

WinEdt FAQ

Copyright © 2005 Volunteers of [WinEdt.org](http://www.winedt.org)

2005-11-14

This "FAQ" is growing slowly; you can help to extend it by answering questions that you think are frequently asked and sending them (the questions *AND* the answers) to me: [<georges.schmitz@heitec.de>](mailto:georges.schmitz@heitec.de) or to [<webmaster@winedt.org>](mailto:webmaster@winedt.org).

Frequently Asked Questions

1. General questions

1.1. What is WinEdt?

WinEdt is a powerful, extremely flexible and versatile native editor and shell for MS Windows with a strong predisposition towards the creation of [La]TeX documents (and much more!)

1.2. What is [WinEdt.org](http://www.winedt.org)?

[WinEdt.org](http://www.winedt.org) is a place, where users of WinEdt can share everything concerning WinEdt. This includes macros ([WinEdt Macro Library](#)), modes, menus and shortcut schemes ([Modes and Menus](#)), help (this FAQ and [tutorials](#)), and whatever you can think of and want to contribute. [WinEdt.org](http://www.winedt.org) is run by volunteers. That is, it lives off contributions. To learn more about contributing, read [this](#).

1.3. How do you pronounce WinEdt?

We need it. ;-)

1.4. A message keeps popping up telling me to register. Is WinEdt not free?

No. WinEdt is shareware. That means you can use it for a trial period of 31 days. Any further use requires a license from the author. Registration of a personal copy of WinEdt for educational purposes carries a fee of US \$40. Students are eligible for a reduced rate. WinEdt's current policy entitles registered users to free or competitive upgrades to new releases. For detailed description of different types of licenses and ordering information please check out the [Registration Page](#) on www.winedt.com.

2. TeX

2.1. My LaTeX (dvips, Ghostview, etc.) button doesn't work!

First of all, WinEdt is not a TeX system. Do you have one installed? (see [next question](#) on where to get one).

Secondly, WinEdt needs to know where the applications are. It is usually very good in detecting them, but in rare cases it might be necessary to help it a bit.

Consult the Configuration Wizards (in the Help menu). Especially the output of the Diagnostic Frames is of interest, should you have not be able to solve the problem yourself.

2.2. Where do I get a TeX system?

The most popular TeX implementation for Windows is MikTeX. You can get it at www.miktex.org. Another widely used system is fpTeX, which you can download from www.fptex.org. Both systems are free.

If you are looking for additional packages for TeX or LaTeX, the place to go is CTAN (Comprehensive TeX Archive Network):

- ctan.tug.org (Vermont, USA)
- ftp.tex.ac.uk (Cambridge, UK)
- ftp.dante.de (Mainz, Germany)

Ghostview and Acrobat Reader are also external programs. They can be obtained from www.cs.wisc.edu/~ghost/ and www.adobe.com respectively.

2.3. How do I do ... with TeX, LaTeX, BibTeX or MakeIndex?

You are looking in the wrong place. Since WinEdt is not a TeX system, it is not the cause for your problems. You should have a look in one of the **below mentioned** LaTeX FAQs or ask in a LaTeX related newsgroup.

3. Configuration

3.1. WinEdt Startup and Profiles

3.1.1. Is there a Command Line Switch to start WinEdt with a document and with the caret on a specific line?

WinEdt allows you to enclose any Macro in its command line.

In your case:

```
d:\winedt\winedt.exe document.txt "[GotoLin(12)]"
```

will do the job. The following is equivalent:

```
d:\winedt\winedt.exe "[Open('document.txt');GotoLin(12)]"
```

You may have to specify the full path to `document.txt`. `Macro.chm` in WinEdt's \Doc\Help folder describes all available macro functions.

For example YAP's Inverse Search should be defined as:

```
winedt.exe -F "[Open('%f');SelPar(%l,8)]"
```

YAP substitutes `%f` and `%l` for actual *filename* and *line number*.

3.1.2. How can I set up WinEdt to open only the file that I double click?

When I double click a file with `.tex` extension it opens in WinEdt, but in addition to this file, all other files that were loaded when I last closed WinEdt are also open.

See information in WinEdt Help for *Command Line Switches*, choose the switch you like (presumably `-V` or `-O`), and then implement it under Options|Configuration Wizard...|File-type Associations. In brief the switches have the following meaning:

- V: start WinEdt in Virgin Mode (no project file, no restore of opened files)
- O (=zero!): do not restore previously opened documents



Something to notice:

The Command Switches under Filetype Associations act globally on all files that are double clicked; it is not possible to set a specific switch for only one file type. The startup behavior of WinEdt remains unaffected by these settings if you start the program through its icon on the desktop.

3.1.3. What is it all about with %B and %b and how can I check their values?

%B *Base Directory* (eg. C:\Programme\WinEdt Team\WinEdt). Sort of WinEdt application home, or in other words the install folder.

%b *Ini Path* (eg. C:\Dokumente und Einstellungen\<user>\Anwendungsdaten\WinEdt). The Ini file and other application data are stored in this place.

The values are the same (%b corresponds to %B) unless you create a user profile or manually define a different shortcut with *-E* or *-e* command line switch (see also [Q: 3.1.4](#)). This means: without a user profile, the default settings (and INI file) of the WinEdt installation is used.

WinEdt can give the values of %B and %b in three ways.

1. In version 5.4, you can see the values in the Project Manager Dialog: go to **Project|Project Manager** and than hit the **About** tab.

Executable: The path shown is equal to the value of %B

Ini File: The path shown is equal to the value of %b

2. Execute a macro from the dialog of **Macros|Define and Run Macro**

Enter in the **Command Line**: `Prompt('%b = "%b" %\%%B = "%B" ') ;`

Leave the **Startup Folder** blank.

Press **OK**

3. Go to **Options|Configuration Wizard** and click the button **Shell Configuration**. In the dialog that pops up, choose the tab **User Profiles**. The value of **Application Data Folder** corresponds to %b. With **Browse Install Folder...** you may open the Windows Explorer at the place corresponding to %B.

3.1.4. Can I start WinEdt with the profile of another user?

Yes, you have to start WinEdt with the special command-line switch *-E* with its argument pointing to the WinEdt INI File of the profile of interest. Eg:

```
WinEdt.exe -E="C:\Dokumente und Einstellungen\<user>\Anwendungsdaten\
WinEdt\WinEdt.ini"
```

Where C:\Dokumente und Einstellungen\<user>\Anwendungsdaten\WinEdt corresponds to %b. Best is to manually define a new shortcut to the WinEdt executable with *-E* command line switch. *-e* can be used in a similar way for the main (default) INI file.

If you work under Win2000/XP in a safe mode (NTFS, administrative work as admin, daily work as normal user) the question arises, how usefull *-E* really is, because you have by default no read/write permissions for the profile of other users (except, the admin makes appropriate settings). Of course this was a different story under Win95/98/ME...

3.2. Editing

3.2.1. What does WinEdt do with my lines?

First of all, WinEdt has more than one way to deal with wrapping. Study the documentation file %B\Doc\Wrapping.txt, to learn all about it.

In summary: You can

- turn off wrapping at all (useful for programming);
- use soft return wrapping, i.e. all lines will be wrapped to screen size and line terminators will only be inserted if you explicitly enter them,
- use WinEdt's default wrapping, which will wrap lines to the Right Margin; paragraphs have to be ended by an empty line (useful for writing TeX documents).

Preferences can be changed in Options|Preferences|Defaults (respectively in Document|Document Settings... for documents which were created with other defaults) and in Options|Settings|Wrapping.

3.2.2. I have wrapping on: how can I configure WinEdt for accepting more than 67 characters per line?

Go to Options|Preferences|Editor and change Right Margin (Chars) to a value that suites your needs.

3.2.3. I want to prevent certain lines from being wrapped.

If you are content with the standard way of wrapping (i.e. where paragraphs have to be ended by an empty line), and you only want to exclude certain lines from being wrapped, there are several things you can do:

- Put a % at the end of the line before the one you don't want to have wrapped (just in TeX mode, because % is a comment), or
- indent the line which you do not wish to be wrapped (Format Indented Lines in Document Settings|Properties needs to be unchecked for this to work), or
- if you never want to have lines wrapped which begin with a certain string (f.i. \begin), go to Options|Settings|Paragraph Breaks and append the command to the list on the left-hand side. The same can be done for strings at the end of a line. F.i. you might want to prevent lines from being wrapped if the one before ends with \\. This can be set in Options|Settings|Comments, in the list of Line Breaks.

3.2.4. Is there any option to comment (insert “%”) large parts of a TeX file automatically?

Select the area you wish to comment. You then have 3 possibilities to choose from:

1. unroll the menu Edit|Move Fill|Insert Comment
2. hit the key combination **Shift-Ctrl-Alt-Right**
3. right-click the selected text and choose Insert Comment

If you do not mark any text, only the current line is commented (see also point [Q: 3.2.5](#)).

3.2.5. I'm editing a table exported from another program and I need to append “\” at the end of each line. Is there a simple way to perform this?

You are able to do this in a similar way to case [Q: 3.2.5](#): Select all the lines of your table and go to Edit|Move Fill|Insert String (**Ctrl-Alt-Right**). This opens a small dialog in which you should check the option Append. Now you can enter the text (“\” in your case) and append it at the end of every selected line. As you may anticipate, this works perfectly well with beginning of lines as well while having the Insert option set.

BLOCK Mode: real magic can be done, if you are in BLOCK Mode and you select some character columns. You have then the ability to insert/append strings directly in between the selected block text. For example “ & ” to insert new columns in your TeX table.

3.2.6. When I typed `\cite{}` or `\ref{}` I used to get a popup list of all bibliography items resp. all labels. This doesn't work anymore.

There are different possibilities:

- **WinEdt doesn't find any items.** To check whether WinEdt finds your items, open the Gather page (Project|Gather...) and have a look at the Bib resp. the Label tab. If it's empty, continue with the [next question](#);
- **The Active String that triggers the popup list, is disabled.** Go to Options|Settings|Active Strings and see whether it includes the item `\cite{}`, with the Action:

```
Exe("%B\Macros\Active Strings\Cite.edt");
```

the box before **On Type** checked and in the field thereafter TeX. Also, if you use the feature of automatic parentheses completion, make sure that before this `\cite{}` you find the item `\cite{}`, defined as Relax; and checked. Verify that there is no other `\cite...` which could override the other definition.

- To check whether the macro that triggers the popup list is actually called, open it (`%B\Macros\Active Strings\Cite.edt` resp. `Ref.edt`) and put something like

```
Prompt("Heureka!");
```

in the first line, then try again.

3.2.7. Doubleclicking on “Graphical” in `\section{Graphical User Interface}` selects the whole content between the braces. I just want the word itself being highlighted.

You surely hit exactly “{” (in front of “Graphical”) and what you observed is a feature of WinEdt implemented as Active String. Disable double-click response for {, (and [in Options|Settings|Active Strings Dialog).

3.2.8. I want a list of common spelling mistakes to be automatically corrected. Can I configure the wanted corrections somewhere?

Options|Settings|Active Strings is the right place to define such responses. Check how is by default defined a response to `\lamda` (a common typo) to immediately correct it to `\lambda`. Similar responses are defined but not enabled for `teh` to `the` etc...

Simply duplicate such an instance by right-clicking it and using **Copy** and **Paste**. Adjust the value of the *Active String* and modify the macro fragment in the field below the label **On Type** to your needs (make sure **On Type** is enabled).

3.2.9. The Gather Interface does not show any (or not all) items.

WinEdt usually searches for gather items either in the current file, or, if a Main File is set, in all files of the Project Tree. Therefore the Tree has to be built, if your Project consists of more than one file. To do this, open your main document and select **Project|Set Main File**. Then you can build the Tree in the **Project|Tree** page. Now click **Reset GDI** in the **Project|Gather** page and then **Collect Items**.

If WinEdt still doesn't find any items, you are probably trying to get a list of bibliography entries. By default, WinEdt searches for bibliography items in all opened files, if no Main File is set. So if you don't want to use the Tree feature, you can also get items, if you open the bibliography database (but beware, WinEdt does not check whether any opened databases are actually used to create your bibliography!). So the probably better possibility is to include the bibliography database in the Tree, too. Add something like

```
%GATHER{yourbibfile.bib}
```

to your LaTeX document and rebuild the Project Tree.

3.2.10. Project Tree (in TeX mode) seems to be able to gather information from included/inserted files only if those commands are at the beginning of a new line. Everything works well with a main document like:

```
\input{preamble}
\begin{document}
  \include{chapter1}
  \include{chapter2}
\end{document}
```

whereas with

```
\typeout{Class and Packages} \input{preamble}
\begin{document}
  \typeout{Chapter1} \include{chapter1}
  \typeout{Chapter2} \include{chapter2}
\end{document}
```

nothing gets gathered at all.

You only have to make a very small change to `WinEdtEx.ini` to adjust this behavior to your needs. Goto **Options|Configurations|Extended Configuration**. This opens `WinEdtEx.ini` from your Local WinEdt Path (%b) or from WinEdt's main path (%B) if you haven't defined a *User Profile*. Search for the section [`INPUT_DIRECTIVES`] (at the end of the file) where you find some lines looking as follows:

```
INPUT="TeX"
MODE="TeX"
MAX_TREE_DEPTH=9
OPEN_ON_DBL_CLICK=1
BEGINNING_OF_LINE_ONLY=1
...
```

Change the value of `BEGINNING_OF_LINE_ONLY` to 0 according to the following rules:

- 0 - any position is admissible
- 1 - Beginning of line or indented
- 2 - Beginning of line only

That's it! You need to restart WinEdt first or say **Macros|Run Startup Macro**, to let the changes become effective.

Notice, that gathering large trees this way may slow down the gathering process, as WinEdt needs to examine the files completely. So consider the default value of `BEGINNING_OF_LINE_ONLY=1` being a safety feature and not a bug.

3.2.11. Is it possible to do a word count for a complete TeX project and not only for the current document?

No. However, on CTAN you'll find a utility that produces a word count from the dvi file. This might be just what you are looking for: it can be more accurate than WinEdt because WinEdt does not know how certain TeX macros get expanded while dvi file contains processed text with all the words...

A tool not located on CTAN is **Translator's Abacus**; a Wordcount program that can count the words in html, pdf, Word, RTF, and text file - no DVI file though, but it should not cause any problems to produce an intermediary pdf file (with pdflatex) to do a word count with this program.

3.3. Modes and Submodes

3.3.1. What are so called submodes good for?

Expanded Question: For example, it says that you could use a certain field to associate a dictionary with a submode. But how? Should I write `:FR, TeX:FR, FR`? What if I want it used for more than one mode? When do more than one dictionary apply to the same file?

A submode only affects certain aspect of a document (eg. language). Your document can be TeX, HTML or plain ASCII and FR can apply to all. That's why you should only specify FR (not `:FR`) as a filter to French dictionaries (you may want to have a different user add-on dictionary for French and English as well!).

For example your document could have mode:

```
TeX:FR:UNIX
```

(the last submode indicates that WinEdt detected UNIX-style line terminators - WinEdt and MikTeX work just fine with such files and there is no need to convert them unless you want to use them in other Windows applications)...

3.3.2. What is the deal with the `-- TeX:FR --` comment that is supposed to set the submode of the current document?

This is an emacs convention to specify modes. It works when the file is loaded or when you explicitly execute the `Get Mode` macro in the documents menu. This is one way to force the correct mode...

This works because WinEdt executes the associated macro as an *On-Open* event handler (but you don't have to worry about this just yet unless you want to change or customize this convention).

Later the mode is preserved in WinEdt's Project file...

3.3.3. Which Convention for Submode Definition should be used?

In the WinEdt.org tutorial the submodes are defined `:FR|*. *` while in the existing (default) submodes the definitions are of the type `:DE|*.de`. Which is recommended? Should I manually name the file bla.de, or bla.de.tex or bla.tex.de beforehand or should I open the Document Settings and choose the submode and save and then the .de will be attached automatically?

That depends on what you want. Some users want to adopt double-extension (file-type) convention to ensure the proper mode and submode. Others prefer emacs convention above and yet others use Document Settings dialog and the fact that the modes are preserved as a part of the project file...

3.4. Dictionaries

3.4.1. How can I disable Spell Checking?

Simply click into the status bar field **Spell**. The color should change from dark red to grey and Spell Checking is disabled.

If you want to fine tune Spell Checking, you should also consider to check the settings in the panes under Options|Dictionary....

3.4.2. Mode dependant Dictionary

Expanded Question: I want to have a French TeX submode, with a French dictionary. I managed to define a submode and to load the dictionary, associating it with the submode. Unfortunately, now, when I activate my FR submode with the Document Settings dialog, the spell checker considers both English and French words as correct. I would like, when in FR submode, French words to be correct, while English ones be recognized as incorrect. How can I do this?

Well, one way is to also define EN submode (beside the FR). And then enable the English word lists ONLY for EN submode.

Or else, you can enable English dictionaries for all modes except FR and DE (should you use both French and German plus English). This is done by the following mode filter:

```
* | FR;DE
```

LaTeX Dict can be enabled for all modes (or else just for Tex). If you want to enter more than one (sub)mode in the "enabled" filter list the modes separated by semicolon:

```
model;mode2;mode3 etc...
```

If the list is followed by

```
|notmodel;notmode2;notmode3 etc...
```

then these modes indicate the modes for which the option is disabled (eg. * | FR). This applies to all (sub)mode-filters.

Depending on your default (most commonly used language) it is a good (but not the only possible) strategy to enable the associated dictionaries with the mode filter:

```
* |second_lang1;second_lang2; etc...
```

and have secondary languages treated by assigning such documents a submode:

```
second_lang1, etc...
```

3.4.3. When I want to browse some dictionary from the dictionary manager, the selected dictionary doesn't open.

Browse here means *browse the disk for new files to include* not *browse inside the files themselves*. If you want to edit or examine the file, then you have to open it manually (which may, of course, be done by right clicking on the file while in the above browsing mode).

3.5. User Interface

3.5.1. I want to add a new menu item to the Accessories Menu. How do I do that?

All menu customizations are performed through the Options|Menu Setup dialog.

- First you have to choose the menu in which you want the new command to go: Double-Click on the Accessories item.

- Example:

or call a batch file:

which in turn would call the application.



Make sure that Windows knows where the utility is located. I.e. its folder must either be part of Windows' PATH environment variable or you have to specify the full path (possibly using a variable like %B, which stands for WinEdt's base directory). If you are having problems, open a DOS window and make sure that the command line itself is correct.)

- or [all in one line]

For further explanations of these macros, consult **Help|Macro Manual** (in the past documentation was in %B\Doc\Macros.txt (WinEdt 5.3 and early 5.4 versions)).

- The full documentation for WinEdt's menus can be found in WinEdt's Help (look for index entry `Menu Setup`).

3.5.2. How can I install a shortcut to a macro

Suppose you have the following macro `... \macro\foo.edt` you want to tie up with a shortcut **Ctrl-Shift-F** without adding a *menuitem*.

To establish this shortcut, go through **Options|Menu Setup** to the tabbed pane **Main Menu**. **Doubleclick** the item **Shortcuts** and move to the end of the **Menu Items** list. **Rightclick** and insert a new item via **Insert|Macro**. Insert `Exe (' ... \macro\foo.edt ') ;` in the **Macro** field, go to the **Shortcut** field and insert your desired shortcut by simply hitting the required keys **Ctrl-Shift-F**.

If you need a key like **Esc** you may need the help of **More Keys**.

3.5.3. How can I add a button to the toolbar?

You can only add icons to the toolbar, that are available as such in menus, so maybe you have to define them first there (invisible menus or popups are possible too). If you want to add a "shortcut" to an external program (via toolbar), you must be aware of WinEdt not being able to evaluate the internal icon definition of programs - you have to extract it yourself.

To publish a new icon in WinEdt's menus, you have to do the following steps:

1. Copy the new icon (best is a 21x20 truecolor BMP file!) into `... \WinEdt\Bitmaps\Buttons`.
2. Open `WinEdt.btn` and add a new line before `-1 The End Of Bitmaps (300 maximum)` as follows: `<ddd> %B\Bitmaps\Buttons\<xxx>.bmp`", where `<ddd>` is a number that is not occupied by existing icons.
3. Restart your WinEdt and try.

If you want a nice icon to appear in menu items too, proceed the same way:

16x16 truecolor BMP files
directory becomes `... \WinEdt\Bitmaps\Images`
and the file for editing becomes `WinEdt.img`

These icons can be referenced under **Options|Menu Setup...|Main Menu**. Double click a specific menu item, and add your new icons under **Images** and hint.

Finally - this all done - just right click the **Tool Bar** with the mouse and choose **Enter Tool Bar Setup** from the popup menu. Then choose the button you want to move (click 1) and click on the place where you want it to appear (click 2). Save the modification with right click and choosing **Apply** in the context menu.

See also [Q: 3.5.1](#) and of course WinEdt's Help (look for index entry **Menu Setup**).

3.5.4. How can I add a custom document format (a template) to **Document|New Document?**

First save your template eg. in `%B\local\templates`. Then generate a file link to this document with the help of **Options|Preferences|Templates**, where you may browse for your file in question. Look at the examples (it is straightforward).

3.5.5. The shortcuts I defined for a popup menu don't work.

This is the way Windows handles menus: If the popup menu is made visible then the shortcuts assigned to its items are not valid and you have to either use the mouse interface, arrows or the underlined letters in the item names (the characters preceded by **&**) to choose an item.

On the other hand, an invisible menu items can only be accessed through double shortcuts and you must assign a shortcut to all its items and use these shortcuts to activate any particular item. The underlined letters are irrelevant in such menus just as the assigned shortcuts are irrelevant in the visible popup menus!

See also [installing a shortcut](#).

3.5.6. Why are buttons sometimes clickable, sometimes not?

Clickable simply indicates the presence of a possible input file for the related command, e.g., the `dvips` button will be clickable if `filename.dvi` exists, where `filename` is the name of the open (or main) TeX/LaTeX file.

So, if the buttons are not clickable, it means that the relevant file was not created by the compilation command - i.e., an error occurred.

3.5.7. The toolbar buttons are enabled all of the time (i.e., they do not grey out when files at my E: drive are not present).

This is a feature which prevents WinEdt from becoming too slow on removable or network drives. If `E:` is a local (or fast enough) drive you can add it to Update Drives in the Preferences|Backup Dialog. Enter something like:

```
C:D:E
```

See also [clickable buttons](#)

3.5.8. Some Shortcuts in WinEdt are broken, what happened?

This can happen, if you upgrade your Windows, or install new software that may use your shortcut in question as Hotkey. Some shortcuts have a predefined meaning and they cannot be reassigned.

For example, **Alt-Tab**, **Alt-F4**, **Alt-ESC**, etc... are interpreted by the Windows System Menu or Task Manager.

And **Ctrl-F4**, **Ctrl-Tab**, and even **Ctrl-F6**, is interpreted by Windows MDI (you'll notice the same in YAP and Adobe Reader - both being MDI applications like WinEdt) ...

But there is not only Microsoft that can hijack your shortcuts. Eg. if you used to have **Shift-Ctrl-I** for the TeX Insert Font Popup Menu and you install ICQ (an Instant Messenger), well, your WinEdt shortcut isn't available anymore, because ICQ uses it as Hotkey to bring its GUI to the foreground. You will have to change the shortcut or disable them all in ICQ.

Depending on your keyboard settings some other keys might be used to change the keyboard layout (German keyboards have **Shift-Ctrl-E** assigned to the EURO symbol). In any case, you'll have to learn to avoid such shortcuts or find a way to disable them. There is nothing WinEdt can do about it (they are intercepted, translated and dispatched by Windows before WinEdt has a chance to have its say)...

WinEdt's **current** shortcuts are listed in Options|Menu Setup...|Shortcut List. In the new Help (CHM Format since March 2005), you'll find a documentation of the default shortcuts under Dialogs → Menu Setup → Shortcut List.

An overview of WinXP specific shortcuts can be found at:

<http://www.microsoft.com/.../keyboardresults.asp...>
and references to other products and OS of Microsoft:
<http://www.microsoft.com/.../keyboard.aspx>

3.5.9. I have upgraded to WinEdt 5.4 [20050317] from previous build and Ctrl-Alt-E no longer pops up the insert environment.

If you did not restore your previous settings (Options|Configurations|Backup **before** you update WinEdt and Options|Configurations|Restore **after** updating), you have to manually activate this short cut again in Options|Menu Setup|Popup Menus

Select the **Environment** item and make sure it is enabled and the short cut is **Ctrl-Alt-E**.

But: this reflects the old Popup Menu invocation, (still working though). Alex changed some settings, so it is easier for mode writers to use the same shortcuts for comparable tasks in different modes (shortcut economics :-)).

Better set the shortcut in **Options|Menu Setup|Main Menu**. Open the first item **Popups** and you'll find an item **Environments**; proceed the same way as before. This menu item will call the desired Popup Menu via a small macro that checks the mode you are currently in (to be found in %B\Menus\Popups, mode writers may extend these macros).

See also the previous entry about **hijacked shortcuts** (especially about EURO symbol shortcut).

3.5.10. How can I replace the command associated to the **ps2pdf** button to execute Acrobat Distiller?



Note

Alex doesn't maintain macros for Acrobat Distiller (work is hard enough to keep up with the changes that every new Acrobat Reader version has in its baggage), but he provided clear instructions on the mailing list on how to set up such a customization yourself.

If you have WinEdt 5.4 start **Options|Execution Modes** interface and select **ps2pdf** accessory.

Change the executable from

```
gswin32c.exe
```

to

```
"%$( 'Acro-Bin' );\acrodist.exe"
```

or if this for some reason fails replace `%$('Acro-Bin');` with full path to `acrodist.exe` enclosed in double quotes, eg.:

```
"C:\Program Files\Adobe\Acrobat 6.0\acrodist.exe"
```

Change parameters from

```
-sOutputFile="%N.pdf" -c save pop -f "%N.ps"
```

to

```
"%N.ps"
```

and switches from

```
-dBATC -dNOPAUSE -sDEVICE=pdfwrite -r600 -dCompatibilityLevel=1.4
```

to empty or something like

```
/N /Q /V
```

with

```

/V - start viewer
/N - run in a new Window
/Q - close Console Window after finished

```

Decide if you want to run it in detached mode or else simply disable all *fancy* execution options for this accessory... No need to modify the macro. You should not use any switches for paper size unless you change the macro so that it will pass switches that distiller *understands* (defaults are for ghostscript) as you can see in the *complicated* macro.

In WinEdt 5.3 you'll find the following comment in `ps2pdf.edt`:

```

// Use Acrobat's Distiller (if available)
// Omit the /V to avoid viewing newly created .pdf files
// Run('acrodist.exe /N /Q /V "%N.ps"', "%P", 0, 1, "PS -> PDF ...", 1, 1);
// Exit;

```

Remove the comments `//` and possibly enter full path for `acrodist.exe` in double quotes:

```

Run('C:\Program Files\Adobe\Acrobat 6.0\acrodist.exe' /N /Q /V
"%N.ps"', "%P", 0, 1, "PS -> PDF ...", 1, 1);

```

3.6. Access to external programs

3.6.1. How can I quickly run an external (DOS) program?

WinEdt offers you two menu items that can assist at command line executions:

- With **Accessories|Windows Accessories|Command Prompt** WinEdt opens a Command Prompt at your current path (usually `%p` or `%P`) inheriting all environment settings of WinEdt
- With **Accessories|Run...** a small dialog opens, inviting you to specify a Command Line (keeping a small history of the latest command executions) and an optional Startup Folder. For example:

Command Line

```
makeindex -s nomencl.ist -o "%N.gls" "%N.glo"
```

Startup Folder

`%P` (or any other path)

corresponds to macro call:

```
Run('makeindex -s nomencl.ist -o "%N.gls" "%N.glo" %P');
```

3.6.2. How can I write a macro, that runs an external (DOS) program?

Everything that is working on the command line (eg. perl script with parameter switches and options), can be run with `Run ()`:

```
Run("script.pl --verbose foo.glo > foo.err");
```

If you need to set the current path, simply specify the second parameter of the `Run ()` macro. If you want to replace filenames dynamically with current (main) file name, you can make use of `%n` or `%N` (postulating `%N` being something like "foo.ext"). In this case make sure to use quotes around the input file's name (`%N` may expand to a string containing whitespaces):

```
Run('script.pl --verbose "%N".glo > "%N".err', 'mypath');
```

More sophisticated execution modes are available with `WinExe()`. Check the macro manual at [Help|Macro Manual](#) for a complete explanation of further parameters that can be declared. If you want to tie such a macro together with a menu item or a shortcut see also [adding menu item](#) or [installing shortcut](#) respectively.

3.7. Project Files, File Handling

3.7.1. What is the difference between *quitting* and *closing* a project?

Both methods close all opened documents and the project file and set the current project to `<temporary project>`.

Quitting does not save the current state of the project and removes the project from the list of recent projects (unless it was a default project). It will not prompt about saving changes to the project (but it prompts with a warning to this subject). It can be used if you are definitely *done* with that project... It cleans up the Recent Files List and the Project Tree as well, leaving WinEdt in a *virgin* state, as if you had started the program with `-V` switch.

Whereas *Closing* the project just closes all the files contained in your project. If you want to happen more here, you can modify the event handler that is executed when closing a project:

```
Exe('%b\Macros\Events\Prj-Close.edt');
```

It is left to this event handler to take care of cleanup, if desired. By default it does not clean anything (but it can be changed), look for the following line:

```
DeleteTree(1, $$0000);
```

Usage of `DeleteTree(UseFlag: 0..1, Flag: $$0000..$$1111);`

- UseFlag: 0..1
 - 0 delete the tree (as before)
 - 1 use flags to select the tasks
- Binary Flags from left to right `$$0000..$$1111`:
 - Clear Gather Control
 - Delete the Tree
 - Clear File List
 - Remove persistent Main File

See also [Q: 3.1.2](#).

3.7.2. If I run the **Open...** or **Save as...** commands from the **File** menu I would like WinEdt to open up the dialog box in the folder of the file currently being edited.

WinEdt starts the common **Open** and **Save As** dialogs in the current folder. By default, the folder is reset to the last opened document and can be set explicitly by **Ctrl-Shift-H** command (`SetFolder()` to the path of the current document). The menu item is either in **Project** or **Document** menu (depending on your version of WinEdt).

I find this handy since it allows more control (when I use **Save As** command I usually want to save the file to a different folder). If you want your asked behavior by default, you have

to delete the current **Open** and **Save As** Menu Items (because they are defined as commands) and then replace them by macro items:

```
SetFolder('%p');CMD('Open...');
```

respectively

```
SetFolder('%p');CMD('Save As...');
```

3.7.3. When opening a file, how can I force WinEdt to always choose the `.tex` file-type?

If you don't want the "Default" mode (`*.*`) to show up as default in the file open dialog, reverse the lines `Default` and `TeX` in the edit control inside the **Preferences|Modes** Dialog. This changes the order of file extension items in the file open dialog.

3.8. Highlighting

3.8.1. Can I define my own colours for highlighting?

Yes. You find the dialog **Colors** and **Syntax Highlighting** under **Options|Highlighting**. In the first tabbed pane **Selections and Colors** choose the 5th button **Customize Basic Colors**, which opens a new dialog, where you can change any of the 16 colours. Note that highlighting can look weird if you choose completely different colours. You can always return to the default colours.

In Version 5.3 of WinEdt go to **Options|Preferences|Colors** and open the context popup menu (Rightclick on **Colors**). Then choose **Change Colors** to redefine the colours.

3.8.2. I would like to have comments highlighted ...

Expanded Question: I have a document mode that uses `!` as a comment character. I would like to have the comments highlighted and therefor I copied the `TeX Comment Switch`, changed the `%` to a `!` and changed `TeX` to `MyMode`. But it doesn't work. What else do I need to do?

Answer: You have to define a Filter Set `[" ! "]` and use it as a Start Switch (from the drop-down list). Changing `%` to `!` in the name does nothing and Filter Sets have to be defined in the **Filter Sets** Tab Page before they can be properly assigned and used in switches...

A good introduction to this subject can be found in [Syntax Highlighting in WinEdt](#) [pdf; KB]

4. Text Manipulation

4.1. Text Manipulation with RegEx

4.1.1. What are Regular Expressions?

Regular expressions are constructs that look like comic-book expletives but can be wondrously powerful tools. Using Regular Expressions you can find (and replace) text that matches a certain pattern. Some characters have a special meaning: F.i. the character `?` stands for "*any (single) character*". So, if you wanted to find all four-letter words that begin and end with `"t"` (tart, text, tilt etc.) you would search for `t???t`.

You can also **tag** expressions to reuse them. F.i., if you wanted to change all `\urls` into `\hrefs`, you would search for `\\url\\{(0*\\)\\}` and replace them with

`\\href\\{mailto:\\0\\}\\{\\0\\}`. Note that the special characters (like `{`, `}` and `\\`) have to be escaped by a `\\` to keep their literal meaning.

The file `%B\\Doc\\RegEx.txt` will tell you more about WinEdt's regular expressions.

4.1.2. How can I extract Hard Returns in paragraphs, but keep the paragraph separation (a single empty line between them)?

Easiest is to use Search|Replace (**Ctrl-R**) with Regular expressions checked:

```
Find: "~<>~"
Replace with: " "
```

or possibly

```
Find: "~<>~{>| }"
```

if you want to prevent (backward) formatting of indented lines.

4.1.3. I want to search with a Regular Expression that can span over more than one line. This does not seem to work with `"*"`.

That's right. The Regular Expression `"*"`, which stands for *any string of arbitrary length*, does not span over line breaks.

If you are using WinEdt **5.3** you can use the expression `"**"`, which allows line breaks. If you are using an earlier version, you can use something along the lines of the following Regular Expression, which will find LaTeX environments:

```
\\begin\\{*\\}@{>|^\\{\\end\\{*\\}\\}\\end\\{*\\}
```

This code will fail on nested environments, as it will stop at the first possible match for `\\end\\{*\\}`. For these, the expression gets more complicated (I've replaced `\\begin\\{*\\}` and `\\end\\{*\\}` with `begin` resp. `end` to make it somewhat more readable):

```
begin@>|{begin*end}|^{end}}end
```

will find a maximum number of two nested environments (as long as the inner environment starts and ends on the same line);

```
begin@>|{begin@>|{begin*end}|^{end}}*end}|^{end}}end
```

will find a maximum number of three nested environments;

```
begin@>|{begin@>|{begin@>|{begin*end}|^{end}}*end}|^{end}}*end}|^{end}}end
```

will find a maximum number of four nested environments, and so on.

Of course these expressions will find all environments which are nested in a smaller degree, too. Environments which contain more than the maximum number, will not be matched correctly.

4.1.4. My macro (or Regular Expression) doesn't work as expected. I think I found a bug.

You may be right. But it might already be known. Check in the [list of bugs](#). If you don't find it there, construct a minimal test macro which shows the behaviour you think is wrong, and send it (preferably) to the WinEdt mailing list or to WinEdt's support (if you do the latter,

please CC to <webmaster@winedt.org>). If it turns out that you've really found a bug, it will probably be fixed in the next release.

4.2. Text Manipulation with Macros

4.2.1. What is a macro?

A macro is a script that contains a number of editing or other commands, which are executed in WinEdt. You can use macros f.i. for repeated editing tasks like typing 168 lines of a LaTeX table. An easy way to learn what a macro is, is the Macro Recorder (Macros|Recorder...), which lets you record and play macros.

You are probably using macros already: F.i. most of the Insert commands are macros, and a lot of responses to Active Strings are macros. Do you type `\textit{...}`? Don't! Press **ALT-CTRL-F** followed by **ALT-I** instead. So whenever you find yourself typing the same things again and again, use a macro!

In the [WinEdt Macro Library](#) you find a lot of more advanced macros, that can make your life with WinEdt (and life in general) much easier.

4.2.2. How do I run a macro?

Go to Macros|Execute Macro... and browse for the macro file. If you want to run a macro more often, read the next question, too.

4.2.3. How can I quickly run a macro?

You have two possibilities, if you want to run a macro frequently:

- Define a new Menu Item (either in a main menu or in a popup menu). This is done in Options|Menu Setup. Insert a macro item and define it as

```
[Exe("%B\path\to\your\macro.edt");]
```

- You can then assign a keyboard shortcut (see [installing a shortcut](#)) to it, or even a Toolbar Item. Or you can define the macro as response to an Active String (in Options|Settings|Active Strings).

To learn more, read the [instructions](#) from the [WinEdt Macro Library](#), and the files `Menu.txt` and `Toolbar.txt` in WinEdt's Doc folder.

4.2.4. WinEdt hang up while executing a macro, have I to kill the program using the Task Manager?

Instead of using the Task Manager to kill WinEdt when it is executing a macro code that is too slow or possibly contains an infinite loop, the following applies:

Press and hold the keys **Shift-ESC** until the status line displays a message `Canceled!!!`. You may have to do this a few times to exit nested loops, but in principle it is possible to cancel all such actions using this shortcut.

4.2.5. The register I want to change always seems to contain an old value. What's happening?

You're probably experiencing WinEdt's string expansion rules. With this macro code:

```
LetReg(1,"Test");
IfStr("%!1","Test","=",>
    "LetReg(1,'It works.');" Prompt('%!1');">
);
```

you will be prompted: `Test` and not, as expected, `It works`, because the whole string

```
"LetReg(1,'It works. '); Prompt('%!1');"

```

will be expanded at once, i.e. before the register is changed. Therefore %!1 does not contain the new value. To prevent the register from being expanded you have two possibilities:

- Either double the character %, i.e.

```
"LetReg(1,'It works. '); Prompt('%%!1');"

```

- You have to double the registers for each expansion level. or, better in most cases, put a ! in front of the string:

```
!"LetReg(1,'It works. '); Prompt('%!1');"

```

4.2.6. How can I duplicate a line?

WinEdt has a macro language that allows you to define your own actions and assign them to the keys you prefer. You could use the following (easy readable, straight forward step by step thinking) macro to duplicate a line:

```
SetInsert(1);
CMD("Go To Beginning Of Line");
CMD("Select To End Of Line");
CMD("Copy");
CMD("Char Right");
NewLine;
CMD("Paste");
RestoreInsert;
```

You can save this content to a file named dupl.edt in for example %b\macros. Then you have to set up a menu entry that executes Exe(" %b\macros\dupl.edt "); and assign it to a key combination (see [installing a shortcut](#) for instructions).

But the macro above is much too long. The WinEdt register %L holds already the content of the current line. So the following line will do the job:

```
CMD("Go To Beginning Of Line");Ins('%L%');

```

Note that %\ has the same meaning as \n in C, namely a new line.

```
CMD('Go To End Of Line');Ins('%\%L');

```

does the same job, but leaves you at the end of the second line. These macros are that short, that you probably don't need to save them in a extra macro file.

Of course, with wrapping on, the new line gets ... well, wrapped! This can be avoided by using InsText instead of Ins, which prevents any reformatting of the inserted text:

```
CMD("Go To Beginning Of Line");InsText('%L%');

```

4.2.7. How can I change a text line by line (cyclically)?

Suppose the the following problem:

You want to use the filenames gathered in the tree of a project. You can store the filenames with `GetTree(6);` in register 6. The problem is that an asterisk is inserted before the filenames and you would like to get rid of them.

The following macro does the job. Register 6 keeps input and result.

```
GetTree(6);
LetRegNum(9,-1);
SetOK(1);
Loop(!` >
    FindInString("%!6", "<\*", 8, 9, 1001,%!9+1); >
    IfOK(!' >
        ReplaceInString("%!6", "<", %!8, %!9, 1, 6); >
    ', ' >
        Stop; >
    '); >
`);
End;
```

- ❶ Initialize resume index in register 9
- ❷ The `Loop` macro allows you to deal with jobs, that have to be executed cyclically. As it is executed at least once, it works similar to a `Repeat/Until` instruction.
Notice the `>` characters at the end of the lines (inside the loop). They instruct WinEdt to concatenate the lines into one single line while parsing the loop-argument.
- ❸ The break is set by `Stop` - usually in combination with a testing environment like `IfOK`
- ❹ Searching takes place in register 6, replacing too. Use of Regular Expression and Resume Index
 - `<`: BOL;
 - `*`: escape the special meaning of `*` in the regular expression)
 - take `<*@\{ }` instead of `<*` if you want discard the whitespace too (indentation)
 - start searching at index `%!9+1` and store the result in register 9 again.
- ❺ *Regular Expressions* (second last parameter) are activated to insert a new *Beginning of Line* with `>`. You can of course use anything else, for example `;` as token separator.

5. Resources

5.1. Where can I get more help?

You can ask on WinEdt's Mailing List. Or, if you prefer that, you can directly contact the WinEdt Team <support@winedt.com>.

If you have a question that is not directly related to WinEdt but rather to TeX, LaTeX and friends, you can search one of the **TeX FAQs**:

- www.tex.ac.uk/faq (English),
- xpt.sourceforge.net/techdocs/Latex (English), excerpt of question and answers, loosely gathered mainly from newsgroup comp.text.tex,
- www.dante.de/faq/de-tex-faq/ (German), or
- www.loria.fr/tex/divers.html (French).

subscribe to a classical **mailing list** (only one known to me so far)

- [German Language TeX Users Group Communication List](#)

or ask various **newsgroups**

- [comp.text.tex](#)
- [de.comp.text.tex](#)
- [fr.comp.text.tex](#)

If you aren't able to register to newsgroups (because of firewalls,...), you may give <http://groups-beta.google.com> a try. This site displays with a delay of some hours the current content of specific newsgroups (in thread style) and you can *google* for a certain subject within a specific group. Take a look at [comp.text.tex](#).

5.2. How do I subscribe to the WinEdt mailing list?

Send an empty message to `<winedt+list-subscribe@wsg.net>`. Make sure that the return address in your e-mail is correct. On subscribing you will be sent instructions on how to remove yourself from the list, should you so wish.

5.3. How do I unsubscribe from the WinEdt mailing list?

Send an empty message to `<winedt+list-unsubscribe@wsg.net>`. Do not send your request to the mailing list! This will have no effect except for annoying the subscribers.

5.4. Is there an archive for the WinEdt Mailing List?

Yes. As of February 13, 2002, the messages on the Mailing List are archived on <http://groups.yahoo.com/group/WinEdt>.

5.5. Are that all frequently asked questions?

Not at all. But [WinEdt.org](#) is the work of volunteers and our time is limited.

If you miss a question, and you know the answer, it would be great if you write it down and send it to me: `<georges.schmitz@heitec.de>`, so that I can include it.

Feel free to notify me of mistakes, or to suggest improvements.

Please do not send questions without answers to me.



Thanks to Robert

We all have to thank Robert! A lot of what you see here, was done by him. I took over the maintenance of the work he started and will add new sets of questions and answers in future.