>0) then vTmp->winfocusset(false);

    end

else begin

    if (aFocusObject<>0) then

        aFocusObject->winfocusset(false);

        RAUS('EvtFocusInit')

        RETURN false;

    end;

    RAUS('EvtFocusInit')

    RETURN true;

end;

end;

// end;

```

```

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

    aFocusObject->winfocusset();

    RAUS('EvtFocusInit')

    RETURN false;

end;

```

// letztes gültiges Objekt merken... 20.08.2009 AI

```
if ((Mode=c_ModeNew) or (Mode=c_ModeNew2) or (Mode=c_ModeEdit) or (Mode=c_ModeEdit2)) and  
  (aEvt:Obj->wpname<>'NB.Main') and (aEvt:Obj->wpname<>'Edit') and (aEvt:Obj->wpname<>'DUMMYN'  
  w_Lastfocus # aevt:obj;
```

```
// 2022-08-08 AH : Quickbuttons sind FOKUSIERBAR
```

```
if (aEvt:Obj<>0) and (aEvt:Obj->wpname=*"bt.Quick*") then RETURN true;
```

```
if (Mode=c_ModeView) and  
  ((aEvt:Obj->wpname='Edit') or (aEvt:Obj->wpname='EditErsatz')) then begin  
  RAUS('EvtFocusInit')  
  RETURN true;  
end;
```

```
if (aEvt:Obj->wpname='RecPrev') then begin  
  Action(c_ModeRecPrev);  
end;
```

```
if (aEvt:Obj->wpname='RecNext') then begin  
  Action(c_ModeRecNext);  
end;
```

```
if (Mode=c_ModeView) then begin // Wieder auf "Edit" focusieren  
  vHDL # Winsearch(gMDI,'Edit');  
  if (vHDL=0) then vHDL # Winsearch(gMDI,'Edit2');  
  if (vHDL<>0) then begin
```



```
// 21.07.2016 zur Not auf EditErsatz springen
```

```
if (vHdl->wpdisabled) then vHdl # Winsearch(gMDI,'EditErsatz');
```

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

```
    vHDL->wpDisabled # false;
```

```
    vHDL->WinFocusSet(false);
```

```
end;
```

```
end;
```

```
RAUS('EvtFocusInit')
```

```
RETURN true;
```

```
end;
```

```
if (Mode=c_ModeList) and (gZLList<>0) then // Wieder auf ZL focusieren
```

```
if (gZList->wpdisabled=false) then begin
```

```
    gZLList->wpDisabled # false;
```

```
    gZLList->WinFocusSet(false);
```

```
    RAUS('EvtFocusInit')
```

```
    RETURN true;
```

```
end;
```

```
// Ermitteln des Frames
```

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

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

```
    RAUS('EvtFocusInit')
```

```
    RETURN TRUE;
```

```
end;
```

```

vX # aEvt:Obj->Wininfo(_wintype);

if (vX=_WinTypeEdit) or
  (vX=_WinTypeFloatEdit) or
  (vX=_WinTypeIntEdit) or
  (vX=_WinTypeTimeEdit) or
  (vX=_WinTypeCheckbox) or
  (vX=_WinTypeTextEdit) or
  (vX=_WinTypeDateEdit) then begin

  if (vX=_WinTypeCheckbox) then
    aEvt:Obj->wpColBkg # Set.Col.Field.Cursor //_WinColCyan;
  else
    aEvt:Obj->wpColFocusBkg # Set.Col.Field.Cursor;
//   aEvt:Obj->wpColFocusBkg # ColFocus;

  try begin
    ErrTryIgnore(_rlocked,_rNoRec);
    ErrTryCatch(_ErrNoProcInfo,y);
    ErrTryCatch(_ErrNoSub,y);
    if (gPrefix<>") then Call(gPrefix+'_Main:EvtFocusInit',aEvt,aFocusObject);
  end;

end;

/* 31.10

if (aEvt:Obj->Wininfo(_WinType)=_WinTypecheckbox) then begin
  aEvt:Obj->wpColBkg # _WinColCyan;

```

```

end;

*/

RAUS('EvtFocusInit')

RETURN true;

end;

//=====

// EvtFocusTerm

//      Fokus wechselt hier weg

//=====

sub EvtFocusTerm (

    aEvt          : event;    // Ereignis

    aFocusObj      : int      // nächstes Objekt

) : logic

local begin

    vOk  : logic;

    vX   : int;

    vHDL : int;

end;

begin

    REIN('EvtFocusTerm:'+aEvt:obj->wpname)

    // falsche Reihenfolge der Events: Term->Init->MdiActivate

    // also hier prüfen, ob gMDI noch stimmt:

    // vHDL # Wininfo(aEVT:Obj, _Winframe);

```

```

// if (vHDL<>gMDI) then begin

//  debug('term gMDI<>vHDL');

//  gMDI # vHDL;

//  end;


//debug('Term :'+aEvt:obj->wpname+'  Var:'+w_name);

// if (w_Name<>gMDI->wpname) then begin

//  debug('korregiere....');

//  if (gMDI->wpcustom<>") then

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

//  if (w_Name<>gMDI->wpname) then

//    debug('AAAAAAARRRRRRRRRRRGGGGGGGGGGGGGGGGGG');

//  end;

```

```

BugFix(__PROCFUNC__, aEvt:Obj);

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

// if (gMdi<>0) then WinSearchPath(gMDI); // 21.11.2011

```

```

// nur Info-Checkbox? ="_I"

```

```

if (aFocusObj<>0) then begin

  vX # aFocusObj->Wininfo(_wintype);

  if (vX=_WinTypeCheckbox) and (aFocusObj->wpcustom='_I') then begin

    RAUS('EvtFocusTerm')

    RETURN false;

  end;

end;

```

vOk # true;

try begin

ErrTryIgnore(_rlocked,_rNoRec);

ErrTryCatch(_ErrNoProcInfo,y);

ErrTryCatch(_ErrNoSub,y);

if (gPrefix<>") then vOk # Call(gPrefix+'_Main:EvtFocusTerm',aEvt,aFocusObj);

end;

if (vOk) then begin

vX # aEvt:Obj->Wininfo(_wintype);

if (vX=_WinTypeEdit) or

(vX=_WinTypeFloatEdit) or

(vX=_WinTypeIntEdit) or

(vX=_WinTypeTimeEdit) or

(vX=_WinTypeTextEdit) or

(vX=_WinTypeDateEdit) then begin

aEvt:Obj->wpColFocusBkg # _WinColParent;

end;

if (vX=_WinTypecheckbox) then begin

aEvt:Obj->wpColBkg # _WinColParent;

end;

end;

if (aFocusObj<>0) then begin

vX # aFocusObj->Wininfo(_wintype);

```

if (Mode=c_ModeView) then begin
    if (vX=_WinTypeEdit) or
        (vX=_WinTypeFloatEdit) or
        (vX=_WinTypeIntEdit) or
        (vX=_WinTypeTimeEdit) or
        (vX=_WinTypeTextEdit) or
        (vX=_WinTypeDateEdit) then begin
        RAUS('EvtFocusTerm')

        RETURN false;
    end;
end;

if (vX=_WinTypeCheckbox) and (aFocusObj->wpcustom='_I') then RETURN false;

end;

RAUS('EvtFocusTerm')

RETURN vOk;

end;

//=====

// EvtChanged

//      Feldveränderungen

//=====

sub EvtChanged(
    aEvt          : event;    // Ereignis

): logic

```

local begin

vA : alpha;

vB : alpha;

vl1,vl2,vl3 : int;

vSuch : alpha;

vAnz : int;

vX : int;

vY : int;

vTds : int;

vFld : int;

vTyp : int;

vTmp : int;

vR : range;

end;

begin

// 25.06.2014

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

BugFix(__PROCFUNC__, aEvt:Obj);

// if (gMdi<>0) then WinSearchPath(gMDI); // 21.11.2011

// Schnellsuche?

if (aEvt:Obj->wpName='ed.Suche') and (gfile<>0) and (Mode=c_ModeList) then begin

if (aEvt:Obj<>Winsearch(gMdi,'ed.Suche')) then RETURN true;

```
$ed.Suche->wplengthmax # 40;
```

```
// if (cSchnellsuche) then begin
```

```
if ("Set.!SchnellSuche"=false) then begin
```

```
WinEvtProcessSet(_WinEvtSystem,FALSE);
```

```
// pro Taste suchen...
```

```
vSuch # $ed.Suche->wpCaption;
```

```
if (Suche(vSuch)) then begin
```

```
    // Liste aktualisieren
```

```
    if (gZLList<>0) then
```

```
        gZLList->WinUpdate(_WinUpdOn, _WinLstRecFromBuffer | _WinLstRecDoSelect);
```

```
end;
```

```
// Range anpassen wegen ggf SwitchMask + Focus auf ed.Suche = alles markiert!
```

```
// 1.3.2011 AI raus:
```

```
// vSuch # $ed.Suche->wpCaption;
```

```
// vTmp # StrLen(vSuch);
```

```
// vR # $ed.Suche->wprange;
```

```
//debug($ed.Suche->wpcaption+' '+aint(vR:Min)+' ':''+aint(vR:Max));
```

```
// $ed.Suche->wprange # RangeMake(vTmp,vTmp);
```

```
// Workaround?
```

```
SySsleep(1);
```



```

WinEvtProcessSet(_WinEvtSystem,true);

end;

end;

end;

//=====

// EvtInit

//      Initialisieren der Applikation

//=====

sub EvtInit(

    aEvt          : event;    // Ereignis

): logic

local begin

    Erx          : int;

    vA           : alpha(1000);

    vVorgabeKey  : int;

    vFilter      : int;

    vX           : int;

    vDLSort      : int;

    vTmp         : int;

    vFont        : font;

end;

begin

```

```
REIN('EvtInit')
```

```
BugFix(__PROCFUNC__, aEvt:Obj);
```

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

```
// Prjekt 1337/133
```

```
// Listen dürfen NICHT refreshen beim Verlassen des Focus beim Klick in andere App
```

```
// hängt an diesem Flag "WinLstRecFocusTermReset" - aber das brauchen wir dringend
```

```
// bei verknüpften RecList !!!
```

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

```
    if (gZLList->wpDbLinkFileNo=0) then
```

```
        gZLList->wpLstFlags # gZLList->wpLstFlags & ~_WinLstRecFocusTermReset;
```

```
end;
```

```
// ERWEITERUNGEN FÜR ALLE FENSTER: 05.10.2018 AH: wenn nicht gefüllt!!
```

```
if (aEvt:obj->WinEvtProcNameGet(_WinEvtCreated)=) then
```

```
    aEvt:obj->WinEvtProcNameSet(_WinEvtCreated, 'App_Main:EvtCreated');
```

```
// für RESIZE/SCALE
```

```
/***/
```

```
// NICHT für Selektions-Dialoge
```

```
// if (StrCut(aEvt:Obj->wpname, 1, 4)<>'Sel.') then begin
```

```
if (StrFind(aEvt:Obj->wpname,'Sel.',0)=0) then begin
```

```
    aEvt:obj->WinEvtProcNameSet(_WinEvtMOuse, 'App_Main:EvtMouse');
```

```
    if (aEvt:Obj->wpMenuNameCntxt=) then begin
```

```

aEvt:Obj->wpMenuNameCntxt # 'Std.Kontext.Window';

if (aEvt:Obj->WinEvtProcNameGet(_WinEvtMenuCommand)=) then begin

    aEvt:Obj->WinEvtProcNameSet(_WinEvtMenuCommand, 'App_Main:EvtMenuCommand');

    aEvt:Obj->WinEvtProcNameSet(_WinEvtMenuInitPOpup, 'App_Main:EvtMenuInitPopup');

end;

if (gZLList<>0) then begin

    gZLList->wpMenuNameCntxt # 'Std.Kontext.RecList';

    gZLList->WinEvtProcNameSet(_WinEvtMenuInitPOpup, 'App_Main:EvtMenuInitPopup');

    gZLList->WinEvtProcNameSet(_WinEvtMenuCommand, 'App_Main:EvtMenuCommand');

end;

end;

end

/***/

// Quickbar

if (Winsearch(aEvt:obj, 'gt.Quickbar')<>0) then Lib_GuiCom:RecallQuickbar(aEvt:Obj);


// globale Fenstervariablen merken

aEvt:Obj->wpcustom # cnvai(VarInfo(WindowBonus));

//debugx('SET custom '+aEvt:Obj->wpname+' auf '+aEvt:Obj->wpcustom);

if (w_Name<>") then begin//todo('PANIK!!!! :'+w_name);

    vTmp # Lib_GuiCom2:GetDBVarHandle(aEvt:Obj);

    if (vTmp<>0) then VarInstance(windowbonus, vTmp);

    if (w_Name<>") then

        Msg(1998,"",0,0,0);

```

end;

w_Name # aEvt:obj->wpname;

w_MDI # aEvt:Obj;

Lib_Pflichtfelder:PflichtfelderListeFuellen(w_Name, var w_Pflichtfeld, var w_Obj4auswahl);

case(aEvt:Obj->wpName) of

'AppFrame' : begin

/**

vFont # _App->wpDialogBoxFont;

vFont:Name # 'Terminal';

vFont:Size # 200;

_App->wpDialogBoxFont # vFont;

debugx("");

***/

//_App->wpStyleTheme # _WinStyleThemeSystem; //2023-04-06 MR auskommentiert unter Rücksprache

//if (gusername='AH') then

/**

set.theme # 'NEW 16'

if (StrFind(StrCnv(Set.Theme,_strupper),'NEW',1)>0) then begin

_App->wpTileTheme # _WinTileThemeEnhanced;

gTmp # cnvia(Set.Theme);

if (gTmp=0) then gTmp # 16;

_App->wpTileSize # gTmp;

```

end;

***/

_App->wpTileTheme # _WinTileThemeEnhanced;

_App->wpTileSize # 16;


// _App->wpStyleTheme # _WinStyleThemeNone;

//_App->wpStyleTheme # _WinStyleThemeSystem;

// Deskriptor des Hauptfensters laden

gFrmMain # $AppFrame;

gFrmMain->wpAreaLeft # 0;

gFrmMain->wpAreaTop # 0;

gFrmMain->wpAreaRight # WinInfo( 0, _WinScreenWidth);

gFrmMain->wpAreaBottom # WinInfo( 0, _WinScreenHeight);


// Testsystem

// ST 2013-05-29: Auch erkennen, wenn mehrere Testsysteme im Einsatz sind

// if ( StrCnv( DbName( _dbaAreaAlias ), _strUpper ) = 'TESTSYSTEM' ) then begin

// if ( StrFind(StrCnv( DbName( _dbaAreaAlias ), _strUpper ), 'TESTSYSTEM', 1) > 0) then begin

if (isTestsystem) then begin

gFrmMain->wpPictureMode # _winPictTile;

gFrmMain->wpPictureName # 'TESTSYSTEM';

```

```

Set.Bild.Hintergrund # ";

end;

// Programmtitel [15.03.2010/PW]

begin

    vA # ";

    // Datenbank: readonly

    if ( DbalInfo( _dbaReadOnly ) > 0 ) then begin

        vA # '_readonly_';

        gFrmMain->wpColBkgApp # _winColRed;

    end;

    // Debugmodus

//@ifdef Prog_ReadOnly

//    if ( cReadOnlyCondition ) then begin

//        Set.Appli.Caption # '!!! BLAUER MODUS !!! %DB%'

//        //vA # '!!! TRANSACTION OPEN !!!';

//        if ( DbalInfo( _dbaReadOnly ) <= 0 ) then

//            gFrmMain->wpColBkgApp # _winColLightBlue

//        else

//            gFrmMain->wpColBkgApp # _winColYellow;

//        end;

//@endif

    // Standardbezeichnung

    if ( Set.Appli.Caption = " ) then

```

Set.Appli.Caption # 'Business Control%_%%_%%_%%Datenbank: %DB%%_%%_%%angemeldet als:

// Bezeichnung generieren

Set.Appli.Caption # Set.Appli.Caption + vA

vA # StrFmt(vA, 10, _strEnd);

vA # Str_ReplaceAll(Set.Appli.Caption, '%_%', vA);

vA # Str_ReplaceAll(vA, '%DB%', DbName(_dbaAreaAlias));

vA # Str_ReplaceAll(vA, '%USER%', gUserName);

vA # Str_ReplaceAll(vA, '%DATE%', CnvAD(today));

if (StrFind(StrCut(vA, 1, 159), 'Business Control', 1, _strCaseIgnore) > 0) then

gFrmMain->wpCaption # vA;

else

gFrmMain->wpCaption # StrCut(vA, 1, 143) + 'Business Control';

/* ALT

vA # vA + 'Business Control'+

StrFmt(vA+vA+vA,30,_StrEnd)+

Translate('Datenbank')+': '+DbName(_DbaAreaAlias) +

StrFmt(vA+vA+vA+vA,30,_StrEnd)+

Translate('angemeldet als')+': '+gUsername+

StrFmt(vA+vA+vA+vA,20,_StrEnd)+

Translate('am')+': '+Cnvad(today);

@ifdef Prog_ReadOnly

if (DbalInfo(_DBaReadOnly)<=0) then begin

```

if (gUsername=cReadOnlyCondition) then begin
    vA # '!!! TRANSACTION OPEN !!!';

    gFrmMain->wpColBkgApp # _wincollightblue;

    end;

end;

@Endif

gFrmMain->wpcaption # vA;

*/

end; // Programmtitel


// Usersettings holen

Lib_GuiCom:RecallWindow(gFrmMain);


// Oberfläche aufbauen

StartMdi( 'Hauptmenue' );

StartMdi( 'Workbench' );

if ( Set.TimerYN ) then

    StartMdi( 'Notifier' );


gToolBar # gFrmMain->WinSearch( 'TB.User' );

// Toolbarfont anpassen...

if (Usr.Font.Size<>0) then begin

    vFont # gToolBar->wpfont;

    vFont:Size # (Usr.Font.Size-4) * 10;

    gToolBar->wpfont # vFont;

end;

```



```

if ( Set.Bild.Hintergrund != " ) then begin

  if (Entwicklerversion()) and (Strcut(Set.Bild.Hintergrund,2,1)<>':') then

    gFrmMain->wpPictureName # '..\' + Set.Bild.Hintergrund;

  else

    gFrmMain->wpPictureName # '*' + Set.Bild.Hintergrund;

end;

Lib_GuiCom:TranslateObject( gToolBar );

Lib_GuiCom:TranslateObject( gFrmMain );


// Toolbar Buttons den Rechten entsprechend deaktivieren (START)

if ((Rechte[Rgt_Auftrag]) = false) then begin

  Erx # gToolBar->WinSearch('Auf_P');

  Erx->wpVisible # false;

End;


if ((Rechte[Rgt_Einkauf]) = false) then begin

  Erx # gToolBar->WinSearch('EK_P');

  Erx->wpVisible # false;

End;


if ((Rechte[Rgt_Adressen]) = false) then begin

  Erx # gToolBar->WinSearch('Adressen');

  Erx->wpVisible # false;

End;

```

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

```
    Erx # gToolbar->WinSearch('Artikel');
```

```
    Erx->wpVisible # false;
```

```
End;
```

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

```
    Erx # gToolbar->WinSearch('Material');
```

```
    Erx->wpVisible # false;
```

```
End;
```

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

```
    Erx # gToolbar->WinSearch('BAG');
```

```
    Erx->wpVisible # false;
```

```
End;
```

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

```
    Erx # gToolbar->WinSearch('Projekte');
```

```
    Erx->wpVisible # false;
```

```
End;
```

```
if// ((gUserName='AH') or (gUserName='ST') or (gUserName='TM') or () and
```

```
    (gUserGroup='PROGRAMMIERER') then begin
```

```
    Erx # gToolbar->WinSearch('DEBUG');
```

```
    Erx->wpVisible # true;
```

```
    Erx # gToolbar->WinSearch('DEBUG2');
```

```
    Erx->wpVisible # true;
```

end;

// Toolbar Buttons den Rechten entsprechend deaktivieren (ENDE)

// TAPI initialisieren

Lib_Tapi:TapiInitialize();

Lib_Jobber:Init();

RAUS('EvtInit')

RETURN true;

end;

'AppFrame.Betrieb' : begin

// Deskriptor des Hauptfensters laden

gFrmMain # \$AppFrame.Betrieb;

vA # 'Business Control'+

StrChar(32,30)+

'Datenbank: '+DbName(_DbAreaAlias)+

StrChar(32,30)+

'angemeldet als: '+gUsername+

StrChar(32,20)+

'am: '+Cnvad(today);

gFrmMain->wpcaption # vA;

// Usersettings holen

```
Lib_GuiCom:RecallWindow(gFrmMain);
```

```
StartMdi('Betrieb');
```

```
if (Entwicklerversion()) and (Strcut(set.Bild.Hintergrund,2,1)<>':') then
```

```
  gFrmMain->wpPictureName # '*'..' + Set.Bild.Hintergrund;
```

```
else
```

```
  gFrmMain->wpPictureName # '*' + Set.Bild.Hintergrund;
```

```
Lib_Jobber:Init();
```

```
RAUS('EvtInit')
```

```
RETURN true;
```

```
end;
```

```
Lib_GuiCom:GetAlternativeName('AppFrame.BDE') : begin
```

```
  gFrmMain # $AppFrame.BDE;
```

```
  vA # 'Business Control'+
```

```
    StrChar(32,30)+
```

```
    'Datenbank: '+DbName(_DbAreaAlias)+
```

```
    StrChar(32,30)+
```

```
    'angemeldet als: '+gUsername+
```

```
    StrChar(32,20)+
```

```
    'am: '+Cnvad(today);
```

```
  gFrmMain->wpcaption # vA;
```

```
  StartMdi('BDE');
```

```
Lib_Jobber:Init();  
  
RAUS('EvtInit')  
  
RETURN true;  
  
end;
```

```
Lib_GuiCom:GetAlternativeName('Mdi.Hauptmenue') : begin  
  
  gMenuName # 'Main';  
  
  gFrmMain->wpMenuname # gMenuName;  
  
  gMenu # gFrmMain->WinInfo(_WinMenu);  
  
  Lib_GuiCom:TranslateObject(gMenu);  
  
  Lib_GuiCom:TranslateObject(aEvt:obj);
```

```
// Fontgröße pro User...
```

```
if (Usr.Font.Size<>0) then begin  
  
  vFont # $TV.Hauptmenue->wpfont;  
  
  vFont:Size # Usr.Font.Size * 10;  
  
  $TV.Hauptmenue->wpfont # vFont;  
  
end;
```

```
RAUS('EvtInit')  
  
RETURN true;  
  
end;
```

```
Lib_GuiCom:GetAlternativeName('Mdi.Workbench') : begin
```

```

gMenuName # 'Main';

Lib_GuiCom:ObjSetPos(aEvt:Obj, gFrmMain->wpArearight - 200, gFrmMain->wpAreaBottom -340);

gFrmMain->wpMenuname # gMenuName;

gMenu # gFrmMain->WinInfo(_WinMenu);

Lib_GuiCom:TranslateObject(gMenu);

Lib_GuiCom:TranslateObject(aEvt:obj);

RAUS('EvtInit')

RETURN true;

end;

```

```

Lib_GuiCom:GetAlternativeName('Mdi.Cockpit') : begin

gMenuName # 'Main';

Lib_GuiCom:ObjSetPos(aEvt:Obj, gFrmMain->wpArearight - 315, 0);

gFrmMain->wpMenuname # gMenuName;

gMenu # gFrmMain->WinInfo(_WinMenu);

Lib_GuiCom:TranslateObject(gMenu);

// Lib_GuiCom:TranslateObject(aEvt:obj);

RAUS('EvtInit')

RETURN true;

end;

```

```

Lib_GuiCom:GetAlternativeName('Mdi.Notifier') : begin

gMenuName # 'Main';

Lib_GuiCom:ObjSetPos(aEvt:Obj, gFrmMain->wpArearight - 315, 0);

```

```
gFrmMain->wpMenuname # gMenuName;  
gMenu # gFrmMain->WinInfo(_WinMenu);  
Lib_GuiCom:TranslateObject(gMenu);  
Lib_GuiCom:TranslateObject(aEvt:obj);  
RAUS('EvtInit')  
RETURN true;  
end;
```

```
Lib_GuiCom:GetAlternativeName('Mdi.Betrieb'): begin
```

```
gMenuName # 'Main.Betrieb';  
gFrmMain->wpMenuname # gMenuName;  
gMenu # gFrmMain->WinInfo(_WinMenu);  
Lib_GuiCom:TranslateObject(gMenu);  
Lib_GuiCom:TranslateObject(aEvt:obj);  
RAUS('EvtInit')  
RETURN true;  
end;
```

```
Lib_GuiCom:GetAlternativeName('Mdi.MC9090'): begin
```

```
gMenuName # 'Main.Betrieb';  
gFrmMain->wpMenuname # gMenuName;  
gMenu # gFrmMain->WinInfo(_WinMenu);  
Lib_GuiCom:TranslateObject(gMenu);  
Lib_GuiCom:TranslateObject(aEvt:obj);
```

```
RAUS('EvtInit')
```

```
RETURN true;
```

```
end
```

```
otherwise begin
```

```
if (StrFind(aEvt:Obj->wpName,'Frame.MC9090',1, _StrCaseIgnore)>0) then begin
```

```
    // Deskriptor des Hauptfensters laden
```

```
    //gFrmMain # $Frame.MC9090;
```

```
    gFrmMain # aEvt:Obj;
```

```
    Lib_guiCom:ObjSetPos(gFrmMain, -4,-30);
```

```
    /*
```

```
    vA # 'Business Control'+
```

```
        StrChar(32,30)+
```

```
        'Datenbank: '+DbName(_DbAreaAlias)+
```

```
        StrChar(32,30)+
```

```
        'angemeldet als: '+gUsername+
```

```
        StrChar(32,20)+
```

```
        'am: '+Cnvad(today);
```

```
    gFrmMain->wpcaption # vA;
```

```
    StartMdi('MC9090');
```

```
    gFrmMain->wpPictureName # '*' + Set.Bild.Hintergrund;
```

```
    */
```

```
RAUS('EvtInit')
```

```
RETURN true;
```



```

    end;

end;

end; // ...case


// UnterObjektliste aufbauen
w_Objects # CteOpen(_CteList);
vTmp # aEvt:Obj->WinInfo(_WinFirst,0);
if (vTmp<>0) then Lib_GuiCom:BuildObjectList(w_objects, vTmp);


// if (gMDI<>gMDIMathCalculator) then begin
    w_Obj2Scale # CteOpen(_CteList);
    if (vTmp<>0) then Lib_GuiCom:BuildScaleList(w_obj2Scale, aEvt:Obj);
// end;


gMDI # aEvt:obj;


// ALLES übersetzen
//if (gFile<>904) then Lib_GuiCom:TranslateObject(aEvt:Obj);
Lib_GuiCom:TranslateObject( gMDI );


if (gZLList<>0) and (w_NoList=n) then begin
    // Fontgröße pro User...
    if (Usr.Font.Size<>0) then begin

```

```
vFont # gZLList->wpfont;  
vFont:Size # Usr.Font.Size * 10;  
gZLList->wpfont # vFont;  
end;
```

```
if (gZLList->wpdisabled=false) then begin //and ((Mode=c_ModeList) or (Mode=c_ModeEdList)) then begin
```

```
if (gZLList->wpDbLinkFileNo=0) and (w_Auswahlmode) then begin
```

```
    // bei "Auswahlfenstern" die Sortierung so beibehalten
```

```
if (gZLList->wpDbLinkFileNo=0) and (gZLList->wpdbfilter=0) then begin
```

```
    vVorgabeKey # 0;
```

```
    if (w_Auswahlmode=n) then begin
```

```
        vVorgabeKey # gZLList->wpdbkeyno;
```

```
    end;
```

```
end;
```

```
end;
```

```
Lib_GuiCom:RecallList(gZLList);    // Usersettings holen
```

```
if (HdlInfo(gZLList,_HdlExists)=1) and
```

```
(gZLList->wpDbLinkFileNo=0) then begin
```

```
    if (vVorgabekey<>0) then gKey # vVorgabeKey;
```

```
    vDLSort # gMDI->winSearch('DL.Sort');
```

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

```

vDLSort->wpautoupdate # false;

// Sortiermenü setzen

vDLSort->WinLstDatLineRemove(_WinLstDatLineAll);

vFilter # RecFilterCreate(901,2);

RecFilterAdd(vFilter,1,_FltAND, _FltEq, gFile);

Erx # RecRead(901,2,_RecFirst,vFilter);

WHILE (Erx=_rOk) do begin

    vX # vX + 1;

    vDLSort->WinLstDatLineAdd(Prg.Key.Name);

//10.02.2020      if (Prg.Key.EchteKeyNr=0) then

//      vA # CnvAl(Prg.Key.Key,_fmtNumLeadZero,0,3)+'|'+Prg.Key.SuchProzedur

//      else

//      vA # CnvAl(Prg.Key.Key,_fmtNumLeadZero,0,3)+'|'+Prg.Key.SuchProzedur;

    if (Prg.Key.EchteKeyNr=0) then Prg.Key.EchteKeyNr # Prg.Key.Key;

    vA # CnvAl(Prg.Key.Key,_fmtNumLeadZero,0,3)+'|'+CnvAl(Prg.Key.EchteKeyNr,_fmtNumLeadZero,0,3);

//debug('Key/ID = '+aint(gKey)+'/'+aint(gKeyID));

    vDLSort->WinLstCellSet(vA,2,_WinLstDatLineLast);

    if ((gKeyID<>0) and (gKeyID=Prg.Key.EchteKeyNr)) or

        ((gKeyID=0) and (gKey=Prg.Key.Key)) then begin

        if (gKeyID=0) then gKeyID # gKey;

        vDLSort->wpcurrentint # vX;

        gSuchProc # Prg.Key.SuchProzedur;

    end;

    Erx # RecRead(901,2,_RecNext,vFilter);

END;

RecFilterDestroy(vFilter);

```

```

Lib_GuiCom:ZLSetSort(0, gKeyID);      // Sortierung setzen

vDLSort->Winupdate(_WinUpdOn,_WinLstRecEvtSkip);

end;

end;

//      RefreshMode();                // Menü & Buttons setzen

end;

//      gZLList->WinFocusSet(true);
//      vHdl # Winsearch(gMDI,'ed.Suche');
//      if (vHdl<>0) then
//          $ed.Suche->Winfocusset(true)
//      else
//          gZLList->WinFocusSet(true);
end;

// bei Selektionen Schluessel auf Selektion setzen

/*** 28.1.2011

if (gZList<>0) and (w_NoList=n) then begin
    if (gZLList->wpdbselection<>0) then begin
        gKey # gZLList->wpdbselection;
//debug('set c '+aint(gKey));

    end;

end;

```

***/

RAUS('EvtInit')

RETURN true;

end;

//=====

// EvtCreated

//

//=====

sub EvtCreated(

 aEvt : event; // Ereignis

) : logic;

local begin

 vTmp : int;

 vHdl1,vHdl2 : int;

end;

begin

 REIN('EvtCreated')

// 12.11.2014

if (w_Command='NEWREPOS') then begin

 RecRead(gFile,0,_recId, cnvia(w_Cmd_Para));

//debugx('inner repos KEY401 '+Mode);

```

w_Command # ";

vTMP # aEvt:Obj->wpDBRecBuf(gFile);

if (vTMP<>0) then begin

//debugx('copy '+w_name);

//   RecBufcopy(gFile, vTMP);

//   if (gZLList<>0) then gZLList->winupdate(_Winupdon, _WinlstRecDoSelect); // 12.11.2014

   if (gZLList<>0) then

      gZLList->winupdate(_Winupdon, _WinlstRecDoSelect | _WinLstRecFromRecId); // 18.07.2018 (Rep

end;

end;

vTmp # winfocusget();

if (vTmp=0) then begin

   RAUS('EvtCreated')

   RETURN true;

end;

if (HdlInfo(vTmp, _HdlExists)<>1) then begin

   RAUS('EvtCreated')

   RETURN true;

end;

// App-Frame start?

if (aEvt:Obj=gFrmMain) then begin

```

```

Usr_Data:Init();

// AFX "AppFrame.Created"
RunAFX('AppFrame.Created','');

// TODO : welches MDI soll vorne sein

// Socket Handle anfordern
// if (gUsername='AH') or (gUsername='FS') then

// Logging initialisieren:
Lib_Logging:InitLogging();

gRemotePort # 8000;
REPEAT
    gRemoteSocket # SckListen('127.0.0.1',gRemotePort); // war 37
    if (gRemoteSocket<=0) then inc(gRemotePort);
UNTIL (gRemoteSocket>0) or (gRemotePort>=8500);
if (gRemoteSocket<0) then begin
    gRemotePort # 0;
debugx('SC-Client konnte keinen Remoteport öffnen!');
end;
//debugx(aint(gRemotePort)+' = '+aint(gRemoteSocket));

Set_Main:Check(); // 05.05.2021 AH

```

RAUS('EvtCreated')

RETURN true;

end;

// BugFix(__PROCFUNC__, aEvt:Obj);

if (vTmp->wpname='DUMMYNEW') then begin

vHdl1 # VarInfo(windowbonus);

vHdl2 # Lib_GuiCom2:GetDBVarHandle(aEvt:Obj);

if (vHdl2<>0) then VarInstance(windowbonus, vHdl2);

if (Mode=c_ModeView) or (mode=c_ModeBald + c_modeView) then begin // Wieder auf "Edit" focusieren

vTmp # 0;

vTmp # gMdi->winsearch('Edit');

if (vTmp<>0) and (vTmp->wpdisabled) then

if (gMdi->winsearch('EditErsatz')<>0) then

vTmp # gMdi->winsearch('EditErsatz');

if (vTmp<>0) then begin

vTmp->winfocusset(true);//false);

end;

end;

if (vHdl1<>0) then VarInstance(windowbonus, vHdl1);

end;

RAUS('EvtCreated')


```

    RETURN(true);

end;

//=====

// EvtClose

//      Schliessen eines Fensters

//=====

sub EvtClose(

    aEvt          : event;      // Ereignis

): logic

local begin

    vParent : int;

    vName   : alpha;

    vHdl    : int;

    vBuf     : int;

    vOK      : logic;

    vTmp     : int;

    vFocus   : int;

    vBonus   : int;

    vMode    : alpha;

end;

begin

    REIN('EvtClose')

//debugx('EvtClose '+aEvT:obj->wpname+' '+w_name+' '+Mode);

```

```
//debugx(aEvt:obj->wpname+' CLOSE  w_Name:'+w_name+' mdi:'+aint(gMDI));
```

```
//if (aEvt:obj->wpname='AppFrame') then begin
```

```
// VarInstance(WindowBonus, cnvIA(aEvt:Obj->wpcustom));
```

```
//debugx('restore '+w_name+' '+aint(gMDI));
```

```
//end;
```

```
if ((aEvt:Obj->wpname='**AppFrame')=false) then
```

```
    BugFix(__PROCFUNC__, aEvt:Obj, true); // 03.03.2021 wenn GESAMT-CLOSE, dann haben die MDI
```

```
Crit_Prozedur:Manage();
```

```
// BUGFIX vTMP # VarInfo(Windowbonus);
```

```
// " if (aEvt:Obj->wpcustom<>" and (vTmp<>cnvia(aEvt:Obj->wpcustom)) then begin
```

```
// "  VarInstance(WindowBonus,cnvIA(aEvt:obj->wpcustom));
```

```
// " end;
```

```
vMode # Mode;
```

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

```
    vName # gUsergroup;
```

```
    if (vName='') then vName # gUserName;
```

```
    if (vName='BETRIEB') or (vName='BETRIEB_TS') or (vName = 'BDE') then begin
```

```
        Lib_GuiCom:RememberWindow(gFrmMain);
```

```
        App_Extras:HardLogout();
```

```
//    Org_Data:Killme();
```

```
//    Winhalt();
```

end;

end;

// Quickbar

if (Winsearch(aEvt:obj, 'gt.Quickbar')<>0) then Lib_Guicom:RememberQuickBar();

// Appframe schliesst kontrolliert alle Unterfenster...

if (aEvt:Obj->wpname='AppFrame') then begin

if (Lib_GuiCom2:CloseAll(true)=false) then begin

RAUS('EvtClose')

RETURN false;

end;

if (Lib_GuiCom2:CloseAll(false)=false) then begin

RAUS('EvtClose')

RETURN false;

end;

Lib_GuiCom:RememberWindow(aEvt:obj); // 28.10.2021 AH

RunAFX('App.Close','');

if (gRemoteSocket>0) then SckClose(gRemoteSocket);

// 30.06.2021 AH Lib_GuiCom:RememberWindow(aEvt:obj);

RAUS('EvtClose')

RETURN true;

end;

// Parentfenster koennen nicht geschlossen werden

if (w_Child<>0) then begin

RAUS('EvtClose')

// StartMdi(a'App Frame');

RETURN false;

end;

if (vMode<>c_ModeCancel) and (aEvt:Obj->wpname='AppFrame') then begin

Lib_GuiCom:RememberWindow(aEvt:obj);

RAUS('EvtClose')

RETURN true;

end;

if (vMode<>c_ModeCancel) and

(aEvt:Obj=gMDIMathCalculator) or (aEvt:Obj=gMDIWorkbench) or (aEvt:Obj=gMDIMenu) or (aEvt:Obj=gMDIStatusBar)

Lib_GuiCom:RememberWindow(aEvt:obj);

RAUS('EvtClose')

RETURN true;

end;

// MDI-Unterfenster?

if (aEvt:Obj->wpname<>'AppFrame') and

(w_CopyToBuf=0) and // 15.04.2021 AH: für EDIT in MoreBufs und X-Close

(mode<>"") and

(mode<>c_ModeView) and

(Mode<>c_ModeCancel) then begin

vBonus # VarInfo(Windowbonus);

Action(c_modeCancel+'X'); // Fenster NICHT schliessen

if (Mode<>c_ModeCancel) and (vBonus=Varinfo(WindowBonus)) then begin // AHGWS

RAUS('EvtClose')

// StartMdi(a'App Frame');

RETURN false;

end;

// Anfängliches Fenster ist ggf. schon weg!!! 01.03.2021

varInstance(Windowbonus, vBonus);

end;

if (gPrefix<>") then begin

vOK # y;

try begin

ErrTryIgnore(_rlocked,_rNoRec);

ErrTryCatch(_ErrNoProcInfo,y);

ErrTryCatch(_ErrNoSub,y);

vOK # Call(gPrefix+'_Main:EvtClose', aEvt);

end;

if (vOK=false) then begin

RAUS('EvtClose')

// StartMdi(a'App Frame');

RETURN false;

end;

end;

// 18.01.2018 AH:

Lib_MoreBufs:Close();

// Sortierung nicht bei "Auswahl" merken !!!

if ((w_AuswahlMode=n) or (w_Context<>"")) and

(gZLList<>0) and (w_NoList=n) then Lib_GuiCom:RememberList(gZLList);

if (gZLList<>0) and (w_NoList=n) then begin // ggf.Filter wieder destroyen

gZLList->wpautoupdate # false;

if (gZLList->wpDbFilter<>0) then begin

RecFilterDestroy(gZLList->wpDbFilter);

gZLList->wpDbFilter # 0;

end;

vHdl # gZLList->wpDbSelection; // Selektionsmengen entfernen

if (vHdl<>0) then begin

gZLList->wpDbSelection # 0;

SelClose(vHdl);

end;

if (w_Sel2Name<>") then begin // temp. Selektionen entfernen

SelDelete(gFile,w_sel2Name);

```

    w_Sel2Name # "";

end;

if (w_SelName<>") then begin // temp. Selektionen entfernen

    SelDelete(gFile,w_selName);

    w_SelName # "";

end;

end;

gFile # 0;

gPrefix # "";

gZLList # 0;


// Pflichtfeldliste loeschen

Lib_Pflichtfelder:PflichtfelderListeLoeschen(var w_Pflichtfeld, var w_Obj4auswahl);


// UnterObjektliste löschen

if (w_Objects<>0) then w_objects->CteClear(true);

if (w_Obj2Scale<>0) then w_obj2Scale->CteClear(true);


if (mode<>") then Lib_GuiCom:RememberWindow(aEvt:obj);


// Elternbeziehung aufheben?

if (w_Parent<>0) then begin

    if (gMdi<>0) then begin

        ErrTrycatch(_ErrHdllInvalid,y);

        try begin

```

```

    gMDI->wpvisible # n;

end;

if (errGet() <> _ErrOK) then begin
    ErrSet(_errOK);

    gMDI # 0;

    RAUS('EvtClose')

    RETURN true;

end;

end;


vTmp # VarInfo(Windowbonus);

if (vTmp>0) and (w_Parent>0) then begin
    if (w_parent->wpcustom <> "") then begin
        VarInstance(WindowBonus,cnvIA(w_parent->wpcustom));

        w_Child # 0;


        if (Mode=c_ModeList) or (Mode=c_ModeList2) then begin
            if (gZLList<>0) and (gFile<>0) then vBuf # RekSave(gFile);

            end;


            VarInstance(WindowBonus,vTmp);

            end;

//    w_Parent->wpdisabled # n;

// 17.03.2021 AH    w_Parent->WinUpdate(_WinUpdActivate);    macht EVTTERM

end;

```



```

    if (vBuf<>0) then RekRestore(vBuf);

end;

RAUS('EvtClose')

RETURN true;

end;

//=====

// EvtTerm

//      Terminieren eines Fensters

//=====

sub EvtTerm(

    aEvt          : event;    // Ereignis

): logic

local begin

    vI           : int;

// vTermProc : alpha;

// vTermPara : alpha;

    vHdl        : handle;

// vHdlBTerm : handle;

    vHdlBAkt    : handle;

    vXParent     : int;

    vXTermProc   : alpha;

```

```

vXTermProcPara : alpha;

vXAufruferMDI : int;

end;

begin

    REIN('EvtTerm')

    //debug('TERM_A:'+aEvt:obj->wpname+ ' prc:'+w_termproc+ ' name:'+w_name+ ' parent:'+cnvai(w_pare

    // vHdlBAkt # BugFix(__PROCFUNC__, aEvt:Obj, true);

    // if (aEvt:obj->wpcustom<>") then VarInstance(WindowBonus,cnvIA(aEvt:Obj->wpcustom));

    //debug('TERM_B:'+aEvt:obj->wpname+ ' prc:'+w_termproc+ ' name:'+w_name+ ' parent:'+cnvai(w_pare

    // Global alle Fenstereinstellungen (Größe & Position) speichern [15.03.2010/PW]

    // Lib_GuiCom:RememberWindow( aEvt:obj );

    //xf (w_Parent<>0) then w_Parent->wpDisabled # false;

    //winsleep(1000);

    //debug('TERM_C:'+aEvt:obj->wpname+ ' prc:'+w_termproc+ ' name:'+w_name+ ' parent:'+cnvai(w_pare

    // if (aEvt:obj=gMDI) then begin

    //     gMDI # 0;

    //debug('CLEAR');

    // end;

    case(aEvt:Obj) of

        gMDIMenu : gMDIMenu # 0;

        gFrmMain : begin

```

```

//Lib_GuiCom:RememberWindow(gFrmMain);

// ai 21.6.2012    gMDIMenu        # 0;

end;


gMdiWorkbench : begin

//Lib_GuiCom:RememberWindow(gMDIWorkbench);

gMDIWorkbench # 0;

end;


gMDINotifier      : begin

vHdl # Winsearch(gMDINotifier, 'rl.Messages');

Lib_GuiCom:RememberList(vHdl);

//Lib_GuiCom:RememberWindow(gMdiNotifier);

gMDINotifier      # 0;

//SysTimerClose(gTimer);

//gTimer # 0;

end;


gMdiTermine      : gMdiTermine      # 0;

gMdiPara         : gMdiPara         # 0;

gMdiPrj          : gMdiPrj          # 0;

gMdiAdr          : gMdiAdr          # 0;

gMdiMat          : gMdiMat          # 0;

gMdiMsl          : gMdiMsl          # 0;

gMdiAuf          : gMdiAuf          # 0;

gMdiLfs          : gMdiLfs          # 0;

```

```

gMdiVSP      : gMdiVSP      # 0;
gMdiEin      : gMdiEin      # 0;
gMdiBdf      : gMdiBdf      # 0;
gMdiRSO      : gMdiRSO      # 0;
gMdiBAG      : gMdiBag      # 0;
gMdiQS       : gMdiQS       # 0;
gMdiErl      : gMdiErl      # 0;
gMdiEKK      : gMdiEKK      # 0;
gMdiOfp      : gMdiOfp      # 0;
gMdiEre      : gMdiERe      # 0;
gMdiArt      : gMdiArt      # 0;
gMdiMath     : gMdiMath     # 0;
gMdiMathVar  : gMdiMathVar  # 0;
gMdiMathAlphabet : gMdiMathAlphabet # 0;
gMdiMathVarMiniPrg : gMdiMathVarMiniPrg # 0;
gMdiRsoKalender : gMdiRsoKalender # 0;
gMdiGantt    : gMdiGantt    # 0;
gMdiMathCalculator : gMdiMathCalculator # 0;

end;

// aktuellen Pfad wieder auf das Clientverzeichnis setzen
FsiPathChange(gFsiClientPath);

//debugx('EvtTerm....');

vHdIBAkt     # BugFix(__PROCFUNC__, aEvt:Obj, true);
vXParent     # w_Parent;
vXTermProc   # w_TermProc;
vXTermProcPara # w_TermProcPara;

```

```
vXAufruferMDI # w_AufruferMDI;
```

```
if (vHdlBAkt<>0) then VarInstance(Windowbonus,vHdlBAkt);
```

```
// AusAuswahlprozedur starten?
```

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

```
ErrTrycatch(_ErrHdlInvalid,y);
```

try begin

```
vXparent->wpname # vxparent->wpname;
```

end;

```
if (errGet() <> _ErrOK) then begin
```

```
ErrSet(_errOK);
```

vXParent # 0;

end;

end;

```
// gibt es ein PARENT?
```

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

```
vXParent->wpdisabled # n;
```

```
vXParent->winupdate(_Winupdon|_WinUpdActivate); // 17.03.2021 AH: Setzt dbRecBufs und VarInstan
```

```
If (vXTermProc<>"") then begin
```

```
WinSearchPath(vXParent);
```

```
VarInstance(Windowbonus,cnvia(vXParent->wpcustom)); // SET Parent-Window <<<<<<<<<<<<<
```

RefreshMode(y);

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

```
Call(vXTermProc, vXTermProcPara) // Aus... Prozedur aufrufen
```

```

else

    Call(vXTermProc);          // Aus... Prozedur aufrufen

end

else begin  // KEINE TermProc

    if (vXParent->wpvisible=n) then vXParent->wpvisible # true;

    WinSearchPath(vXParent);

    if (cnvia(vXParent->wpcustom)<>0) then VarInstance(Windowbonus,cnvia(vXParent->wpcustom)); //

    RefreshMode(y);

end;

if (HdlInfo(gZLList,_HdlExists)>0) then begin

    if (gZLList<>0) and

        ((Mode=c_ModeList) or (Mode=c_ModeView)) then begin  // ggf. Liste refreshen

            gZLList->WinUpdate(_WinUpdOn, _WinLstRecFromRecid | _WinLstRecDoSelect);

            gZLList->winfocusset(true);

        end;

end;

if (gPrefix<>") and

    ((Mode=c_ModeEdit) or (Mode=c_ModeNew) or (Mode=c_ModeEdit2) or (Mode=c_ModeNew2)) then

    if (VarInfo(windowbonus)>0) then begin

        vHdl # Winfocusget();

//    Winsleep(100);  // WORKAROUND 12.07.2012 AI

        if (vHdl<>0) then Call(gPrefix+'_Main:Refreshifm',vHdl->wpname);

    end;

end;

```

```
// VarInstance(Windowbonus,vHdlBTerm); //
```

```
end
```

```
else begin // KEIN Parent !
```

```
// gibt's einen Aufrufer? ja, dann dahin zurück
```

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

```
    if (HdlInfo(vXAufruferMDI, _HdlExists)>0) then begin
```

```
        vXAufruferMDI->WinUpdate(_WinUpdActivate);
```

```
        RAUS('EvtTerm')
```

```
        RETURN true;
```

```
    end;
```

```
end;
```

```
// einfach wieder MainMenu starten
```

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

```
    if (HdlInfo(gMDIMenu, _HdlExists)=1) then gMDIMenu->winupdate(_WinUpdActivate);
```

```
end;
```

```
end;
```

```
// 01.07.2016 AH:
```

```
if (StrCnv(aEvt:Obj->wpname, _Strupper) = StrCnv(Lib_GuiCom:GetAlternativeName('MDI.Betrieb'), _Strupper)) then
```

```
    App_Betrieb:TransCheck(false);
```

```
end;
```

```
RAUS('EvtTerm')
```

```
RETURN true;
```

```

end;

//=====

// EvtDragInit

//      Sourceobjekt auswählen

//=====

sub EvtDragInit(

    aEvt      : event;  // Ereignis

    aDataObject : int;   // Drag-Datenobjekt

    aEffect    : int;    // Rückgabe der erlaubten Effekte (_WinDropEffectNone = Cancel)

    aMouseBtn  : int     // Verwendete Maustasten

) : logic

local begin

    vTxtBuf : int;

    vFormat : int;

end;

begin

    BugFix(__PROCFUNC__, aEvt:Obj);

    if (gFile=0) then RETURN false;

    RecRead(gFile, 0, _RecID, aEvt:obj->wpDbRecID);

    if (Lib_Workbench:CreateDragData(gFile, aDataObject)) then begin

        aEffect # _WinDropEffectCopy | _WinDropEffectMove;

```



```

    RETURN (true);

end

RETURN false;

end;

//=====

// EvtDragTerm

//      D&D beenden

//=====

sub EvtDragTerm(

    aEvt      : event;  // Ereignis

    aDataObject : int;   // Drag-Datenobjekt;Durchgeführte Dragoperation (_WinDropEffectNone = abgebrochen)

    aEffect     : int

) : logic

local begin

    vFormat : int;

    vTxtBuf : int;

end;

begin

    BugFix(__PROCFUNC__, aEvt:Obj);

    if (aDataObject->wpFormatEnum(_WinDropDataText)) then begin

        // Format-Objekt ermitteln.

        vFormat # aDataObject->wpData(_WinDropDataText);

```

```

// Format schliessen.

aDataObject->wpFormatEnum(_WinDropDataText) # false;

end;

end;

//=====

// EvtPosChanged

//

//=====

sub EvtPosChanged(

aEvt      : event;  // Ereignis

aRect      : rect;  // Größe des Fensters

aClientSize : point;  // Größe des Client-Bereichs

aFlags      : int    // Aktion

) : logic

local begin

vRect      : rect;

vHdl        : int;

end

begin

//debug('EVTPOSCHANGED:'+aEvt:Obj->wpname+' '+gMDi->wpname+' '+w_name+' '+gZLList->wpname+' '+gZLList->wname);

/** Organigramm

// WORKAROUND

if (aEvt:obj=gMdiMenu) then begin

vHdl # Winsearch(aEvt:obj,'TV.Hauptmenue');

```

```

vRect      # vHdl->wpArea;

vRect:right  # aRect:right-aRect:left-4;

vRect:bottom  # aRect:bottom-aRect:Top-28;

vHdl->wparea  # vRect;

RETURN true;

end;

**/

BugFix(__PROCFUNC__, aEvt:Obj);

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 (w_NoList) then RETURN true;  // 08.01.2020 AH

if (aFlags & _WinPosSized != 0) then begin

    if (gZLList<>0) then vHdl # gZLList;

    else if (gDataList<>0) then vHdl # gDataList

```

```
else RETURN true;
```

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

```
vRect:right  # aRect:right-aRect:left-4;
```

```
vRect:bottom  # aRect:bottom-aRect:Top-28 - w_QBHeight;
```

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

```
end;
```

```
RETURN (true);
```

```
end;
```

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

```
//=== ENDE DER EVENTS =====
```

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

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

```
// MAIN
```

```
//  Haupt-Applikation
```

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

```
MAIN
```

```
local begin
```

```
  vName  : alpha;
```

```
  vGroup : alpha;
```

```
end;
```

```
begin
```

```

// ALLE EVENTS ERLAUBEN:

WinEvtProcessSet(_WinEvtAll,true);


// Uhrzeit vom Server holen

DbacControl(_DbacTimeSync);


VarAllocate(VarSysPublic); // public Variable allokieren

gCodepage # _Sys->spCodepageOS;

// _Sys->spOptions # _deadLockRTE; // 27.05.2021 AH 08.07.2021 AH RAUS !!!!!


// Debug initialisieren

Lib_Debug:InitDebug();


// UserInfo(_UserCurrent);

// if (UserInfo(_Username)='AH') then DEBUG('DebugInit: ok');


// Initialisieren

Liz_Data:SystemInit(var Set.Module);


//if (gUsername='AH') then set.Module # "";

// _app->wpflags # _app->wpflags | _Winappbuf2fldoff; // 20.09.2019 AH: TEST!!!


_App->wpPicDpiDefaultX # 300;

_App->wpPicDpiDefaultY # 300;

_App->wpOleDropEffectStandard # _WinDropEffectcopy;

```

```
// _App->wpflags # _App->wpflags | _WinAppWaitCursorEvtOs;
```

```
_App->wpflags # _App->wpflags | _WinAppWaitCursorEvtArrow;
```

```
_App->wpflags # _App->wpflags | _WinAppEditTextROContextMenu;
```

```
// _App->wpflags # _App->wpflags | _WinAppRadioFocusChecked;
```

```
// _WinAppEditSelectAll
```

```
//_App->wpFlags # _App->wpFlags | _WinAppNotebookPageDelayed; damit gehen Textbuffer im Auftrag
```

```
//_App->wpFlags # _App->wpFlags | _WinAppGlobalvar
```

```
RecBufClear(999);
```

```
// UserInfo(_UserCurrent);
```

```
// if (UserInfo(_Username)='AH') then DEBUG('SystemInit: ok');
```

```
// AFX initialisieren
```

```
Lib_SFX:InitAFX();
```

```
// SOUND initialisieren
```

```
Lib_Sound:Initialize();
```

```
// ODBC initalizieren
```

```
Lib_ODBC:Init();
```

```
// User aufräumen
```

```
App_Extras:UserCleanup();
```

```
App_Extras:EntwicklerVersion();
```

```
// UserInfo(_UserCurrent);
```

```
// if (UserInfo(_Username)='AH') then DEBUG('Tapilnit: ok');
```

```
// INI Dateien leeren??
```

```
Set.Client.Pfad # FSIPAth();
```

```
if (Set.Reset.INI) and (DbalInfo(_DbidUserCount)=1) then begin
```

```
    RecRead(903,1,_RecLock);
```

```
    Set.Client.Pfad # FSIPAth();
```

```
    Set.Reset.INI # n;
```

```
    RecReplace(903,_recUnlock);
```

```
    Lib_Initialize:ReadIni();
```

```
    App_Extras:INICleanUp();
```

```
end;
```

```
Usr_Data:MerkeKleinstenClient();
```

```
gUserGroup # UserInfo(_UserGroup,gUserID);
```

```
if (gUserGroup='') then gUserGroup # gUserName;
```

```
vGroup # gUserGroup;
```

```
GV.Sys.Username # gUsername;
```

```
GV.Sys.UserID # gUserID;
```

```
Usr.Username # gUsername;
```

```
RecRead( 800, 1, 0 );
```

```
gUserSprachnummer # Usr.Sprachnummer;
```

```
gUserSprache # Set.Sprache1.Kurz;  
case (gUserSprachnummer) of  
  1 : gUserSprache # Set.Sprache1.Kurz;  
  2 : gUserSprache # Set.Sprache2.Kurz;  
  3 : gUserSprache # Set.Sprache3.Kurz;  
  4 : gUserSprache # Set.Sprache4.Kurz;  
  5 : gUserSprache # Set.Sprache5.Kurz;  
end;
```

```
Set.Client.Pfad # FSIPath();
```

```
Usr_Data:LoadINI();
```

```
Crit_Prozedur:Manage(TRUE);    // 26.03.2015 AH : einmalig SQL-Prüfen
```

```
Lib_Rec:ClearFile( 773 );      // 2022-09-20 AH
```

```
Lib_Rec:ClearFile( 774 );
```

```
Lib_Rec:ClearFile( 827 );
```

```
Lib_Cache:Init();
```

```
// globale Errorbehandlung
```

```
ErrCall('Lib_Debug:RunTimeErrorCatcher');
```

```
// Speziell Programmiersettings...
```

```
// if (Entwicklerversion()) then begin
```

```
//   Set.LFS.mitVersandYN # y;
```



```
// end;
```

```
//@ifdef Prog_ReadOnly
```

```
// if (cReadOnlyCondition) then TRANSON;
```

```
//@Endif
```

```
if (SFX_Std_Anonymisiere:FirstOpen()) then begin
```

```
    // Haupframe starten
```

```
    case (gUserGroup) of
```

```
        // BETRIEBSMASKE anzeigen
```

```
        'BETRIEB','BETRIEB_TS' : begin
```

```
            if (Usr_Data:CountThisUserThisPC(>1) then begin    // 02.07.2021 AH Proj. 2222/43/1
```

```
                WindialogBox(0,'Login','Dieser Betriebsuser ist hier schon angemeldet!', _WinlcoError, _Windialogok
```

```
            end
```

```
        else begin
```

```
            Set_Main:Check();
```

```
            Windialog('AppFrame.Betrieb',_Windialogcenter | _windialogapp);
```

```
        end;
```

```
    end;
```

```
// BDE MASKE ANZEIGEN
```

```
'BDE' : begin
```

```
    Windialog('AppFrame.BDE',_Windialogcenter | _windialogapp);
```

```
end;
```

```

// BETRIEBSMASKE anzeigen

'MC9090' : begin

  //Windialog('AppFrame.MC9090',_Windialogcenter | _windialogapp);

  vName # 'Frame.MC9090';

  REPEAT

    VarAllocate(Windowbonus);

    Lib_guiCom:ObjSetPos(gFrmMain, -4,-30);

    // ggf. anderes Objekt benutzen

    vName # Lib_GuiCom:GetAlternativeName(vName);

    Windialog(vName,_WinDialogCenter | _windialogapp);

    vName # "";

    if (gSelected=1) then vName # 'Frame.MC9090';
    if (gSelected=2) then vName # 'Frame.MC9090.2';
    if (gSelected=3) then vName # 'Frame.MC9090.3';
    if (gSelected=4) then vName # 'Frame.MC9090.4';
    if (gSelected=5) then vName # 'Frame.MC9090.5';
    if (gSelected=6) then vName # 'Frame.MC9090.6';
    if (gSelected=7) then vName # 'Frame.MC9090.7';

    VarFree(WindowBonus);

  UNTIL (vName="");

end;

```

```
// JOB-SERVERMASKE anzeigen
```

```
'JOB-SERVER' : begin
```

```
    Set_Main:Check();
```

```
    if (gUsername='FILESCANNER') then
```

```
        FileScanner_Frame()
```

```
    else
```

```
        Job_Frame();
```

```
end;
```

```
// standard User mit NORMALER Maske
```

```
otherwise begin
```

```
//    Set_Main:Check(); 05.05.2021 AH ins EvtCreated, wegen Performance
```

```
_App->wpvisible # false;
```

```
// 13.01.2016 AH
```

```
/** RAUS 30.06.2016
```

```
if (!_App->wpInstallCtxOffice) then
```

```
    _App->wpInstallCtxOffice # true;
```

```
if (!_App->wpInstallCtxOffice) then begin
```

```
    // Office-Erweiterung nicht installiert
```

```
    WinDialogBox(0, 'Fehler',
```

```

        'Office-Erweiterung konnte nicht installiert werden. Fehlerwert:' + StrChar(13) + StrChar(10) +
        ErrMapText(WinInfo(0, _WinErrorCode), 'DE', _ErrMapSys) + '(' +
        CnvAI(WinInfo(0, _WinErrorCode)) + ')', _WinlcoError, _WinDialogOK, 0);
    end;
**/

```

```

// APPLIKATIONSFENSTER STARTEN
Windialog('AppFrame',_Windialogcenter | _windialogapp);
_App->wpvisible # true;
end;
end;
end;

```

```

// Transaktion noch aktiv?
if (gBlueMode) then begin
    Transcount # Transcount + 1;
    TRANSBRK;
end;

```

```

gFrmMain # 0;
gMDI    # 0;

```

```

if (TransActive) then begin
    WHILE (TransActive) do
        TRANSBRK;

```

```

//@ifdef Prog_ReadOnly

```

```
//@else
```

```
    Msg(001103,"0,0,0,0);
```

```
//@Endif
```

```
end;
```

```
RunAFX('Logout',vGroup);
```

```
if (Set.Installname='BSC') then
```

```
    Lib_DotNetServices:Remote_Quit();
```

```
Lib_Cache:Term();
```

```
// SOUND beenden
```

```
Lib_Sound:Terminate();
```

```
// User aufräumen
```

```
App_Extras:UserCleanup();
```

```
// ODBC beenden
```

```
Lib_ODBC:Term();
```

```
// TAPI beenden
```

```
Lib_Tapi:TAPITerm();
```

```
// Job-Thread beenden
```

```
Lib_Jobber:Term();
```

// BCS_COM beenden

Lib_BCSCOM:UnloadDLL();

// Debugger beenden

Lib_Debug:TermDebug();

// AFX beenden

Lib_SFX:TermAFX();

// vom Organigramm abmelden

Org_Data:Killme();

// Rest aufräumen...

if (gUserINI<>0) then TextClose(gUserINI);

varfree(VarSysPublic);

Liz_Data:SystemTerm();

// Abschalten ausser bei Programmierern

if (vGroup<>'PROGRAMMIERER') then WinHalt();

// ST:

//WinDialogBox(0,'TODO','NICHT FREI WEGEN: BA-INPUT-KEIN SCHROTT',_WinIcoInformation,_WinDi

// AH:

//WinDialogBox(0,'TODO','NICHT FREI WEGEN: SQL-SYNC',_WinIcoInformation,_WinDialogOk,0)

```
// DS:
```

```
//WinDialogBox(0,'TODO','NICHT FREI WEGEN: BA-INPUT-KEIN SCHROTT',_WinIcoInformation,_WinDi
```

```
end;
```

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