

Workshop

»Einrichtung des TUSTEP-Arbeitsplatzes
(Tipps und Tricks zur Konfiguration,
Automatisierung und Dateiformaten)«

21. ITUG-Jahrestagung 2014

Vrije Universiteit Amsterdam

Faculteit der Godgeleerdheid

Skript

Matthias Schneider (schneiderm@uni-trier.de)

Kompetenzzentrum für elektronische
Erschließungs- und Publikationsverfahren
in den Geisteswissenschaften/

Trier Center for Digital Humanities

Universität Trier

1. Oktober 2014

www.kompetenzzentrum.uni-trier.de

www.m-schneider.eu

Inhalt

Grundlagen/Ausgangssituation	1
Tipps und Tricks bei der Konfiguration des TUSTEP-Arbeitsplatzes	1
Gestaltung der Benutzeroberfläche	1
Unterstützung bei der Arbeit mit dem Editor	2
Hilfen bei der Texteingabe/Textbearbeitung	2
Punktmakros	2
Syntax-Highlighting	2
Umgang mit Textdaten	2
Tastaturmakros	2
Definition von Funktionstasten	3
Weitere Tipps und Tricks	4
Hilfen bei der Arbeit mit strukturierten Texten (Tags)	4
Menüs (=Maus-/Makroleisten)	4
einfache permanente Makroleiste	4
einfache temporäre Mausmakroleiste	5
Ausführliches Beispiel: MEINFACH.TF	6
Tipps und Tricks für die Ablaufsteuerung und Automatisierung	10
Eingabe von Kommandos direkt im Editor	10
Nutzung von TUSCRIPT zur ereignisabhängigen Ablaufsteuerung	10
Stapelverarbeitung	11
parametrisierter Skriptaufruf mit TUE	11
parametrisierter Aufruf von TUSCRIPT-Makros	12
automatisches Backup	12
E-Mail versenden	13
Tipps und Tricks zum Umgang mit unterschiedlichen Dateiformaten	14
Aufruf von Fremddateien unter Windows mit TUSCRIPT	14
Transformation XML → TXT	15
Informationsquellen/Literatur	16

Grundlagen/Ausgangssituation (vgl. Ott 2000)

- TUSTEP setzt auf Funktionalität und *Portabilität/Plattformunabhängigkeit* → reduzierte graphische Oberfläche und syntaxbasiertes Arbeiten
- Bsp.: #SATZ → Markup statt »What you see is what you get« (WYSIWYG)
- TUSTEP kann für alle Stationen geisteswissenschaftlicher textbasierter Arbeiten genutzt werden. Von der Datenerfassung über die Datenmodifikation bis zur Publikation in Print-, HTML- oder Datenbankform muss die Plattform nicht verlassen werden (*Integration*).
- Die Benutzeroberfläche wird unter verschiedenen Windows-Versionen (XP, Vista, 7, 8), unterschiedlichen Linux-Distributionen inklusive deren Terminals/Shells (aktuell werden 18 Linux-Distributionen getestet, optimale Ergebnisse erzielt die Konsole aus der KDE-Desktopumgebung) sowie Mac OS X betrieben, daher ist das Ausnutzen von betriebssystemseitigen Graphikmöglichkeiten nur eingeschränkt möglich.
- TUSTEP setzt auf einen *professionellen* Nutzer, der sich durch die entsprechende Konfiguration des *modularen* Programmpakets für seine Bedürfnisse eine passende Benutzeroberfläche zusammenstellen kann.
- Die in der default-Einstellung nicht barrierefreie Darstellung der Benutzeroberfläche kann recht leicht barrierefreier gestaltet werden.¹

Tipps und Tricks bei der Konfiguration des TUSTEP-Arbeitsplatzes

Gestaltung der Benutzeroberfläche

- Editorfenster mit den wichtigsten Elementen (TUSTEP-Handbuch, Kapitel »Editor«)

```
DATEN*BEISPIEL
 1.1 | Kurzer Satz in einer Zeile. Es folgt ein
 1.2 | langer Satz, der nicht in eine einzige Zeile
      | passt. Für ihn sind noch zwei Fortsetzungszeilen
      | notwendig. Danach folgt
 1.3 | wieder ein kurzer Satz, der in eine Zeile passt.
      |
      |
*=1.1  Gib Anweisung >
          01;30          SPLIT          INSERT          16:46
```

Es besteht aus

- einer Kopfzeile, in der angezeigt wird, welche Daten ediert werden.
- einem Textfenster mit 6 bis 76 Zeilen, in dem die Daten angezeigt werden.
- einer Meldungszeile, in der z. B. Fehlermeldungen angezeigt werden.
- einer Anweisungszeile, in der die aktuelle Satzposition angezeigt wird und die Anweisungen angefordert werden.
- einer Statuszeile, in der die Cursor-Position, verschiedene Einstellungen und die Uhrzeit angezeigt wird.

¹ <http://tustep.wikispaces.com/Erleichterungen+im+Umgang+mit+TUSTEP>

Unterstützung bei der Arbeit mit dem Editor

Hilfen bei der Texteingabe/Textbearbeitung²

Punktmakros (TUSTEP-Handbuch, Kapitel »Makros aufrufen/definieren/löschen/abfragen«, Makro mit Abkürzungspunkt)

- Definition: $Y, abbr.=INS: |<abbreviation>|$ → fügt $<abbreviation>$ an die aktuelle Cursorposition ein (INSERT MODE)
- Aufruf: Eingabe der Zeichenfolge »abbr« → ALT+P, »abbr« wird zu $<abbreviation>$ aufgelöst
- ALT+Y zeigt alle definierten Punktmakros an, Punktmakros können auch aus diesem Menü heraus eingefügt werden

Syntax-Highlighting (TUSTEP-Handbuch, Kapitel »Colorierung definieren/wechseln/löschen/abfragen«)

- $Cg_n, f_n:xx=Definition$
- Farbauswahl anzeigen: CTRL+F
- $C1, 1:84=|<h1>*</h1>|<i>*</i>|$
- $Cg_n, f_n=$ Farbdefinition löschen
- $Cg_n=$ Farbdefinitionen der gesamten Farbgruppe löschen

Umgang mit Textdaten

- Einfügen von Textbausteinen via Menüs (s. MEINFACH.TF)
- Einfügen von Markup via Mausektionen (s. temporäre Mausleisten)
- Einzelzeichen verschieben ALT+→, ALT+←
- Datensätze verschieben ALT+↑, ALT+↓
- Zeile verdoppeln und einfügen CTRL+B, CTRL+B, Page_Dn
- Eingabe von {...} CTRL+K+erstes Zeichen, z. B. CTRL+K+& → {&}
- Übersicht über die Konventionen der {}-Parameter CTRL+K, Blank

Tastaturmakros (TUSTEP-Handbuch, Kapitel »Makros aufrufen/definieren/löschen/abfragen«, »Tastenkombinationen für Makroaufrufe«, »Tastenkombinationen für Steuerbefehle«, »Steuerbefehle im Editor«)

- Definition von benutzerspezifischen Tastenkombinationen bspw. zur Textdatenmanipulation
- Anzeigen einer Liste mit allen Makros Y
- Definition eines Makros $Y, name=$ makroanweisungen
- alle definierten Makros löschen und Standardeinstellung wieder einstellen $Y, -STD-$

² <http://tustep.wikispaces.com/Editor+-+Tipps>

- Bsp.: `Y, SC_N=INDENT` → `SHIFT+CTRL+N` rückt den aktuellen Datensatz genau so weit vom linken Rand wie den vorangegangenen Datensatz ein
 - Bsp.: `Y, CA_B=BEG_REC, INS:"<bi>", SPLIT, INS:"</bi>, CUR_UP, END_REC` → `CTRL+ALT+B` fügt am Beginn des Datensatzes die Codierung für einen bibliographischen Eintrag ein (`<bi>`), teilt die aktuelle Zeile, fügt in der neuen Zeile die entsprechende Endekennung (`</bi>`) ein und positioniert den Cursor in der ersten Zeile, damit hier gleich weitergeschrieben werden kann
 - Bsp.: `Y, CA_U=CLR_CMD_LINE, "zu, , %, | |", CUR_LE` → `CTRL+ALT+U` löscht den Inhalt der Anweisungszeile, fügt die Zeichenfolge `zu, , %, , | |` ein und positioniert den Cursor
- Funktionalität: Suche String inklusive Umgebung und ignoriere Akzente, Umlaute, u. ä. (%%), s. TUSTEP-Handbuch Kapitel »Ignorieren von Auszeichnungen/Schriftumschaltungen«

Definition von Funktionstasten (TUSTEP-Handbuch, Kapitel »Tastenkombinationen für Funktionsaufrufe«, »Steuerbefehle im Editor«)

Es stehen 60 Funktionstasten zur Verfügung. F1-F12 können über die Funktionstasten auf der Tastatur, F13-F52 außerdem über die kombinierte Verwendung von `Shift`, `CTRL`, `Plus` aus dem Numblock bzw. `CTRL+B` und der jeweiligen Funktionstaste aufgerufen werden. Die Funktionstaste F52 kann beispielsweise mit folgender Tastenkombination aufgerufen werden: `CTRL+B, CTRL+B, CTRL+F12`. Sämtliche Funktionstasten können zusätzlich als Anweisung eingegeben werden. Die Eingabe von F32 ist somit gleichbedeutend mit der Tastenkombination `CTRL+B, Shift+F12`. Die Funktionstasten F1-F12 sind von TUSTEP vorbelegt, sie können jedoch beliebig verändert werden. Alle Funktionstasten können darüber hinaus per Anweisung aufgerufen werden. F18, F29 etc. in der Anweisungszeile ruft also nach Bestätigung mit `Return` die korrespondierende Funktionstaste auf.

- Anzeige aller Funktionstastenbelegungen `F`
 - Definition vornehmen `Fn=definition`
 - Bsp. Anzeige des TUSTEP-Handbuchs mit der Funktionstaste F50 `F50=x #*ZEBE`
 - Bsp. Unterbrechung der Arbeit im Editor und Ausführung der letzten Editor-Datei als Programm mit `TUE F16 = x #T,<EDITOR>`
 - Bsp. Aufruf eines parametrisierten Satzlaufes inkl. Konvertierung der Postscript-Datei in ein PDF (Linux-Funktion)
- ```
F57=x #*SATZ,konfig_wo.tf,MO=satz,PAR=a4ms,AUSF=s'd,
SCH=palatino,OPT=noghost #*ps2pdf,konfig_wo.ps,konfig_wo.pdf
```
- Löschen einer Definition `F50=`
  - Definitionen auf Standardeinstellung zurücksetzen `F=-STD-`

## Weitere Tipps und Tricks

- Lesezeichen/Bookmarks setzen ALT+B
- Lesezeichen aufrufen ALT+I

Achtung! Wenn die Datensatznummern derjenigen Sätze, die mit einem Lesezeichen markiert sind, verändert werden (z. B. mit der Anweisung U!), dann verändert sich auch die Adresse des Lesezeichens.

Hilfen bei der Arbeit mit strukturierten Texten (Tags) (TUSTEP-Handbuch, Kapitel »Suchen und Prüfen von Tags«)

- Tagprüfung (Prüfung auf Wohlgeformtheit) TP
- Tagliste (alphabetische Liste aller Tags) TL
- Tagliste wie TL, aber hierarchisch TLH

Makrofunktionen auf vorgelegten Tastenkombinationen

- Endtag ergänzen ALT+E
- Anfangstag ergänzen ALT+A
- Sprung zum Anfangstag ALT+R
- Sprung zum Endtag ALT+V

Menüs (=Maus-/Makroleisten) (TUSTEP-Handbuch, Kapitel »Makroleisten definieren/löschen/abfragen«)

Mittels Makroleisten besteht die Möglichkeit, benutzereigene Menüstrukturen im TUSTEP-Editor aufzubauen. Diese können für diverse Aufgaben genutzt werden: Einfügen von Textbausteinen oder Code Snippets, Buttons als Ersatz für die Eingabe komplexer Kommandoaufrufe, Änderung von Einstellungen für unterschiedliche Aufgaben (Colorierung, Funktionstasten), Aufruf externer Dokumente (TUSTEP-Handbücher), Einfügen von Datenbankeinträgen...

einfache permanente Makroleiste

- Aufgabe: Anlegen einer permanenten, sequenziellen Übungsdatei ml\_def1.tf
- Konfigurierung einer temporären Makroleiste mit zwei Feldern: HILFEN und DMANAGER.
- Über die Felder sollen das TUSTEP-Handbuch sowie die Satzbeschreibung resp. der TUSTEP-Dateimanager angesteuert werden.

```
#- Datei anlegen
#DA,ml_def1.tf,SEQ-AP
#- Datei öffnen
#E,ml_def1.tf
#- folgende Zeilen sind in die Definitionsdatei
#- einzutragen
#- Definition der Felder <cou>HBUCH</cou>, <cou>SBUCH</cou>
#- und <cou>DMANAGER</cou>, die gem. Syntax einen obligatorischen
```



```
#- Hilfstext enthalten müssen
Y,*=@HBUCH:"Hilfstext",@SBUCH:"Hilfstext",@DMANAGER:"Hilfstext"
 = innerhalb von Definitionsdateien sind Kommentarzeilen
 = wie in dieser Zeile per "=" zu kennzeichnen
 = Beschriftung der Makrofelder:
Y,@HBUCH=hbuch_auf:"Handbuch aufrufen"
Y,@SBUCH=sbuch_auf:"Satzbeschreibung aufrufen"
Y,@DMANAGER=dmanager_auf:"Dateimanager aufrufen"
 = Festlegung der Makrofunktionen, die ausgeführt werden,
 = sobald eines der Felder angeklickt wird
Y,hbuch_auf=CLR_CMD_LINE, "x #*ZEBE",ENTER
Y,sbuch_auf=CLR_CMD_LINE, "x #*ZEBE,SATZMAKRO",ENTER
Y,dmanager_auf=CLR_CMD_LINE, "x #*D",ENTER
- die Makroleiste kann mit folgendem Kommando als Editordefinition eingestellt werden: #E,DE=ml_def1.tf
```

#### einfache temporäre Mausmakroleiste (TUSTEP-Handbuch, Kapitel »Mausaktionen für Makroaufrufe«)

- Aufgabe: anlegen einer permanenten, sequenziellen Übungsdatei ml\_def2.tf
- Konfigurierung einer temporären Mausleiste, die bei der Markierung von Text mit der rechten (!) Maustaste aufgerufen wird.
- Die Mausleiste soll drei Aktionen zur Auswahl anbieten: Text kursivieren, Text auskommentieren, Auswahl ignorieren.

```
#- Datei anlegen
#DA,ml_def2.tf,SEQ-AP
#- Datei öffnen
#E,ml_def2.tf
#- folgende Zeilen sind in die Definitionsdatei
#- einzutragen
= rechte Maustaste gedrückt halten, führt zu einer
= farblich hervorgehobenen Markierung
= Makro für die rechte Maustaste
Y,M_RP=SET_INS,SAVE_CUR,MRK_INI,MRK_CHG:0C
 = Loslassen der rechten Maustaste →
 = markierten Bereich in den Zwischenspeicher und
 = Zwischenspeicher ggf. zuvor löschen; Aufruf der
 = temporären Makroleiste TAGS
Y,M_RR=MRK_REP,SWITCH:C_?TAGS
Y,?TAGS="Bitte Aktion auswählen",
 KURS:"Textstelle kursivieren",
 AKOMM:"Textstelle auskommentieren",
 IGN:"Markierung aufheben"
 = Festlegen der Makrofunktionen
Y,C_KURS="</i>",<EXCH_CUR,"<i>
```

```
Y,C_AKOMM="-->",EXCH_CUR,"<!--"
```

```
Y,C_IGN=MRK_IGN
```

- die temporäre Mausleiste kann mit folgendem Kommando als Editordefinition eingestellt werden: #E,DE=ml\_def2.tf

### Ausführliches Beispiel: MEINFACH.TF<sup>3</sup>

```
#= DEFINITIONEN Definitionen
#= TEXT Textbausteine
#= INFO_TEXT Infotext Textbausteine
#= INFO_HBUCH Infotext Hilfen
#= INFO_CONTACT Kontaktdaten
#= INFO_MAUSS Infotext Mausleiste
#= HILFEN Hilfen
#= MAUSS Mausleiste
#=: DEFINITIONEN Definitionen #- 12.03.14 21:33:18 08.01.14
10:30:11
=
=
= ** MLEIST.TF **
= Diese kurze Makroleiste dient der
= Demonstration des Aufbaus und der
= Funktionsweise einer Nutzer-basierten
= Editor-Makroleiste inkl. temporärer
= Mausleiste.
=
= Eine erweiterte Makroleiste für Produktiv-
= zwecke ist erhältlich auf:
= http://www.tustep.wikispaces.com
= (Suche nach "mleist.tf")
= oder per Mail an: mail@m-schneider.eu
=
= M. Schneider (KompZe Trier), März 2014
= =====
= Definitionsteil (Nennung der
= Schaltflächen mit jeweiliger Farbe)
= =====
Y,*=Text:1F,Mausleiste:8F,Hilfen:47
= =====
= Die für die Makroleiste notwendigen
= Definitionen etc. stehen in
= dieser Datei:
= =====
```

---

3 <http://tustep.wikispaces.com/Editormakros>

```
d=+
= =====
= Festlegen, welche Segmente bei der
= Auswahl welcher Schaltfläche angesteuert
= werden sollen:
= =====
Y,TEXT=SELECT:text
Y,MAUSLEISTE=SELECT:maus
Y,Hilfen=SELECT:hilfen
= =====
= Einzufügende Textbausteine:
= =====
Y,TEXT_SG=INS:"Sehr geehrte Damen und Herren,",SPLIT
Y,TEXT_MFG=INS:"Mit freundlichen Grüßen",SPLIT,"Ihr",SPLIT
Y,TEXT_INFO=DISPLAY:info_text
= =====
= Temporäre Mausleiste
= =====
Y,MAUS_INFO=DISPLAY:info_maus
= =====
= Mausmakros mit rechter Maustaste:
= Klick mit rechter Maustaste -->
= Insert Mode, aktuelle Cursorposition
= wird gemerkt, Markierung eines Bereichs wird begonnen, das
= Markierte farblich hervorgehoben
= =====
Y,M_RP=SET_INS,SAVE_CUR,MRK_INI,MRK_CHG:0C
= Loslassen der rechten Maustaste -->
= markierten Bereich in einen Puffer speichern,
= alten Inhalt des Puffers überschreiben

= temporäre Mausleiste zum Taggen
= anzeigen:
Y,M_RR=MRK_REP,SWITCH:c_?tag2
= Abfragen, was mit dem markierten Bereich geschehen soll:
Y,?TAG2="Markierten Bereich mit Satz-Tags versehen?",
h1:"<h1>":0A,
h2:"<h2>":07,
h3:"<h3>":08,
p:"<p>":4A,
fn:"<fn>":B9,
ital:"<i>":8E,
bold:"":4F,
nix:"Mark. aufh.":0A
= Im Folgenden wird definiert, was zu geschehen hat, wenn
```

Matthias Schneider – Tipps und Tricks zum TUSTEP-Arbeitsplatz

= eines der Felder der temporären Mausleiste angeklickt wird:

Y,C\_H1="</h1>",EXCH\_CUR,"<h1>"

Y,C\_H2="</h2>",EXCH\_CUR,"<h2>"

Y,C\_H3="</h3>",EXCH\_CUR,"<h3>"

Y,C\_P="</p>",EXCH\_CUR,"<p>"

Y,C\_FN="</fn>",EXCH\_CUR,"<fn>"

Y,C\_ITAL="</i>",EXCH\_CUR,"<i>"

Y,C\_NIX=MRK\_IGN

= =====

= Definition von Handbuchaufrufen, Eingabe von

= Sonderzeichen + Anzeige von Kontaktdaten etc.

= =====

Y,HILFEN\_HBUCH=CLR\_CMD\_LINE,INS:"x #\*zebe",ENTER

Y,HILFEN\_IBUCH=CLR\_CMD\_LINE,INS:"x #\*zebe,import",ENTER

Y,HILFEN\_SBUCH=CLR\_CMD\_LINE,INS:"x #\*zebe,satzmakro",ENTER

Y,HILFEN\_CONF=CLR\_CMD\_LINE,INS:"x #\*zebe,config",ENTER

Y,HILFEN\_SUCHE=CLR\_CMD\_LINE,INS:"x #suche",ENTER

Y,HILFEN\_SZEI=SELECT\_CHAR

Y,HILFEN\_INFO=DISPLAY:info\_hbuch

Y,HILFEN\_KONT=DISPLAY:info\_contact

#=: TEXT Textbausteine #=- 12.03.14 21:30:18 08.01.14 10:18:30

TEXT\_SG: Sehr geehrte Damen... einfügen

TEXT\_MFG: Mit freundlichen Grüßen einfügen

TEXT\_INFO: Infotext anzeigen

#=: INFO\_TEXT Infotext Textbausteine #=- 08.01.14 10:17:06

```
|=====INFO-TEXT=====|
| Hier können einige Textbausteine eingebaut |
| werden. |
| Hierzu wird an die betreffende Stelle im Text |
| geklickt, an welcher der Textbaustein eingebracht |
| werden soll. Anschließend wird der Baustein aus |
| diesem Menü ausgewählt. |
| |
| Zum Beenden dieses Infotextes: Leertaste drücken |
|=====|
```

#=: INFO\_HBUCH Infotext Hilfen

#=-

08.01.14 10:26:28

```
|=====INFO-TEXT=====|
| Über das Menü "Hilfen" können die verschiedenen |
| TUSTEP-Handbücher aufgerufen werden: |
| |
| - TUSTEP-Handbuch |
| - Handbuch für den Import/Export RTF-TUSTEP |
| - Handbuch für das Satzmakro |
| |
```

## Matthias Schneider – Tipps und Tricks zum TUSTEP-Arbeitsplatz

```
| Zum Beenden dieses Infotextes: Leertaste drücken |
|=====|
#=: INFO_CONTACT Kontaktdaten #=-
09.01.14 18:15:07 08.01.14 10:34:50
Diese Makroleiste dient als exemplarischer Rahmen
zur vereinfachten, insbesondere
einhändigen, Bedienung des TUSTEP-Editors.
Rückfragen können gerne gerichtet werden an:
#:(08)Matthias Schneider (Trier), mail@m-schneider.eu
#=: INFO_MAUS Infotext Mausleiste #=-
12.03.14 21:35:45 08.01.14 10:29:43
|=====INFO-TEXT=====|
|
| Mit gedrückter rechter (!) Maustaste können Text-
| teile markiert und anschließend mit Tags für den
| Satz versehen werden.
|
| Zum Beenden dieses Infotextes: Leertaste drücken
|=====|
#=: HILFEN Hilfen #=-
12.03.14 21:27:39 08.01.14 10:32:55
HILFEN_HBUCH: #:(0E)Handbuch anzeigen
HILFEN_IBUCH: #:(0E)*Import/*EXPORT-Handbuch anzeigen
HILFEN_SBUCH: #:(0E)*SATZ-Handbuch anzeigen
HILFEN_CONF: #:(0E) "Installation & Konfiguration" anzeigen
HILFEN_SUCHE: #:(0B)Suche in TUSTEP-Beschreibung
HILFEN_SZEI: #:(0A)Sonderzeichen eingeben
HILFEN_INFO: #:(70)Infotext anzeigen
HILFEN_KONT: #:(70)Kontaktdaten anzeigen
#=: MAUS Mausleiste #=- 08.01.14 10:33:23
MAUS_INFO: Infotext zu Mausleiste anzeigen
```

Die Definitionen der MEINFACH.TF sind unter dem oben angegebenen Link abzurufen. Die Datei kann mit dem folgenden Kommando als Editoreinstellung festgelegt werden: #E,DE=meinfach.tf. Die Standardeinstellungen sind mit #E,DE-STD- wiederherzustellen. Im Wiki ist eine wesentlich erweiterte Version der MEINFACH.TF als MLEIST.TF zu finden.

## Tipps und Tricks für die Ablaufsteuerung und Automatisierung

Eingabe von Kommandos direkt im Editor

Gib Anweisung > x #\*D

x → verlasse den Editor und führe das nachfolgende Kommando aus (hier Aufruf des TUSTEP-Dateimanagers)

Nutzung von TUSCRIPT zur ereignisabhängigen Ablaufsteuerung

- Ablaufsteuerung durch die Kombination von Makros und parametergesteuerten Programmen

```
$$! quelle = "quelldatei.rtf", ziel = "zieldatei.xml", isolier = "ja"
$$ MODE TUSCRIPT, {}
```

```
-...
```

```
-...
```

```
IF (isolier .EQ. "ja") THEN
ERROR/STOP OPEN ("isoturb.p", READ, -STD-)
anw = "#T, isoturb.p, , tmp3"
FILE "komm" = anw
ELSE IF (isolier .EQ. "nein") THEN
anw = ""
FILE "komm" = anw
ELSE
ERROR/STOP "Bitte für Variable >isolier< ja oder nein angeben."
END IF
DATA #T, komm
```

- ereignisabhängige Steuerung von Einstellungsdateien

```
$$!
```

```
$$ MODE TUSCRIPT, {}
```

- Datei festlegen, welche die einzustellenden Definitionen enthält  
einstell = "einstell.tf"

```
p_einstell = "p*einstell.tf"
```

- Makrovariablen gleichlautend in TUSTEP-Variablen schreiben,  
- damit Kommando #EDIERE ausgeführt werden kann

```
DEFINE/TUSTEP einstell = einstell
```

```
DEFINE/TUSTEP p_einstell = p_einstell
```

```
IF ((einstell .NE. 'FILE') .AND. (p_einstell .NE. 'FILE')) THEN
```

```
 ERROR/STOP "Die angegebene Datei scheint nicht zu existieren."
```

```
END IF
```

```
IF (einstell .EQ. 'FILE') THEN
```

```
 EXECUTE #E, DE={einstell}
```

```
 PRINT "Einstellungsdatei {einstell} liegt im aktuellen Projekt"
```

```
 PRINT "und wurde definiert."
```

```
ELSEIF (p_einstell .EQ. 'FILE') THEN
```

```
EXECUTE #E,DE=<p_einstell>
PRINT "Einstellungsdatei {p_einstell} liegt im Projekt p"
PRINT "und wurde definiert."
END IF
```

### Stapelverarbeitung, Bsp. TUSTEP → TXT<sup>4</sup>

```
#- kurzes Skript zur Konvertierung mehrerer TUSTEP-Files in
#- einzelne Textdateien.
#- Äquivalent können Textdateien in TUSTEP-Dateien
#- konvertiert werden.
#- temporäre Datei "liste" anlegen
#DA,liste,FR=-
#- Dateinamen in die Datei TUSTEP.EDT schreiben, welche
#- die Zeichenfolge ».tf« als Extension aufweisen.
#LI,DA,-STD-,POS=|.tf{||}
#DA,liste,FR=-
#- Dateiextension wird für die nachfolgende automatische Verar-
#- beitung entfernt
#-
#KO,-STD-,liste,LO=+,PA=*
XX |.tf||
*EOF
#- das, was zwischen #TUE und EOF steht, soll auf die Dateien
#- angewandt werden, die in der Datei "liste" enthalten sind
#TUE,*,SCHL=liste
#- (1) die Dateien werden zum Lesen angemeldet, (2) es werden
#- Textdateien als Ziel angelegt sowie die (3) die
#- TUSTEP-Dateien mit der Zeichencodierung ANSI in die
#- gleichnamige Textdatei exportiert.
#AN,?0.tf
#DA,?0.txt,FDF-AP
#U,?0.tf,?0.txt,CO=ANSI,LO=+
*EOF
```

parametrisierter Skriptaufruf mit TUE (TUSTEP-Handbuch, Kapitel »Ausführen einer Kommandofolge«)

Mit der Parametrisierung von Programmaufrufen können regelmäßig für Standardaufgaben benutzte Skripte ohne Änderung der Programmdateien direkt ausgeführt werden. Für die Platzhalter ?1-?9 werden die beim Programmaufruf angegebenen Werte eingetragen.

---

<sup>4</sup> Unter Stapelverarbeitung wird hier die einheitliche, automatische und sequentielle Verarbeitung mehrerer Dateien mit den gleichen Parametern verstanden. Hiervon ist der Betrieb von TUSTEP im Batch-Modus abzugrenzen. S. hierzu im TUSTEP-Handbuch das Kapitel »Starten von TUSTEP als System-Kommando«.

```
#DE,,*
quelle = ?1
ziel = ?2
#KO,<quelle>,<ziel>,MO=-STD-,LO=+,PA=*
XX 1 |<*>|
*EOF
Programmaufruf: #T,skript.p,,quelldatei.tf'zieldatei.tf
Die Daten der Quelldatei werden in eine Zieldatei kopiert, wobei
alle Tags gelöscht und die Daten der Zieldatei ggf. überschrieben
werden (LO=+).
```

Es können bis zu neun Parameter an beliebiger Stelle in das auszuführende Programm eingebaut werden.

```
Inhalt der Datei s1
#KO,?1,?2,LO=?3,MO=?4,PA=*
XX 1 |aa|bb|
*EOF
#UM,?2,?5,CO=?6,LO=?7
exemplarischer Programmaufruf: #T,s1,,qu'zi'+-STD-'zil.txt'ansi'+
Die Daten der Quelldatei (qu) werden in eine Zieldatei (zi) kopiert, wobei alle Zeichen-
folgen »aa« in »bb« ausgetauscht werden. Außerdem wird die Zieldatei ggf. überschrieben
(+), die Datensatznummerierung wird möglichst nicht geändert (-STD-) und als Zieldatei
für den Export als Textdatei wird im Zeichenformat ANSI die Datei zil.txt angegeben,
deren Inhalt ggf. zu überschreiben ist (+).
```

parametrisierter Aufruf von TUSCRIPT-Makros

```
$$! source="", target=""
$$ MODE TUSCRIPT, {}
txt = FILE (source)
BUILD X_TABLE x_tab1 = *
DATA |<*>|
txt = EXCHANGE (txt, x_tab1)
FILE/ERASE/PRINT $target = txt
Exemplarischer Programmaufruf: $?$makroname,qu,zi
Die Daten der Quelldatei (qu) werden in eine Zieldatei (zi) kopiert, wobei alle Tags
gelöscht werden. Die Daten der Zieldatei werden ggf. überschrieben (FILE/ERASE...).
```

automatisches Backup (z. B. beim Aufruf von Skripten oder in der TUSTEP.INI)

```
#- alle XML/TXT- und TUSTEP-Dateien werden
#- vor Ausführung eines Skriptes auf eine Banddatei
#- gesichert
#- Banddatei zur Sicherung anlegen
#DA,backup.tape,TAPE-AP
```



```
#*MBUPDATE,BA=backup.tape,MO=RE,POS=|.xml|.txt|.tf|
#- exemplarischer Skriptaufruf
#KO,quelle,ziel,,+,*
. . . .
*EOF
```

Für Informationen zum Standardmakro #\*MBUPDATE kann auf Kommandoebene #INF,MBUPDATE genutzt werden. Die angegebenen Dateien werden in die Banddatei gesichert, sofern sie noch nicht oder nur in einer älteren Version in der Banddatei vorliegen.

### E-Mail versenden (TUSTEP-Handbuch, Kapitel »Makrofunktion zum Abschicken einer E-Mail«)<sup>5</sup>

```
$$!
$$ MODE TUSCRIPT, { }
- TUSTEP-Datei für Mailtext anlegen
- Daten in der Datei löschen für den Fall, dass die Prozedur mehrfach
- ausgeführt wird
datei = "textdatei"
ERROR/STOP CREATE (datei, SEQ=0, -STD-)
ERROR/STOP ERASE (datei)

- Datum und Uhrzeit abfragen
datum = DATE(2)
uhrzeit = TIME(1)

- Definition des Mail-Textes als Sternvariable
mailto = *
DATA Hier steht ein Probetext:
DATA Für die Projektdatenbank wurde ein Update
DATA eingespielt:
DATA Datum und Uhrzeit des Updates:
DATA {datum}, {uhrzeit} Uhr
DATA
DATA TUSTEPianische Grüße
DATA Ihr Admin

- der oben stehende Text wird in eine TUSTEP-Datei geschrieben
FILE $datei = mailto

- Variablen werden belegt: Empfänger, Betreff, Textdatei
an = "arbeitsgruppe@uni-xy.de"
wg = "E-Mail-Versand mit TUSCRIPT"
```

---

<sup>5</sup> Diese Makrofunktion ist derzeit nur für Windows implementiert. <http://tustep.wikispaces.com/E-Mail+-versenden>

```
text = FILE ($datei)
```

```
- Makrofunktion zum Mail-Versand
status = SEND_MAIL (an, -, wg, text, -)
```

Das Makro kann unter Windows mit folgendem Kommando aufgerufen werden `??$makroname`.

Das vorstehende Beispielmakro versendet eine Mail mit automatisierten Angaben an eine Mailing-Liste. Sinnvolle Verwendungsmöglichkeiten ergeben sich beispielsweise bei Projekten, in denen mehrere Mitarbeiter an einer Datenbank arbeiten und Informationen über Veränderungen an der Datenbank an die Beteiligten versandt werden sollen. Wird TUSTEP als Serveranwendung betrieben, ist zudem die Generierung von automatischen Nachrichten oder Dokumentauslieferungen denkbar.

```
Vollständige Makrofunktion SEND_MAIL (mailadr, ccadr, subject, text,
files)
```

Als Argumente der Makrofunktion können die Mail-Empfänger, CC-Empfänger, Betreff, die Textdatei sowie anzuhängende Dateien (Anlagen) angegeben werden.

## Tipps und Tricks zum Umgang mit unterschiedlichen Dateiformaten

- mithilfe des TUSTEP-Dateimanagers (`#*D`) können auch Fremddateien (Nicht-TUSTEP-Dateien) angelegt, angemeldet und unter Windows aufgerufen werden<sup>6</sup>

Aufruf von Fremddateien unter Windows mit TUSCRIPT (TUSTEP-Handbuch, Kapitel »Starten und Beenden von Anwendungen«)

```
Satzroutine
Verarbeitung von Fußnoten, Apparaten....
#SA,.....
Aufruf des Satzergebnisses (Postscript-Datei)
#MAKRO
$$ MODE TUSCRIPT, {}
erg = "satzerg.ps"
BROWSE $erg
*EOF
```

Bsp.: Am Ende einer umfangreichen Satzroutine wird das Ergebnis des Satzlaufes (eine Postscript-Datei) über die Makroanweisung `BROWSE` aufgerufen. Die Anweisung `BROWSE` ist derzeit nur unter Windows verfügbar, da auf die Windows-eigene Registry-Funktion zurückgegriffen wird, welche bestimmte Dateiextensionen einem spezifischen Programm zuordnet. Mit der Anweisung `BROWSE` können neben lokalen Dateien auch URLs im Standardbrowser aufgerufen werden.

---

<sup>6</sup> <http://tustep.wikispaces.com/Datei-Manager>

### Transformation XML → TXT

- mit TUSCRIPT können Fremddateien wie XML und TXT direkt ohne #UMWANDLE (=Konvertierung nach TUSTEP) verarbeitet werden
- das nachfolgende Skript liest eine XML-Datei datensatzweise ein und entfernt alle Tags hieraus, um die Daten anschließend als UTF-8 Textdatei abzuspeichern
- zu ergänzen ist ggf. der Umgang mit Zeichencodes wie &#x2019; oder &amp;

```
$$!
$$ MODE TUSCRIPT, {}
source = "xmldatei.xml"
target = "ziel.txt"
status = OPEN (source, READ, -STD-)
status = CREATE (target, FDF-P, -STD-)
BUILD C_GROUP tz = "|?{-}<{+}|"
BUILD X_TABLE xx1 = "|<{00}{C:tz}>||"
ACCESS q: READ/RECORDS/UTF8 $source s, txt
ACCESS z: WRITE/ERASE/RECORDS/UTF8 $target s, a+txt+e
COMPILE
LOOP/999999
 READ/NEXT/EXIT q
 txt = EXCHANGE (txt, xx1)
 WRITE/NEXT z
ENDLOOP
END COMPILE
ENDACCESS/PRINT q
ENDACCESS/PRINT z
```

## Informationsquellen/Literatur

Anzeige der Onlinehilfe: #HilfE auf Kommandoebene oder Anweisung »hilfe« bzw. CTRL+O auf der Editor-/Anweisungsebene

TUSTEP-Beschreibung durchsuchen: #SUCHE

TUSTEP-Handbuch. Quellen: #\*ZEBE (Kommandoebene), »handbuch.pdf« im TUSTEP-Installationsverzeichnis.

TUSTEP Import & Export. Quellen: #\*ZEBE, IMPORT (Kommandoebene), »importexport.pdf« im TUSTEP-Installationsverzeichnis.

Beschreibung des Standard-Makros #\*Satz. Quellen: #\*ZEBE, SATZMAKRO (Kommandoebene), »satzmakro.pdf« im TUSTEP-Installationsverzeichnis.

Installation & Konfiguration. Aufruf. Quellen: #\*ZEBE, CONFIG (Kommandoebene), »config.pdf« im TUSTEP-Installationsverzeichnis.

Bader, Winfried: Lernbuch TUSTEP. Einführung in das Tübinger System von Textverarbeitungsprogrammen, Tübingen 1995.

*Gute Beschreibung von grundlegenden Funktionen und Bedienungsgrundlagen für den Einsteiger. Aufgrund des Alters ergeben sich Abweichungen in der Syntax (<>-Parameter). Die sonstigen Funktionen sind nach wie vor weitgehend unverändert zu verwenden.*

TUSTEP-Wiki: <http://tustep.wikispaces.com/>

*Informationssammlung mit Kurzeinführungen, Programmbeispielen sowie Tipps & Tricks rund um TUSTEP.*

Ott, Tobias: Erste Schritte in TUSTEP: <http://www.tustep.uni-tuebingen.de/tustein.htm>

*Bietet eine Übersicht über grundlegende Befehle und Anweisungen im Editor sowie beim Dateimanagement.*

Ott, Wilhelm: Strategies and Tools for Textual Scholarship: The Tübingen System of Text Processing Programs (TUSTEP), in: Literacy and Linguistics Computing, Jg. 15, Nr 1/2000, S. 93–108.

*Darstellung der Designprinzipien von TUSTEP in Verbindung mit Vorstellung exemplarischer Verwendungsmöglichkeiten.*

Schneider, Matthias; Tobner, René: Einführende Erläuterungen zur Arbeit mit TUSTEP. Importieren und Exportieren von externen Dateiformaten, Tipps und Tricks zum Editor, Trier 2014: <http://tustep.wikispaces.com/Einfuehrungen>.

*Kurze Einführung zu Bedienungsgrundlagen und Einstellungsmöglichkeiten des Editors sowie Datenaustausch zwischen TUSTEP und Fremdprogrammen.*

Stahl, Peter: Tustep für Einsteiger. Eine Einführung in das »Tübinger System von Textverarbeitungs-Programmen«, Würzburg 1996.

*Bietet einen guten Überblick zu grundlegenden Funktionen und Bedienungsoptionen. Aufgrund des Alters ergeben sich Abweichungen in der Syntax (<>-Parameter). Die sonstigen Funktionen sind nach wie vor weitgehend unverändert zu verwenden.*

Homepage von TUSTEP bei der Universität Tübingen: <http://www.tustep.uni-tuebingen.de/>

*Hier kann TUSTEP kostenlos als Open Source für Linux, Mac OS und Windows bezogen werden.*

Homepage der *International TUSTEP User Group*: <http://www.itug.de/>

*Hier kann u. a. die TUSTEP-Mailingliste subskribiert werden, über die Anfragen zu konkreten Problemstellungen sowie TUSTEP- und allgemein DH-Nachrichten sowie Veranstaltungshinweise kommuniziert werden. Des Weiteren sind hier Informationen zu TUSTEP-bezogenen Veranstaltungen und Kursen zu finden.*