```
@A+
//===== Business-Control ============================================
//
//  Prozedur    Form_Parse_Mat
//                  OHNE E_R_G
//  Info
//    parst die @-Kommandos in den Material Formularen
//
//
//  07.11.2012  AI  Erstellung der Prozedur
//  28.05.2013  TM  KÖRNUNG2 hinzugefügt
//  19.10.2020  TM  Auftragspos.-Vorgaben drucken wenn vorhanden (Änderung STD!) zu 2063/11
//
//  Subprozeduren
//  SUB AnalyseBereich(aWert : float; aMin : float; aMax : float; aDec : int;) : alpha;
//  SUB Analyse(aWert : float; aDec : int;) : alpha;
//  SUB Parse200(var aLabels : alpha; var aInhalt : alpha; var aZusatz : alpha; aText : alpha(4096); aKomb
//  SUB Parse200Multi(var aLabels : alpha; var aInhalt : alpha; var aZusatz : alpha; aObjName : alpha; aKo
//
//=====================================================================
@I:Def_Global
@I:Def_Form


//=====================================================================
//  AnalyseBereich
//=====================================================================
```

```
sub AnalyseBereich(
  aWert : float;
  aMin  : float;
  aMax  : float;
  aDec  : int;) : alpha;
local begin
  vOK   : logic;
end;
begin

  // keine Angabe
  vOK # (aMin=0.0) and (aMax=0.0);

  // nur min
  if (aMin<>0.0) and (aMax=0.0) then begin
    vOK # (aWert>=aMin);
    end
  // bis max
  else begin
    vOK # (aWert>=aMin) and (aWert<=aMax);
  end;

  if (vOK) then
    RETURN cnvaf(aWert, _FmtNumNoGroup,0, aDec)
  else
    RETURN 'F E H L E R';
```

```
end;


//=========================================================================
//  Analyse
//=========================================================================
sub Analyse(
  aWert : float;
  aDec  : int;) : alpha;
local begin
  vOK   : logic;
end;
begin
  if (aWert=0.0) then RETURN ' '
  RETURN cnvaf(aWert, _FmtNumNoGroup,0, aDec)
end;


//=========================================================================
//  Parse200
//=========================================================================
sub Parse200(
  var aLabels   : alpha;
  var aInhalt   : alpha;
  var aZusatz   : alpha;
  aText         : alpha(4096);
```

```
  aKombi        : logic) : int;
local begin
  Erx     : int;
  vZeilen : int;
  vFeld   : alpha(4096);
  vTitel  : alpha(4096);
  vPre    : alpha(4096);
  vPost   : alpha(4096);
  vAdd    : alpha(4096);
end;
begin

  // Sonderbefehl?
  if (StrCut(aText,1,2)='my') then begin
    try begin
      ErrTryIgnore(_rlocked,_rNoRec);
      ErrTryCatch(_ErrNoProcInfo,y);
      ErrTryCatch(_ErrNoSub,y);
      vZeilen # Call(FRM.Prozedur+':Parse',var aLabels, var aInhalt, var aZusatz, aText, aKombi);
      RETURN vZeilen;
    end;
  end;

  if (mat.Kommission !='') then begin
    erx # RecLink(401,200,16,0);
    if (erx > _rLocked) then
```

```
    RecBufClear(401);
end

vFeld   # Str_Token(aText, '|', 1);
vTitel  # Str_Token(aText, '|', 2);
vPost   # Str_Token(aText, '|', 3);
vPre    # Str_Token(aText, '|', 4);

case (StrCnv(vFeld, _StrUpper)) of
  'ADR.NAME'    :   vAdd  # Adr.Name;
  'ADR.ANREDE'  :   vAdd  # Adr.Anrede;
  'ADR.ZUSATZ'  :   vAdd  # Adr.Zusatz;
  'ADR.STRASSE'  :  vAdd  # "Adr.Straße";
  'ADR.ORT'  :      vAdd  # Adr.Ort
  'ADR.PLZ'  :      vAdd  # Adr.Plz;
  'ADR.LAND' :      vAdd  # Lnd.Name.L1;
  'ZEUGNIS' :       vAdd  # Mat.Zeugnisart;
  'MATERIALNR' :    vAdd # aint(Mat.Nummer);
  'CHARGENNR' :     vAdd  # Mat.Chargennummer;
  'COILNR'  :       vAdd  # Mat.Coilnummer;
  'WERKSNR'  :      vAdd  # Mat.Werksnummer;
  'RINGNR'  :       vAdd  # Mat.Ringnummer;
  'KOMMISSION' :    vAdd  # Mat.Kommission;
  'WERKSTOFFNR' :   vAdd  # Mat.Werkstoffnr;
  'GÜTE' :          vAdd  # "Mat.Güte";
  'ABMESSUNG' :     begin
```

```
            if (Mat.Dicke<>0.0) then vAdd # anum(Mat.Dicke, Set.Stellen.Dicke);
            if (Mat.Breite<>0.0) then begin
              if (vAdd<>'') then vAdd # vAdd + ' x ';
              vAdd # vAdd + anum(Mat.Breite, Set.Stellen.Breite);
              if ("Mat.Länge"<>0.0) then
                vAdd # vAdd + ' x ' + anum("Mat.Länge", "Set.Stellen.Länge");
            end;
          end;
'DICKE' :     begin
            if (Mat.Dicke=0.0) then RETURN 0;
            vAdd # anum(Mat.Dicke, Set.Stellen.Dicke);
          end;
'BREITE' :    begin
            if (Mat.Breite=0.0) then RETURN 0;
            vAdd # anum(Mat.Breite, Set.Stellen.Breite);
          end;
'LÄNGE' :     begin
            if ("Mat.Länge"=0.0) then RETURN 0;
            vAdd # anum("Mat.Länge", "Set.Stellen.Länge");
          end;
'STÜCK' :       vAdd  # aint(Mat.Bestand.Stk);
'GEWICHT' :     vAdd  # anum(Mat.Bestand.Gew, Set.Stellen.Gewicht);
'GEWICHT.NETTO' :   vAdd  # anum(Mat.Gewicht.Netto, Set.Stellen.Gewicht);
'GEWICHT.BRUTTO' :  vAdd  # anum(Mat.Gewicht.Brutto, Set.Stellen.Gewicht);
```

```
'AUF.BEST.DATUM' :    if (Auf.Best.Datum<>0.0.0) then vAdd # cnvad(Auf.Best.Datum);
'AUF.BEST.NUMMER' :   vAdd  # Auf.Best.Nummer;
'AUF.P.KUNDENARTNR' : vAdd  # Auf.P.KundenArtnr;
'AUF.P.GÜTE' :        vAdd  # "Auf.P.Güte";
'AUF.P.WERKSTOFFNR' : vAdd  # Auf.P.Werkstoffnr;
'NORM' :              vAdd  # MQu.NachNorm;

'AUFTRAGSNR' :        if (Mat.Auftragsnr<>0) then         vAdd # aint(Mat.Auftragsnr);
'LFS.NUMMER' :        if (Lfs.Nummer<>0) then             vAdd # aint(Lfs.Nummer);
'LFS.DATUM' :         if (Lfs.LieferDatum<>0.0.0) then    vAdd # cnvad(Lfs.LieferDatum);


// 2020-10-19 TM: Auftragspos.-Vorgaben drucken wenn vorhanden
'MINSTRECK' :         if (Auf.P.Streckgrenze1<>0.0) then    vAdd # cnvaf(Auf.P.Streckgrenze1          ,_F
                      else if (MQu.M.Von.StreckG<>0.0) then    vAdd # cnvaf(MQu.M.Von.StreckG          ,_Fmt
                      else vAdd # ' ';
'MINZUG' :            if (Auf.P.Zugfestigkeit1<>0.0) then    vAdd # cnvaf(Auf.P.Zugfestigkeit1          ,_FmtN
                      else if (MQu.M.Von.ZugFest<>0.0) then    vAdd # cnvaf(MQu.M.Von.Zugfest          ,_Fmt
                      else vAdd # ' ';
'MINDEHNUNG' :        if (Auf.P.DehnungB1<>0.0) then    vAdd # cnvaf(Auf.P.DehnungB1          ,_FmtN
                      else if (MQu.M.Von.Dehnung<>0.0) then    vAdd # cnvaf(MQu.M.Von.Dehnung          ,_F
                      else vAdd # ' ';
'MINRP02' :           if (Auf.P.DehnGrenzeA1<>0.0) then vAdd # cnvaf(Auf.P.DehnGrenzeA1        ,_FmtN
                      else if (MQu.M.Von.DehnGrenzA<>0.0) then vAdd # cnvaf(MQu.M.Von.DehnGrenzA
                      else vAdd # ' ';
'MINRP10' :           if (Auf.P.DehnGrenzeB1<>0.0) then vAdd # cnvaf(Auf.P.DehnGrenzeB1        ,_FmtN
```

```
                else if (MQu.M.Von.DehnGrenzB<>0.0) then vAdd # cnvaf(MQu.M.Von.DehnGrenzB
                else vAdd # ' ';
'MINKÖRNUNG' :      if ("Auf.P.Körnung1"<>0.0) then vAdd # cnvaf("Auf.P.Körnung1"      ,_FmtNumn
                else if ("MQu.M.Von.Körnung"<>0.0) then  vAdd # cnvaf("MQu.M.Von.Körnung"      ,_Fr
                else vAdd # ' ';
'MINHÄRTE' :        if ("Auf.P.Härte1"<>0.0) then vAdd # cnvaf("Auf.P.Härte1"      ,_FmtNumnogroup,0
                else if ("MQu.M.Von.Härte"<>0.0) then    vAdd # cnvaf("MQu.M.Von.Härte"      ,_FmtN
                else vAdd # ' ';

'MAXSTRECK' :       if (Auf.P.Streckgrenze2<>0.0) then    vAdd # cnvaf(Auf.P.Streckgrenze2      ,_F
                else if (MQu.M.Bis.StreckG<>0.0) then    vAdd # cnvaf(MQu.M.Bis.StreckG      ,_FmtN
                else vAdd # ' ';
'MAXZUG' :          if (Auf.P.Zugfestigkeit2<>0.0) then    vAdd # cnvaf(Auf.P.Zugfestigkeit2      ,_Fmt
                else if (MQu.M.Bis.ZugFest<>0.0) then    vAdd # cnvaf(MQu.M.Bis.Zugfest      ,_FmtNu
                else vAdd # ' ';
'MAXDEHNUNG' :      if (Auf.P.DehnungB2<>0.0) then    vAdd # cnvaf(Auf.P.DehnungB2      ,_Fmt
                else if (MQu.M.Bis.Dehnung<>0.0) then    vAdd # cnvaf(MQu.M.Bis.Dehnung      ,_Fm
                else vAdd # ' ';
'MAXRP02' :         if (Auf.P.DehnGrenzeA2<>0.0) then vAdd # cnvaf(Auf.P.DehnGrenzeA2      ,_FmtN
                else if (MQu.M.Bis.DehnGrenzA<>0.0) then vAdd # cnvaf(MQu.M.Bis.DehnGrenzA      ,
                else vAdd # ' ';
'MAXRP10' :         if (Auf.P.DehnGrenzeB2<>0.0) then vAdd # cnvaf(Auf.P.DehnGrenzeB2      ,_FmtN
                else if (MQu.M.Bis.DehnGrenzB<>0.0) then vAdd # cnvaf(MQu.M.Bis.DehnGrenzB      ,
                else vAdd # ' ';
'MAXKÖRNUNG' :      if ("Auf.P.Körnung2"<>0.0) then vAdd # cnvaf("Auf.P.Körnung2"      ,_FmtNum
                else if ("MQu.M.Bis.Körnung"<>0.0) then  vAdd # cnvaf("MQu.M.Bis.Körnung"      ,_Fmt
```

```
                else vAdd # ' ';
'MAXHÄRTE' :        if ("Auf.P.Härte2"<>0.0) then vAdd # cnvaf("Auf.P.Härte2"        ,_FmtNumnogroup,
                else if ("MQu.M.Bis.Härte"<>0.0) then    vAdd # cnvaf("MQu.M.Bis.Härte"          ,_FmtNur
                else vAdd # ' ';



'MINCHEMIE_C'  :    if (Auf.P.Chemie.C1<>0.0)      then vAdd # cnvaf(Auf.P.Chemie.C1        ,_FmtNu
                else if (MQu.ChemieVon.C<>0.0)  then vAdd # Cnvaf(MQu.ChemieVon.C       ,_FmtNumr
                else vAdd # ' ';
'MINCHEMIE_SI' :    if (Auf.P.Chemie.Si1<>0.0)     then vAdd # cnvaf(Auf.P.Chemie.Si1        ,_FmtNu
                else if (MQu.ChemieVon.Si<>0.0) then vAdd # Cnvaf(MQu.ChemieVon.Si      ,_FmtNumn
                else vAdd # ' ';
'MINCHEMIE_MN' :    if (Auf.P.Chemie.Mn1<>0.0)     then vAdd # cnvaf(Auf.P.Chemie.Mn1        ,_Fm
                else if (MQu.ChemieVon.Mn<>0.0) then vAdd # Cnvaf(MQu.ChemieVon.Mn      ,_FmtNum
                else vAdd # ' ';
'MINCHEMIE_P'  :    if (Auf.P.Chemie.P1<>0.0)     then vAdd # cnvaf(Auf.P.Chemie.P1        ,_FmtNum
                else if (MQu.ChemieVon.P<>0.0)      then vAdd # Cnvaf(MQu.ChemieVon.P       ,_FmtNu
                else vAdd # ' ';
'MINCHEMIE_S'  :    if (Auf.P.Chemie.S1<>0.0)     then vAdd # cnvaf(Auf.P.Chemie.S1        ,_FmtNum
                else if (MQu.ChemieVon.S<>0.0)      then vAdd # Cnvaf(MQu.ChemieVon.S       ,_FmtNu
                else vAdd # ' ';
'MINCHEMIE_AL' :    if (Auf.P.Chemie.Al1<>0.0)     then vAdd # cnvaf(Auf.P.Chemie.Al1        ,_FmtNu
                else if (MQu.ChemieVon.Al<>0.0) then     vAdd # Cnvaf(MQu.ChemieVon.AL      ,_FmtNu
                else vAdd # ' ';
'MINCHEMIE_CR' :    if (Auf.P.Chemie.Cr1<>0.0)     then vAdd # cnvaf(Auf.P.Chemie.Cr1        ,_FmtN
                else if (MQu.ChemieVon.CR<>0.0) then     vAdd # Cnvaf(MQu.ChemieVon.CR      ,_FmtN
```

```
                else vAdd # ' ';
'MINCHEMIE_V'  :    if (Auf.P.Chemie.V1<>0.0)     then vAdd # cnvaf(Auf.P.Chemie.V1        ,_FmtNum
                else if (MQu.ChemieVon.V<>0.0) then     vAdd # Cnvaf(MQu.ChemieVon.V       ,_FmtNu
                else vAdd # ' ';
'MINCHEMIE_NB' :    if (Auf.P.Chemie.Nb1<>0.0)     then vAdd # cnvaf(Auf.P.Chemie.Nb1        ,_Fmt
                else if (MQu.ChemieVon.Nb<>0.0) then     vAdd # Cnvaf(MQu.ChemieVon.NB      ,_FmtN
                else vAdd # ' ';
'MINCHEMIE_TI' :    if (Auf.P.Chemie.Ti1<>0.0)     then vAdd # cnvaf(Auf.P.Chemie.Ti1        ,_FmtNu
                else if (MQu.ChemieVon.Ti<>0.0) then     vAdd # Cnvaf(MQu.ChemieVon.TI      ,_FmtNum
                else vAdd # ' ';
'MINCHEMIE_N'  :    if (Auf.P.Chemie.N1<>0.0)     then vAdd # cnvaf(Auf.P.Chemie.N1        ,_FmtNum
                else if (MQu.ChemieVon.N<>0.0) then     vAdd # Cnvaf(MQu.ChemieVon.N       ,_FmtNu
                else vAdd # ' ';
'MINCHEMIE_CU' :    if (Auf.P.Chemie.Cu1<>0.0)     then vAdd # cnvaf(Auf.P.Chemie.Cu1        ,_Fmt
                else if (MQu.ChemieVon.Cu<>0.0) then     vAdd # Cnvaf(MQu.ChemieVon.CU      ,_FmtN
                else vAdd # ' ';
'MINCHEMIE_NI' :    if (Auf.P.Chemie.Ni1<>0.0)     then vAdd # cnvaf(Auf.P.Chemie.Ni1        ,_FmtNu
                else if (MQu.ChemieVon.Ni<>0.0) then     vAdd # Cnvaf(MQu.ChemieVon.NI      ,_FmtNum
                else vAdd # ' ';
'MINCHEMIE_MO' :    if (Auf.P.Chemie.Mo1<>0.0)     then vAdd # cnvaf(Auf.P.Chemie.Mo1        ,_Fm
                else if (MQu.ChemieVon.Mo<>0.0) then     vAdd # Cnvaf(MQu.ChemieVon.MO      ,_FmtN
                else vAdd # ' ';
'MINCHEMIE_B'  :    if (Auf.P.Chemie.Mo1<>0.0)     then vAdd # cnvaf(Auf.P.Chemie.Mo1        ,_FmtN
                else if (MQu.ChemieVon.B<>0.0) then     vAdd # Cnvaf(MQu.ChemieVon.B       ,_FmtNum
                else vAdd # ' ';
'MINCHEMIE_FREI1'  :  if (Auf.P.Chemie.Frei1.1<>0.0)     then vAdd # cnvaf(Auf.P.Chemie.Frei1.1
```

```
                else if (MQu.ChemieVon.Frei1<>0.0) then  vAdd # Cnvaf(MQu.ChemieVon.Frei1    ,_FmtN
                else vAdd # ' ';


'MAXCHEMIE_C'  :      if (Auf.P.Chemie.C2<>0.0)      then vAdd # cnvaf(Auf.P.Chemie.C2         ,_FmtNu
                else if (MQu.ChemieBis.C<>0.0) then      vAdd # Cnvaf(MQu.ChemieBis.C        ,_FmtNum
                else vAdd # ' ';
'MAXCHEMIE_SI' :      if (Auf.P.Chemie.Si2<>0.0)      then vAdd # cnvaf(Auf.P.Chemie.Si2         ,_FmtNu
                else if (MQu.ChemieBis.Si<>0.0) then     vAdd # Cnvaf(MQu.ChemieBis.Si       ,_FmtNumr
                else vAdd # ' ';
'MAXCHEMIE_MN' :      if (Auf.P.Chemie.Mn2<>0.0)      then vAdd # cnvaf(Auf.P.Chemie.Mn2         ,_Fm
                else if (MQu.ChemieBis.Mn<>0.0) then     vAdd # Cnvaf(MQu.ChemieBis.Mn       ,_FmtNu
                else vAdd # ' ';
'MAXCHEMIE_P'  :      if (Auf.P.Chemie.P2<>0.0)      then vAdd # cnvaf(Auf.P.Chemie.P2         ,_FmtNu
                else if (MQu.ChemieBis.P<>0.0) then      vAdd # Cnvaf(MQu.ChemieBis.P        ,_FmtNumr
                else vAdd # ' ';
'MAXCHEMIE_S'  :      if (Auf.P.Chemie.S2<>0.0)      then vAdd # cnvaf(Auf.P.Chemie.S2         ,_FmtNu
                else if (MQu.ChemieBis.S<>0.0) then      vAdd # Cnvaf(MQu.ChemieBis.S        ,_FmtNumr
                else vAdd # ' ';
'MAXCHEMIE_AL' :      if (Auf.P.Chemie.Al2<>0.0)      then vAdd # cnvaf(Auf.P.Chemie.Al2         ,_FmtN
                else if (MQu.ChemieBis.Al<>0.0) then     vAdd # Cnvaf(MQu.ChemieBis.AL       ,_FmtNum
                else vAdd # ' ';
'MAXCHEMIE_CR' :      if (Auf.P.Chemie.Cr2<>0.0)      then vAdd # cnvaf(Auf.P.Chemie.Cr2         ,_Fmt
                else if (MQu.ChemieBis.CR<>0.0) then     vAdd # Cnvaf(MQu.ChemieBis.CR       ,_FmtNu
                else vAdd # ' ';
'MAXCHEMIE_V'  :      if (Auf.P.Chemie.V2<>0.0)      then vAdd # cnvaf(Auf.P.Chemie.V2         ,_FmtNu
                else if (MQu.ChemieBis.V<>0.0) then      vAdd # Cnvaf(MQu.ChemieBis.V        ,_FmtNumr
```

```
                else vAdd # ' ';
'MAXCHEMIE_NB' :    if (Auf.P.Chemie.Nb2<>0.0)    then vAdd # cnvaf(Auf.P.Chemie.Nb2       ,_Fm
                else if (MQu.ChemieBis.Nb<>0.0) then    vAdd # Cnvaf(MQu.ChemieBis.NB      ,_FmtNu
                else vAdd # ' ';
'MAXCHEMIE_TI' :    if (Auf.P.Chemie.Ti2<>0.0)    then vAdd # cnvaf(Auf.P.Chemie.Ti2       ,_FmtNu
                else if (MQu.ChemieBis.Ti<>0.0) then    vAdd # Cnvaf(MQu.ChemieBis.TI      ,_FmtNumr
                else vAdd # ' ';
'MAXCHEMIE_N'  :    if (Auf.P.Chemie.N2<>0.0)    then vAdd # cnvaf(Auf.P.Chemie.N2       ,_FmtNu
                else if (MQu.ChemieBis.N<>0.0) then    vAdd # Cnvaf(MQu.ChemieBis.N       ,_FmtNum
                else vAdd # ' ';
'MAXCHEMIE_CU' :    if (Auf.P.Chemie.Cu2<>0.0)    then vAdd # cnvaf(Auf.P.Chemie.Cu2       ,_Fm
                else if (MQu.ChemieBis.Cu<>0.0) then    vAdd # Cnvaf(MQu.ChemieBis.CU      ,_FmtNu
                else vAdd # ' ';
'MAXCHEMIE_NI' :    if (Auf.P.Chemie.Ni2<>0.0)    then vAdd # cnvaf(Auf.P.Chemie.Ni2       ,_FmtN
                else if (MQu.ChemieBis.Ni<>0.0) then    vAdd # Cnvaf(MQu.ChemieBis.NI      ,_FmtNum
                else vAdd # ' ';
'MAXCHEMIE_MO' :    if (Auf.P.Chemie.Mo2<>0.0)    then vAdd # cnvaf(Auf.P.Chemie.Mo2       ,_Fr
                else if (MQu.ChemieBis.Mo<>0.0) then    vAdd # Cnvaf(MQu.ChemieBis.MO      ,_FmtNu
                else vAdd # ' ';
'MAXCHEMIE_B'  :    if (Auf.P.Chemie.B2<>0.0)    then vAdd # cnvaf(Auf.P.Chemie.B2       ,_FmtNu
                else if (MQu.ChemieBis.B<>0.0) then    vAdd # Cnvaf(MQu.ChemieBis.B       ,_FmtNumr
                else vAdd # ' ';
'MAXCHEMIE_FREI1'  :  if (Auf.P.Chemie.Frei1.2<>0.0)    then vAdd # cnvaf(Auf.P.Chemie.Frei1.2
                else if (MQu.ChemieBis.Frei1<>0.0) then  vAdd # Cnvaf(MQu.ChemieBis.Frei1    ,_FmtNu
                else vAdd # ' ';
```

```
'RAU_OS' :        if (Mat.RauigkeitA1<>0.0) then vAdd # anum(Mat.RauigkeitA1, 2);
'RAU_US' :        if (Mat.RauigkeitC2<>0.0) then vAdd # aNum(Mat.RauigkeitC1, 2);
'MECH_SONST' :    vAdd # Mat.Mech.Sonstiges1;

'STRECK_ODERFEHLER' :       vAdd # AnalyseBereich(Mat.Streckgrenze1,    MQu.M.Von.StreckG, MQ
'ZUG_ODERFEHLER' :          vAdd # AnalyseBereich(Mat.Zugfestigkeit1,   MQu.M.Von.ZugFest, MQu.
'DEHNUNG_ODERFEHLER' :      vAdd # AnalyseBereich(Mat.DehnungA1,        MQu.M.Von.Dehnung,
'RP02_ODERFEHLER' :         vAdd # AnalyseBereich(Mat.RP02_V1,          MQu.M.Von.DehnGrenzA, M
'RP10_ODERFEHLER' :         vAdd # AnalyseBereich(Mat.RP10_V1,          MQu.M.Von.DehnGrenzB, M
'KÖRNUNG_ODERFEHLER' :      vAdd # AnalyseBereich("Mat.Körnung1",       "MQu.M.Von.Körnung",
'HÄRTE_ODERFEHLER' :        vAdd # AnalyseBereich("Mat.HärteA1",        "MQu.M.Von.Härte", MQu.M
'CHEMIE_C_ODERFEHLER'  :    vAdd # AnalyseBereich(Mat.Chemie.C1,      MQu.ChemieVon.C     ,M
'CHEMIE_SI_ODERFEHLER' :    vAdd # AnalyseBereich(Mat.Chemie.Si1,     MQu.ChemieVon.Si    ,MQ
'CHEMIE_MN_ODERFEHLER' :    vAdd # AnalyseBereich(Mat.Chemie.Mn1,     MQu.ChemieVon.Mn
'CHEMIE_P_ODERFEHLER'  :    vAdd # AnalyseBereich(Mat.Chemie.P1,      MQu.ChemieVon.P     ,MQ
'CHEMIE_S_ODERFEHLER'  :    vAdd # AnalyseBereich(Mat.Chemie.S1,      MQu.ChemieVon.S     ,MQ
'CHEMIE_AL_ODERFEHLER' :    vAdd # AnalyseBereich(Mat.Chemie.AL1,     MQu.ChemieVon.Al    ,M
'CHEMIE_CR_ODERFEHLER' :    vAdd # AnalyseBereich(Mat.Chemie.CR1,     MQu.ChemieVon.Cr    ,
'CHEMIE_V_ODERFEHLER'  :    vAdd # AnalyseBereich(Mat.Chemie.V1,      MQu.ChemieVon.V     ,MQ
'CHEMIE_NB_ODERFEHLER' :    vAdd # AnalyseBereich(Mat.Chemie.NB1,     MQu.ChemieVon.Nb
'CHEMIE_TI_ODERFEHLER' :    vAdd # AnalyseBereich(Mat.Chemie.TI1,     MQu.ChemieVon.Ti    ,MQ
'CHEMIE_N_ODERFEHLER'  :    vAdd # AnalyseBereich(Mat.Chemie.N1,      MQu.ChemieVon.N     ,M
'CHEMIE_CU_ODERFEHLER' :    vAdd # AnalyseBereich(Mat.Chemie.CU1,     MQu.ChemieVon.Cu
'CHEMIE_NI_ODERFEHLER' :    vAdd # AnalyseBereich(Mat.Chemie.NI1,     MQu.ChemieVon.Ni    ,MQ
'CHEMIE_MO_ODERFEHLER' :    vAdd # AnalyseBereich(Mat.Chemie.MO1,     MQu.ChemieVon.Mo
'CHEMIE_B_ODERFEHLER'  :    vAdd # AnalyseBereich(Mat.Chemie.B1,      MQu.ChemieVon.B     ,MQ
```

```
'CHEMIE_FREI1_ODERFEHLER'  : vAdd # AnalyseBereich(Mat.Chemie.Frei1.1, MQU.ChemieVon.Frei

'STRECK' :        vAdd # Analyse(Mat.Streckgrenze1, 2);
'ZUG' :           vAdd # Analyse(Mat.Zugfestigkeit1, 2);
'DEHNUNG' :       vAdd # Analyse(Mat.DehnungA1, 2);
'DEHNUNGB' :      vAdd # Analyse(Mat.DehnungB1, 2);
'RP02' :          vAdd # Analyse(Mat.RP02_V1, 2);
'RP10' :          vAdd # Analyse(Mat.RP10_V1, 2);
'KÖRNUNG' :       vAdd # Analyse("Mat.Körnung1", 2);
'KÖRNUNG2' :      vAdd # Analyse("Mat.Körnung2", 2);
'HÄRTE' :         vAdd # Analyse("Mat.HärteA1", 2);
'CHEMIE_C'  :     vAdd # Analyse(Mat.Chemie.C1,5);
'CHEMIE_SI' :     vAdd # Analyse(Mat.Chemie.Si1,5);
'CHEMIE_MN' :     vAdd # Analyse(Mat.Chemie.Mn1,5);
'CHEMIE_P'  :     vAdd # Analyse(Mat.Chemie.P1, 5);
'CHEMIE_S'  :     vAdd # Analyse(Mat.Chemie.S1, 5);
'CHEMIE_AL' :     vAdd # Analyse(Mat.Chemie.AL1,5);
'CHEMIE_CR' :     vAdd # Analyse(Mat.Chemie.CR1,5);
'CHEMIE_V'  :     vAdd # Analyse(Mat.Chemie.V1, 5);
'CHEMIE_NB' :     vAdd # Analyse(Mat.Chemie.NB1,5);
'CHEMIE_TI' :     vAdd # Analyse(Mat.Chemie.TI1,5);
'CHEMIE_N'  :     vAdd # Analyse(Mat.Chemie.N1, 5);
'CHEMIE_CU' :     vAdd # Analyse(Mat.Chemie.CU1,5);
'CHEMIE_NI' :     vAdd # Analyse(Mat.Chemie.NI1,5);
'CHEMIE_MO' :     vAdd # Analyse(Mat.Chemie.MO1,5);
'CHEMIE_B'  :     vAdd # Analyse(Mat.Chemie.B1, 5);
```

```
    'CHEMIE_FREI1'  : vAdd # Analyse(Mat.Chemie.Frei1.1, 5);
    otherwise           begin
                        // Allgemeiner Befehl?
                        if (ParseAllgemein(var aLabels, var aInhalt, var aZusatz, vFeld, vTitel, vPost, vPre, aKomb

                        // unbekannt?
                        inc(vZeilen);
                        AddLIZ(var aLabels, var aInhalt, var aZusatz, '?'+vTitel, '?'+vFeld, vPre, vPost, aKombi);
                       end;
  end;

  if (vAdd<>'') then begin
    inc(vZeilen);
    AddLIZ(var aLabels, var aInhalt, var aZusatz, vTitel, vAdd, vPre, vPost, aKombi);
  end;

  RETURN vZeilen;
end;



//=========================================================================
//  Parse200Multi
//=========================================================================
sub Parse200Multi(
  var aLabels : alpha;
  var aInhalt : alpha;
```

```
  var aZusatz : alpha;
  aObjName    : alpha;
  aKombi      : logic;
  ) : int
local begin
  vCap        : alpha(4096);
  vRow        : alpha(4096);
  vToken      : alpha(4096);
  vA          : alpha(4096);
  vPre,vPost  : alpha(4096);
  vI,vJ       : int;
  vK,vL       : int;
  vOK         : logic;
  vZ          : int;
  vZeilen     : int;
  vTmp,vMax   : int;
end;
begin

  aLabels # '';
  aInhalt # '';
  aZusatz # '';

  vCap # GetCaption(aObjName);

  vJ # 1 + Lib_Strings:Strings_Count(vCap, StrChar(13)+StrChar(10));
```

```
FOR vI # 1 loop inc(vI) WHILE (vI<=vJ) do begin
  vRow # Str_Token(vCap, StrChar(13)+StrChar(10), vI);
  vOK # n;
  vL # Lib_Strings:Strings_Count(vRow, '@');

  vZ # 0;
  if (vL=0) then begin//and (vRow<>'') then begin
     AddLIZ(var aLabels, var aInhalt, var aZusatz, '', vRow, ',', aKombi);
    inc(vZ);
    end
  else begin
    vMax # 0;
    FOR vK # 1 loop inc(vK) WHILE (vK<=vL) do begin
      vToken # Str_Token(vRow, '@', vK+1);
      vTmp # Parse200(var aLabels, var aInhalt, var aZusatz, vToken, aKombi);
      if (vTmp<0) then RETURN vTmp;
      vMax # Max(vTmp, vMax);
    END;
    vZ # vZ + vMax;
  end;
  if (vZ>0) then begin
    aInhalt # aInhalt + StrChar(10);
    aLabels # aLabels + StrChar(10);
    aZusatz # aZusatz + StrChar(10);
    vZeilen # vZeilen + vZ;
  end;
```

```
  END;

  RETURN vZeilen;

end;

//=============================================================================

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