WinEdt

Quick Guide

Aleksander Simonič

Contents

List of Figures

Acknowledgements

Ralf Heckmann supplied the template for LaTeX and KOMA-Script package using Palatino and Bera fonts. This template is used to typeset this document.[1]

Luca introduced and applied text styling commands, fixed a few inaccuracies, and improved a few TeX examples.

Adriana proofread and edited the draft version. She fixed numerous language mistakes, cut out some fat, and slightly rearranged the text in order to improve the *flow* (for those that notice such things).

> *Here is where you will be acknowledged if you contribute a chapter* (*or a section*) *of general interest that will improve this manual and help other users* (*new and experienced*) *to learn how to better use WinEdt. There is plenty of room for improvements and additional information but it will require the involvement of the WinEdt Community. This is your chance to do something about it!*

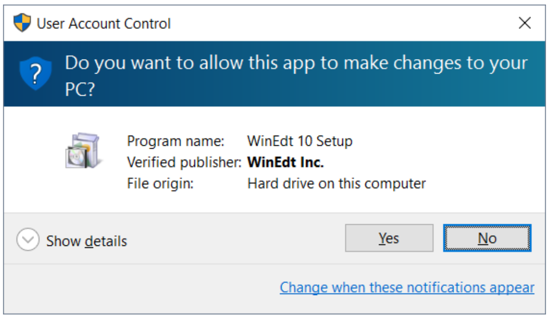
[1] LaTeX sources for this guide are available for download in a self-contained zipped archive. Unpack its contents in a directory of your choice, open the main document `QuickGuide.tex` and set it as the main file in WinEdt's tree interface. Now you are ready to compile it with PDFTeXify button (or the universal compile shortcut F9). After the compilation is complete the resulting pdf document will be opened in your PDF Viewer (such as Adobe Reader or SumatraPDF). Please think about the environment before you decide to print the whole document...

Figure 1: TEX Lion: ready and willing…

1 Installing WinEdt and TEX

1.1 Installing WinEdt is easy

Download and execute the setup file `winedt102-64.exe` (or `winedt102-32.exe` if you choose to install the 32-bit version).[1] As long as you've downloaded a legitimate copy of WinEdt (eg. from our web site www.winedt.com) you will receive the standard Windows User Account Control warning that displays WinEdt Inc. as the *verified publisher* of this product.



User Account Control

Do you want to allow this app to make changes to your PC?

Program name: WinEdt 10 Setup
Verified publisher: **WinEdt Inc.**
File origin: Hard drive on this computer

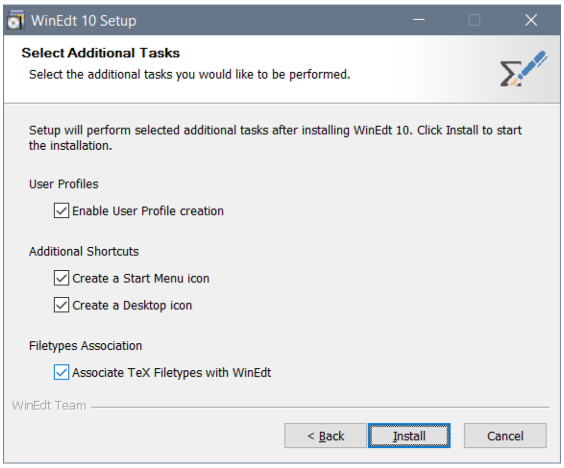
Figure 1.1: WinEdt Setup UAC Warning

It is now safe to proceed with the installation. The Installer Wizard will guide you. For most users the default options in the Wizard should be used. If you are creating a portable installation, however, disable User Profile Creation: this way WinEdt will create a `Local` folder inside its install folder and this folder will contain all your custom settings. . .

During the installation you can decide if you want this version of WinEdt associated with TEX files (and friends).[2] Filetype associations can later be made, removed, or repaired (on a selective basis) through WinEdt's Options -> Configuration Wizard. However, on Windows with UAC-enabled this will require starting WinEdt with elevated privileges (which are already in effect during the installation).

[1] It is recommended that you download and install the latest version from our web site. Ordering a CD is not necessary as no extras are included on it. If you really need or want a physical copy you can save yourself some money by burning your own CD.

[2] This is the default and recommended for most users.



WinEdt 10 Setup

Select Additional Tasks

Select the additional tasks you would like to be performed.

Setup will perform selected additional tasks after installing WinEdt 10. Click Install to start the installation.

User Profiles

☑ Enable User Profile creation

Additional Shortcuts

☑ Create a Start Menu icon

☑ Create a Desktop icon

Filetypes Association

☑ Associate TeX Filetypes with WinEdt

Figure 1.2: WinEdt Setup

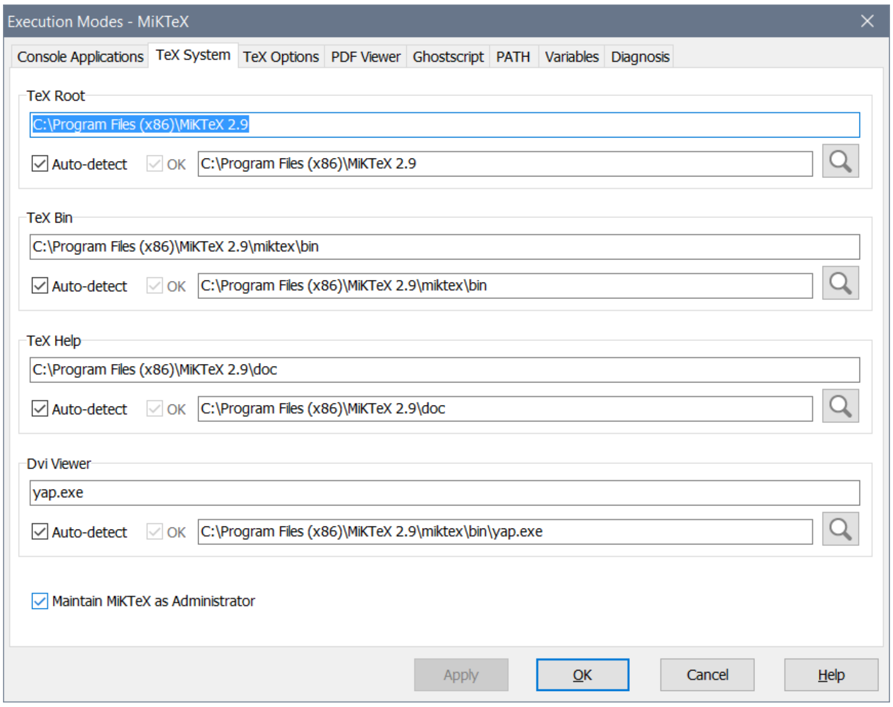
After the installation is completed start WinEdt from the shortcut on your desktop: now you are ready to go. If you are planning to use WinEdt on a regular basis you should consider pinning it to your taskbar. This is done from the Windows popup menu that is displayed when you right-click on the WinEdt icon on your desktop. The same menu can also be used to unpin the program if you find your taskbar overloaded with applications.

1.2 What about TEX?

If you are planning to use WinEdt for writing LATEX documents you must also install a TEX System such as MiKTEX or TEX Live; both TEX Systems run well on Windows, the choice is yours.[3] If you haven't done it yet this would be a good time! After the TEX installation is successfully completed restart WinEdt and it will automatically detect your TEX System and connect with its accessories.

> If in doubt, you can always start Execution Modes dialog in WinEdt's Options menu and check the last page: Diagnosis. It explains everything. Help in this dialog offers a guide on how to manually fix any problems in a rare situation when automatic detection fails.

[3] MiKTEX download is considerably smaller and the installation is faster. MiKTEX also has its Package Manager and Update Wizard to help you keep it current. However, MiKTEX may have to use its install-on-demand feature during a compilation in order to download and install missing packages that you may be using in your documents. This can make compilation slow until all missing packages have been installed. TEX Live, on the other hand, comes loaded with just about every supported LATEX package. This makes the download huge and the installation slower than MiKTEX. It also requires an annual update in order to keep its large package repository up-to-date. However, once it is installed it runs fast.



TeX Root

C:\Program Files (x86)\MiKTeX 2.9

☑ Auto-detect ☑ OK C:\Program Files (x86)\MiKTeX 2.9

TeX Bin

C:\Program Files (x86)\MiKTeX 2.9\miktex\bin

☑ Auto-detect ☑ OK C:\Program Files (x86)\MiKTeX 2.9\miktex\bin

TeX Help

C:\Program Files (x86)\MiKTeX 2.9\doc

☑ Auto-detect ☑ OK C:\Program Files (x86)\MiKTeX 2.9\doc

Dvi Viewer

yap.exe

☑ Auto-detect ☑ OK C:\Program Files (x86)\MiKTeX 2.9\miktex\bin\yap.exe

☑ Maintain MiKTeX as Administrator

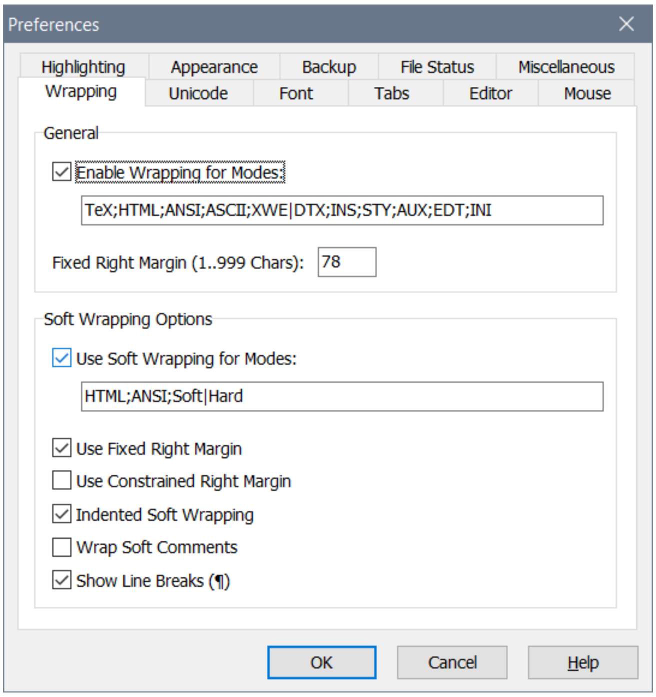
Figure 1.3: Execution Modes: TEX System

> *Some users may also choose to install additional accessories such as Ghostscript and GSView. However, for those of us that do not intend to work with legacy postscript output format the only thing missing is a suitable PDF Viewer. SumatraPDF allows trouble-free previewing of compiled pdf documents with support for Forward and Inverse Search to switch between sources and the compiled pdf document. This is further explained in the next chapter...*

1.3 What Next?

If you have a properly installed TEX system you are ready to proceed to the next chapter and see WinEdt in action working on this document.

> The default settings use Notepad-like Soft Wrapping and UTF-8 encoding for TEX Documents, as expected by new users. If these are not not your preferences you should make some changes now before starting to work on your documents (this can save you from a lot of problems and confusion later)! The Options -> Preferences dialog offers some basic choices pertaining to your preferred Wrapping, Unicode, and Backup strategy. Help in the dialog explains how.



General

☑ Enable Wrapping for Modes:

TeX;HTML;ANSI;ASCII;XWE|DTX;INS;STY;AUX;EDT;INI

Fixed Right Margin (1..999 Chars): 78

Soft Wrapping Options

☑ Use Soft Wrapping for Modes:

HTML;ANSI;Soft|Hard

☑ Use Fixed Right Margin

□ Use Constrained Right Margin

☑ Indented Soft Wrapping

□ Wrap Soft Comments

☑ Show Line Breaks (¶)

Figure 1.4: Preferences: Wrapping

1.4 Upgrading

Upgrading from WinEdt 10 series is easy!

> *You don't have to uninstall the previous build of WinEdt 10 in order to upgrade to the latest version. In fact you should not uninstall it if you want to preserve custom settings in your Application Data or Local folder.*

Here are the steps that you should perform for smooth upgrading:

- Install the new version (over the old one).
- Restart WinEdt; new default settings are used and your last project is loaded. . .
- Execute Options -> Maintenance -> Upgrade Personal Configuration menu command (to merge your personal settings with the new defaults).

The "What's New?" section in WinEdt's Manual explains what has been added, changed, or fixed for the latest build. Check it out!

2 Getting Started

So you've downloaded and installed WinEdt and started it for the first time. This is what you see:

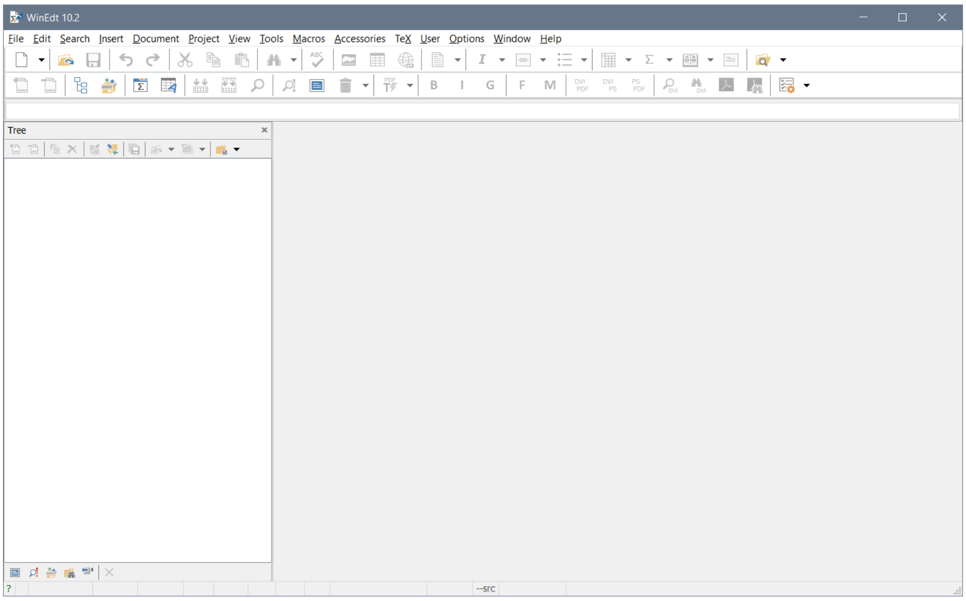

WinEdt 10.2
File Edit Search Insert Document Project View Tools Macros Accessories TeX User Options Window Help
Tree
--src

Figure 2.1: WinEdt running for the first time

Not particularly impressive or useful as is. On the left is an empty tree control that will later be used to display the project structure, Table of Contents, and other items collected in the relevant documents. On top is the (legacy) default 2-row toolbar with many buttons disabled (grayed out). There's more to say about the toolbar interface, but for now let us point out that while many such buttons are currently disabled that does not mean there's a bug or indicate something's wrong with your WinEdt/TEX installation. That the TEX compiler, converters and previewer buttons are disabled simply indicates that currently there is nothing to compile, convert, or preview.[1]

[1] If at this stage you have any doubt about what TEX System and other accessories WinEdt detected on your computer, start the Options -> Execution Modes dialog and check the last, Diagnosis, tab page. The report produced there will clearly indicate whether you forgot to install a TEX system such as MiKTEX or TEX Live (in which case, now is the time to do it!)...

2.1 Setting up a TEX project in WinEdt

Now we open the main file used to create this document: `QuickGuide.tex`. Use the button on the left of the toolbar in the tree interface to set this document as the main file for this WinEdt TEX project.

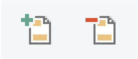
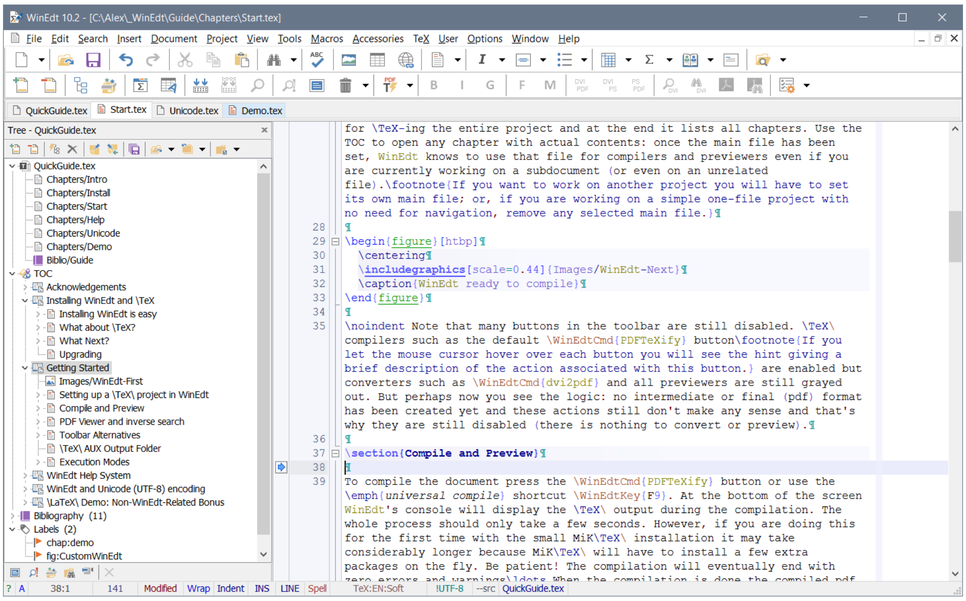


Figure 2.2: Set and Remove Main File Buttons

With the main file set, WinEdt builds the project tree with all included chapters and displays the TOC branch for this document.[2] This main file consists of the preamble required for TEX-ing the entire project and at the end it lists all chapters. Use the TOC to open any chapter with actual contents: once the main file has been set, WinEdt knows to use that file for compilers and previewers even if you are currently working on a subdocument (or even on an unrelated file).[3]



WinEdt 10.2 - [C:\Alex_WinEdt\Guide\Chapters\Start.tex]
File Edit Search Insert Document Project View Tools Macros Accessories TeX User Options Window Help
QuickGuide.tex Start.tex Unicode.tex Demo.tex
Tree - QuickGuide.tex
QuickGuide.tex
Chapters/Intro
Chapters/Install
Chapters/Start
Chapters/Help
Chapters/Unicode
Chapters/Demo
Biblio/Guide
TOC
Acknowledgements
Installing WinEdt and \TeX
Installing WinEdt is easy
What about \TeX?
What Next?
Upgrading
Getting Started
Images/WinEdt-First
Setting up a \TeX\ project in WinEdt
Compile and Preview
PDF Viewer and inverse search
Toolbar Alternatives
\TeX\ AUX Output Folder
Execution Modes
WinEdt Help System
WinEdt and Unicode (UTF-8) encoding
\LaTeX\ Demo: Non-WinEdt-Related Bonus
Bibliography (11)
Labels (2)
chap:demo
fig:CustomWinEdt
? A 38:1 141 Modified Wrap Indent INS LINE Spell TeX:EN:Soft !UTF-8 --src QuickGuide.tex

```
for \TeX-ing the entire project and at the end it lists all chapters. Use the
TOC to open any chapter with actual contents: once the main file has been
set, WinEdt knows to use that file for compilers and previewers even if you
are currently working on a subdocument (or even on an unrelated
file).\footnote{If you want to work on another project you will have to set
its own main file; or, if you are working on a simple one-file project with
no need for navigation, remove any selected main file.}¶
¶
\begin{figure}[htbp]¶
  \centering¶
  \includegraphics[scale=0.44]{Images/WinEdt-Next}¶
  \caption{WinEdt ready to compile}¶
\end{figure}¶
¶
\noindent Note that many buttons in the toolbar are still disabled. \TeX\
compilers such as the default \WinEdtCmd{PDFTeXify} button\footnote{If you
let the mouse cursor hover over each button you will see the hint giving a
brief description of the action associated with this button.} are enabled but
converters such as \WinEdtCmd{dvi2pdf} and all previewers are still grayed
out. But perhaps now you see the logic: no intermediate or final (pdf) format
has been created yet and these actions still don't make any sense and that's
why they are still disabled (there is nothing to convert or preview).¶
¶
\section{Compile and Preview}¶
¶
To compile the document press the \WinEdtCmd{PDFTeXify} button or use the
\emph{universal compile} shortcut \WinEdtKey{F9}. At the bottom of the screen
WinEdt's console will display the \TeX\ output during the compilation. The
whole process should only take a few seconds. However, if you are doing this
for the first time with the small MiK\TeX\ installation it may take
considerably longer because MiK\TeX\ will have to install a few extra
packages on the fly. Be patient! The compilation will eventually end with
zero errors and warnings\ldots When the compilation is done the compiled pdf
```

Figure 2.3: WinEdt ready to compile

Note that many buttons in the toolbar are still disabled. TEX compilers such as the default PDFTeXify button[4] are enabled but converters such as dvi2pdf and all

[2] Also, the name of the main document now shows in the last panel in the status line.

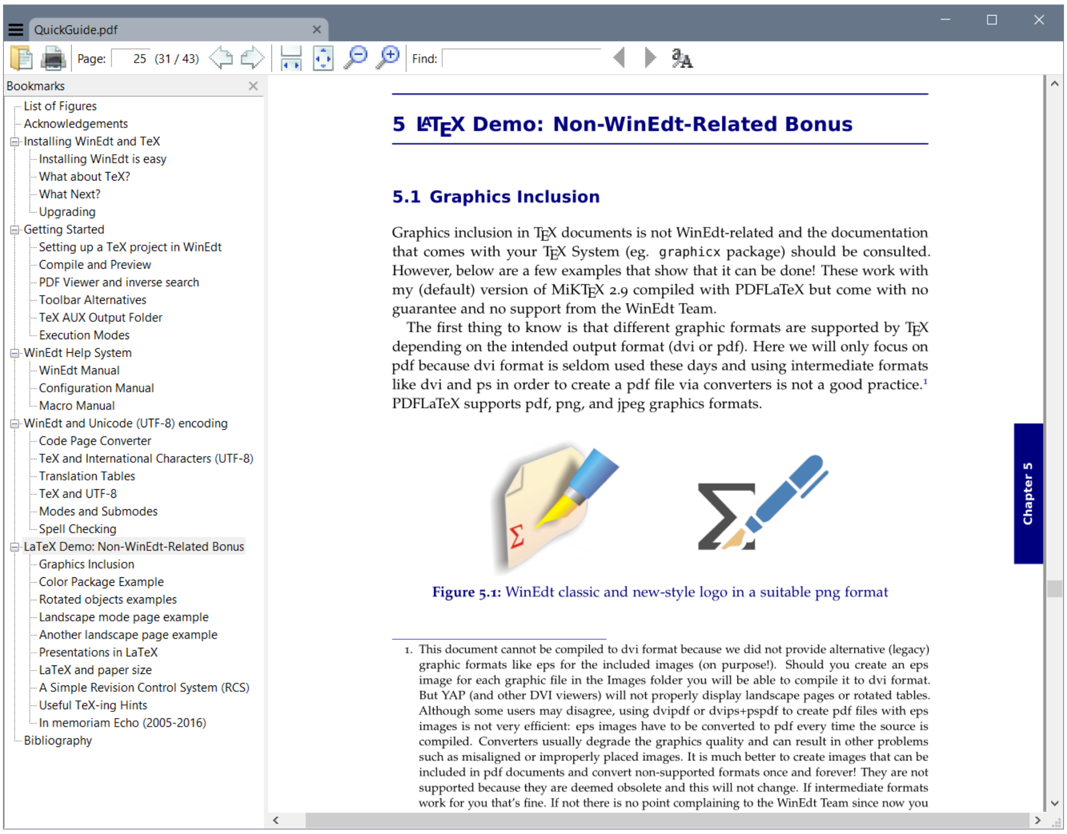
[3] If you want to work on another project you will have to set its own main file; or, if you are working on a simple one-file project with no need for navigation, remove any selected main file.

[4] If you let the mouse cursor hover over each button you will see the hint giving a brief description of the action associated with this button.

previewers are still grayed out. But perhaps now you see the logic: no intermediate or final (pdf) format has been created yet and these actions still don't make any sense and that's why they are still disabled (there is nothing to convert or preview).

2.2 Compile and Preview

To compile the document press the PDFTeXify button or use the *universal compile* shortcut F9. At the bottom of the screen WinEdt's console will display the TEX output during the compilation. The whole process should only take a few seconds. However, if you are doing this for the first time with the small MiKTEX installation it may take considerably longer because MiKTEX will have to install a few extra packages on the fly. Be patient! The compilation will eventually end with zero errors and warnings... When the compilation is done the compiled pdf document will be displayed in the default PDF Viewer on your computer.



- List of Figures
- Acknowledgements
- Installing WinEdt and TeX
  - Installing WinEdt is easy
  - What about TeX?
  - What Next?
  - Upgrading
- Getting Started
  - Setting up a TeX project in WinEdt
  - Compile and Preview
  - PDF Viewer and inverse search
  - Toolbar Alternatives
  - TeX AUX Output Folder
  - Execution Modes
- WinEdt Help System
  - WinEdt Manual
  - Configuration Manual
  - Macro Manual
- WinEdt and Unicode (UTF-8) encoding
  - Code Page Converter
  - TeX and International Characters (UTF-8)
  - Translation Tables
  - TeX and UTF-8
  - Modes and Submodes
  - Spell Checking
- LaTeX Demo: Non-WinEdt-Related Bonus
  - Graphics Inclusion
  - Color Package Example
  - Rotated objects examples
  - Landscape mode page example
  - Another landscape page example
  - Presentations in LaTeX
  - LaTeX and paper size
  - A Simple Revision Control System (RCS)
  - Useful TeX-ing Hints
  - In memoriam Echo (2005-2016)
- Bibliography

5 LATEX Demo: Non-WinEdt-Related Bonus

5.1 Graphics Inclusion

Graphics inclusion in TEX documents is not WinEdt-related and the documentation that comes with your TEX System (eg. `graphicx` package) should be consulted. However, below are a few examples that show that it can be done! These work with my (default) version of MiKTEX 2.9 compiled with PDFLaTeX but come with no guarantee and no support from the WinEdt Team.

The first thing to know is that different graphic formats are supported by TEX depending on the intended output format (dvi or pdf). Here we will only focus on pdf because dvi format is seldom used these days and using intermediate formats like dvi and ps in order to create a pdf file via converters is not a good practice.[1] PDFLaTeX supports pdf, png, and jpeg graphics formats.

Figure 5.1: WinEdt classic and new-style logo in a suitable png format

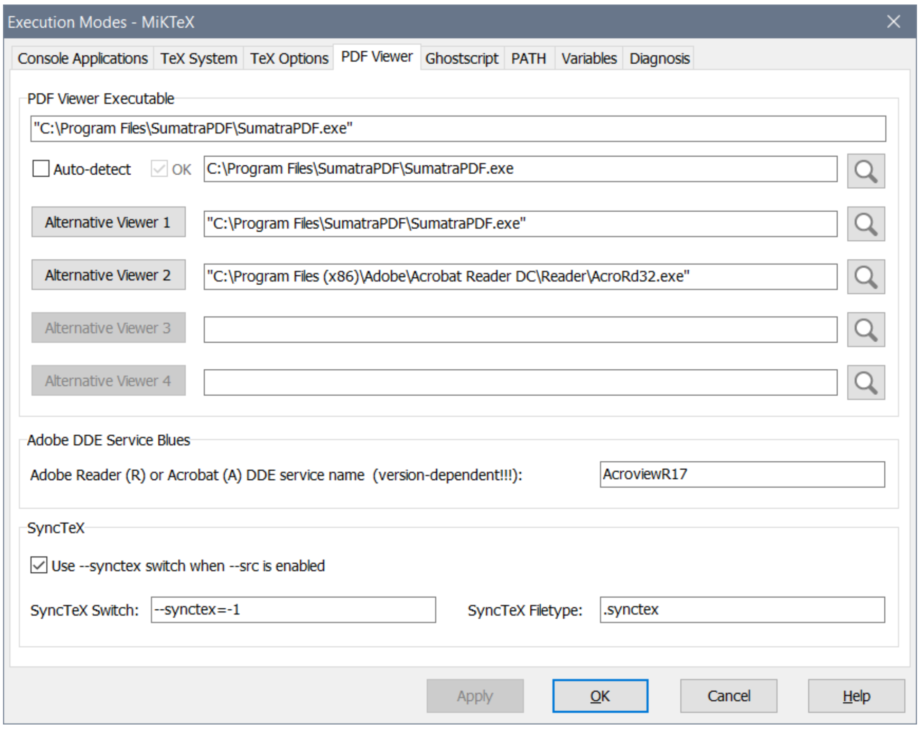
[1] This document cannot be compiled to dvi format because we did not provide alternative (legacy) graphic formats like eps for the included images (on purpose!). Should you create an eps image for each graphic file in the Images folder you will be able to compile it to dvi format. But YAP (and other DVI viewers) will not properly display landscape pages or rotated tables. Although some users may disagree, using dvipdf or dvips+pspdf to create pdf files with eps images is not very efficient: eps images have to be converted to pdf every time the source is compiled. Converters usually degrade the graphics quality and can result in other problems such as misaligned or improperly placed images. It is much better to create images that can be included in pdf documents and convert non-supported formats once and forever! They are not supported because they are deemed obsolete and this will not change. If intermediate formats work for you that's fine. If not there is no point complaining to the WinEdt Team since now you

Figure 2.4: This document displayed in SumatraPDF viewer

2.3 PDF Viewer and inverse search

Most users have Adobe Reader or Acrobat as the default pdf viewer on their computer. That is fine. However, Adobe is *not* the most suitable working previewer for pdf files generated externally (eg. by PDFLaTeX).

SumatraPDF is a light-weight PDF Viewer that does not lock the pdf files it is previewing: it automatically refreshes their contents when they are recompiled, and it supports forward and inverse search, described below, based on synctex technology.[5] It is strongly recommended that you install this application for TeXing.[6] In fact SumatraPDF may be the only external application besides your TeX System that you need in order to make compiling and previewing problem-free. It is easy to use alternative PDF Viewers in WinEdt: start Options -> Execution Modes dialog, go to the PDF Viewer page, the click on the Help button for details…



PDF Viewer Executable

"C:\Program Files\SumatraPDF\SumatraPDF.exe"

□ Auto-detect ☑ OK C:\Program Files\SumatraPDF\SumatraPDF.exe

Alternative Viewer 1 "C:\Program Files\SumatraPDF\SumatraPDF.exe"

Alternative Viewer 2 "C:\Program Files (x86)\Adobe\Acrobat Reader DC\Reader\AcroRd32.exe"

Alternative Viewer 3

Alternative Viewer 4

Adobe DDE Service Blues

Adobe Reader (R) or Acrobat (A) DDE service name (version-dependent!!!): AcroviewR17

SyncTeX

☑ Use --synctex switch when --src is enabled

SyncTeX Switch: --synctex=-1 SyncTeX Filetype: .synctex

Figure 2.5: Execution Modes: PDF Viewer Alternatives

In SumatraPDF you can double-click anywhere in the document and WinEdt will display the corresponding source. This is called *Inverse Search* and is implemented in the PDF Viewer. *Forward Search* is started in the editor and results in the PDF Viewer displaying the compiled text corresponding to the source position. In WinEdt you

[5] This functionality is superior to the older dvi src specials used by DVI Viewers such as YAP.

[6] You don't have to replace Adobe as your default pdf viewer

have a button or shortcut to initiate this action. It can also be done by clicking on the current line bookmark in the left margin (or double-clicking on the left margin in front of the desired line). If you are not familiar with these concepts[7] try it: it works very well!

2.4 Toolbar Alternatives

Notice that many buttons in the toolbar are still disabled: converters like DVIPS, dvi2pdf, ps2pdf and also Dvi Preview or GSView are still greyed out. This makes sense since we did not create any intermediate (legacy) output formats such as dvi or ps! Those who are used to such legacy formats have their reasons[8] for sticking to them but creating pdf output with one of the PDF engines is all the TEX-ing most of us will ever want. And with vertical space being such a precious commodity on most wide screens we don't really need or want a two-row toolbar where one-row will do just as well (see Figure 2.7).

Many users will use F9 for compiling. And, as pointed out above, forward PDF search from WinEdt 10 can be initiated by clicking on the current line bookmark in the left margin (or double-clicking on the left margin in front of the desired line).[9] This practically eliminates the need for the Forward PDF Search button in the toolbar. It *can* be useful to show whether this command is currently enabled – eg. the existence of pdf and synctex files, and many of us use such buttons for feedback like this rather than actually clicking on them on a regular basis: another argument for a smaller 1-row toolbar! The Options -> Toolbar menu contains a few predefined alternatives for toolbars.

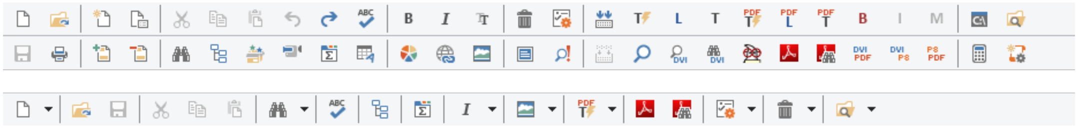


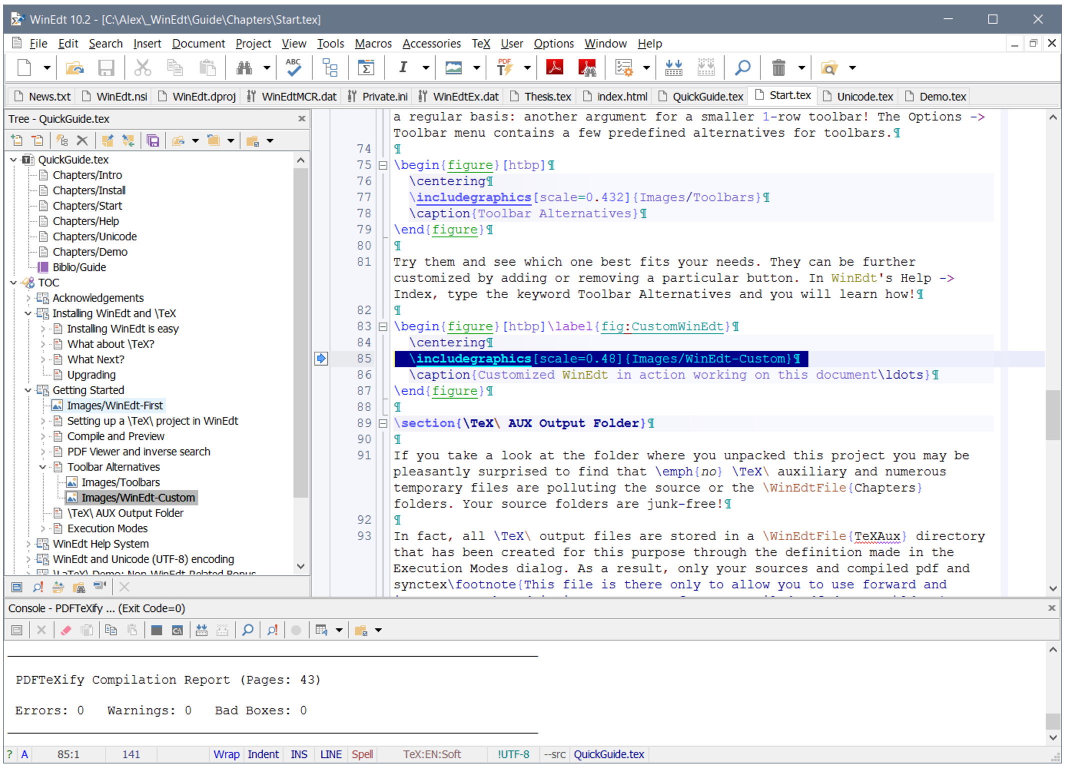
Figure 2.6: Toolbar Alternatives

Try them and see which one best fits your needs. They can be further customized by adding or removing a particular button. In WinEdt's Help -> Index, type the keyword Toolbar Alternatives and you will learn how!

[7] Help in the Execution Modes dialog explains the Forward/Inverse search functionality in detail; read it up as it is very useful and will save you a lot of time.

[8] See Chapter 5 to learn how to import images or control paper size in LATEX without the need for intermediate formats. For example, most users do not have to install Ghostscript or GSView unless they really want to work with postscript output format. Beware though: this document cannot be compiled into dvi or ps because we did not provide any legacy eps version of graphics. . .

[9] This is customizable by "event handlers" but for most users the default settings will be enough.



WinEdt 10.2 - [C:\Alex_WinEdt\Guide\Chapters\Start.tex]
File Edit Search Insert Document Project View Tools Macros Accessories TeX User Options Window Help
News.txt WinEdt.nsi WinEdt.dproj WinEdtMCR.dat Private.ini WinEdtEx.dat Thesis.tex index.html QuickGuide.tex Start.tex Unicode.tex Demo.tex
Tree - QuickGuide.tex
QuickGuide.tex
Chapters/Intro
Chapters/Install
Chapters/Start
Chapters/Help
Chapters/Unicode
Chapters/Demo
Biblio/Guide
TOC
Acknowledgements
Installing WinEdt and \TeX
Installing WinEdt is easy
What about \TeX?
What Next?
Upgrading
Getting Started
Images/WinEdt-First
Setting up a \TeX\ project in WinEdt
Compile and Preview
PDF Viewer and inverse search
Toolbar Alternatives
Images/Toolbars
Images/WinEdt-Custom
\TeX\ AUX Output Folder
Execution Modes
WinEdt Help System
WinEdt and Unicode (UTF-8) encoding
Console - PDFTeXify ... (Exit Code=0)
? A 85:1 141 Wrap Indent INS LINE Spell TeX:EN:Soft !UTF-8 --src QuickGuide.tex

```
a regular basis: another argument for a smaller 1-row toolbar! The Options ->
Toolbar menu contains a few predefined alternatives for toolbars.¶
¶
\begin{figure}[htbp]¶
  \centering¶
  \includegraphics[scale=0.432]{Images/Toolbars}¶
  \caption{Toolbar Alternatives}¶
\end{figure}¶
¶
Try them and see which one best fits your needs. They can be further
customized by adding or removing a particular button. In WinEdt's Help ->
Index, type the keyword Toolbar Alternatives and you will learn how!¶
¶
\begin{figure}[htbp]\label{fig:CustomWinEdt}¶
  \centering¶
  \includegraphics[scale=0.48]{Images/WinEdt-Custom}¶
  \caption{Customized WinEdt in action working on this document\ldots}¶
\end{figure}¶
¶
\section{\TeX\ AUX Output Folder}¶
¶
If you take a look at the folder where you unpacked this project you may be
pleasantly surprised to find that \emph{no} \TeX\ auxiliary and numerous
temporary files are polluting the source or the \WinEdtFile{Chapters}
folders. Your source folders are junk-free!¶
¶
In fact, all \TeX\ output files are stored in a \WinEdtFile{TeXAux} directory
that has been created for this purpose through the definition made in the
Execution Modes dialog. As a result, only your sources and compiled pdf and
synctex\footnote{This file is there only to allow you to use forward and
```

```
PDFTeXify Compilation Report (Pages: 43)

Errors: 0   Warnings: 0   Bad Boxes: 0
```

Figure 2.7: Customized WinEdt in action working on this document. . .

2.5 TEX AUX Output Folder

If you take a look at the folder where you unpacked this project you may be pleasantly surprised to find that *no* TEX auxiliary and numerous temporary files are polluting the source or the `Chapters` folders. Your source folders are junk-free!

In fact, all TEX output files are stored in a `TeXAux` directory that has been created for this purpose through the definition made in the Execution Modes dialog. As a result, only your sources and compiled pdf and synctex[10] files are placed next to the main file: *nice and neat!*[11]

2.6 Execution Modes

The settings and definitions in the Execution Modes dialog govern how your compilers and accessories (TEX and friends) associated with WinEdt will behave. To

[10] This file is there only to allow you to use forward and inverse search and it is not a part of your compiled pdf document. . .

[11] For the more discerning, this feature can be further configured in the TeX Options page of Execution Modes. As always, clicking on Help in that dialog will explain the details.

illustrate another practical use of the Execution Modes dialog in connection with this project, notice that its source folder, while clean of TeX auxiliary and temporary files, does contain two non-standard packages: `chapterthumb.sty` and `xrcs.sty`. If you need to share the project with other people that also don't have them installed, it's best to leave them in your source document folder.

If, on the other hand, you're working on your own project which *you don't need to share with anyone* (apart from the final result), *and* you want to keep your source folder as neat as possible, you can place those files in a `Packages` subfolder.

> But then TeX will not find them unless we inform it where to look for them. To do that, use the `TEXINPUTS` variable to let TeX know that `Packages` subfolder should be searched for such packages. In the Variables page of Execution Modes you can define `TEXINPUTS` as:
>
> ```
> .;./Packages//;
> ```
>
> Now you can move them into the `Packages` subfolder and TeX will find them without any other effort.

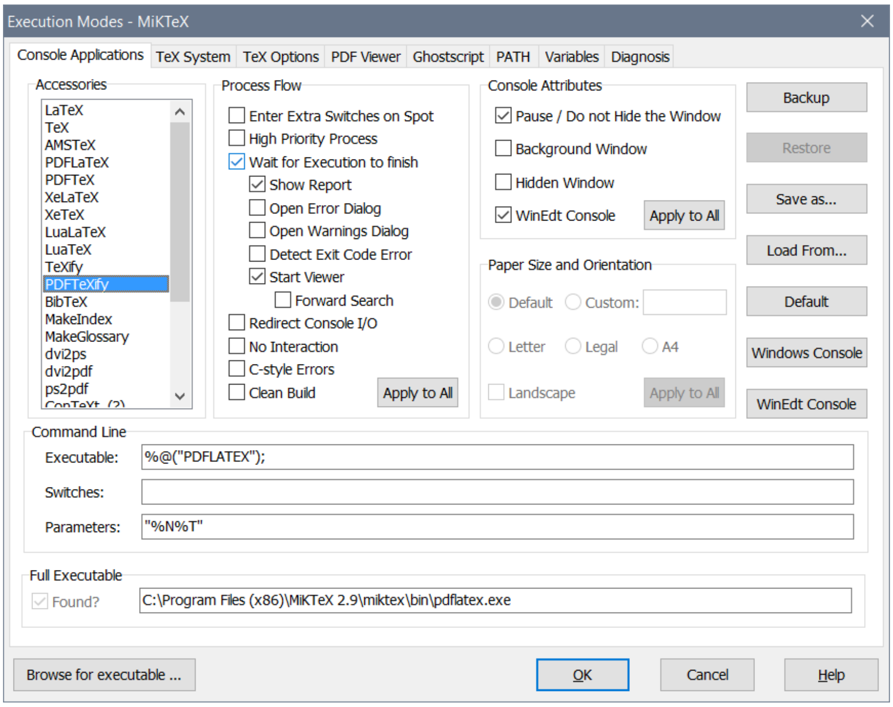
`TEXINPUTS` and `BIBINPUTS` *environment variables* can be used to incorporate your private (relative or fixed) locations for sty or bib files into TeX's input search path. If too many packages are installed using this method TeX will become slow but for a few private packages or bibliography databases this is an admissible and simple solution that will allow you to keep your source directory free of such files and also share bibliography databases among different projects.

The Execution Modes dialog offers many possibilities that you may have not been aware of. You are urged to become familiar with this dialog and take advantage of what can be done, from defining an alternative, TeX-friendly, PDF Viewer to checking that your TeX system is properly installed, and lots in between...

> *Help in the Execution Modes dialog is thorough and in-depth. It explains how WinEdt interacts with external accessories*[12] *and what available alternatives*[13] *can be easily incorporated in your custom version. If you are new to TeX-ing or WinEdt you will greatly benefit from reading the dozen* (*or so*) *illustrated pages of Help associated with this dialog...*

[12] Including and especially your TeX System and friends.

[13] Including PDF Viewers, pdf converters, etc...



Execution Modes - MiKTeX

Console Applications | TeX System | TeX Options | PDF Viewer | Ghostscript | PATH | Variables | Diagnosis

Accessories: LaTeX, TeX, AMSTeX, PDFLaTeX, PDFTeX, XeLaTeX, XeTeX, LuaLaTeX, LuaTeX, TeXify, PDFTeXify (selected), BibTeX, MakeIndex, MakeGlossary, dvi2ps, dvi2pdf, ps2pdf, ConTeXt (?)

Process Flow
- □ Enter Extra Switches on Spot
- □ High Priority Process
- ☑ Wait for Execution to finish
  - ☑ Show Report
  - □ Open Error Dialog
  - □ Open Warnings Dialog
  - □ Detect Exit Code Error
  - ☑ Start Viewer
    - □ Forward Search
- □ Redirect Console I/O
- □ No Interaction
- □ C-style Errors
- □ Clean Build

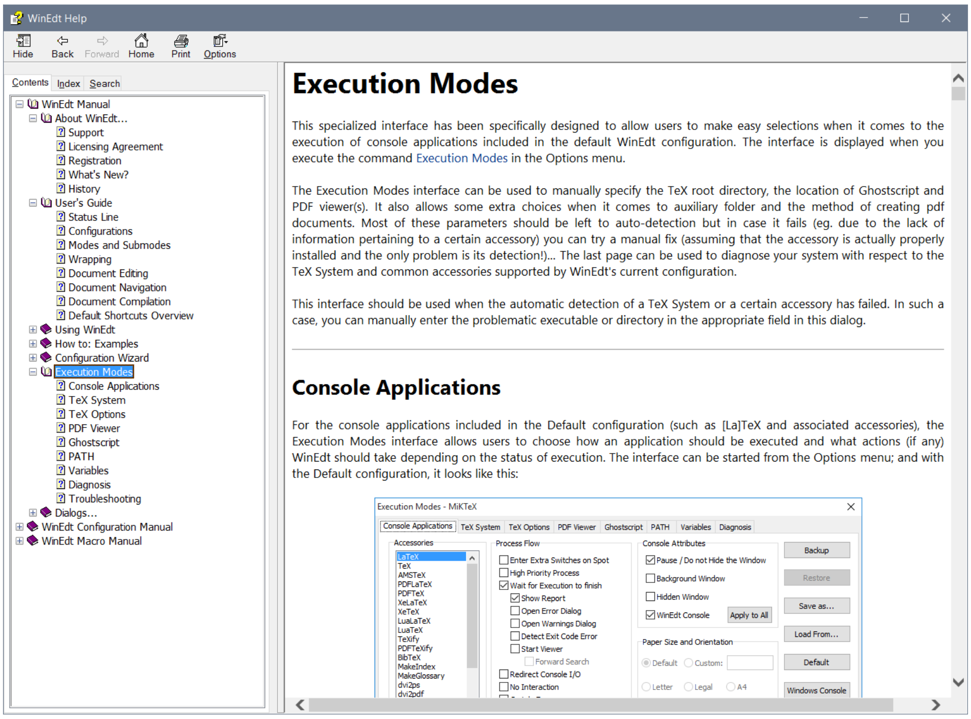
Console Attributes
- ☑ Pause / Do not Hide the Window
- □ Background Window
- □ Hidden Window
- ☑ WinEdt Console

Paper Size and Orientation
- ● Default ○ Custom:
- ○ Letter ○ Legal ○ A4
- □ Landscape

Command Line
- Executable: %@("PDFLATEX");
- Switches:
- Parameters: "%N%T"

Full Executable
- ☑ Found? C:\Program Files (x86)\MiKTeX 2.9\miktex\bin\pdflatex.exe

Figure 2.8: Execution Modes Dialog



WinEdt Help

- WinEdt Manual
  - About WinEdt...
    - Support
    - Licensing Agreement
    - Registration
    - What's New?
    - History
  - User's Guide
    - Status Line
    - Configurations
    - Modes and Submodes
    - Wrapping
    - Document Editing
    - Document Navigation
    - Document Compilation
    - Default Shortcuts Overview
  - Using WinEdt
  - How to: Examples
  - Configuration Wizard
  - Execution Modes
    - Console Applications
    - TeX System
    - TeX Options
    - PDF Viewer
    - Ghostscript
    - PATH
    - Variables
    - Diagnosis
    - Troubleshooting
  - Dialogs...
- WinEdt Configuration Manual
- WinEdt Macro Manual

Execution Modes

This specialized interface has been specifically designed to allow users to make easy selections when it comes to the execution of console applications included in the default WinEdt configuration. The interface is displayed when you execute the command Execution Modes in the Options menu.

The Execution Modes interface can be used to manually specify the TeX root directory, the location of Ghostscript and PDF viewer(s). It also allows some extra choices when it comes to auxiliary folder and the method of creating pdf documents. Most of these parameters should be left to auto-detection but in case it fails (eg. due to the lack of information pertaining to a certain accessory) you can try a manual fix (assuming that the accessory is actually properly installed and the only problem is its detection!)... The last page can be used to diagnose your system with respect to the TeX System and common accessories supported by WinEdt's current configuration.

This interface should be used when the automatic detection of a TeX System or a certain accessory has failed. In such a case, you can manually enter the problematic executable or directory in the appropriate field in this dialog.

Console Applications

For the console applications included in the Default configuration (such as [La]TeX and associated accessories), the Execution Modes interface allows users to choose how an application should be executed and what actions (if any) WinEdt should take depending on the status of execution. The interface can be started from the Options menu; and with the Default configuration, it looks like this:

Figure 2.9: Execution Modes: Context-sensitive Help

3 WinEdt Help System

As already heavily hinted, WinEdt comes with extensive, detailed, and up-to-date documentation in its on-line HTML-Help manual. It is easily opened by pressing `F1`, and here you will find the description of all available WinEdt options and macro functions. Help is also indexed[1] to allow quick access to certain topics that might be of interest to you. You should find it definitely worthwhile to spend a few minutes investigating what's there and how it's organized.

However, with the possible exception of a few sections in the WinEdt Manual, the on-line documentation was not written as a gentle introduction for first-time users. This guide is an attempt to be just that and to provide examples and hands-on guidance for setting up TEX projects with WinEdt as the center of control. In this guide we have tried to avoid duplicating information that is already available in the Help Manual.

> *And as you have seen, you don't have to study WinEdt's documentation in order to start working on your TEX documents. The program is ready to go, by default configured for MiKTEX or TEX Live accessories.*

That being said let us briefly describe what you *will* find in WinEdt's Help. It is organized in three parts as described below, starting with the most accessible one.

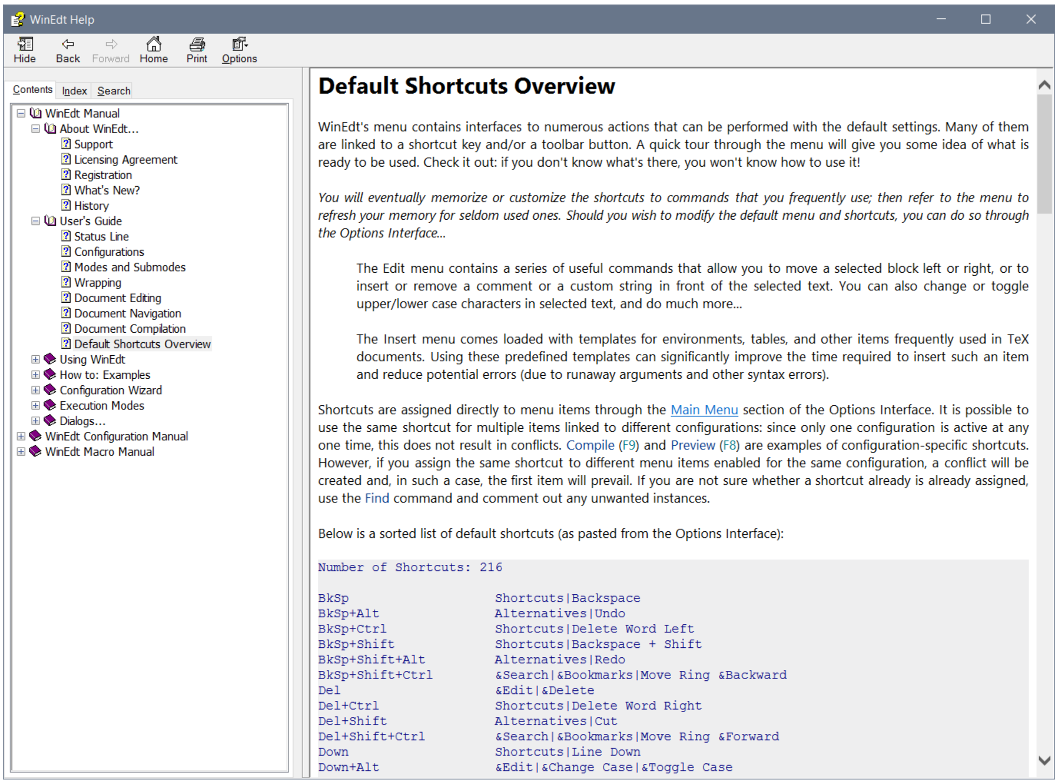
3.1 WinEdt Manual

Among the main topics in the Manual you will see the User's Guide, which explains some important concepts (such as modes, configurations, active strings) that make WinEdt more than just a simple (Notepad-like) editor with a few TEX-related buttons. Many sections in this part are heavily illustrated and written with a first-time user in mind.[2] The User's Guide section of the Manual also contains a section, Default Shortcuts Overview, that lists all keyboard shortcuts used in the default settings.[3]

[1] The Index can be used to quickly find the information linked to a common keyword. If you still cannot find the relevant information, use the third tab page of the navigation bar to perform a full search through the manual. Do not expect WinEdt's documentation to deal with MiKTEX- or LATEX-specific issues. Such information can be found in your TEX System's doc folder.

[2] Still, if (for the time being) you are not interested in such functionality you can ignore these concepts and simply use WinEdt as is.

[3] Shortcuts, too, are customizable...



Default Shortcuts Overview

WinEdt's menu contains interfaces to numerous actions that can be performed with the default settings. Many of them are linked to a shortcut key and/or a toolbar button. A quick tour through the menu will give you some idea of what is ready to be used. Check it out: if you don't know what's there, you won't know how to use it!

You will eventually memorize or customize the shortcuts to commands that you frequently use; then refer to the menu to refresh your memory for seldom used ones. Should you wish to modify the default menu and shortcuts, you can do so through the Options Interface...

> The Edit menu contains a series of useful commands that allow you to move a selected block left or right, or to insert or remove a comment or a custom string in front of the selected text. You can also change or toggle upper/lower case characters in selected text, and do much more...
>
> The Insert menu comes loaded with templates for environments, tables, and other items frequently used in TeX documents. Using these predefined templates can significantly improve the time required to insert such an item and reduce potential errors (due to runaway arguments and other syntax errors).

Shortcuts are assigned directly to menu items through the Main Menu section of the Options Interface. It is possible to use the same shortcut for multiple items linked to different configurations: since only one configuration is active at any one time, this does not result in conflicts. Compile (F9) and Preview (F8) are examples of configuration-specific shortcuts. However, if you assign the same shortcut to different menu items enabled for the same configuration, a conflict will be created and, in such a case, the first item will prevail. If you are not sure whether a shortcut already is already assigned, use the Find command and comment out any unwanted instances.

Below is a sorted list of default shortcuts (as pasted from the Options Interface):

```
Number of Shortcuts: 216

BkSp                   Shortcuts|Backspace
BkSp+Alt               Alternatives|Undo
BkSp+Ctrl              Shortcuts|Delete Word Left
BkSp+Shift             Shortcuts|Backspace + Shift
BkSp+Shift+Alt         Alternatives|Redo
BkSp+Shift+Ctrl        &Search|&Bookmarks|Move Ring &Backward
Del                    &Edit|&Delete
Del+Ctrl               Shortcuts|Delete Word Right
Del+Shift              Alternatives|Cut
Del+Shift+Ctrl         &Search|&Bookmarks|Move Ring &Forward
Down                   Shortcuts|Line Down
Down+Alt               &Edit|&Change Case|&Toggle Case
```

Figure 3.1: WinEdt's On-line HTML Help System

There are Help topics for the Configuration Wizard, Execution Modes and Preferences dialogs, all of which can be used to accomplish the most commonly requested customizations.[4] And if you scroll through the menus, you will see many other dialogs, such as Search Menu -> Find dialog, or Tool -> Spell Checking dialog, which offer options for you to enable your preferred strategies. . .

> These and other WinEdt dialogs provide a context-sensitive help. Press the Help button in such dialogs and you'll get relevant information pertaining to the contents of the dialog and the meaning of the available options. This information can be also accessed through the navigation bar (Dialogs. . .) in the WinEdt Manual. Again, you don't have to read it until you encounter the need to change something in the defaults. . .

Dialogs aside, managing the majority of WinEdt options and customizing the program to any specific needs you have, will require the use of the Options Interface. . .

[4] It is strongly recommended that you take a look at those dialogs in order to get an idea of what can be easily accomplished. In particular, the Diagnosis tab page in the Execution Modes dialog will help you correctly diagnose and fix any problems with external accessories (such as your TEX System).

3.2 Configuration Manual

The Options Interface has all the branches of WinEdt's customizable configurations:

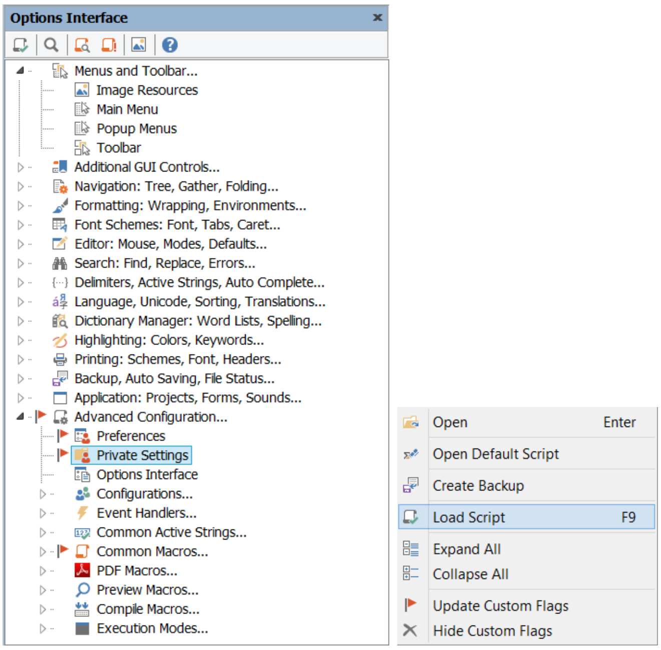

Options Interface
Menus and Toolbar...
Image Resources
Main Menu
Popup Menus
Toolbar
Additional GUI Controls...
Navigation: Tree, Gather, Folding...
Formatting: Wrapping, Environments...
Font Schemes: Font, Tabs, Caret...
Editor: Mouse, Modes, Defaults...
Search: Find, Replace, Errors...
Delimiters, Active Strings, Auto Complete...
Language, Unicode, Sorting, Translations...
Dictionary Manager: Word Lists, Spelling...
Highlighting: Colors, Keywords...
Printing: Schemes, Font, Headers...
Backup, Auto Saving, File Status...
Application: Projects, Forms, Sounds...
Advanced Configuration...
Preferences
Private Settings
Options Interface
Configurations...
Event Handlers...
Common Active Strings...
Common Macros...
PDF Macros...
Preview Macros...
Compile Macros...
Execution Modes...
Open Enter
Open Default Script
Create Backup
Load Script F9
Expand All
Collapse All
Update Custom Flags
Hide Custom Flags

Figure 3.2: WinEdt Options Interface

The Configuration Manual explains how to use this interface, lists all configuration sections, and describes their properties. You should read the preface and possibly the Introduction to the Configuration Manual before you attempt customizations through the Options Interface.[5]

> As mentioned, the most commonly required customizations - such as wrapping, font, etc. - can be done through the Options -> Preferences dialog. However, advanced customizations - such as adding menu items, making changes to the toolbar, adding dictionaries - have to be done through the Options Interface, for which the Configuration Manual should be consulted.

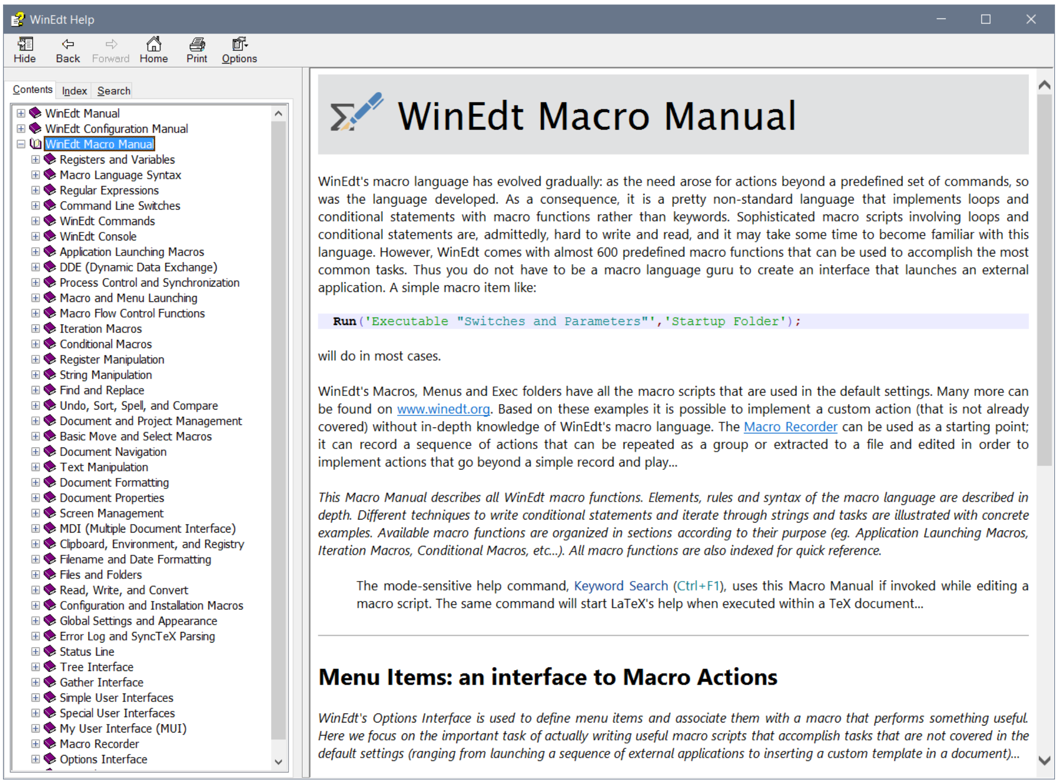
[5] Many users are initially intimidated by the numerous options and possibilities and are afraid of breaking something. However, if you approach it with an open mind you will soon realize that this is actually a very good way to manage your customizations even if you are not a WinEdt expert. Should something go wrong the Options Menu -> Maintenance menu has a command Rebuild Default which will restore the default settings; you can then return and fix any mistake and then use Rebuild Local settings.

3.3 Macro Manual

Additional information for more demanding and advanced users is available through the Macro Manual. The manual explains the syntax of WinEdt's macro language and describes all available functions, together with their parameters. The topics in the Macro Manual also cover some other advanced issues, most notably:

- Command Line Switches
- Registers and Variables (such as %N, %T, %P. . .)
- Regular Expressions
- etc. . .

An in-depth understanding of WinEdt's macro language is *not* required to add extra accessories or slightly modify the existing behavior. Numerous examples in this manual explain how. Comments and existing definitions in the configuration files also provide additional guidance.



- WinEdt Manual
- WinEdt Configuration Manual
- WinEdt Macro Manual
  - Registers and Variables
  - Macro Language Syntax
  - Regular Expressions
  - Command Line Switches
  - WinEdt Commands
  - WinEdt Console
  - Application Launching Macros
  - DDE (Dynamic Data Exchange)
  - Process Control and Synchronization
  - Macro and Menu Launching
  - Macro Flow Control Functions
  - Iteration Macros
  - Conditional Macros
  - Register Manipulation
  - String Manipulation
  - Find and Replace
  - Undo, Sort, Spell, and Compare
  - Document and Project Management
  - Basic Move and Select Macros
  - Document Navigation
  - Text Manipulation
  - Document Formatting
  - Document Properties
  - Screen Management
  - MDI (Multiple Document Interface)
  - Clipboard, Environment, and Registry
  - Filename and Date Formatting
  - Files and Folders
  - Read, Write, and Convert
  - Configuration and Installation Macros
  - Global Settings and Appearance
  - Error Log and SyncTeX Parsing
  - Status Line
  - Tree Interface
  - Gather Interface
  - Simple User Interfaces
  - Special User Interfaces
  - My User Interface (MUI)
  - Macro Recorder
  - Options Interface

WinEdt Macro Manual

WinEdt's macro language has evolved gradually: as the need arose for actions beyond a predefined set of commands, so was the language developed. As a consequence, it is a pretty non-standard language that implements loops and conditional statements with macro functions rather than keywords. Sophisticated macro scripts involving loops and conditional statements are, admittedly, hard to write and read, and it may take some time to become familiar with this language. However, WinEdt comes with almost 600 predefined macro functions that can be used to accomplish the most common tasks. Thus you do not have to be a macro language guru to create an interface that launches an external application. A simple macro item like:

```
Run('Executable "Switches and Parameters"','Startup Folder');
```

will do in most cases.

WinEdt's Macros, Menus and Exec folders have all the macro scripts that are used in the default settings. Many more can be found on www.winedt.org. Based on these examples it is possible to implement a custom action (that is not already covered) without in-depth knowledge of WinEdt's macro language. The Macro Recorder can be used as a starting point; it can record a sequence of actions that can be repeated as a group or extracted to a file and edited in order to implement actions that go beyond a simple record and play...

This Macro Manual describes all WinEdt macro functions. Elements, rules and syntax of the macro language are described in depth. Different techniques to write conditional statements and iterate through strings and tasks are illustrated with concrete examples. Available macro functions are organized in sections according to their purpose (eg. Application Launching Macros, Iteration Macros, Conditional Macros, etc...). All macro functions are also indexed for quick reference.

> The mode-sensitive help command, Keyword Search (Ctrl+F1), uses this Macro Manual if invoked while editing a macro script. The same command will start LaTeX's help when executed within a TeX document...

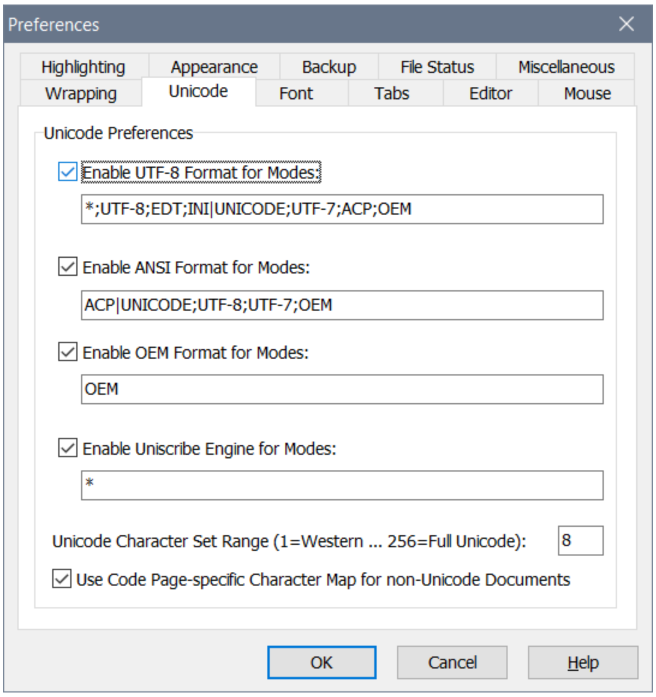
Menu Items: an interface to Macro Actions

WinEdt's Options Interface is used to define menu items and associate them with a macro that performs something useful. Here we focus on the important task of actually writing useful macro scripts that accomplish tasks that are not covered in the default settings (ranging from launching a sequence of external applications to inserting a custom template in a document)...

Figure 3.3: WinEdt Macro Manual

4 WinEdt and Unicode (UTF-8) encoding

WinEdt is a unicode editor with support for UTF-8 or code page-specific encoding. UTF-8 is the default format for TEX documents. This can be configured through the Unicode section of the Options Interface - or through the Unicode page in the Preferences dialog (Help explains the details):



Unicode Preferences

☑ Enable UTF-8 Format for Modes:

*;UTF-8;EDT;INI|UNICODE;UTF-7;ACP;OEM

☑ Enable ANSI Format for Modes:

ACP|UNICODE;UTF-8;UTF-7;OEM

☑ Enable OEM Format for Modes:

OEM

☑ Enable Uniscribe Engine for Modes:

*

Unicode Character Set Range (1=Western ... 256=Full Unicode): 8

☑ Use Code Page-specific Character Map for non-Unicode Documents

Figure 4.1: Preferences: Unicode

UTF-8 is the best choice for TEX documents encoding. However, if you have to work with legacy documents that were created in your default code page-specific format WinEdt will treat such documents properly and will preserve their encoding. Users that have to deal with code pages that are not native to their version of Windows will have to use the CP converter to tell WinEdt how to treat such documents. WinEdt is capable of handling both Unicode and (legacy) code page-specific documents.

4.1 Code Page Converter

If a document's mode ends with the submode :CPnum then the indicated code page is used to load the file in unicode format. For example: TeX:CP1251 uses Cyrillic code page.

A comment in the beginning of a TEX document:

```
% !Mode:: "TeX:UTF-8"
```

will ensure that a document is properly loaded and saved. A similar convention is used by emacs:

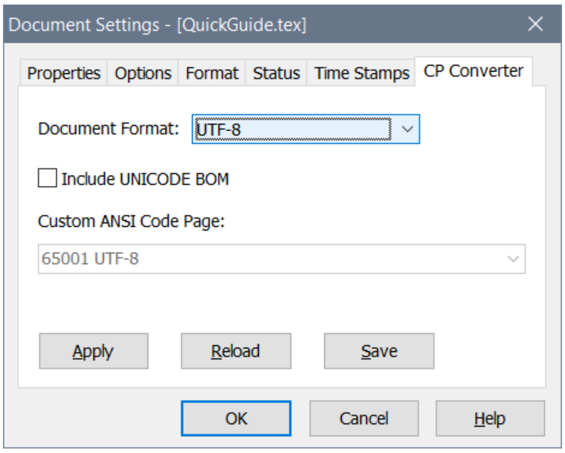
```
% -*-coding: utf-8 -*-
```

WinEdt understands emacs coding directive for UTF-8.

Some unicode or UTF-8 documents start with a Byte Order Mark (BOM). Unicode-aware applications can determine the coding of a document from its BOM. Windows Notepad always includes BOM in unicode or UTF documents. Unfortunately, the BOM signature also causes problems with many applications and compilers (including TEX with UTF-8 encoding) and that makes it rather useless. . .

Without BOM and without any convention as described above it is sometimes hard to distinguish between UTF-8 and ANSI (code page-specific) documents.

The Document Settings dialog has a CP Converter tab page. It can be used to change a document's format or reload the document in the proper code page in the rare case of WinEdt not being able to automatically determine the correct encoding.



Document Settings - [QuickGuide.tex]

Document Format: UTF-8

□ Include UNICODE BOM

Custom ANSI Code Page: 65001 UTF-8

Figure 4.2: Document Settings: Code Page Converter

Help in the dialog explains how to use this functionality.

4.2 TEX and International Characters (UTF-8)

Putting

```
\usepackage[utf8]{inputenc}
```

enables you to use UTF-8 (unicode) coding in LATEX documents. As long as you open the document in WinEdt in UTF-8 mode you see the same characters in WinEdt as in your compiled document (as is the case with this UTF-8 document):

```
À Á Â Ã Ä Å Æ Ç È É Ê Ë Ì Í Î Ï Ñ Ò Ó Ô Õ Ö Ø Œ Ù Ú Û Ü Ý Ÿ Š ß ¡
à á â ã ä å æ ç è é ê ë ì í î ï ñ ò ó ô õ ö ø œ ù ú û ü ý ÿ š € ¿
† ‡ § ¶ © ® Č Š Ž č š ž
```

À Á Â Ã Ä Å Æ Ç È É Ê Ë Ì Í Î Ï Ñ Ò Ó Ô Õ Ö Ø Œ Ù Ú Û Ü Ý Ÿ Š ß ¡
à á â ã ä å æ ç è é ê ë ì í î ï ñ ò ó ô õ ö ø œ ù ú û ü ý ÿ š € ¿
† ‡ § ¶ © ® Č Š Ž č š ž

Not all UTF-8 characters are currently supported by LATEX unless you load extra packages. For example the € (€) symbol requires:

```
\usepackage{textcomp} % required for \texteuro
\usepackage{eurosym}  % required for \euro
% get a "nicer" looking euro symbol:
%\let\texteuro\euro   % if you want \texteuro=\euro
```

Note the difference between the shape of the `\euro` (€) and `\texteuro` (€) symbols. Such issues are non-WinEdt related and you will have to consult TEX's documentation or, if needed, seek help on the appropriate forum (such as TEX Newsgroup where LATEX related topics are discussed).

You might have noticed that the preamble of this document also contains:

```
\catcode‘\•=13
\def•{•} % WinEdt bullet (U+007F)-> Unicode Bullet (U+2022)
```

This allows TEX to process an empty tabular environment as inserted by WinEdt's auto-completion functionality. Bullets are represented by •:

•	•	•	•	•
•	•	•	•	•
•	•	•	•	•
•	•	•	•	•

In WinEdt the shortcut Ctrl+Space (*Tools menu ->* Next Bullet) *lets you move through placeholders and fill in the actual data.*

4.3 Translation Tables

If you prefer your documents to contain plain TEX notation for international characters (eg. `\'{A}` stands for À) then you should consider applying WinEdt's read and write translation tables. This will make working with WinEdt more comfortable and it is required if you want to take advantage of WinEdt's spell checking ability with international dictionaries. UTF-8 encoding is a better solution in most cases!

WinEdt can convert certain strings into their unicode equivalents when the file is being read and then translate these characters back to the original strings representing international characters in TEX notation.

Suitable translation tables for TeX mode are already defined (but not enabled) in the default settings: see Options Interface. The help in this interface provides the details.

For example, the default `TeX_Read` and `TeX_Write` translation tables contain definitions like:

```
"{\ss}" -> "ß"                    "ß" -> "{\ss}"
"{\AA}" -> "Å"                    "Å" -> "{\AA}"
"{\AE}" -> "Æ"                    "Æ" -> "{\AE}"
"{\aa}" -> "å"                    "å" -> "{\aa}"
"{\ae}" -> "æ"                    "æ" -> "{\ae}"
"{\OE}" -> "Œ"                    "Œ" -> "{\OE}"
"{\oe}" -> "œ"                    "œ" -> "{\oe}"
"{\O}" -> "Ø"                     "Ø" -> "{\O}"
"{\o}" -> "ø"                     "ø" -> "{\o}"
"\c{C}" -> "Ç"                    "Ç" -> "\c{C}"
"\c{c}" -> "ç"                    "ç" -> "\c{c}"
"\^{A}" -> "Â"                    "Â" -> "\^{A}"
"\~{A}" -> "Ã"                    "Ã" -> "\~{A}"
"\""{A}" -> "Ä"                   "Ä" -> "\""{A}"
                     ...
```

Note that the last item is not a typo! To specify double quotes inside a double-quoted string they have to be repeated twice! Failing to observe this convention may completely corrupt WinEdt's translation table.

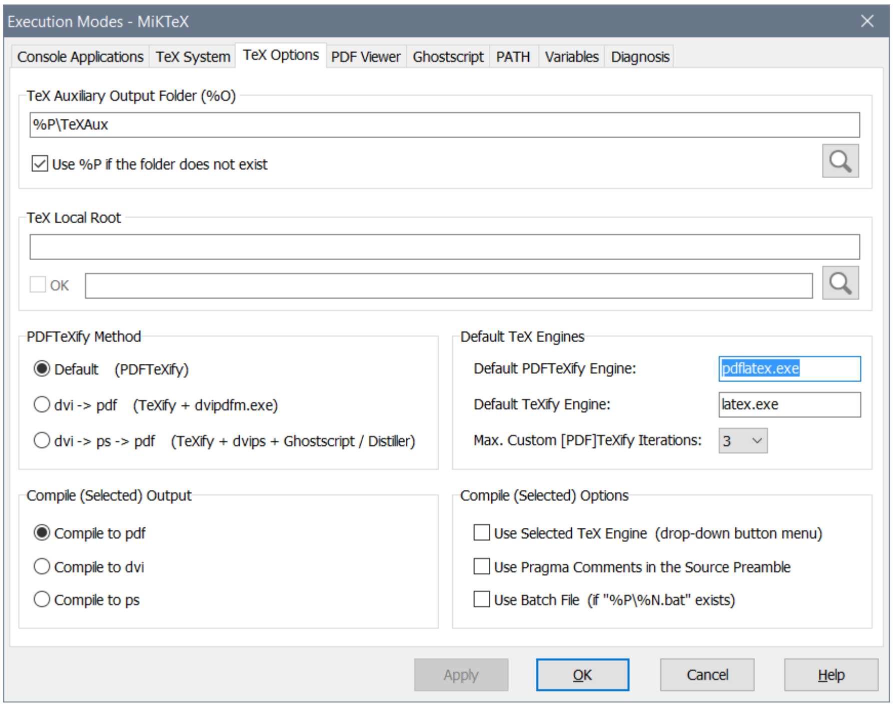
The read translation table supports two notations (eg. `\^{A}` and `{\^A}`). The write translation table `TeX_Write` is the inverse of the read translation table (except that it uses the first, preferable, notation where applicable). You should use translation tables with some care: make a backup copy of your documents until you verify that the tables are set up correctly. Careless application of translation tables may irreversibly corrupt your documents (just like a global replace)!

4.4 TEX and UTF-8

While WinEdt supports unicode pretty well[1] this is not necessarily the case for all TEX engines. For example, pdfTeX is not a unicode application and supports only limited unicode ranges, which may be enough for most international users that use the Latin alphabet but certainly is not sufficient for CJK or Middle East users. But any TEX engine's limitation in this respect is not a WinEdt-related issue and you should seek information or help with this on LATEX forums.

TEX Systems such as MiKTEX and TEX Live include alternative TEX engines that have a better (or different) approach to handling UTF-8 documents containing characters outside the range supported by pdfTeX. And WinEdt provides an easy interface to alternative TEX engines such as XeTeX and LuaTeX. All you do is select your alternative from the drop-down button that lists different TEX compilers. Once an alternative has been selected you do not have use the drop-down portion of the button to activate the selected compiler until you want to change it again.

Furthermore, if you want to make XeLaTeX or LuaLaTeX your default PDFTeXify engine you can easily do so in the TeX Options page of the Execution Modes dialog.



TeX Auxiliary Output Folder (%O)

%P\TeXAux

☑ Use %P if the folder does not exist

TeX Local Root

□ OK

PDFTeXify Method

● Default (PDFTeXify)

○ dvi -> pdf (TeXify + dvipdfm.exe)

○ dvi -> ps -> pdf (TeXify + dvips + Ghostscript / Distiller)

Default TeX Engines

Default PDFTeXify Engine: pdflatex.exe

Default TeXify Engine: latex.exe

Max. Custom [PDF]TeXify Iterations: 3

Compile (Selected) Output

● Compile to pdf

○ Compile to dvi

○ Compile to ps

Compile (Selected) Options

□ Use Selected TeX Engine (drop-down button menu)

□ Use Pragma Comments in the Source Preamble

□ Use Batch File (if "%P\%N.bat" exists)

Figure 4.3: Execution Modes: TeX Options

[1] WinEdt uses the powerful MS Uniscribe library with support for bidirectional text and complex text processing.

4.5 Modes and Submodes

A document's mode is initially determined from its filetype and is stored as a local attribute of the file in WinEdt's File List (Project File). This works in most cases for the main mode. However, bilingual users might want to tie certain attributes (such as dictionaries) to submodes that may not be apparent from the filetype.

Instead of setting such modes through the Document Settings dialog or adopting a practice to name your files with more than one filetype (eg. `Paper.fr.tex`) it is possible to enter submodes (as comments) in the first (or second) line of a document.

WinEdt modes can be specified as a comment:

```
% !Mode:: "Mode:Submode:Submode"
```

WinEdt also recognizes mode specification as used by emacs:

```
% -*- mode: TeX -*-
% -*- coding: utf-8 -*-
```

It is also possible to specify mode and submode in a single comment:

```
% -*- TeX:DE:Soft:UTF-8 -*-
```

However, as Emacs might not recognize such specification, it is better to use WinEdt's convention `!Mode:: "TeX:DE:Soft:UTF-8"` as described above.

Furthermore, for TEX documents WinEdt also detects the language submode from `babel` and UTF-8 coding from the `inputenc` package:

```
// Determine Language Submodes from babel:
// \usepackage[french,german,italian,spanish]{babel}

// Determine Coding (UTF-8) from the preamble:
// \usepackage[utf8]{inputenc}
```

This functionality is implemented through event handler macros that are executed before a document is loaded into WinEdt, and which ensure that WinEdt opens and treats the document properly.[2] The actual macro that is by default called from this event handler is

```
%b\Macros\Events\GetMode.edt
```

If for some reason mode detection (or some portion of it) from comments is unwanted for your style of work you can edit this macro and comment out unwanted portions or make any other desirable changes. However, note that this macro is executed frequently and it has to be fast or else you'll notice delays when opening documents or even when collecting data in previously unopened documents. . .

[2] Event handlers are defined in the Advanced section of the Options Interface. Modes and submodes are explained further in WinEdt's Help Manual -> User Guide. . .

4.6 Spell Checking

WinEdt is by default configured to use English dictionaries. Bilingual users may immediately stumble upon the problem of using different dictionaries depending on their preferred or alternative languages. Other such dictionaries are available on www.winedt.org, and the *"How to..."* section in WinEdt's Help explains how to download, install, and configure additional dictionaries. With properly used modes and submodes WinEdt can be configured for multi-lingual users.

WinEdt underlines misspelled and (another common typo) repeated words. Right-clicking on such a word brings up a spelling dialog that offers suitable suggestions based on the currently enabled dictionaries. Right-clicking on a repeated word prompts you with the option to remove the duplicate. It is also possible to spell check the whole document (or a selected portion of it) or to manually highlight the next or previous typing mistake by using the items in the Tools Menu and their associated shortcuts.

WinEdt is often used to edit documents containing markup languages such as TeX. Not all environments in such documents should be subject to spell-checking as they may not contain plain text. WinEdt can be configured to turn spell checking off in such places. By default it is already loaded with an extensive list of such exceptions for TeX documents. More can be added by users with special needs. . .

WinEdt Dictionaries (Word Lists)

Any word list (a text file consisting of one word per line) can be converted into a WinEdt dictionary by sorting it as a dictionary (Sort Lines dialog) and saving it as a unicode (UTF-16) file with BOM and UNIX line terminators (Document Settings dialog). Neglecting to properly sort and save a dictionary will result in increased loading time and possible problems with the spell checking functionality! Words have to be at most 255 characters long and a dictionary can contain at most 16,000,000 lines. These are reasonable limitations for all situations arising in practice. Lines at the beginning of a dictionary starting with % characters are considered comments and are not subject to sorting. That's all. The hard part is compiling a complete word list for any available language.[3]

WinEdt's dictionaries use unicode (UTF-16) format with BOM and LF as line terminator. This means that WinEdt 6 (or earlier) ANSI dictionaries are not compatible with the unicode versions of WinEdt. However, the CP Converter section of the Documents Settings dialog makes it is easy to convert any non-unicode dictionary into the new format. Help in the Document Settings dialog gives the details, showing, as an example, how conversion of the Russian dictionary into a unicode (ready to use) format was done. Of course, all dictionaries included in the default settings are already properly formatted and ready to use!

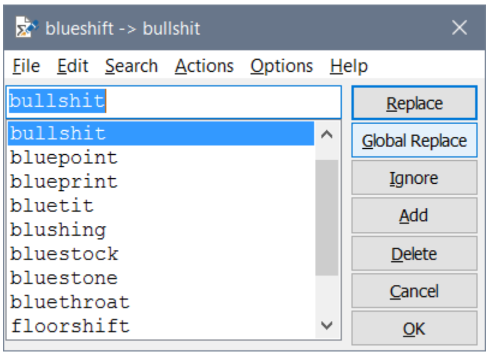
[3] See the cautionary tale below.

When adding a word to the dictionary or performing a Global Replace while spell checking a document one is strongly advised to pay attention. Here is a true cautionary tale illustrating how things can and did go wrong.

Cosmic *bullshit* or a *bluefish* universe?

We'll omit the names and places, but let's say a well known astronomer was writing an article on *blueshift*. The word *blueshift* is not in WinEdt's dictionaries. This would not be a real problem for us simple down-to-earth types, as it is easy to add a word to the User Dictionary during spell checking (eg. by right-clicking on an underlined word). This could and should be the end of it...

Alas, apparently this is not how our universe works. For reasons that could not be completely fathomed by our less sophisticated minds, the esteemed user in question decided to perform a Global Replace with the first suggestion offered by WinEdt. The picture is worth a thousand words:



```
bullshit

bullshit
bluepoint
blueprint
bluetit
blushing
bluestock
bluestone
bluethroat
floorshift
```

Figure 4.4: *Blueshift* gone bad...

The nuisance of an underlined (misspelled) word was gone and no further attention was paid to the issue. That is, until the final draft was received by a (I can only assume somewhat puzzled) co-author. At first, I received an angry email about the whole incident. However, after some explanations and reflection on the actions taken, we eventually shared a laugh. The problem was easily fixed by replacing the word *bullshit* back to *blueshift* and this saved the day without the potential embarrassment that would have resulted had the "spell-checked" article been submitted to higher authorities for consideration.

Since those days, the algorithm used by WinEdt to compile a list of suggestions for misspelled words has been improved and refined (*bullshit* and *bluefish* are no longer the top two suggestions in this case). However, the first suggestion is not always the correct one: especially if the correct word is not present in WinEdt dictionaries (word lists). Keep this in mind when performing spelling replacements!

5 LaTeX Demo: Non-WinEdt-Related Bonus

5.1 Graphics Inclusion

Graphics inclusion in TeX documents is not WinEdt-related and the documentation that comes with your TeX System (eg. `graphicx` package) should be consulted. However, below are a few examples that show that it can be done! These work with my (default) version of MiKTeX 2.9 compiled with PDFLaTeX but come with no guarantee and no support from the WinEdt Team.

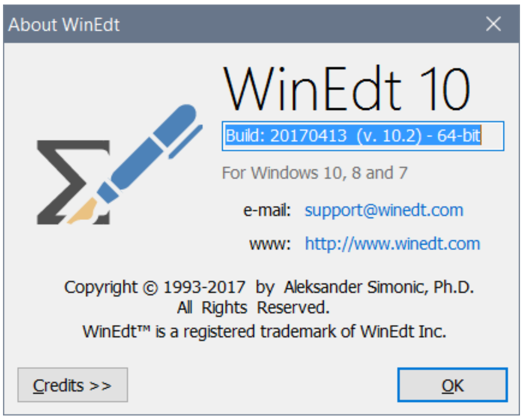
The first thing to know is that different graphic formats are supported by TeX depending on the intended output format (dvi or pdf). Here we will only focus on pdf because dvi format is seldom used these days and using intermediate formats like dvi and ps in order to create a pdf file via converters is not a good practice.[1] PDFLaTeX supports pdf, png, and jpeg graphics formats.

Figure 5.1: WinEdt classic and new-style logo in a suitable png format

[1] This document cannot be compiled to dvi format because we did not provide alternative (legacy) graphic formats like eps for the included images (on purpose!). Should you create an eps image for each graphic file in the Images folder you will be able to compile it to dvi format. But YAP (and other DVI viewers) will not properly display landscape pages or rotated tables. Although some users may disagree, using dvipdf or dvips+pspdf to create pdf files with eps images is not very efficient: eps images have to be converted to pdf every time the source is compiled. Converters usually degrade the graphics quality and can result in other problems such as misaligned or improperly placed images. It is much better to create images that can be included in pdf documents and convert non-supported formats once and forever! They are not supported because they are deemed obsolete and this will not change. If intermediate formats work for you that's fine. If not there is no point complaining to the WinEdt Team since now you know how we feel about it. Besides WinEdt is completely irrelevant to such issues...

Figure 5.2: SV Adriana docked

JPEG is a good format for pictures (non-vector graphics) like the one above. PNG format is suitable for vector graphics or computer screenshots like the one below:


About WinEdt

WinEdt 10

Build: 20170413 (v. 10.2) - 64-bit

For Windows 10, 8 and 7

e-mail: support@winedt.com

www: http://www.winedt.com

Copyright © 1993-2017 by Aleksander Simonic, Ph.D.
All Rights Reserved.
WinEdt™ is a registered trademark of WinEdt Inc.

Figure 5.3: About WinEdt Dialog

Now you know it can be done! Learn how to use `graphicx` *package if you need to do something more sophisticated…*

5.2 Color Package Example

This section is borrowed from MiKTeX's Samples folder. You should definitely consult more documentation and examples that come with your TeX System...

Text starts off in green a little red nested blue text returning to green

1. magenta cmyk black
2. predefined blue gray text

Black text on red background

Light background

Dark background

Black text, blue background, red frame

White text, blue background, red frame

This is how it is done:

```
\usepackage{color}
  ...
      \begin{enumerate}
        \item \textcolor[cmyk]{0,1,0,0}{magenta cmyk} black
        \item \color[gray]{0.5} \textcolor{blue}{predefined blue}
          gray text
      \end{enumerate}

      \definecolor{Light}{gray}{.80}
      \definecolor{Dark}{gray}{.20}
      \colorbox{red}{Black text on red background}
      \par\colorbox{Light}{%
        \textcolor{Dark}{Light background}}
      \par\colorbox{Dark}{%
        \textcolor{white}{Dark background}}

      \fcolorbox{red}{blue}{Black text,
            blue background, red frame}

      \fcolorbox{red}{blue}{\color{white}%
        White text, blue background, red frame}
```

5.3 Rotated objects examples

rotation = 30°				
1	2	3	4	5
A	B	C	D	E
a	b	c	d	e

rotation = 330°				
1	2	3	4	5
A	B	C	D	E
a	b	c	d	e

Table 5.1: Rotated tables

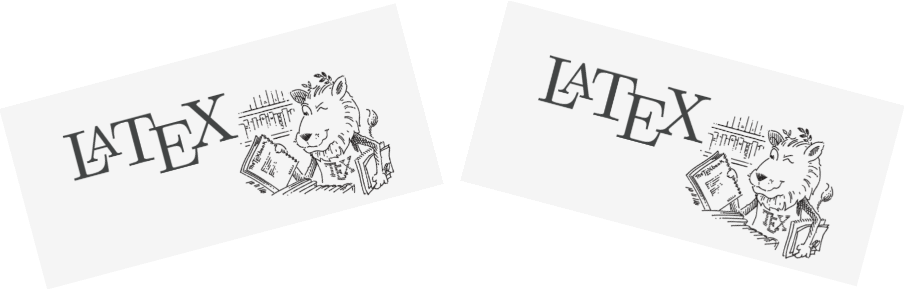


Figure 5.4: LaTeX Logo rotated by ±15°

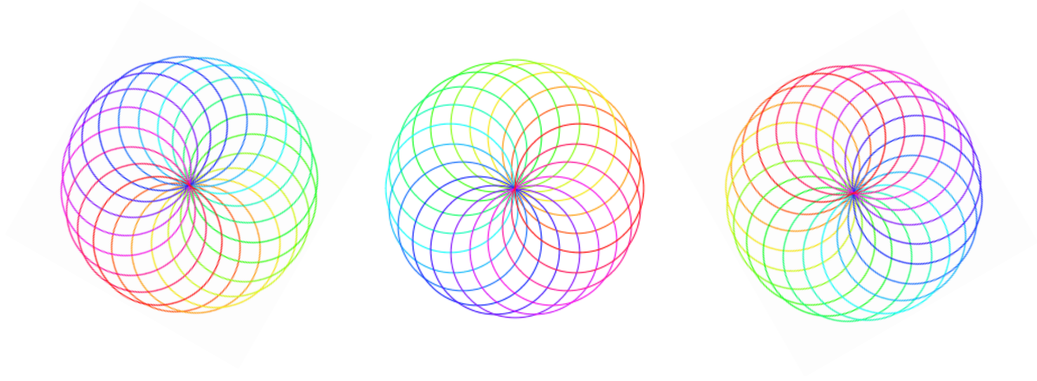


Figure 5.5: Rotated circles

Chapter 5

5.4 Landscape mode page example

The preamble must load the following package for this to work:

```
\usepackage{lscape} %landscape pages support
```

Name	*Base colors/notions*	*Parameter range*	*Default*
rgb	*red, green, blue*	$[0,1]^3$	
cmy	*cyan, magenta, yellow*	$[0,1]^3$	
cmyk	*cyan, magenta, yellow, black*	$[0,1]^4$	
hsb	*hue, saturation, brightness*	$[0,1]^3$	
Hsb	*hue°, saturation, brightness*	$[0,H] \times [0,1]^2$	$H = 360$
tHsb	*hue°, saturation, brightness*	$[0,H] \times [0,1]^2$	$H = 360$
gray	*gray*	$[0,1]$	
RGB	*Red, Green, Blue*	$\{0,1,\ldots,L\}^3$	$L = 255$
HTML	*RRGGBB*	$\{\texttt{000000},\ldots,\texttt{FFFFFF}\}$	
HSB	*Hue, Saturation, Brightness*	$\{0,1,\ldots,M\}^3$	$M = 240$
Gray	*Gray*	$\{0,1,\ldots,N\}$	$N = 15$
wave	*lambda* (nm)	$[363,814]$	

L, M, N are positive integers; H is a positive real number

Table 5.2: Table in landscape mode example (from `xcolor`'s documentation)

5.5 Another landscape page example

The preamble must load the following package for this to work:

```
\usepackage{lscape} %landscape pages support
```

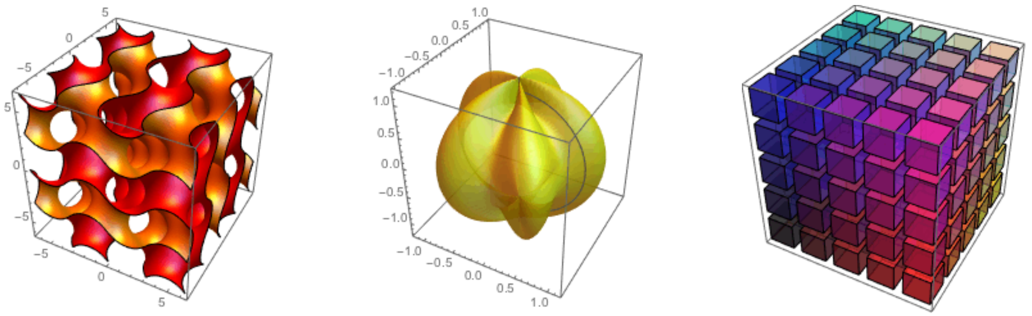


Figure 5.6: Mathematica 3D-objects in landscape mode

5.6 Presentations in LaTeX

Presentation packages and software that can be used with LaTeX are not WinEdt-related topics. However, since we frequently get asked about such things we posted a question to WinEdt's Mailing List and the response was overwhelming.

Most users were of the opinion that `beamer` is currently the best when it comes to ease of use and the quality of the output. Alternatives `texpower` and `seminar` have also been mentioned…

You can use MiKTeX's Package Manager to install `beamer`. MiKTeX's doc folder has all the documentation and examples that will help you start working on your own presentations. Some users mentioned that they had to upgrade their MiKTeX in order to be able to compile the examples, which rely on up-to-date packages. If you encounter any such problems you may have to do the same…

5.7 LaTeX and paper size

The best way to set the paper size in LaTeX documents is to use the `geometry` package. TeX itself does not have a notion of output paper size and this package is essential if you need to change the paper size.

The following will properly handle paper size in the printed document:

```
\usepackage[letterpaper]{geometry} % or a4paper
```

Some users use dvi2pdf or even dvips and then ps2pdf because these conversions seem to handle their choice of paper size correctly while PDFLaTeX does not. However, this does not make much sense as default paper size settings in these converters may change in future versions. Furthermore, and as already mentioned, using intermediate formats to produce a pdf document is not very efficient and tends to result in problems.

Learning how to use the `geometry` package is a much better way to handle paper size issues. And this package can do much more. You should read its documentation to learn about it. Type *geometry* in the interface that can be started from WinEdt's Help Menu -> LaTeX Doc (or Shift+Ctrl+F1 shortcut) and you can open the pdf manual describing this package[2] in detail.

[2] Both the package and its documentation are, of course, a part of your TeX system and not WinEdt! In fact WinEdt is completely irrelevant when it comes to such issues. If you need assistance you should seek help on TeX forums.

5.8 A Simple Revision Control System (RCS)

On www.winedt.org you'll find a link to the page that describes how to use RCS or CS-RCS with WinEdt. RCS (Revision Control System) is the proper way to deal with revisions…

However, simple revisions or corrections done by the copy editor and intended for the authors can be handled in a much simpler manner. WinEdt provides a sample LaTeX package `xrcs.sty` that can be used for such editing. The package defines two macros `\RCSAdd{...}` and `\RCSDel{...}`. These two macros can be used to mark simple additions and deletions, respectively. In WinEdt the environments are colored in blue and red (as defined in the Switches section of the Options Interface). Depending on the options the compiled document can contain additions and/or deletions (in color or plain text).

Furthermore, the package also provides a tag `\RCSMark{...}` which can be used to mark the argument with a yellow marker and `\RCSRem{...}` which can be used to include remarks. All four RCS tags are defined as switches in WinEdt's default highlighting scheme for TeX mode.

The `xrcs.sty` package provides the following options (with the default values displayed in red):

`active`	`inactive`
`marker`	`nomarker`
`remarks`	`noremarks`
`new`	`nonew`
`old`	`noold`

Examples of usage:

```
\usepackage[active,new,old,remarks,marker]{xrcs}
\usepackage[active]{xrcs} % Only Additions- in blue colors
\usepackagep[active,old,nonew]{xrcs} % Only old text - in red
\usepackage[nomarker}{xrcs} % Only Additions: final version
```

In your preamble you have to also include the `color` package:

```
\usepackage{color}
```

Text example:

```
\RCSMark{IMPORTANT:} WinEdt's \RCSDel{menu}\RCSRem{use capitals!}
\RCSAdd{Menu} should be thought of as an Action List...
```

with `\usepackage[active,new,noold,marker]{xrcs}` is processed as:

IMPORTANT: WinEdt's Menu should be thought of as an Action List…

Beside the highlighting definitions for switches `\RCS*{...}` WinEdt also has a popup menu Edt RCS containing some commands that can make the revisions easier. This popup menu is displayed in response to the Alt+R keystroke. The properties of the popup can be adjusted through the Options Interface (Popup Menus)...

The `xrcs.sty` file is included with this document's sources. Feel free to make changes and improvements... Note that the WinEdt Team does not provide support for this package: it's take it or leave it...

```
% -------------------------------------------------------------
% File: xrcs.sty
%
% A (very) simple Revision Control System for LaTeX2e/WinEdt
% *************************************************************
\NeedsTeXFormat{LaTeX2e}
\ProvidesPackage{xrcs}[2005/01/30 v0.002 RCS]
\RequirePackage{color}
\newif\ifMarker \Markertrue
\newif\ifRemarks\Remarkstrue
\newif\ifAddDel \AddDeltrue
\newif\ifAddNew \AddNewtrue
\newif\ifAddOld \AddOldfalse
\DeclareOption{active}{\AddDeltrue}
\DeclareOption{inactive}{\AddDelfalse}
\DeclareOption{marker}{\Markertrue}
\DeclareOption{nomarker}{\Markerfalse}
\DeclareOption{remarks}{\Remarkstrue}
\DeclareOption{noremarks}{\Remarksfalse}
\DeclareOption{new}{\AddNewtrue}
\DeclareOption{nonew}{\AddNewfalse}
\DeclareOption{old}{\AddOldtrue}
\DeclareOption{noold}{\AddOldfalse}
\ExecuteOptions{inactive,noold,noremarks,new,marker}
\ProcessOptions
% -------------------------------------------------------------
\def\RCSMark#1{\ifMarker{\colorbox{yellow}{#1}}\else#1\fi}
\def\RCSRem#1{\ifRemarks{\textsf{#1}}\fi}
\def\RCSDel#1{\ifAddOld\ifAddDel{\color{red}#1}\else#1\fi\fi}
\def\RCSAdd#1{\ifAddNew\ifAddDel{\color{blue}#1}\else#1\fi\fi}
%--------------------------------------------------------------
```

Once again, this is a very simplified revision system; it is somewhat primitive and it is lacking all the features available in proper RCS... However, it may be of some interest since it is very simple to use: in any text editor it is easy to search for `\RCS`...

5.9 Useful TEX-ing Hints

Check the source code of this document in WinEdt. Pay attention to comments included in the preamble. . .

For author-year references use:

```
\usepackage{natbib}
```

Specify your bibliography database in a different folder:

```
\bibliography{Biblio/articles.bib}
```

Note that you have to specify the path UNIX-style (using forward instead of backward slash as folder separator). Avoid spaces in filenames (some TEX accessories may not work properly with spaces in the filename specification).

Most TEX Systems allow you to place your bib files in a separate folder on your localtexmf tree. For details consult the documentation that comes with your TEX System. MiKTEX users can create a bibtex folder in their localtexmf tree, place their bib files there, and refresh the FNDB in MiKTEX's Options interface.

For fancy pdf files use:

```
\usepackage{hyperref}
\hypersetup{
   pdftitle={Shown in AR File Information},
   pdfstartview=FitH,   % Fit the page horizontally
   bookmarks=true,      % Open Bookmarks in AR
}
% more options can be found in
% TEXMF/doc/latex/hyperref/manual.pdf
```

To manually correct the hyphenation of a word that was not properly handled by TEX (eg. Weltauffassung) put the following in the preamble:

```
\hyphenation{Welt-auf-fas-sung}
```

To prevent long titles in your table of contents (generated by LATEX) use alternative short title:

```
\section[Short Title for TOC]{Long long long title}
```

You can find everything about TEX and LATEX on: TUG. . .

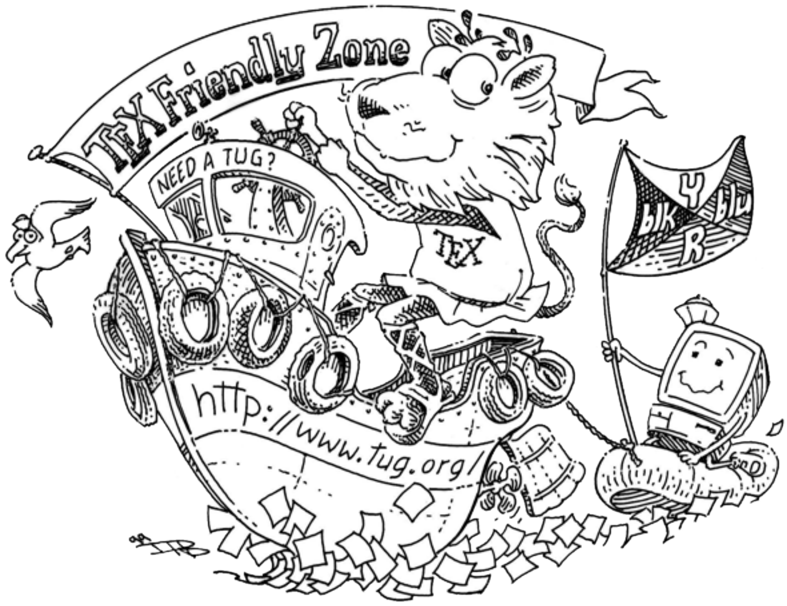


Figure 5.7: Happy TEX-ing from TUG and the WinEdt Team

5.10 In memoriam Echo (2005-2016)

Echo, our faithful companion and a long-time CSO (Chief Security Officer) at WinEdt HQ, passed away on January 21, 2016 (just over a month short of her eleventh birthday).

Figure 5.8: All Things Must Pass

She is being missed, no kidding...

Bibliography

TUG Online Book Catalogue. *Books about TeX and Friends*. TUG Web Site, 2017. URL https://www.tug.org/books.
Everybody loves books, and everybody loves TeX. On this url you'll find it all...

Donald E. Knuth. *The TeXbook*. Addison-Wesley, 1984. ISBN 0-201-13448-9.
The basis of everything, covering plain TeX.

Leslie Lamport. *LaTeX: A Document Preparation System*. Addison-Wesley, second edition, 1986. The basis of everything, covering LaTeX and LaTeX2e.

Helmut Kopka and Patrick Daly. *A Guide to LaTeX2e*. Addison-Wesley, fourth edition, 2003. ISBN 0-321-17385-6. Covers core LaTeX from the ground up, and discusses commonly used packages for graphics, web integration, and more.

Frank Mittelbach, Michel Goossens, Johannes Braams,and Chris Rowley. *The LaTeX Companion*. Addison-Wesley, second edition, 2004. ISBN 0-201-36299-6.
Covers core LaTeX, plus a vast array of additional packages.

David Griffiths and Desmond Higham. *Learning LaTeX*. SIAM, second edition, 1997. ISBN 978-0-898713-83-1. A short example-based book covering core LaTeX and a few packages.

Herbert Voß. *Typesetting Mathematics with LaTeX*. UIT Cambridge, 2010. ISBN 978-1-906860-17-2. A practical book on typesetting mathematics with LaTeX, covering many packages.

Herbert Voß. *Typesetting tables with LaTeX*. UIT Cambridge, 2011. ISBN 978-1-906860-25-7. A practical book on typesetting tables with LaTeX, covering many packages.

George Grätzer. *First Steps in LaTeX*. Birkhauser, 1999. ISBN 0-8176-4132-7.

George Grätzer. *Math into LaTeX*. Birkhauser Boston and Springer Verlag New York, 2000. ISBN 0-8176-4131-9 and 3-7648-4131-9.

George Grätzer. *More Math into LaTeX*. Birkhauser, 2007. ISBN 978-0-387-32289-6.