

Oktober 2009

CAP-Suite, CAP, infor R6.1, R6.3 und R7.1

Verschnittoptimierung

(nur CAP Oberfläche ab 08.2009)

Im CAP ist jetzt eine Zuschnitt Optimierung implementiert.

Im Normblatt für die Merkmalerfassung kann in der Spalte M (=Merkmal) eine Option „O“ ausgewählt werden:



Die Optimierung wird durch folgende **Systemvariablen** gesteuert.

| | |
|------------------|---|
| \$O_SchnittBrt | Schnittbreite in mm. |
| \$O_SchnittArt | "G" (Guillotine) oder "N" (nested). |
| \$O_LgTeilLänge | Tafellänge in mm. |
| \$O_LgTeilBreite | Tafelbreite in mm. |
| \$O_TeilAnzahl | Anzahlelemente als Stück. |
| \$O_TeilBreite | Elementbreite in mm. |
| \$O_TeilLänge | Elementlänge in mm. |
| \$O_TeilDrehbar | "N" (nein) oder "D" (drehbar) |
| \$O_BmpName | Pfad und Name der BMP-Datei die automatisch erzeugt wird. |

Die Berechnung erfolgt in diesem Beispiel nach jeder Wertänderung.

```
* BL_CUT_001: Zuschnitt Optimierung, Systemvariablen belegen,
* die Berechnung und Zeichnen der Zuschnitt Optimierung erfolgt automatisch
* -----
* TEST: "Ein".

$O_SchnittBrt = Schnittbreite.
$O_SchnittArt = Schnitt.

$O_LgTeilLänge = Tafellänge.
$O_LgTeilBreite = TafelBreite.

$O_TeilAnzahl = AnzahlElemente.
$O_TeilBreite = Elementbreite.
$O_TeilLänge = Elementlänge.

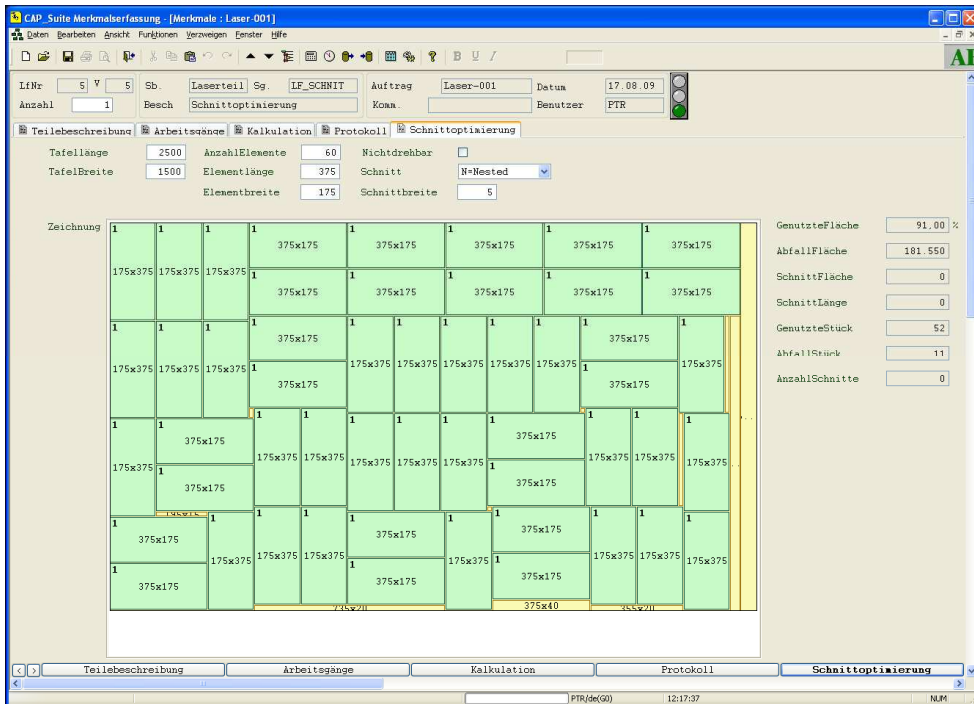
$O_BmpName = ".\\Cut\\" + $ANr + ".BMP".

WENN: Nichtdrehbar == "1".
  DANN: $O_TeilDrehbar = "N".
  SONST: $O_TeilDrehbar = "D".
ENDE.
```

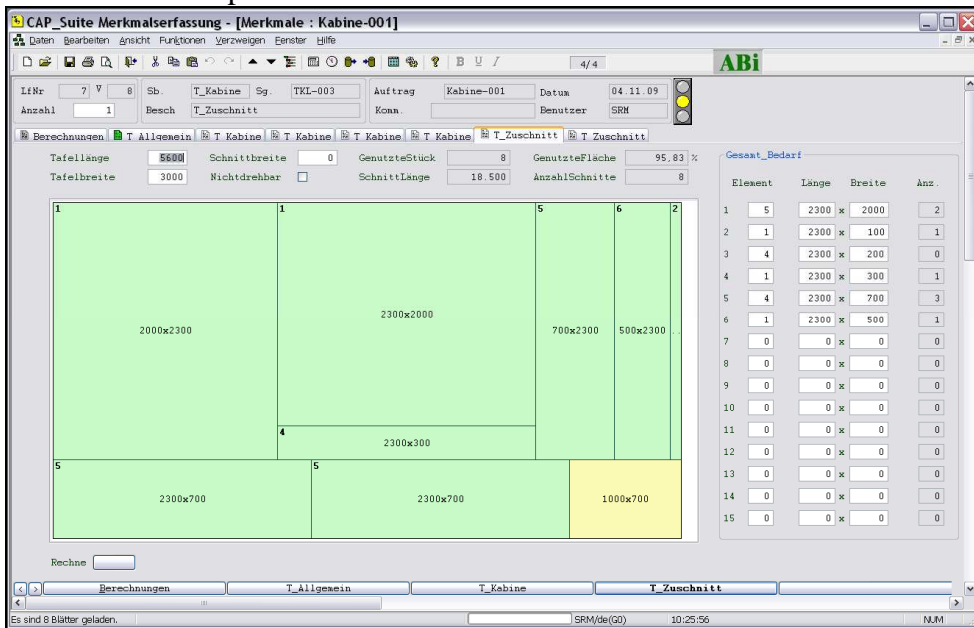
```
* BL_CUT_002: Zuschnitt Optimierung
* Werte aus der Systemvariablen in die Maske übertragen
* -----
* TEST: "Ein".

GenutzteFläche = $O_BelegteFläche.
AbfallFläche = $O_AbfallFläche.
SchnittFläche = $O_SchnittFläche.
SchnittLänge = $O_SchnittLänge.
GenutzteStück = $O_AnzahlVerwend.
AbfallStück = $O_AnzahlAbfall.
AnzahlSchnitte = $O_AnzahlSchnitte.
```

Ergebnis:



Ein weiteres Beispiel:



Systemvariablen:

(CAP Oberfläche ab 08.2009)

Für die Verschnittoptimierung gibt es 14 neue Systemvariablen die alle mit: \$O_??? anfangen.

TXT_RTF:

RTF_TXT:

(ab 04.2009)

Befehle sind nicht mehr notwendig, da Befehle RTF_RTF: und TXT_TXT: diese Funktionalität ebenfalls bieten.

RTF_RTF:

TXT_TXT:

(ab 04.2009)

Beide Befehle fügen Dateien/Texte zusammen. Lediglich das Ausgabeformat ist unterschiedlich. Durch die Syntaxerweiterung werden die Befehle RTF_TXT: und TXT_RTF: nicht mehr benötigt.

RTF_RTF: „Quelle1“ [(„PfadQuelle1“) „Quelle2“ [(„PfadQuelle2“) „Ziel.RFT“ („PfadZiel“) [„-“].

„Quelle1“ Name der 1. Datei.
 [(„PfadQuelle1“) Pfad, wo sich diese Datei befindet.
 „Quelle2“ Name der 2. Datei, kann weglassen werden.
 [(„PfadQuelle2“) Pfad, wo sich die 2. Quelldatei befindet. Parameter kann weglassen werden, dann wird ggf. „PfadQuelle1“ angenommen.
 „Ziel.RFT“ Name der Zieldatei. Die Dateierweiterung bestimmt das Zielformat: „.RTF“ oder „.TXT“ beide Quelldateien werden in das Zielformat konvertiert.
 Bei Datei1.RTF wird die Formatierung an die Datei2.TXT weitergegeben.
 Bei Datei1.TXT wird die Windows Formatierung angenommen.
 Bei Datei2.RTF wird die Formatierung beibehalten.
 Bei Datei2.TXT wird die letzte RTF Formatierung übernommen.
 Quellen und Ziel können auch TEXT: Bereich sein: „{TEXT:}“.
 [(„PfadZiel“) Pfad der Zieldatei, Parameter kann weglassen werden, dann wird „PfadQuelle1“ angenommen.
 [„-“] unterdrückt die Ausgabe einer Leerzeile am Ende des Textes.

```
PfadQuelle = "RTF/RTF_QUELLE/" .
PfadZiel   = "RTF/RTF_ZIEL/" .
Quelle1    = "Quelle1.RTF" .
Quelle2    = "Quelle2.RTF" .
Quelle3    = "Quelle3.TXT" .
Ziel1     = "Ziel1.RTF" .
Ziel2     = "Ziel2.RTF" .
Ziel3     = "Ziel3.RTF" .
Ziel4     = "Ziel4.RTF" .
Ziel5     = "Ziel5.RTF" .
Variable  = $Heute.
```

```
* kopiert Inhalt von Quelle1 und Quelle2 in Ziel1 und übersetzt dabei.
RTF_RTF: Quelle1 (PfadQuelle) Quelle2 Ziel1 (PfadZiel).
```

```
* kopiert Inhalt von Quelle1 und Quelle2 in TEXT: mit Übersetzung
* Speicherbereich von TEXT: wird anschließend in Ziel2 exportiert.
RTF_RTF: Quelle1 (PfadQuelle) Quelle2 "{TEXT:}" .
```

```
EXPORT: Ziel2 (PfadZiel). TRANSFER: Ziel2. BEENDEN: Ziel2.

*Text im Speicher vorbereiten
TEXT:      "{\\rtf1\\ansi\\deff0\\deftab720{\\fonttbl.
TEXT: + "{\\f0\\fmodern\\fprq1\\fcharset238 Courier New\\;}}".
TEXT: + "{\\colortbl\\red0\\green0\\blue0\\;}" + "\\plain\\f0\\fs20\\cf0".
TEXT: + "----" + " " + "Ressourcenliste \"0815\"angelegt von ".
TEXT: + $Login_User + " am <CAP:F>Variable</CAP:F> ---\\par}".

* RTF direkt in die Datei oder
RTF_RTF: Quelle1 (PfadQuelle) "{TEXT:}" Ziel3 (PfadZiel).

* in TEXT: schreiben und dann exportieren.
EXPORT:   Ziel4 (PfadZiel).
RTF_RTF:  Quelle1 (PfadQuelle) "{TEXT:}" "{TEXT:}".
TRANSFER: Ziel4.
BEENDEN:  Ziel4.

* RTF + TXT zusammenfassen in eine RTF Datei.
RTF_RTF:  Quelle1 (PfadQuelle) Quelle3 Ziel5 (PfadZiel).
```

QUELLEN:

Quelle1.RTF

Test mit **RTF1** <CAP:F>Variable</CAP:F> Text.

Quelle2.RTF

Test mit **RTF2** <CAP:F>Variable</CAP:F> Text.

Quelle3.TXT

Test mit TXT1 <CAP:F>Variable</CAP:F> Text.

ERGEBNISSE:

Ziel11.RTF

Test mit **RTF1** 16.03.2009 Text.Test mit **RTF2** 16.03.2009 Text.

Ziel12.RTF

Test mit **RTF1** 16.03.2009 Text.Test mit **RTF2** 16.03.2009 Text.

Ziel13.RTF

Test mit **RTF1** 16.03.2009 Text.--- Ressourcenliste "0815"angelegt von PTR am 16.03.2009 ---

Ziel14.RTF

Test mit **RTF1** 16.03.2009 Text.--- Ressourcenliste "0815"angelegt von PTR am 16.03.2009 ---

Ziel15.RTF

Test mit **RTF1** 16.03.2009 Text.Test mit TXT1 16.03.2009 Text.

Lizenzierung in R6.1:

(ab 04.2009)

Problem: wenn keine SML lizenziert ist, kann infor R6.1 nicht "lizenziert werden.

Lj4: **mnuCAPELTest.Click** ruft den Lj4 **bfrmMG.GlbEL.Plausi** und dort wird die E-Logik mit dem Parameter 01 = „SML“ statt 16 = „E-Logik testen“ aufgerufen.

Solche Aufrufe gibt es mehrmals: siehe IDM-Verwendung von bfrmMG.GlbEL.Plausi

```

Sub mnuCAPELTest.Click()
/*-----
@BeginComment
mnuCAPELTest.Click

@F(Syntax)
Sub mnuCAPELTest.Click (
)

@F(Beschreibung)
Start test of CAP decisionlogic

@F(Parameter)

@F(Vorbedingung)
keine

@F(Nachbedingung)
keine

@EndComment
-----*/
/*
25 June 1999 RIESTERER (i)
03.07.2000 Riesterer ask for saving
07.08.2000 Riesterer set current dummy-vt to ZwiSp
2008-08-06 Petrasch EL-Aufruf geändert, Dummies + Abfrage gelöscht
*/
    Dim sELName As tString
    Dim sMsg As tString = ""
    Dim sSaveMsg As tString = ""

    sELName = FormThis.chdHead.txtName.Value

    If CAPEL.al_HasChanged() Then
//      Lja.SPY10(sSaveMsg, "steCAP_Save_Elogik", "ELogik", sELName)
//      If Root.VMsgBox(cMsgQuest, sSaveMsg) = IDYES Then
//        CAPEL.al_Save()
//      End If
    End If

Lja.MGE00()
If Lja.MGE16(sELName, "", sMsg) = 101 Then // 10->16 = E-Logik testen
  If sMsg <> "" Then
    Root.VMsgBox(cMsgStop, sMsg)
  End If
End If
Lja.MGE99()

End Sub // mnuCAPELTest.Click

```

EL-Editor:

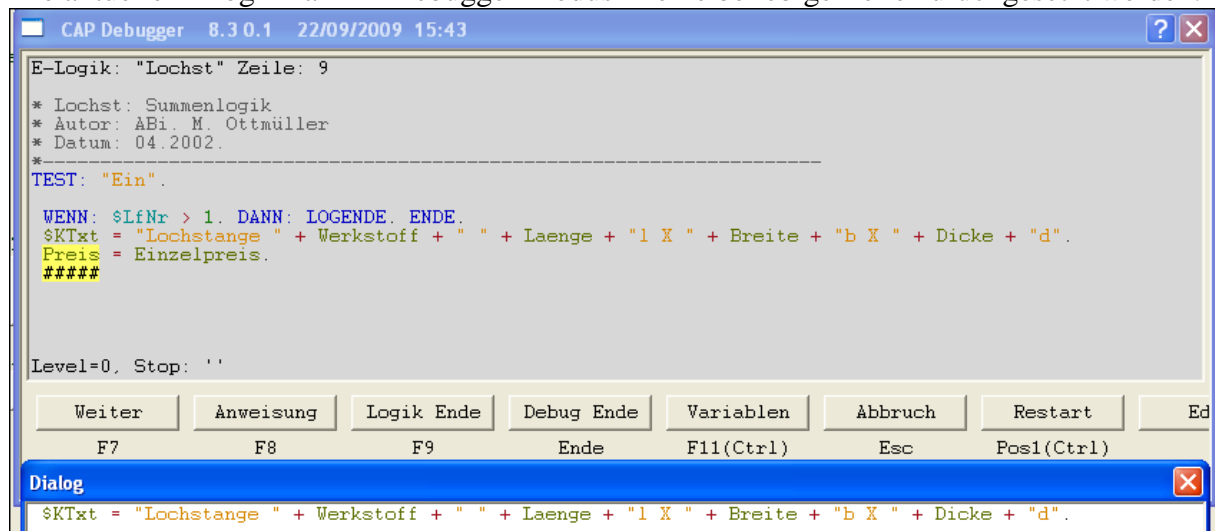
(CAP Oberfläche ab 08.2009)

Die aktuelle E-Logik kann in Debuggermodus getestet werden auch ohne dass TEST: „Ein“ eingetragen wird. Taste „Strg“ drücken und dann auf das Icon „Lupe“ klicken.

Debugger:

(08.2009)

Die aktuelle E-Logik kann in Debugger Modus in eine beliebige Zeile zurückgesetzt werden.



Dazu wird mit der Tastenkombination **Strg + Pos1** ein Fenster geöffnet, in dem auf die gewünschte Zeile positioniert werden kann. Nach Return steht der Debugger in der ausgewählten Zeile.

- Bei der Positionierung oberhalb der aktuellen Zeile werden die zwischenzeitlich geänderten Variablenwerte, Vorgänge... nicht zurückgesetzt.
- Bei der Positionierung nach der aktuellen Zeile werden die dazwischenliegenden Anweisungen nicht ausgeführt.

SML-Suche:

(CAP Oberfläche ab 10.2009)

Normalerweise werden bei der SML-Suche alle (in Gültigkeit des Normblattes „Lo-Flachmat“ hinterlegte) Werte angeboten. Dis entspricht aber nicht den im System hinterlegten SML-Ausprägungen.

Um bei der SML-Suche nur die tatsächlich vorhandene Werte anzuzeigen anzubieten kann ein zweites Normblatt erstellt werden. Dieser bekommt die gleiche Struktur wie das erste, aber in der Gültigkeit werden E-Logiken eingetragen die, die Auswahl entsprechend den in System gefundenen Treffern, einschränken.

Suche über SML

Normblatt Beschreibung

von _____ bis _____

Werkstoff

Dicke mm

Breite mm

O.K.
Zurück
Loeschen
Hilfe

Name Benutzer Datum

Beschr.

| Merkmal | Position | KBez | Len | Dez | M | Einh. | Gültigkeit |
|---------|-------------|---------|-----|-----|---|-------|--------------------------|
| 1 | Werkstoff | WS | 2 | | S | | :AL;MS;ST;* |
| 2 | Dicke | D | 2 | 0 | S | mm | :5;6;7;8;9;10;* |
| 3 | Breite | B | 2 | 0 | S | mm | :20;30;40;50;* |
| 4 | Matkosten | MatKost | 8 | 2 | M | Euro | |
| 5 | Pe | Pe | 3 | | M | | :Stk;kg;m;m ² |
| 6 | Bezeichnung | @Bez | 40 | | K | | |

Name Benutzer Datum

Beschr.

| Merkmal | Position | KBez | Len | Dez | M | Einh. | Gültigkeit |
|---------|-------------|---------|-----|-----|---|-------|--------------------------|
| 1 | Werkstoff | WS | 2 | | S | | ELK:Lo-Flachmat21 |
| 2 | Dicke | D | 2 | 0 | S | mm | ELK:Lo-Flachmat22 |
| 3 | Breite | B | 2 | 0 | S | mm | ELK:Lo-Flachmat23 |
| 4 | Matkosten | MatKost | 8 | 2 | M | Euro | |
| 5 | Pe | Pe | 3 | | M | | :Stk;kg;m;m ² |
| 6 | Bezeichnung | @Bez | 40 | | K | | |

```
* Lo-Flachmat21:
$Sb      = "Lo-Halbz".
Werkstoff = "?".
SELSML:  "Lo-Flachmaterial".
```

```
* Lo-Flachmat22:
$Sb      = "Lo-Halbz".
Dicke    = "?".
SELSML:  "Lo-Flachmaterial".
```

```
* Lo-Flachmat23:
$Sb      = "Lo-Halbz".
Breite   = "?".
SELSML:  "Lo-Flachmaterial".
```

Tabellen oder Tabellenzeilen automatisch erstellen

(CAP/VARIANTS ab 08.2009)

Aus E-Logiken können neue Tabellen oder Tabellenzeilen erzeugt werden, Alle Merkmale/Spalten der Tabelle müssen als Variablen vorhanden sein.

```
Durchmesser = 50.
Maschine    = "TEST".
Preis2      = 31,55.
TabellenName = "Lo_Tab2".

TABSUCH: TabellenName.
WENN: $Fehler = "F". DANN:
  LESEN: TabellenName "TAB" (). */ Vorlage lesen
  DUP:   TabellenName "TAB" (). */ Tabellenzeile erstellen
ENDE.

LESEN: TabellenName "TAB" (). */ Vorlage lesen
DUP:   "TabelleNEU" "TAB" (). */ neue Tabelle erstellen mit 1. Zeile
```

Tabellendefinition:

| Merkm. | Position | Merkm. | KBez | Len | Dez | M | Einh. |
|--------|----------|-------------|-------|-----|-----|---|-------|
| 1 | | Durchmesser | DN | 2 | 0 | B | |
| 2 | | Maschine | Masch | 6 | | M | |
| 3 | | Preis2 | P2 | 5 | 2 | M | EUR |

Tabelleninhalt vor der E-Logik:

| | | | |
|--------|------------------|-----------|---------|
| Name | Lo_Tab2 | Normblatt | Lo_Tab2 |
| Beschr | Preise Maschinen | | |
| | DN | Masch | P2 |
| 1 | 10 | 8610 | 19,00 |
| 2 | 30 | 8620 | 22,00 |
| 3 | 40 | 8630 | 24,00 |
| 4 | | | |

Tabelleninhalt nach der E-Logik:

| | | | |
|--------|------------------|-----------|---------|
| Name | Lo_Tab2 | Normblatt | Lo_Tab2 |
| Beschr | Preise Maschinen | | |
| | DN | Masch | P2 |
| 1 | 10 | 8610 | 19,00 |
| 2 | 30 | 8620 | 22,00 |
| 3 | 40 | 8630 | 24,00 |
| 4 | 50 | TEST | 31,55 |

Neue Tabelle:

| | | | |
|--------|------------------|-----------|---------|
| Name | TabelleNEU | Normblatt | Lo_Tab2 |
| Beschr | Preise Maschinen | | |
| | DN Masch | | P2 |
| 1 | 50 TEST | | 31,55 |

MERKMAL:

(CAP/VARIANTS ab 10.2009)

MERKMAL: Controlname ("A").

Option „A“ markiert die ControlGrafik mit einem Rahmen.

MERKMAL:

(CAP Oberfläche ab 10.2009)

MERKMAL: Feldname ("X").

Option „X“ setzt Fehlerflag (Grafik „X“ oder „!“ und die interne Markierung) des Feldes zurück. Dies bewirkt dass das Feld von nachfolgendem Ablauf als korrekt behandelt wird. Stand die Ampel auf rot, wird sie auf grün umgeschaltet, es sei denn ein anderes Feld ist noch als Fehlerhaft markiert.

Solid Works

(CAP/VARIANTS ab 10.2009)

Aus E-Logiken können Solid Works Dokumente und Makros gesteuert werden. Um die Befehle anzuwenden muss Solid Works zuerst gestartet werden.

SLW:

(CAP/VARIANTS ab 10.2009)

SLW: "Parameter".

Initialisiert die Solid Works Sitzung und prüft welche Datei offen ist.

- "Dateiname". Ist diese Datei in SolidWorks noch nicht geöffnet, wird sie gestartet und die Systemvariable \$Fehler mit „“, sonst \$Fehler = "F" belegt..
- "" prüft ob Solid Works gestartet ist. Liefert \$Fehler = „“ oder \$Fehler = "F".
- "?" prüft ob Solid Works gestartet ist und liefert ggf. Name der Datei: \$Fehler = „Dateiname“ oder \$Fehler = "F".

SLWENDE:

(CAP/VARIANTS ab 10.2009)

SLWENDE: .

Schließt das Programm Solid Works.
Dieser Befehl sollte nur in Ausnahmefällen verwendet werden!

SLWDOKENDE:

(CAP/VARIANTS ab 10.2009)

SLWDOKENDE: .

Schließt das Dokument in dem Solid Works.

SLWMAKRO:

(CAP/VARIANTS ab 10.2009)

SLWMAKRO: "Dateiname" ("MakroName" "ProzedurName").

Die Makros sind in Dateien gespeichert und können mehrere Prozeduren beinhalten

- "Dateiname" ist Pfad und Dateiname in dem das Makro gespeichert ist.
- "MakroName" ist Name des Makros selbst.
- "ProzedurName" ist Name der Prozedur in diesem Makro.

SLWSCHREIBEN:

(CAP/VARIANTS ab 10.2009)

SLWSCHREIBEN: (Merkmal1 [Merkmal2 Merkmal3...]).

Beschreibt die Variablen „custom property“, die von Makro gelesen werden können.

SLWLESEN:

(CAP/VARIANTS ab 10.2009)

SLWLESEN: (Merkmal1 [Merkmal2 Merkmal3...]).

Liest die Variablen „custom property“, die von Makro belegt wurden.

SLWSPARAM:

(CAP/VARIANTS ab 10.2009)

SLWSPARAM: (DlqLinear."Dl@linear austragung1" D1@Skizze1 [...]).

Ändert die Abmessungen von Solid Works Objekten direkt über ihre Namen (ggf. mit Alias Namen).

SLWLPARAM:

(CAP/VARIANTS ab 10.2009)

SLWLPARAM: (DlqLinear."Dl@linear austragung1" D1@Skizze1 [...]).

Liest die Abmessungen von Solid Works Objekten direkt über ihre Namen (ggf. mit Alias Namen).