Aquamacs Emacs

Manual

July 14, 2019

Contents

1 Aquamacs Emacs: a User-friendly Emacs Distribution **3**

1.1 Terminology in this Manual . . . 4

2 Tutorial: Aquamacs for Beginners **4**

2.1 What Makes Aquamacs Like Other Text Editors . . . 4

2.2 What Makes Aquamacs Emacs Different from Other Text Editors . . . 4

3 In Depth: The Aquamacs Interface **5**

3.1 Tabs and Windows . . . 6

3.2 Configuring the Toolbars . . . 6

3.3 File . . . 7

3.4 Edit . . . 7

3.5 Options . . . 7

3.5.1 Line wrapping . . . 8

3.5.2 Using the Options key and more... . . . 9

3.5.3 Languages of the World - Dealing with Different Character Sets . 9

3.5.4 The Right Look: Colors and Fonts for Frames and Modes . . . 9

3.6 Window . . . 10

3.7 Tools . . . 10

3.8 Help . . . 11

4 Aquamacs for Emacs Veterans **11**

4.1 Bringing some Emacs 24 behavior back . . . 11

4.2 Key Bindings . . . 11

4.3 Customizing Aquamacs . . . 11

4.4 LaTeX Support . . . 15

4.4.1 Skim: A LaTeX Previewer . . . 15

4.5 Command Line Tools 16
4.6 Vanilla Emacs Mode 17

5 Requirements 17

6 Extending Aquamacs 17
6.1 Plug-in Interface 17
6.2 Load Path 17
6.3 Disabling a plug-in or a load-path 18
6.4 Startup files 18
6.5 Changing Aquamacs source files 19

7 Getting Help 19

8 Aquamacs Lives From Your Donations 20

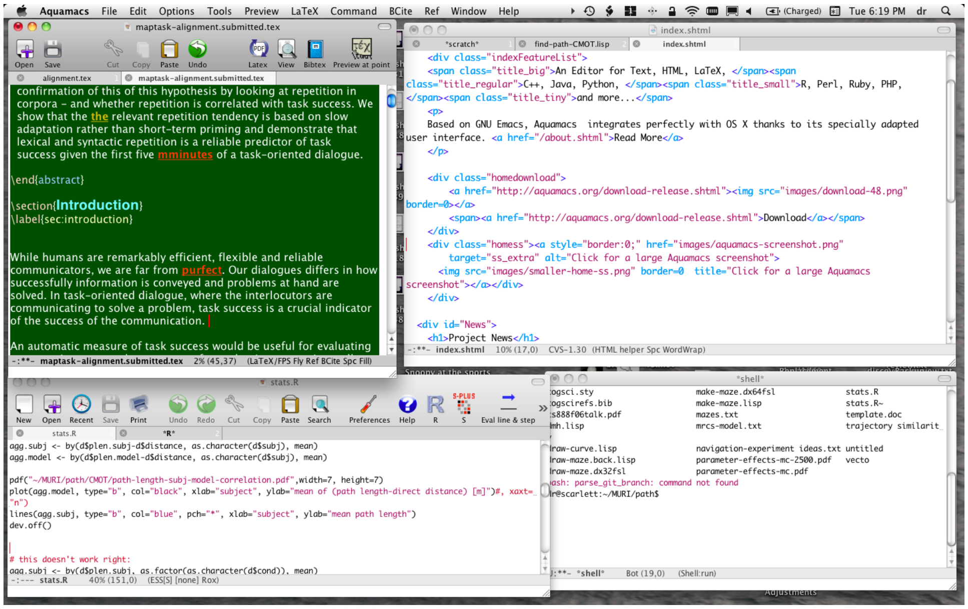
9 Contribute to Aquamacs 20

10 Acknowledgments 20

11 Nightly Development Builds 21

12 Obtaining the Source Code 21

13 Licenses 21



```
confirmation of this of this hypothesis by looking at repetition in
corpora - and whether repetition is correlated with task success. We
show that the the relevant repetition tendency is based on slow
adaptation rather than short-term priming and demonstrate that
lexical and syntactic repetition is a reliable predictor of task
success given the first five mminutes of a task-oriented dialogue.

\end{abstract}

\section{Introduction}
\label{sec:introduction}


While humans are remarkably efficient, flexible and reliable
communicators, we are far from purfect. Our dialogues differs in how
successfully information is conveyed and problems at hand are
solved. In task-oriented dialogue, where the interlocutors are
communicating to solve a problem, task success is a crucial indicator
of the success of the communication.

An automatic measure of task success would be useful for evaluating
-:**-  maptask-alignment.submitted.tex   2% (45,37)   (LaTeX/FPS Fly Ref BCite Spc Fill)
```

```
    <div class="indexFeatureList">
    <span class="title_big">An Editor for Text, HTML, LaTeX, </span><span
class="title_regular">C++, Java, Python, </span><span class="title_small">R, Perl, Ruby, PHP,
</span><span class="title_tiny">and more...</span>
    <p>
    Based on GNU Emacs, Aquamacs  integrates perfectly with OS X thanks to its specially adapted
user interface. <a href="/about.shtml">Read More</a>
    </p>

    <div class="homedownload">
       <a href="http://aquamacs.org/download-release.shtml"><img src="images/download-48.png"
border=0></a>
       <span><a href="http://aquamacs.org/download-release.shtml">Download</a></span>
    </div>
    <div class="homess"><a style="border:0;" href="images/aquamacs-screenshot.png"
       target="ss_extra" alt="Click for a large Aquamacs screenshot">
      <img src="images/smaller-home-ss.png" border=0  title="Click for a large Aquamacs
screenshot"></a></div>
    </div>

  <div id="News">
    <h1>Project News</h1>
-:**-  index.shtml   10% (17,0)   CVS-1.30  (HTML helper Spc WordWrap)
```

```
agg.subj <- by(d$plen.subj-d$distance, as.character(d$subj), mean)
agg.model <- by(d$plen.model-d$distance, as.character(d$subj), mean)

pdf("~/MURI/path/CMOT/path-length-subj-model-correlation.pdf",width=7, height=7)
plot(agg.model, type="b", col="black", xlab="subject", ylab="mean of (path length-direct distance) [m]")#, xaxt=
"n")
lines(agg.subj, type="b", col="blue", pch="*", xlab="subject", ylab="mean path length")
dev.off()


# this doesn't work right:
agg.subj <- by(d$plen.subj, as.factor(as.character(d$cond)), mean)
-:---  stats.R      40% (151,0)    (ESS[S] [none] Rox)
```

```
ogsci.sty                make-maze.dx64fsl               stats.R
ogscirefs.bib            make-maze.lisp                  stats.R~
s888f06talk.pdf          mazes.txt                       template.doc
mh.lisp                  mrcs-model.txt                  trajectory similarit

raw-curve.lisp           navigation-experiment ideas.txt untitled
raw-maze.back.lisp       parameter-effects-mc-2500.pdf   vecto
raw-maze.dx32fsl         parameter-effects-mc.pdf
ash: parse_git_branch: command not found
r@scarlett:~/MURI/path$
J:**-  *shell*      Bot (19,0)     (Shell:run)
```

Figure 1: Aquamacs combines the legendary power of Emacs with user-friendly customizations to provide a more Aqua-specific user experience.

1 Aquamacs Emacs: a User-friendly Emacs Distribution

Aquamacs is an freely-available Aqua-native build of the powerful Emacs text editor (http://www.gnu.org/software/emacs/emacs.html). By "Aqua-native," we mean more than just the fact that this version of Emacs runs as a standard OS X application. Aquamacs features extensive customization that enables it to conform better with Apple's standard Human Interface Guidelines (HIG) than standard versions of the editor do.

Emacs is a text editor of legendary power and configurability, but it also has an enormously complex interface that, while consistent across platforms, is usually at odds with the specific interface conventions of the particular platform on which it is being used. The original GNU version of Emacs for the Mac, called Carbon Emacs, is no different.

Aquamacs Emacs implements the standard OS X keyboard shortcuts and other interface conventions, integrating Emacs into the Aqua environment to a far greater degree than other versions of Emacs. This allows Mac users who might be unfamiliar with Emacs' complex standard interface to harness its amazing editing power in a familiar way.

You can always download the latest version of Aquamacs from the project home page, http://aquamacs.org. Just download the disk image (DMG), move the Aquamacs application bundle to your hard drive, and launch.

This documentation aims to introduce Aquamacs to novice users of Aquamacs Emacs, to help them get started with this powerful text editor. The documentation also aims to introduce Aquamacs to experienced users of Emacs, who may find aspects of its interface inconsistent with their experience.

The Aquamacs documentation will focus on the following areas:

- What's New in this Release
- Tutorial: Aquamacs for Beginners
- In-Depth: The Aquamacs Interface
- Aquamacs for Emacs Veterans

Our hope is that using Aquamacs will be a rewarding experience both for new users, who come to appreciate the power of Emacs without the steep learning curve, and for experienced Emacs users, who may find Emacs' integration into the Aqua environment an unexpectedly pleasant surprise.

1.1 Terminology in this Manual

GNU Emacs uses a terminology that is different from what users of modern, graphical environments are used to. Windows become *frames*, and documents are held in *buffers*. We will concentrate on the Emacs terminology in this manual in places where this is not confusing.

Aquamacs Emacs is an extensive *distribution* of GNU Emacs with modified defaults – almost up to the point where it could be called a *fork* - a completely new program. However, Aquamacs always contains the latest version of GNU Emacs. It is useful to understand the relationship between Emacs and Aquamacs. In this manual, we will use the term *Emacs* to refer to the core which is used on many different operating systems and built and distributed by the GNU Project. We will use the term *Aquamacs Emacs* (or just *Aquamacs*) to refer to the present implementation.

2 Tutorial: Aquamacs for Beginners

2.1 What Makes Aquamacs Like Other Text Editors

When you first launch Aquamacs, you will see that it is like many other text editors: you can type text, cut and paste text, and save and close a file using the menubar or standard OS X keyboard shortcuts (Command-S for save, Command-X for cut, Command-V for paste, and so on). If you are writing one of the many text formats that is supported by Aquamacs, such as HTML, you will also note Aquamacs' use of *syntax coloring*, which sets certain parts of the text—such as HTML markup—in a different color than the text content. This makes editing the text and adjusting the markup easier.

If you are used to the way Emacs does things, you may be surprised about how Aquamacs works: Aquamacs differs significantly from Emacs. While the useful Emacs key bindings are supported and Emacs packages usually just work, Aquamacs is adapted to be comparable to younger text editors (than Emacs). It also comes with a range of packages that give you greater comfort in editing certain types of files.

2.2 What Makes Aquamacs Emacs Different from Other Text Editors

If you look at some of the menu items and keyboard shortcuts, you will see some of the features that make Emacs different from other text editors. Although Aquamacs

OS X Term	Emacs Term
Window	Frame
Tab/pane	Window
Document	Buffer
Cursor	Point
Mouse pointer	Pointer
Keyboard shortcut	Key (binding)

Table 1: Key Emacs terms and their Apple counterparts.

has been designed to present many of these features in an Aqua-friendly way, it does not hide these features. Aquamacs is a complete editing environment.

- **Sophisticated text processing.** Aquamacs features text editing capabilities that go far beyond the average text editor. For instance, Aquamacs features several kinds of search and replace: it can replace text incrementally, it can search and replace text by complex patterns of characters (regular expressions) and not just by word matching, and so on. Aquamacs also features support for virtually every kind of text file imaginable: computer code such as C/C++, HTML, LaTeX, XML, and other formats.

- **Buffers.** One of the features that makes Emacs such a productive editing environment for experienced users is the concept of *buffers.* A buffer is, simply said, a document that is being edited. It can be displayed in any window, and you can even display a buffer twice. But buffers don't just hold text files. They can hold messages from a program that's running, they can be shown in a window that you can use to actually send commands that Aquamacs executes, and other functions. The Buffers menu in Aquamacs Emacs allows you to switch quickly between windows, to send execute or preview the code you are writing with a couple of keystrokes, and to monitor logs of commands you are executing.

- **Integration with additional tools.** Aquamacs' "Tools" menu provides access to file comparison and version control, compiling and debugging of program code, the ability to read e-mail and newsgroups, and more.

In addition to its large number of features, Aquamacs also defines some interface terms differently than other OS X applications. See Table 1 for more information.

This list provides just a small sample of the functionality available in Emacs. Aquamacs' customizations make Emacs much easier to learn; it is possible to get started and become productive quickly. However, harnessing all of Emacs' power, even with assistance from the familiar Aqua user interface, will take time.

3 In Depth: The Aquamacs Interface

In this section, we will walk through the Aquamacs interface step by step, and will introduce relevant points about how Aquamacs Emacs solves particular editing problems in a distinctive way. We will specifically emphasize the functions that Aquamacs offers as opposed to GNU Emacs, which readers might be familiar with anyways.

3.1 Tabs and Windows

Aquamacs has an optional tabbed interface for organizing multiple open buffers. A tab is a GUI handle for a particular buffer. If you have "Show Tabs" checked in the Options menu, then each displayed buffer and any buffers that are subsequently created will be associated with a tab in the tab bar, just above the text area. (Note that the tab bar is hidden whenever it would display only a single tab. You may want to turn off "Show Buffers in New Frames" in the Options menu, in order to get full benefit from tabs.) If the tab bar cannot display all tabs at once, scroll bars appear to allow you to choose the displayed subset of tabs. Additional tab-related functions are available from the Window menu and from the tab bar's context menu.

Tabs in Aquamacs do not act like containers – one can't open a file in an existing tab. (This is in contrast to tab implementation in a number of popular internet browsers.) Rather, tabs provide a convenient way to navigate existing buffers. Because Emacs allows you to simultaneously display the same file in multiple windows or frames, it is possible to have multiple tabs (in different windows) that are associated with the same buffer.

To access tabs quickly, the key commands A-M-0,1,2 through 9 may be used; A-M-1 (normally: Command-Option-1) will select the first, leftmost tab in the window.[1]

Note that Tab functionality may be impaired in modes that use the Emacs Header Line. This includes the Buffer List (C-x C-b) and *psvn*.

A related option lets Aquamacs open new buffers in their own, extra frames. The option may be combined with Tabs, when you want to create tabs manually (Command-T), but visit new files in separate windows. Normally, we would not recommend to combine the two options.

Note that when tabs are switched on, or when buffers are shown in new frames, then closing a window has a different implication: it will remove the buffer from memory (in Emacs terminology, it will *kill* the buffer). In case there are unsaved modifications, you will be asked whether you'd like to save them. If neither tabs nor "Show Buffers in New Frames" is on, then closing a window using Command-W, or closing a whole frame with the red closer button has no such implication. The buffer remains in memory and is available from the Window menu. It is not automatically saved.

3.2 Configuring the Toolbars

The easiest way to set up icons in the toolbars is to right click on the toolbar that you'd like to customize and select *Customize Toolbar...*. This will show a panel with all available icons. Drag and drop those into the toolbar.

Note that if a mode provides its own set of icons or additional icons, Aquamacs will store a separate configuration for that toolbar. This way, you can set up toolbars for different tasks. (Note that if modes, perhaps after an upgrade, add or remove icons, Aquamacs will adopt a new customization set for the toolbar.)

Click on the button in the top right corner of the frame to remove the toolbar completely. Use the menu items *Options / Appearance / Adopt Face and Frame Parameters as Default* and *Options / Save Options* to make this change stick.

[1] Use the function 'tabbar-define-access-keys' to configure the keys in your `Preferences.el` file.

3.3 File

The File menu includes basic operations for opening, closing, and printing files. Opening and saving files uses standard Mac keyboard shortcuts (Command-O, Command-S), and uses standard Aqua dialog boxes.

You can create a new buffer: Aquamacs allows you to choose from a list of recently and commonly used editing modes. A similar function lets you change the current editing mode.

You can also open a directory. When called by mouse from the menubar, this brings up a standard Aqua dialog box. When called up with a keyboard command (C-x d), it brings up a directory name in the "minibuffer" (small space for commands at the bottom of the main window, or frame).

Printing is fully supported, although no "Page Setup" option is available at this point. You will see a print screen (in the system's Preview.app) from where you can select a printer and the usual options. This may take a few seconds to appear with very large buffers. (Printing of embedded images such in LaTeX Preview mode is currently not supported.)

You may export your buffer to PDF or to HTML format. This operation preserves all the formatting, including any word-wrapping - so you may want to choose a font and resize the frame so that the lines are wrapped in the intended way (if Soft Wrapping is used).

The paper size for printing and PDF export can be configured by customizing the variable 'ps-paper-type' (use Options / Customize Aquamacs / Specific Option).

3.4 Edit

The Edit menu is the heart of Aquamacs' textual wizardry. Aquamacs supports all customary editing functions, such as cut, copy, paste, and simple search and replace. In Aquamacs these basic functions are supported by standard Aqua keyboard shortcuts. There is a great deal more functionality, however, than the average text editor. For instance, Aquamacs Emacs allows you to go to a specific line number in the file you are editing, to the top or bottom of the buffer, and so on. It also supports searches with *regular expressions,* which are sophisticated text patterns that go beyond simply matching a specific set of characters (or "string"). Aquamacs Emacs also stores more than twenty of the most recently-copied items on the clipboard, and these are accessible from the menu in case you need to paste these items again. The Edit menu also supports "bookmarks," a feature that allows you to save your place in a specific file.

Aquamacs lets you spell-check your texts, either while typing ('flyspell-mode') or for the whole buffer with an interactive interface ('ispell').

3.5 Options

The Options menu is where you can easily customize your settings. The options that you can configure include syntax coloring, matching of parentheses (useful for text markup that depends on open and close brackets), how to display buffers and frames, fonts, and other settings. For more information on fonts, please see Section 3.5.4.

If you want to delve deeply into customizing Aquamacs, select "Customize Emacs:Top-Level Customization Group" in the Options menu. A new frame titled "Customize Group: Emacs" will open. Scroll down and find the "Aquamacs Group" listing, and push the “Go to Group” button. This will open a new frame with all of the configuration options for Aquamacs —power users can customize Aquamacs this way, or via the Emacs Lisp programming language, elisp—and each option is documented there. Table 4 displays a complete list of the options.

3.5.1 Line wrapping

Text is usually shown and read line by line. Several lines commonly form a paragraph. However, text is not necessarily stored this way in a file: sometimes, a *line* stored in a file is much wider than what would fit on a computer screen, and is really a paragraph. This makes sense: authors of documents do not know how wide the reader's computer screen is, or how wide the window is going to be that is used to read and edit the text. Aquamacs supports this mode of editing with *word wrapping*: lines are wrapped at word boundaries. This way, a buffer line is displayed as a paragraph (several visual lines) on the screen. In Aquamacs, this mode of operation is called **word wrap** and is available from the Options menu. Internally, word wrapping enables *Visual Line Mode*.[2]

What's the alternative? **Hard line breaking** means that one line shown on the screen corresponds to one line in the buffer and file. This means that even if a window is resized, the lines still stay the same, and word wrap will occur in the same place on every machine that displays the file. Line Breaking, called *Auto Fill* in classical Emacs, changes the actual buffer and file contents, unlike all the other options discussed here.

Long lines can also be **truncated** (cut off) or be continued (**wrap**) on the next line.

Line wrapping can be set to Truncate, Wrap, Word Wrap and Line Breaking (auto fill) under *Options / Line Wrapping* for the current buffer. A *Set as Default* will set it as default for all buffers that have no explicit line wrap setting.

Note that Word Wrapping (‘visual-line-mode’) affects the function of the cursor movement keys (arrows, and C-a, C-e, C-p and C-n). In Aquamacs, this can be easily customized. The default is as such:

- In Word Wrapping (visual-line-mode), C-aenp and arrow keys move according to visual lines.

- Without Word Wrapping (visual-line-mode off), arrow keys move visually, but C-aenp move non-visually according to buffer lines as in Emacs 22 and Aquamacs 1.x.

You may configure this behavior via the ‘line-move-visual’ customization variable. Set it to *t* for unconditional visual movement of all keys, and *nil* for unconditional logical (buffer) movement of all keys.

Aquamacs allows you to automatically set the right type of line wrapping for text files using the “Detect Line Wrap in Text Files” function. (You can control the default, e.g., for new files, by setting ‘auto-word-wrap-default-function’ to ‘set-word-wrap’ using Options / Customize Aquamacs / Specific Option.)

A useful function to convert hard wrapped text into soft wrapped text can be found in the Edit menu: *Remove Hard Line Breaks* will turn (non-indented!) paragraphs into soft-wrapped lines (turn on soft wrapping to see).

[2] Earlier Emacs versions provided ‘longlines-mode’, which is similar, but is unreliable and deprecated.

Similarly, to manually reformat text that is laid out line by line, using hard line wraps, use *Wrap and Re-Format*. Emacs terminology calls this process *filling*.

3.5.2 Using the Options key and more...

The keyboard has always been the essential user interface to Emacs - it's what makes Emacs so efficient as an editor for daily tasks. There are keyboard bindings for pretty much every task. Many bindings involve pressing the "Meta" modifier key - it's a key just like Control or Shift, which all go together with another (normal) key. "Meta" has only really existed on Unix keyboards long time ago – nowadays, computers have other keys instead. Therefore, you will need to press another key on your Macintosh keyboard. *By default, this is the Option key*, but you can use the ESC key instead (in this case, you may press ESC first, then the rest of the keys).

Inputting characters with the Option key on non-English keyboards. In most non-English keyboard layouts, the Option key also serves to input characters such as { or \ or @. Using Option as Meta would inhibit you from inputting those characters. You have three options to get around this. *Either*, you deselect "Option Key for Meta" in the Options menu (under "Option Key"), in which case you will have to use ESC for Meta, *or* you toggle back and forth between the modes using Command-;, *or* you use Option for Meta, but enable one of the emulation modes provided in the same menu under "Option Key". For those keyboard layouts supported, **this option is the most convenient**. The emulation will allow you to input all common characters with the Option key, while using Option for Meta in general.

Alternatively, Aquamacs allows you to use another modifier key – such as the Function key on laptops. The customization group "Aquamacs" contains appropriate settings, reachable via the menu items in Options / Customize Emacs.

3.5.3 Languages of the World - Dealing with Different Character Sets

Computer keyboards were designed to input text in languages with a small character set. Aquamacs Emacs lets you use your keyboard with a variety of *input methods* in order to input text in languages with many characters, among them Asian languages.

By default, Aquamacs uses the standard Mac user interface to input characters, called "System Input Method".

The Multilingual Environment, present in GNU Emacs, provides a number of predefined input methods. Aquamacs Emacs extends this: On Mac OS X, because of the particular ways of user interface and unicode handling, these standard Emacs language environments are not enough to read and write some international languages. One needs to set additional parameters, especially, the language specific coding systems.

You can find these new language environments (Korean, Japanese, Chinese-GB and Chinese-BIG5) – among other ones – under "Aquamacs Multilingual Environment".

3.5.4 The Right Look: Colors and Fonts for Frames and Modes

Aquamacs allows you to alter specific features of the current frame via functions in the Options menu. "Font for *-mode..." lets you choose from the installed system fonts.

You can also choose foreground and background colors. *These settings only apply to buffers in the current major mode*, provided *Autofaces* is turned on (as is by default). Without Autofaces, the settings apply to all frames and set the frame default.

With *Autofaces*, Aquamacs Emacs lets you pick a style specific to the current mode. For example, you can use different font settings when you are editing C or LaTeX, then when you are editing a text file. Note that all settings not sepcified for the mode are taken over from the default. Specifically, color and font choices are prioritized in this order: *1.* Mode-specific face, *2.* Autoface Default Face, *3.* Frame default.

You can use "Adopt Frame Parameters as Frame Default" to accept the appearance of the current frame for future frames; this includes the visibility of the tool bar, fringes, and more (but not position and size).

If you're more of an Emacs expert and you'd like to configure things manually by means of elisp or the customization buffer, configure the following faces. With Autofaces, you can customize the faces directly (via M-x customize-face). The face name for a major mode is always named after the major mode, e.g. "latex-mode-default" is the autoface face for 'latex-mode'. The Autoface default face is named 'autoface-default'. The default for all frames is the 'default' face.

You can also customize the variables for the frame default: "default-frame-alist" and "initial-frame-alist", as documented in the Emacs manual. You can, alternatively, use "Adopt Frame Parameters as Frame Default" to accept the appearance of the current frame for future frames; this includes the visibility of the tool bar.

3.6 Window

The Window menu allows you to navigate through the windows and frames that you may have open. Note that a "buffer" is not synonymous with a window or frame, in that you can split a frame and have more than one buffer contained within. (Multiple windows/frames is a feature of Aquamacs; standard Emacs does not support this.) In addition to standard frames that display open files, there are a few other important buffer types. One is the "scratch" buffer, which is simply a buffer to type notes into; this can also be the starting point for a file to save, and a buffer to type configuration commands for Aquamacs (an advanced feature). Another is the "Messages" buffer, which displays a log of output from Emacs commands and operations. Finally, there is the "info" buffer, in which Aquamacs Emacs displays built-in user help, tutorials and other documentation in Emacs' "info" format.

From the Window menu, you can also open a new "frame,", or split the open window into two separate ones. There are keyboard shortcuts for all of these commands.

3.7 Tools

The Tools menu provides access to a variety of functions, including integration with version control systems, running shell commands, searching external files for text, and compiling and debugging software code. The Tools menu also provides access to newsgroups and e-mail.[3]

[3] The functionality provided in the Tools menu is, to say the least, diverse, and is part of the attraction of Emacs for a large number of users—particularly advanced users. It is not necessary to use Aquamacs as an e-mail client to appreciate its considerable power and utility, however.

3.8 Help

The Help menu contains a wealth of information about Aquamacs Emacs and GNU Emacs. Except for the help provided in this document, Aquamacs' Help menu also provides documentation of the Mac-specific customizations with the present User Manual. The general Emacs user help is comprehensive and detailed to the point of possibly overwhelming the inexperienced user. The beginner should definitely start with the Emacs tutorial contained in the Help menu. While geared toward the traditional Emacs interface instead of the OS X Aquamacs version, the tutorial is a good introduction to Emacs' unique capabilities. And, as you gain more experience, you will appreciate the depth of the Emacs documentation.

4 Aquamacs for Emacs Veterans

While experienced users of Emacs on other platforms can continue to use all the key combinations to which they are accustomed, we recommend that they use the Aquamacs-specific conventions to get the most benefit from the applications. Many of the standard Emacs behaviors and interface conventions have been modified in Aquamacs in the interest of providing a more Aqua-native experience. In this section, we discuss some of the ways that Emacs conventions are mapped to Aqua conventions, and outline some advanced ways that users can modify Aquamacs to their specific preferences.

4.1 Bringing some Emacs 24 behavior back

Expert users who find themselves confused by some changes to the Emacs behavior that they are used to, either due to the upgrade to Emacs 24, or to Aquamacs, may undo configuration changes quickly by using the Emacs Lisp code shown at `http://www.emacswiki.org/emacs/AquamacsEmacsCompatibilitySettings` in their Preferences.el file.

4.2 Key Bindings

Emacs has a well-defined set of keyboard shortcuts, to which Aquamacs adds its own to accomodate OS X conventions. See Table 2 and Table 3 for details. Note that the descriptions of keys in Aquamacs conform to the Mac standards.

Almost all key bindings are provided by a minor mode, *osx-key-mode*. This mode can be turned off. Bindings may also be removed or overriden in its key map, *osx-key-mode-map* (note: Emacs minor mode key bindings have priority compared to global bindings).

Modifier keys may be adjusted using the 'ns-...-modifier' customization variables, such as 'ns-alternate-modifier', or 'ns-right-command-modifier'.

4.3 Customizing Aquamacs

One of the distinguishing features of Emacs is the degree to which it can be customized by the end user. Emacs includes its own internal scripting language, elisp, which allows the user to customize such things as keyboard shortcuts, window settings, fonts,

Shortcut	**Elisp Command**	**Function**
Command-N	new-frame-with -new-scratch	Create new buffer
Command-O	find-file other-frame	Open a file
Command-W	close-window	Close selected window deleting buffer
Command-Shift-S	write-file	Save as
Command-A	mark-whole-buffer	Select all text
Command-V	cua-paste (yank)	Paste text
Command-C	clipboard-kill-ring-save	Copy text
Command-Option-C	aquamacs-clipboard-kill-ring-save-secondary	Copy text from secondary selection
Command-X	clipboard-kill-region	Cut text
Command-Option-X	aquamacs-clipboard-kill-secondary	Cut text from secondary selection
Command-S	save-buffer	Save file
Command-L	goto-line	Go to specified line
Command-F	isearch-forward	Search
Command-G	isearch-repeat-forward	Repeat search
Command-E	aquamacs-use-selection-for-find	Use selected text for next search
Command-;	spellcheck-now	Jump to next spelling error
Command-M	iconify-or-deiconify-frame	Minimize window to the Dock
Command-.	keyboard-quit)	Keyboard quit
Command-,	customize	Show Customization Buffer
Control-;	toggle-mac-option-modifier	Change Option key function
Command- / / ’	(un)comment-region-or-line	Comment out or in the current line or region if marked
Command-Backspace	kill-whole-visual-line	Deletes the current line
Command-Delete	kill-visual-line	Deletes the remainder of the current line
Command-Q	aquamacs-save-buffers-kill-emacs	Save file, exit program
Command-Z	undo	Undo
Command-Shift-Z	redo	Redo
Mouse-drag		select text
Option-Mouse-drag		select text as secondary selection

Table 2: Mac-specific keyboard shortcuts implemented in Aquamacs. This assumes that the “Option Key is Meta” (Options menu). Commands named here are usually the traditional Emacs commands rather than direct bindings.

Emacs Command Key	Aquamacs / Mac Command Key
C-*	Control-*
A-*	Command-*
M-*	Option-*[4] (or Esc)

Table 3: Aqua-specific command keys implemented in Aquamacs.

and more. The Aquamacs customizations themselves are implemented in elisp.[5]

We recommend that user customizations be placed in `Preferences.el` in the Library folder (see Section 6.2). The usual `~/.emacs` file is still loaded for compatibility.

Note that Emacs Lisp settings in a Preference.el file take priority over any other defined customizations such as those chosen in the Options menu, in customization buffers (Aquamacs / Preferences and elsewhere), or `~/.emacs`.

Below are some specific customization options and groups that may be of interest:

- **Frame.** Aquamacs opens new files (and other buffers) in new frames. That is usually more convenient and allows you to use the graphical user interface of today's computers, which did not exist when Emacs was conceived almost three decades ago. If you do not like this behavior, perhaps because you are used to traditional Emacs, just deselect "Display Buffers in Separate Frames" in the Options menu and save your choice with "Save Options." (The associated global minor mode is called "one-buffer-one-frame-mode".)

- **Autofaces.** These styles provide a convenient way to tie default faces to specific major modes so that every frame showing a buffer in a particular major mode is styled that way. Such faces can determine background and foreground color as well as font settings. Such settings always override any global choices made with, for instance, 'default-frame-alist' or the 'default' face. The autofaces may be customized ('customize-face') directly or set through the GUI. Note that these faces apply *per buffer*.

- **ns-alternate-modifier.** This is the modifier to use for the Mac alt/option key. The value can be alt, hyper, or super for the respective modifier. If the value is nil then the key will act as the normal Mac option modifier, and the option key can be used to compose characters depending on the chosen Mac keyboard setting.

- **Additional customization.** Of course, Aquamacs Emacs offers you almost all the customization possibilities that Emacs has. Under "Customize Emacs," you will find a sub-menu that allows you to browse the vast space of customization settings. Beware: some of them are complex and not easy to understand. If you would like to tinker with some general Aquamacs-specific behavior, you can customize the group "Aquamacs."

- **The Scratch Buffer**

[5] The Aquamacs customizations are stored in elisp files in the application bundle. It is possible to modify these files directly, but we discourage this practice and provide no support for it.

Variable
Aquamacs Set Creator Codes After Writing Files
Aquamacs Quick Yes Or No Prompt
Aquamacs Ring Bell On Error
Aquamacs Known Major Mode
Aquamacs Default Styles
Aquamacs Buffer Specific Frame Styles
Aquamacs Auto Frame Parameters Flag
One Buffer One Frame Mode
Tabbar Mode
Smart Frame Positioning Mode
Mac Option Modifier
Mac Control Modifier
Mac Command Modifier
Mac Function Modifier
Aquamacs Known Buffer Modes
OS X Key Mode
OS X Key Mode Mouse-3 Behavior
Visual Scroll Margin
Emulate Mac ... Keyboard Mode

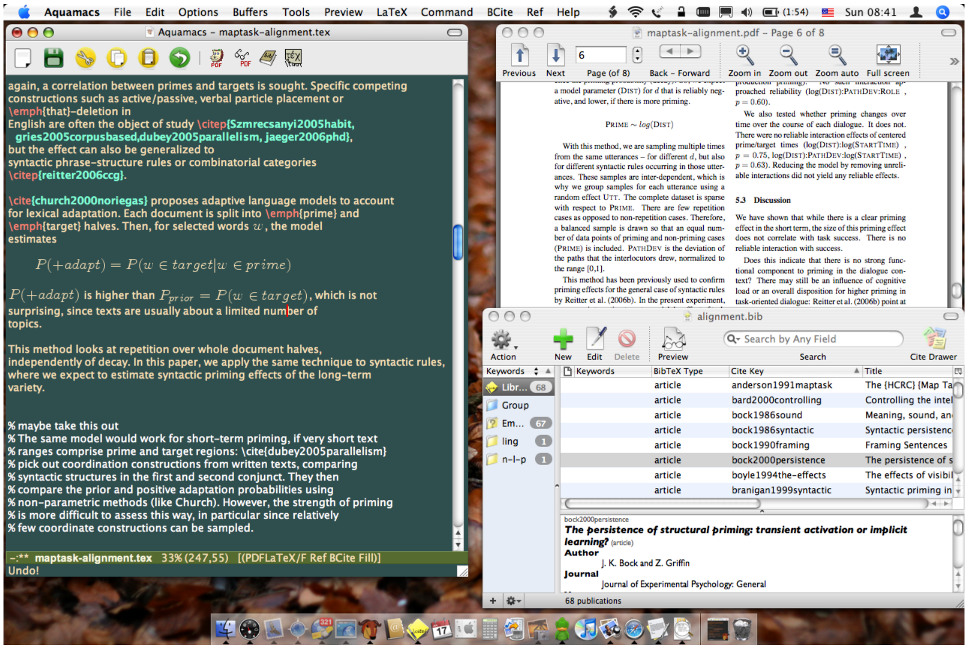
Table 4: Selection of Aquamacs-specific variables that can be customized in the Aquamacs customization group.

  As mentioned above, the Aquamacs scratch buffer is simply a buffer into which notes can be typed. For this reason, the scratch buffer is opened in text mode, as opposed to lisp interaction mode (the default mode for the plain GNU Emacs). Customize 'initial-major-mode' (and restart) to change this. In Aquamacs, the scratch buffer is automatically retained between sessions. It can also be saved using the normal save commands; you are free to save its contents to another file.

- **Want some GNU Emacs 24 behavior back?** In most cases that represents no problem. Aquamacs provides a customization group called "Aquamacs-is-more-than-Emacs" which contains all the settings whose default values differ from the ones in GNU Emacs (24) or the appropriate package. We recommend that you tinker with these one at a time and check their (often wider-than-you-think-reaching, sometimes detrimental) effects.

  Also, see Table 4 for a partial list of such settings.

The range of possible customizations—including restoring some of Emacs' traditional interface conventions—is beyond the scope of this help document. However, we provide a wiki for users to share their modifications. See http://www.emacswiki.org/emacs/AquamacsEmacs for more details.



```
again, a correlation between primes and targets is sought. Specific competing
constructions such as active/passive, verbal particle placement or
\emph{that}-deletion in
English are often the object of study \citep{Szmrecsanyi2005habit,
  gries2005corpusbased,dubey2005parallelism, jaeger2006phd},
but the effect can also be generalized to
syntactic phrase-structure rules or combinatorial categories
\citep{reitter2006ccg}.

\cite{church2000noriegas} proposes adaptive language models to account
for lexical adaptation. Each document is split into \emph{prime} and
\emph{target} halves. Then, for selected words w, the model
estimates

      P(+adapt) = P(w ∈ target|w ∈ prime)

P(+adapt) is higher than P_prior = P(w ∈ target), which is not
surprising, since texts are usually about a limited number of
topics.

This method looks at repetition over whole document halves,
independently of decay. In this paper, we apply the same technique to syntactic rules,
where we expect to estimate syntactic priming effects of the long-term
variety.


% maybe take this out
% The same model would work for short-term priming, if very short text
% ranges comprise prime and target regions: \cite{dubey2005parallelism}
% pick out coordination constructions from written texts, comparing
% syntactic structures in the first and second conjunct. They then
% compare the prior and positive adaptation probabilities using
% non-parametric methods (like Church). However, the strength of priming
% is more difficult to assess this way, in particular since relatively
% few coordinate constructions can be sampled.

-:** maptask-alignment.tex   33% (247,55)   [(PDFLaTeX/F Ref BCite Fill)]
Undo!
```

a model parameter (DIST) for d that is reliably negative, and lower, if there is more priming.

$$\text{PRIME} \sim log(\text{DIST})$$

With this method, we are sampling multiple times from the same utterances – for different d, but also for different syntactic rules occurring in those utterances. These samples are inter-dependent, which is why we group samples for each utterance using a random effect UTT. The complete dataset is sparse with respect to PRIME. There are few repetition cases as opposed to non-repetition cases. Therefore, a balanced sample is drawn so that an equal number of data points of priming and non-priming cases (PRIME) is included. PATHDEV is the deviation of the paths that the interlocutors drew, normalized to the range [0,1].

This method has been previously used to confirm priming effects for the general case of syntactic rules by Reitter et al. (2006b). In the present experiment,

proached reliability (log(DIST):PATHDEV:ROLE , $p = 0.60$).

We also tested whether priming changes over time over the course of each dialogue. It does not. There were no reliable interaction effects of centered prime/target times (log(DIST):log(STARTTIME) , $p = 0.75$, log(DIST):PATHDEV:log(STARTTIME) , $p = 0.63$). Reducing the model by removing unreliable interactions did not yield any reliable effects.

5.3 Discussion

We have shown that while there is a clear priming effect in the short term, the size of this priming effect does not correlate with task success. There is no reliable interaction with success.

Does this indicate that there is no strong functional component to priming in the dialogue context? There may still be an influence of cognitive load or an overall disposition for higher priming in task-oriented dialogue: Reitter et al. (2006b) point at

Keywords	BibTeX Type	Cite Key	Title
	article	anderson1991maptask	The {HCRC} {Map Ta
	article	bard2000controlling	Controlling the intel
	article	bock1986sound	Meaning, sound, an
	article	bock1986syntactic	Syntactic persistenc
	article	bock1990framing	Framing Sentences
	article	bock2000persistence	The persistence of s
	article	boyle1994the-effects	The effects of visibil
	article	branigan1999syntactic	Syntactic priming in

bock2000persistence
The persistence of structural priming: transient activation or implicit learning? (article)
Author
J. K. Bock and Z. Griffin
Journal
Journal of Experimental Psychology: General

Figure 2: Aquamacs offers extensive support for LATEX documents and can be combined with viewer applications such as Skim and bibliographic databases such as BibDesk.

4.4 LATEX Support

One special feature of Aquamacs is its extensive support for the editing of LATEXdocuments, especially Emacs' AUCTEX mode.

AUCTeX comes with its own manual, accessible via the LaTeX menu (when in latex mode after loading a .tex file), "Read the AUCTeX manual". If you have a question, please consult the manual. If that doesn't work, or you suspect you've found a bug in AUCTeX, please turn to the AUCTeX mailing lists, accessible here: http://www.gnu.org/software/auctex/mailing-lists.html.

To access the enhanced LATEX functionality that Aquamacs offers, you will need to install a LaTeX system. We recommend to install the *MacTeX distribution*, available at http://www.tug.org/mactex/. Alternatively, you can install a smaller distribution provided by Gerben Wierda (http://www.rna.nl/tex.html. Both install packages are complete and user-friendly. (While you can obtain TEX from other sources, the binary distributions are particularly well supported in Aquamacs.)

Note that AUCTeX, as it comes with Aquamacs, is configured to compile documents directly to PDF, a standard on the Mac. If you prefer to produce DVI and Postscript documents with LaTeX, switch off PDF Mode in Command / TeXing Options.

4.4.1 Skim: A LATEX Previewer

When editing LATEXfiles, you will often need to view the compiled results. Apple's Preview.app can view PDF files. But for regular use you may prefer a program like *Skim* which has features specifically supporting LATEXediting. Skim is available here: http://skim-app.sourceforge.net/ . Skim can automatically reload PDFs every time they are recompiled. It also includes a configuration preset to allow it to jump back and forth between Aquamacs and Skim, showing the documents at the corrsponding locations. An up-to-date LaTeX distribution such as MacTeX 2009 is required to use the “SyncTeX” feature.

Clicking on a position in the PDF file in Skim while holding the Command and Shift keys causes Aquamacs to move your cursor to the corresponding point in the underlying source file, opening it if necessary. To enable Skim with SyncTeX, enable Aquamacs in Skim’s Preferences dialog. Aquamacs does not need to be configured further – it should recognize Skim when it is running (you need to start it yourself).

If you want to be able to start Skim from inside Aquamacs, do the following.

Having opened a Latex buffer in Aquamacs, go “Menu>Latex>Customize Auctex” and then click on ”Extend this Menu.” Then, once more, go “Menu>Latex>Customize Auctex.” In the list of items you see now, drag the mouse to ”Tex Command,” in the list you see then, drag it to ”Tex View”, and in the list which finally opens, click on ”Tex View Program Selection.” You now have a customization buffer opened. There, you see a line which contains the words ”output-pdf.” In the line below that line, you see the word ”Viewer.” To the right of it, there is a button ”Value Menu.” Click on it, and choose ”Skim” in the list that pops up.

Finally, before you close the customization buffer, do not forget to click on the button ”State” and, in the list then popping up, to click on ”Save for Further Sessions”. That’s all.

Now you can use the view icon in the toolbar, or the key combination C-c C-v, or “Menu>Command>View” to open Skim with your pdf-output file.

Furthermore, things are configured so that these three ways to call Skim also yield syncing from source to pdf-output. Syncing from source to pdf-output can also be obtained by clicking on a position in your source-file while holding the Command and Shift keys.

(If you had not rejected Apple’s default in the installation of Leopard or Snow Leopard to put X11 on your system, you may also consider to choose xdvi for ”output-dvi.”)

4.5 Command Line Tools

Aquamacs can install command line tools. Run M-x aquamacs-install-command-line-tool, or choose *Install Command Line Tools* from the Tools menu. The following commands will be made available from a terminal running the system’s default login shell:

- `aquamacs`: This tool will start up a new Aquamacs or run an existing one. (If multiple Aquamacs.app application bundles are present in the system, unexpected behavior may occur. This is a OSX limitation.) You may give arguments to the tool and use it as `EDITOR` setting.

- `emacsclient`: This is a tool standard to every GNU Emacs distribution. It will open a file or execute arbitrary commands in a running Emacs/Aquamacs instance. (Aquamacs automatically starts the associated server so that you do not need to call M-x server-start.)

- `emacs`: This tool is installed only if direct install is possible by Aquamacs into /usr/local/bin, on Mac OSX 10.11 or later. Emacs will start up vanilla mode

Aquamacs directly in the text terminal, with no Aquamacs customizations, emulating GNU Emacs. This allows you to run a late-version Emacs with your system, without the need for installing another package.

4.6 Vanilla Emacs Mode

Aquamacs may be run in *vanilla Emacs mode*. This will disable most Aquamacs customizations and prevent loading of included packages. Aquamacs will be most similar to GNU Emacs. The mode is intended primarily for terminal use.

To enable the mode, start the Aquamacs process from a terminal, with the `--no-site-file` parameter.

The mode is also enabled by the command line tool `emacs`.

5 Requirements

Aquamacs requires Mac OS X 10.6 or later. The downloadable Aquamacs runs on all modern Macs with 64-bit capable Intel processors. Versions for older Macs (32-bit Intel, PPC) may be compiled by the end user.

6 Extending Aquamacs

Aquamacs supports Plugins that can be packaged to provide point-and-click installers (via Apple's PackageMaker). The interface is simple. Adding standard Emacs packages is easy as well.

6.1 Plug-in Interface

All files named *site-start.el* anywhere in the load-path are loaded at startup, before the user's init files are loaded. For example, such a file may be placed in the user's Library folder, that is in
`~/Library/Application Support/Aquamacs Emacs/myPlugin`.

The load-path is automatically extended to cover all subdirectories in standard paths (see Section 6.2).

If a plug-in needs to execute code earlier in the initialization phase, that is, before any Aquamacs- specific code is executed (but after Emacs initilization code has been executed), this code can be provided in a file *site-prestart.el* anywhere in the load-path.

Info files (M-x info) can be provided in a directory called `info` inside the plugin directory. Aquamacs will automatically add this path to 'Info-default-directory-list'. Plugins should provide a `dir` file inside the `info` directory containing an overview (see the Emacs documentation for 'info').

Note that plug-ins should not make any assumptions about whether a frame exists or is visible.

6.2 Load Path

Aquamacs is able to find configuration files and Emacs packages from any of the following standard locations (load-path) in Library folders.

- `/Library/Application Support/Aquamacs Emacs/`
- `/Library/Preferences/Aquamacs Emacs/`

The following paths are available in the load-path for individual preference settings. *They should not be used if you distribute a point-and-click installer package.*

- `~/Library/Application Support/Aquamacs Emacs/`
- `~/Library/Preferences/Aquamacs Emacs/`

All subsequent directories are added recursively, so Aquamacs will find any path that starts below. Type `C-h v load-path` for more information.[6]

6.3 Disabling a plug-in or a load-path

If an (emtpy) file named *.nosearch* exists in a specific path, Aquamacs does not load any *site-start* and *site-prestart* source files in this or descending paths and does not add that or subsequent paths to the load-path.

6.4 Startup files

In short, most settings are stored in these files:

- `~/.emacs`
- `~/Library/Preferences/Aquamacs Emacs/Preferences.el`
- `~/Library/Preferences/Aquamacs Emacs/customizations.el`
  [Automatically written when you "Save Options"]

There are further source files that will be loaded. So, the complete answer is as follows.

At startup, Aquamacs will load this file:

- `~/.emacs`

At startup, by default it will load the following files at startup:

- `~/Library/Preferences/Aquamacs Emacs/customizations.el`
- `/Library/Application Support/Emacs/Preferences.el`
  [deprecated: don't use!]

[6] The following paths are loaded for backward compatibility reasons:

- `/Library/Application Support/Emacs/`
- `/Library/Preferences/Emacs/`
- `~/Library/Preferences/Emacs/`
- `~/Library/Preferences/Emacs/`

. Do not use these paths.

- /Library/Application Support/Aquamacs Emacs/Preferences.el
- ~/Library/Application Support/Emacs/Preferences.el

  [deprecated: don't use!]
- ~/Library/Application Support/Aquamacs Emacs/Preferences.el

The location of the first customizations file is given by the 'custom-file' variable.

Aquamacs will also automatically load all files names `site-start.el` in the load path list, which includes the above paths.

Note that Emacs (including Aquamacs) will preferentially load a `foo.elc` file if present, in lieu of a `foo.el` file, even if the `.el` file is newer.

To trouble-shoot your own customizations, first start Aquamacs ”without customizations”, which you can do via a function found in the menu ”Help/Diagnose and Report Bug”.

6.5 Changing Aquamacs source files

If you'd like to tinker with the Emacs lisp source files that come with Aquamacs, you can do so quite easily. When defining a new function or, more generally, to evaluate an expression, C-x C-e is your friend.

However, you may notice that some of your changes aren't active after you've saved your work (into the Aquamacs application bundle) and restart. That is because some of the code is precompiled into the Aquamacs binary and not loaded from the source (or .elc / .el.gz) files. To alleviate that problem, create a file
`~/Library/Application Support/Aquamacs Emacs/site-prestart.el`
with the following content:

```
(setq aquamacs-reload-preloaded-files t)
```

This will cause to re-read the otherwise preloaded files from disk.

7 Getting Help

There are many options for getting help with Aquamacs.

From within Aquamacs, you can access user documentation from the menu or from specific key combinations. This manual is available from the Help menu, as is a very complete Emacs documentation, the *Emacs Manual*. To extend Emacs using the built-in programming language Emacs Lisp, refer to the *Emacs Lisp Reference*, which is also available from the Help menu. (The latter can be used via the Mac's convenient Help Books, or via Emacs' powerful Info interface.)

Emacs is self-documenting. The Help menu (under *Describe*) contains documentation for every function available to the user. You can reach these functions easily: `C-h k` (plus key/menu entry) brings up help for any key or menu binding; `C-h f` (plus function name) gives help for an elisp function; autocompletion support is available with (tab); and `C-h a` brings up apropos, a search function.

For help from other Aquamacs/Emacs users, the best place to begin is the OS X Emacs mailing list. The searchable list archives are located at http://dir.gmane.org/gmane.emacs.macintosh.osx. For more information on subscribing to the list, see http://email.esm.psu.edu/mailman/listinfo/macosx-emacs. Another option for general Emacs help is the gnu.emacs.help newsgroup.

Remember: almost every question has been asked before. Google will give you an instant answer, provided you know the right keywords.

Apart from answering questions at the OS X Emacs mailing list, you can also file bug reports on Aquamacs. Use the "Send Bug Report" function in the Help / Diagnose and Report Bug menu. The same menu has a function to restart Aquamacs without any of your customizations – please use this function to ensure that it is really a bug in Aquamacs that you are seeing. (Your settings may be flawless, but even so we'd like to know which ones will cause the bug to show up!)

In addition to requesting help, you can also offer it in these mailing lists. *Please note that in case we have helped you with a configuration problem on the mailing list, we may ask you to write a little note in the Aquamacs Wiki so others can benefit, too.* Remember: Aquamacs is an Open Source community project!

8 Aquamacs Lives From Your Donations

Aquamacs Emacs is a project that depends on your support. The project has no other source of substantial income: the application is free, the downloads are free.

Free software is developed because programmers love their work. Donations will never remunerate them financially as a job in the software industry would. But every donation is a recognition of the programmer's hard work, and the bigger ones actually help paying some bills.

If you would like to say "thank you", or if you would like to see the project developed further, *please make a donation.*

http://aquamacs.org/donate

9 Contribute to Aquamacs

If you'd like to join us in working on Aquamacs, download the source code from Git and take a look. Do ask if you have questions! When you have a patch or a less specific suggestion for improvements, send it to the development mailing list.

We're a very open team. Everyone is allowed to suggest modifications to every piece of code. There is no "ownership" of code, so please join us to make Aquamacs even better!

10 Acknowledgments

Who's behind Aquamacs?

It's a free software project, and naturally, Aquamacs could not exist without the help of developers, designers, writers and users. The following people have contributed code, bug reports and ideas to the Aquamacs project at one point of the other.[7]

[7] We apologize for not being able to list everyone. If you feel that someone should be mentioned, please e-mail us.

Part of the development team of Aquamacs 3.x are: David Reitter (co-founder, lead developer). Many contributions in bug reports and fixes came from Konrad Podczeck, Nathaniel Cunningham, and other regular users.

Kevin Walzer (co-founder) wrote the first manual. Aquamacs contains icon artwork by Adrian Chromenko, Jessica Walker (Aquamacs 2.0/3.0 application icon) and David Reitter, Jasper Hauser, and the Nuovext Project. The Aquamacs website was designed by Ted Roden.

We would also like to acknowledge the contributions of the many authors whose source code and hints on public forums have already been integrated into the build. They are too numerous to mention here.

Aquamacs is based on GNU Emacs, which is the work of Richard Stallman and many other developers, in particular Andrew Choi (original GNU Emacs-to-MacOS and OSX ports), Steven Tamm, Mitsuharu Yamamoto, Adrian Robert. Aquamacs also could not exist without the latest code contribtions to GNU Emacs (now lead by Stefan Monnier) and to its Mac/OSX/Nextstep ports (Jan djärv, Yamamoto Mitsuharu, and others). We also would like to acknowledge the work of the many authors of the major modes included.

Last but not least, Aquamacs exists because of the dedication of its users, who donate money, report bugs and help us fix them.

11 Nightly Development Builds

Ready to install, the Aquamacs project offers nightly built binary packages that contain the latest in Aquamacs development. These builds are not considered "stable", but experimental.

See http://aquamacs.org/nightlies.shtml for downloads.

12 Obtaining the Source Code

You may download the source code for each release of Aquamacs, or download the source of the current development version from http://aquamacs.org/development.shtml. You need Git (http://git-scm.com) in order to download it.

The author offers to provide you with the source code for a time period of at least three years from the date of release. Source code should be obtained from the aforementioned Git repository or, if this is impossible, by written request from the author.

13 Licenses

Aquamacs: the Emacs distribution, (c) 2005–2018 by Free Software Foundation, Inc., and by David Reitter. Aquamacs is licensed under the terms of the GNU General Public License, either Version 3, or (at your option) any later version.

For details of this license, type C-h C-c in Aquamacs (Help: Copying Conditions) or visit http://www.fsf.org/licensing/licenses/gpl.html. The Aquamacs documentation is licensed under the terms of the GNU Free Documentation License. For details of this license, please see http://www.gnu.org/copyleft/fdl.html#SEC1.

The license does not apply to the toolbar icons contained in the Aquamacs .app bundle. They may be used and redistributed within Aquamacs, but not separately or in another application. Some of the icons may be redistributed, however - see the COPYING file in the etc/images directory in the Aquamacs .app bundle or in the source code.

The Aquamacs binary distribution includes compiled versions of the following libraries, which are distributed under the specified licenses:

- libffi: https://sourceware.org/libffi/, distributed under the following license:

  > libffi - Copyright (c) 1996-2014 Anthony Green, Red Hat, Inc and others. See source files for details.
  >
  > Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:
  >
  > The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.
  >
  > THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

- GMP: https://gmplib.org, distributed under the GNU LGPL v3 and GNU GPL v2.

- GnuTLS: https://gnutls.org/, distributed under the GNU LGPLv2.1+.

- Nettle (a low-level cryptographic library): https://www.lysator.liu.se/~nisse/nettle/, distributed under the GNU LGPLv3. This is also the source for the bundled Hogweed library.

- p11-kit: https://p11-glue.github.io/p11-glue/p11-kit.html, distributed under the following license:

```
/*
 * Copyright (c) 2011, Collabora Ltd.
 *
 * Redistribution and use in source and binary forms, with or without
 * modification, are permitted provided that the following conditions
 * are met:
```

```
 *
 *     * Redistributions of source code must retain the above
 *       copyright notice, this list of conditions and the
 *       following disclaimer.
 *     * Redistributions in binary form must reproduce the
 *       above copyright notice, this list of conditions and
 *       the following disclaimer in the documentation and/or
 *       other materials provided with the distribution.
 *     * The names of contributors to this software may not be
 *       used to endorse or promote products derived from this
 *       software without specific prior written permission.
 *
 * THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS
 * "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT
 * LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS
 * FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE
 * COPYRIGHT OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT,
 * INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING,
 * BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS
 * OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED
 * AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY,
 * OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF
 * THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH
 * DAMAGE.
 *
 * Author: Stef Walter <stefw@collabora.co.uk>
 */
```

- libtasn1: https://www.gnu.org/software/libtasn1, distributed under the GNU Lesser General Public License version 2.1 or later.n

- libunistring: https://www.gnu.org/software/libunistring/, distributed under the GNU Lesser General Public License (LGPL).

- libxml2: http://www.xmlsoft.org/, distributed under the MIT License.