Arbeitsmenüs für das Editierprogramm TextPad unter Windows

H. Hollatz, B. Truthe

1. Juli 2015

Zusammenfassung

Es werden 6 Arbeitsmenüs mit über 90 Menüpunkten vorgestellt, die das Editierprogramm TextPad bereichern. Zunächst geht es um die Nutzung von TextPad als LaTeX-Editor. Insbesondere wird dabei ein hauptdokumenten-freies Erarbeiten größerer LaTeX-Dokumente vorgestellt. Das 2. Menü betrifft die C/C++-Programmierung; das dritte dient der Arbeit mit normalen Textdateien. Den Abschluß bilden das Konzept einer losen Projektbearbeitung mit einem entsprechenden Arbeitsmenü, ein Menü für globale Aufgaben und eines für mehr private. Es ist leicht möglich, weitere Menüs oder Menü-Punkte hinzuzufügen.

1. Einführung

Die Arbeitsmenüs entstanden aus dem Bedürfnis, für verschiedene Aufgaben den gleichen Editor zu nutzen. Nur wenige Editoren bieten eine geeignete Schnittstelle. So besitzt der Editor `NEdit, www.nedit.org` unter `UNIX` eine eigene Sprache zum Programmieren neuer Aktionen. Unter `WINDOWS` hat der Editor `TextPad, www.textpad.com` eine Schnittstelle, mit deren Hilfe Parameter, deren Werte sich während der Arbeit mit dem Editor ändern, an andere Anwendungsprogramme übergebbar sind.
Es gibt 6 **Arbeitsmenüs**:

```
LaTeX, Language, File, Project, Global, Private.
```

Mit jedem der ersten beiden Menüs ist je eine erweiterbare **Menü-Dateitypliste** verbunden:

```
lst_LaTeX=.tex .sty .cls
lst_Language=.c .cpp .h
```

Das System arbeitet mit **Arbeitsumgebungen**. Mit einer Arbeitsumgebung – Name auf der Umgebungsvariablen `work` – ist eine Menge von Ordnern verbunden; ihre Pfade stehen in einer Ordnerliste, der **Arbeitsordnerliste**; sie ist über die Umgebungsvariable `dirs_File` verfügbar. Der Pfad zum Basisordner wird als Punkt repräsentiert. Der Basisordner muss zur Arbeitsumgebung gehören. Da jeder assoziierte Ordnerpfad als Umgebungsvariable verwendet wird, sind Leerzeichen in einem Ordnernamen oder Dateinamen

unzulässig und führen zu Fehlern; ersatzweise verwende man z. B. den Unterstrich. Es ist zweckmäßig, den Namen des Basisordners als Namen der betreffenden Arbeitsumgebung zu verwenden. Jedem mit einer Arbeitsumgebung assoziierten Ordner ist eine Typliste für Dateinamen zugeordnet, die **Ordner-Dateitypliste**. Mittels der Menü-Dateitypliste werden aus der Arbeitsordnerliste Ordner ausgewählt (**Menü-Ordnerliste**): Ein Ordner aus der Arbeitsordnerliste gehört genau dann zur Menü-Ordnerliste, wenn seine Dateitypliste mit der Menü-Dateitypliste einen Typ gemeinsam hat. Es gibt drei Menü-Ordnerlisten: `dirs_File`, `dirs_LaTeX` und `dirs_Language`. Wenn z. B. die Ordnerlisten lauten

```
dirs_File=. .\LaTeX .\CC
dirs_LaTeX=.\LaTeX
dirs_Language=.\CC
```

so könnten die zugeordneten Ordner-Dateityplisten wie folgt repräsentiert sein:

```
.=.cmd .exe .txt
.\CC=.c .h
.\LaTeX=.sty .tex
```

Bei der Ordnerauswahl innerhalb eines Menüs werden nur die Ordner aus der Menü-Ordnerliste zur Auswahl gestellt, wobei bei wiederholter Selektion innerhalb eines Menü-Aufrufes ein schon einmal ausgewählter Ordner nicht noch einmal zur Auswahl steht. Bei der Dateiauswahl aus einem Ordner werden jene Dateien zur Auswahl gestellt, die den Filter Ordner-Dateitypliste erfolgreich durchlaufen haben und nicht geöffnet sind. Eine angezeigte Auswahlliste wird durch das Kleinerzeichen `<` bzw. `enter` geschlossen, falls man die Auswahl nicht wünscht.
Die Arbeitsmenüs sind über `cmd`-Dateien realisiert. Es werden der Schriftsatz `Lucida Console` und die Codepage 1252 eingestellt. Daher sind bei der Installierung Administratorenrechte erforderlich. Menü-Punkte sind mit Icons in das Menü `Extras` von `TextPad` einbindbar; die eingebundenen Menü-Punkte werden beim Aufruf eines Arbeitsmenüs nicht angezeigt. Bei jedem Aufruf eines Arbeitsmenüs werden die folgenden Parameter in der angegebenen Reihenfolge übergeben:

- Menüpunkt,
- Name der aktiven Datei,
- Zeilen-Nummer der aktuellen Cursor-Position,
- Spalten-Nummer der aktuellen Cursor-Position,
- markiertes Wort.

2. Menü LaTeX

An LaTeX-Dokumenten werden Texte, Artikel, Berichte, Briefe, Bücher und Folien unterstützt. Die einzelnen Dokumentvorlagen stehen in der Datei `LaTeX_Dok.tcl` bzw. in `LaTeX_Doc.tcl`. Jedes Dokument beginnt mit einem Eröffnungsmakro und endet mit dem Makro `\Ende`:

normaler Text: `\Text ... \Ende`

Artikel: `\Artikel ... \Ende`

Bericht: `\Bericht ... \Ende`

Brief: `\Brief ... \Ende`

Buch: `\Buch ... \Ende`

Folien: `\Folien ... \Ende`

Größere Dokumente werden gewöhnlich in Teilen erzeugt. So wird man einen umfangreichen Bericht abschnittsweise erstellen; oft werden die Abschnitte auch von verschiedenen Personen geschrieben. Jeder einzelne Abschnitt ist dann ein relativ eigenständiges Dokument; alle werden durch ein Hauptdokument zusammengeführt. Die Dokumentklassen `Artikel`, `Bericht`, `Text` und `Buch` gestatten das unabhängige Übersetzen einzelner Teile des gesamten Dokumentes, ohne die Teile für das Übersetzen des Gesamtdokumentes ändern zu müssen. Dadurch entfällt die Definition eines Hauptdokumentes.
Beispiel: Es seien bereits zwei Kapitel `k1.tex, k2.tex` eines Buches geschrieben; jedes der beiden Dokumente hat die Struktur `\Buch ... \Ende`. Durch die Datei

```
\input _latex
\Buch
\input k1
\input k2
\Ende
```

wird das aktuelle Gesamtdokument beschrieben.
Hinter dieser Technik steht eine Datei namens `_latex.tex`, die jedem LaTeX-Dokument in der Form `\input _latex` voranzustellen oder in die Datei `latex.ltx` aus der LaTeX-Distribution einzuspeichern ist; im letzteren Falle sind die Format-Dateien zu erneuern. Zu jeder Dokumentumgebung gehört eine `sty`-Datei; z. B. zur Umgebung `\Text` die Datei `tp_Text.sty`. Die zur Umgebung korrespondierende `sty`-Datei enthält die Informationen für den Dokumententyp und sollte in den Arbeitsordner kopiert werden, um sie dort den spezifischen Bedürfnissen anzupassen. Es sei erwähnt, dass für die Dokumente `Artikel`, `Bericht` und `Buch` eine 14pt-Schrift verfügbar ist. Außerdem darf man in deutschsprachigen Texten bei deutscher Tastatur das vollständige deutsche Alphabet (auch Umlaute) verwenden.
Das LaTeX-Menü enthält folgende Menüpunkte:

Open_LaTeX_file: Über die Menü-Ordnerliste ist ein Ordner auszuwählen und aus diesem sind über die Ordner-Dateitypliste und die Menü-Dateitypliste Dateien auszuwählen, die dann gleichzeitig geöffnet werden. Es werden nur die Namen von nicht geöffneten Dateien angezeigt. Falls der Name der aktiven Datei keiner `tex`-Datei entspricht, wird dieser Menüpunkt aufgerufen.

LaTeX_to_dvi: Es wird das Programm `latex.exe` mit der aktuellen Datei aufgerufen, eine `dvi`-Datei erzeugt und diese angezeigt, falls der Menü-Punkt direkt aufgerufen worden ist.

LaTeX_to_pdf: Aus der aktuellen Datei wird eine `pdf`-Datei erzeugt und diese angezeigt, falls der Menü-Punkt direkt aufgerufen worden ist.

LaTeX_from_header: Falls ein anderer Compiler aufgerufen werden soll oder die Datei eine spezielle Bearbeitung mittels Zusatzprogrammen erfordert, ist die abzuarbeitende Befehlsfolge an den Anfang des Dokumentes zu schreiben und mit `\iffalse%%`, `\fi%%` zu umschließen:

```
\iffalse%%
pdflatex.exe -src-specials %file%
musixflx.exe %file%
pdflatex.exe -src-specials %file%
pdflatex.exe -src-specials %file%
%file%.pdf
# -L %work_LaTeX% +rA %file% -system del --end
\fi%%
```

In diesem Falle enthält das Dokument Noten, die durch `musixflx.exe` zu bearbeiten sind. Nach dem Anzeigen des Ergebnisses werden alle Arbeitsdateien gelöscht.

Set_LaTeX_header: Aus einer Menüliste ist eine Befehlsfolge in richtiger Reihenfolge auszuwählen; aus dieser wird eine Befehlsfolge erzeugt und diese an den Anfang des aktuellen Dokumentes geschrieben, um sie als Standard-Befehlsfolge für das aktuelle Dokument zu verwenden.

Error_filter: Aus der `log`-Datei wird eine neue Datei erzeugt und geöffnet; sie enthält die Fehlernachrichten (einschließlich `overfull` und `underfull`).

dvi_to_ps: Erzeugen einer `ps`-Datei aus einer `dvi`-Datei für das aktuelle Dokument.

View: Es wird eine zum aktuellen LaTeX-Dokument gehörende `dvi`-, `pdf`- oder `ps`-Datei ab Basisordner gesucht und angezeigt.

View_error: In der durch `Error_filter` erzeugten Datei ist eine Zeilennummer markiert; der diesen Fehler enthaltene Quelltext wird geöffnet und der Cursor an die Fehlerposition gesetzt.

View_logfile_LaTeX: Anzeigen der `log`-Datei.

BibTeX: Um für das aktuelle Dokument ein Literaturverzeichnis zu erstellen, wird das Programm `bibtex.exe` aufgerufen.

Delete_working_files: Alle durch LaTeX erzeugten Arbeitsdateien für die aktuelle LaTeX-Datei werden gelöscht.

ps_to_ps: Aus der zum aktuellen Dokument gehörenden `ps`-Datei wird eine neue erzeugt, die für geänderten Ausdruck vorbereitet ist. Dabei gibt es folgende Möglichkeiten: Ein- oder zweiseitiger Druck sowie 2, 4, 8 oder 9 Dokumentseiten auf einer Druckseite (bei zweiseitigem Druck gibt es desweiteren die Möglichkeit, als Buch zu drucken).

Edit_documentstyle: Die zum aktuellen Dokument gehörende `sty`-Datei wird in den aktuellen Ordner kopiert und zum Editieren geöffnet.

Make_index: Erzeugen eines Indexregisters (bei einem Buch).

Set_action_sequence: Aus dem LaTeX-Menü darf man folgende Menüpunkte gleichzeitig oder wiederholt auswählen:

`LaTeX_to_dvi`,
`LaTeX_to_pdf`,
`output_size`,
`LaTeX_from_header`,
`Run_musictex`,
`View`,
`Error_filter`,
`Delete_working_files`,
`dvi_to_ps`,
`ps_to_ps`,
`ps_to_pdf`.

Dabei speichert der Menü-Punkt `output_size` die Dokument-Größe an den Anfang der aktuellen Datei in der Form

`% Seitenanzahl prozentuale_Füllung_der_letzten Seite`.

Die gewählte Aktionsfolge wird auf der Umgebungsvariablen `LATEX.menu` gemerkt und bei jedem Aufruf des Menü-Punktes `Execute_action_sequence` abgearbeitet. Beispiel: Man möchte das aktuelle LaTeX-Dokument als `pdf`-Datei in Buchform erstellen:

`LaTeX_to_dvi`,
`LaTeX_to_dvi`,
`dvi_to_ps`,
`Delete_working_files`,
`ps_to_ps`,
`ps_to_pdf`.

Execute_action_sequence: Bei Auswahl dieses Menüpunktes wird die letzte, sich gemerkte Menüpunktfolge wiederholt.

Execute_for_open_documents: Aus den geöffneten LaTeX-Dokumenten dürfen welche ausgewählt werden. Auf jedes ausgewählte Dokument wird die gemerkte Menüpunktfolge angewendet.

Weitere unmittelbare Hilfen beim Erstellen von LaTeX-Dokumenten bieten

- die Datei `LaTeX_Doc.tcl`, Dokument-Umgebungen für englisch-sprachige Texte,
- die Datei `LaTeX_Dok.tcl`, Dokument-Umgebungen für deutsch-sprachige Texte,

- die Datei `LaTeX_Env.tcl`, Textumgebungen,
- die Datei `LaTeX_Mathe.tcl`, mathematische Umgebungen,
- die Datei `LaTeX_Symbols.tcl`, Symbolkodierungen.

Alle diese Dateien sollten im Ordner `USER` von `TextPad` zu finden sein, damit sie beim `TextPad`-Start eingebunden werden.
Die hier genannten Menüpunkte sind nur Beispiele. Ein Blick in die Datei `tp_LaTeX.cmd` sollte beim Erstellen weiterer Menüpunkte hilfreich sein.

3. Menü Language

Die normalen `C/C++`-Compiler sind mit leistungsfähigen Editoren ausgestattet, so dass es hier nicht darum gehen kann, diese zu ersetzen. Vielmehr geht es darum, zusätzliche Entwurfsfunktionen zu kreieren. So wird z. B. mittels der Datei `C.tcl` das Erstellen von Programmen beschleunigt. Natürlich ist dies nur ein Vorschlag. Hier wird der frei verfügbare Übersetzer `Dev-Cpp, http://wxdsgn.sourceforge.net/` verwendet.
Das Menü enthält folgende Menüpunkte:

Open_source_code: Aus einem Ordner der Menü-Ordnerliste dürfen Dateien ausgewählt werden, um sie gleichzeitig zu öffnen. Es werden nur die Namen von nicht geöffneten Dateien angezeigt. Die dabei angezeigte Dateiliste eines Ordners enthält nur Dateinamen mit Endungen, die dem Menü über die Umgebungsvariable `lst_Language` zugeordnet und über die Ordner-Dateitypliste gefiltert sind. Sollte der Name der aktiven Datei keiner `c`- oder `cpp`-Datei entsprechen, wird dieser Menüpunkt aufgerufen.

Compile_source_code: Es wird der `C/C++`-Compiler aufgerufen; im Falle einer `c`-Datei ist es der `C`-Übersetzer (`gcc.exe`) und im Falle einer `cpp`-Datei der `C++`-Übersetzer (`g++.exe`). Enthält die aktuelle Datei kein Hauptprogramm, wird ein Objektmodul erzeugt und im aktuellen Ordner in einer Modulbibliothek (`a`-Datei) abgelegt, die als Grundnamen den Ordnernamen hat. Die in einem Ordner erzeugte Modulbibliothek wird beim Übersetzen jedes Hauptprogramms, das sich in diesem Ordner befindet, dem Linker übergeben.

Compile_from_header: Gelegentlich ist eine dateispezifische Befehlsfolge abzuarbeiten; diese ist am Anfang der aktuellen Datei abzulegen und mit `/*:Win32` und `*/` zu umschließen. Diese Befehlsfolge wird sodann anstelle der Standard-Reaktion abgearbeitet.

View_log_file: Anzeigen der `log`-Datei.

Run_program: Das aus dem aktuellen Sourcecode erzeugte Programm wird nach Abfrage der Aufrufparameter aufgerufen. Falls die aktuelle Datei kein Sourcecode ist, werden alle `exe`-Dateien aus Ordnern der Menü-Ordnerliste zur Auswahl gestellt.

4. Menü File

Das Menü enthält folgende Menüpunkte:

Open_file: Aus einem Ordner der Menü-Ordnerliste sind editierbare Dateien auszuwählen, um sie gleichzeitig zu öffnen. Bereits geöffnete Dateien erscheinen nicht in der Liste. Alle aktuell geöffneten Dateien werden sich mit ihren Cursor-Positionen auf der Umgebungsvariablen `FILE.open` gemerkt.

Open_high: Im aktuellen Text ist ein Dateiname markiert; die entsprechende Datei wird gesucht und geöffnet.

Add_to_project: Die aktuelle Datei wird zum Projekt der aktuellen Arbeitsumgebung hinzugefügt.

New_file: In dem ausgewählten Ordner wird eine neue Datei angelegt, deren Name interaktiv erfragt wird.

Save_as: In einem auszuwählenden, mit dem Menü assoziierten Ordner wird eine Kopie der aktuellen Datei mit dem eingebenen Namen abgelegt.

Copy_file: Über diesen Menüpunkt kann man Dateien eines Ordners in einen anderen Ordner kopieren oder bewegen, gegebenenfalls in einer anderen Arbeitsumgebung.

Create_copy: Im aktuellen Ordner wird eine Kopie der aktuellen Datei angelegt.

File_to_LaTeX: Die aktuelle Datei wird in ein LaTeX-Dokument transformiert, das sowohl als selbständiges Dokument behandelbar ist als auch als Input-Datei für ein anderes LaTeX-Dokument dienen kann.

html_to_txt: Die aktuelle `html`-Datei wird in eine `txt`-Datei transformiert.

File_Converter: `PDFCReator` muss installiert sein und der Pfad auf der Umgebungsvariablen `Path` stehen. Nachdem die zu transformierenden Dateien und der Ausgabe-Typ ausgewählt sind, werden die Transformation ausgeführt.

Delete_associated_files: Löschen von Dateien, die mit der Arbeitsumgebung assoziiert sind.

Delete_other_files: Löschen von Dateien, die nicht mit der Arbeitsumgebung assoziiert sind.

Upload_files: Ausgewählte Dateien werden zu einem -ftp-Server (`UNIX`) gesendet. Die benötigten Daten werden beim Einrichten einer Arbeitsumgebung abgefragt.

Download_files: Ausgewählte Dateien werden vom `ftp`-Server (`UNIX`) geholt; es können nur Dateien geholt werden, die vorher mittels `Upload_files` gesendet wurden.

Create_folder: In einem ausgewählten Ordner werden ein neuer Ordner angelegt, eine Dateitypliste zugeordnet und der neue Ordner mit der aktuellen Arbeitsumgebung verbunden.

5. Menü Project

Ein Projekt ist eine Sammlung von solchen Dateien, die eine wohlbestimmte Einheit bilden und über die aktuelle Arbeitsumgebung erreichbar sind; pro Arbeitsumgebung gibt es ein Projekt. So kann ein `C++`-Programm mit seinen zahlreichen Inputdateien ein Projekt bilden; ein größeres LaTeX-Dokument, wie etwa ein Buch, wird meist aus mehreren Dateien bestehen, die für sich eine gewisse Selbständigkeit haben. Schließlich kann man sich auch eine Lieder-, Rezepte- oder Sprüchesammlung als Projekt vorstellen. Der Projektname stimmt mit dem Namen der aktuellen Arbeitsumgebung überein. Die Namen der zu einem Projekt gehörenden Dateien sind in einer Projektdatei nach Ordnern sortiert abgelegt. Beim Einrichten einer Arbeitsumgebung wird ein entsprechendes Projekt erstellt. Ein Projekt wirkt wie ein Superordner, in dem dateiübergreifende Aktionen ausführbar sind. Das verwendetete Packprogramm `7zip` akzeptiert keine gleichen Dateinamen in verschiedenen Ordnern.
Das Menü enthält folgende Menüpunkte:

Open_project: Die Projektdatei der aktuellen Arbeitsumgebung wird geöffnet.

Open_high_project: In der Projektdatei ist in einer Zeile ein Projektmitglied markiert, das geöffnet wird.

Open_member: Die ausgewählten Projektmitglieder des Projektes werden geöffnet.

Extend: Das Projekt wird interaktiv um neue Dateien erweitert.

Clean_project: Aus dem Projekt werden die Namen aller nicht mehr existierenden Ordner und Dateien gelöscht.

Set_gather_filelist: Aus der Namensliste aller editierbaren, assoziierten Dateien werden Dateien ausgewählt und ihre Namen gemerkt. Danach werden diese zu einer Datei zusammengelegt, um sie zu öffnen. Damit ist es möglich, mehrere Dateien einheitlich zu editieren.

Gather_with_filelist: Die sich zuletzt gemerkten Dateien werden zusammengelegt.

Gather: Aus der Namensliste aller editierbaren, assoziierten Dateien werden Dateien ausgewählt, diese zu einer zusammengelegt, um sie danach zu öffnen.

Expand: Zusammengelegte Dateien werden expandiert.

Set_pack_sequence: Es erscheint eine Menüliste mit folgenden Menüpunkten:

`All_project_files`,
`All_associated_files`,
`Last_packed_file`,
`File_selection_from_project`,
`File_selection_from_working_environment`,
`Create_pack_file`,
`Copy_into_extern_archive`,

`Move_into_extern_archive`,
`Ftp_send`,
`Ftp_send_with_move`,

aus der man auszuwählen hat, welche Packoperationen ausgeführt werden sollen. Die ausgewählten Menüpunkte werden auf der Umgebungsvariablen `PROJECT.Pack` gespeichert und sofort ausgeführt. Es werden `zip`-Dateien unterstützt. Bei der Auswahl ist unbedingt auf die richtige Arbeitsreihenfolge zu achten. Sollte auch der Menüpunkt `Ftp_send` ausgewählt worden sein, erfolgt ein Upload (`UNIX`) der gepackten Datei analog zum Menüpunkt `Upload_files` aus dem Menü `File_menu`.

Execute_pack_sequence: Die sich gemerkten Packoperationen werden ausgeführt.

Pack: Es erscheint eine Menüliste mit folgenden Menüpunkten:

`All_project_files`,
`All_associated_files`,
`File_selection_from_project`,
`File_selection_from_working_environment`,
`Last_packed_file`,

aus der man die Packoperation auszuwählen hat; es werden `zip`-Dateien unterstützt und im Basisordner abgelegt.

Unpack: Nach Auswahl des Ortes, wo sich die zu entpackende Datei befindet, werden ausgewählte Dateien entpackt. Sollte die Umgebungsvariable `ftp_server` definiert sein, steht auch ein `ftp`-Transfer (`UNIX`) der gepackten Datei zur Auswahl.

6. Menü Global

In diesem Menü befinden sich jene Menüpunkte, die globalerer Natur sind. Bei einigen ändert sich die Arbeitsumgebung, so dass ein Neustart von `TextPad.exe` erforderlich ist.

Close_TextPad: Das `TextPad`-Programm wird geschlossen; gleichzeitig werden die Namen der aktuell geöffneten Dateien gemerkt, damit beim Neustart von `TextPad` (mit diesem Menü) diese wieder geöffnet werden können. Falls die Umgebungsvariable `out_archive` gesetzt ist, werden alle während der Sitzung benutzten Arbeitsumgebungen einzeln im angegebenen Ordner als `zip`-Datei gepackt.

Run_program: Aus allen zugeordneteten, ausführbaren Dateien ist eine auszuwählen, um sie auszuführen. Falls das Kommando `%work%.cmd` existiert und ausgewählt wurde, wird es – bis auf den ersten Parameter – mit den gleichen Parameter aufgerufen wie das Kommando `tp_Global.cmd`, also

- Name der aktiven Datei (ohne Pfad),
- Zeilen-Nummer der aktuellen Cursor-Position,
- Spalten-Nummer der aktuellen Cursor-Position,

- markiertes Wort.

Update_working_environment: Die aktuelle Arbeitsumgebung wird aktualisiert. Dazu darf man mehrere unter folgenden Menüpunkten auswählen:

Update_file_type_list: Für ausgewählte Ordner darf die Dateityp-Liste erneuert werden.

Update_variables: Es dürfen Umgebungsvariable gesetzt oder neu definiert werden. Diese gelten dann nur für die aktuelle Arbeitsumgebung.

Clean_working_environment: Über Auswahl-Listen dürfen Dateien gelöscht werden; das zugeordnete Projekt und die Umgebungsvariablen werden entsprechend korrigiert.

Connect_folder: Aus der Ordner-Liste, die nur nicht-zugeordnete Ordnernamen enthält, dürfen Ordner ausgewählt werden; diesen wird eine Dateityp-Liste zugeordnetet.

Disconnect_folder: Die ausgewählten Ordner werden der Arbeitsumgebung und seinem Projekt entzogen.

Change_working_environment: Es wird zu einer anderen Arbeitsumgebung übergegangen und ein Neustart von `TextPad.exe` ausgeführt.

Delete_working_environment: Es dürfen andere Arbeitsumgebungen gelöscht werden.

Create_working_environment: Es darf eine weitere Arbeitsumgebung angelegt werden. Als Name für die Arbeitsumgebung wird der Name des Basisordners empfohlen.

Update_menu: Es wird ein neues Menü erstellt, um das alte, unter `Extras` abgelegte Menü zu ersetzen; danach ist ein Neustart von `TextPad.exe` nötig. Man beachte, dass nach der Auswahl die Menüpunkte in der angegebenen Reihenfolge erscheinen und daher die Anordnung ergonomisch günstig zu wählen ist. Wegen ihrer seltenen Anwendungshäufigkeit sind einige Menü-Punkte aus der Auswahl ausgeschlossen.

Update_root_parameter: Es werden neue Werte für jene Umgebungsvariable abgefragt, die für alle Arbeitsumgebungen gelten; danach Neustart von `TextPad.exe`.

7. Menü Private

In diesem Menü sollte der Nutzer alle jene Menüpunkte sammeln, die mehr spezielleren Charakter haben. Aktuell gibt es im Menü Private die folgenden Menüpunkte.

LaTeX_helpbook: Es werden die `.chm`-Dateien der LaTeX-Hilfe des bekannten LaTeX-Hilfebuches von Petr N. Vabischchevich (siehe `www.latexsoft.com`) zur Auswahl gestellt.

Open_high_LaTeX: Der Menüpunkt arbeitet analog zu `Open_high`; gegebenenfalls wird an den markierten Dateinamen die Endung `.tex` angehängt, also nach einem LaTeX-Dokument gesucht, was zu öffnen ist.

Create_letter: Es soll ein Brief erstellt werden. Dazu ist zunächst ein Briefordner auszuwählen. Falls sich in ihm kein Briefkopf befindet, wird die Datei `tp_Brief.sty` aus dieser Distribution in den Briefordner kopiert und zum Editieren geöffnet. Neue Daten sind anstelle der Hinweise in der Form `<...>` einzutragen. Es wird das Paket `dinbrief` aus der LaTeX-Distribution verwendet. Wenn der Briefkopf im Briefordner existiert, wird ein LaTeX-Dokument im Briefordner angelegt, das die Briefvorlage aus der Datei `LaTeX\tp_Brief.dcl` dieser Distribution enthält; in diesem ist der Brief zu schreiben.

Letters: Dieser Menüpunkt gibt einem Brief eine Addresse und verwendet eine Adressendatei (Dateityp `adr`) aus dem aktuellen Ordner, die Blöcke folgender Struktur enthält:

```
[Sascha]
TITLE=Herrn Geheimrat
NAME=Alexander Tschechow
STR=Puschkinstr. 17
TOWN=39279 Rachmanin
SALUT=Hallo Sascha,
```

Die erste Zeile enthält den Blocknamen (ID-NAME); jede weitere Zeile beginnt mit einem Kennwort, gefolgt von seinem Wert. Jeder Block endet mit einer Leerzeile oder mit einem neuen Block. Für einen Brief sind die obigen Kennwörter nötig. In einem Block dürfen noch weitere Informationen stehen, wie z. B.:

```
PHONE=039245 3805
MOBILE=
EMAIL=Sascha.Tschechow@web.de
INTERNET=http://www.tschechov.ru
OTHERS=geb. 11.11.11
COUNTRY=Niemandsland
```

Dabei ist die Reihenfolge beliebig. Man schreibe nun ein LaTeX-Dokument mit der Vorlage `Brief` aus der Datei `LaTeX\tp_Brief.dcl`; für das Dokument ist ein Name zu wählen, der nicht als Blockname in der Adressendatei auftritt. Beim Aufruf des Menüs wird man aufgefordert, über die Blocknamen aus der Adressendatei Personen auszuwählen, die den aktuellen Brief erhalten sollen. Aus dem einer Person zugeordneten Datenblock werden die benötigten Werte ausgewählt, der aktuelle Brief in eine LaTeX-Datei mit den Adressdaten kopiert und alle Zeilen, die `{<` oder `{>` enthalten, gelöscht. Während des Anzeigens der `pdf`-Datei ist das Druck-Kommando abzusetzen. Man beachte, dass jeder Brief einen Briefkopf hat, der in der Datei

`tp_Brief.sty` gestaltet sein muss. Folgende Philosophie ist empfehlenswert: Gewöhnlich hat man Briefpartner, denen man mit mehr oder weniger leicht modifizierten Briefköpfen schreiben möchte. Folglich richte man für jeden Briefkopf einen Ordner ein, kopiere dahin `tp_Brief.sty` und gestalte den Briefkopf. In jedem dieser Ordner wird man sodann eine Adressendatei über alle jene Briefpartner haben, denen man Briefe mit dem entsprechenden Briefkopf senden möchte.

Sort_adr_file: Die aktuelle `adr`-Datei wird alphabetisch nach den Blocknamen geordnet.

Scratching_of_deleted_files: Gelöschte Dateien sind mit speziellen Programmen rekonstruierbar. Mit diesem Menüpunkt wird der freie Speicherplatz eines Mediums mit zufällig erzeugten Daten überschrieben. Durch mehrmaliges Überschreiben erhöht sich die Sicherheit, dass gelöschte Dateien nicht rekonstruierbar sind.

Create_collection: Aus einer Sammlung von LaTeX-Texten ein Buch als `.tex`-Datei erstellen. Es ist die im obigen Abschnitt Menü LaTeX dargestellte hauptdokumentenfreie Methode zu installieren. Im Basisordner befinden sich eine Titel-Datei `Titel.tex` (mit dem Vorwort), gegebenenfalls eine Literatur-Datei `Literatur.tex` und die zur Sammlung gehörende `.sty`-Datei `%work%.sty`; hier ist `work` der Name der Arbeitsumgebung; standardmäßig ist dies die mitgelieferte Datei `Buch.sty`. Die Kapitel des zu erstellenden Buches sind in Ordnern abgelegt; jeder Kapitel-Ordner `folder` enthält eine LaTeX-Datei `folder.tex`, die die erste Kapitel-Seite mit der Kapitel-Überschrift enthält:

```
\BUCH{\%work\%.sty}\Buch
\hhchapter{Kapitel-Überschrift}
\vspace*{10cm}
\centerline{\Pnb Kapitel-Überschrift}
\newpage
\Ende
```

Im Kapitel-Ordner befinden sich außerdem die zum Kapitel gehörenden Texte als `.tex`-Datei; jede solche Datei hat die folgende Grundstruktur:

```
\BUCH{\%work\%.sty}\Buch
\centerline{\pnb Überschrift}
...
\Ende
```

Der pnb-Befehl erklärt die zu verwendende Schrift und ist in der Datei `%work%.sty` definiert. Dateinamen dürfen keine Umlaute oder Leerzeichen enthalten; alternativ nehme man den Unterstrich für das Leerzeichen. Falls man die Reihenfolge der Texte im Buch weiß, wähle man die Dateinamen so, dass ihre alphabetische Reihenfolge mit der Reihenfolge im Buch übereinstimmt (z. B. durch Nummerierung).

8. Installation

Die Installierung ist als Administrator vorzunehmen, da Änderungen in der `Registry`-Datenbank vorgenommen werden. Dies betrifft die Einstellung auf die Code-Seite 1252 für die Anzeige im Windows-Befehlsscript-Fenster. Diese Code-Seite ist nötig, damit auch Umlaute korrekt behandelt werden.

- Man installiere `TextPad` (www.textpad.com) in `D:\home\Textpad`; dabei sollten sich im Ordner `D:\home` alle jene Ordner befinden, für die Arbeitsbereiche vorgesehen sind. Man achte darauf, dass sich im Pfadnamen kein Leerzeichen befindet. `TextPad` konfiguriere man so, dass beim Aufruf mehrere Dateien zugelassen sind, nur eine Instanz existiert und das Arbeitsverzeichnis dem aktiven Dokument folgt. Man mache durch TextPad-Aufruf erste Erfahrungen; insbesondere registriere man sich.

- Man entpacke `menu_TextPad.zip` in den TextPadt-Ordner; es entstehen insbesondere die neuen Ordner `ins` und `LaTeX`.
  Nun installiere man das Menü wie folgt:
  Das Programm `C:\Windows\System32\cmd.exe` ist mit Administratorenrechten zu starten. Dazu wähle man die Datei aus, drücke die rechte Maustaste und wähle den Menüpunkt „als Administrator ausführen“. Sodann wechsle man im `cmd`-Fenster in den Installationsordner, z. B. `cd /d D:\home\TextPad` und rufe die Datei `ins\tp_first.cmd` auf: `call ins\tp_first.cmd`. Neben den Änderungen in der Registry werden die Arbeitsumgebung für TextPad eingerichtet und ein Standard-Menü installiert.

- Die Menü-Punkte lassen sich über `Update_menu` im Menü `Global` ändern.
  Auf dem Desktop erscheint eine Verknüpfung mit dem Startkommando `tp_ini.cmd`.
  Weitere Arbeitsumgebungen werden im Menü `Global` dadurch erzeugt, dass man den Menüpunkt `Create_working_environment` aufruft.

- Die Datei `tp_ini.cmd` im `TextPad`-Ordner ist das Startkommando für alle Arbeitsumgebungen; beim Aufruf werden der zuletzt geöffnete Arbeitsbereich und jene Dateien geöffnet, die zuletzt geöffnet waren.

- Startet man `TextPad` ohne das Startkommando `tp_ini.cmd` ist das Menu nicht aktiv.

- Über den Menüpunkt `Change_working_environment` im Menü `Global` wechselt man die Arbeitsumgebung.