

TEA Manual

Peter "Roxton" Semiletov

peter.semiletov@gmail.com

TEA Manual

by Peter "Roxton" Semiletov

The book is a public domain

Table of Contents

1. An introduction is necessary	1
2. Interface.....	2
2.1. Main window	2
2.2. Save and Open dialog windows	2
2.3. About drag and drop	2
3. How TEA works with different charsets?	4
4. How to edit and preview SRT-subtitles.....	5
5. The transcendental shining of the File menu	6
6. Edit menu, or how to cut without scissors and violence.....	10
6.1. Common editing operations.....	10
7. Search menu	11
7.1. Common searching functions.....	11
7.2. Marking menu	12
8. Markup	13
8.1. About all that stuff	13
8.2. HTML Tools.....	13
8.3. Common.....	13
8.4. List stuff	14
9. Functions	15
9.1. Common.....	15
9.2. LaTeX.....	15
9.3. IDE	15
9.4. Calendar	16
9.5. Spell-checking.....	16
9.6. UNIVersal Text AnalyZer.....	17
9.7. User menu	17
9.8. Snippets	18
9.9. Format	18
9.10. Insert.....	19
9.11. Case.....	21
9.12. Numbers	21
9.13. String.....	21
9.14. Sort	24
9.15. Filters	24
9.16. Morse code.....	25
9.17. Tools.....	26
10. Run.....	27
11. Nav[igation]	28
12. View.....	29
13. Preferences, I suppose... ..	30
13.1. Intro	30
13.2. Switches	30
13.3. Encoding	31

13.4. Functions	31
13.5. Interface.....	31
13.6. Editor.....	31
13.7. Browsers.....	32
13.8. Paths	32
13.9. Maintenance	32
14. What is Kwas?.....	33
14.1. Actions	33
14.2. Ins to editor	33
15. Command line options.....	34
16. About the author.....	35
A. Resources	36

Chapter 1. An introduction is necessary

Tibi et igni

At the autumn of the year 2000 I started to develop the text editor (for Windows) which called *TEA* (something like Text Editing and Authoring program). It was popular in ex-USSR countries such as Ukraine, Russia, Belarus and others; I made 11 versions until 2004. Then I stopped *TEA* for Windows (http://http://my.elvisti.com/roxtton/tea_w32.html) and started the creation of *TEA for Linux* (currently known simple as *TEA*). It is my first Linux program, and, actually, my first C-program at all :) I am not a programmer. I'm a writer and a journalist, and I want the text editor that fit to my needs. So I try to do a such one. *By the way, at 2006 TEA (Win32) was resurrected, but it has no any localization except Russian. TEA (Win32) is a public domain, written on Delphi (yes, the proprietary one!) using Scintilla as the text editing engine..*

I completely understand that my English is truly BAD (articles, tenses and other stuff), but I tries to write this manual good as I can.

If you have any questions, ideas, propositions or patches, feel free to write me - peter.semiletov@gmail.com (<mailto:peter.semiletov@gmail.com>). By the way, I understand English good as my native Russian, so don't look at how terrible I write.

Please read this manual entire, because *TEA* in some cases is a very non-standart editor.

Now playing: Nirvana - I Hate Myself And I Want To Die

Chapter 2. Interface

2.1. Main window

Some words about an interface. Attention please! Our airplane is safe but a bit unusual. Here is a layout of TEA control elements, from the top to the bottom:

- Window bar (sic!).
- Menu. Behold and explore.
- The toolbar. Hardcoded, non-customizable, but nice and neat. You can turn it off from **Preferences > Interface > Show the main toolbar**. And I remember my promise to make the configurable toolbars, please wait.
- Text-tabs area.
- Text field called **Logmemo** destined for the information output.
- Text entry (I call it **the Famous text entry**) for input of miscellaneous values such as a text for searching, tags for enclosing selections with them, etc. The **Execute** button directly from the right of the famous text entry doing the same function as pressing Enter in the text entry.

For example, if you want go to the line 77, just enter 77 into the Famous text entry and press Alt-G. And, if you need to find something, type there some text and press Enter or Ctrl-F.

The **Open** button is very useful if you want to open a file quickly. Just paste filename (with a full path) to the Famous text entry and press the **Open** button. Please note that in that case TEA uses an encoding which selected at **Preferences > Encoding > Default charset for File open**.

- Statusbar, the nice and wide one. Look at that and have a joy! ;)

2.2. Save and Open dialog windows

TEA implements slightly non-standart Save/Open dialog windows. They are big by default, and you can define their size at Preferences > Interface. The standart GTK+2 file chooser widget has no filename entry when it works in "File open"-mode. The TEA's dialog has it. There are two entries available: **Directory** and **Filename**. The **Directory** entry is used in the dual way. To navigate into the some directory, enter an URI, a full path or just directory name into this entry, and then press Enter or push the **Go to** button. To create a new directory IN THE CURRENT ONE, enter the directory name into the **Directory** entry and press the **Create new** button.

To save or open a file use the **Filename** entry - it works similar to the same thing in the KDE's file selector. Even the **Enter** key works! ;)

2.3. About drag and drop

When you drop some text file[s] into the TEA window, that file[s] will open in the editor. If a file is the image, TEA inserts it into the current text document as the IMG-tag (or media object for Docbooc). It works fine with Konqueror, Gqview, etc. By drag and drop action, TEA supposes that you drop files with the encoding that you select in **Preferences > Encodings > Default charset for File open.**

Now playing: Scorn - Logghi Barogghi - Weakener

Chapter 3. How TEA works with different charsets?

TEA can handle all charsets that Linux supports through the *iconv library*. Basically you can use two character sets for the file save operation - there are UTF-8 and the current locale's charset. For the file opening action you can select them and in addition there is a one more option - **autodetect**.

When you choose an autodetection, TEA tries to determine the actual encoding automatically, and if it fails, you need to select the proper encoding manually. The autodetection function has a polymorphic nature and divided into more than one sub-functions. For example, TEA has a simply **autodetect** and **Japanese autodetect** functions. When you choose the first one, you must also select some charsets which TEA needs to check. All that charsets are available at **Preferences > Encodings > Autodetect section** - you may see a lot of checkboxes there.

So when you open the file, you can choose the **autodetect** charset, and TEA will try to autodetect encodings which you select at "Autodetect" Preferences section.

And another case is **Japanese autodetect** - it is separated from the "usual" **autodetect** function due to its inner implementation.

How to add more charsets to the list of available encodings? Go to **File > Preferences > Encoding**, switch to **Configure charsets** page. There are two lists - **Available** and **Selected**. Select charset in **Available** list and press **Add** button to add this encoding to **Selected** list. Charsets from **Selected** will be available in File open/save dialogs, Kwas file manager et cetera.

To remove an item from **Selected** list, select it and press **Delete** button between the lists.

Menus/lists with available charsets updates: for Open/Save dialogs - when you reopen them; for Kwas - when you start a new Kwas window.

TEA has an almost obsolete **View > Co** menu. From this menu you can switch encoding for the current document. Encodings are the same as in "File open/save as" dialogs. When you switch an encoding, it means that the current file's content will be reloaded with the different encoding. *So use it carefully, only if you want to change an encoding when you see some kind of crap in the place of a text.*

Chapter 4. How to edit and preview SRT-subtitles

With TEA you can edit and preview SRT-subtitles. To work with them properly, you need:

1. Open SRT-file as a usual text file.
2. "Open" a movie file using the **File > Open different > Open movie**.
3. Edit your SRT-file. Save it.
4. When you activate **View > Preview in Mplayer**, TEA finds the *start time* of the current subtitle section, and runs Mplayer with movie which you have selected before by **Open movie**. TEA also talk Mplayer to jump to the time what is needed for the current subtitle section (actually, after such "rewind", due to Mplayer's nature, it shows the next subtitle AFTER the current one, so choose in TEA previous subtitle to view correctly the current one. A weird sentence, isn't it? But, try to use and you'll understand).

Chapter 5. The transcendental shining of the File menu

New - makes a new empty document in the UTF-8 encoding. Of course you can save it in the charster what you like.

New Kwas - opens a new Kwas window. Please read the chapter What is Kwas?.

Crapbook - Crapbook is a simply text file that stored at `$HOME/.config/tea/crapbook.txt`. You can call it by Alt-M. TEA saves this file automatically when you close TEA or `crapbook.txt` itself. I think this is useful for writing some notes, quotes and the similar stuff.

Add to... > Add to Bookmarks - adds a new bookmark record into Bookmarks list. Each record includes fields: filename, charset, position. To edit the bookmarks file (which is a simple text file with a list of items) go to **File > Manage config files > Bookmarks file**. Edit it and then save. Changes will be applied immediately.

Add to... > Add to the autosaving list - adds the current file to a special list of files. These files will be saved automatically when you close it. To edit the file with that list go to **File > Manage config files > Autosaving list**

This feature is useful if you want to keep some files as collectors for a different info. You can, for example, make a shortcut in KDE to run something like `tea ~/my_links.txt`, and put `my_links.txt` into the autosaving files list.

Open - opens a file[s]. There are a charset selector in the File open dialog window. Choose the correct encoding or you will be angry. For details about how TEA deals with charsets please read the chapter How TEA works with different charsets?

By open multiply files, they must be in the same encoding (the charset that selected), or shit can happen. I cannot say about the nature of that hypothetical shit, but now you are warned.

TEA uses File open widget introduced in GTK+ 2.4. It supports some keyboard shortcuts for different actions:

- Location popup - Control-L
- Directory up - Alt-Up
- Directory down - Alt-Down
- Goto home directory - Alt-Home

About RTF files - yes, TEA can OPEN them. Not to save. To open it correctly you must define the default encoding for RTF. It can be done in **Preferences > Encoding > Default charset for RTF**. If you are an English-speaking user, do not worry, it is not affects you much.

An another question is *how* TEA opens RTF's. It really sucks, but it works. To just read RTF quickly, use TEA. But if you want to edit them, then use OOO, or KWord, or AbiWord, or whatever else.

By the way, TEA can open OpenDocument, OpenOffice.org Writer, KWord, Microsoft Word and AbiWord files in the read-only mode. So you can use TEA as the viewer of those files, or to convert them into the plain text (using **Save as** with another filename). I can notice that TEA reads KWord slightly better than OOO and Abiword formats. To open Microsoft Word, you need to have the Antiword (<http://www.winfield.demon.nl>) program installed. And to open OpenDocument, OpenOffice.org Writer and KWord you need the following programs: *gunzip*, *bzip2* and *tar*.

About archived files. Currently TEA supports in the read-only mode files compressed with .gz, bz2, zip, tar.gz, tar.bz2 and tar.zip. There are two limitations for that. In a such mode TEA can open only UTF-8 files (or with a normal English text). And, TEA can handle an archive if it contains just a single file.

Open different - from this menu you can quickly open snippets, sessions and templates files for editing. Here you may find also the **Open movie** menu item. It is destined for SRT-subtitles editing. For the further info read the chapter about View menu.

The **Open different** has also an item called **Open from the Famous text entry**. How it works? Enter or paste the full file name into the Famous text entry, then use this function and enjoy.

Templates - from this menu user templates are available. All templates are stored by default in *\$HOME/.config/tea/templates*. To save the file as a template, use **Save different > Save as template** menu item.

Sessions - the stuff like in the Opera browser - you can store lists of files, and then load them as well. There are also the one menu item - **Open different > Open session file** - to open a session file in TEA like a normal text file, without loading a filelist from it. TEA stores to each session 3 elements per file - the filename, the encoding and the cursor position.

To store the files as a session, use **Save different > Save session** menu item.

Now playing: Yab Yum - Dead Moon Ritual (Source Mutation Mix)

Save different > Backup - TEA makes copy of the currently editing text as the backup-file.

Save - you know, right? Just a one note - document will be saved in same encoding as you've opened it.

Save as - by the way, in this dialog you can select from the list any of the encodings that available. So if you want to convert your file into another charset, use this feature.

Save different > Save as template - saves a current file as template in a default templates directory `$HOME/.config/tea/templates`.

Save different > Save version - save a copy of the current file with the name, based on this file name and current date and time.

Manage utility files - from here you can open directly any of TEA configuration files (saved at `$HOME/.config/tea/`). The changes will be applied after editing and saving. I do not recommend you edit the Main config, because TEA has a fine tool for that - the **Preferences window**, but do as you wish.

Current music is: Nirvana - Talk To Me

Manage utility files > Autoreplace words file - here you can open the file with words for autoreplacing (which you can turn ON at **Preferences > Editor > Autoreplace**). This file contains lines in a very simply format: `string1=string2`. For example:

```
Linux=GNU/Linux
sxe=straight edge
FSOL=Future Sound Of London
hl=highlighting
```

After you save this file, the inner list of autoreplacing will be updated. To use it, just turn autoreplacing on, and start to type your text. All words from the list there will be replaced/expanded when you enter their short equivalent and press SPACE or any punctuation key. *Autoreplacing is case-sensitive.*

Manage utility files > Hotkeys config - using this file you can assign your hotkey/shortcut for any menu item. All hotkeys are stored in the `$HOME/.config/tea/tea_hotkeys` file. This file **MUST** contain only UTF-8 data. All changes will be applied after you save this file. If the new hotkey overlaps an older one for the same menu item, then changes will work after the TEA restart. Pre-defined hotkeys are not changeable.

Each line of `tea_hotkeys` has a simple format: `menu item caption=hotkey`. For example:

```
Reverse=Alt Shift R
Read this fine manual=F1
```

Some notes. The menu item's caption is case-sensitive. So you must write it exactly as in the TEA menu. Otherwise, the hotkey is not case-sensitive, so you can write freely something like that: "shift ctrl F5" or so. Available modifiers are: *Alt, Ctrl, Shift*. You must divide them by spaces.

You can assign a hotkey to any menu item, even for dynamical-created items like bookmarks.

Chapter 6. Edit menu, or how to cut without scissors and violence.

6.1. Common editing operations.

Indent/Unindent - it moves text to the right or to the left on one tabulation character or some spaces (look at **Preferences, Editor page**). For an indent you can also use a TAB key. Both of functions works with a single line or multiply lines as well.

Append to the Crapbook - add the selection to the Crapbook. It's not important that Crapbook is opened or not - this function works in any case.

Swop - swap the selected text with a clipboard content. I.e. text from the selection moves to the clipboard, and at the same time the content of clipboard is moving to the place of a selection.

Move line up/down - doing that. Please note that there is no need to select a line. The operation works with a current line (where the cursor is placed).

Copy all - copy all text to the clipboard.

Copy current URL - before using that, you must point with cursor to the URL at your text.

Chapter 7. Search menu

7.1. Common searching functions

Want to find something? Write it to the **Famous text entry** and press Enter or Ctrl-F. Want to **Find next**? Press F3. Note that TEA always search from the current cursor position.

If you need more traditional search tool, you can use the **Search and Replace window**. It is far from the ideal, but somehow usable.

Replace - replaces the current selection with a content of the Famous text entry. The common use of that - find/find next some text, and when the text that founded will be selected, use this function.

Goto line - enter the line number into the famous text entry, and use this function.

Replace all - again, dealing with the Famous text entry. But now in a special format: *text to find~text for replace*. For example: *hell~heck*.

Find and mark find the string (from the Famous text entry) in the current document and mark all found entries.

Find in files. TEA has a simple built-in utility to search text within files. It's a single thread function and it has no any **Stop** button so you'll need some patience while TEA tries to find the text in files. Drink some tea...

Probably that some fields of this window need an explanation. Here it is. The **And where to find** directory field - that's from which directory TEA will start the search. The search is recursive.

File pattern - only two metachars are supported there: * and ?.

To open a file from the list of files those found, double click on the list item.

Also you can save the list of files those found as a TEA session to load files later at any time. Press **Save as a session** button to save the list as a session. Also you can copy all found files to some directory using the **Copy to the directory** button.

TEA searches the text in plain text, KOffice KWD, AbiWord ABI, OpenOffice.org sxw, and OpenDocument ODT files.

For the plain text files, on search, TEA tries to use all character encodings that included at charset list in the File Open dialog window.

Mark lines > n characters. Put the maximum line length to the Famous text entry and use this function. All lines those are larger than N (the value in the Famous text entry) characters will be marked.

Scan for local links - scans current document for href-links to local files, and add them (if any) into the menu **Nav > Links**, from which you can open a text file in TEA, and a picture in the built-in or external viewer (like a **Open at cursor**).

Scan for SRC's - similar to **Scan for local links**, but now it fills **Nav > Links** menu with all local SRC-stuff - for most cases with images. It's very useful if you want to see some banner or another picture and don't want to find it in the code manually.

7.2. Marking menu

Mark. Mark is a special selection mode. It gives you the ability to select text multiply. To *mark* some text, select the text normally and apply **Mark**. To "deselect" the marked text, point to it with the cursor and use **Mark** again. The another way to disable marked selection is the **Unmark all** menu item. You can copy all marked text using **Copy marked** function.

Mark by regexp - search through text by regexp defined at the Famous text entry and mark all words those founded according to regexp. For example, regexp = "p*r". So the words like "peter", "poor", "pair" will be marked. This kind of search is a case-sensitive.

Metachar * means the any characters, and ? stands for an any single one character.

For example, if you define the pattern such that "?og", so the "dog" will be found, but not the "frog". But when you define "*og", TEA marks all words like *dog, frog, bulldog, fog* and other similar to those.

current music: The Pixies - Santo

Mark all fuzzy and empty translations - it's useful for the .po-files editing.

current music: The Pixies - Evil Hearted You

Mark each n-th line - use this function to mark each n-th line. Type the number into the Famous text entry first.

Mark lines those contains a string - marks all lines those contains a string from the Famous text entry.

Chapter 8. Markup

8.1. About all that stuff

Let me say some words about Markup menu. It can act in the several modes according to the mode choosed from the **Markup mode** menu. There are HTML, XHTML, Wikipedia, LaTeX and Docbook modes are available. But! Not all menu items of the Markup menu supports all markup modes. For example, the **Make a table** function can create tables with HTML, Wikipedia and Docbook rules, but not a TeX table.

8.2. HTML Tools

Document height - calculate a total document weight including all SRC-staff (images, flashes etc). Look for the result at the Logmemo.

Build-in template - insert the hardcoded HTML template into the current document.

Enclose selected link into tags - doing that. If you want to enclose a *www.foo.bar* or *ftp.foo.bar*, or *mailto:foo@bar.com* with a *a href*-tag, so use this function and be happy.

Strip tags - kill all tags from HTML-document.

Convert tags to entities - if you want to show some HTML-tagged code in your HTML-document, so select your code and use this function. You'll get, for example:

```
&lt;b&gt;demo&lt;/b&gt;
```

Text to HTML - this function converts a plain text into the HTML-formatted code using CSS.

8.3. Common

Some of functions those described below are works in a dual-way - if there no text selected, then will be inserted a pair of tags, otherwise the selected text will be enclosed with tags.

Close the current tag - use this fuction to close the current tag. Type any tag with the ">" at the end, then use this menu item. Also you can turn on the automatic tag closing at **Preferences > Editor > Autoclose Tags**. All that stuff works with Docbook, HTML etc.

Comment - enclose the selection into the comment tags. It works fine for TeX, HTML, CSS, Pascal and C/C++ files, according to syntax rules.

Color - opens the color selection dialog. If no text selected, then a HTML-color inserts into the cursor position, otherwise the selected text will be embraced with a span-tag with the color property CSS.

Image - opens the file open dialog. You can select an image[s], which will be inserted as img-tags (with filled attributes such as height and width) or image object (Docbook) into the current document. There are supported all formats what GTK+ 2 usually understands - PNG, JPEG, GIF, WBMP etc.

Table stuff > Make table - I don't like any wizards, but I always have dreamed about the tool to create empty tables quickly. And I did it. Again we use the Famous text entry. It is very simply. Just type: *rows count~columns count*, and apply this function. For example, we want to make 2x4 table - 2 rows, with 4 cells per each row. So our template will be such of this: 2~4

Catch it? And now for something completely different.

8.4. List stuff

Itemized list - works in LaTeX and Docbook modes. An example of the itemized list:

- Scorn
- Nirvana
- Guano Apes

Enumerated list - works in LaTeX (as enumerated list) and Docbook (as orderedlist) modes. An example of the enumerated list:

1. Scorn
2. Nirvana
3. Guano Apes

Chapter 9. Functions

9.1. Common

To use text-processing functions, you must select some text first. There are exceptions, when the function process the whole text. But usually it works with the selection (except functions those inserting some values). Sometimes it works in two modes - with selection or a whole text, or selection or a current word. Try and you will be enlightened.

Document stats - outputs text statistics for the whole text or the selection into the Logmemo.

9.2. LaTeX

How to work with TeX/LaTeX documents in TEA? First of all, set up **Preferences > Commands > LaTeX** options. They are good by default, but check them. And DO NOT add the &-character after the command lines. Of course you must have all that LaTeX stuff installed.

To make *dvi*-file from the current LaTeX document, use **Process with LaTeX** function and see to Logmemo for errors. If all right, now you can convert *dvi* to *ps* or *PDF*. Use **dvi to PS** or **dvi to PDF** functions. TEA saves all *dvi*'s and resulting files into the directory where your LaTeX document is saved. TEA cares about their filenames, don't think about it.

To view a **dvi**-file, use **View dvi** menu item. To view PostScript and PDF files, use **View PS** and **View PDF** functions.

So the usual way to produce PDF from LaTeX with TEA is: **Process with LaTeX**, then **dvi to PDF**, then **View PDF** to view the result.

Or, alternatively, if you have *pdflatex* installed, use **Process with pdflatex** and then **View PDF**.

9.3. IDE

Among other features, TEA supports some kind of IDE (Intergated Development Environment) functions, such as project handling. TEA also can call **make** utility and run a binary file.

How it works? TEA IDE project does not contain any files except of a single project file. To create it use a **New project** function. You will see a dialog window with some entries. There are:

Project name - the name of your project.

Makefile directory - that where TEA can find makefile of your project to call **make** utility.

Source directory - where you keep source files of the project.

Target executable - filename (without the full path) of the binary file. TEA supposes that this file is in the Source directory. TEA runs **target executable** when you activate **Run** menu item.

Project properties menu item - use it to edit properties of the current project (do not forget to save it after that).

When you **Make** your program using TEA, it shows error and warning messages in the Logmemo. You can double-click on some error message string, and TEA jump to that line at the source file. Colors of error message lines you can set at **Preferences > Colors > IDE** section.

Refresh tags with ctags - creates the tags file using **ctags** at the directory of the current file.

In the addition to IDE functions you can use **View > Switch header/source** menu item. Note: it works only if the source and header files are in the same directory. Only C/C++ files are supported. Also you can use **Nav > Browse the symbol's declaration** to jump to the declaration of the C/C++ function that is under the cursor. It requires *tags* file from **ctags**. If TEA can't find this file, TEA run **ctags** to create the *tags* file. Otherwise TEA uses the existing *tags* file, even if it is outdated. For sure you always can refresh it with **IDE > Refresh ctags**.

9.4. Calendar

TEA has a good calendar (actually, it is a GTK+ 2 widget). But TEA provide some additional functionality. With **Days between** section you can calculate how many days are between two selected dates. Select the first date in the calendar and press the **date1** button. Then select another date and press the **date2** button. Press **Calculate** button to see the result at Logmemo.

Below the calendar widget you see another buttons beginning from **Insert...**. With them you can insert a date into the current document in different formats.

Current music is: Guano Apes - Electic Nights

9.5. Spell-checking

Spell-checker languages - TEA uses Aspell (<http://aspell.net>) for the spell-checking. At this menu you can find the list of dictionaries which installed for aspell (from the distro or so). To spell check, use one of those menu items. It works with the whole text, not with the selection only.

The menu item with the language name is used also to set the default spell-checker language. Default value is "en". To do a simple check with that default language, use **Spell check** menu item.

Suggest a word - shows the list with suggestions for the current misspelled word (the word under cursor). To use this function, use **Spell check** first.

Possibly incorrect words will be colored with the color you can define in **Preferences > Colors**.

TEA can just show you errors. You must fix them for yourself, by the hand. When you fix the error you can see that the fixed word is colored as before. That is because the nature of TEA spellchecker - it updates manually, so to update highlighting select the same spell checking a menu item again. To turn error marks off, use **View > Hide error marks**.

9.6. UNiversal Text AnalyZer

What is UNITAZ? It is a fine tool for the text analysis. It can show you the list of words from the current document, and show the count of each word. **Call UNITAZ with sorting abc** - do analysis and then sort words list, **Call UNITAZ plain** - do analysis without sorting. **Call UNITAZ with sorting by count** - the same, but with sorting by the word's count.

Extract words - extracts all words from the current document. It is useful, for example, to make some sort of a dictionary or so. Warning! All that functions are relatively slow, so be patient and do not think that TEA has frozen. Wait, read an interesting book, drink some cola, click here and there with your mouse, read an interesting book again, think twice about the Universe and the place of our planet in the Milky Way galaxy, drink some cola, read an int...

Also UNITAZ shows how many words are total, and amount of *unique* words - words count WITHOUT repetitions of that words.

9.7. User menu

In TEA you can create your own menu to run some programs or open with them a current file. To edit the user menu go to the **File > Manage config files > User-menu config**. It opens the user menu

configuration file. Edit it and save. The menu will update. The format of this file is a simple one. There are a list with something like:

```
Opera=opera %s &
Kwrite=kwrite %s &
```

So, as you understand, the "%s"-macro means the current file name. To start some program as a background process, you must add a &-sign to the end of line. Or else TEA in the frozen state will wait for the program termination.

9.8. Snippets

A snippet is the piece of code, which you can insert into the text. TEA keeps each snippet as a single file in *\$HOME/.tea/snippets* directory. File name of a snippet is a menu item in Snippets-menu, i.e. the content of Snippets menu is the list of files from *\$HOME/.tea/snippets*. To create a new snippet, you should do:

1. Write some text. 2. **File > Save different > Save as snippet**. UTF-8 only! 3. Enjoy :)

You can create a snippet that use a text selection in a some way. For example, you want to make a snippet which encloses the selected text into some HTML-tags. The %s macro represents a text selection. Here is the example of such snippet:

```
<a href="%s">%s</a>
```

When this snippet will be applied, %s will be replaced with a selected text. If there no text is selected, snippet content will be inserted into the text.

9.9. Format

Browser to text. What it does? It reformates a text copied from the HTML-browser. Internally, this function deletes blanks between paragraphs and inserts spaces before each paragraph. You can define the format of the paragraph at the Famous text entry. The idea comes when Sveta started to do that manually and wasted a lot of time.

Kill formatting - kills the formatting (tabs, new lines, double spaces etc.)

Wrap raw at position - do the same as doing **fold** when it without the -s parameter. I.e. it wraps each line at position, which you must enter into the famous text entry.

Wrap on spaces at position - do the same as doing **fold -s**. It wraps the each line (preserving words dividing) at the position, which you must enter into the famous text entry.

9.10. Insert

***Now playing* string from Amarok** - does this. Please note that you can customize the format-string at **Preferences > Functions > Now Playing string format**. There is *%s* macro that means artis and song names.

Dump menu - write all menu items names into the new file. It is useful when you want to add some entries into the hotkeys config.

Date/Time - inserts a date/time in a giver format. The format must be defined by you in the **Preferences > Functions > Date And Time Format**

A quote from *Edition 0.10, last updated 2001-07-06, of The GNU C Library Reference Manual, for Version 2.2.x of the GNU C Library* (the quoted text is edited, to make it short, by Roxton):

- *%a* - The abbreviated weekday name according to the current locale.
- *%A* - The full weekday name according to the current locale.
- *%b* - The abbreviated month name according to the current locale.
- *%B* - The full month name according to the current locale.
- *%c* - The preferred calendar time representation for the current locale.
- *%C* - The century of the year. This is equivalent to the greatest integer not greater than the year divided by 100.
- *%d* - The day of the month as a decimal number (range 01 through 31).
- *%D* - The date using the format *%m/%d/%y*.
- *%e* - The day of the month like with *%d*, but padded with blank (range 1 through 31).
- *%F* - The date using the format *%Y-%m-%d*. This is the form specified in the ISO 8601 standard and is the preferred form for all uses.
- *%g* - The year corresponding to the ISO week number, but without the century (range 00 through 99). This has the same format and value as *%y*, except that if the ISO week number (see *%V*) belongs to the previous or next year, that year is used instead.
- *%G* - The year corresponding to the ISO week number. This has the same format and value as *%Y*, except that if the ISO week number (see *%V*) belongs to the previous or next year, that year is used instead.
- *%h* - The abbreviated month name according to the current locale. The action is the same as for *%b*.
- *%H* - The hour as a decimal number, using a 24-hour clock (range 00 through 23).
- *%I* - The hour as a decimal number, using a 12-hour clock (range 01 through 12).

- %j - The day of the year as a decimal number (range 001 through 366).
- %k - The hour as a decimal number, using a 24-hour clock like %H, but padded with blank (range 0 through 23). This format is a GNU extension.
- %l - The hour as a decimal number, using a 12-hour clock like %I, but padded with blank (range 1 through 12). This format is a GNU extension.
- %m - The month as a decimal number (range 01 through 12).
- %M - The minute as a decimal number (range 00 through 59).
- %n - A single \n (newline) character.
- %p - Either AM or PM, according to the given time value; or the corresponding strings for the current locale. Noon is treated as PM and midnight as AM.
- %P - Either am or pm, according to the given time value; or the corresponding strings for the current locale, printed in lowercase characters. Noon is treated as pm and midnight as am.
- %r - The complete calendar time using the AM/PM format of the current locale.
- %R - The hour and minute in decimal numbers using the format %H:%M.
- %s - The number of seconds since the epoch, i.e., since 1970-01-01 00:00:00 UTC. Leap seconds are not counted unless leap second support is available. This format is a GNU extension.
- %S - The seconds as a decimal number (range 00 through 60).
- %t - A single \t (tabulator) character.
- %T - The time of day using decimal numbers using the format %H:%M:%S.
- %u - The day of the week as a decimal number (range 1 through 7), Monday being 1.
- %U - The week number of the current year as a decimal number (range 00 through 53), starting with the first Sunday as the first day of the first week. Days preceding the first Sunday in the year are considered to be in week 00.
- %V - The ISO 8601:1988 week number as a decimal number (range 01 through 53). ISO weeks start with Monday and end with Sunday. Week 01 of a year is the first week which has the majority of its days in that year; this is equivalent to the week containing the year's first Thursday, and it is also equivalent to the week containing January 4. Week 01 of a year can contain days from the previous year. The week before week 01 of a year is the last week (52 or 53) of the previous year even if it contains days from the new year.
- %w - The day of the week as a decimal number (range 0 through 6), Sunday being 0.
- %W - The week number of the current year as a decimal number (range 00 through 53), starting with the first Monday as the first day of the first week. All days preceding the first Monday in the year are considered to be in week 00.
- %x - The preferred date representation for the current locale.
- %X - The preferred time of day representation for the current locale.
- %y - The year without a century as a decimal number (range 00 through 99). This is equivalent to the year modulo 100.
- %Y - The year as a decimal number, using the Gregorian calendar. Years before the year 1 are numbered 0, -1, and so on.

- `%z` - RFC 822/ISO 8601:1988 style numeric time zone (e.g., -0600 or +0100), or nothing if no time zone is determinable. A full RFC 822 timestamp is generated by the format "`%a, %d %b %Y %H:%M:%S %z`" (or the equivalent "`%a, %d %b %Y %T %z`").
- `%Z` - The time zone abbreviation (empty if the time zone can't be determined).
- `%%` - A literal `%` character.

By default TEA uses "`%d/%m/%Y %T`" as the formatting string, so `date_time=%d/%m/%Y %T`

9.11. Case

UPCASE/lowcase - do that with the selected text or the word under cursor.

cRAcKerIZE - MakES tHe TEXT SuCh THIs.

9.12. Numbers

Arabian to Roman - for example, was *1977*, will be *MCMLXXVII*.

Counter - generates a list of numbers. You must define the format string in the FAMOUS text entry. The format is: `start_value~end_value~[step]`. By the way, default value of "step" = 1. Example (with step = 5):

1~10~5

And as a result we have:

5
10
15
20
25
30
35
40
45
50

9.13. String

Split by the delimiter - actually, this function replaces all delimiters in a text with the new line character. Set the delimiter value in the Famous text entry.

Split after the delimiter - adds the line line character after the each delimiter found in a text. Set the delimiter value in the Famous text entry.

CSV-like table to LaTeX table - use this function if you want to convert a CSV-like table into the LaTeX table format. How it works?

Open the OpenOffice.org Calc or Gnumeric, and save your table sheet as a CSV-file. It is a simple text file where cells are separated with some delimiting character (separator), usually comma (comma-separated file). But if you have commas in cells, use another character as a delimiter. If Gnumeric you should use **Save as - Export as text files** where you can define the separator.

Then open your CSV-file in TEA. Select the text. Use **CSV-like table to LaTeX table**. Voila!

Split to TeX paragraphs - splits the text to TeX paragraphs, adding empty lines between strings. To convert your plain text into TeX, your first step should be use of this function.

Convert usual quotes to TeX quotes - yes it is. Just a one note - it process double-quotes only.

Escape and quote/Unescape and unquote - you know, all that coders stuff, when you need escape/unescape some string or a filename full of that damned whitespaces.

Numerate lines - numerates selected lines. Define the format of numeration in the famous text entry. The format is a weird one: *printf-like format string~step of the counter~initial value of the counter*. The two last parameters are optional and equal to 1 by default. Let's imagine that you have a list of music bands:

Nirvana
Scorn
Napalm Death
Defecation
Neck
JR Ewing
Fall
Meathook Seed
The Doors
Led Zeppelin

Now you want to add a numbering to that list, and in the custom format. So, TEA can do it for you. You write the format string into the FAMOUS text entry. The format string is a very printf-like, i.e. you can use two macros - `%d` for a counter and `%s` for a string. To be more clear - `%d` represents the counter, and `%s` represents a string.

Here is some examples of format-string:

```
%d.)%s
%d.)%s~10
%d.)%s~10~4
```

The second line represents a format-string with a step parameter. Ten is the step value. 3-rd line is the format-string with the step and with the initial value of the counter. It is equal to 4.

You can also use `%d` after `%s`, i.e. `%s (%d)` give us a result:

```
Nirvana (1)
Scorn (2)
Napalm Death (3)
Defecation (4)
Neck (5)
JR Ewing (6)
Fall (7)
Meathook Seed (8)
The Doors (9)
Led Zeppelin (10)
```

Convert tabs to spaces - be sure to enter the tab size into the Famous text entry before. The value of a tab-size is the number of characters per a one tab.

Convert spaces to tabs - and now the Famous text entry content = how many spaces to find for replace each of them to the tab.

Reverse - reverse a text. For example, was *roxton*, will be *notxor*.

Antispam e-mail - makes a selected mailto-link possibly invisible to that damned spammer e-mail harvesters, by converting an address into integer-coded entities. For example, if you will look at the source of that document, so this link (mailto:tea@list.ru) will look like a heap of garbage. I hope that spam harvesters do not understands it. I took an idea from a some issue of LinuxGazette (<http://linuxgazette.net/>).

Remove blank lines - removes blank lines form selected text. I do not think that it is a needful thing, but... Maybe it will be useful to someone.

Remove duplicates - removes duplicated **lines**.

Apply a template to each line - and again we use the Famous text entry. For example, I want to add br-tag at the end of each line of the selected text. So I type into the entry:

```
%s<br>
```

And then I apply that function and get br-tag added to the end of each line. In another case, I want to enclose an each line into a pair of li-tags. I type:

```
<li>%s</li>
```

Then I apply the function. So, as you understand, the *%s* macro = text of the line. And another example:

```
<a href="%s">%s</a>
```

9.14. Sort

Sort lines - Sort lines in the alphabetical order.

Reverse order of lines - better I will show you an example. Was:

```
Line 1
Line 2
Line 3
```

Will be:

```
Line 3
Line 2
Line 1
```

9.15. Filters

Extract at the each line - extracts some text between two delimiters, at the each line. The first and the second separators must be written into the Famous text entry in the form such as: *first_delimiter~last_delimiter*. For example, the text is:

```
g_print ("hello\n");
g_print ("world\n");
```

We put the "~" (double quotes, tilde, double) template into the famous text entry. Select the text and apply this function. After that you shall have as the result:

```
hello\n
world\n
```

Extract at the each line before a separator and **Extract at the each line after a separator** - a custom case of the previous function. But here is only one delimiter is used in the Famous text entry. For example, to cut off all comments in the C++-like text, you can set // as the delimiter value and use the **Extract at the each line before a separator** function.

Kill lines with a phrase. Select a text. Enter some phrase into the Famous text entry. Apply this function. As a result, all *lines* containing your phrase will be deleted from the selection.

Kill all lines except with a phrase. Select some text. Enter some phrase into the famous text entry. Apply this function. As a result, all lines *without* your phrase will be deleted from the selection.

Kill all lines < N characters - Kill in the selection all lines smaller than N (value in the Famous text entry) characters.

Kill all lines > N characters - Kill in the selection all lines smaller than N (value in the Famous text entry) characters.

9.16. Morse code

Encode to Morse code EN. I am not a guru in Morse codes, but I hope that my effort to implement such function is right. With this menu item you can translate the English text into the Morse code. For example, was:

Drum and bass

Will be:

.....

Note that TEA puts a single space between two Morse codes, and two spaces between words (in Morse codes too).

Decode from Morse code EN. The inverse action for the previous function. You can decode any English Morse-coded message. You must to know that TEA supposes that there are single spaces between Morse-codes, and double spaces between words, as in the example written before.

9.17. Tools

Mass encode tool - use this too to encode multiply files from the one encoding to another. Optionally, you can turn on the **Convert ends of lines** option and choose LF or CR/LF mode.

Chapter 10. Run

This menu is designed for run the current file with the external programs, browsers for example. To add some items to menu, go to **File > Manage config files > External programs list config**, edit that file and then save it. The configuration file with that stuff has the simple ini-like format.

An example of the command line for a browser starting: **konqueror=konqueror %s & . %s** is a needful macro for a current filename, so use it in the properly place.

Chapter 11. Nav[igation]

Save the position - it saves the position of cursor. Then you can jump to that position using **Jump to the saved position** menu item.

Links - here are names of local files from links in the current document - you can get them by **Search > Scan for local links** (I've described it previously). More of that, from this menu you can switch to yet opened (in TEA) document.

Go to recent tab - switch to the last accessed tab.

Focus the text/Focus the Famous text entry - useful, if you want to assign some hotkeys for switch between a text tab and the Famous text entry.

Go to the block start/end - really does it. For C-like languages or PHP. It moves the cursor to the current block's start or the end ({ and } characters). I think it is useful.

Open at cursor - if you have the link to a local file at your HTML-document, you can open it with this function. Just point to the filename in the text and use it (pressing F2 will be more quick). If you point to an image-file, it will be opened with built-in image viewer. You can also use an external viewer - setup it with **Preferences > Commands**. For example, it can be the command line **display %s&**. Also in that case you need turn on the option **Preferences > Switches > Use external image viewer**

If the file is not an image, **Open at cursor** open it in TEA as text file with a same encoding as of the current file's one. If the file is opened yet, it becomes the current one.

If you have a link to the local "a name=" - label, press F2 to jump to the place of a definition of that label.

Edit at cursor - the same thing as the previous one, but it calls for a file not the viewer, but the external editor. Go to **Preferences > Commands** and setup there the command line to run a such editor. Example of a command line: **gimp %s&**

Nav > Browse the symbol's declaration - jump to the declaration of the C/C++ function that is under the cursor. It requires *tags* file from **ctags**. If TEA can't find this file, TEA run **ctags** to create the *tags* file. Otherwise TEA uses the existing *tags* file, even if it is outdated. For sure you always can refresh it with **IDE > Refresh tags with ctags**.

Chapter 12. View

Highlighting mode (TEA *original* version only) - from this submenu you can select a highlighting mode locally, for the current document.

Hide highlighting (TEA *original* version only) - hide the highlighting in the document. It removes syntax hl., and spellchecker marks too.

Toggle images visibility - show or hide images near their IMG/SRC tags directly in the text. Currently it works for HTML and XHTML files only.

Refresh highlighting (TEA *original* version only)- TEA *original* has a very ugly ability to highlight a code. With this function, you can update the highlighting manually. Use it often, have a joy.

Word wrap - do you need an explanation? And why I write that paragraph?

Line numbers - the song remains the same...

Preview with Mplayer - this item is related to the chapter How to edit and preview SRT-subtitles.

Imageplane - open the thumbnailed image browser. From there you can view images in the built-in image viewer or insert image-tags/elements into the current document. To view image in a full-size click on the thumbnail. To insert image tag, turn on **Insert tag** option and then click some thumbnail. To set this option by default use **Preferences > Switchers > Imageplane insert tags by default**.

By the way, Imageplane can use thumbnails from *\$HOME/.thumbnails/normal*. It is the place that proposed by Free Desktop specification. Some programs (for example, GQview) already use that directory for storing their thumbnails. TEA does not store any thumbnails, it just reads them from here.

Encoding - use this menu to **reload** the current file with an encoding that you choose.

Chapter 13. Preferences, I suppose...

13.1. Intro

Starting from TEA 3.0, we have the magic **File > Preferences** window. No text-config editing anymore. Use GUI - it is not a heresy. Now I can kick off more of config variables descriptions because they have self-explaining text labels in Preferences window.

Tabbed pages and two big buttons are there. The buttons are: **Close** and **Save and apply**. The first one just closes the Preferences window, and nothing more. But **Save and apply** button doing an important job - it saves all options into the good old plain text config, and then initiates reload the config. If you do not press **Save and apply**, so no changes will stored. TEA do not saves its main config on exit. TEA saves config just when you press the **Save and apply** button.

And now it's a time to give you some basic information about Preferences sections.

13.2. Switches

Miscellaneous switches. There are:

Show the common close button - show or hide a big button (near the main menu) which closes the current document.

Restore the last session on start - when this option is enabled, TEA saves the session on exit and restore it on start.

Save all on the abnormal termination (SIGTERM, SIGHUP, SIGABRT, SIGINT) - save all currently opened files if an abnormal termination happen. But if the memory was corrupted, the corrupted data may be saved. Default: off. But I think that more wise will be use the crash-file and the **Save the crashfile on terminate**. It is similar to the previous one option, but it works with a single file only (the current one), and the text will be saved to the special file called *crash_file* which is located at the TEA configuration directory. You can open it later from **File > Manage utility files > Open the crashfile**.

Determine a scripts highlighting by the content - it is for the syntax highlighting. Turn it on, if you want to auto-highlight Bash-scripts which are without the *.sh*-extension.

Use snippets - snippets works if this option is turned ON. For what is that? In some cases you may fund that TEA scans snippets directory to slowly, and maybe you want turn it off temporary.

Show line numbers - to show them or not to show by default.

Word wrap - to wrap words or not to wrap - yes, by default.

Scan for links on file open. If *on*, then TEA will scans local links automatically on file open. By default is off.

Do backup - Check this box if you want TEA make backup on file save.

13.3. Encoding

Here you can turn on encoding autodetection, and check charsets which you want to autodetect. For details about how TEA deals with charsets please read the chapter How TEA works with different charsets?

13.4. Functions

The format of the Color function - it affects on the result of **Markup > Color** function. Use *@color* macro to define where must be the generated color value, and a *@text* macro as a replacement of the selected text (where it will be inserted into the output string).

13.5. Interface

Here you can set file save/open windows height and width in per cents, and turn off **Show the full path in the window caption** if you want to see just file name in the caption.

Show the common close button - show or hide the shared close button (similar to the close button in Firefox) that closes the current document.

13.6. Editor

Smart Home/End - if enabled, the cursor moves to the paragraph's start/ending.

Tab size in spaces - defines the default tab size, in the count of characters, so, 3 = approx. 3 spaces.

Enable the autosaving - yes, it is. But you need to define the **Autosaving interval**, in minutes. The changes will be applied after you shall restart TEA.

13.7. Browsers

Here you can change default hardcoded command lines to for run a wide range of browsers. Also with the option **Browser for the manual** you may define the browser for TEA manual. In addition to that, for use your browser turn ON **Use this browser** option.

By the way, if no documentation browser defined, TEA will try to find one of good browsers installed on your box and will use that one.

13.8. Paths

Default file saving directory and **Use default saving directory**. Use this entries (and according to these checkboxes) to override default directories for File Save As/Open dialog windows.

Add this extension to the file which is *Saving as* - use this option, if you want to add a default extension to the "saving as" file (in a case when the file name has no extension).

13.9. Maintenance

Put the launcher to the desktop/Remove the launcher from the desktop - TEA supports some FreeDesktop specifications, so if your desktop environment supports it too (as KDE or Gnome does), you can use these two buttons to quickly put or remove the TEA launcher.

Chapter 14. What is Kwas?

Kwas is designed as a replacement of the standart "File open" window in TEA, so do not expect much from this file navigator. You can open it in several windows, you can navigate through the directories, and for sure you can open files in the desired codepage and also view images. In the future I will extend the features list, but for now that is all.

You can use the text entry as the address line for the Kwas. Just type there the path that you need and press the Enter key. Pressing Backspace at the files list = directory up. Kwas has own menus.

14.1. Actions

From here you can:

Refresh Kwas - update the files list. Kwas can't do that automatically.

Get file info - get the information about a current (selected) file. Watch for the output in Logmemo. TEA uses the standart **file** utility for that.

Run with... - run the current (yes, selected) file with a desired program. Type a command into the Famous text entry. The command must be a program name only, i.e. just **opera** or **xmms**.

Now playing: Scorn - Anamnesis - Rarities 1994-1997 - Maker Of Angels

14.2. Ins to editor

There are also **Ins to editor** menu:

Insert image - inserts the selected (active) image file as an image tag/element into the current text document.

Insert link - inserts the selected (active) file as a HREF-tag into the current text document.

Chapter 15. Command line options

--crapbook - start TEA with the Crapbook already opened.

--charset=*charset_name* - set the charset for the opening file. For example, you want to open file1 with CP1251 charset, and file2 with UTF-8. So you write: **tea --charset=cp1251 file1 --charset=utf-8 file2**

Chapter 16. About the author

Realname: Peter Semiletov, nick: Roxton (reversed XOR and NOT operators); was born in Kiev (USSR), at 1977. Currently I work as a journalist writing mostly about Free Software/Open Source and digital sound processing. Apart from my daily job, periodically I sing in underground punk/grindcore bands, write novels and stories, make music using virtual synths and live guitar. Also I'm the drummer in the band Kakayato (it can be translated like as "someone"). I am a vegetarian (due to ethical reason) and an atheist. My home site is here (<http://www.roxton.kiev.ua>)

My preferences are (in the chaotic order):

- Literature: Russian realism classics (Sholohov, Shedrin, Tynyanov, Shookshin, Andreev etc.)
- Music: The Pixies, Nirvana, Scorn, Guano Apes, Modest Musorgsky, Sergey Prokofiev, Bach, Brahms, Napalm Death, Fall and many more.
- Food: fry potato and cola
- Movies: Soviet films (Parajanov, Kalatosov, Danelia, Row, Shookshin, Chuhray, Ptooshko, Gabriadze, Michaeljan), old horror zombie movies, movies by Lynch, Kubrick, McGigan, Gilliam, Kurosava, Kitano, Imoe, Farelly brothers, Coscarelly, Costner, Coen bros., Buster Keaton, Fatty Arbuckle, etc.
- OS distro: Mandriva
- Mail client: Sylpheed-Claws
- Spreadsheet: Gnumeric
- Desktop: KDE
- Media players: Amarok, Mplayer, Beep Media Player
- IDE: KDevelop
- Games: Fallout, Baldur's Gate, Quest For Glory, Mortal Kombat, Resident Evil, Blood, Prehistorik 2, old games for NES (Famicom) and SEGA Genesis/Megadrive; games for Spectrum.
- Browsers: Firefox, Opera
- File managers: mc, Konqueror
- Guitars: relatively cheap Stratocaster clone (made in Taiwan)
- Mediators: Fender .73 mm M (for the electric guitar), Fender .88 M/H (for the acoustic guitar)

Appendix A. Resources

TEA site (<http://tea.linux.kiev.ua>)

Live Journal TEA Community, seems like a calm pool...
(<http://www.livejournal.com/userinfo.bml?user=tea4linux>)

TEA announce newsletter - to subscribe just send an empty letter to
tea-announce-subscribe@linux.kiev.ua (<mailto:tea-announce-subscribe@linux.kiev.ua>)

Peter Semiletov's homepage (in Russian language) (<http://www.roxton.kiev.ua>)