A short manual for TEXworks

Alain Delmotte, Stefan Löffler, and others

lowering the entry barrier to the TEX world

Contents

1
Introduction

Donald E. Knuth decided to create a new typesetting system, which would be called TeX, because there had been a change in the printing system used for the volumes of his book *The Art of Computer Programming* and Knuth found the result of the new system awful.

The goal of TeX was then to have a system which would always produce the same documents independently of the actual machine they were processed on. Knuth also designed the *Computer Modern* family of typefaces and the METAFONT language for font description.

The work initiated in 1977 was finished (the languages were "frozen") in 1989. TeX and METAFONT are not evolving any more except for minor bug fixes (TeX versions are numbered following the decimals of π—now 3.1415926—and METAFONT the decimals of the number "e"—now 2.718281).

TeX provides basic tools (commands/instructions/"primitives") to define typesetting; almost every detail has to be defined, but the language allows the creation of macros for repeatedly used constructs. So collections of macros are loaded through format files (i.e., pre-compiled large macro collections).

Knuth created an original default format (600 commands, more or less) which is called *Plain TeX*. This facilitates creating documents.

The most widely used format is LaTeX (Leslie Lamport, 1985), which provides more global commands and structures for documents (article, book,...) allowing easier and faster work, but sometimes with loss of flexibility due to the more or less rigid framework. But there are many other formats and TeX-variants in use as well, such as $\mathcal{AMS}$-TeX, $\mathcal{AMS}$-LaTeX, ConTeXt, or XeTeX, each having specific goals and advantages (and drawbacks).

To extend the format, one loads "packages" which are collections of macros specific to some aspect of typesetting.

From its specification in the late 1970s, the TeX family had to evolve until

now, last version March 2008, to take into account the developments in the typesetting world outside TeX.

Some of the problems to answer were/are:

- taking into account other languages with "alphabets" larger than the ASCII[1] one or with non-Latin characters altogether,

- having more fonts, there is not much variety in the fonts created with METAFONT (few font creators use it),

- creating documents in other formats than the normal DVI[2],

- using the rich possibilities of other typesetting systems and formats like PostScript and PDF,

- having more calculation and scripting facilities,...

To answer these questions and others, many "engines" and programmes have been created around TeX, including pdftex, pdflatex, dvips, ps2pdf, and METAPOST, which opens the TeX world to the possibilities of PostScript and PDF. X$_\exists$TeX and X$_\exists$LaTeXto be able to use the "normal" fonts found on the different machines and to be able to cope with writing systems different from the left to right systems which originated in Europe (Latin and Cyrillic letters and associates)—right to left, vertically, pictograms,...Or LuaTeX and LuaLaTeX to have a powerful scripting language.

To use TeX and the systems of its family, one has to create a "source" document as TeX is only a system to "transform" a source document into a (beautifully) typeset document. This source is a simple text with typesetting instructions and one needs a programme to create it: the editor.

There are many editors able to create a TeX source; some are general editors, others are specifically designed for TeX: here TeXworks comes in.

TeXworks is a project to create a text editor for use with the TeX family of tools; we will refer to these as (La)TeX. Instead of creating a new sophisticated program, equipped with multiple tool-bars to meet any need, TeXworks provides a simple editor, offering at first sight only a limited set of tools for text editing as well as a single button and a menu to typeset a (La)TeX text.

---

[1] "American Standard Code for Information Interchange": a character encoding scheme including only Latin characters found in English, some common punctuation characters, and a few other symbols such as % or $

[2] "Device Independent": format of files produced by TeX

The idea to create the editor came to *Jonathan Kew*, the initiator and leader of the project, after a long period of reflection on the reasons why potential users tend to keep away from (LA)TEX, as well as pondering the success of the **TEXShop** editor on the Mac.

Finally the goal was also to provide the same editor on many operating systems: TEXworks currently runs on Linux, Mac OS X and Windows. The interface is always the same and the program offers the same functionality on all three platforms.

After this introduction, the second section of this manual explains how to install the software. In the third section, we describe the interface and create a first document as well as show the basics of TEXworks. In the forth and fifth section, the advanced tools provided by TEXworks are presented; you should read these sections only after mastering the basic working of TEXworks. These advanced tools allow much more effective working practices. The sixth section gives a brief introduction to scripting. This section focuses on using ready-made scripts, not on writing your own scripts (which is beyond the scope of this manual and will be presented elsewhere). After that, the seventh section in which some pointers to further information about TEXworks and sources for help are compiled concludes the main part.

Finally, the appendices provide additional information how TEXworks can be customized, about the regular expression search/replace system, and how TEXworks can be compiled from source. A short bibliography and an index conclude this manual.

1.1   Icons and style

Because a picture is often worth a thousand words, icons and special styling is used throughout this manual to avoid cumbersome paraphrases or mark specialties. Keyboard keys are usually depicted as $\boxed{A}$, with the exception of a few special keys. These are: $\boxed{\text{Shift} \Uparrow}$, $\boxed{\text{Page} \uparrow}$, $\boxed{\text{Page} \downarrow}$, $\boxed{\leftarrow}$ (return), $\boxed{\uparrow}$, $\boxed{\downarrow}$, $\boxed{\leftarrow}$, $\boxed{\rightarrow}$, $\boxed{}$ (space), $\boxed{\longleftarrow}$ (backspace), and $\boxed{\leftrightharpoons}$ (tab).

In addition, mouse clicks are depicted as 🖱 (left click) and 🖱 (right click; on Mac OS X with a one-button mouse, this is usually available by holding down $\boxed{\text{Ctrl}}$ while clicking).

Apart from input instructions, several passages throughout this manual are marked by special styling.

Information that is only valid or relevant for a particular operating system is marked like this:

Win This only concerns you if you use Windows.
Of course you can also read it if you use another operating system.
It just will not be of much use to you.

Code examples are set in a fixed-space, typewriter font, with lines above and below to set it apart from the rest of the text:

```
Hello \TeX-World!
```

Closely related to this, chapter 3 contains several tutorials, which are typeset just like the code examples above, but with an additional notebook icon next to it.

2

Installation

TeXworks is only a text editor; to be able to create documents with (LA)TeX and to typeset them to PDF, we also need what is called a *TeX distribution*. This is a collection of programs and other files which will be automatically called by TeXworks during its work. Thus, you need to install a distribution: we will do that *before* starting TeXworks for the first time, as this way, TeXworks will automatically find what it needs.

TeX Live (`http://www.tug.org/texlive/`), a combination of teTeX, MacTeX and XEmTeX, is available for all three operating systems (Linux, Mac OS X, Windows). The current version is TeX Live 2010. Note that you need a reasonably recent version of TeX Live (2008 or later at the time of writing) to use all features of TeXworks.

*nix For Linux: most Linux distributions include a TeX distribution, but it may not be installed by default and you will have to use the Linux package management tools to do that. As an alternative to TeX Live, one can use **teTeX** (`http://www.tug.org/teTeX/`, a predecessor system to TeX Live.

Mac For the Mac: **MacTeX**, a new distribution based on gwTeX and XeTeX, is available; see `http://www.tug.org/mactex/`.

Win For Windows: a very popular distribution is **MiKTeX** (`http://www.miktex.org/`). MikTeX has an update programme, which has also been ported to Linux. You can also use the XEmTeX distribution (`http://www.xemtex.org/`).

2.1   Under Windows

Most of the larger TeX distributions already contain TeXworks as a package. Sometimes, these versions even have some distribution-specific enhancements.

So, the preferred way of installing TEXworks on Windows is to use the package manager of your distribution. In this case, you can skip the next few paragraphs. Be sure to read the end of this section, though, as it provides important information about customizing TEXworks to your needs.

If you want to obtain an "official" version, obtain TEXworks by downloading the setup from the TEXworks web site `htp://tug.org/texworks/` after the installation of the TEX distribution.

Simply install TEXworks by running the setup file. During the installation, you will be asked where to install the program, if you want to create shortcuts, and if you want to always open `.tex` files with TEXworks. There are reasonable default values that should work well for most users.

If you want full control over how and where TEXworks is put, you can also download the `.zip` archive from the website and unpack it wherever you like. Note that in this case, shortcuts and file associations must be created manually.

When you start TEXworks for the first time, it creates a folder named `TeXworks` in your home folder[1]. This folder will contain some sub-folders for auto-completion, configuration, dictionaries, templates, and interface translation/localisation files—we will see these in more detail later.[2]

NB. At the time of writing, if `<your name>` contains any non-ASCII characters (for example accented characters), some functions of TEXworks may not work correctly. For example, the spell-checker and forward/reverse synchronization between the source and `.pdf` will be impaired.

2.2   Under Linux

Several common Linux distributions already have packages for TEXworks. They are adequate for most users and facilitate installing TEXworks considerably.

If your Linux distribution does not provide recent, adequate packages, you need to build TEXworks from source yourself, which is fairly easy on Linux. After the installation of the TEX distribution, go to `http://code.google.com/p/texworks/wiki/Building` and follow the instructions suitable for your Linux distributions. Also see section C.

Once the program is installed, start TEXworks. The folders `.TeXworks` and `.config/TUG` will be created in your home directory.

---

[1]on Windows XP, your home folder is `C:\Documents and Settings\<your name>`. On Windows Vista/7, your home folder is `C:\Users\<your name>`.

[2]TEXworks will save its preferences in the registry: `\HKEY_USERS\S-...\Software\TUG\TeXworks`. If this is erased, it will be recreated with default values at the next use.

2.3   Under Mac OS X

MacTeX 2010 already contains TeXworks as a package. So, the preferred way of installing TeXworks on the Mac is to use the package manager of your distribution. In this case, you can skip the next few paragraphs. Be sure to read the end of this section, though, as it provides important information about customizing TeXworks to your needs.

If you want to obtain an "official" version, obtain TeXworks by downloading the archive from the TeXworks web site `htp://tug.org/texworks/` after the installation of the TeX distribution.

You need to get `TeXworks-Mac-0.3r567.zip`[3] which contains everything you need.

It is a standalone `.app` package that does not require any Qt files installed into `/Library/Frameworks`, or other libraries into `/usr/local/lib`. Just copy the `.app` anywhere you like and run it.

On Mac OS X, the TeXworks resource folder will be created in your `Library` folder (`~/Library/.TeXworks/`), inside your home directory. Preferences are stored in `~/Library/Preferences/org.tug.TeXworks.plist` which you can delete if you ever suspect it is causing problems.

2.4   Ready!

Finally, some files may need to be added to the "personal" files that TeXworks creates. As the exact location of these depends on your platform, this will be referred to as `<resources>` or the **TeXworks resource folder** throughout this manual. On Linux, this is `~/.TeXworks`, on Windows XP it is `C:\Documents and Settings\<your name>\TeXworks`, on Windows Vista/7 it is `C:\Users\<your name>\TeXworks`, and on Mac OS X it is `~/Library/.TeXworks/` by default. The easiest way to locate this folder in recent versions of TeXworks is to use the ***Help→Settings and Resources...*** menu item. It opens a dialog which shows you where TeXworks saves its settings and where it looks for resources.

After installation and first run, have a look in the sub-folders of the TeXworks resource folder and delete any `qt_temp.xxxx` files; they are temporary files left behind and could interfere with the normal ones, which are installed in the same folder, later on.

---

[3]Version at the time of writing this manual.

First steps

Let's now see how to create a first document: for this you'll need to type some text in the editor window of TEXworks. (LA)TEX is *not* WYSIWYG[1] software, so you'll have to type the text and the instructions for formatting it and you'll see the result only after "typesetting" the text. This looks a little bit dry, but one very quickly gets used to it and it is well worth the effort.

3.1   Interface summary

When one opens the editor, it shows a very sparse interface: a title bar, a menu bar, two small toolbars, a large typing zone (white) and, at the bottom, a status bar. We are in the *source/editor* window. If you have already typeset the document previously, the resulting `.pdf` will be shown on the right hand side in the *preview* window.
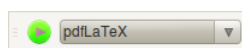
The first toolbar has a button to typeset and an drop-down menu to choose the format for typesetting (we'll choose `pdfLaTeX`). Knowing that the keyboard shortcut for typesetting is Ctrl T (Mac OS X: ⌘ T) and that we almost never change the format, we could even hide this toolbar. The selection of a format for compiling can also be changed through the **Typeset** menu.

The second toolbar provides the standard functions: New document, Open, Save | Undo, Redo | Cut, Copy, Paste | Search, Replace.

---

[1] *What You See Is What You Get.*
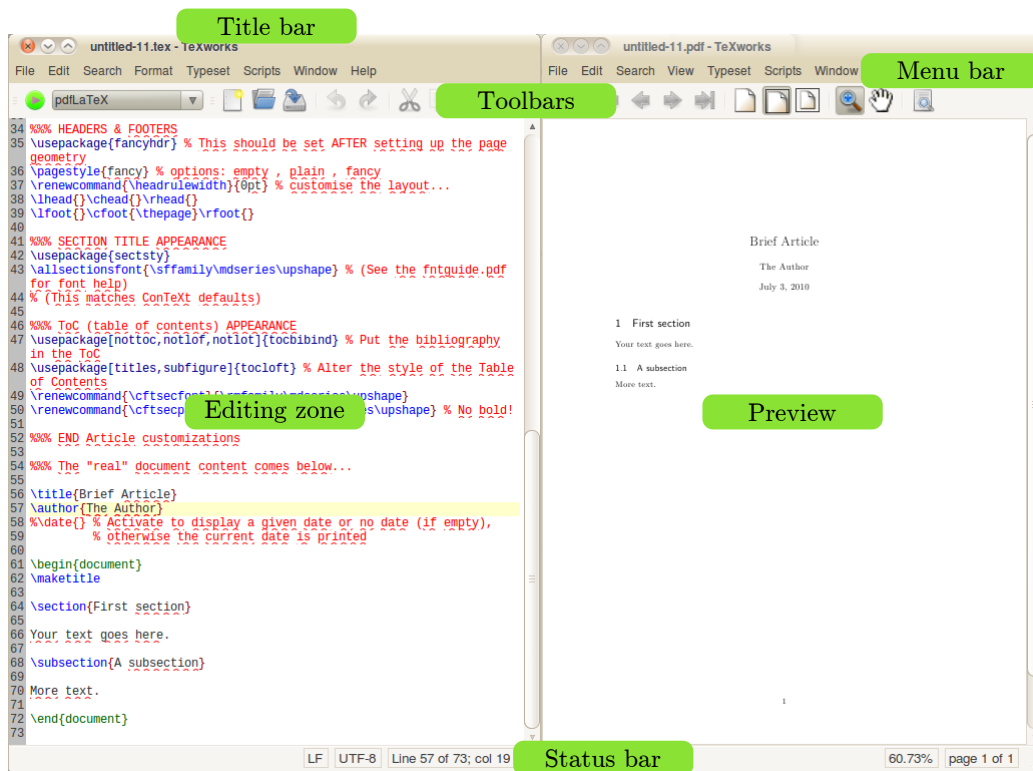
Even though they are not looking like real buttons, the widgets in the status bar can be clicked. The widgets showing the current position (line or page, respectively), for example, open a dialog to enter a line or page to jump to when clicked. The other widgets typically open contextual menus where some settings can be changed.

3.2 Creating a document

3.2.1 Writing the document

As an example of the use of TeXworks, we will work with LaTeX, but any other TeX system is possible. In particular, if you need to use some special fonts[2]—a mandatory font for an official template, non-Latin alphabets, etc.—the XeTeX system is very powerful.[3]

---

[2]You can only use fonts (LA)TeX knows, most of which are coming in packages included in your distribution. You cannot use your "normal" fonts, unfortunately. For more information, see for example `http://faq.tug.org/` and `http://www.tug.dk/FontCatalogue/`.

[3]See the bibliography for pointers to XeTeX and XeLaTeX.

Let's create our first document now. Enter the following text exactly as shown. To show some of the features of TeXworks/LaTeX, it is in French intentionally.

```
\documentclass{article}

\usepackage[utf8]{inputenc}
\usepackage[T1]{fontenc}
\usepackage{geometry}
\geometry{a4paper}

\usepackage[frenchb]{babel}

\title{Premier document}
\author{Un TeXnicien}
\date{}

\begin{document}
\maketitle

Voici un texte accentué en français!

\end{document}
```

Save the file in a folder for test documents (e.g., `<home>\TeXworks_tests`); call the file `first.tex`. Note that it should have a `.tex` extension.

3.2.2   Typesetting the document and viewing it

Next we start typesetting[4] by clicking the green button 🟢 or by ⌨Ctrl ⌨T (Mac OS X: ⌨⌘ ⌨T).

A new panel opens between the typing area and the status bar: the *output panel*, labeled *Log*; everything LaTeX is doing is displayed there[5]. When LaTeX finishes this panel disappears (if there no error occurred) and a new window will appear; in this new window, the *Preview window*, you can see a page with a title "Premier document" followed by the name of the author "Un TeXnicien", both centred, the text "Voici un texte accentué en français!", and a page number at the bottom centre.

---

[4]We also use the words "compilation" and "to compile" for the same action; indeed LaTeX works on the source file to produce a `.pdf` output, so there is a compilation.

[5]see page 12 for a picture of the panel

Notice that the mouse cursor is like a magnifier in the new window. If you press (and hold) the left button of the mouse you can see the text under the magnifier much bigger (it is a magnifier, isn't it!); you can move the magnifier and so inspect the text in detail.

To go back to the source, you can just click in its window or better use ⌨Ctrl ⌨ (Mac OS X ⌨⌘ ⌨). This shortcut toggles between the two windows. See also section 5.1 to automatically move to a specific location in the output from the source or vice versa.

3.2.3 The work of LaTeX

Let's shortly analyse the result to understand what LaTeX did and why, now. Introductions and full tutorials can be found on the internet: see for example *lshort*[6] which should be installed as part of your TeX distribution, and is also available from CTAN.[7]

First, we ask to create a document of the *article* class: this defines the global layout of the document.

Next, we say that the input document (the source) is saved with the Unicode encoding *utf-8* and that it may contain characters which are not present in the standard ASCII without accents. We also want to use an output encoding T1 (the modern TeX encoding); we also want an *A4* document and not the default *US letter* size. Finally, we make it clear that the typography should follow the French rules using the **babel** package.[8] Those general instructions for the work are done by packages called with options.

Lastly, we finish the declaration part of the document, the *preamble*, giving the title, the author, and the date of the document; here we specify no date.

Next comes the body of the document, which describes the actual content, between the lines `\begin{document}` and `\end{document}` (these are LaTeX commands).

Let's do some experiments to show the effect of these instructions. For this, we put a % in front of the instructions; the % and everything after it will be considered as comment, which will be ignored by LaTeX.[9]

Comment out the line `\usepackage[utf8]{inputenc}`, and typeset the file. You should see that the accented characters are now displayed incorrectly

---

[6] *The (Not So) Short Introduction to LaTeX 2ε*

[7] *Comprehensive TeX Archives Network*, a network of mirror servers of the central CTAN; there, one can find almost everything about TeX, LaTeX, and more: `http://www.ctan.org`

[8] This influences, e.g., automatic hyphenation of words or the way punctuation characters are typeset

[9] Notice that the comments are, by default (this can be changed), coloured red by TeXworks, so we see them well.

in the preview window. If, in addition, you also comment out the line
\usepackage[frenchb]{babel}, LaTeX will give an error. Just hit ⟨←⟩ to
continue the typesetting.

After these experiments, let's modify the text as follows:

```
\begin{document}
\maketitle
\tableofcontents

\section{Petite démonstration}

Voici un texte accentué en français!
Suite du texte entré après avoir fait un retour chariot. Dans
l'éditeur on peut demander un passage automatique à la ligne
du texte saisi; mais le numéro de ligne n'est incrémenté que par un
retour chariot.

Nouvelle ligne en passant une ligne dans la source: c'est la
manière d'indiquer un changement de paragraphe.

\end{document}
```

Redo the previous experiments and observe the changes which appear.

Note that entering only one carriage return doesn't create a new paragraph. In LaTeX, one has to have an empty line for that. In TeXworks, the line number of the source (on the right in the status bar) numbers the lines created with carriage return, not the wrapped lines.

3.3   And when errors occur?

When you create a document for typesetting with LaTeX, you cannot avoid making mistakes: forgetting a closing brace or an \end{} to close an environment, using mathematical commands without switching to mathematical mode, etc. When you compile and there is an error, LaTeX stops, giving you a chance to deal with the problem. This is shown by the stopping of the scrolling action in the output panel, and an error message being displayed, with LaTeX waiting for an instruction to know what it should do.
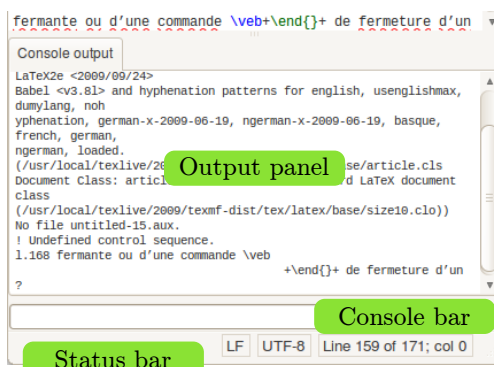
One sees the *typing cursor* in the line between the output panel and the status bar: the *console bar*.

The error message is on many lines, for example like this:

```
! Undefined control sequence.
l.168 ... fermante ou d'une commande \veb
                                          +\end{}+ de fermeture d'un...
?
```

LaTeX says that it doesn't recognize the command name, sometimes suggests to see the manual or to type h (plus ⏎) for help, points to the line number where it noticed the error[10] (here 168), and shows the place of the error at the cut of the line (here at \veb). Finally, it shows that it waits for an action from us by displaying a single question mark.

There are different possible actions:

- Type ⏎ and ask to continue as if nothing happened; sometimes this allows to finish compiling, but there will be an error in the result.

- Type h ⏎ to ask for help; this help is not always clearer than the error message, but often gives a clue.

- Type i ⏎ to tell LaTeX that we will propose a replacement text. Enter the text followed by ⏎ ; it will be used, beginning at the start of the error, but you should correct the source afterwards as LaTeX never changes that.

---

[10]Unfortunately, this does not always have to be the place where the actual mistake was made in the sources. This is discussed later.

- Type x ⟵ to stop compilation. This is the traditional (L#)T#X way to kill a typesetting process.

  We can also kill the typesetting by repeating the action used to start it: the green typesetting button will have changed to a red one with a white cross ❌. By clicking on that button or by hitting [Ctrl] [T] (Mac OS X: [⌘] [T]) again, the L#T#X process is terminated. The output panel is still visible and so one can still see the error message.

You should note that sometimes an error appears far from its actual position. For example, when opening an environment but not closing it, L#T#X doesn't see the error before it encounters another end of environment without closing of the first one. The error is often only picked up at the `\end{document}` command, which shows that another environment was not closed!

Sometimes, an error still occurs during subsequent runs of (L#)T#X even after it was corrected in the document. This can happen because (L#)T#X creates a number of intermediary files which can still contain the original, erroneous code. Therefore, it is advisable to remove those files after fixing an error. T#Xworks provides a command to facilitate this—see section 5.6.

After an error occurred, the output panel remains open—even after consecutive typesetting runs—to help you fix the problem (which can sometimes take several attempts and restarts of (L#)T#X). Once all problems are solved, you can close the panel by the **Windows→Hide Output Panel** menu item. As an alternative, you can also configure to output panel to hide automatically as soon as the typesetting process finishes successfully by setting **Edit→Preferences. . .→Typesetting→Hide output panel** to *On success*. Since one can easily overlook other problems in the document (e.g., undefined references) that do not cause (L#)T#X to fail, this option is only recommended to expert users.

To help you in finding and fixing error (at least if you are using L#T#X), T#Xworks comes with a small script that extracts all errors, warnings, and other noteworthy messages from the console output and presents them to you in a simple, tabular form.
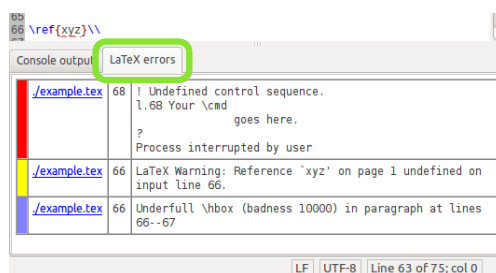
Here, you see a list of LaTeX messages, color-coded and sorted by severity. Red represents errors, yellow indicates warnings, and blue stands for over- and underfull box warnings. Next to the colour bar, you see the name of the file in which the error was detected. Next to that, you see the line number (if the script was able to determine that), as well as an excerpt from the console output telling you what the error was. Moreover, the filename is a link which will take you to the file (and, if a line number could be determined, also the line) where LaTeX reported the error. Hopefully, you can quickly fix any errors that may occur this way.

In case you are not using LaTeX, this script may be of little use, particularly if the console output is formatted differently. In this case, you can simply disable the error parsing hook script (see section 6.2).

3.4   Changing TeXworks parameters for convenience

If the default font of the editor doesn't suit you, it is possible to change it from **Format→Font...** by selecting a new one in the dialogue box which appears. This change will apply only to the current window, and until TeXworks is restarted.

From the **Typeset** menu or from the drop-down on the **Typesetting tool bar**, you can change the compilation format. Again this change will only be temporary and for the current document.

To have permanent changes, you need to change the *preferences* through the **Edit→Preferences...** menu item, using the **Editor** tab for the font and the **Typesetting** tab for the compilation format: the default format is at the bottom of the tab.

4

Going further: Editing tools

When you have had some practise with TeXworks, you'll find the need for more effective tools. Many of them are bundled with TeXworks. We are going to see some of them now.

4.1 Creating a document from a template

Most documents you will create will use the same instructions in the preamble, the same layout settings, similar heading and so on. You can use predefined templates to get started quickly or create your own with all of these settings already in place.

Use ***File→New from template...*** or Ctrl Shift ⇧ N (Mac OS X: ⌘ Shift ⇧ N). A dialogue box opens to allow you to select one of the templates. After selecting one and pressing OK, a document is created and you can start to work.

If you want to create a personal template, you just have to create a suitable document with everything you always want to do (and perhaps marking places to fill in the rest) and save it as a `.tex` file in the `<resources>\templates` folder, or a sub-folder of it, if you wish.

4.2 Creating a project using several source files

When the source becomes long, it is sometimes difficult to navigate and maintain it. In that case, it is useful to split the source into different smaller files: one file will be the main document, with the preamble and the `document`

environment, as well as calls to the "sub-documents"[1], which could in turn contain separate chapters, for example.

But there might be a problem if you want to start typesetting/compilation in a sub-document: as there is neither a preamble nor a `document` environment there, LATEX will stop immediately with an error.

To tell TEXworks that it should typeset the main document, one adds at the very beginning of the sub-document the instruction:

```
% !TeX root = path/main_file.tex
```

for example:

```
% !TeX root = manual.tex
```

If the main file is in the same folder, its name is enough, as in the above example. Otherwise, you must also give the path to the main document (preferably relative to the sub-document in question, e.g., `../manual.tex`). Notice that the slash `/` and not the backslash `\` should be used as directory separator even on Windows.

Further, with MiKTeX, the call to a sub-document `\input{name.tex}` should include the extension `.tex` to ensure proper SyncTeX functionality (see section 5.1).

4.3 Spell-checking

You can turn on automatic spell-checking of your source document from ***Edit →Spelling→<language>***.

During typing, every word the spell-checker considers wrong is underlined by a red wavy line. A right-click on the word opens a contextual menu in which there are some replacement suggestions. Click on the desired word to make the replacement.

Before using the spell-checker, you need to install dictionaries in the right folder of TEXworks: `<resources>\dictionaries`.

On Linux, the dictionaries are usually taken from the folder `/usr/share/myspell/dicts`—the default path for myspell dictionaries. Note, though, that the maintainer of your TEXworks package may have changed this to

---

[1]Called by the commands `\input{}` or `\include{}`, see LATEX manuals for more information.

reflect the file system layout of your Linux distribution. You can override this default by setting the `TW_DICPATH` environment variable before running TEXworks.

One can use the available dictionaries for OpenOffice.org and other free software;[2] if you have Mozilla Thunderbird with spell-checking, you can copy its `.aff` and `.dic` files as well, for example. It is possible to ask TEXworks to enable spell-checking by default by setting a dictionary in ***Edit →Preferences. . .→Editor→Spell-check language***.

4.4   Search and replace

4.4.1   Standard functions

The options of the menu ***Search—Find. . .***, ***Find again***, ***Replace. . .***, ***Replace again***, and ***Go to Line. . .*** (Ctrl F , Ctrl G , Ctrl R , Ctrl Shift ⇑ R , and Ctrl L , respectively)—are standard actions (Mac OS X: ⌘ F , ⌘ G , ⌘ R , ⌘ Shift ⇑ R , and ⌘ L); the first and the third open a dialogue box:
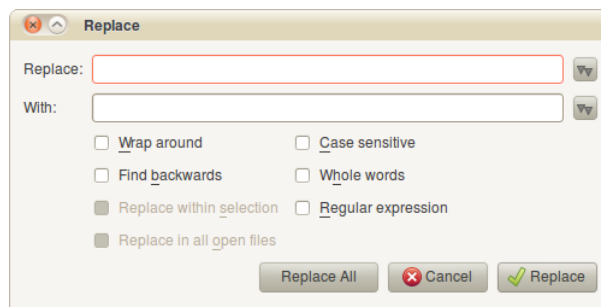
Here, the usual options are available: *Wrap around*, *Find backwards*, *Search/Replace within selection*, or *Find all occurrences*. The following options are also usual: *Case sensitive* and *Whole words*. By default, the search is forward, towards the end of the document.

The option *Search/Replace in all open files* is also a frequent choice, but not as much as the others; this allows, for example, replacement in all the files of a project—pay attention, though, as this is very powerful.

The last option, *Regular expression*, is detailed in the next sub-section.

In the ***Search*** menu there are other options:

---

[2]See, for example, `http://lingucomponent.openoffice.org/download_dictionary.html`.

Copy to Find copies the currently selected text into the ***Find*** field of the Find dialogue or the ***Replace*** field of the Replace dialogue; you still need to open the dialogues separately

Copy to Replace copies the currently selected text into the ***With*** field of the ***Replace*** dialogue

Find Selection uses the current selection for a search without opening the ***Find*** dialogue—very fast

Show selection scrolls the view to the currently selected text—useful if word wrapping is turned off and you moved in the document using the vertical scroll bar on the right

4.4.2   Advanced search and replace (regular expressions)

The regular expressions provide a very powerful tool, but they require some effort to be well understood. To understand them fully would require a manual of its own[3], so we'll only give some simple ideas of use. For more advanced uses, as well as lists of the most used codes, see section B.

Suppose we have the following text:

```
Voici du texte pour tester les expressions régulières dans du texte
accentué.
Voici du texte pour tester les expressions régulières dans du texte
accentué.
Voici du texte pour tester les expressions régulières. Voici du
texte pour tester les expressions régulières.
truc            truc
tél.: 010-99-99-99
tél.: 00.32.10.99.99.99
tél.: 00/32-10/99.99.99
```

We want to

1. insert an empty line after each "accentué" (to create paragraphs in LaTeX), but not after the three telephone numbers;

2. replace each *tab* character between the two words "truc" of the fourth paragraph by three spaces; and finally

---

[3]Such manuals exist on the internet.

3. make the telephone numbers consistent by replacing the various punctuation characters by spaces.

For 1., in the dialogue box **Replace** (⌜Ctrl⌟ ⌜R⌟) for *Replace:* we put ›\n‹[4] and in *With:* ›\n\n‹. ›\n‹ is the code to match or insert a line feed. You will need to select the first four paragraphs and the beginning of the fifth (the first telephone number) and to tick the *Replace within selection* and *Regular expression* options; if this was not done and an empty line has been inserted after each line, select the telephone lines and do the reverse action: replace ›\n\n‹ by ›\n‹. So we replaced one line feed by two, creating an empty line.

For 2., use ›\t‹ and ›␣␣␣‹[5]. ›\t‹ is the code which represents a tab, while a space is typed in literally (here represented as ␣).

For 3., find ›-|\.|/‹ and replace with ›␣‹. Here, ›|‹ provides alternatives (-, ., or /); for the dot we have used ›\.‹ because the dot alone is a regular expression code which represents any character and we would have replaced all the characters by spaces! We therefore have to use a code—prefixing the dot with a backslash tells specifies that the normal meaning of the dot should be used instead of the special meaning it usually has in regular expressions.

If one has strings of the same character but of different lengths (for example 3, 4, or 5 times the sames character e) and one wants to truncate all these strings to a string with less characters (for example 2), one can ask to replace the string ›e{3,5}‹ by ›ee‹.

If one wants to insert the same string at the beginning of some paragraphs separated or not by an empty line, for example ›\noindent␣‹ or ›\item␣‹, one can replace ›\n\n‹ or ›\n‹ by ›\n\n\\noindent␣‹ or ›\n\\noindent␣‹. Pay attention, we have a double \ in front of noindent to get one (\noindent) because \ is an escape character in regular expressions (we've met it before in the expression \.)!

If it were making sense, we could replace all the letters between "a" and "m" by "$" using ›[a-m]‹ and ›$‹.

---

[4]the ›‹ are used here only to show the limits of the entered text and they should not actually be entered.

[5]These are three space characters.

4.5   Other tools for editing and error tracking

4.5.1   Standard tools

It is always possible to undo an action using ***Edit→Undo*** or `Ctrl` `Z` (Mac OS X: `⌘` `Z`): this way you can undo stepwise! The inverse action, redo, is available as ***Edit→Redo*** or `Ctrl` `Shift ⇑` `Z` (Mac OS X: `⌘` `Shift ⇑` `Z`).[6]

TEXworks also provides the standard editing tools such as the clipboard; therefore one can select, cut/copy and paste a piece of text normally.

You can select with the mouse by dragging over the desired text, or by double-clicking to select a word. Using the keyboard, holding down `Shift ⇑` while moving using the arrow keys will select text. You can also move and select word by word moving left or right holding `Ctrl` `Shift ⇑` down (`⌘` `Shift ⇑` on Mac OS X). The clipboard shortcuts are the ones you'll find in almost every program: `Ctrl` `X` to cut, `Ctrl` `C` to copy, and `Ctrl` `V` to paste (`⌘` `X` , `⌘` `C` and `⌘` `V` , respectively, on Mac OS X).

You can easily change the case of a selection—put everything upper case or lower case—using ***Edit→Change case*** and next, depending on the desired effect, ***ALL UPPERCASE***, ***all lowercase***, or ***Toggle Case*** (which toggles the case of each letter individually).

It is also convenient to show the line numbers, as all error messages refer to these numbers; you can toggle the line numbers, on the left of the editing panel, from ***Format→Line Numbers***.

4.5.2   Commenting

When preparing a document with (LA)TEX, it is often useful to prevent compilation of a portion of text to be able to locate an error; you can do this piece by piece until you find the part which causes the error. For that, commenting the source block by block is needed.

We have seen that the symbol % marks the beginning of a comment. To comment a big piece of text, it is sufficient to select it and ask to mark it as comment ***Format→Comment*** or `Ctrl` `Shift ⇑` `[` (Mac OS X: `⌘` `Shift ⇑` `[`). To remove the comment, select the lines and choose ***Format→Uncomment*** or `Ctrl` `Shift ⇑` `[` (Mac OS X: `⌘` `Shift ⇑` `[`).[7]

---

[6] `Ctrl` `Y` and `Alt` `Shift ⇑` `Backspace` work as well on Windows. `⌘` `Y` work as well on Mac OS X.

[7] On some keyboards, like the French one, it is impossible to use `Ctrl` `Shift ⇑` `[` or

4.5.3   Matching delimiters

A frequent error is to forget a closing symbol: parenthesis, bracket, square bracket, *etc.* TEXworks helps with a tool to show the pairs of symbols: when the cursor moves over one of these symbols, its partner is briefly highlighted. You can also select an entire block using ***Edit→Balance Delimiters*** or by the shortcut ⎡Ctrl⎤ ⎡B⎤ (Mac OS X: ⎡⌘⎤ ⎡B⎤). Thus, you will immediately see the scope of the block.

4.5.4   Smart quotes

Another similar error, but this time semantic and not hindering typesetting, is in the use of quotes when one wants to give focus to some text.

There are two types of quotation marks in English: the 'single' quotes and the "double" quotes. They are formed by ' and '; these are not the quotation marks used in programming and found on the keyboard: " and '. Using the TEXworks smart quotes system, one can use the latter as normal to automatically produce the typographically correct single/double opening and close quotes.

In a `.tex` document, select one of the smart quotes system: ***Format →Smart Quotes→TeX Ligatures***, ***→TeX Commands***, ***→Unicode Characters***. Then, when you want to start a quoted section in your text, let's say enclosed in double quotes, type ", then the text to be quoted, and finish again by "; TEXworks will automatically insert the correct opening quotes `` and later the correct closing ones ''. The three options give the same result in the typeset document, but ***TeX Ligatures*** should work best in most cases.

Finally, it is possible to define personal quotation marks systems (in the file `smart-quotes-modes.txt` in the `configuration` folder of the resource folder).

4.6   Auto-completion

Another tool which rapidly becomes indispensable is auto-completion. Indeed, when you use (LA)TEX, you have to continuously enter codes to, for example, create environments; you also have to remember to close every group you open.

Auto-completion allows you to type a keyword, hit the ⎡⇥⎤ key, and have TEXworks insert the (LA)TEX command or environment code automatically.

⎡Ctrl⎤ ⎡Shift ⇑⎤ ⎡⎤ ; these shortcuts can be changed, however—see section A.2

As an example to insert "LaTeX", we have to type `\LaTeX`. This is not difficult, but entering "`\`" followed by the word "LaTeX" with alternating capitals and lower case letters could become annoying after a while.[8] With auto-completion, you just enter `latex` and hit ⌨ to get `\LaTeX`. You just have to take care that there is no *letter* directly preceding or succeeding `latex`—e.g., `alatex`—, or else the mechanism might not pick up the correct keyword.

Another example is `bmin`, which gives

```
\begin{minipage}{}
•
\end{minipage}•
```

---

with the cursor between the empty pair of curly brackets where you need to enter the size of the minipage. See the section A.3 for a list of the keywords for auto-completion. Notice the "•" in the minipage environment. They are placeholders which can be reached by ⌈Ctrl⌋ ⌨ (⌥ ⌨ on the Mac), repeating this shortcut cycles forward through the placeholders; by ⌈Ctrl⌋ ⌈Shift ⇑⌋ ⌨ (⌥ ⌈Shift ⇑⌋ ⌨), you can also cycle backwards.

If a partial keyword is given, repeatedly hitting ⌨ will cycle through possible completions. For example, `bali` (the `b` commonly indicates the beginning of an environment, `\begin{}`) creates the `align` environment after one ⌨ , next `align*`, and after that, in succession, `alignat`, `alignat*`, `aligned`, `alignedat`, and `alignedat` with options; to access the last environments directly, they have their own codes which start by `bali` (`balis`, `baliat`, `baliats`, `balied`, `baliedat` and `baliedato`).

If you want to create your own keywords, you can add a `.txt` file in the `completion` folder inside the resources folder. The entries in the file should have the following format:

```
bfigo:=\begin{figure}[#INS#]#RET##RET#\end{figure}•
\bibliography{#INS#}•
```

In the first case, `bfigo` is the assigned keyword (with `:=`) to be converted into a `figure` environment with an optional argument; there are two carriage returns (`#RET#`) after the `begin`, i.e., an empty line, and the cursor is placed between the square brackets (at the position of `#INS#`). "•" is a place holder as introduced before.

---

[8]In particular with keyboard layouts where `\` is not directly accessible.

In the second case, we give ourselves a shortcut, which will let us type the first part of `\bibliography{}` and have TEXworks convert it to the full name plus braces (with the cursor between them). In this case, the keyword is the instruction itself.

Note that the `.txt` file containing the auto-completion information needs to be UTF-8 encoded—this is the default encoding for all files created with TEXworks.

Going further: Other tools

5.1 SyncTeX'ing between source and preview

When you are reading a document in the preview window and see something to change, it is convenient to go immediately to the corresponding place in the source. To do so, hold down ⟦Ctrl⟧ (Mac OS X: ⌘) and click at the appropriate place in the preview; the cursor will move and highlight the corresponding location in the source window. The same is true in the other direction: ⟦Ctrl⟧ 🖱 in the source will highlight the same line in the preview window.[1]

Here a remark for users under Windows: this only works if **all** the names for folders/files/. . . do **not** have accented characters. If, for example, your document is in `C:\Documents and Settings\Propriétaire\My Documents\thesis` it will not work because of the é in `Propriétaire`!

5.2 Special comment strings

Special comments, at the very beginning of the files, can be used to manage two other aspects of the compilation.

By default, TEXworks uses the "utf-8" encoding for loading and saving files, but some files could be saved in another encoding. Common ones are "latin1", which is the dominant encoding on Windows when using western languages, and "Apple Roman" which is dominant on Mac OS.

---

[1]It is also possible to use a right-click (🖱) to open a context window and select "jump to PDF" or "jump to source".

To set a different encoding for a specific file one can put the following at the beginning of that file:

```
% !TeX encoding = latin1
```

Note that without this line, you must tell TEXworks the correct encoding manually. Otherwise, your data could be corrupted! To override TEXworks's default choice of encoding, use the menu that appears when you click on the editor window's middle status bar widget.

If you opened a file in TEXworks that was not saved as utf-8 but is lacking the `% !TeX encoding` line, it might be displayed with (some) weird characters. In that case, you can specify the correct encoding via the status bar widget menu in the same way, but it is *imperative* that you then use ***Reload using selected encoding*** from the same menu! This forces TEXworks to open the document again with the encoding you selected, the weird characters should be replaced by normal ones, and only then it is safe to continue to work normally. To avoid having to repeat this procedure each time you open this file, you should either switch to utf-8 for saving it in the future or add a proper `% !TeX encoding` line.

If we want to compile a file with another programme than the default TEX or LATEX, we put at the beginning of the file:

```
% !TeX program = <the_programme>
```

for example:

```
% !TeX program = xelatex
```

Pay attention to this last instruction. You have to use the name of the programme here which should be used for the whole project, as the first encountered programme when starting typesetting is used (which is the one from the sub-document you are in). TEXworks will use that programme, even if another name appears in the main document!

When opening a document which contains a `% !TeX program` line, the specified programme will become the one to use and its name will appear in the drop down menu in the toolbar; you can, however, override this by selecting a different one from the drop down list, if you want.

In addition, you can set the spell checking language by a similar comment line:

```
% !TeX spellcheck = <language_code>
```

The language codes available on your system are listed in parentheses in **Edit →Spelling** next to the human-readable name of the language.

5.3   Formatting the source for legibility

To facilitate legibility of the source, one can use indentation as programmers do:

```
\begin{itemize}
    \item First element of the list;
    \item second element;
    \item last element:
    \begin{itemize}      % beginning of a sub-list
        \item first sub-element;
        \item second sub-element.
    \end{itemize}
\end{itemize}
```

This increases legibility, but works well only on short lines, without text wrapping; or if one chooses not to use text wrapping by unchecking **Format →Wrap lines**.

The command **Format→Indent** or the shortcut [Ctrl] [↹] (Mac OS X: [⌘] [↹]) will indent the line, or the selected lines, by inserting a tab character. You can repeat the process to increase the indent.

To remove one level of indentation, use **Format→Unindent** or the shortcut [Ctrl] [↹] ([⌘] [↹] on Mac OS X).[2]

As *indent* only indents the first part of very long (wrapped) lines, this is not very satisfactory in some cases. But one can ask TEXworks to split a long line (longer than the width of the editing window) into short ones adding a hard coded line feed. **Format→Hard Wrap...** opens a dialog box in which you can specify the width of the lines; you can also re-format lines which have already been split.

---

[2]See the modified shortcuts if your keyboard layout does not allow these actions.

5.4   Showing the tags

When a document is becoming long and you want to move to a specific place
(a chapter, a section, a subsection, ...) you normally need to scroll the editing
window to find the desired location, or use the Find dialog if you remember a
keyword in the chapter's title.

To the same end, though a lot more comfortable, you can also use the
structural information in the document to navigate the source: the menu item
Window→Show→Tags opens a panel showing the information detected
by TEXworks. Clicking on an item in the panel moves the cursor to the
corresponding part in the source. That panel, like any other, can be resized
by dragging its border.

The same action is possible in the PDF window from **Window→Show
→Table of contents**, but this only works if one has created structure tags
in the PDF file using the package `hyperref`.

5.5   Organising the windows

By default, the editor/source window opens on the left and the preview one
on the right (when the corresponding PDF file exists), thus splitting the
screen in two.

You can change the position of the windows in the **Window** menu.
→Stack and **→Side by side** give the default effect if there is only one
document open. If not, **→Stack** creates a mosaic with all the windows. The
other options allow to place the windows for your convenience. It is also
always possible to resize and move the windows manually, of course.

For the preview you can change the way it is presented and of course the
zoom by **View→Actual size**, **→Fit to width** and **→Fit to window**; you
can also zoom in and out. Shortcuts exist for all these actions and are shown
next to the menu items.

5.6   Cleaning the working folder

Very soon when one uses (LA)TEX, one discovers that the working folder is
cluttered by many files which have the name of the source file but different
extensions: `.aux`, `.log`, `.toc`, `.lof`, `.lot`, `.bbl`, ...

All these are files needed by (LA)TEX to be able to create the table of
contents, lists of figures/tables, the bibliography, the cross references and,
also very importantly, to keep track of what it did (in the `.log` file).

Apart from the external files, images, pictures, . . . , the only files required are the `.tex` files, the sources of the document. One can erase all the others. Sometimes, this is even necessary when (LA)TEX gets stuck after an error.

This can be done using a TEXworks command from the ***File*** menu with the →***Remove Aux files. . .*** item.

When you use this command, a dialog box opens in which you can check/uncheck the files you want to remove.[3] The dialog box will only list files that actually exist in the folder; if you removed all these auxiliary files before, you get a message box saying that there is no file to remove at the moment.

The list of auxiliary files which are taken into account is defined in the file `texworks-config.txt` in the `configuration` folder of the TEXworks resources folder. You could add some if required.

5.7   Changing the configuration

We have seen in section 2 (on page 5) that the first time you use TEXworks, it creates a resource folder and also that it saves default preferences.

It is possible to define a personal place where one wants the resource folder and the preferences. This can be handy when one wants a portable system (e.g., on an USB stick) or when one wants to easily access the templates or completion folders for modifications.

For this, create a file `texworks-setup.ini` in the programme folder in which you specify the path to the folder containing the completion, configuration, dictionaries,. . . folders and the configuration file (`texworks.ini`); there will be two lines:

```
inipath=C:/myfolder/TW_conf/
libpath=C:/myfolder/TW_conf/
```

`inipath` for the configuration file and `libpath` for the necessary folders. Here, `TW_conf` would replace the resource folder `TeXworks`. Note that the referenced folder (here `TW_conf`) should exist—it will not be created—, and that the `/` is used even on Windows (instead of the common \).

If one wants to put the resource folder in the programme folder as a subfolder, one can use an instruction like `inipath=./TW_conf/`; all relative paths are taken to be relative to the TEXworks programme folder (on Mac OS X, the folder containing the app package is used).

---

[3]The name of the main file is used to list the possible candidates for deletion.

One can also add a line like

```
defaultbinpaths=C:/Program Files/MiKTeX 2.7/miktex/bin
```

---

to specify where the programmes of the TeX distribution are located; but this instruction is not yet completely operational, especially under Windows.

6

Advanced use: Scripting

6.1 Introduction to Scripting

All the functions and utilities described so far were built into TEXworks
by default. While some of them could be configured or customized to a
certain extent, they are intended to suit the most common needs of a general
audience. However, the TEX world is very large and diverse. In order to
enable users to address their special needs—from simply making some text
bold to fulfilling special requirements for the next book or scientific paper
you want to publish—, the core functionality of TEXworks can be extended
or modified by the use of scripts.

Scripts are simple text files that you can open, read, or modify in any
text editor (including TEXworks, of course). They are written in a specific
scripting language that is essentially a programming language. At the time
of writing, TEXworks supports QtScript[1] (built-in), Lua (with a plugin), and
Python (with a plugin). To see which scripting languages are available on
your system, use the ***Scripts→Scripting TEXworks→About Scripts...***
menu item.

Writing scripts is beyond the scope of this manual, but is documented
elsewhere[2]. Here, only the installation and usage of scripts will be discussed.

TEXworks distinguishes between two types of scripts: standalone scripts
and hook scripts. The primary purpose of standalone scripts is to add new
functionality to the program. If you need a new function, such as a command
to make the selected text bold, a standalone script is the one to choose. These
scripts get an item in the ***Scripts*** menu, and you can run them simply by

---

[1]A scripting language similar to JavaScript provided by Qt.

[2]See, for example, Paul Norman's page `http://twscript.paulanorman.com/docs/index.html`.

clicking on that menu item (or by using a keyboard shortcut, if the script provides one).

Hook scripts, on the other hand, are meant to extend existing TeXworks functions. They are hooked into the code at specific places, e.g., after the typeset process has finished or after a file was loaded, and can add or modify whatever TeXworks is doing. One example for this would be a script that analyses a newly loaded file and sets the spell-checking language based on `babel` commands found in the document. Thus, hook scripts do not show up in the *Scripts* menu but are instead run automatically when the TeXworks function they modify is used.

You can easily determine which type of script you have by opening the script file. Near the top of the file, you should find a line similar to

---

```
// Type: standalone
```

Alternatively—once the script is installed—, you can use the dialogue available from *Scripts*→*Scripting TeXworks*→*Manage Scripts* to display this information.

6.2   Installing Scripts

A word of caution first: do not install scripts from a source you do not trust! Before installing scripts, you should make sure that the file you are about to install indeed does what you expect. Scripts are very powerful—they can do almost everything a normal program can do. So while there are some security precautions built into TeXworks, you should still be aware that scripts could potentially harm your computer and cause (among other things) crashes and data loss. In particular, scripts can read, create, and modify arbitrary files on your hard drive.

That said, installing scripts is very simple. Script files are generally installed in `<resources>/scripts` or subdirectories of it. These subdirectories are shown as submenus of the *Scripts* menu, so they can be used to group and categorize scripts. This is especially useful if you use many different scripts that would otherwise make the *Scripts* menu very confusing. One easy way to open the `scripts` folder is the *Scripts*→*Scripting TeXworks* →*Show Scripts Folder* menu item.

Since scripts are usually simple plain-text files, they do not come with fancy installers. To install them, simply copy or decompress (if archived, e.g., in a .zip file) the script file—and any other required files that you may have received—into `<resources>/scripts` or a subdirectory of it.

After having installed a new script file, TEXworks needs to become aware of it. It automatically scans for all scripts during start-up, so you could close all TEXworks windows and restart the application. An alternative is provided by the ***Scripts→Scripting TEXworks→Reload Script List*** menu item which rescans all scripts without otherwise interfering with the program.

You can also disable scripts (or whole directories of scripts) if you want to. This can be useful if you do not need some scripts for some time and do not want them to clutter the ***Scripts*** menu, but do not want to uninstall them entirely. Or if you want to prevent hook scripts from being run automatically. To do this, open the "Manage Scripts" dialogue with the ***Scripts→Scripting TEXworks→Manage Scripts*** menu item. Simply uncheck the script you want to disable and it won't bother you again.

6.3   Using Scripts

Using scripts is simple. Hook scripts are used automatically—you don't need to do anything. Standalone scripts show up in the ***Scripts*** menu or one of its submenus. If you cannot find a script you are looking for, or if you find a script you do not know the purpose of, you can use the "Manage Scripts" dialogue to get additional information (like the author, a brief description, etc.) about it.

Some scripts need to run other programs on your system. One example would be a script that opens the pdf in the system's default previewer, e.g., for printing. Since running arbitrary commands can in some situations be particularly dangerous, this functionality is disabled by default. You will notice this when a dialogue pops up informing you of an error in the script, or a similar message is displayed in the status bar. To enable scripts to execute system commands, open the preferences dialogue via the ***Edit →Preferences. . .*** menu item. There, go to the "Scripts" tab and check the "Allow scripts to run system commands" option. If you want to disable this function again later just uncheck the option. Note that this option applies equally to all scripts—there is currently no way to allow command execution only for some scripts.

7

Beyond this manual

In this manual, the authors tried to give an overview over TeXworks and a concise introduction to get you started. TeXworks is constantly evolving and improving, however, so the information presented here will never be complete.

Additional, frequently updated documents are posted in the wiki hosted by Google Code at `http://code.google.com/p/texworks/w/list`. Particularly noteworthy are the following pages:

SpellingDictionaries describes how to obtain and install dictionaries for the spell-checker on various systems. `http://code.google.com/p/texworks/wiki/SpellingDictionaries`

TipsAndTricks provides a compilation of useful things to know at a glance, such as the `% !TEX root` construct. `http://code.google.com/p/texworks/wiki/TipsAndTricks`

AdvancedTypesettingTools lists the configurations for several typesetting tools that are not included in TeXworks by default, such as latexmk or the dvips workflows. `http://code.google.com/p/texworks/wiki/AdvancedTypesettingTools`

If you run into problems with TeXworks, it is advisable to browse the mailing list archives accessible via `http://tug.org/pipermail/texworks/`. If you use TeXworks regularly or are interested in learning about problems and solutions when using it for some other reason, you can also consider subscribing to the list at `http://tug.org/mailman/listinfo/texworks` to stay up-to-date. For the occasional post to the mailing list, you can also use the ***Help→Email to mailing list*** menu item. Please make sure you replace the default subject by something describing your issue and to include

34

all information that might help resolving it. That way, you are much more likely to get many helpful replies.

If you find a bug in TEXworks or want to suggest a new feature you would like to see in a future version, you should have a look at the issue list at Google Code (`http://code.google.com/p/texworks/issues/list`). Before posting a new item, please make sure that a similar report or request is not already on the list and that the issue list is indeed the right place, though. If in doubt, please ask on the mailing list first.

Happy TEXing!

A

Customizing TEXworks

A.1   Syntax highlighting

Among its many other features, TEXworks also include syntax highlighting. This means that certain things like LATEX commands, environments, or comments are coloured, underlined, or highlighted in some other way. TEXworks also provides the ability to switch between different highlighting schemes[1], and to define your own ones. This is useful if you often work with types of files for which no highlighting scheme is provided by default, or if you want to adjust the highlighting schemes to better match your system's colour scheme.

To modify the highlighting schemes, you have to edit the plain-text file `<resources>/configuration/syntax-patterns.txt`. This file can contain any number of individual sections, each defining a single highlighting scheme to be displayed in the menu structure of TEXworks. To define a section, just write the name enclosed in square brackets on a line of its own. Naturally, these names should not include the `]` character. By default, the following two sections are defined:

```
[LaTeX]
[ConTeXt]
```

In addition, you can add comments to the file by starting a line with `#`. Empty lines are ignored.

Each section consists of an arbitrary number of styling rules. Each such instruction consists of three parts: a formatting instruction, a spell-check

---

[1]Use ***Format***→***Syntax Coloring*** to change the highlighting scheme for the current document, and ***Edit***→Preferences...→***Editor***→***Syntax Coloring*** to set the default one.

flag, and a regular expression[2] defining what part of a text to match. These parts must all be on the same line, and separated by whitespaces (e.g., spaces or tabstop characters). Take for example the following line from the default `LaTeX` section:

```
red     Y     %.*
```

The first part, `red`, defines the format (in this case, a red foreground colour is specified). The second part, `Y`, defines that spellchecking should be enabled for text that matches this particular rule. Sometimes, it is useful to put `N` here to disable spellchecking. For example, if spellchecking would be enabled for LaTeX commands, most documents would be flooded with red underlines indicating misspelled words when in fact they are only special commands. Finally, the third part specifies that this rule should be applied to all text preceded by `%`.

Let us take a closer look at the three parts of each rule. In its most general form, the first part—the format instruction—looks like

```
<foreground_colour>/<background_colour>;<fontflags>
```

The `<fontflags>` can be specified independent of the colours (note, though, that it must always be preceded by a `;`). The background colour (together with the `/`) can be omitted, but if you specify it, you also have to specify the foreground colour.

Each colour can either be specified by an SVG name[3] or by a hexadecimal value (`#rrggbb`) similar as in web documents. The `<fontflags>` can be any combination of the letter `B` (bold), `I` (italic), and `U` (underlined).

Examples of valid formatting instructions are:

```
red
white/#000000
;B
blue;I
#000000/#ffff00;U
```

---

[2] For some details on regular expressions, see B
[3] See `http://www.w3.org/TR/SVG/types.html#ColorKeywords` for a list of valid names.

A.2 Keyboard shortcuts

The use of keyboard shortcuts greatly facilitates typing in and the management of the source and the preview windows. Their use is much more effective than the use of buttons for frequently-used actions.

Below, you'll find the shortcuts for source and preview windows. Note that on Mac OS X, `Ctrl` actually refers to the *Command key*, which is the usual modifier for keyboard shortcuts. Although the keyboard shortcuts are specified with `Ctrl`, this will appear as the *Command-key* symbol in menus. (To refer to the actual *Control key* on the Mac, the shortcut file should use the name `Meta`).

Note that the shortcuts listed below are the default shortcuts for the English interface of TEXworks. Different languages may use different shortcuts.

All the shortcuts can be redefined either to create new shortcuts or to modify the existing ones to match personal uses or change shortcuts not adapted to one particular keyboard layout. The list of possible actions to associate with shortcuts is given after the predefined shortcuts.

To define your own shortcuts, put a file named `shortcuts.ini` in the `<resources>/configuration` folder, next to `auto-indent-patterns.txt`, `delimiter-pairs.txt`, ..., `texworks-config.txt`.

For example, this file could contain:

```
actionHard_Wrap = Shift+F3
actionLast_Page = Ctrl+End
actionFirst_Page = Ctrl+Home
actionWrap_Lines = F3
actionLine_Numbers = F4
actionBalance_Delimiters = F9
```

The first line defines that using [Shift ⇑] [F3] should open the hardwrap dialogue box in the source window; the second ([Ctrl] [End]) should bring you to the last page and [Ctrl] [Home] (third line) should take you to the first page; with [F3] you want to wrap/unwrap lines in the source, with [F4] you will show/hide line numbers and with [F9] you intend to select the text between corresponding delimiters in the source.

A.2.1 Predefined shortcuts

For working in the source window:

Shortcut	Action
Ctrl+'	Go to Preview
Ctrl+=	Show Selection
Ctrl+A	Select All
Ctrl+Alt+S	Save All
Ctrl+B	Balance Delimiters
Ctrl+C	Copy
Ctrl+E	Copy to Find
Ctrl+F	Find...
Ctrl+G	Find Again
Ctrl+H	Find Selection
Ctrl+L	Go to Line...
Ctrl+N	New
Ctrl+O	Open...
Ctrl+Q	Quit TeXworks
Ctrl+R	Replace...
Ctrl+S	Save
Ctrl+Shift+E	Copy to Replace
Ctrl+Shift+N	New from Template...
Ctrl+Shift+R	Replace Again
Ctrl+Shift+S	Save As...
Ctrl+Shift+Z	Redo
Ctrl+Shift+[	Uncomment
Ctrl+Shift+]	Comment
Ctrl+T	Typeset
Ctrl+V	Paste
Ctrl+W	Close
Ctrl+X	Cut
Ctrl+Z	Undo
Ctrl+[	Unindent
Ctrl+\	Hide Output Panel
Ctrl+]	Indent

Moving the cursor (hold $\boxed{\text{Shift} \Uparrow}$ to select):

Shortcut	Action
→	1 character right
Ctrl+→	1 word right
←	1 character left
Ctrl+←	1 word left
↑	1 line up
↓	1 line down
PgUp	1 screen up
PgDown	1 screen down
Home	Begin of line
Ctrl+Home	Begin of document
End	End of line
Ctrl+End	End of document

For working in the preview window:

Shortcut	Action
Ctrl+'	Go to Source
Ctrl++	Zoom In
Ctrl+-	Zoom Out
Ctrl+1	Actual Size
Ctrl+2	Fit to Width
Ctrl+3	Fit to Window
Ctrl+Backspace	Clear
Ctrl+C	Copy
Ctrl+F	Find...
Ctrl+G	Find Again
Ctrl+J	Go to Page...
Ctrl+N	New
Ctrl+O	Open...
Ctrl+Q	Quit TeXworks
Ctrl+Shift+F	Full Screen
Ctrl+Shift+N	New from Template...
Ctrl+Shift+Z	Redo
Ctrl+T	Typeset
Ctrl+V	Paste
Ctrl+W	Close
Ctrl+X	Cut

Shortcut	Action
Ctrl+Z	Undo
End	Last Page
Home	First Page
PgDown	Next Page
PgUp	Previous Page

A.2.2   Actions listed alphabetically

actionAbout_Scripts	actionPaste
actionAbout_TW	actionPlace_on_Left
actionActual_Size	actionPlace_on_Right
actionApply_to_Selection	actionPreferences
actionAutoIndent_None	actionPrevious_Page
actionAuto_Follow_Focus	actionQuit_TeXworks
actionBalance_Delimiters	actionRedo
actionClear	actionRemove_Aux_Files
actionClose	actionReplace
actionComment	actionReplace_Again
actionCopy	actionRevert_to_Saved
actionCopy_to_Find	actionSave
actionCopy_to_Replace	actionSave_All
actionCut	actionSave_As
actionFind	actionScroll
actionFind_Again	actionSelect_All
actionFind_Selection	actionSelect_Image
actionFirst_Page	actionSelect_Text
actionFit_to_Width	actionSettings_and_Resources
actionFit_to_Window	actionShow_Hide_Console
actionFont	actionShow_Scripts_Folder
actionFull_Screen	actionShow_Selection
actionGoToHomePage	actionSide_by_Side
actionGo_to_Line	actionSmartQuotes_None
actionGo_to_Page	actionStack
actionGo_to_Preview	actionSyntaxColoring_None
actionGo_to_Source	actionTile
actionHard_Wrap	actionTo_Lowercase
actionIndent	actionTo_Uppercase
actionLast_Page	actionToggle_Case

```
actionLine_Numbers          actionTypeset
actionMagnify               actionUncomment
actionManage_Scripts        actionUndo
actionNew                   actionUnindent
actionNew_from_Template     actionUpdate_Scripts
actionNext_Page             actionWrap_Lines
actionNone                  actionWriteToMailingList
actionOpen                  actionZoom_In
actionOpen_Recent           actionZoom_Out
```

A.2.3 Actions listed by menu

For the source window:

Edit
```
actionBalance_Delimiters    actionRedo
actionClear                 actionSelect_All
actionCopy                  actionTo_Lowercase
actionCut                   actionTo_Uppercase
actionNone                  actionToggle_Case
actionPaste                 actionUndo
actionPreferences
```

File
```
actionClose                 actionRemove_Aux_Files
actionNew                   actionRevert_to_Saved
actionNew_from_Template     actionSave
actionOpen                  actionSave_All
actionOpen_Recent           actionSave_As
actionQuit_TeXworks
```

Format
```
actionApply_to_Selection    actionLine_Numbers
actionAutoIndent_None       actionSmartQuotes_None
actionComment               actionSyntaxColoring_None
actionFont                  actionUncomment
actionHard_Wrap             actionUnindent
actionIndent                actionWrap_Lines
```

Help
```
actionAbout_TW              actionSettings_and_Resources
actionGoToHomePage          actionWriteToMailingList
```

	Scripts
`actionAbout_Scripts`	`actionShow_Scripts_Folder`
`actionManage_Scripts`	`actionUpdate_Scripts`

	Search
`actionCopy_to_Find`	`actionGo_to_Line`
`actionCopy_to_Replace`	`actionReplace`
`actionFind`	`actionReplace_Again`
`actionFind_Again`	`actionShow_Selection`
`actionFind_Selection`	

	Typeset
`actionTypeset`	

	Window
`actionAuto_Follow_Focus`	`actionShow_Hide_Console`
`actionGo_to_Preview`	`actionSide_by_Side`
`actionPlace_on_Left`	`actionStack`
`actionPlace_on_Right`	`actionTile`

For the preview window:

	Edit
`actionClear`	`actionPreferences`
`actionCopy`	`actionRedo`
`actionCut`	`actionUndo`
`actionPaste`	

	File
`actionClose`	`actionOpen`
`actionNew`	`actionOpen_Recent`
`actionNew_from_Template`	`actionQuit_TeXworks`

	Help
`actionAbout_TW`	`actionSettings_and_Resources`
`actionGoToHomePage`	`actionWriteToMailingList`

	Scripts
`actionAbout_Scripts`	`actionShow_Scripts_Folder`
`actionManage_Scripts`	`actionUpdate_Scripts`

	Search
`actionFind`	`actionFind_Again`

	Typeset	
`actionTypeset`		

	View	
`actionActual_Size`	`actionLast_Page`	
`actionFirst_Page`	`actionNext_Page`	
`actionFit_to_Width`	`actionPrevious_Page`	
`actionFit_to_Window`	`actionZoom_In`	
`actionFull_Screen`	`actionZoom_Out`	
`actionGo_to_Page`		

	Window	
`actionGo_to_Source`	`actionSide_by_Side`	
`actionPlace_on_Left`	`actionStack`	
`actionPlace_on_Right`	`actionTile`	

A.2.4 Other actions

In addition to the static actions listed above, there are also actions for scripts. These are dynamic in nature, as they are created on-the-fly for the available scripts (which may change when you install scripts, remove them, or change some settings). All of these actions are of the form `Script:␣<script_title>`, where `<script_title>` must be replaced appropriately. If you have a script that shows up as `My Script`, for example, the corresponding action would be named `Script:␣My␣Script`.

A.3   Roots for completion

We give here the keywords for auto-completion as they are supplied by TₑXworks. They are given in the files `tw-basic.txt`, `tw-context.txt` (initially empty) and `tw-latex.txt` in the `<resource>\completion` folder.

We give them in three columns: the first two show the keywords, the third the (LA)TₑX code produced. In some cases there is only the code, this means that you can start to enter the (LA)TₑX code and try to complete it with ⌨ .

During completion, the system inserts line feeds and puts the cursor at the first place where one has to enter information to complete the typing. To represent the line feeds we used $\mathcal{R}$ and $\mathcal{I}$ for the input point.

So, a line like "\begin{abstract}$\mathcal{RIR}$\end{abstract} •" should be interpreted as

---

`\begin{abstract}`

`\end{abstract}`•

---

with the cursor being position on the central, empty line.

It is possible to see that the keywords have some pattern. The mathematical variables have a keyword starting with `x`, when they are in a mathematical environment; when they are used alone in the text you add `d` in front. For example, `xa` and `dxa` give `\alpha`, if there is a capital there is a `c`, as `xo` for `\omega` and `xco` for `\Omega`. The keywords for environments start with `b`: `bali` for `\begin{align}` (`b` is a mnemonic for `\begin`). When the environment has possible options, there is one or more `o` added to the base name: `bminp` gives `\begin{minipage}{}...` while `bminpo` gives `\begin{minipage}[]{}...`.

Keywords defined in `tw-basic.txt` (defined in T_EX):

---

xa	\xa	\alpha
xb	\xb	\beta
	\bsk	\bigskip
		\bigskip$\mathcal{R}$
xch	\xch	\chi
xd	\xd	\delta
xcd	\xcd	\Delta
xe	\xe	\epsilon
xet	\xet	\eta
xg	\xg	\gamma
xcg	\xcg	\Gamma
		\hskip
		\indent
		\input
xio	\xio	\iota
xl	\xl	\lambda
xcl	\xcl	\Lambda
	\msk	\medskip
		\medskip$\mathcal{R}$
xm	\xm	\mu
		\noindent
xn	\xn	\nu

xo	\xo	\omega
xco	\xco	\Omega
		\par
xcph	\xcph	\Phi
xph	\xph	\phi
xp	\xp	\pi
xcp	\xcp	\Pi
xcps	\xcps	\Psi
xps	\xps	\psi
xr	\xr	\rho
		\scriptsize
xs	\xs	\sigma
xcs	\xcs	\Sigma
		\smallskip$\mathcal{R}$
	\ssk	\smallskip$\mathcal{R}$
xt	\xt	\tau
tex	\tex	\TeX
		\TeX
texs	\texs	\TeX\
		\TeX\
xth	\xth	\theta
xcth	\xcth	\Theta
xu	\xu	\upsilon
xcu	\xcu	\Upsilon
xve	\xve	\varepsilon
xvph	\xvph	\varphi
xvp	\xvp	\varpi
xvr	\xvr	\varrho
xvs	\xvs	\varsigma
xvth	\xvth	\vartheta
		\vskip
xcx	\xcx	\Xi
xx	\xx	\xi
xz	\xz	\zeta

Keywords defined in `tw-latex.txt` (defined in LATEX):

ncol	\ncol	&
dd	\dd	\($\mathcal{I}$ \) •
dxa	\dxa	\(\alpha\)
dxb	\dxb	\(\beta\)
dxch	\dxch	\(\chi\)
dxd	\dxd	\(\delta\)
dxcd	\dxcd	\(\Delta\)
dxe	\dxe	\(\epsilon\)
dxet	\dxet	\(\eta\)
dxg	\dxg	\(\gamma\)
dxcg	\dxcg	\(\Gamma\)
dxio	\dxio	\(\iota\)
dxl	\dxl	\(\lambda\)
dxcl	\dxcl	\(\Lambda\)
dxm	\dxm	\(\mu\)
dxn	\dxn	\(\nu\)
dxo	\dxo	\(\omega\)
dxco	\dxco	\(\Omega\)
dxcph	\dxcph	\(\Phi\)
dxph	\dxph	\(\phi\)
dxp	\dxp	\(\pi\)
dxcp	\dxcp	\(\Pi\)
dxcps	\dxcps	\(\Psi\)
dxps	\dxps	\(\psi\)
dxr	\dxr	\(\rho\)
dxs	\dxs	\(\sigma\)
	\dxcs	\(\Sigma\)
dxt	\dxt	\(\tau\)
dxth	\dxth	\(\theta\)
dxcth	\dxcth	\(\Theta\)
dxu	\dxu	\(\upsilon\)
dxcu	\dxcu	\(\Upsilon\)
dxve	\dxve	\(\varepsilon\)
dxvph	\dxvph	\(\varphi\)
dxvp	\dxvp	\(\varpi\)
dxvr	\dxvr	\(\varrho\)
dxvs	\dxvs	\(\varsigma\)
dxvth	\dxvth	\(\vartheta\)
dxx	\dxx	\(\xi\)

dxcx	\dxcx	\(\Xi\)
dxz	\dxz	\(\zeta\)
		\addtocounter{$\mathcal{I}$}{ • }
	\adc	\addtocounter{$\mathcal{I}$}{ • }
adcount		\addtocounter{$\mathcal{I}$}{ • }$\mathcal{R}$
	\adl	\addtolength{$\mathcal{I}$}{ • }
		\addtolength{$\mathcal{I}$}{ • }
adlen		\addtolength{$\mathcal{I}$}{ • }$\mathcal{R}$
		\author{$\mathcal{I}$}$\mathcal{R}$
		\begin{
babs	\babs	\begin{abstract}$\mathcal{RIR}$\end{abstract} •
balis	\balis	\begin{align*}$\mathcal{RIR}$\end{align*} •
baliats	\baliats	\begin{alignat*}{$\mathcal{I}$}$\mathcal{R}$ • $\mathcal{R}$\end{alignat*} •
baliat	\baliat	\begin{alignat}{$\mathcal{I}$}$\mathcal{R}$ • $\mathcal{R}$\end{alignat} •
baliedat	\baliedat	\begin{alignedat}$\mathcal{RIR}$\end{alignedat} •
baliedato	\baliedato	\begin{alignedat}[$\mathcal{I}$]$\mathcal{R}$ • $\mathcal{R}$\end{alignedat} •
balied	\balied	\begin{aligned}{$\mathcal{I}$}$\mathcal{R}$ • $\mathcal{R}$\end{aligned} •
bali	\bali	\begin{align}$\mathcal{RIR}$\end{align} •
bapp	\bapp	\begin{appendix}$\mathcal{RIR}$\end{appendix} •
barr		\begin{array}$\mathcal{RIR}$\end{array} •
bbmat	\bbmat	\begin{bmatrix}$\mathcal{RIR}$\end{bmatrix} •
bcase	\bcase	\begin{cases}$\mathcal{RIR}$\end{cases} •
bcent	\bcent	\begin{center}$\mathcal{RIR}$\end{center} •
bcenum	\bcenum	\begin{compactenum}$\mathcal{R}$\item$\mathcal{RIR}$\end{compactenum} •
bcenumo	\bcenumo	\begin{compactenum}[$\mathcal{I}$]$\mathcal{R}$\item$\mathcal{R}$ • $\mathcal{R}$\end{compactenum} •
bcitem	\bcitem	\begin{compactitem}$\mathcal{R}$\item$\mathcal{RIR}$\end{compactitem} •
bcitemo	\bcitemo	\begin{compactitem}[$\mathcal{I}$]$\mathcal{R}$\item$\mathcal{R}$ • $\mathcal{R}$\end{compactitem} •
bdes	\bdes	\begin{description}$\mathcal{R}$\item[$\mathcal{I}$]$\mathcal{R}$ • $\mathcal{R}$\end{description} •
bdoc	\bdoc	\begin{document}$\mathcal{RRIRR}$\end{document}
benu	\benu	\begin{enumerate}$\mathcal{R}$\item$\mathcal{RIR}$\end{enumerate} •
benuo	\benuo	\begin{enumerate}[$\mathcal{I}$]$\mathcal{R}$\item$\mathcal{R}$ • $\mathcal{R}$\end{enumerate} •
beqns	\beqns	\begin{eqnarray*}$\mathcal{RIR}$\end{eqnarray*} •
beqn	\beqn	\begin{eqnarray}$\mathcal{RIR}$\end{eqnarray} •
bequs	\bequs	\begin{equation*}$\mathcal{RIR}$\end{equation*} •
bequ	\bequ	\begin{equation}$\mathcal{RIR}$\end{equation} •
bfig	\bfig	\begin{figure}$\mathcal{RIR}$\end{figure} •
bfigo	\bfigo	\begin{figure}[$\mathcal{I}$]$\mathcal{R}$ • $\mathcal{R}$\end{figure} •
bflaligs	\bflaligs	\begin{flalign*}$\mathcal{RIR}$\end{flalign*} •
bflalig	\bflalig	\begin{flalign}$\mathcal{RIR}$\end{flalign} •
bfll	\bfll	\begin{flushleft}$\mathcal{RIR}$\end{flushleft} •
bflr	\bflr	\begin{flushright}$\mathcal{RIR}$\end{flushright} •

bgaths	\bgaths	\begin{gather*}$\mathcal{RIR}$\end{gather*} •
bgathed	\bgathed	\begin{gathered}$\mathcal{RIR}$\end{gathered} •
bgathedo	\bgathedo	\begin{gathered}[$\mathcal{I}$]$\mathcal{R}$ • $\mathcal{R}$\end{gathered} •
bgath	\bgath	\begin{gather}$\mathcal{RIR}$\end{gather} •
bite	\bite	\begin{itemize}$\mathcal{R}$\item$\mathcal{RIR}$\end{itemize} •
biteo	\biteo	\begin{itemize}[$\mathcal{I}$]$\mathcal{R}$\item$\mathcal{R}$ • $\mathcal{R}$\end{itemize} •
blett	\blett	\begin{letter}{$\mathcal{I}$}$\mathcal{R}$ • $\mathcal{R}$\end{letter} •
blist	\blist	\begin{list}{$\mathcal{I}$}{ • }$\mathcal{R}$\item$\mathcal{R}$ • $\mathcal{R}$\end{list} •
bminpo	\bminpo	\begin{minipage}[$\mathcal{I}$]{ • }$\mathcal{R}$ • $\mathcal{R}$\end{minipage} •
bminp	\bminp	\begin{minipage}{$\mathcal{I}$}$\mathcal{R}$ • $\mathcal{R}$\end{minipage} •
bmults	\bmults	\begin{multline*}$\mathcal{RIR}$\end{multline*} •
bmult	\bmult	\begin{multline}$\mathcal{RIR}$\end{multline} •
bpict	\bpict	\begin{picture}$\mathcal{RIR}$\end{picture} •
bpmat	\bpmat	\begin{pmatrix}$\mathcal{RIR}$\end{pmatrix} •
bquot	\bquot	\begin{quotation}$\mathcal{RIR}$\end{quotation} •
bquo	\bquo	\begin{quote}$\mathcal{RIR}$\end{quote} •
bsplit	\bsplit	\begin{split}$\mathcal{RIR}$\end{split} •
bsubeq	\bsubeq	\begin{subequations}$\mathcal{RIR}$\end{subequations} •
btabb	\btabb	\begin{tabbing}$\mathcal{RIR}$\end{tabbing} •
btbls	\btbls	\begin{table*}$\mathcal{RIR}$\end{table*} •
btabls	\btabls	\begin{table*}$\mathcal{RIR}$\end{table*} •
btablso	\btablso	\begin{table*}[$\mathcal{I}$]$\mathcal{R}$ • $\mathcal{R}$\end{table*} •
btblso	\btblso	\begin{table*}[$\mathcal{I}$]$\mathcal{R}$ • $\mathcal{R}$\end{table*} •
btbl	\btbl	\begin{table}$\mathcal{RIR}$\end{table} •
btabl	\btabl	\begin{table}$\mathcal{RIR}$\end{table} •
btblo	\btblo	\begin{table}[$\mathcal{I}$]$\mathcal{R}$ • $\mathcal{R}$\end{table} •
btablo	\btablo	\begin{table}[$\mathcal{I}$]$\mathcal{R}$ • $\mathcal{R}$\end{table} •
btabs	\btabs	\begin{tabular*}{$\mathcal{I}$}{ • }$\mathcal{R}$ • $\mathcal{R}$\end{tabular*} •
btabx	\btabx	\begin{tabularx}{$\mathcal{I}$}{ • }$\mathcal{R}$ • $\mathcal{R}$\end{tabularx} •
btab	\btab	\begin{tabular}{$\mathcal{I}$}$\mathcal{R}$ • $\mathcal{R}$\end{tabular} •
bbib	\bbib	\begin{thebibliography}{$\mathcal{I}$}$\mathcal{R}$\bibitem{ • }$\mathcal{R}$ • $\mathcal{R}$\end{thebibliography} •
bindex	\bindex	\begin{theindex}$\mathcal{RIR}$\end{theindex} •
btheo	\btheo	\begin{theorem}$\mathcal{RIR}$\end{theorem} •
btitpg	\btitpg	\begin{titlepage}$\mathcal{RIR}$\end{titlepage} •
btrivl	\btrivl	\begin{trivlist}$\mathcal{RIR}$\end{trivlist} •
bvarw	\bvarw	\begin{varwidth}{$\mathcal{I}$}$\mathcal{R}$ • $\mathcal{R}$\end{varwidth} •
bverb	\bverb	\begin{verbatim}$\mathcal{RIR}$\end{verbatim} •
bvers	\bvers	\begin{verse}$\mathcal{RIR}$\end{verse} •
		\bfseries
bfd		\bfseries
bibitemo		\bibitem[$\mathcal{I}$]{ • }$\mathcal{R}$ •

		$\bibitem[\mathcal{I}]\{\bullet\}\mathcal{R}\bullet$
bibitem		$\bibitem\{\mathcal{I}\}\mathcal{R}\bullet$
		$\bibitem\{\mathcal{I}\}\mathcal{R}\bullet$
bibstyle	\bibstyle	$\bibliographystyle\{\mathcal{I}\}$
biblio		$\bibliography\{\mathcal{I}\}$
		$\bibliography\{\mathcal{I}\}$
		$\bottomrule\mathcal{R}$
botr		$\bottomrule\mathcal{R}$
		$\boxed\{\mathcal{I}\}$
		$\caption\{\mathcal{I}\}\mathcal{R}$
		$\cdots$
center		$\centering$
		$\centering$
		$\chapter\{\mathcal{I}\}$
chap		$\chapter\{\mathcal{I}\}\mathcal{R}$
		$\citep\{\mathcal{I}\}$
		$\citet\{\mathcal{I}\}$
		$\cite\{\mathcal{I}\}$
		$\cline\{\mathcal{I}\}$
		$\cmidrule(\mathcal{I})\{\bullet\}$
cmidr		$\cmidrule(\mathcal{I})\{\bullet\}$
cmidro		$\cmidrule[\mathcal{I}](\bullet)\{\bullet\}$
		$\cmidrule[\mathcal{I}](\bullet)\{\bullet\}$
		$\date\{\mathcal{I}\}\mathcal{R}$
		$\ddddot\{\mathcal{I}\}$
		$\dddot\{\mathcal{I}\}$
		$\ddots$
		$\ddot\{\mathcal{I}\}$
		$\documentclass[\mathcal{I}]\{\bullet\}\mathcal{R}$
		$\documentclass\{\mathcal{I}\}\mathcal{R}$
		$\dots$
		$\dotsb$
		$\dotsc$
		$\dotsi$
		$\dotsm$
		$\dotso$
emd		$\em$
em		$\emph\{\mathcal{I}\}$
		$\emph\{\mathcal{I}\}$
		$\end\{\mathcal{I}\}\mathcal{R}$
		$\eqref\{\mathcal{I}\}$

		\fboxrule{$\mathcal{I}$}
		\fboxsep{$\mathcal{I}$}
fbox		\fbox{$\mathcal{I}$}
		\fbox{$\mathcal{I}$}
		\footnotesize
foot		\footnote{$\mathcal{I}$}
		\footnote{$\mathcal{I}$}
frac		\frac{$\mathcal{I}$}{ • }
		\frac{$\mathcal{I}$}{ • }
fboxoo	\fboxoo	\framebox[$\mathcal{I}$][•]{ • }
		\framebox[$\mathcal{I}$][•]{ • }
		\framebox[$\mathcal{I}$]{ • }
fboxo	\fboxo	\framebox[$\mathcal{I}$]{ • }
geometry		\geometry{ • }
		\geometry{ • }
		\headwidth
hw		\headwidth
		\hline$\mathcal{R}$
href		\href{$\mathcal{I}$}{ • }
		\href{$\mathcal{I}$}{ • }
		\hspace*{$\mathcal{I}$}
		\hspace{$\mathcal{I}$}
incgo		\includegraphics[$\mathcal{I}$]{ • }$\mathcal{R}$
		\includegraphics[$\mathcal{I}$]{ • }$\mathcal{R}$
		\includegraphics{$\mathcal{I}$}$\mathcal{R}$
incg		\includegraphics{$\mathcal{I}$}$\mathcal{R}$
		\include{$\mathcal{I}$}$\mathcal{R}$
		\intertext{$\mathcal{I}$}
		\item$\mathcal{R}\mathcal{I}$
ito		\item[$\mathcal{I}$]$\mathcal{R}$ •
		\item[$\mathcal{I}$]$\mathcal{R}$ •
itd		\itshape
		\itshape
lbl	\lbl	\label{$\mathcal{I}$}
		\label{$\mathcal{I}$}
		\Large
		\large
		\LaTeX
latex	\latex	\LaTeX
		\LaTeX\
latexs	\latexs	\LaTeX\

		\LaTeXe
latexe	\latexe	\LaTeXe
latexes	\latexes	\LaTeXe\
		\LaTeXe\
		\ldots
		\listoffigures$\mathcal{R}$
listf	\listf	\listoffigures$\mathcal{R}$
		\listoftables$\mathcal{R}$
listt	\listt	\listoftables$\mathcal{R}$
mboxoo	\mboxoo	\makebox[$\mathcal{I}$][$\bullet$]{$\bullet$}
		\makebox[$\mathcal{I}$][$\bullet$]{$\bullet$}
		\makebox[$\mathcal{I}$]{$\bullet$}
mboxo	\mboxo	\makebox[$\mathcal{I}$]{$\bullet$}
mpar	\mpar	\marginpar{$\mathcal{I}$}
		\marginpar{$\mathcal{I}$}
		\mathbf{$\mathcal{I}$}
mbf	\mbf	\mathbf{$\mathcal{I}$}
mcal	\mcal	\mathcal{$\mathcal{I}$}
		\mathcal{$\mathcal{I}$}
mit	\mit	\mathit{$\mathcal{I}$}
		\mathit{$\mathcal{I}$}
mnorm	\mnorm	\mathnormal{$\mathcal{I}$}
		\mathnormal{$\mathcal{I}$}
		\mathrm{$\mathcal{I}$}
mrm	\mrm	\mathrm{$\mathcal{I}$}
msf	\msf	\mathsf{$\mathcal{I}$}
		\mathsf{$\mathcal{I}$}
		\mathtt{$\mathcal{I}$}
mtt	\mtt	\mathtt{$\mathcal{I}$}
mbox		\mbox{$\mathcal{I}$}
		\mbox{$\mathcal{I}$}
		\mdseries
mdd		\mdseries
		\midrule$\mathcal{R}$
midr		\midrule$\mathcal{R}$
multc	\multc	\multicolumn{$\mathcal{I}$}{$\bullet$}{$\bullet$}
		\multicolumn{$\mathcal{I}$}{$\bullet$}{$\bullet$}
multic		\multicolumn{$\mathcal{I}$}{$\bullet$}{$\bullet$}
nct		\newcolumntype{$\mathcal{I}$}{$\bullet$}
newct		\newcolumntype{$\mathcal{I}$}{$\bullet$}
		\newcolumntype{$\mathcal{I}$}{$\bullet$}

		$\newcommand\{\mathcal{I}\}[\bullet][\bullet]\{\bullet\}\mathcal{R}$
ncmoo		$\newcommand\{\mathcal{I}\}[\bullet][\bullet]\{\bullet\}\mathcal{R}$
newcoo		$\newcommand\{\mathcal{I}\}[\bullet][\bullet]\{\bullet\}\mathcal{R}$
newco		$\newcommand\{\mathcal{I}\}[\bullet]\{\bullet\}\mathcal{R}$
ncmo		$\newcommand\{\mathcal{I}\}[\bullet]\{\bullet\}\mathcal{R}$
		$\newcommand\{\mathcal{I}\}[\bullet]\{\bullet\}\mathcal{R}$
		$\newcommand\{\mathcal{I}\}\{\bullet\}\mathcal{R}$
ncm		$\newcommand\{\mathcal{I}\}\{\bullet\}\mathcal{R}$
newc		$\newcommand\{\mathcal{I}\}\{\bullet\}\mathcal{R}$
nenvoo		$\newenvironment\{\mathcal{I}\}[\bullet][\bullet]\{\bullet\}\{\bullet\}\mathcal{R}$
		$\newenvironment\{\mathcal{I}\}[\bullet][\bullet]\{\bullet\}\{\bullet\}\mathcal{R}$
neweoo		$\newenvironment\{\mathcal{I}\}[\bullet][\bullet]\{\bullet\}\{\bullet\}\mathcal{R}$
nenvo		$\newenvironment\{\mathcal{I}\}[\bullet]\{\bullet\}\{\bullet\}\mathcal{R}$
neweo		$\newenvironment\{\mathcal{I}\}[\bullet]\{\bullet\}\{\bullet\}\mathcal{R}$
nenv		$\newenvironment\{\mathcal{I}\}\{\bullet\}\{\bullet\}\mathcal{R}$
newe		$\newenvironment\{\mathcal{I}\}\{\bullet\}\{\bullet\}\mathcal{R}$
		$\newenvironment\{\mathcal{I}\}\{\bullet\}\{\bullet\}\mathcal{R}$
nlen		$\newlength\{\mathcal{I}\}\mathcal{R}$
		$\newlength\{\mathcal{I}\}\mathcal{R}$
newlen		$\newlength\{\mathcal{I}\}\mathcal{R}$
newlin		$\newline\mathcal{R}$
nline		$\newline\mathcal{R}$
		$\newline\mathcal{R}$
npg	\npg	$\newpage\mathcal{R}$
newpg		$\newpage\mathcal{R}$
		$\newpage\mathcal{R}$
		$\newtheorem\{\mathcal{I}\}[\bullet]\{\bullet\}\mathcal{R}$
		$\newtheorem\{\mathcal{I}\}\{\bullet\}\mathcal{R}$
		$\newtheorem\{\mathcal{I}\}\{\bullet\}[\bullet]\mathcal{R}$
		$\nocite\{\mathcal{I}\}$
		$\normalsize$
		$\pagebreak\mathcal{R}$
pgref		$\pageref\{\mathcal{I}\}$
		$\pageref\{\mathcal{I}\}$
pgs		$\pagestyle\{\mathcal{I}\}\mathcal{R}$
		$\pagestyle\{\mathcal{I}\}\mathcal{R}$
pars		$\paragraph^*\{\mathcal{I}\}\mathcal{R}$
		$\paragraph^*\{\mathcal{I}\}\mathcal{R}$
		$\paragraph[\mathcal{I}]\{\bullet\}\mathcal{R}$
paro		$\paragraph[\mathcal{I}]\{\bullet\}\mathcal{R}$
		$\paragraph\{\mathcal{I}\}\mathcal{R}$

par		\paragraph{$\mathcal{I}$}$\mathcal{R}$
pboxo	\pboxo	\parbox[$\mathcal{I}$]{ • }{ • }
		\parbox[$\mathcal{I}$]{ • }{ • }
parboxo		\parbox[$\mathcal{I}$]{ • }{ • }
parbox		\parbox{$\mathcal{I}$}{ • }
		\parbox{$\mathcal{I}$}{ • }
	\pbox	\parbox{$\mathcal{I}$}{ • }
		\pbox{$\mathcal{I}$}{ • }
pbox		\pbox{#INS}{ • }
rboxoo	\rboxoo	\raisebox{$\mathcal{I}$}[•][•]{ • }
		\raisebox{$\mathcal{I}$}[•][•]{ • }
rboxo	\rboxo	\raisebox{$\mathcal{I}$}[•]{ • }
		\raisebox{$\mathcal{I}$}[•]{ • }
		\raisebox{$\mathcal{I}$}{ • }
rbox	\rbox	\raisebox{$\mathcal{I}$}{ • }
ref		\ref{$\mathcal{I}$}
		\ref{$\mathcal{I}$}
rnewcoo		\renewcommand{$\mathcal{I}$}[•][•]{ • }$\mathcal{R}$
rncmoo		\renewcommand{$\mathcal{I}$}[•][•]{ • }$\mathcal{R}$
		\renewcommand{$\mathcal{I}$}[•][•]{ • }$\mathcal{R}$
rnewco		\renewcommand{$\mathcal{I}$}[•]{ • }$\mathcal{R}$
rncmo		\renewcommand{$\mathcal{I}$}[•]{ • }$\mathcal{R}$
		\renewcommand{$\mathcal{I}$}[•]{ • }$\mathcal{R}$
rncm		\renewcommand{$\mathcal{I}$}{ • }$\mathcal{R}$
rnewc		\renewcommand{$\mathcal{I}$}{ • }$\mathcal{R}$
		\renewcommand{$\mathcal{I}$}{ • }$\mathcal{R}$
		\rmfamily
rmc		\rmfamily
		\rule[$\mathcal{I}$]{ • }{ • }
		\rule{$\mathcal{I}$}{ • }
scd		\scshape
		\scshape
secs		\section*{$\mathcal{I}$}$\mathcal{R}$
		\section*{$\mathcal{I}$}$\mathcal{R}$
seco		\section[$\mathcal{I}$]{ • }$\mathcal{R}$
		\section[$\mathcal{I}$]{ • }$\mathcal{R}$
		\section{$\mathcal{I}$}$\mathcal{R}$
sec		\section{$\mathcal{I}$}$\mathcal{R}$
		\setlength{$\mathcal{I}$}{ • }
hw2tw		\setlength{\headwidth}{\textwidth}$\mathcal{R}$
		\sffamily

sfd		\sffamily
		\slshape
sld		\slshape
sqrto	\sqrto	\sqrt[$\mathcal{I}$]{ $\bullet$ }
sqrt	\sqrt	\sqrt{$\mathcal{I}$}
stcount		\stepcounter{$\mathcal{I}$}$\mathcal{R}$
spars	\spars	\subparagraph*{$\mathcal{I}$}
		\subparagraph*{$\mathcal{I}$}
		\subparagraph[$\mathcal{I}$]{ $\bullet$ }
sparo	\sparo	\subparagraph[$\mathcal{I}$]{ $\bullet$ }
		\subparagraph{$\mathcal{I}$}
spar	\spar	\subparagraph{$\mathcal{I}$}
ssecs	\ssecs	\subsection*{$\mathcal{I}$}$\mathcal{R}$
		\subsection*{$\mathcal{I}$}$\mathcal{R}$
sseco	\sseco	\subsection[$\mathcal{I}$]{ $\bullet$ }$\mathcal{R}$
		\subsection[$\mathcal{I}$]{ $\bullet$ }$\mathcal{R}$
ssec	\ssec	\subsection{$\mathcal{I}$}$\mathcal{R}$
		\subsection{$\mathcal{I}$}$\mathcal{R}$
		\subsubsection*{$\mathcal{I}$}$\mathcal{R}$
sssecs	\sssecs	\subsubsection*{$\mathcal{I}$}$\mathcal{R}$
		\subsubsection[$\mathcal{I}$][$\bullet$]$\mathcal{R}$
	\ssseco	\subsubsection[$\mathcal{I}$][$\bullet$]$\mathcal{R}$
ssseco		\subsubsection[$\mathcal{I}$]{ $\bullet$ }$\mathcal{R}$
sssec	\sssec	\subsubsection{$\mathcal{I}$}$\mathcal{R}$
		\subsubsection{$\mathcal{I}$}$\mathcal{R}$
toc	\toc	\tableofcontents$\mathcal{R}$
tableofcontents		\tableofcontents$\mathcal{R}$
		\tableofcontents$\mathcal{R}$
tilde	\tilde	\textasciitilde
bf	\bf	\textbf{$\mathcal{I}$}
		\textbf{$\mathcal{I}$}
---		\textemdash\
--		\textendash\
it	\it	\textit{$\mathcal{I}$}
		\textit{$\mathcal{I}$}
	\rm	\textrm{$\mathcal{I}$}
sc	\sc	\textsc{$\mathcal{I}$}
		\textsc{$\mathcal{I}$}
sf	\sf	\textsf{$\mathcal{I}$}
		\textsf{$\mathcal{I}$}
		\textsl{$\mathcal{I}$}

sl	\sl	\textsl{$\mathcal{I}$}
tt	\tt	\texttt{$\mathcal{I}$}
		\texttt{$\mathcal{I}$}
		\textup{$\mathcal{I}$}
up	\up	\textup{$\mathcal{I}$}
tw	\tw	\textwidth
		\textwidth
		\text{$\mathcal{I}$}
		\thanks{$\mathcal{I}$}$\mathcal{R}$
		\title{$\mathcal{I}$}$\mathcal{R}$
topr		\toprule$\mathcal{R}$
		\toprule$\mathcal{R}$
ttd		\ttfamily
		\ttfamily
upd		\upshape
		\upshape
url		\url{$\mathcal{I}$}
		\url{$\mathcal{I}$}
		\usepackage[$\mathcal{I}$]{ • }$\mathcal{R}$
usepo		\usepackage[$\mathcal{I}$]{ • }$\mathcal{R}$
usep		\usepackage{$\mathcal{I}$}$\mathcal{R}$
		\usepackage{$\mathcal{I}$}$\mathcal{R}$
		\vdots
		\vspace*{$\mathcal{I}$}$\mathcal{R}$
		\vspace{$\mathcal{I}$}$\mathcal{R}$
		{abstract}$\mathcal{RIR}$\end{abstract} •
		{align*}$\mathcal{RIR}$\end{align*} •
		{alignat*}{$\mathcal{I}$}$\mathcal{R}$ • $\mathcal{R}$\end{alignat*} •
		{alignat}{$\mathcal{I}$}$\mathcal{R}$ • $\mathcal{R}$\end{alignat} •
		{alignedat}{$\mathcal{I}$}$\mathcal{R}$ • $\mathcal{R}$\end{alignedat} •
		{aligned}$\mathcal{RIR}$\end{aligned} •
		{aligned}[$\mathcal{I}$]$\mathcal{R}$ • $\mathcal{R}$\end{aligned} •
		{align}$\mathcal{RIR}$\end{align} •
		{appendix}$\mathcal{RIR}$\end{appendix} •
		{array}$\mathcal{RIR}$\end{array} •
		{bmatrix}$\mathcal{RIR}$\end{bmatrix} •
		{cases}$\mathcal{RIR}$\end{cases} •
		{center}$\mathcal{RIR}$\end{center} •
		{compactenum}$\mathcal{R}$\item$\mathcal{RIR}$\end{compactenum} •
		{compactenum}[$\mathcal{I}$]$\mathcal{R}$\item$\mathcal{R}$ • $\mathcal{R}$\end{compactenum} •
		{compactitem}$\mathcal{R}$\item$\mathcal{RIR}$\end{compactitem} •

$\{compactitem\}[\mathcal{I}]\mathcal{R}\backslash item\mathcal{R} \bullet \mathcal{R}\backslash end\{compactitem\} \bullet$
$\{description\}\mathcal{R}\backslash item[\mathcal{I}]\mathcal{R} \bullet \mathcal{R}\backslash end\{description\} \bullet$
$\{document\}\mathcal{R}\mathcal{R}\mathcal{I}\mathcal{R}\mathcal{R}\backslash end\{document\}$
$\{enumerate\}\mathcal{R}\backslash item\mathcal{R}\mathcal{I}\mathcal{R}\backslash end\{enumerate\} \bullet$
$\{enumerate\}[\mathcal{I}]\mathcal{R}\backslash item\mathcal{R} \bullet \mathcal{R}\backslash end\{enumerate\} \bullet$
$\{eqnarray*\}\mathcal{R}\mathcal{I}\mathcal{R}\backslash end\{eqnarray*\} \bullet$
$\{eqnarray\}\mathcal{R}\mathcal{I}\mathcal{R}\backslash end\{eqnarray\} \bullet$
$\{equation\}\mathcal{R}\mathcal{I}\mathcal{R}\backslash end\{equation\} \bullet$
$\{figure\}\mathcal{R}\mathcal{I}\mathcal{R}\backslash end\{figure\} \bullet$
$\{figure\}[\mathcal{I}]\mathcal{R} \bullet \mathcal{R}\backslash end\{figure\} \bullet$
$\{flalign*\}\mathcal{R}\mathcal{I}\mathcal{R}\backslash end\{flalign*\} \bullet$
$\{flalign\}\mathcal{R}\mathcal{I}\mathcal{R}\backslash end\{flalign\} \bullet$
$\{flushleft\}\mathcal{R}\mathcal{I}\mathcal{R}\backslash end\{flushleft\} \bullet$
$\{flushright\}\mathcal{R}\mathcal{I}\mathcal{R}\backslash end\{flushright\} \bullet$
$\{gather*\}\mathcal{R}\mathcal{I}\mathcal{R}\backslash end\{gather*\} \bullet$
$\{gathered\}\mathcal{R}\mathcal{I}\mathcal{R}\backslash end\{gathered\} \bullet$
$\{gathered\}[\mathcal{I}]\mathcal{R} \bullet \mathcal{R}\backslash end\{gathered\} \bullet$
$\{gather\}\mathcal{R}\mathcal{I}\mathcal{R}\backslash end\{gather\} \bullet$
$\{itemize\}\mathcal{R}\backslash item\mathcal{R}\mathcal{I}\mathcal{R}\backslash end\{itemize\} \bullet$
$\{itemize\}[\mathcal{I}]\mathcal{R}\backslash item\mathcal{R} \bullet \mathcal{R}\backslash end\{itemize\} \bullet$
$\{letter\}\{\mathcal{I}\}\mathcal{R} \bullet \mathcal{R}\backslash end\{letter\} \bullet$
$\{list\}\{\mathcal{I}\}\{ \bullet \}\mathcal{R}\backslash item\mathcal{R} \bullet \mathcal{R}\backslash end\{list\} \bullet$
$\{minipage\}[\mathcal{I}]\{ \bullet \}\mathcal{R} \bullet \mathcal{R}\backslash end\{minipage\} \bullet$
$\{minipage\}\{\mathcal{I}\}\mathcal{R} \bullet \mathcal{R}\backslash end\{minipage\} \bullet$
$\{multline*\}\mathcal{R}\mathcal{I}\mathcal{R}\backslash end\{multline*\} \bullet$
$\{multline\}\mathcal{R}\mathcal{I}\mathcal{R}\backslash end\{multline\} \bullet$
$\{picture\}\mathcal{R}\mathcal{I}\mathcal{R}\backslash end\{picture\} \bullet$
$\{pmatrix\}\mathcal{R}\mathcal{I}\mathcal{R}\backslash end\{pmatrix\} \bullet$
$\{quotation\}\mathcal{R}\mathcal{I}\mathcal{R}\backslash end\{quotation\} \bullet$
$\{quote\}\mathcal{R}\mathcal{I}\mathcal{R}\backslash end\{quote\} \bullet$
$\{split\}\mathcal{R}\mathcal{I}\mathcal{R}\backslash end\{split\} \bullet$
$\{subequations\}\mathcal{R}\mathcal{I}\mathcal{R}\backslash end\{subequations\} \bullet$
$\{tabbing\}\mathcal{R}\mathcal{I}\mathcal{R}\backslash end\{tabbing\} \bullet$
$\{table*\}\mathcal{R}\mathcal{I}\mathcal{R}\backslash end\{table*\} \bullet$
$\{table*\}[\mathcal{I}]\mathcal{R} \bullet \mathcal{R}\backslash end\{table*\} \bullet$
$\{table\}\mathcal{R}\mathcal{I}\mathcal{R}\backslash end\{table\} \bullet$
$\{table\}[\mathcal{I}]\mathcal{R} \bullet \mathcal{R}\backslash end\{table\} \bullet$
$\{tabular*\}\{\mathcal{I}\}\{ \bullet \}\mathcal{R} \bullet \mathcal{R}\backslash end\{tabular*\} \bullet$
$\{tabularx\}\{\mathcal{I}\}\{ \bullet \}\mathcal{R} \bullet \mathcal{R}\backslash end\{tabularx\} \bullet$
$\{tabular\}\{\mathcal{I}\}\mathcal{R} \bullet \mathcal{R}\backslash end\{tabular\} \bullet$
$\{thebibliography\}\mathcal{R}\mathcal{I}\mathcal{R}\backslash end\{thebibliography\} \bullet$

$\{\text{theindex}\}\mathcal{RIR}\setminus\text{end}\{\text{theindex}\}$ •
$\{\text{theorem}\}\mathcal{RIR}\setminus\text{end}\{\text{theorem}\}$ •
$\{\text{titlepage}\}\mathcal{RIR}\setminus\text{end}\{\text{titlepage}\}$ •
$\{\text{trivlist}\}\mathcal{RIR}\setminus\text{end}\{\text{trivlist}\}$ •
$\{\text{varwidth}\}\{\mathcal{I}\}\mathcal{R}$ • $\mathcal{R}\setminus\text{end}\{\text{varwidth}\}$ •
$\{\text{verbatim}\}\mathcal{RIR}\setminus\text{end}\{\text{verbatim}\}$ •
$\{\text{verse}\}\mathcal{RIR}\setminus\text{end}\{\text{verse}\}$ •

There are also environment codes (above) without `\begin{` (which is itself a keyword); this allows to finish the environment name alone by ⌨ if one started to input it manually.

Regular expressions

As TeXworks is built on Qt4, the available regular expressions—which are often referred to as **regexp**—are a subset of those found in Qt4. See the site of Qt4[1] for more information. It is possible to find other information about regexps on the net[2] or from books. But pay attention that not all systems (programming languages, editors, ...) use the same set of instructions; there is no "standard set", unfortunately.

B.1 Introduction

When searching and replacing, one has to define the text to be found. This can be the text itself (e.g., "Abracadabra"), but often it is necessary to define the strings in a more generic and powerful way to avoid repeating the same operation many times with only small changes from one time to the next; if, for example, one wants to replace sequences of the letter **a** by ones of the letter **o**, but only those sequences of 3, 4, 5, 6 or 7 **a**; this would require repeating (and slightly adjusting) the find and replace procedure 5 times. Another example: replace all vowels by **§**—again, this would take 5 replace operations. Here come the regular expressions!

A simple character (a or 9) represents itself. But a set of characters can be defined: [**aeiou**] will match any vowel, [**abcdef**] the letters **a**, **b**, **c**, **d**, **e**, and **f**; this last set can be shortened as [**a-f**] using "**-**" between the two ends of the range. This can even be combined: [**a-zA-Z0-9**] will match all letters and all numbers.

---

[1] `http://doc.trolltech.com/4.4/qregexp.html#details`—this section is based on the information provided there

[2] see, for example, Wikipedia

To define a complementary set[3], one uses "`^`": the caret negates the character set if it occurs at the beginning, i.e., immediately after the opening square bracket. [`^`**abc**] matches anything except **a**, **b**, **c**.

B.2   Codes to represent special sets

When using regexps, one very often has to create a search expession which represents other strings in a generic way. If you are looking for a string that matches email addresses, for example, the letters and symbols will vary; still, you could search for any string which corresponds to the structure of an email address (`<text>@<text>.<text>`, roughly). To facilitate this, there are abbreviations to represent letters, figures, symbols, . . .

These codes replace and facilitate the definition of sets; for example, to instead of manually defining the set of digits [**0-9**], one can use "**\d**". The following table lists the replacement codes.[4]

---

[3]A set of characters that are not allowed to occur for this regular expression to match the text

[4]simplified from Qt4 at trolltech, see note 1

Element	Meaning
c	Any character represents itself unless it has a special regexp meaning. Thus c matches the character c.
\c	A special character that follows a backslash matches the character itself except where mentioned below. For example, if you wished to match a literal caret at the beginning of a string you would write "\^".
\n	This matches the ASCII line feed character (LF, Unix newline, used in TeXworks).
\r	This matches the ASCII carriage return character (CR).
\t	This matches the ASCII horizontal tab character (HT).
\v	This matches the ASCII vertical tab character (VT; almost never used).
\xhhhh	This matches the Unicode character corresponding to the hexadecimal number hhhh (between 0x0000 and 0xFFFF). \0ooo (i.e., zero-ooo) matches the ASCII/Latin-1 character corresponding to the octal number ooo (between 0 and 0377).
. (dot)	This matches any character (including newline). So if you want to match the dot character iteself, you have to escape it with "\.".
\d	This matches a digit.
\D	This matches a non-digit.
\s	This matches a white space.
\S	This matches a non-white space.
\w	This matches a word character or "_").
\W	This matches a non-word character.
\1, ...	The n-th back-reference, e.g. \1, \2, etc.; used in the replacement string with capturing patterns—see below

Using these abbreviations is better than describing the set, because the abbreviations remain valid in different alphabets.

Pay attention that the end of line is often taken as a white space. Under TeXworks the end of line is referred to by "\n".

B.3   Repetition

One doesn't work only on single letters, digits, symbols; most of the time, these are repeated (e.g., a number is a repetition of digits and symbols—in the right order).

To show the number of repetitions, one uses a so called "quantifier": **a{1,1}** means at least one and only one **a**, **a{3,7}** means between at least 3 and at most 7 **a**; **{1,1}** is redundant, of course, so **a{1,1} = a**.

This can be combined with the set notation: **[0-9]{1,2}** will correspond to at least one digit and at most two, the integer numbers between 0 and 99. But this will match any group of 1 or 2 digits within any arbitrary string (which may have a lot of text before and after the integer); if we want this to match only if the whole string consists *entirely* of 1 or 2 digits (without any other characters preceding or following them), we can rewrite the regular expression to read **^[0-9]{1,2}$**; here, the **^** specifies that any match must start at the first character of the string, while the **$** says that any matching substring must end at the last character of the string, so the string can only be comprised of one or two digits (**^** and **$** are so-called "assertions"—more on them later).

Here is a table of quantifiers.[5] E represents an arbitrary expression (letter, abbreviation, set).

E{n,m}	Matches at least **n** occurrences of the expression and at most **m** occurrences of the expression.
E{n}	Matches exactly **n** occurrences of the expression. This is the same **E{n,n}** or as repeating the expression n times.
E{n,}	Matches at least **n** occurrences of the expression.
E{,m}	Matches at most **m** occurrences of the expression.
E?	Matches zero or one occurrence of E. This quantifier effectively means *the expression is optional* (it may be present, but doesn't have to). It is the same as **E{0,1}**.
E+	Matches one or more occurrences of E. This is the same as **E{1,}**.
E*	Matches zero or more occurrences of E. This is the same as **E{0,}**. Beware, the ***** quantifier is often used by mistake instead of the **+** quantifier. Since it matches zero or more occurrences, it will match even if the expression is not present in the string.

B.4   Alternatives and assertions

When searching, it is often necessary to search for alternatives, e.g., apple, pear, or cherry, but not pineapple. To separate the alternatives, one uses **|**: apple|pear|cherry. But this will not prevent to find pineapple, so we have to

---

[5]simplified from Qt4 at trolltech, see note 1

specify that apple should be standalone, a whole word (as is often called in the search dialog boxes).

To specify that a string should be considered standalone, we specify that it is surrounded by word separators/boundaries (begin/end of sentence, space), like **\bapple\b**. For our alternatives example we will **group** them by parentheses and add the boundaries **\b(apple|pear|cherry)\b**. Apart from **\b** we have already seen ˆ and **$** which mark the boundaries of the whole string.

Here a table of the "assertions" which do not correspond to actual characters and will never be part of the result of a search. [6]

ˆ	The caret signifies the beginning of the string. If you wish to match a literal ˆ, you must escape it by writing \ˆ
$	The dollar signifies the end of the string. If you wish to match a literal **$**, you must escape it by writing \$
\b	A word boundary.
\B	A non-word boundary. This assertion is true wherever \b is false.
(?=E)	Positive lookahead. This assertion is true if the expression **E** matches at this point.
(?!E)	Negative lookahead. This assertion is true if the expression **E** does not match at this point.

Notice the different meanings of ˆ as assertion and as negation inside a character set!

B.5   Final notes

Using rexexp is very powerful, but also quite dangerous; you could change your text at unseen places and sometimes reverting to the previous situation is not possible entirely. If you immediately see the error, you can try ⌨Ctrl ⌨Z .

Showing how to exploit the full power of regexp would require much more than this extremely short summary; in fact it would require a full manual on it own.

Also note that there are some limits in the implementation of regexps in TEXworks; in particular, the assertions (ˆ and $) only consider the whole file, and there are no look-behind assertions.

Finally, do not forget to "tick" the regexp option when using them in the *Find* and *Replace* dialogs and to un-tick the option when not using regexps.

---

[6]simplified from Qt4 at trolltech, see note 1

C

Compiling T_EXworks

A complete guide how to compile T_EXworks is far beyond the scope of this manual. However, most users should find precompiled versions suitable for their system come either with their T_EX distribution or their operating system. If this is not the case, several precompiled versions can also be downloaded from `http://code.google.com/p/texworks/downloads/list`.

Compiling T_EXworks yourself is only necessary if your system is not (yet) supported, if you want to always have the latest features (and bugs), or generally want to help in improving T_EXworks further. To this end, there are some documents giving detailed instructions to compile T_EXworks on different machines.

*nix `http://code.google.com/p/texworks/wiki/Building`

Mac `http://code.google.com/p/texworks/issues/detail?id=107`

Win `http://code.google.com/p/texworks/wiki/BuildingOnWindowsMinGW`

Acknowledgements

Microsoft, Windows, and the Windows logo are registered trademarks of Microsoft Corporation in the United States and other countries.

Apple, Mac, and Mac OS are trademarks of Apple Inc., registered in the U.S. and other countries.

Google is a trademark of Google Inc.

Unless noted otherwise, all icons are either part of TeXworks or part of the Tango Icon Library (`http://tango.freedesktop.org/Tango_Icon_Library`).

Bibliography

[1] **D. Knuth**, *The T_EXbook*, Addison Wesley, 1986-1992

[2] **D. Knuth**, *The METAFONT book*, Addison Wesley, 1986-1992

[3] **L. Lamport**, *LaTeX: A Document Preparation System*, Addison Wesley, 1985 (LaTeX 2.09), 1994 (LaTeX 2_ε)

[4] **M. Goossens, F. Mittelbach & A. Samarin**, *The LaTeX Companion*, Addison Wesley, 1994

[5] **M. Goossens**, *The XeTeX Companion*, July 2009, `http://xml.web.cern.ch/XML/lgc2/xetexmain.pdf`

[6] **D. J. Perry**, *Creating Scholarly Multilingual Documents Using Unicode, OpenType, and XeTeX*, June 2009, `http://scholarsfonts.net/xetextt.pdf`

Index