

CAP-News 55

CAP-News ist eine in unregelmäßigen Abständen erscheinende Information zum Produktkonfigurator CAP-Suite und Infor mit CAP oder VARIANTS.

Hier werden die neuen Befehle, Funktionen und Möglichkeiten beschrieben.

In „CAP-News komplett“ sind alle in der letzten Zeit erschienen Informationen zusammengefasst.

Der komplette Befehlsumfang ist in der jeweiligen Programmhilfe und in der aktuellen Schulungsunterlage enthalten.

November 2023

CAP-Suite und CAP-/ VARIANTS ab R7.1

- CAP-Suite	1
Tabelle mit Grafiken	1
- CAP-Online:	2
- InforCOM	2
TABNEXT:.....	2
Tabelle mit E-Logik anlegen	3
Tabellenzeile anlegen/ändern/löschen.....	3
UP-Sätze	4
- InforCAP.....	4
Attribute der Merkmale ~	4
Tabellen mit FILTER öffnen.....	4
SML-Ergebniszeilen um Spalten mit z.B. Bestand erweitern	5

- CAP-Suite

Tabelle mit Grafiken

Normblatt

Name	TabelleMitGrafik	Benutzer	ptr	Datum	23.07.18	
Beschr.	TabelleMitGrafik					
Merkmal	Position					
Merkmal	KBez	Len	Dez	M	Einh.	Gültigkeit
1	TKey	Key	10		S	
2	TGrafik	Grafik	10		G	
3	TText	Text	45		T	
4	TKnopf	Knopf	0		E	ELE:TabelleMitGrafik

E-Logik

* TabelleMitGrafik: Auswahl der Grafik für die aktuelle Tabellenzeile

```
Dateiname= "".
Dateipfad = $FDrv + $FDir + "SYSINFO\BMP\\".
OEFFNEN: DateinameDateipfad (Dateipfad "BMP").
WENN: $Fehler = "".
    DANN: TGrafik = MyFile.
    SONST: TGrafik = "".
ENDE.
```

Dialog bzw. Auswahl der Grafik:

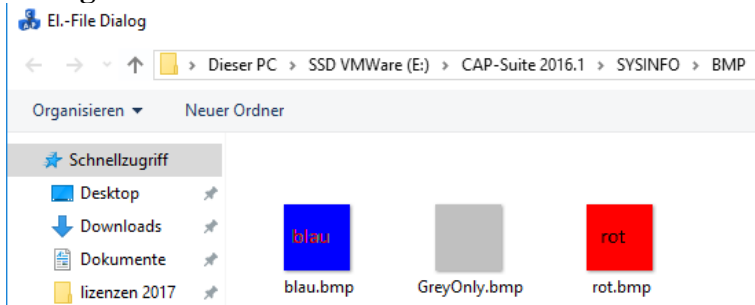


Tabelle mit der Grafik:



- CAP-Online:

- InforCOM

TABNEXT:

TABNEXT: "Tabellenname".

(ab 2023)

Syntax ist identisch wie beim Befehl TABELLE:

Beim nächsten Aufruf wird in der Tabelle ab der aktuellen Tabellenzeile gesucht.

Dadurch wird die Suche in der selber Tabelle wesentlich schneller.

Bei Änderung des Tabellennamens wird die vorherige Tabelle geschlossen bzw. freigegeben.

Leerer Tabellename (**TABNEXT: ""**) beendet die aktuelle Tabellensuche.

Ohne diesen Eintrag wird beim nächsten Start der E-Logik an der letzten Stelle weitergesucht.

Bei Änderung des Tabellennamens wird die vorherige Tabelle geschlossen bzw. freigegeben.

Tabelle:

Name	Ausprägung	Normblatt	Auspraegung
Beschr.	Ausprägung		
	Ob	Ausp	
1	Dekor	Cremeweiß	
2	Dekor	Grafitgrau	
3	Dekor	Kirschbaum	
4	Dekor	Kobaltblau	
5	Dekor	Seidengrau	
6	Furnier	Buche	
7	Furnier	Eiche	
8	Furnier	Esche	

```
Oberflaeche ="Furnier".
* TABNEXT z.B. in einen SOLANGE: Schleife abfragen.
*
TABNEXT: "Ausprägung". */ 6 Buche
TABNEXT: "Ausprägung". */ 7 Eiche
TABNEXT: "Ausprägung". */ 8 Esche
TABNEXT: "Ausprägung". */ $Fehler = "F"
TABNEXT: ". */ Suchindex wieder auf die 1. Zeile setzen
```

Tabelle mit E-Logik anlegen

(ab 2023)

Beispiel:

```
* eine neue Tabelle anlegen
LESEN: "Tabelle1" "TAB" (). */ Kopiervorlage lesen
DUP: "Tabelle2" "TAB" (). */ Tabelle "Tabelle2" wird angelegt
* gibt es Spaltennamen = Variablen, werden die Werte gleich in die 1. Zeile
eingetragen
```

Tabellenzeile anlegen/ändern/löschen

(ab 2023)

Beispiel:

```
LESEN: "Ausprägung" TAB.
Oberflaeche = "1Glas".
Auspraegung = "matt".
DUP: "Ausprägung" TAB.
$TabZnr =? 10
```

	Oberflaeche	AuspraegungT
1	1Glas	matt
2	Dekor	Buche

```
* (Zur Info: Das ist jetzt Zeile 10, weil InforCOM die Zeilennummern nicht
neu durchnummeriert, sondern immer die höchste + 1 nimmt)

LESEN: "Ausprägung" $TabZnr TAB (*).
Oberflaeche = "1Blech".
Auspraegung = "Rost".
SCHREIBEN: "Ausprägung" TAB.
```

	Oberflaeche	AuspraegungT
1	Blech	Rost
2	Dekor	Buche

LESEN: "Ausprägung" \$TabZNr TAB.
DEL: TAB.

	Oberflaeche	AuspraegungT
1	Dekor	Buche

UP-Sätze

(ab 2023)

In der Konfiguration kann der Eintrag: COMMON.CAP.UPGEN auf M gesetzt werden. Dann werden nur die UP-Sätze berücksichtigt, die VARIANTS-Merkmale haben.

- InforCAP

Attribute der Merkmale ~

(ab 2023)

Die Eigenschaft eines Merkmals (Falsch, aktiv...) kann abgefragt werden.

MERKMAL: ("F" "A") Länge.
 WENN: Länge ~ "F" UND: Länge ~ "A".
 DANN: MELDUNG: "Eigenschaft F und A sind gesetzt".
 ENDE.
 WENN: Länge ~ "F" ODER: Länge ~ "A".
 DANN: MELDUNG: "Eigenschaft F oder A sind gesetzt".
 ENDE.

Tabellen mit FILTER öffnen

(ab 2023)

Tabelle kann mit einem Filter geöffnet werden:

Table öffnen/speichern ✕

Tabellenname: OK

Filter / Abbruch

Normblattname: Suchen

Filter

Dann werden nur die Tabellenzeilen angezeigt, die dem Filter entsprechen:

ZDG	Brüniert	9,00	ZDG	Brüniert	9,00
ZDG	Verchromt	10,00	ZDG	Verchromt	10,00
ZDG	Vermessingt	11,00	ZDG	Vermessingt	11,00
ZDG	Vernickelt	12,00	ZDG	Vernickelt	12,00
ZDG	Weiß	13,00	ZDG	Weiß	13,00

SML-Ergebniszeilen um Spalten mit z.B. Bestand erweitern

(ab 2022)

Sachbereich	HALBZEUG		Sachgruppe		FLACHMAT	
Ressource	Werkst.	Dicke in mm	Breite in mm	Lagerbest. in meter	Verf. Best. in meter	
F602-0013	AL	5	20	10.000	10.000	
F602-0019	AL	5	30	0.000	0.000	
F602-0025	AL	5	40	0.000	0.000	
F602-0031	AL	5	50	0.000	0.000	
F602-0014	AL	6	20	0.000	0.000	
F602-0020	AL	6	30	0.000	0.000	
F602-0026	AL	6	40	0.000	0.000	
F602-0032	AL	6	50	0.000	0.000	

```

* HALBZEUG_SML, Startet bei der SML-Suche und kann die Ergebniszeilen ändern.
* Startet automatisch durch die Namenskonvention: $Sb_SML
* Jan.Petrasch@ABi-CAP.com
* 29.11.2022
* -----
* Geändert: Kommentar
* -----
*TEST: "Ein".

FRAGE: "Sollen die Ergebniszeilen um die Lagerbestände erweitert werden?".
WENN: $Fehler = "F". DANN: LOGENDE. ENDE.

WEITER: $Nb. */z.B. "Flachmat".

Flachmat.
WENN: $ZNr = 0.
  DANN: TEXT: = "Werkst.\tDicke\tBreite\tLagerbest.\tVerf.Best.\r".
        TEXT: + " \tin mm\tin mm \tin meter \tin meter ".
  SONST: Dl_MNr = $MNr. Gutmenge = 0. Verfuegbar = 0.
        VQL: "USvqlCAPGutmenge" (DfPart.MNr) (DfPart.Gutmenge).
        LJ4: "CAP.VerfuegbarerBbestandSML" (Dl_MNr Verfuegbar).
        TEXT: = Werkstoff[7.7] + "\t" + Dicke[5.0] + "\t" + Breite[6.0] +
              "\t" + Dl_Gutmenge[9.3] + "\t" + Verfuegbar[9.3].

  ENDE.
LOGENDE.

WENDE. */ alle anderen Normblätter des SML-Sachbereiches
* -----

```

```

ScriptDataDef VQLStatement USvqlCAPGutmenge

#{+Comment}
// 29.11.2022 Jan.Petrasch@ABi-CAP.com (i)
// relDl = Lagerbewegungen

SELECT DfPart.Gutmenge = SUM(DfPart.Gutmenge)
  FROM relDl BY relDl
 WHERE Dfpart.MNr = :Dfpart.MNr
#{-Comment}

End // ScriptDataDef VQLStatement USvqlCAPGutmenge

```

```

Function CAP.VerfuegbarerBbestandSML(
  ByVal MNr      As tString,
  VBestand As tReal
) As tBoolean
/*-----
Verfügbare Menge ermitteln

```

```

@F(Parameter)
IN: MNr           As String Ressourcennummer

@F(Rückgabewert)
VBestand         As tReal
VerfuegbarerBbestand As tBoolean

@EndComment
-----*/
/*
29.11.2022 Jan.Petrasch@ABi-CAP.com (i)
*/

Dim hItemVt      As tHandle //VTabMitMNR
Dim hZwiSpActVt As tHandle //ZwiSp.ActVt
Dim Modus        As tByte   //welcher Bestand zurückliefern?
Dim Tmp          As tByte   //Wert aus vM4Aufrufer zwischenspeichern
Dim pName        As tString //Name der ActVt

hItemVt = VtCreate("viwDbVt", "E-Logik", 0) // VTab erstellen
VtInsertLine (hItemVt, 1, True)           // eine leere Zeile einfügen
// übergebenen Parameter in die VTab übertragen
VtSetStr (hItemVt, "DbSatz.DbPart.MNR", MNr, cVtUseActLine, False)

hZwiSpActVt = VtCreateGlbHandle(pName) // Handle der ActVt
VtPushGlbHandle (hItemVt)             // VTabMitMNR -> ZwiSp
Lja.M5A02(1,2)                        // ZwiSp -> Work-Vt
Lja.M5B9(-1)                          // Ressourcendaten auslesen
VtPushGlbHandle (hItemVt)             // VTabMitMNR -> ZwiSp

Modus = 2
// 1 = Bestand unter Berücksichtigung aller geplanten Zugänge oder
// 2 = physischer Bestand - Reservierungen
// 3 = physischer Bestand - alle Bedarfe ungeachtet des Zustands
//   (nur für Kommissionierung!!)

Tmp = vM4Aufrufer // Wert von vM4Aufrufer merken
vM4Aufrufer = Modus // und setzen für nachfolgende Programme

vM4VPEigenbedarfAbziehen = 1 // = Eigenbedarf nicht berücksichtigen
bfrmM4.VerfPruefInit(vM4Aufrufer) //Initialisierung der Prüfung
dlgM4VerfPr.IdleAction() // Abbrechen-Dialog: Verfügbarkeitsprüfung

vM4Aufrufer = Tmp // vM4Aufrufer wieder restaurieren
bfrmM4.VerfPruefExit(Modus) // Exit-Routine der Verfügbarkeitsprüfung
VtPushGlbHandle (hZwiSpActVt)
HdlDestroy (hZwiSpActVt) // Handle löschen

VBestand = VtGetReal (hItemVt, "DbCalcTerm.ZSeg(2).Meng") // VTab->Variable

VtDestroy (hItemVt) // VTab löschen
VerfuegbarerBbestandSML = True // -> $Fehler = ""

End Function // CAP.VerfuegbarerBbestandSML

```