

Fremd-Dateien (bisher: System-Dateien) {24}

Die bisher als System-Dateien bezeichneten Dateien heißen jetzt Fremd-Dateien; sie können mit TYP=PDF statt (wie bisher) SDF eingerichtet werden; in der mit #LISTE erzeugten Dateiliste werden sie als Typ PDF- (statt bisher SDF-)Dateien gekennzeichnet.

#DEFINIERE, ZEILEN=++, SPALTEN=++ {110}

Das TUSTEP-Fenster in der Höhe und (im Editor) an die Bildschirmbreite und -höhe anpassen, jedoch max. 80 Zeilen und 160 Spalten.

#MBAUSGABE, #MBEINGABE, #MBINFORMIERE {157}

Mit diesen Kommandos können jetzt auch PDF-Dateien auf virtuelle Bänder geschrieben bzw. von dort gelesen werden. Dateititel werden mit übertragen.

Bei der Information über den Inhalt von Banddateien mit #MBINFORMIERE sind PDF-Dateien daran zu erkennen, dass statt der Zahl der Sätze die Größe der Datei in KByte angegeben wird.

#UMWANDLE

MODUS = TK | KT | TX | XT {212}

Beim Umwandeln wird das letzte Änderungsdatum der Datei mitgeführt.

MODUS = -& bzw. +& {213}  
(nur bei Code=ISO)

Beim Umwandeln einer TUSTEP- in eine Fremd-Datei mit MODUS=-& werden Zeichen, die im Code ISO-8859-1 nicht darstellbar sind, in Character-Entities der Form &#xXXXX; umgewandelt (z.B. #g+b#g- in &#x03B2;); die anderen Zeichen werden wie bei Modus -1 umgewandelt; außerdem gelten die gleichen Besonderheiten wie beim Umwandeln von TUSTEP nach UTF-16 oder UTF-8.

Beim Umwandeln einer Fremd- in eine TUSTEP-Datei mit MODUS=+& werden Character-Entities in die entsprechenden TUSTEP-Codierungen umgewandelt (z.B. &uuml; in ä und &#x03B2; in #g+b#g-); das Zeichen ^ hat die gleiche Bedeutung wie bei Modus +1; außerdem gelten die gleichen Besonderheiten wie beim Umwandeln von UTF-16 oder UTF-8 nach TUSTEP.

Editor

## Holen, Retten

h,datei,segment,träger {235}  
 r,datei,segment,träger

Wird als dritte Angabe ein Träger angegeben, so wird die Datei, falls sie noch nicht angemeldet (bei Anweisung r: zu Schreiben angemeldet) ist, vor dem Holen bzw. Retten auf diesem Träger zum Lesen bzw. Schreiben angemeldet. War die Datei nicht angemeldet, bleibt sie nach dem Holen bzw. Retten zum Lesen angemeldet.

## Einstellen der Farben {281}

Mit Ctrl+F können jetzt auch Farben für das Feld, das die Satznummern enthält, und für Popups eingestellt werden.

## Mausaktionen für Makroaufrufe {291}

Außer mit Shift, Ctrl und ALT können Mausaktionen jetzt auch mit Shift+ALT, Ctrl+ALT und Shift+Ctrl kombiniert und damit die Makronamen

SA\_x, CA\_x und SC\_x erzeugt werden; x steht dabei für die Kürzel der bisher mit Shift, Ctrl und ALT kombinierbaren Mausaktionen.

## Blättern, Scrollen {301}

SHW\_BEG zeigt die Sätze vom Dateianfang an.  
 SHW\_CUR zeigt die Sätze um die aktuelle Satzposition.  
 SHW\_END zeigt die Sätze vom Dateieinde her.

## Markieren {307}

Markieren ist jetzt auch in der Anweisungszeile möglich.

SET\_FND:"text" Define text for MRK\_FND

Definiert die für text angegebene Zeichenfolge zum Suchen mit MRK\_FND ohne vorangehende Markierung eines Textes.

## Satznummer {320}

REC\_NR

Schreibt die aktuelle Satznummer (d.i. die in der Anweisungszeile nach "<<\*>>" stehende Satznummer) an der aktuellen Cursor-Position ins Editorfenster.

## Zwischenablage von Windows {322}

MRK\_INS\_CB

Löscht den markierten Text, ohne ihn vorher im

Zwischenspeicher zu merken, und fügt den Inhalt der Zwischenablage an dieser Stelle ein.

Weitere Steuerbefehle {326}

JOIN ist jetzt: Join records  
(Tastenkombination: Plus-Plus-Return)

JOIN\_LINE Join lines (wie bisher: JOIN)  
(keine Tastenkombination vorgesehen)

## Makros

Datenausgabe in Dateien {349}

```
FILE/ERASE/PROGRAM/PRINT "dateiname" = variable
ENDFILE/PRINT
```

Mit der Option PRINT wird zusätzlich eine Meldung mit Angaben zu den ausgegebenen Daten ins Ablaufprotokoll ausgegeben.

Fehlermeldungen und Fehlerflag {353}

```
ERROR ... filefunction (...)
```

An Stelle der letzten Variablen kann auch eine der Makrofunktionen zur Dateiverwaltung CREATE, OPEN, CHECK, ERASE, DELETE, CLOSE, REARRANGE, COPY oder RENAME angegeben werden, z.B.:

```
$$ ERROR/STOP "QUELLE: ", OPEN (datei,READ,-STD-)
```

Liefert diese "OK", wird das Fehlerflag nicht gesetzt.

Suchtabellen, Austausch Tabellen, Recherchiertabellen

```
RELEASE S_TABLE | X_TABLE | R_TABLE name1, name2, ... {361}
```

Es können auch mehrere Tabellennamen, jeweils durch Komma getrennt, angegeben werden.

Wörterbücher {365}

```
DICTIONARY name APPEND[/QUIET] key, num, cnt,
value1[, value2][; erg]
```

APPEND sucht das in der Variablen key stehende Schlüsselwort und ergänzt die dazugehörigen Textteile durch die in den Variablen value1 und value2 stehenden Textteile.

Soll statt Apostroph eine andere Zeichenfolge als Trennzeichen ergänzt werden, so kann diese als letzte in der DICTIONARY-Anweisung nach einem Strichpunkt angegeben werden.

## Makrofunktionen zur Dateiverwaltung

SET status = CHECK (name) {397}

Ist nur das Argument name angegeben, so wird nur geprüft, ob es einen TUSTEP-konformen Dateinamen enthält.

SET status = REARRANGE (datei) {400}

Reorganisiert Daten der angegebenen TUSTEP-Datei.

SET status = COPY (datei1, datei2) {400}

Kopiert datei1 unverändert nach datei2.

SET status = COMPARE (datei1, datei2) {400}

Funktionswert = YES wenn Dateien (einschl. Satznummern) identisch, sonst NO

SET status = PROJECT (projekt) {401}

Definiert den zu benutzenden Projektnamen

## Makrofunktionen für Datei-Inhalte

SET zeit = MODIFIED (dateiname, segmentname) {405}

Liefert letztes Änderungsdatum eines Segments.

## Makrofunktionen für Datei- und Projektnamen

SET var = FILES (CURRENT | SCRATCH | MACRO) {407}

CURRENT liefert Namen aller angemeldeten Dateien.  
 SCRATCH liefert Namen aller Scratch-Dateien.  
 MACRO liefert Namen aller mit #DEFINIERE definierten Makro-Dateien.

SET var = FILE\_NAMES () {407}

() liefert die mit #DEFINIERE oder DEFINE definierten Dateinamen.

## Makrofunktionen für sonstige Informationen

SET var = COLS (modus) {414}

modus = CURRENT liefert eingestellte Spaltenzahl.  
 = LIMIT liefert unter Windows maximale Spaltenzahl für TUSTEP-Fenster.  
 = PIXEL liefert unter Windows die Anzahl der Pixel für die Schirmbreite.

SET var = ROWS (modus) {414}

modus = CURRENT liefert eingestellte Zeilenzahl.  
 = LIMIT liefert unter Windows maximale Zeilenzahl für TUSTEP-Fenster.

= PIXEL liefert unter Windows die Anzahl der Pixel für die Schirmhöhe.

SET var = SOURCE (MACRO\_FILE | MAIN\_FILE | CURRENT\_FILE) {414}

Liefert Quellen-Information zum ausgeführten Makro:  
Namen der Datei, die das Makro enthält  
Namen der Datei, aus der das Makro ggf. geholt wurde  
Namen der Datei, aus der die Makroanweisungen gerade  
gelesen werden (z.B. aufgrund einer INCLUDE-Anweisung).

SET var = SOURCE (MACRO\_SEGMENT | MAIN\_SEGMENT | CURRENT\_SEGMENT)

liefert den Segment- statt den Datei-Namen.

Makrofunktionen für beliebige Variableninhalte

SET kurz = SHORTEN (lang, n1, " ... ", n2[, tab]) {418}

Ist tab angegeben, so werden nicht Zeichen, sondern  
Wörter zusammengefügt. Als Worttrenner gelten die  
in der Suchtabelle tab enthaltenen Suchzeichenfolgen.

SET var = SEARCH\_ALL (var, stab, apos, epos[, npos]) {440}

Sucht alle Zeichenfolgen; die Anfangspositionen  
werden als Funktionswert, die Zeichenpositionen nach  
den gefundenen Zeichenfolgen in npos übergeben, ggf.  
durch Apostroph getrennt. Ist var eine Sternvariable  
so erfolgt die Suche zeilenweise; die Ergebnisse  
werden ebenfalls in Sternvariablen abgelegt.

Makrofunktionen für Teilzeichenfolgen

SET var = REMOVE (var, #nummern[, raus]) {447}

Zu auswahl kann eine Variable (hier: nummern)  
angegeben werden, die auch mehrere durch Apostroph  
getrennte Zahlen enthalten kann.

SET var = SELECT (liste, #nummern[, rest]) {448}

Zu auswahl kann eine Variable (hier: nummern)  
angegeben werden, die auch mehrere durch Apostroph  
getrennte Zahlen enthalten kann.

Ist als drittes Argument eine Variable angegeben,  
so werden die nicht ausgewählten Teilzeichenfolgen  
bzw. Zeilen dieser Variablen zugewiesen.

Makrofunktion zum Codieren

SET var = ENCODE (var, ISO/ENTITIES) {475}

Mit diesen Modi können Daten vom TUSTEP-Code in den  
angegebenen Code umgewandelt werden. Dabei werden  
Zeichen, die im angegebenen Code nicht darstellbar  
sind, in Character-Entities der Form &#xXXXX;  
umgewandelt (z.B. #g+b#g- in &#x03B2;).

## Makrofunktionen zur Anzeige einer Meldung/Anfrage

SET var = DISPLAY (zeile:spalte, farben, kopf, text, {478}  
 button1, button2, button3, ...)

## Makrofunktionen zur Anzeige eines Eingabefeldes

SET var = ASK (zeile:spalte, farben, kopf, text, feld, {479}  
 länge, stab, button1, button2, button3, ...)

SET var = EDIT (zeile:spalte, farben, kopf, feld, {481}  
 breite, höhe, button1, button2, button3, ...)

## Makrofunktionen zur Anzeige eines Auswahlfeldes

SET var = CHOOSE (zeile:spalte, farben, kopf, text, {484}  
 liste, breite, höhe, start, auswahl,  
 button1, button2, button3, ...)

Zeigt ein Fenster mit einer Auswahlliste und wartet, bis eine der Schaltflächen aktiviert (angeklickt), eine Funktionstaste gedrückt oder ENTER oder CANCEL eingegeben wird.

Für die oben angeführten Makrofunktionen DISPLAY, ASK, EDIT, CHOOSE gilt:

Soll der Cursor nach dem Anzeigen des Fensters automatisch auf eine bestimmte Schaltfläche positioniert werden, so muss diese Schaltfläche durch einen Stern vor dem entsprechende Variablennamen markiert werden.

## Lesen einer Texteinheit

READ/IGNORE daten {500}

Liest die Texteinheit, die auf die zuletzt gelesene folgt, unabhängig vom Inhalt der für die laufende Nummer/Satznummer vorgesehenen Variablen

## Schreiben einer Texteinheit / einer Textportion

WRITE/ADJUST daten {499}

erhöht bei Bedarf die Satznummer automatisch. {508}

WRITE/BREAK daten {508}

schreibt keine Daten in die Datei, sondern bewirkt, dass mit der nächsten WRITE-Anweisung ein neuer Satz begonnen wird, so als wäre die Option NEXT angegeben.

MODIFY ACCESS daten NEXT | ADJUST | \* schrittweite {499}

legt fest, um wieviel die Satznummer ggf. für {508}  
 WRITE/NEXT bzw. WRITE/ADJUST bzw. für die einzelnen Zeilen beim Schreiben einer Sternvariablen jeweils erhöht wird.

## Export einer Datei

## DOWNLOAD /PREVIEW

Die Option PREVIEW bewirkt, dass die Datei, {531}  
 die mit #DRUCKE,TYP=WIN-10 bzw TYP=WIN-12  
 beschrieben wurde, nach dem Übertragen  
 angezeigt wird.

DOWNLOAD /BROWSE/BINARY ... {531}

Die Option BROWSE bewirkt, dass die für die  
 Endung des Dateinamens definierte Anwendung  
 nach dem Übertragen gestartet wird

## Definition eines Makrofensters

FIELD feldname: felddtyp feldfarbe {534}

MODIFY FIELD name COLOR feldfarbe {535}

Zu feldfarbe können jetzt 6 statt bisher 4  
 Codes für Farben angegeben werden.  
 Sind mehr als vier Farbwerte angegeben, so  
 wird der normale Text in zeilenweise abwechselnden  
 Farben angezeigt und dabei im Status ON für  
 jede zweite Zeile statt des ersten Farbwertes  
 der fünfte Farbwert und im Status OFF für jede  
 zweite Zeile statt des dritten Farbwertes der  
 sechste Farbwert verwendet.

EDIT[/MIXED\_SORT] {541}

Damit in einem SELECT-Feld der Cursor richtig  
 positioniert werden kann, müssen die Zeilen so  
 sortiert sein, wie sie mit der Makrofunktion  
 ALPHA\_SORT sortiert würden. Sind sie so sortiert,  
 wie sie mit der Makrofunktion MIXED\_SORT sortiert  
 würden (z.B. Dateilisten), muss bei der zu diesem  
 Feld gehörenden EDIT-Anweisung die Option  
 MIXED\_SORT angegeben werden.

FIELD name: BUTTON/CENTER ... {545}

FIELD name: OUTPUT/CENTER ... {547}

Der Feldinhalt wird in der Feldmitte angezeigt.

Parametergesteuerte Programme

## #FAUFBEREITE

Parameter STR: wie bei #KOPIERE (siehe unten) {738}

## #KOPIERE

Parameter STR {796}

Zur Berücksichtigung von tags in / nach getrennten Wörtern kann statt 1 bzw. 2 einer der folgenden Werte angegeben werden:

3 = wie 2, jedoch wird angenommen, dass im nachfolgenden Eingabesatz alle Zeichen bis zum ersten Leerzeichen bzw. bis zur ersten öffnenden spitzen Klammer zum getrennten Wort gehören.

4 = wie 2, jedoch wird angenommen, dass im nachfolgenden Eingabesatz alle Zeichen bis zum ersten Leerzeichen, das außerhalb von spitzen Klammern steht, zum getrennten Wort gehören.

Parameter EEZ: {817}

zweiter Wert = 2: Der Ersetzungstext soll jeweils vor der Zeichenfolge eingesetzt werden.

## #RVORBEREITE

Parameter STR: wie bei #KOPIERE (siehe oben) {898}

## #SVORBEREITE

Parameter STR: wie bei #KOPIERE (siehe oben) {942}

Standardmakros

## #\*DESI

Zur Spezifikation ORT kann jetzt auch STARTLEISTE angegeben werden.

## #\*MBUPDATE

Es können jetzt auch Fremddateien gerettet bzw. geholt werden.

Die bisherigen Angaben zu MODUS wurden aufgeteilt auf die Spezifikationen MODUS und AUSWAHL.

## #\*PS2GHOSTVIEW

Mit diesem Makro kann die Registry so geändert werden, dass BROWSE bei .PS-Datein (wieder) Ghostview aufruft.



Satz

## Allgemeines

Neue Fehlermeldung: Bezüge auf vertikale Positionen, die auf der betr. Seite noch nicht definiert sind, werden mitgezählt; am Ende der Protokoll-Datei wird ggf. eine entsprechende Meldung ausgegeben.

Fußnotenzeilen, die länger als 4000 Zeichen sind, werden in der Protokolldatei angemahnt.

Wenn mehr als 20 Freiräume für eine Seite vorgemerkt werden sollen, wird eine Fehlermeldung ausgegeben.

Pro Seite sind jetzt bis zu 60.000 Zeichen (statt bisher 40.000) zugelassen.

## Parameter

{997}

Zu PARAMETER können jetzt zwei Dateien angegeben werden. Der Inhalt der zweiten Parameterdatei wird nach dem Inhalt der ersten Parameterdatei ausgewertet. Für Reihenfolge und Anzahl der Parameter gelten die gleichen Regeln wie wenn nur eine Parameterdatei angegeben ist.

## Schriften

;" hinter der Schriftnummer einer Schrift mit ungewöhnlich kleiner Versalhöhe bewirkt, dass die Akzente über Versalien um ca. 20% weniger weit nach oben verschoben werden.

{1005}

## Seiten- und Spaltenmontage

SEI, 5. Wert (NRZUS):

8 = Seitennummer zwischen Halbgeviertstriche setzen

9 = 1 + 8

{1016}

## Zeilenumbruch

{1019}

AUS, 2. Wert (IEINRZ): negativer Zahlenwert -n bedeutet, dass Ausgangszeilen, bei denen weniger als n Punkt zur vollen Zeile fehlen, ausgetrieben werden.

AUS, 4. Wert (ISPMIN) kann "-1" sein, um anzugeben, dass Spaltenbreiten auch dann ins Protokoll ausgegeben werden sollen, wenn sie nicht vom Soll abweichen, aber wenigstens 1 Spatium in der Zeile ist und die Zeile weder auf Mitte zentriert noch nach rechts geschoben ist.

## Lebende Kolummentitel

{1023}

KOL, 1. und 3. Wert:

Kolummentitel können an die in der gleichen Zeile stehende Seitennummer herangerückt werden mit der

Angabe 71, 72, ..., 79 (Kolumnentitel oben) bzw. 71 oder 72 (Kolumnentitel unten). Der Kolumnentitel selbst muss dazu linksbündig (mit Zentrieranweisungen für nach rechts geschobene Kolumnentitel) und mit der gleichen Breite gesetzt werden wie die Seitennummer (gegebenenfalls einschließlich deren Verschiebung)

Zwischenüberschriften {1028}

&x& Titelzeilen der Stufe 1.x

Mit &A&...&A&{ bis &W&...&W&{ stehen zusätzlich zu &l&{...&l&{ bis &9&{...&9&{ weitere 23 Titelzeilen der Stufe 1.x zur Verfügung. Es sind die gleichen Angaben wie zu &n& möglich.

Fußnoten {1030}

FN, 1. Wert (IFNLIN) und 8. Wert (IFNLAB):

Ist IFNLAB größer als IFNLIN, so wird der Wert von IFNLAB auch für IFNLIN eingesetzt. Es ist also nicht möglich, die Fußnotenlinie durch Angabe eines größeren Wertes für IFNLAB beliebig nahe an den Haupttext heranzurücken.

Die Fußnotenlinie, die bisher immer aus Linienelementen einer 12-Punkt-Schrift erzeugt wurde, wird jetzt aus Linienelementen mit 4/3 der Grundschriftgröße aufgebaut. Dies gibt vor allem bei großen Schriftgraden eine ausgewogenere Strichstärke. Soll eine andere Schriftgröße m zugrundegelegt werden, so kann zu IFNLNG (9. Wert) n:l bzw. n+m:l angegeben werden.

Wenn die Fußnoten in zwei Spalten unter einspaltigem Text gesetzt werden, werden jetzt Hurenkinder auch in den Fußnoten vermieden.

FNN, 1. Wert (IFNVZ): {1033}

21 bzw. 22: wie 1 bzw. 2, aber Fußnoten, deren Wortlaut identisch ist mit einer auf der Seite bereits ausgegebenen Fußnote, nicht ausgeben; die Fußnoten-Verweisnummern im Text werden angepasst.

Mit der gleichen Wirkung kann auch 31 bzw. 32 statt 11 bzw. 12 angegeben werden.

Makros {1036}

Die Auflösungen der mit den Parametern MAC, MAA, MAH und die in PIs der Form <?tus ...?> enthaltenen Steueranweisungen können zusammen 64000 Zeichen (statt bisher 32000) umfassen.

Der Parameter MAC <!-- > wird nicht mehr erzeugt.

## Steueranweisungen

## Seitenumbruch

&&=n&&{ und &=n&&{ {1046}  
 Spaltenwechsel und Weiterschalten der Spaltennummer  
 auf Spalte n

&!N(r) &!N(R) &!N(a) &!N(A) &!N(\_)  
{1046}  
 schaltet die Seitennummer auf römische (klein),  
 römische (groß) und arabische Ziffern um.  
 Die Seitennummer wird dabei nicht verändert.  
 Mit &!N(\_) wird die Ausgabe der Seitennummer  
 bis zur nächsten Anweisung &!N(r) &!N(R) &!N(a)  
 oder &N(A) unterdrückt.

## Spaltenkopftext

{1052}

&!T: statt &!T{ merkt Spaltenkopftext und setzt  
 ihn gleichzeitig an der Stelle, an der er  
 definiert wird.

&!T\ in Kombination mit &!T= bzw. &!T/

&!T\ immer: Spaltenkopftext für erste Spalte  
 auf linker Seite

&!T= (wenn auch &!T\ definiert ist): Spaltenkopftext  
 für erste Spalte auf rechter Seite

&!T/ (wenn auch &!T\ definiert ist): Spaltenkopftext  
 für letzte Spalte auf rechter Seite.

Zusammen mit &!T\ ist nur eine der Anweisungen  
 &!T= bzw. &!T/ möglich.

## Freiraum

{1060}

\$\$!nnn\$\$ { \$\$!-nnn\$\$ { \$\$\!nnn\$\$ { \$\$!nnn\$\$ { {1061}  
 wie

\$\$nnn\$\$ { \$\$-nnn\$\$ { \$\$\nnn\$\$ { \$\$\nnn\$\$ {  
 aber Vielfache von Grundschrift-Zeilenabständen  
 (auch mit Grafik-Aufruf, z. B. \$\$!nnn#iii\$\$ {)

\$\$mmm/!nnn\$\$ { \$\$-mmm/!nnn\$\$ { wie {1063}

\$\$mmm/nnn\$\$ { \$\$-mmm/nnn\$\$ {  
 aber Vielfache von Grundschrift-Zeilenabständen  
 auch mit Grafikaufruf nach #, z.B. \$\$!mmm/nnn#iii\$\$ {)

\$\$nnn#iii+-eee/aaa\$\$ { {1063}

\$\$nnn#iii+eee/aaa\$\$ {

\$\$mmm/nnn#iii+-eee/aaa\$\$ {

\$\$mmm/nnn#iii+eee/aaa\$\$ {

/aaa gibt an, wie weit Grafiken vom linken Rand  
 eingerückt werden sollen, die aufgrund der Angaben  
 zu +-eee bzw. +eee nicht eingerückt werden.  
 aaa kann auch negativ sein.

- $\$ \$ + ! n n n \$ \$ \{ \$ \$ + - ! n n n \$ \$ \{ \$ \$ + m : ! n n n \$ \$ \{$  wie {1066}  
 $\$ \$ + n n n \$ \$ \{ \$ \$ + - n n n \$ \$ \{ \$ \$ + m : n n n \$ \$ \{$   
 aber nicht Punktzahl, sondern nnn-te Zeile im  
 Grundschriftaster
- Verringern / Vergrößern der Spaltenhöhe {1068}
- $\$ \$ / ! n n n \$ \$ \{ \$ \$ / - ! n n n \$ \$ \{ \$ \$ / + ! n n n \$ \$ \{$  wie  
 $\$ \$ / n n n \$ \$ \{ \$ \$ / - n n n \$ \$ \{ \$ \$ / + n n n \$ \$ \{$   
 aber Vielfache von Grundschrift-Zeilenabständen
- $\$ \$ // n n n \$ \$ \{$  und  $\$ \$ // ! n n n \$ \$ \{$  {1068}  
 verringert eine bereits verringerte Spaltenhöhe  
 um den größeren der beiden zuvor und aktuell  
 angegebenen Werte.
- $\$ \$ / = n n n \$ \$ \{$  Verändert die Höhe des Satzspiegels {1068}  
 für den Rest des Textes.
- Zusätzlicher Durchschuss
- $\&!D-(nn)$   
 Verringern des Durchschuss nach der Zeile, in {1068}  
 der diese Anweisung steht, um nn Punkt (wie  $\&!D-n$ ,  
 aber zweistellig; nn = max. aktueller Zeilenvorschub).
- $\&!D \backslash n$  {1069}  
 Verringern des Durchschuss nach der Zeile, in  
 der diese Anweisung steht, um n/8 Punkt.  
 Im Unterschied zu  $\&!D/n$  wird die Schriftgrundlinie  
 nicht einmalig verschoben, sondern der Durchschuss  
 nach jeder Zeile verringert, in der diese Anweisung  
 steht.
- $\&!D=n$  und  $\&D=(nn)$  {1069}  
 Zusätzlicher Durchschuss von n bzw. nn Punkt  
 vor der Zeile, in der diese Anweisung steht.
- Zeilenumbruch; Spatien
- $\&!B-$  schaltet Blocksatz aus {1073}  
 $\&!B+$  schaltet Blocksatz (wieder) ein
- Silbentrennung {1074}
- Kerning-Anweisungen wie  $\&!(+20)$  oder  $\&!(-10)$   
 werden nicht als Sonderzeichen gewertet (wichtig  
 für die Frage, ob ein Wort mehr Sonderzeichen  
 als Buchstaben enthält und deswegen u.U. nicht  
 getrennt wird).  
 Kommentar mit blanks mitten im Wort führte u.U.  
 zu einer zu späten Silbentrennung.
- Überstreichungen {1094}
- Bei Überstreichungen von kursiven Texten mit #5+ bzw.  
 #6+ wird Linie etwas weiter rechts gesetzt.

## Schriftgrößenänderung {1095}

&!K(nn/l) und &!K(mm:nn/l)

&!K(nn/+l) und &!K(mm:nn/+l)

&!K(nn/-l) und &!K(mm:nn/-l)

Hinter der neuen Schriftgröße kann, durch Schrägstrich von dieser getrennt, jetzt auch eine Veränderung des Zeilenabstandes (absolut oder relativ, ganzzahlig in Punkt) angegeben werden.

## Akzente {1098}

%; steht unter a A E (neu: auch unter u) am rechten Buchstabenrand; ihre Positionierung wurde überarbeitet.

%- über m und w sowie über breiten Versalien wird länger als über normalen Kleinbuchstaben (= alle Buchstaben mit Dicke von mehr als 620/1000)

&!%(nnn) gibt an, ab welcher Buchstaben-Dicke == der Akzent %- über den Buchstaben länger werden soll. Voreinstellung ist 620 (siehe oben).

## Sonderzeichen {1110}

#[2014] wird zu Geviertstrich (bisher: Blockade)

## Makros {1126}

<?:t: xxxx?>

<?tus xxxx?>

Makros ("PI"s) der Form <?:t: xxxx?> und der Form <?tus xxxx?> werden beim Satz durch xxxx aufgelöst, es sei denn, dass sie über Parameter anders aufgelöst sind.

Makros für die Satzumgebung

## #\*GRAFIK {1176}

Das Makro wurde überarbeitet, um auch große EPS-Dateien (mehr als 1 Mio. Zeilen in der Ausgangsdatei) verarbeiten zu können.

## #\*PSAUS {1190}

GRAFIK = name!

Wenn die Grafiken in den zu GRAFIK angegebenen Dateien auf Kopf- und Grafikdatei aufgeteilt sind oder in der 4. Zeile jeder Grafik die zugehörige BoundingBox-Information steht, werden statt der Grafiken Gauraster in der Größe der Grafiken ausgegeben. Grafik-Nummer und -Name werden in das Grauraster eingetragen.

Dies spart beim Testen mit großen Grafikdateien viel Zeit bei `#*PSAUS` und bei der Inspektion mit (z.B.) Ghostview.

`NUMMERN = n`

Zur Spezifikation `NUMMERN` können bis zu 8 (durch Apostroph getrennte) Nummern von Grafiken angegeben werden, die trotz ! hinter dem zu `GRAFIK` angegebenen Dateinamen nicht als Grauraster, sondern als Grafik ausgegeben werden sollen.

`NUMMERN = ALLE`

Trotz ! hinter dem zu `GRAFIK` angegebenen Dateinamen sollen alle Grafiken ausgegeben und nicht als Grauraster wiedergegeben werden.

`NUMMERN = -n`

Bei ! hinter dem zu `GRAFIK` angegebenen Dateinamen sollen nur die (bis zu 8) Grafiken, deren Nummer negativ angegeben ist, als Grauraster ausgegeben werden; die übrigen Grafiken werden unverändert ausgegeben.

`SCHRIFTEN=name`

Zu `SCHRIFTEN` kann zusätzlich zu der Datei mit den Dicktentabellen eine (mit `#*PSKERNFILE` vorbereitete) Datei mit Kerning-Tabellen angegeben werden. Die Kerning-Information wird beim Erstellen der PostScript-Datei berücksichtigt. Die Schriftnummern müssen mit den Schriftnummern in der Datei mit den Dicktentabellen übereinstimmen.

`SATZPROT=prot'protp'protf`

Sollen auch die Fehlermeldungen vom dem Satz der Fußnoten am rechten Rand mit ausgegeben werden, so ist als dritte Datei zu `SATZPROT` die Datei mit den Satzprotokoll des Fußnotensatzes anzugeben.

Die Meldungen des Satzprotokolls können jetzt auch bei zweispaltigem Satz mit ausgegeben werden.

`#*PSKERNFILE,QUELLE=, ZIEL=, LOESCHEN=, NUMMER=` {1209}

Schreibt die Kerning-Information aus dem AFM-File eines PostScript Typel-Fonts in eine Datei, die vom Makro `*PSAUS` ausgewertet werden kann. Zu `NUMMER` ist die Schriftnummer anzugeben, die auch beim Makro `#*PSFONT` für den betreffenden Font gewählt wurde.

`#*PSMONT, ..., PRUEFEN=-` {1210}

Ergibt die Prüfung auf Identität des Vorspanns Abweichungen, so entfällt der Fehlerhalt. Mit `PRUEFEN=-` kann die Prüfung ganz unterdrückt werden (Vorsicht: Sollte nur gewählt werden,

---

wenn sicher ist, dass sich Unterschiede nicht auf die Ausgabe auswirken).

#\*SILKOR {1217}

Trennungen in Wörtern, die in der Silbentrennungsliste Referenz mit Unterscheidungsnummer haben, können nicht korrigiert werden (Fehlerkommentar)

#\*SILLIST {1219}

Mehrfach getrennte Wörter werden als ganze Wörter (bisher: nur die beiden Wortbestandteile vor und nach der Trennstelle) in die Liste aufgenommen. Sie erhalten eine Referenz mit Unterscheidungsnummer.

#\*SILMARKE {1222}

Der Apostroph wird nicht mehr als Worttrenner, sondern (analog zu Akzenten) als Wortbestandteil gewertet.

#\*SATZ

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

\* \* \* \* \*

Corrigenda zum gedruckten Handbuch 2008

Seite 205, letzte Zeile:

Statt "ggf. nach den Daten" muss es heißen:  
"ggf. vor den Daten"

Seite 336, Zeile 20:

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

Seite 464, Seitenmitte, erste Zeile nach "Beispiel:"

Statt |%<| muss es heißen: |%<<|  
also (die ganze Zeile):  
|<%|0|%/|1|%\|2|%<<|3|%:|4|

Seite 1045, 10 Zeile:

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