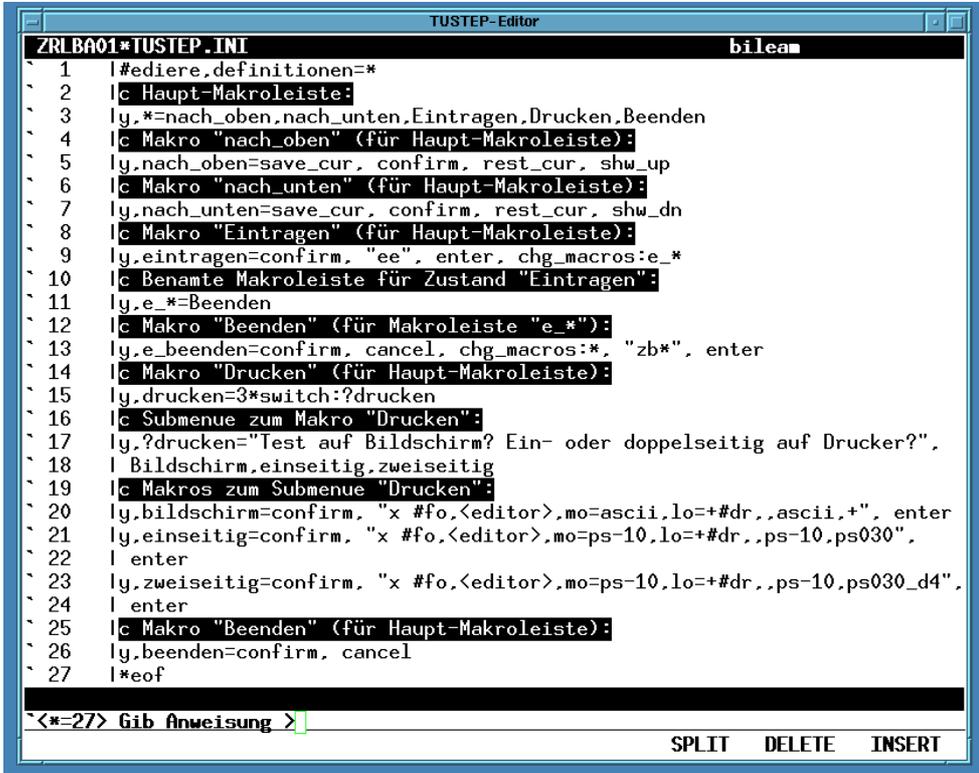


Der TUSTEP-Editor leicht gemacht.

Selbstdefinierte Makroleisten für projektspezifische Anwendungen

Die Verbindung von Maus und Makros macht es möglich, dem TUSTEP-Editor eine leicht zu bedienende projektspezifische Oberfläche zu geben. Dies erweitert die in der letzten BI (95/3+4, S. 9–11) beschriebenen Möglichkeiten, die Kommando-Makros zur Gestaltung projektspezifischer Arbeitsoberflächen bieten, auf die Arbeit im TUSTEP-Editor: ein TUSTEP-Super-User stellt anderen TUSTEP-BenutzerInnen Makros zur Verfügung, die die Arbeit erleichtern und die Einarbeitungszeit erheblich verkürzen. (Zur Verwendung der Maus vgl. auch BI 94/7+8, S. 13.)

Das folgende Beispiel erleichtert die Benutzung von Grundfunktionen im Editor: Blättern, Eintragen von neuem Text, Ausdrucken und Beenden des Editors. Erklärt wird das Zusammenspiel von Maus und Makroleisten. Mit diesem Verfahren lassen sich auch wesentlich komplexere Anwendungen als die hier vorgestellten verwirklichen. Abb. 1 zeigt das TUSTEP-Programm, mit dem Makroleisten und die dazugehörigen Editormakros definiert werden. Die Abb. 2, 3 und 4 zeigen die Sicht von BenutzerInnen bei der Anwendung der Makros.



```
ZRLBA01*TUSTEP.INI                                     bileam
1  #ediere.definitionen=*
2  |c Haupt-Makroleiste:
3  |y,*=nach_oben,nach_unten,Eintragen,Drucken,Beenden
4  |c Makro "nach_oben" (für Haupt-Makroleiste):
5  |y,nach_oben=save_cur, confirm, rest_cur, shw_up
6  |c Makro "nach_unten" (für Haupt-Makroleiste):
7  |y,nach_unten=save_cur, confirm, rest_cur, shw_dn
8  |c Makro "Eintragen" (für Haupt-Makroleiste):
9  |y,eintragen=confirm, "ee", enter, chg_macros:e_*
10 |c Benannte Makroleiste für Zustand "Eintragen":
11 |y,e_*=Beenden
12 |c Makro "Beenden" (für Makroleiste "e_*"):
13 |y,e_beenden=confirm, cancel, chg_macros:*. "zb*", enter
14 |c Makro "Drucken" (für Haupt-Makroleiste):
15 |y.drucken=3*switch:?drucken
16 |c Submenue zum Makro "Drucken":
17 |y.?drucken="Test auf Bildschirm? Ein- oder doppelseitig auf Drucker?",
18 | Bildschirm,einseitig,zweiseitig
19 |c Makros zum Submenue "Drucken":
20 |y.bildschirm=confirm, "x #fo,<editor>.mo=ascii.lo=+#dr,.ascii,+", enter
21 |y.einseitig=confirm, "x #fo,<editor>.mo=ps-10.lo=+#dr,.ps-10.ps030",
22 | enter
23 |y.zweiseitig=confirm, "x #fo,<editor>.mo=ps-10.lo=+#dr,.ps-10.ps030_d4",
24 | enter
25 |c Makro "Beenden" (für Haupt-Makroleiste):
26 |y.beenden=confirm, cancel
27 |*eof

<*=27> Gib Anweisung >
SPLIT DELETE INSERT
```

Abb. 1

Mit der Anweisung `y,*=name1,name2,name3,...` (siehe Abb. 1, Zeile 3) wird eine Makroleiste definiert, die beim Editoraufwurf in der *message line* (drittletzte Zeile) sichtbar ist (siehe Abb. 2). Jeder der angegebenen Namen wird in einem eigenen Feld dieser Makroleiste angezeigt. Bewegt man den Mauszeiger in eines dieser Felder und klickt dort die linke Maustaste, so wird ein Editormakro mit diesem Namen aufgerufen. Für jedes Feld dieser Makroleiste muß also sinnvollerweise ein entsprechendes Editormakro definiert werden. Die Zeilen 5 und 7 in Abb. 1 definieren die beiden Makros `nach_oben` und `nach_unten`

für das Blättern in der Datei. Durch Klick mit der Maus in das Feld `nach_oben` wird das so definierte Makro ausgeführt: Es wird in der Datei (ausgehend von der Zeile, in der der Cursor steht) um einen Bildschirm nach oben geblättert. An der Makroleiste verändert die Ausführung dieses Makro nichts.

Anders verhält es sich beim Makro `eintragen` (Abb. 1, Zeile 9). Es ermöglicht das fortlaufende Eintragen von neuem Text am Ende der Datei. Damit ist der Editor jedoch in einem neuen Zustand, in dem die meisten der in der Makroleiste angezeigten Makros nicht sinnvoll sind bzw. – wie das Makro `beenden` –

eine andere Bedeutung haben müssen. Darum wird eine eigene, benannte Makroleiste definiert mit der Anweisung `y,name*=name1,name2,name3,...` (Abb. 1, Zeile 11, die in diesem Fall nur den einen Feldnamen `Beenden` enthält). Das

Wechseln der angezeigten Makroleiste geschieht in Editormakros mit der Steueranweisung `CHG_MACROS:name*`. So wird im Makro eintragen die Makroleiste gewechselt (Abb. 1, Zeile 9).

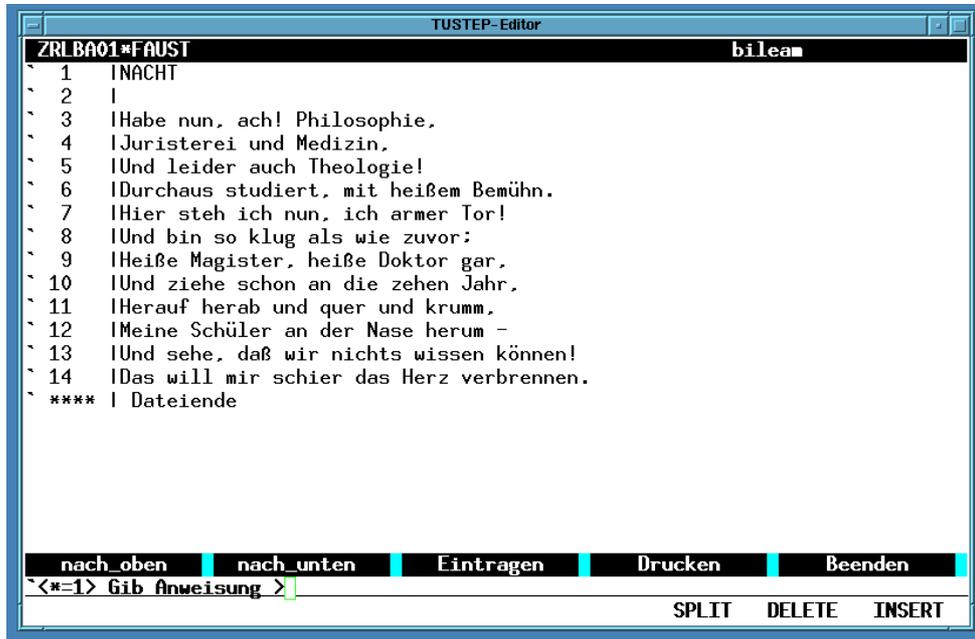


Abb. 2

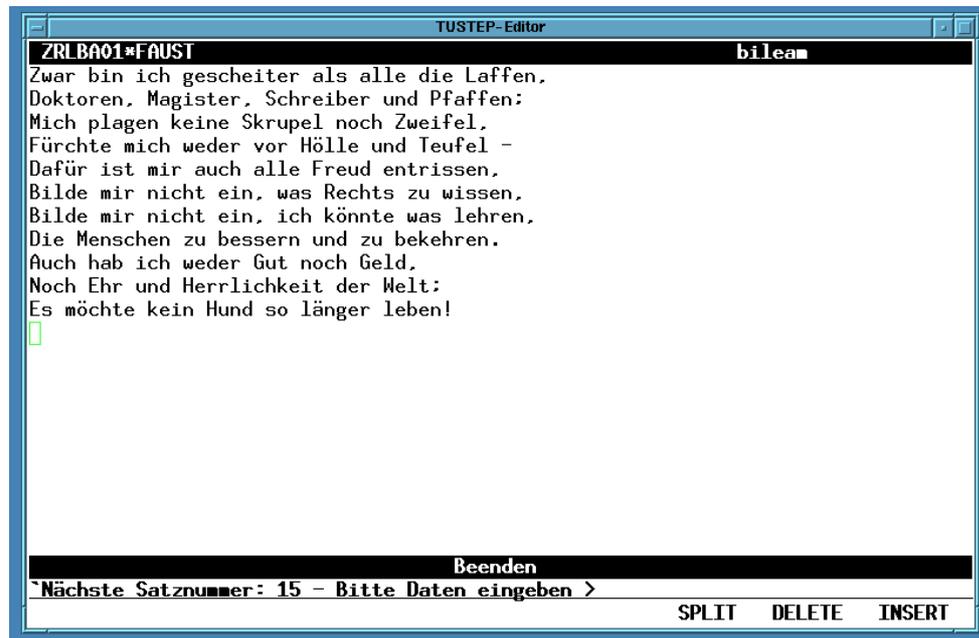


Abb. 3

Das Resultat eines Aufrufs des Makros `eintragen` sieht man in Abb. 3: der Editor ist in einem anderen Zustand (»Bitte Daten eingeben«) und die andere, benannte Makroleiste wird angezeigt. Wird das Feld dieser Makroleiste mit der Maus angeklickt, so wird ein Makro ausgeführt, dessen Namen sich aus dem Namen der Makroleiste und dem Namen des Feldes zusammensetzt, im Beispiel also

`e_beenden`. Dieses Makro wird in Abb. 1 Zeile 13 definiert: das `Eintragen` wird beendet, es wird wieder die unbenannte Makroleiste angezeigt und der Text bis zum zuletzt eingetragenen Satz aufgeblättert. Identische Namen in den verschiedenen Makroleisten (»Beenden«) können so kontextabhängig unterschiedliche (aus BenutzerInnensicht aber intuitiv ähnliche) Bedeutung haben: beim

Mausklick auf die beim Eintragen sichtbare Makroleiste wird das Eintragen beendet und die unbenannte Haupt-Makroleiste angezeigt; wird in dieser Haupt-Makroleiste Beenden angeklickt, so wird das in Abb. 1, Zeile 26 definierte Makro ausgeführt, und der Editor wird beendet.

Das Makro drucken, das von der Haupt-Makroleiste aufgerufen werden kann, veranschaulicht noch einen anderen Mechanismus der Makroleistenprogrammierung: das Anzeigen von Submenues (Abb. 4).

Bei Drucken soll gewählt werden können, ob das Ausdrucken als *preview* am Bildschirm oder auf einseitig oder auf zweiseitig bedrucktem Papier erfolgt. Mit dem Aufruf des Makros drucken kann also nicht sofort die Leistung erbracht werden; sie muß durch eine weitere Eingabe spezifiziert werden. Die Steueranweisung dafür ist SWITCH:?name (Abb. 1, Zeile 15). Damit wird die mit y,?name="Meldung",name1,name2,name3,... (Abb. 1, Zeile 17–18) definierte Meldung in der *command line*

(zweitletzte Zeile; im Beispiel: Test auf Bildschirm? Ein- oder doppelseitig auf Drucker?) und eine Makroleiste mit den entsprechenden Feldern in der *status line* (letzte Zeile; im Beispiel: Bildschirm|einseitig|zweiseitig) angezeigt. In diesem Zustand wird auf genau eine Tasteneingabe (z. B. einen Mausclick) gewartet, die als Makroname interpretiert wird und ein entsprechendes Makro aufruft. Wird mit der Maus ein Feld der in der letzten Zeile angezeigten Makroleiste angeklickt, so wird ein Makro mit dem entsprechenden Namen aufgerufen. (In Abb. 1, Zeile 20–24 sind die Makros zu dieser Leiste für die verschiedenen Druckbefehle definiert.) Wird kein der Eingabe entsprechendes Makro gefunden, so wird ein Warnton ausgegeben und die Anzeige der Meldung und der Makroleiste verschwindet wieder. (Die Angabe 3*switch:?drucken in Abb. 1, Zeile 15 läßt in diesem Zustand drei Fehlversuche zu, bevor das Submenue wieder verschwindet.)



Abb. 4

Das Beispiel sollte zeigen, wie man häufig wiederkehrende Funktionen im Editor für sich und andere mit der Maus erleichtern kann. Wichtig ist zu unterscheiden zwischen den Makroleisten in der *message line* (drittletzte Zeile), die als Angebot dienen, und denen in der *status line* (letzte Zeile), die auf eine Eingabe warten und BenutzerInnen zwingen, entsprechend zu reagieren. Bei konsequenter Ausnutzung dieser Möglichkeit lassen sich ganze Editorsitzungen, z. B. zur Recherche, gestalten,

ohne daß andere als die vom TUSTEP-Super-User vorgesehenen Eingaben akzeptiert werden. Für Sitzungen, in denen Daten erfaßt und korrigiert werden sollen, ist diese restriktive BenutzerInnenführung nicht möglich. Man muß sich darauf beschränken, eine dem Projekt entsprechende Palette von Angeboten bereitzustellen.

Winfried Bader
bader@zdv.uni-tuebingen.de