

Dezember 2009

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

Debugger

(CAP/VARIANTS ab 11.2009)

Debugger erscheint nach Rückkehr aus einem Unterprogramm erneut.

Per Tastenkombination **Strg + F9** (lokales **Log Ende**) wird die aktuelle E-Logik mit allen Ihren Unterlogiken beendet.

Reorganisation der Merkmale nach Normblattänderung

(Nur CAP-Oberfläche ohne infor ab 11.2009)

Wenn ein Normblattes mit aktiver Verwendung in der Merkmalerfassung geändert wird, werden die dazugehörigen Merkmale reorganisiert.

ACHTUNG: mehrere Änderungen gleichzeitig können zum Datenverlust führen!

Beim Einzelimport von Normblatt über die XML-Schnittstelle, wird das Normblatt zuerst angezeigt. Beim Speichern wird dann die Notwendigkeit der Reorganisation geprüft.

Vor der Reorganisation wird folgendes geprüft:

1. Anzahl Merkmale
 2. Änderung in Feldlängen oder Dezimalstellen
 3. Merkmalnamen (nur bei Feldlänge > 0)
 4. Prüfung ob Spalten gelöscht werden müssten.
 5. Prüfung ob betreffende Daten noch lokale Verwendung haben.
- Ggf. wird eine Frage gestellt ob das so richtig ist.

SELEKT: SELSML:

(CAP/VARIANTS ab 12.2002)

So wie es bereits beim Befehl SELEKT: beschrieben ist, funktioniert es auch in der SML-Suche Das Fragezeichen ("?") unterbricht die schlüsselkette, das Ausrufezeichen ("!") nicht.

Tabelle oder SML:

Schlüssel1	Schlüssel2	Schlüssel3.
X	A	B
X	Y	Z

E-Logik:

```
Schlüssel1 = "X".
Schlüssel3 = "B".
Schlüssel2 = "?".
SELEKT: "Tabelle".  */findet beide Zeilen, da Schlüssel 3 ignoriert wird.
$Sel_1 =? "A"
$Sel_2 =? "Y"

Schlüssel1 = "X".
Schlüssel3 = "B".
Schlüssel2 = "!".
```

```
SELEKT: "Tabelle". */findet nur die 1. Zeile
$Sel_1 =? "A"
```

EDPBOOLMODE:

(CAP in ABAS ab 10.2009)

EDPBOOLMODE: "Text".

Befehl EDP: liefert „Ja“ oder „Nein“ bzw. „True“ oder „False“, je nach der Sprache.

EDPBOOLMODE: "Num".

Befehl EDP: liefert 1 oder 0.

Normblatt

(CAP-Oberfläche ab 10.2009)

In der Spalte „Hinweistext“ werden die Tooltips eingetragen. Wenn dieser Text mit */ eingeleitet wird, handelt sich um einen Kommentar der nicht angezeigt wird.

ZA_KOPIEREN:

(ab 10.2009)

Inhalt (des TEXT: Bereiches) in der (Windows) Zwischenablage kopieren.

ZA_KOPIEREN: Variable | "{TEXT}" | "Text".

ZA_EINFUEGEN:

(ab 10.2009)

Inhalt der (Windows) Zwischenablage (in den TEXT: Bereich) übernehmen.

ZA_EINFUEGEN: Variable | "{TEXT}".

```
x = "Hallo Welt_Variable".
ZA_KOPIEREN: x.
ZA_EINFUEGEN: y.
Y =? "Hallo Welt_Variable"

ZA_KOPIEREN: "xyz".
ZA_EINFUEGEN: A.
A =? "xyz"

TEXT: = "Hallo Welt_TEXT".
ZA_KOPIEREN: "{TEXT}".
TEXT: ".
ZA_EINFUEGEN: "{TEXT}".
```

WEITER:

(ab 11.2009)

Es kann eine Sprungmarke „WENDE.“ verwendet werden.

Dadurch wird eine Funktionalität ähnlich dem CASE ... END CASE erreicht.

Beispiel 1:

```

*$Sg = "global".
$Sg = "lokal1".

WEITER: $Sg.

global. */ <-----
*Sachgruppe "global" berechnen
X = 1.
LOGENDE.

lokal1. */ <-----
*Sachgruppe "lokal" berechnen
X = 2.
LOGENDE.

WENDE. */ <-----
*alle anderen Sachgruppen
X = 3.

```

Beispiel 2:

```

*$Sg = "global".
$Sg = "lokal1".

OK = 0.
WEITER: $Sg.

global. */ <-----
*Sachgruppe "global" berechnen
X = 1.
OK = 1. WEITER: "WENDE".

lokal1. */ <-----
*Sachgruppe "lokal" berechnen
X = 2.
OK = 1. WEITER: "WENDE".

WENDE. */ <-----
*alle anderen Sachgruppen
WENN: OK = 1. DANN: WEITER: "Schluss". ENDE.
X = 3.

Schluss. */ <-----
* Befehle für alle Sachgruppen
X = 4.

```

Je nach dem welcher Sachbereich gültig ist, wird das entsprechende Programmstück durchlaufen. Wird ein Sachbereich (ausgewählte Option, gefundene Ressource...) gesucht für die keine Sprungmarke existiert, kam bisher eine harte Fehlermeldung dass die Sprungmarke nicht gefunden wurde. Mit dem „WENDE.“ wird das Programm an dieser Stelle fortgeführt. Ist keine Sprungmarke „WENDE.“ vorhanden, ändert sich nichts an der Funktionalität.

E-Logiken mit festen Namen

(CAP Oberfläche ab 11.2009)

Im **CAP-Set** gibt es einen neuen Menüpunkt: **ANSICHT / ALLES AUFKLAPPEN/ZUKLAPPEN**. Dabei wird die E-Logik **\$\$b_Z** gestartet. Das \$\$b steht stellvertretend für Name des Sachbereiches. Die \$\$b_Z startet auch beim Doppelklick auf die Zeilennummer. Durch Systemvariable \$Disabled = „Yes“ oder „No“ kann die Sichtbarkeit der Zeilen gesteuert werden.

Beim Abschließen der Merkmalerfassung wird zusätzlich zu der Summenlogik auch eine **\$\$b_C** Logik gestartet. Die Summenlogik läuft pro Merkmalsblatt 1x durch, die \$\$b_C läuft insgesamt nur 1x.

CAD Schnittstelle

(ab 09.2009)

CAD:

CAD: "Befehl" oder "{TEXT:}".
Befehl wird in IntelliCAD ausgeführt

CADENDE:.

CAD schließen.

CADLESEN:

CADLESEN: [BlockIndex] ItemIndex.

Beim **ItemIndex** = 0 wird nur geprüft ob CAD aktiv ist: \$Fehler = „“.

Beim **ItemIndex** > 0 werden jeweils folgende Systemvariablen ausgelesen:
\$CAD_Name, \$CAD_Id, CAD_Layer, \$CAD_Line, \$CAD_Color,
\$CAD_MinX, \$CAD_MinY, \$CAD_MaxX, \$CAD_MaxY

Manche Objekte liefern zusätzliche Variablen:

Circle	\$CAD_Circumf, \$CAD_Area, \$CAD_Radius, \$CAD_CenterX, \$CAD_CenterY
BlockInsert	\$CAD_BlockName, \$CAD_CenterX, \$CAD_CenterY
Arc	\$CAD_Radius, \$CAD_Length
Line	\$CAD_Length
LWPolyline	\$CAD_Area, \$CAD_Length, \$CAD_Closed, \$CAD_Circumf
Polyline	\$CAD_Area, \$CAD_Length, \$CAD_Closed, \$CAD_Circumf
Ellipse	\$CAD_Area, \$CAD_Radius, \$CAD_CenterX, \$CAD_CenterY
PointEntity	-
Text	\$CAD_Text, \$CAD_Height
MText	\$CAD_Text, \$CAD_Height

Beim **BlockIndex** können die Unterelemente ebenfalls ausgelesen werden.
Verschachtelte Blöcke werden nicht unterstützt.

CADSCHREIBEN:

CADSCHREIBEN: [BlockIndex] ItemIndex.

Beim **ItemIndex** = 0 werden jeweils folgende Systemvariablen übergeben:
\$CAD_Visible, \$CAD_Height, \$CAD_Width, \$CAD_MinX,
\$CAD_MinY

Beim **ItemIndex** > 0 müssen jeweils folgende Systemvariablen vorhanden sein:

Circle \$CAD_Radius, \$CAD_CenterX, \$CAD_CenterY
Line -

Befehl CADSCHREIBEN: wurde nur in Teilen realisiert. Die Funktionalität kann durch den Befehl CAD: realisiert werden.

```
* HilfeCAD: Beispiel für SCHREIBEN und LESEN in IntellicAD
$ANr := "Test".
*CAPPRG.INI
*[User]
*CADPATH=C:\ICAD6CLASSIC\
*CADEXE=icad.exe /L
GETINI: "USER" ("CADPath").          */ CAD-Pfad holen
GETINI: "USER" ("CADEXE").           */ CAD-Aufruf holen

CADENDE:.                             */ CAD ggf. schließen
STARTEN: CADPATH + CADEXE () "F" "S". */ CAD starten
WENN: $Fehler = "F". DANN: "CAD konnte nicht gestartet werden". ENDE.

$CAD_MinX   = 800.    */ Koordinaten von links
$CAD_Width  = 880.    */ Fensterbreite waagrecht (nach rechts)
$CAD_MinY   = 100.    */ Koordinaten von oben
$CAD_Height = 850.    */ Fensterhöhe senkrecht (nach unten)
$CAD_Visible = "Yes". */ Fenster soll sichtbar sein
CADSCHREIBEN: 0.     */ Einstellungen für das CAD-Fenster übertragen

* Zeichnung zuerst erstellen:
CAD: "_cmdecho 1".          */ Befehlsprotokoll einschalten
CAD: "_color red".         */ Farbe setzen
CAD: "_line 0,0 100,100".  */ Linie zeichnen
CAD: ".                    */ Befehl abschließen
CAD: "_zoom _e".          */ maximal vergrößern

CAD: "_color blue".       */ Farbe setzen
CAD: "_circle 50,50 D 100". */ Kreis zeichnen
CAD: "_zoom _e".         */ maximal vergrößern

CAD: "_color yellow".     */ Farbe setzen
CAD: "_arc 0,0 30,80 100,100". */ Bogen zeichnen
CAD: "_zoom _e".         */ maximal vergrößern

CAD: "_color green".      */ Farbe setzen
TEXT: "_Polyline 0,0 20,30 50,60 20,10 a 10,10 _close".
CAD: "{Text:}".
CAD: "_zoom _e".         */ maximal vergrößern

CAD: "_color white".      */ Farbe setzen
TEXT: "text 10,50 10 0 " + $Heute.
CAD: "{Text:}".

* Zeichnungsdaten auslesen und in den Textbereich reinstellen
Index      = 1.
$Fehler    = ".
KreisIndex = 0. */Einen Kreis Gefunden?
TEXT: ".
CADLESEN: Index.

SOLANGE: $Fehler = ". DANN:
WENN: $Fehler = "F". DANN: SOABBRUCH:. ENDE.

WENN: $CAD_Name = "BlockInsert". DANN:
TEXT:+ Index[2.0] + " Id=\\" + $CAD_Id[4.4] + "\\" Name=\\" +
      $CAD_BlockName[-20,30] + "\\tEinfügekpunkt X=" +
      $CAD_CenterX[6.2] + " Y=" + $CAD_CenterY[6.2] + "\\n".
ENDE.
WENN: $CAD_Name = "Circle" ! "Ellipse". DANN:
TEXT:+ Index[2.0] + " Id=\\" + $CAD_Id[4.4] + "\\" Name=\\" +
      $CAD_Name + "\\" Radius=" + $CAD_Radius[6.2] +
      " Fläche=" + $CAD_Area[8.2] + " Umfang=" + $CAD_Circumf[7.2] +
      " Layer=" + $CAD_Layer + "\\n" +
      "\\t\\t\\tX: " + $CAD_CenterX[0,2] + " Y: " + $CAD_CenterY[0,2] + "\\n".
KreisIndex = Index.
KreisUmfang = $CAD_Circumf.
ENDE.
```

```

WENN: $CAD_Name = "Polyline" ! "LightWeightPolyline". DANN:
  TEXT:+ Index[2.0] + " Id=\" + $CAD_Id[4.4] + "\" Name=\" +
    $CAD_Name + "\"\tFläche= " + $CAD_Area[8.2] +
      " Umfang= " + $CAD_Circumf[7.2] + " Layer= " + $CAD_Layer + "\n".
  WENN: $CAD_Closed <> "Yes". DANN:
    TEXT:+ "\t\tAchtung Polyline ist nicht geschlossen!\n".
  ENDE.
WENN: $CAD_Name = "Polyline" ! "LightWeightPolyline". DANN:
  RECHNE: "HilfeCADPolyline".
  ENDE.
WENN: $CAD_Name = "Mtext" ! "Text". DANN:
  TEXT:+ Index[2.0] + " Id=\" + $CAD_Id[4.4] + "\" Name=\" +
    $CAD_Name + "\"\t\t\tHöhe= " + $CAD_Height[8.2] +
      " Layer= " + $CAD_Layer + "\r\t\t\t Text:\\" + $CAD_Text + "\"\n".
  ENDE.
WENN: $CAD_Name = "Line" ! "Arc". DANN:
  TEXT:+ Index[2.0] + " Id=\" + $CAD_Id[4.4] + "\" Name=\" +
    $CAD_Name + "\" Länge= " + $CAD_Length[0,2] +
      "\tLayer= " + $CAD_Layer + "\n".
  ENDE.
  Index +1.
  CADLESEN: Index.
  SOENDE.

* Eigenschaften eines Objektes direkt verändern
WENN: KreisIndex > 0. DANN:
  $CAD_Radius = KreisUmfang / $PI / 2. */Umfang -> Radius
  $CAD_CenterX = 10,5.
  $CAD_CenterY = 30,9.
  CADSCHREIBEN: KreisIndex. */ Kreis verschieben
  CAD: "_regen". */ Zeichnung neu aufbauen/regenerieren
  CAD: "_zoom_e". */ auf Maximum vergrößern
  CADLESEN: KreisIndex.
  TEXT:+ KreisIndex[2.0] + " Id=\" + $CAD_Id[4.4] + "\" Name=\" +
    $CAD_Name + "\" Radius= " + $CAD_Radius[6.2] +
      " Fläche= " + $CAD_Area[8.2] + " Umfang= " + $CAD_Circumf[7.2] +
      " Layer= " + $CAD_Layer + "\n" +
      "\t\t\tX: " + $CAD_CenterX[0,2] + " Y: " + $CAD_CenterY[0,2] + "\n".
  ENDE.
  CADENDE: . */CAD-Programm schließen

```

Unterprogramm für die Polylinie:

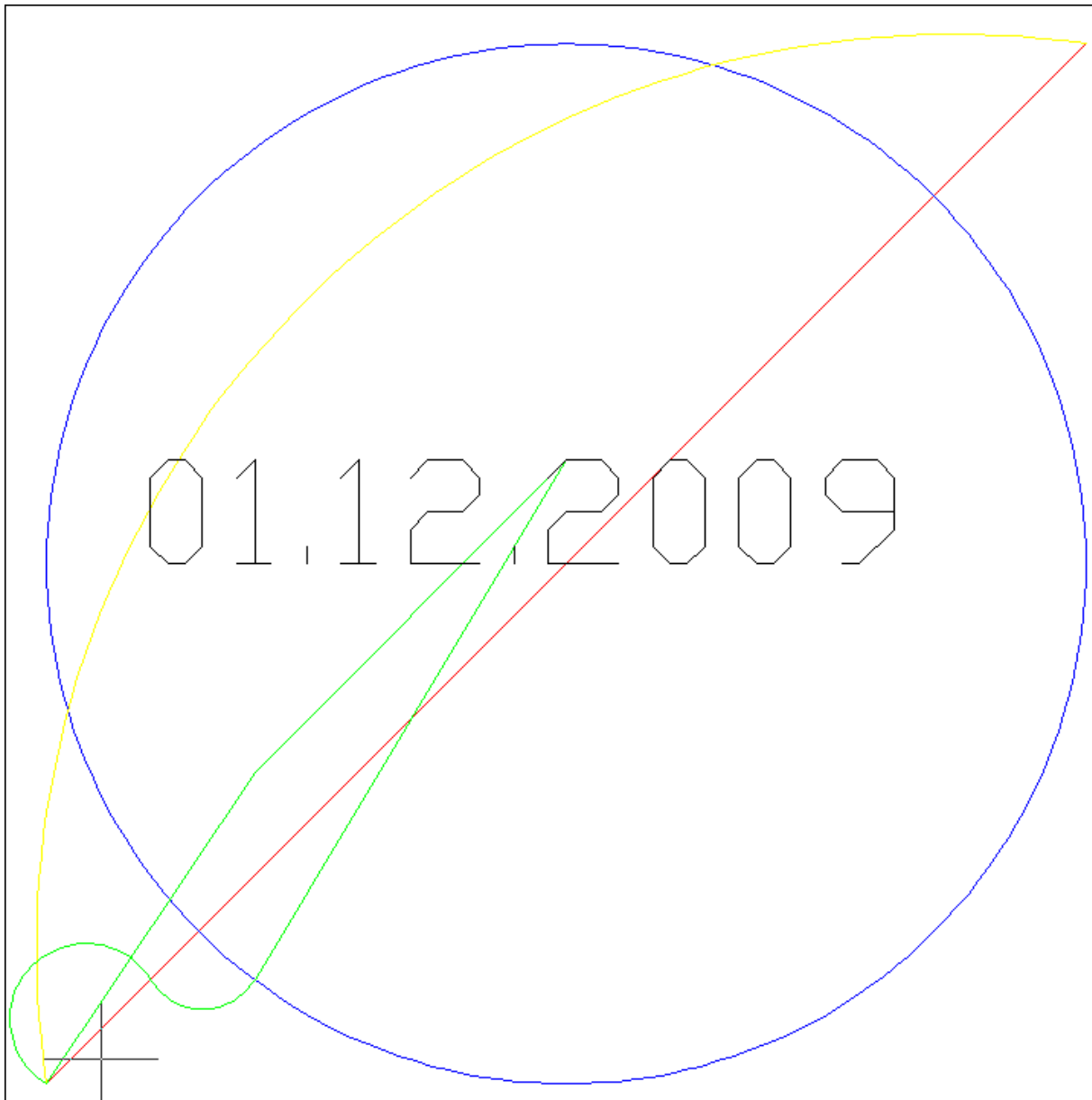
```

* HilfeCADPolyline: Unterprogramm um die Objekte der Polylinie auszulesen

UP_Index := 1.
CADLESEN: Index (UP_Index). */ Segment der Polylinie lesen
WENN: $Fehler = ". DANN:
  TEXT:+ " Ix= " + UP_Index[2.0] + " x= " + $CAD_PointX[9.4] +
    " y= " + $CAD_PointY[9.4] + " Kr.= " +
      $CAD_Bulge[+6.6] + "\n".
  UP_Index +1.
  RESTART:.
  ENDE.

```

Die Zeichnung (vor der Verschiebung des Kreises):



Erstellter Text:

1	Id=" CF"	Name="AVE_RENDER	"	Einfüangepunkt X=	0,00	Y=	0,00
2	Id=" F1"	Name="DIRECT	"	Einfüangepunkt X=	-3,19	Y=	-12,49
13	Id=" 111"	Name="AVE_GLOBAL	"	Einfüangepunkt X=	0,00	Y=	0,00
14	Id=" 138"	Name="Line"	Länge= 141,42	Layer=	0		
15	Id=" 139"	Name="Circle"	Radius= 50,00	Fläche=	7853,98	Umfang=	314,16
			X: 50,00	Y: 50,00			
16	Id=" 13A"	Name="Arc"	Länge= 164,27	Layer=	0		
17	Id=" 14B"	Name="LightWeightPolyline"		Fläche=	361,07	Umfang=	175,28
				Layer=	0		
	Ix= 1	x=	0,0000	y=	0,0000	Kr.=	+0,000000
	Ix= 2	x=	20,0000	y=	30,0000	Kr.=	+0,000000
	Ix= 3	x=	50,0000	y=	60,0000	Kr.=	+0,000000
	Ix= 4	x=	20,0000	y=	10,0000	Kr.=	-0,566190
	Ix= 5	x=	10,0000	y=	10,0000	Kr.=	+1,280776
18	Id=" 14C"	Name="Text"		Höhe=	10,00	Layer=	0
		Text:	"02.12.2009"				
15	Id=" 139"	Name="Circle"	Radius= 50,00	Fläche=	7853,98	Umfang=	314,16
			X: 10,50	Y: 30,90			

Systemvariablen:

(CAP Oberfläche ab 11.2009)

\$OK = Zustand in der Merkmalerfassung.

Sind alle Felder fehlerfrei, ist die Variable \$OK = „True“, sonst \$OK = „False“.

Durch Abfrage dieser Variable kann z.B. wenn die Merkmalerfassung nicht korrekt ist verhindert werden: dass Artikelstämme angelegt werden, Daten Exportiert werden....

\$Disabled = „YES“ oder „NO“ im Set (z.B. Ressourcenliste, Fertigungsauftrag...)

Die aktuelle Zeile wird angezeigt oder ausgeschaltet. Diese Systemvariable kann in einer E-Logik eingesetzt werden, die durch alle SET-Positionen durchläuft.

Das wäre z.B. die E-Logik im Sachbereich/Sachgruppe sowie die \$Sb_Z Logik.

\$ELogik = „Test“ Name der aktuellen E-Logik.

\$CAD_Area	<-	die Fläche der Polylinie
\$CAD_BlockName	<-	Name des Blockes
\$CAD_Bulge	<-	Ausbuchtung in der Polylinie
\$CAD_CenterX	<->	Mitte des Kreises, der Ellipse oder des Blockes
\$CAD_CenterY	<->	Mitte des Kreises, der Ellipse oder des Blockes
\$CAD_Circumf	<-	Kreisumfang im Kreis und Polylinie
\$CAD_Closed	<-	Ist die Polylinie geschlossen?
\$CAD_Color	<-	Farbe des Objektes
\$CAD_Height	<-	Texthöhe
	->	Höhe des CAD Fensters
\$CAD_Id	<-	Id Nummer des Objektes
\$CAD_Layer	<-	Name des Layers (Ebene)
\$CAD_Length	<-	Länge des Objektes
\$CAD_Line	<-	Linienart des Objektes
\$CAD_MaxX	<-	größte Ausdehnung in X
\$CAD_MaxY	<-	größte Ausdehnung in Y
\$CAD_MinX	<-	kleinste Ausdehnung in X -> linke Seite des CAD-Fensters
\$CAD_MinY	<-	kleinste Ausdehnung in Y -> Oberkante des CAD-Fensters
\$CAD_Name	<-	Name des Objektes (_line, _arc...)
\$CAD_PointX	<-	Knotenpunkt der Polylinie
\$CAD_PointY	<-	Knotenpunkt der Polylinie
\$CAD_Radius	<->	Radius des Kreises, Ellipse oder Bogens
\$CAD_Text	<-	Text als Zeichenkette
\$CAD_Visible	->	Sichtbarkeit des CAD Fensters
\$CAD_Width	->	CAD Fensterbreite