

TUSTEP-Editor. Funktionen, die man kennen sollte

4. Oktober 2018

25. ITUG-Jahrestagung, Potsdam

Matthias Schneider (schneiderm@uni-trier.de, www.m-schneider.eu)

Editordefinitionen einrichten

- temporäre Einstellung im TUSTEP-Editor (mit der Möglichkeit, die Definitionen in eine Datei zu überführen)
- Einstellung im Segment EDIT der Datei TUSTEP.INI

Empfehlung: Segment EDIT in der Datei TUSTEP.INI

Empfehlungen für grundlegenden Editoreinstellungen

- Colorierungen
- Punktmakros (insb. für häufig genutzte Textbausteine; auch beim Programmieren oder beim Satz)
- Zeigeanweisungen per Tastaturmakro
- ggf. Mausleiste

Colorierungen

- dienen der visuellen Hervorhebung, z.B. von XML-Tags, inhaltlichen Schlagwörtern oder Satzsteueranweisungen
- 9 Farbgruppen zu je 9 Farbdefinitionen sind möglich
- Neben einfachen Stringfolgen ist die Benutzung von Pattern Matching zur Auswahl der hervorzuhebenden Zeichenfolgen möglich.¹
- Definition:
 - Farbauswahl anzeigen lassen: STRG + F (im Editor)

C1,1=84: |<h1>*</h1>|<i>*</i>|

¹ Vgl. im TUSTEP-Handbuch Kapitel »Editor Organisatorische Anweisungen Colorierung definieren/wechseln/löschen/abfragen«.

C1,1= → löscht die Farbdefinition 1 der Farbgruppe 1
C1= → löscht alle Farbdefinitionen der Farbgruppe 1

Anzeigen der aktuell definierten Colorierungen:

C

Aufgaben:

- Definieren Sie in der Farbgruppe 2 folgende Colorierungen:
 - Alle Überschriften (<h1> + Text + </h1>) sollen in roter Schrift auf schwarzem Grund hervorgehoben werden.
 - Die XML-Tags für Absätze (<p>) sollen in Blau auf weißem Grund erscheinen.

Allgemeine Editormakros²

Editormakros können genutzt werden, um beispielsweise Textbausteine zu schreiben, eine Folge von TUSTEP-Anweisungen oder -Kommandos auszuführen, Editoreinstellungen zu verändern oder Nicht-TUSTEP-Programme aufzurufen. Sie können mit einer Funktionstaste, einer Tastenkombination oder mit der Maus aufgerufen werden.

Y	Anzeigen aller definierten Makros
Y,-STD-	Alle Editormakros auf Standardeinstellung zurücksetzen
Y,SC_N=INDENT	Rückt den aktuellen Datensatz genauso weit vom linken Rand ein wie den vorangehenden. Aufruf: SHIFT+STRG+N.
Y,CA_B=BEG_REC,INS:<bi>“,SPLIT,</bi>“,CUR_UP,END_REC	Fügt am Beginn des Datensatzes die Codierung für einen bibliographischen Eintrag ein (<bi>), teilt die Zeile, fügt in der neuen Zeile die Endekennung ein und positioniert den Cursor am Ende der Zeile, so dass sofort weitergeschrieben werden kann. Aufruf: STRG+ALT+B.
Y,CA_U=CLR_CMD_LINE,INS:<ZU,,%%, “,CUR_LE	Löscht den Inhalt der Anweisungszeile, für die Zeichenfolge ZU, ,%%, ein und positioniert den Cursor um eine Position nach links. Aufruf: STRG+ALT+U.

- Die verfügbaren Steuerbefehle finden sich im Kapitel „Editor/Steuerbefehle im Editor“ im Handbuch.

² Vgl. im Handbuch „Editor/Makros aufrufen/definieren/löschen/abfragen“.

- Vgl. für den Aufruf von Editormakros (insb. die verfügbaren Editormakros) im Handbuch „Editor/Tastenkombinationen für Makroaufrufe“ (s.u.) sowie für die vorbelegten Tastenkombinationen „Editor/Tastenkombinationen (Auswahl)“

ALT+a	A	Shift+ALT+n	SA_n
ALT+x	x	Shift+ALT+9	SA_9
ALT+z	Z		
		Ctrl+ALT+0	CA_0
Shift+ALT+a	SA_A	Ctrl+ALT+n	CA_n
Shift+ALT+x	SA_x	Ctrl+ALT+9	CA_9
Shift+ALT+z	SA_Z		
		Shift+Ctrl+0	SC_0
Ctrl+ALT+a	CA_A	Shift+Ctrl+n	SC_n
Ctrl+ALT+x	CA_x	Shift+Ctrl+9	SC_9
Ctrl+ALT+z	CA_Z		
		Shift+Z0	S_0
Shift+Ctrl+a	SC_A	Shift+Zn	S_n
Shift+Ctrl+x	SC_x	Shift+Z9	S_9
Shift+Ctrl+z	SC_Z		
		Ctrl+Z0	C_0
ALT+0	M_0	Ctrl+Zn	C_n
ALT+n	M_n	Ctrl+Z9	C_9
ALT+9	M_9		
		ALT+Z0	A_0
Shift+ALT+0	SA_0	ALT+Zn	A_n
		ALT+Z9	A_9
		NUM+Z0	N_0
		NUM+Zn	N_n
		NUM+Z9	N_9
		ALT+F1	F_1
		ALT+Fn	F_n
		ALT+F10	F_10
		ALT+F11	F_11
		ALT+F12	F_12

ABBILDUNG 1: AUSZUG AUS DEM HANDBUCH,
"TASTENKOMBINATIONEN FÜR
MAKROAUFRUFE"

Aufgaben:

1. Definieren Sie ein Editormakro, das mit der Tastenkombination STRG+ALT+B aufgerufen wird und folgende Funktionalität umsetzt: Wechsel an den Beginn des Datensatzes, Umschalten in den INSERT MODE und Einfügen einer Stringgruppe nb für „Jedes beliebige Zeichen außer Leerzeichen“.
2. Definieren Sie ein Editormakro, das mit der Tastenkombination STRG+ALT+I aufgerufen wird und folgende Funktionalität umsetzt: Wechsel an den Beginn des Datensatzes, Einfügen von drei Leerzeichen (= auskommentieren von Parametern in parametergesteuerten Programmen), Wechsel in den nächsten Datensatz.
3. Definieren Sie ein Editormakro, das mit der Tastenkombination STRG+ALT+U aufgerufen wird und folgende Funktionalität umsetzt: Wechsel an den Beginn des Datensatzes, Entfernen von drei Leerzeichen, Wechsel in den nächsten Datensatz (= Auskommentierung – s.o. – aufheben).
4. Definieren Sie ein Editormakro, das mit der Funktionstaste F16 aufgerufen wird und folgende Funktionalität umsetzt: Ausführen der aktuell im Editor geöffneten Datei mit #TUE.
5. Definieren Sie ein Editormakro, das mit der Funktionstaste F17 aufgerufen wird und folgende Funktionalität umsetzt: Ausführen der aktuell im Editor geöffneten Datei mit #MAKRO.

Makros mit Abkürzungspunkt (Punktmakros)

Punktmakros sind eine spezielle Art der Editormakros, die z.B. genutzt werden können, um Abkürzungen aufzulösen, Textbausteine oder sonstige Textteile (z.B. Codesnippets, Kommandofolgen usw.) an der Cursor-Position einzufügen. Zur Vermeidung von manuellen Tippfehlern, zur Beschleunigung von sich wiederholenden Eingaben oder auch dem regelmäßigen Aufbau komplexerer Kommandofolgen beim Programmieren ist diese Möglichkeit sehr nützlich:

- Definiert werden die Punktmakros mit einer Anweisung nach dem Muster:³
Y,komm.=INS:“<kommentar></kommentar>“,12*CUR_LE
- Zum Aufruf des Punktmakros wird an der passenden Stelle im Textfeld die Zeichenfolge „komm“ eingegeben oder aufgesucht, der Cursor im oder nach dem Wort platziert und die Tastenkombination ALT+P gedrückt. Daraufhin wird „komm“ zu „<kommentar></kommentar>“ aufgelöst und der Cursor in der Mitte platziert, so dass unmittelbar ein Eintrag vorgenommen werden kann.
- Bsp. 2) HTML-/XML-Kommentar:
Y,ko.=INS:“<!-- -->“,3*CUR_LE

³ Man beachte den Punkt am Ende des Makronamens.

- Bsp. 3) Rahmen für ein #VERGLEICHE:
`Y,VGL.=BEG_REC,INS:"#DE,,*",SPLIT,"VA = ",SPLIT,"VB =
",SPLIT,"*EOF",SPLIT,"#AN,<VA>'<VB>",SPLIT,"#DA,prt001,FR=-
",SPLIT,"#VE,<VA>,<VB>,MO=w,LO=+,PR=prt001",SPLIT,"#DR,prt001,WIN-
10,+",SPLIT,"#DA,prt001.txt,FDF-
AP",SPLIT,"#U,prt001,prt001.txt,CO=ANSI,LO=+",SPLIT,"#+Ergebnisausgabe
vgl. prt001.txt"4`
- Mit ALT+Y kann man sich alle definierten Punktmakros anzeigen lassen. Wählt man durch Anklicken mit der linken Maustaste oder mit den Pfeiltasten und RETURN das beispielsweise definierte Makro „ko.“ aus, so wird das oben geschilderte Makro ausgeführt.

Einfache Makroleisten

Auch wenn in der Standardeinstellung keine Menüleisten im TUSTEP-Editor angezeigt werden, kann der Benutzer zwei Menüleisten (= Makroleisten) nach eigenen Bedürfnissen konfigurieren und mithilfe einer Konfigurationsdatei oder der Datei TUSTEP.INI in den Editor integrieren. Diese können für diverse Aufgaben genutzt werden, z.B. zum Einfügen von Textbausteinen oder Code Snippets, als subsidiäre Buttons für die Eingabe komplexer Anweisungsfolgen, für die Änderung von Einstellungen für unterschiedliche Aufgaben (Colorierung, Funktionstasten) oder für den Aufruf externer Dokumente (TUSTEP-Beschreibungen) oder Webseiten.

- Anlegen einer permanenten Datei für die Definition der permanenten Makroleiste (#DA,ml_def1.tf,SEQ-AP)
- Konfiguration einer Makroleiste mit drei Schaltflächen „Handbuch aufrufen“, „ITUG.de aufrufen“ und „Dateimanager aufrufen“:

```
= Definition der Felder HBUCH , ITUG
= und DMANAGER , die gem. Syntax einen obligatorischen
= Hilfstext enthalten müssen
Y,*=@HBUCH:"Hilfstext",@ITUG:"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,@ITUG=itug_auf:"ITUG.de 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,itug_auf=2*confirm,clear,echo_off,
```

⁴ Auf Grund der Länge muss zur Eingabe der Makrodefinition in der Regel die Anweisungszeile maximiert werden. Hierzu nacheinander auf dem NumPad „+“ und „X“ drücken.

```
|B~#MAKRO|,LF,
|~$$BROWSE "http://www.itug.de"|,LF,
|~*EOF|,ENTER
Y,dmanager_auf=CLR_CMD_LINE, "x #*D",ENTER
```

- Einstellen der Makroleiste: #E,DE=m1_def1.tf

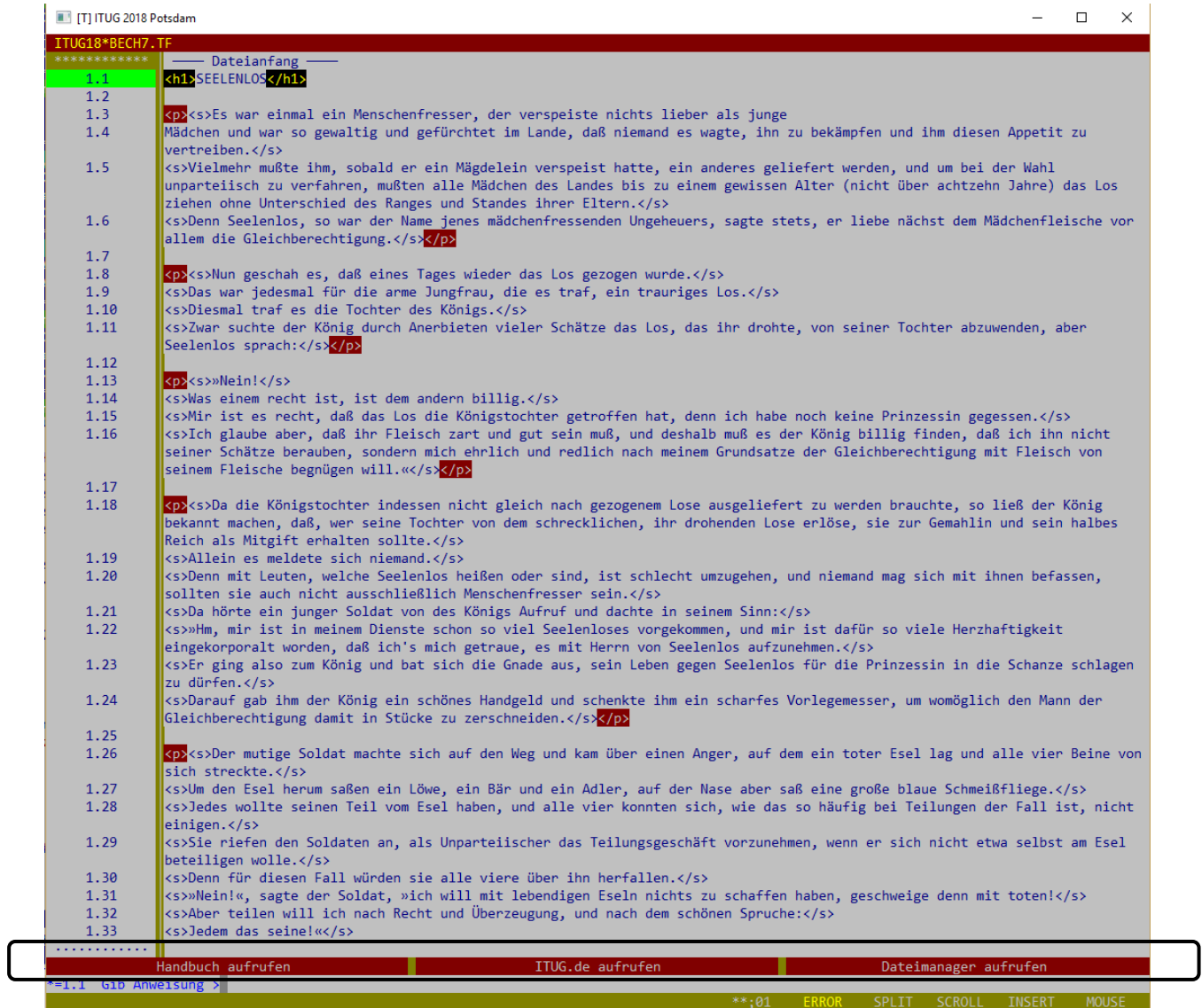


ABBILDUNG 2: SCREENSHOT MIT MAKROLEISTE

Einfache temporäre Mausleisten für Mauseaktionen

- Anlegen einer permanenten Datei für die Definition der permanenten Makroleiste (#DA,m1_def2.tf,SEQ-AP)
- Konfigurierung einer temporären Makroleiste, die bei der Markierung von Text mit der rechten (!) Maustaste aufgerufen wird.
- Die Makroleiste soll drei Aktionen zur Auswahl anbieten: Text kursivieren, Text auskommentieren, Auswahl ignorieren:

```
= 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 #(pfr)  
= 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
```

- Einstellen der Makroleiste: #E,DE=m1_def2.tf
- Beide hier gezeigten Makroleisten können problemlos zusammen genutzt werden.

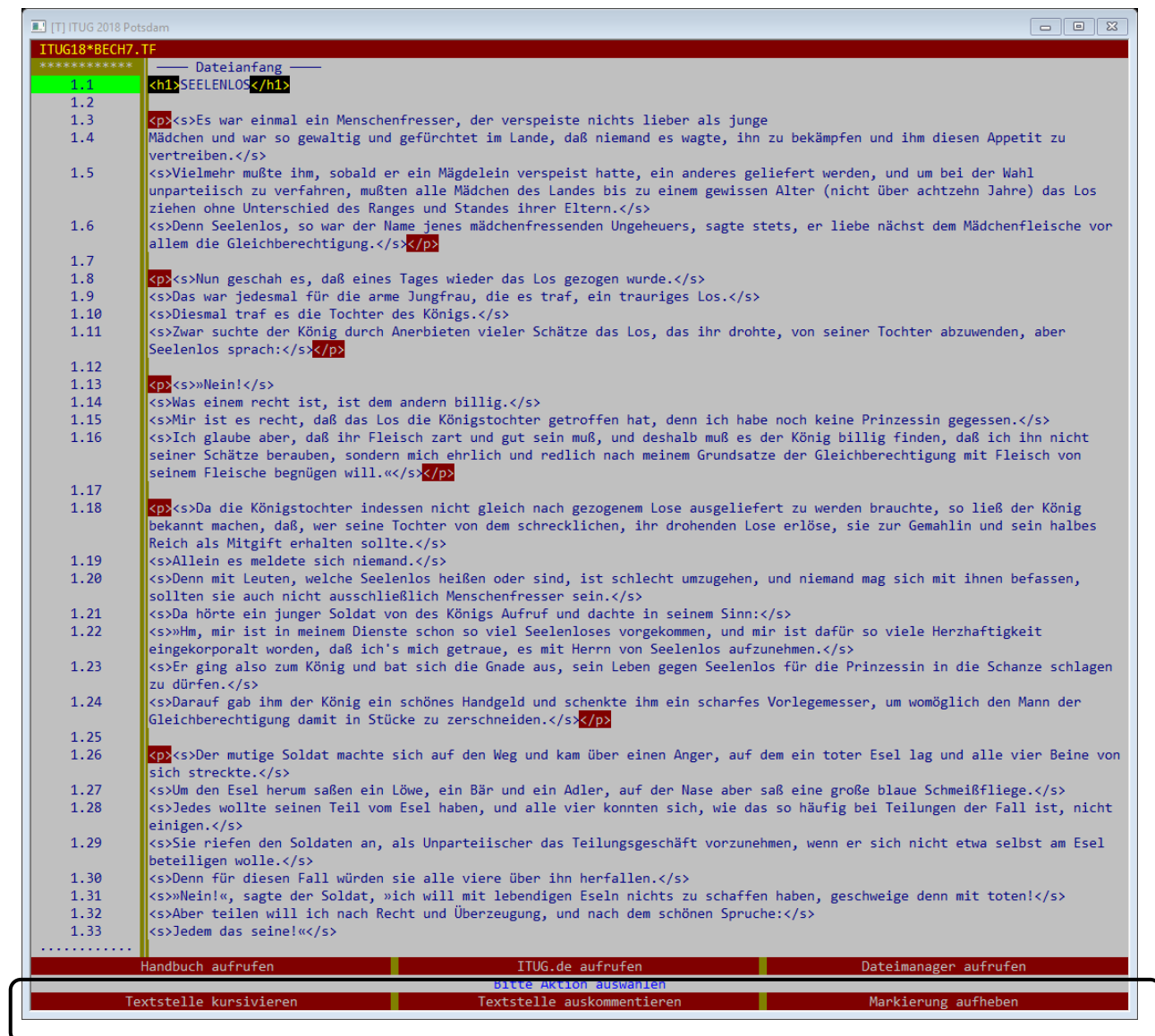


ABBILDUNG 3: SCREENSHOT MIT MAKROLEISTE UND TEMPORÄRER MAUSLEISTE

Als Beispiel für einen umfangreicheren Einsatz von Makroleisten vgl. die Datei mleist.tf:
<http://85.214.95.119/docuwiki/doku.php?id=praesentation:start>.

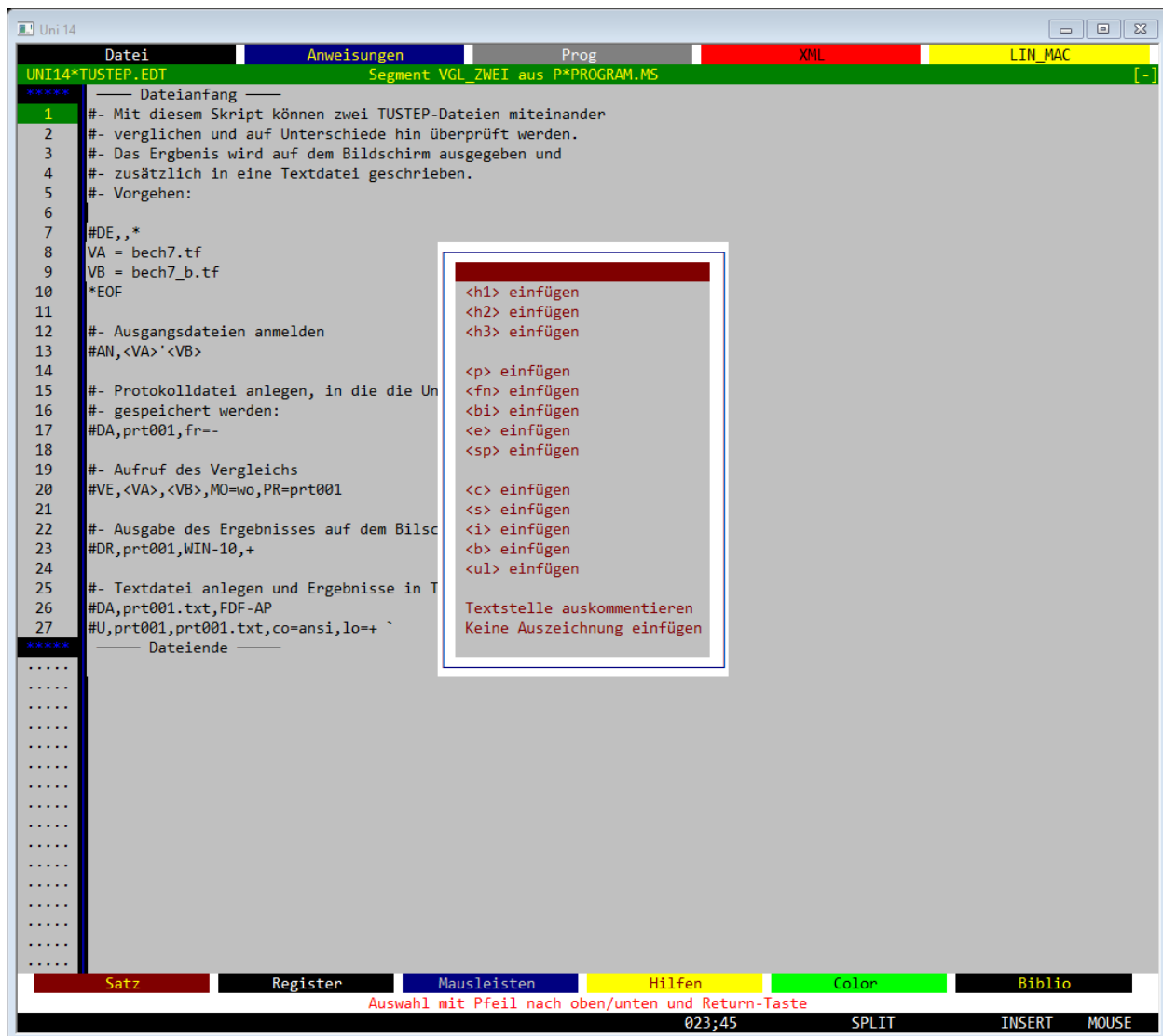


ABBILDUNG 5: SCREENSHOT MIT POPUP ALS MAUSMAKROFUNKTION

Editordefinitionen zurücksetzen/löschen

- #E,DE=+ → Löscht die ggf. im Editor geänderten Definitionen und übernimmt wieder die im Segment EDIT der Datei TUSTEP.INI gespeicherten Definitionen. Gemarkte Editoranweisungen bleiben erhalten, ausschließlich im Editor definierte Zeichen- und Stringgruppen hingegen nicht (= nicht im Segment EDIT vorhanden)
- #E,DE=-STD- → Funktioniert wie #E,DE=+, die gemerkten Anweisungen bleiben erhalten, allerdings werden nicht die Definitionen aus dem Segment EDIT eingestellt, sondern die von TUSTEP voreingestellten Standardwerte. Zeichen- und Stringgruppen aus dem Segment Edit sind damit ebenfalls nicht eingestellt.

- $\#E, DE=!$ → Funktioniert wie $\#E, DE=+$, jedoch werden auch gemerkte Anweisungen etc. gelöscht. Es wird also für den Editor der Stand hergestellt, der für den Beginn einer Sitzung gilt (“neu initialisieren”).