

Neuerungen in TUSTEP Version 2008

gegenüber Version 2007 (Übersicht)

Stand: 12.11.2007

(*"+"*: hinter dem Seitenverweis: im gedruckten Handbuch 2008 noch nicht beschrieben)

TUSTEP-Fenster

Im TUSTEP-Fenster steht neben den Schriftarten Lucida Console und Rasterschrift jetzt auch die Courier New zur Verfügung.

Funktionstasten auf Kommando-Ebene

{114}

F6 (CALL_E) ruft das Standard-Makro #*E (öffnet Fenster zum Auswählen und anschließenden Edieren von Dateien).

Aufruf

{63}

Bei der Definition von Remote-Sitzungen kann unter DELAY die zur Synchronisation der Verbindung benötigte Wartezeit bis zum Start der TUSTEP-Sitzung (in Sekunden; Voreinstellung: 5) angegeben werden.

Starten des Editors unter Windows

{83}

Über die System-Variable TUSTEP-Editor kann eine Definitionsdatei für den Editor angegeben werden.

Kommandos

#DEFINIERE,ZEILEN=80

{110}+

Das TUSTEP-Fenster kann bis zu 80 Zeilen hoch sein, der Scroll-Bereich kann bis zu 800 Zeilen umfassen.

#LISTE,MODUS=SEGMENTE

{143}

Listet Namen von Dateien mit geholten Segmenten; markiert Dateien mit noch nicht wieder zurückkopierten Segmenten.

#KOPIERE

Einsetzen von Zahlenwerten:

{801}

Sollen nur vorhandene Zahlenwerte ersetzt und überzählige I-Variablen ignoriert werden, so ist im Parameter EIZ für die zu ignorierenden Variablen ein um 100 erhöhter Wert anzugeben.

Austauschen von Zeichenfolgen mit Bedingungen

{804}

Als 5. Wert zu Parameter ERZ kann angegeben werden, wie oft der Parameter XXB unmittelbar hintereinander (ausgehend vom Ergebnis der vorangehenden Ausführung) ausgeführt werden soll.

#SVORBEREITE

Neuer Parameter FSZ {931}+
Zusatzangeben zu Parameter FS: Gibt an, ob die
Sortiereinheiten oder die Fortsetzungseinheiten
markiert sind

Standard-Makros

#*CMD

Richtet ein Script ein, das mit einem System-Kommando
aufgerufen werden kann. Beim Aufruf des Scripts wird
TUSTEP aufgerufen und das gleichnamige Makro ausgeführt.

#*DESI, ORT=..., ORDNER=...

erlaubt, Icons im Startmenü unter "Programme" bzw.
auf dem Desktop im angegebenen Ordner anzulegen.

#*IMPORT

kann außer Dateien, die von WinWord im Format
"Word 2003 XML-Dokument" abgespeichert wurden,
jetzt auch RTF-Dateien nach TUSTEP importieren.

#*KOMA

schreibt in die zur Spezifikation DATEI angegebene
Datei ein Makro, das als Vorlage zum satz- oder
portionsweisem Kopieren von Dateien dienen kann.

#*UPLOAD und #*DOWNLOAD

falls Angabe zu PFAD mit \ endet, wird der Name der
QUELL-Datei (bei #*UPLOAD) bzw. der ZIEL-Datei (bei
#*DOWNLOAD) - ohne Projektname - ergänzt.

#*TUSTEP2STICK

kopiert TUSTEP so auf einen USB-Stick, dass TUSTEP
auf anderen Windows-Rechnern, auf denen TUSTEP nicht
installiert ist, aufgerufen werden kann. Dort kann
unter Start-->Ausführen l:\name\start.exe eingegeben
werden (l = Laufwerksbuchstabe des USB-Sticks,
name = der beim Aufruf dieses Makros vergebene
Name der Anwendung).

#*E, LESEN=..., SCHREIBEN=..., TRAEGER=..., MAKRO=...

Anmelden (zum Lesen oder Schreiben) und Edieren der
angegebenen Datei.

#*E

Edieren einer Datei, die über ein Fenster ausgewählt
werden kann.

#*ES

Edieren oder Löschen von Scratch-Dateien, die über ein Fenster ausgewählt werden können.

Editor

Definitionen

tpl, tsl leere Tags definieren {126}
(in Definitions-Datei)

y,fn_#=zfstab (Suchen im Dateinamen) {243}

y,ft_#=zfstab (Suchen im Dateititel)

y,fc_#=zfstab (Suchen im ersten Satz)

(Beim Aufruf des Editors bzw. Wechseln der Datei wird geprüft, ob im Dateinamen / Dateititel / in der ersten Datenzeile eine der zu zftab angegebenen Zeichenfolgen vorhanden ist, und ggf. das zugehörige Makro fn_# / ft_# / fn_# aufgerufen.)

Editor-Datei abfragen/wecheln {238}

d,? Mit DEL kann ein Name aus der angezeigten Liste gelöscht werden.

Textfenster {239}+

Wechseln zwischen zwei Textfenstern durch beliebige Maus-Aktivität (ohne Extra-Click) möglich.

Funktionstasten {241}

F6=d,? statt bisher: f6=x #tue,<editor>
(bisherige Belegung bei Bedarf in TUSTEP.INI eintragen!)

Editorfenster

Das Editorfenster kann bis zu 80 Zeilen hoch sein. {282}

Tastenkombinationen für Makroaufrufe {287}

Tastenkombination Makroname

SHIFT+ALT+a SA_A

Ctrl+ALT+a CA_A

SHIFT+Ctrl+a SC_A

Steuerbefehle

CLR_CMD_LINE Text in Anweisungszeile löschen {304}

APP_MRK:text text an Zwischenspeicher anhängen {308}

HIDE_MACROS Makroleiste nicht mehr anzeigen {314}

EDIT_FILE Namen der Editor-Datei ausgeben {318}

BROWSE:"file" (Windows:) Starten der zur Endung des Dateinamens definierten Anwendung {320}

BRS_MRK wie BROWSE:"file", für die Datei, deren Name im Zwischenspeicher steht {320}

REC_NR_ON	Daten mit Satznummern anzeigen	{322}
REC_NR_OFF	Daten ohne Satznummern anzeigen	
ENCODE_ON	Daten im eingestellten Code anzeigen	{322}
ENCODE_OFF	Daten in Eingabe-Codierung anzeigen	
SAVE_CMD	Anweisungszeile merken und löschen	{323}
REST_CMD	Anweisungszeile wieder herstellen	
Tastenkombinationen für Steuerbefehle		
SHIFT+Backspace	DEL	{301}
SHIFT+Return	ENTER	{323}

Makros

Interpretationsmodi einstellen		{335}
\$\$ MODE STATEMENT	Nicht gekennzeichnete Zeilen werden als Makro-Anweisungen angesehen.	
\$\$ MODE DATA	Nicht gekennzeichnete Zeilen werden als Daten angesehen.	
\$\$ MODE TUSCRIPT	wie MODE STATEMENT und MODE VARIABLE, außerdem geschweifte Klammern für die Steuerzeichen "Klammern".	{336}+

Wertzuzuweisungen

\$\$ set @var = ...	Wertzuzuweisung erfolgt auf die Variable, deren Name in var steht.	{340}
---------------------	--	-------

Signalton

\$\$ BEEP "tonfolge"	Gibt die angegebene Tonfolge als Signal aus (vgl. #SIGNAL).	{351}
----------------------	---	-------

Wörterbücher

\$\$ DICTIONARY name CREATE/INFIX		{362}
\$\$ DICTIONARY name INFIX[/COUNT] key, num, cnt, txt	für Wörterbücher, deren Textteil aus dem Schlüsselwort + zusätzlichen Zeichen besteht.	{363}
\$\$ DICT name UNLOAD/FILE[/COUNTER] "datei", "trenner"		{364}
\$\$ DICT name LOAD/FILE[/COUNTER] "datei", "trenner"	Schreibt den Inhalt des Wörterbuchs in die angegebene Datei und umgekehrt.	{364}

Compilieren von Makroanweisungen

\$\$ COMPILE	Die zwischen den beiden Anweisungen stehenden Makro-Anweisungen werden vor der Ausführung compiliert.	{369}
\$\$ ENDCOMPILE		

\$\$ SECTION/COMPILE, \$\$ SUBMACRO/COMPILE	{ 371 }
Option /COMPILE zum Compilieren von Sektionen und Submakros.	
Starten und Beenden von Anwendungen	
\$\$ BROWSE/CANCEL "titel"	{ 374 }
(Windows:) beendet Anwendungen, deren Fenstertitel mit dem angegebenen Titel übereinstimmt.	
Anstehende Kommandos löschen	{ 375 }
\$\$ CANCEL	löscht alle noch zur Ausführung anstehenden Kommandos.
Vergleichen von Zeichenfolgen	
\$\$ IF ("zf1" .HN. "zf2")	{ 379 }
ist wahr, wenn <i>zf1</i> ein Tag ist, das den Namen <i>zf2</i> hat.	
\$\$ IF ("zf1" .HA. "zf2")	{ 379 }
ist wahr, wenn <i>zf1</i> ein Tag ist, das ein Attribut <i>zf2</i> hat.	
Prüfen von Zeichenfolgen mit Hilfe von Schlüsselwörtern	
\$\$ IF ("zeichenfolge" .EQ. TITLE)	{ 381 }
ist wahr, wenn <i>zeichenfolge</i> der Titel eines Fensters ist.	
Prüfen von Dateinamen und Dateien	
\$\$ IF ("zeichenfolge" .EQ. FILE)	{ 383 }
\$\$ IF ("zeichenfolge" .EQ. DATEI)	ist wahr, wenn <i>zeichenfolge</i> der Name einer Datei auf dem Standard-Träger für permanente Dateien ist.
\$\$ IF ("zeichenfolge" .EQ. READ .ON. traeger)	{ 384 }
ist wahr, wenn <i>zeichenfolge</i> der Name einer zum Lesen oder zum Schreiben angemeldeten Datei auf <i>traeger</i> (Systemvariable) ist.	
\$\$ IF ("zeichenfolge" .EQ. WRITE .ON. traeger)	{ 384 }
ist wahr, wenn <i>zeichenfolge</i> der Name einer zum Schreiben angemeldeten Datei auf <i>traeger</i> (Systemvariable) ist.	
\$\$ IF ("zeichenfolge" .EQ. TAPE)	{ 384 }
ist wahr, wenn <i>zeichenfolge</i> der Name einer angemeldeten Band-Datei ist.	
Status-Abfragen	{ 387 }
\$\$ IF (CGI)	ist wahr, wenn der Aufruf des Makros als CGI-Makro erfolgte.

\$\$ IF (CMD) ist wahr, wenn der Aufruf des Makros
 als CMD-Makro erfolgte.

Makrofunktionen zur Dateiverwaltung

DELETE (projekt, traeger) {396}
 löscht das Projekt *projekt* auf dem
 angegebenen Träger.

RENAME (datei_alt, datei_neu) {396}
 benennt eine Datei um.

RENAME (projekt_alt, projekt_neu, traeger) {396}
 benennt Projekt auf dem angegebenen
 Träger um.

Makrofunktionen für Datei-Inhalte

USED (name) Zahl der mit Daten belegten Bytes {399}

TYPE (name) Typ (SDF, SEQ, RAN, TAPE) {399}

TITLE (name) Dateititel {400}

Makrofunktionen für beliebige Variablen-Inhalte

VALUE (@var) liefert den Inhalt der Variablen, {411}
 deren Name in *var* steht.

CAPS (var) Buchstaben werden Großbuchstaben {411}

SUBSTITUTE (var,+,apos,epos,trenner,dict,modus,num,anz) {427}
 unterteilt jeden Bereich der Variablen
 var in Schlüsselwörter und tauscht
 diese anhand des Wörterbuchs *dict* aus.

RENUMBER (var, stab, apos, epos, num, schr, anz) {428}
 ersetzt im angegebenen Bereich der
 Variablen *var* die (hinter den in *stab*
 definierten Zeichenfolgen stehenden)
 Nummern anhand der Variablen *num* und
 und *schr* durch neue.

EXCHANGE (var,xtab,apos,epos,xnum,xanz,num,anz,xtabx,nmal) {430}+
 Das neue Argument *nmal* gibt an, wie
 oft (über das Ergebnis vorhergehender
 Durchläufe) das Austauschen wiederholt
 werden soll.

Makrofunktionen für Teilzeichenfolgen

APPEND (var1, var2) {437}
 fügt *var2* als neue Teilzeichenfolge
 an *var1* (auch Sternvariable) an.

COLLECT (var1, var2) {438}
 fügt die in *var2* enthaltenen Teil-
 zeichenfolgen, die noch nicht in *var1*
 enthalten sind, hinten an *var1* an.

FILTER (liste, positiv, negativ, raus)	{442}	
wählt aus der Variablen <i>liste</i> Teilzeichenfolgen aus, die die Bedingungen der Recherchiertabellen <i>positiv</i> bzw. <i>negativ</i> erfüllen. Die nicht ausgewählten Zeichenfolgen werden (bei Bedarf) in die Variable <i>raus</i> geschrieben.		
FILTER_INDEX (liste, positiv, negativ, raus)	{443}	
wie FILTER, als Funktionswert (und ggf. in <i>raus</i>) werden jedoch nicht die ausgewählten Teilzeichenfolgen, sondern ihre Positionen in <i>liste</i> geliefert.		
Makrofunktionen zum (De-)Codieren	{466}	
DECODE (var, UTF16) (jetzt auch UTF16)		
ENCODE (var, UTF16)		
Makrofunktionen zur Anzeige einer Meldung / Anfrage		
MESSAGE (kopf, text)	{470}	
Zeigt Popup-Fenster mit Titel <i>kopf</i> und Meldung <i>text</i> an.		
QUESTION (kopf, text, YES NO)	{470}	
Zeigt Frage <i>text</i> in Popup-Fenster mit Titel <i>kopf</i> an; wird mit ENTER statt mit Click auf die Antwort-Schaltfläche geantwortet, gilt die im Aufruf voreingestellte Antwort.		
Makrofunktionen für Makrofenster		
COLOR (feldname)	{479}	
liefert die aktuelle Farbe des im Aufruf angegebenen Feldes.		
Definition des Dateizugriffs		
ACCESS .../RAW	wie /RECORDS, aber für Dateien, deren Satznummern nicht aufsteigend und nicht alle verschieden sind.	{488}
ACCESS .../RAW/REVERSE	Die Sätze werden in umgekehrter Reihenfolge gelesen.	{488}
ACCESS .../UTF16 (jetzt auch UTF16)		{489}

Satz

Allgemeines

Die Silbentrennungs-Algorithmen wurden erstmals seit Version 2004 überarbeitet; insbesondere wurden die Regeln für zusammengesetzte Wörter und - soweit mit den Regeln fürs Deutsch kompatibel - für lateinische Wörter erweitert.

Spezifikationen

PROTOKOLL = p1'p2 {984}+

In die zweite Protokolldatei wird zusätzlich (mit Zeilennummer 999) die Gesamthöhe jeder Seite ausgegeben.

Parameter

Auswahl der Daten

Bei Parameter BER kann jetzt statt eines Bereichs auch eine Anfangsstelle angegeben werden. {989}+

Schriften

"!" hinter der Schriftnummer einer Schrift mit ungewöhnlich kleiner x-Höhe bewirkt, dass die Akzente unter den Buchstaben nicht zu tief gesetzt werden. {991}+

"!!" hinter der Schriftnummer schiebt die Akzente weniger weit nach oben als !.

Fußnoten

{1015}+

FN, 2. Wert (IFNZLI) > 1 = linksbündig mit festem Abstand zum Text.

FN, 4. Wert (IFNEN1) n = wie bisher
 n+m = wie n, aber zuvor ganze erste Zeile um m einrücken (m = linker Rand für FN-Nummer).

Apparate

APn 6. Wert: 1 statt (voreingest. 0) für lange (textlastige) Apparateinträge. {1019}

Makros

Makros, die in der Regel die selbe Auflösung haben und nur gelegentlich hierarchie-abhängig anders aufgelöst werden müssen, können sowohl mit MAC als auch mit MAH definiert werden. Die hierarchie-abhängige Definition hat jeweils Vorrang. {1021}

Steueranweisungen

Verändern der Satzbreite

`&!S(1,Pnn)` {1038}+
 stellt die Satzbreite auf den mit `&!M(nn)` gemerkten Wert ein.

Fußnoten {1040}+

`&!f-` schaltet Ausgabe der Fußnoten-Linie für aktuelle Seite aus.

Überlaufende Fußnoten werden nicht (wie bisher fälschlicherweise) auf einer Seite weitergeführt, für die eine spaltenhohe Abbildung vorgemerkt war.

Freiraum

`$$\nnn$$` {1045}+
 Wird auf einer Seite mehr als 1 Freiraum nach unten geschoben, erfolgt eine Fehlermeldung.

`$$\nnn$$`
 schiebt den Freiraum nach unten und verringert die Satzspiegelhöhe der Folgespalte (bei mehrspaltigem Satz: aller Folgespalten, die zur selben Seite gehören), um die Höhe des Freiraums.

Zentrieren

`@-N` hebt nur Blocksatz, nicht (wie bisher fälschlich) auch Mitte- oder Rechtszentrierung auf.

Satzzeichen, Sonderzeichen

"-" zwischen Ziffer und Buchstabe (auch Akzentbuchstabe) wird Divis. {1088}+

Sonstige Zeichen und Symbole

Von den von `#UMWANDLE` nicht in TUSTEP-Zeichencodes umcodierten, sondern mit `#[xxxx]` wiedergegebenen Unicode-Zeichen zwischen `#[2000]` und `#[202F]` werden folgende Codierungen vom Satzprogramm akzeptiert: {1090}+

<code>#[2000]</code> (<i>en quad</i>)	= Zwischenraum 500 Bildlinien
<code>#[2001]</code> (<i>em quad</i>)	= Zwischenraum 1000 Bildlinien
<code>#[2002]</code> (<i>en space</i>)	= Zwischenraum 500 Bildlinien
<code>#[2003]</code> (<i>em space</i>)	= Zwischenraum 1000 Bildlinien
<code>#[2004]</code> (<i>three-per-m-space</i>)	= Zwischenraum 333 Bildlinien
<code>#[2005]</code> (<i>four-per-m-space</i>)	= Zwischenraum 250 Bildlinien
<code>#[2006]</code> (<i>six-per-m-space</i>)	= Zwischenraum 167 Bildlinien
<code>#[2007]</code> (<i>figure space</i>)	= Zwischenraum 500 Bildlinien
<code>#[2009]</code> (<i>thin space</i>)	= Zwischenraum 200 Bildlinien
<code>#[200A]</code> (<i>hair space</i>)	= Zwischenraum 100 Bildlinien
<code>#[2010]</code> (<i>hyphen</i>)	= - (Divis)

#[2011] (<i>non-breaking hyphen</i>)	= - (Divis)
#[2012] (<i>figure dash</i>)	■ (Rückweisung)
#[2013] (<i>en dash</i>)	= - (Gedankenstrich, wie \-)
#[2014] (<i>em dash</i>)	■ (Rückweisung)
#[2016] (<i>double vertical line</i>)	=
#[2018] (<i>left single quotation mark</i>)	= `
#[2019] (<i>right single quotation mark</i>)	= ’
#[201A] (<i>single low-9 quotation mark</i>)	= ,
#[201B] (<i>single high-reversed-9 quotation mark</i>)	= `
#[2021] (<i>double dagger</i>)	= †
#[202F] (<i>narrow no-break space</i>)	= Zwischenraum 200 Bildlinien

Linien

&!?(a,l,n,wa,wl) {1092}

Wie &!|(a,l,n), jedoch Wellenlinie mit der Amplitude wa und der (halben) Wellenlänge wl; Angaben = Achtelpunkt.

Griechische Schrift

#.w als zusätzliche Codierung für Digamma {1095}

%; = iota subscriptum auch unter anderen Buchstaben als αηω {1095}

Russische Schrift

^z ^Z für x X zusätzlich möglich {1104}+

Erweiterungen des 50T1-Codes {1116}

&!(2812D254yyyyllllaaaabbbbnn)

&!(2812D255llllaaaabbbbnn)

Makros für die Satzumgebung

#*GRAFIK

ABSTAND=-STD- {1155}+

Die Angaben zum Verschieben werden der BoundingBox der Grafik entnommen: es wird um den für den linken Rand angegebenen X-Wert nach links und um den für den oberen Rand angegebenen Y-Wert nach unten verschoben

SCHNEIDEN=-STD-

{1155}+

Die Angaben zum Schneiden werden der
BoundingBox der Grafik entnommen:
was über die an die linke obere Ecke
der Aussparung verschobene BoundingBox
hinaussteht, wird abgeschnitten.

##PSAUS, ..., PARAMETER = *

{1174}+

Für das Fehlerprotokoll in der zweiten PS-Datei
können die gleichen Auswahlparameter angegeben
werden wie bei ##FPROT. Über den zusätzliche
Parameter XXP kann der Wortlaut der ausgewählten
Meldungen verändert werden.

{1177}+

##SATZ

Wichtige Programmänderungen seit der letzten Version
werden mit dem Aufruf ##SATZ,,NEUERUNGEN angezeigt.

* * * * *

Corrigenda zum gedruckten Handbuch 2008

Seite 110, 8. und 10. Zeile: {110}

 statt "(n=10 bis 60)" muss es heißen:
 "(n=10 bis 80)"

Seite 336, Zeile 20: {336}

 Statt "werden die Modi VARIABLE und DATA (s.o.)"
 muss es heißen:
 "werden die Modi VARIABLE und STATEMENT (s.o.)"

Seite 803, 9. Zeile von unten: {803}

 Statt "können vier Zahlenwerte" muss es heißen:
 "können fünf Zahlenwerte"

Seite 1045, 10 Zeile:

 Statt "Wie \$\$-nnn\$\${" muss es heißen:
 "Wie \$\$nnn\$\${"