

ab oder ruft ein CGI-Script auf einem WWW-Server auf. Die Antwort des WWW-Servers wird in die angegebene Datei geschrieben.

Schleifen

\$\$ LOOP variable = Anfangswert TO Endwert STEP Schrittweite {436}
 ist eine Alternative zu der Anweisungsform, bei der statt TO und STEP jeweils Kommata stehen.

Stapelspeicher

\$\$ STACK Beim Lesen von einem leeren Stapelspeicher {450}
 verhindert die Option QUIET (wie bisher schon die Option EXIT beim Lesen innerhalb von Schleifen), dass eine Fehlermeldung ausgegeben und das Fehlerflag gesetzt wird.

Makrofunktionen zur Dateiverwaltung

COMPARE (name1, name2) {489}
 drei statt nur zwei Funktionswerte, wenn die Daten verglichen werden konnten:
 - "YES", wenn die Daten und die Satznummern identisch sind,
 - "DATA", wenn nur die Daten identisch sind,
 - "NO", wenn die Daten nicht identisch sind.

Makrofunktionen zur Dateisperre

LOCK (name, modus, limit) {490}
 limit ist eigenes, durch Komma statt durch Schrägstrich von modus abgetrenntes Argument.

Makrofunktionen für Dateiinhalte

FILE (name, auswahl, anzahl, code) {494}
 Das neue Argument CODE erspart bei Fremddateien die Umwandlung der Daten in die TUSTEP-Codierung mit DECODE. Vorgesehen sind: ISO, UTF8, UTF16, BINARY oder "-" (ohne Anführungszeichen) oder BINARY/NLF.

WRITE (pfad, daten, code, limit) {495}
 Neue Makrofunktion:
 Schreibt den Inhalt der Variablen daten in die mit pfad angegebene Datei.

READ (pfad, daten, code, limit) {496}
 Neue Makrofunktion:
 Liest den Inhalt der mit pfad angegebenen Datei und speichert ihn in die Variable daten.

Makrofunktionen für beliebige Variablen-Inhalte

FIND_DIFF (var1, apos1, epos1, var2, apos2, epos2, char1,
char2) {538}

Falls var1 und var2 Sternvariablen sind,
kann nach char2 nach einem Komma noch
eine Variable nummer angegeben werden,
in die die laufende Nummer der Zeile
gespeichert wird, in der ein Unterschied
gefunden wurde.

COMBINE (werte, erg2, erg3, erg4) {552}

Beim Zusammenfassen von aufeinanderfolgenden
Zahlenwerten können jetzt bis zu drei Variablen
angegeben werden, die Zeichenfolgen enthalten,
die an die jeweils erste Zahl angehängt werden.
Der Inhalt der ersten Variable wird dort
eingefügt, wenn nur zwei aufeinanderfolgende
Zahlenwerte vorhanden sind; der Inhalt der
zweiten Variablen gilt für 3, der Inhalt der
dritten Variablen für 4 und mehr aufeinander-
folgende Zahlenwerte.

Makrofunktionen für Tags

RENAME_ATTRIBUTE (tag, altername, neuername) {563}

ersetzt die bisherige Funktion
RENAME_ATTRIBUTE (tag, namen)

Makrofunktion zum Decodieren

DECODE (variable, modus) {577}

Als modus ist jetzt auch HEX6 (für Zahlen,
die mit 6 Hexadezimalzeichen dargestellt sind)
und HEX möglich. Mit dem Modus HEX wird jeder
Hexadezimal-Code (jeweils 2 Bytes) in das
entsprechende Zeichen (ein Byte) umgewandelt.

Makrofunktion zum Codieren

ENCODE (variable, modus) {578}

Als modus ist jetzt auch HEX6 (für Zahlen,
für die 6 Hexadezimalzeichen erforderlich sind)
und HEX möglich. Mit dem Modus HEX wird jedes
Zeichen (jedes Byte) in den entsprechenden
Hexadezimal-Code umgewandelt.

Dateizugriffe, Allgemeines

ACCESS daten: modus/option "dateiname" ... {597}

Wenn die Daten durch frei wählbare Zeichenfolgen
oder XML-Tags in Portionen eingeteilt werden
(Option STREAM), können mit der READ-Anweisung
auch Daten statt aus einer Datei aus einer
Variablen gelesen werden und mit der WRITE-

Anweisung in eine Variable statt in eine Datei geschrieben werden. In diesem Fall muss nach den Optionen statt eines Dateinamens der Name der entsprechenden Variablen angegeben werden.

Ist nach `modus/option` eine Variable statt `{598}` eines (in Anführungszeichen stehenden) Dateinamens angegeben, so muss zu Optionen an erster Stelle `FILE` oder `VARIABLE` angegeben werden.

Der "Dateizugriff" auf eine Variable erfolgt nach den gleichen Regeln wie auf eine Fremd-Datei, jedoch kann die Angabe eines Codes entfallen, wenn die Daten nicht umcodiert werden sollen.

#SATZ

Parameter

SCH	neu: 32001 TusLibertine0-Heb (so genannt in <code>tustep*szschrift</code>)	{1158}
AUS	Für den Mindesteinzug in Ausgangszeilen kann zu <code>IEINZR</code> (zweiter Wert) außer 1 (= mindestens 10 Punkt Einzug) ein beliebiger Wert (z.B. 40 für mindestens 40 Punkt Einzug, falls die Satzausgabe um den Faktor 4 größer aufbereitet wird) angegeben werden.	{1170}
MAH	Sollen bei Tags mit Attributen Attributwerte unverändert in die Makro-Auflösung übernommen werden, so kann statt des Attributs ein <code>"**"</code> angegeben werden; in der Makro-Auflösung wird an der Stelle, an der der Wert des Attributs eingesetzt werden soll, <code>"{=an}"</code> angegeben, wobei für <code>n</code> die laufende Nummer des im Tag durch <code>"**"</code> ersetzten Attributs ist.	{1191}
	Beispiel: Mit dem Parameter <code>MAH <page n="**"/></code> (Seite <code>{=a1}</code>) wird erreicht, dass aus dem Quelltext <code>"weiter unten <page n="13"/></code> beschrieben." beim Satz folgender Text erzeugt wird: <code>"weiter unten (Seite 13) beschrieben."</code>	{1192}
	MAH <code><see kap="**" pg="**"></code> (Seite <code>{=a2}</code> , <code>"{=a1}"</code>) ergibt beim Quelltext <code>"haben wir bereits <see kap="Einleitung" pg="3"></code> beschrieben" in der Satzausgabe: <code>"haben wir bereits (Seite 3, "Einleitung")</code> beschrieben"	

Steueranweisungen

#9+ ... #9-	tiefgestellte Unterstreichung	{1258}
#49+ ... #49-	Unterpunktierung + Unterstreichung	
^&^#x00A0;	wird, wenn als 3. Wert von Parameter LAU eine 1 angegeben ist, wie #._ behandelt (non-breaking space, das aber zum Zeilenlängen-Ausgleich genutzt wird)	{1278}

Makros für die Satzumgebung#*FUNO

KENNUNGEN	Zu KENNUNGEN können jetzt auch Zeichenfolgen für den Anfang und das Ende der Fußnoten sowie für die Rückschaltung auf Fußnoten-Nummer 1 angegeben werden, die nicht aus dem Text entfernt werden sollen (z.B. XML-Tags <fussnote> </fussnote>). Dazu muss ein "+" oder " +" hinter der betr. Zeichenfolge angegeben werden	{1345}
-----------	---	--------

#*PS2PDF zusätzliche Spezifikationen:

BREITE=nnn	Breite der auszugebenden PDF-Seiten in Punkt
HOEHE=nnn	Höhe der auszugebenden PDF-Seiten in Punkt

#*PSAUS {1364}

PARAMETER=	in den Parametern sind jetzt auch Kommentare möglich (sie beginnen mit drei Spatien)
------------	--

#*PSDICKTEN {1378}

ZEICHENSATZ=HB	hebräischer Zeichensatz
----------------	-------------------------

#*PSFONT {1382}

MODUS=HEB	Dem vorliegenden AFM-File liegt der Encoding Vector für hebräische Fonts zugrunde. (Schriftnummern 45001 bis 45049 für hebräische Fonts reserviert)
-----------	--

#*PSFONTVOR

ZEICHENSATZ=HEB	Zeichensatz für das Hebräische auswählen	{1384}
-----------------	--	--------

#*TAGS {1409}

META=tag	Zur Spezifikation META angegebene Namen werden jetzt auch innerhalb anderer Elemente von der Auswertung ausgenommen
----------	---

(kommt z.B. in Sammelwerken vor, die Meta-Daten für die einzelnen Beiträge haben)

#*TAGUEB

{1412}

MODUS==STD- (Voreinstellung): Groß- und Kleinschreibung in den Tags wird unterschieden (wie in XML)

MODUS=- Groß- und Kleinbuchstaben in den Tags gelten als gleichwertig

#*XMLZIEL

{1417}

MODUS=S | TS | PS wie MODUS ==STD- | T | P
aber Silbentrennung aufheben

TAGENTF=tag Zur Spezifikation TAGENTF angegebene Tag-Namen schließen auch Elemente ein, die mit Tags mit Attributen gekennzeichnet sind.

