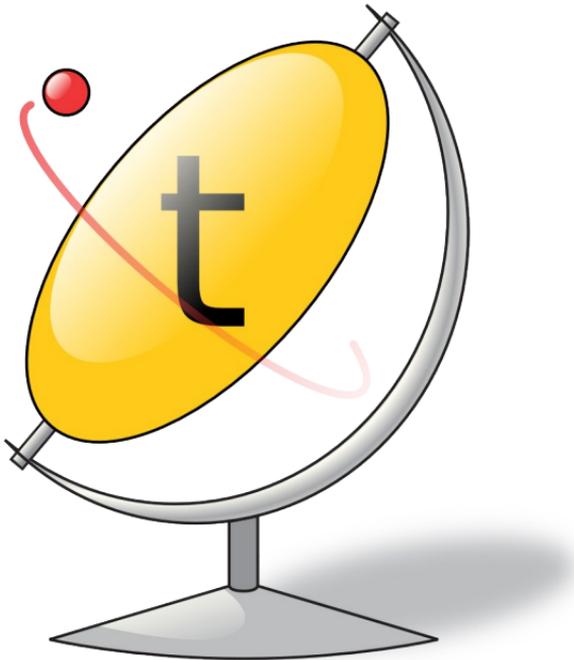


Textadept

A fast, minimalist, and remarkably extensible text editor



Quick Reference

Fourth Edition

Mitchell

Textadept Quick Reference

Textadept is a fast, minimalist, and remarkably extensible cross-platform text editor for programmers. This quick reference contains a wealth of knowledge on how to script and configure Textadept using the Lua programming language. It groups the editor's rich API into a series of tasks in a convenient and easy-to-use manner.

This book covers how to:

- Leverage Textadept's important files and folders
- Adeptly navigate and manipulate text
- Mark lines and text visually
- Show interactive lists and call tips
- Prompt for user input in various ways
- Spawn asynchronous, interactive child processes
- Configure colors, themes, and other settings
- Define lexers for highlighting source code
- And much more

Mitchell is the author and principal developer of Textadept and commands over 10 years of experience with Lua.

foicica.com

FOURTH EDITION

Textadept
Quick Reference

Mitchell

Textadept Quick Reference

by Mitchell

Copyright © 2013, 2015, 2016, 2018 Mitchell. All rights reserved.

Contact the author at mitchell@foicica.com.

Although great care has been taken in preparing this book, the author assumes no responsibility for errors or omissions, or for damages resulting from the use of the information contained herein. All product names mentioned in this book are trademarks of their respective owners.

Editor: Ana Balan

Cover Designer: Mitchell

Interior Designer: Mitchell

Indexer: Mitchell

Printing history:

December 2013: First Edition

May 2015: Second Edition

October 2016: Third Edition

August 2018: Fourth Edition

ISBN: 978-0-9912379-4-4

Preface to the Fourth Edition

This book is an updated version of the third edition of *Textadept Quick Reference*. It includes many of the new features introduced in the Textadept 9.x releases, and covers the backwards-incompatible changes made for Textadept 10.x. In a nutshell, this book covers the following new topics:

- Textadept's new and improved lexer syntax.
- Its simplified configuration file.
- The new color selection dialog.

For a comprehensive list of changes between Textadept versions, please refer to the editor's *CHANGELOG.md* or *doc/CHANGELOG.html* files, which are distributed with the application. The online version is located at <https://foicica.com/textadept/CHANGELOG.html>.

Contents

Introduction	1
Download	2
Conventions	2
Terminology	2
Environment Variables	3
Important Files and Directories	3
Command Line Options	8
Define Custom Options	9
Global Variables	9
Platform Variables	10
Handle Events	10
Create Buffers and Views	11
Query View Information	12
Handle Buffer and View Events	13
Work with Files and Projects	13
Detect or Change File Encodings	15
Query File Information	16
Handle Input and Output Events	17
Work with Sessions	18
Configure Session Settings	18
Move Around	18
Move Within Lines	18
Move Between Lines	19
Move Between Pages	20
Move Between Buffers and Views	20

Other Movements	21
Handle Movement Events	21
Manipulate Text	22
Retrieve Text	22
Set Text	23
Delete Text	25
Transform Text	26
Undo and Redo	28
Employ the Clipboard	29
Handle Text Events	30
Select Text	30
Make Simple Selections	31
Make Multiple Selections	34
Make Rectangular Selections	35
Query Selection Information	36
Search for Text	38
Simple Search	40
Search and Replace	41
Interact with the Find & Replace Pane	42
Incremental Search	44
Handle Find & Replace Events	44
Query Buffer Information	45
Query Position Information	45
Query Line and Line Number Information	46
Query Measurement Information	47
Configure Line Margins	47
Query Margin Information	49
Handle Margin Events	49

Mark Lines with Markers	50
Bookmark Lines	54
Query Marker Information	54
Annotate Lines	54
Query Annotated Lines	55
Mark Text with Indicators	55
Highlight Words	58
Query Indicator Information	58
Handle Indicator Events	58
Show an Interactive List	59
Display an Autocompletion List	59
Display a User List	61
Configure List Behavior and Display	61
Display Images in Lists	62
Query Interactive List Information	65
Handle Interactive List Events	65
Show a Call Tip	66
Configure Call Tip Display	67
Query Call Tip Information	67
Handle Call Tip Events	68
Fold or Hide Lines	68
Query Folded or Hidden Line Information	69
Scroll the View	70
Prompt for Input with Dialogs	71
Prompt with MessageBox Dialogs	71
Prompt with Inputbox Dialogs	73
Prompt with File Selection Dialogs	75
Prompt with a Textbox Dialog	76

Prompt with Dropdown Dialogs	77
Prompt with a Filtered List Dialog	79
Prompt with an Option Dialog	81
Prompt with a Color Dialog	83
Manipulate the Command Entry	84
Issue Lua Commands	85
Compile and Run Code	85
Configure Compile and Run Settings	86
Handle Compile and Run Events	87
Spawn Processes	87
Configure Textadept	89
Configure Indentation and Line Endings	90
Configure Character Settings	91
Configure the Color Theme	92
Create or Modify a Color Theme	93
Configure the Display Settings	100
Configure File Types	109
Configure Key Bindings	110
Configure Key Settings	114
Configure Snippets	114
Configure Miscellaneous Settings	117
Define a Lexer	117
Declare the Lexer Configuration	117
Construct Patterns	118
Define Tokens	122
Define Rules	122
Assign Styles	122
Specify Fold Points	123
Embed Lexers	124

Query Lexer Properties and Rules	125
Handle Lexer Events	125
Manually Style Text	125
Refresh Styling	125
Assign Plain Text Styles	126
Style Plain Text	127
Query Style Information	127
Miscellaneous	128
Handle Miscellaneous Events	128
Appendix: Image Formats	133
XPM Image Format	133
RGBA Image Format	134
Index of Key and Mouse Bindings	135
Lua API Index	145
Concept Index	157

Introduction

Textadept is a fast, minimalist, and remarkably extensible cross-platform text editor for programmers. Written in a combination of C and Lua¹ and relentlessly optimized for speed and minimalism for over nine years, Textadept is an ideal editor for programmers who want endless extensibility without sacrificing speed or succumbing to code bloat and featuritis.

Textadept runs in both graphical and text-based user interface environments. The text-based version of the editor is referred to as the “terminal version”, since it executes within a terminal emulator.

Textadept Quick Reference is designed to help the user “get things done” when it comes to scripting and configuring Textadept. Its pragmatic approach assumes the user has a working knowledge of both Lua and Textadept. This book is broken up into a number of descriptive sections with conveniently grouped tasks that cover nearly every aspect of Textadept’s Application Programming Interface (API). For the most part, the contents of each task are not listed in conceptual order. They are listed in procedural order, an order the user would likely follow when writing Lua scripts. This quick reference serves as a complement to Textadept’s comprehensive Manual and extensive API documentation.

While this book aims to be a complete reference, it does omit some of the less useful features of Textadept’s API. For example, although many of Textadept’s table fields are both readable and writable, this reference sometimes chooses to cover only one of those operations. (Unless a field is marked “Read-only” or “Write-only”, it is readable and writable.) This book also does not cover Lua’s standard libraries.

Finally, the facilities in this book are designed to be used primarily in user-written Lua scripts and in the occasional “one-shot” Lua command. If the user keeps this in mind, he or she can realize Textadept’s full potential.

1 <https://www.lua.org>

Download

Textadept binary packages for Windows, Mac OSX, and Linux platforms are available from <https://foicica.com/textadept>. Each package is self-contained and need not be installed. Textadept's source code is also included in each archive. The user may compile the application manually by following the instructions in the editor's Manual.

Conventions

This book uses the following conventions.

Italic

Used for filenames and for introducing new terms.

Constant width

Used for environment variables, command line options, and Lua code, including functions, tables, and variables.

Constant width italic

Used for user-specified parameters.

[]

Used for optional function arguments, except in code examples that index Lua tables. Unless otherwise specified, optional arguments default to `nil`.

Terminology

This book uses the following terminology.

Buffer

An object that contains editable text.

View

An object that contains a single buffer.

Caret

Either the visual that represents the text insertion point or the end point of a text selection.

Anchor

The start point of a text selection or search.

Virtual Space

The space past the ends of lines.

Lexer

A Lua module that highlights the syntax of source code written in a particular programming language. Textadept refers to a programming language by its lexer's name.

Style

A collection of display settings specific to source code comments, strings, keywords, and other ranges of text.

Language Module

A Lua module automatically loaded by Textadept when editing source code in a particular programming language. The module's name matches the language's lexer name. Not all languages have language modules.

Environment Variables

Textadept utilizes the following environment variables.

HOME or USERHOME

The user's home directory. Textadept's user data and preferences exist in a *.textadept/* sub-directory, denoted as *~/.textadept/* throughout this book.

On Windows, this directory is typically *C:\Users\user name*. On Mac OSX, it is */Users/username/*. On Linux and BSD it is often */home/username/*.

LANG

The user's default locale. Textadept will display localized text and messages in it if possible.

TA_LUA_PATH

TA_LUA_CPATH

The Textadept equivalent of *LUA_PATH* and *LUA_CPATH*. Used by Lua's *require()* function for finding modules.

Important Files and Directories

Textadept allows the user to configure and customize the editor using several important files and directories contained

Search for Text

Textadept supplies a variety of tools to search for text: a simple search API, a more complex API for search and replace, a Find & Replace Pane for interactive search and replace, and an incremental find entry. The first two tools make use of the search flags defined in Table 2 and the regular expression syntax in Table 3. The last two tools make use of the search flags defined in the section “Interact with the Find & Replace Pane” on page 42 and the regular expression syntax in Table 3.

NOTE

Textadept’s regular expressions are based on the C++11 standard’s ECMAScript syntax².

Table 2. Buffer search flags

Bit Flag	Description
<code>buffer.FIND_MATCHCASE</code>	Match search text case sensitively.
<code>buffer.FIND_WHOLEWORD</code>	Match search text only when it is surrounded by non-word characters. <code>buffer.word_chars</code> is a string that contains all word characters.
<code>buffer.FIND_WORDSTART</code>	Match search text only when the previous character is a non-word character.
<code>buffer.FIND_REGEXP</code>	Interpret search text as a regular expression. (See Table 3.)

Table 3. Regular expression and Lua pattern special characters

Regex	Lua	Meaning
<code>.</code>	<code>.</code>	Matches any character.
<code>[[[:alpha:]]</code>	<code>%a</code>	Matches any letter.
<code>\d</code>	<code>%d</code>	Matches any digit.

2 <http://www.cplusplus.com/reference/regex/ECMAScript/>

Regex	Lua	Meaning
<code>[[:lower:]]</code>	<code>%l</code>	Matches any lower case character.
<code>[[:punct:]]</code>	<code>%p</code>	Matches any punctuation character.
<code>\s</code>	<code>%s</code>	Matches any space character.
<code>[[:upper:]]</code>	<code>%u</code>	Matches any upper case character.
<code>\w</code>	<code>%w</code>	Matches any alphanumeric character.
<code>[[:xdigit:]]</code>	<code>%x</code>	Matches any hexadecimal digit.
<code>[set]</code>	<code>[set]</code>	Matches any character in <i>set</i> , including ranges (e.g. <code>[A-Za-z]</code>).
<code>[^set]</code>	<code>[^set]</code>	Matches the complement of <i>set</i> .
<code>*</code>	<code>*</code>	Matches the previous item (Regex) or <i>character class</i> (Lua) zero or more times. The previous Lua expressions listed are character classes.
<code>+</code>	<code>+</code>	Matches the previous item or class one or more times.
<code>*?</code>	<code>-</code>	Matches the previous item or class zero or more times, but as few times as possible.
<code>+?</code>		Matches the previous item one or more times, but as few times as possible.
<code>?</code>	<code>?</code>	Matches the previous item or class once, or not at all.
<code>{m,n}</code>		Matches the previous item between <i>m</i> and <i>n</i> times.
<code>{m,}</code>		Matches the previous item at least <i>m</i> times.
<code>{m}</code>		Matches the previous item exactly <i>m</i> times.
<code> </code>		Matches either the previous item or the next item.
	<code>%bxy</code>	Matches a balanced string that starts with <i>x</i> and ends with <i>y</i> .
	<code>%f[set]</code>	Matches a position where the next character belongs to <i>set</i> , but the previous character does not.
<code>\<</code>		Matches the beginning of a word.
<code>\></code>		Matches the end of a word.
<code>\b</code>		Matches a word boundary.
<code>^</code>	<code>^</code>	Matches the beginning of a line unless inside a set.

Regex	Lua	Meaning
\$	\$	Matches the end of a line unless inside a set.
((The beginning of a captured matching region.
))	The end of a captured matching region.
(?: ...)		Consider the matched sequence as a single, uncaptured item.
\n	%n	Represents the <i>n</i> th captured matching region's text. In replacement text, "0" represents all matched text.
\x	%x	Represents non-alphanumeric character <i>x</i> , ignoring any special meaning it may have by itself.

Simple Search

The user can perform simple searches anchored at the caret position. Example 8 demonstrates a simple (but not comprehensive) method of jumping to the beginning of the next occurrence of the word under the caret.

Example 8. Find the next instance of the current word

```
-- Determine the word under the caret.
local pos = buffer.current_pos
local s = buffer:word_start_position(pos, true)
local e = buffer:word_end_position(pos, true)
local word = buffer:text_range(s, e)
-- Perform the search.
local flags = buffer.FIND_MATCHCASE +
              buffer.FIND_WHOLEWORD
buffer:search_anchor()
pos = buffer:search_next(flags, word)
if pos ~= -1 then buffer:goto_pos(pos) end
```

`buffer:search_anchor()`

Anchors the position that `buffer:search_next()` and `buffer:search_prev()` start at to the caret position.

`buffer:search_next(flags, text)`

`buffer:search_prev(flags, text)`

Searches for and selects the next or previous occurrence of string `text` using search flags bit-mask `flags`, returning that occurrence's position or `-1` if `text` was not found.

Concept Index

Symbols

~/textadept/, 3, 8, 10

A

annotations, 54
autocompleting code, 59
autocompletion list
 configuring, 61
 displaying, 59
 images in,
 displaying, 62, 64
 information, 65
autopaired characters, 25

B

block comments, 28
bookmarks, 21, 54, 97
brace matching, 98, 107
buffers
 creating, 12
 line information in, 46
 list of open, 9
 manipulating text in (see
 manipulating text)
 measurements, 47
 moving around in (see
 moving around)
 moving between, 20
 position information
 in, 45
 searching and replacing
 in (see searching
 for text)
 selecting text in (see
 selecting text)

C

call tip
 configuring, 67, 94, 99
 displaying, 66
 information, 67

character classifications, 91
clipboard operations, 29
code autocompletion, 59
code folding, 68, 106, 123
color dialog, 83
color theme
 bookmarks, 97
 carets, 95
 changing, 92
 color definitions, 93, 95
 highlighted words, 99
 indicators, 98
 location of, 6
 long lines, 99
 margins, 96
 markers, 97
 matching braces, 98
 selections, 96
 styles for, 94
 whitespace, 99
Command Entry, 84
 Lua commands with,
 issuing, 85
command line options, 8-10
commenting code, 27
compiling and running
 code, 85-87
configuring Textadept
 ~/textadept/, 8
 autopaired characters, 25
 block comments, 28
 character classifications,
 91
 color theme (see color
 theme)
 compile and run
 code, 86
 display settings (see
 display settings)
 file types, 110

- configuring Textadept
 - (continued)
 - key bindings (see key bindings)
 - line endings, 90
 - line indentation, 90
 - locale, 5
 - matching braces, 107
 - sessions, 8, 18
 - snippets (see snippets)
 - typeover characters, 25

D

- deleting text, 25
- dialogs
 - color, 83
 - dropdown, 77-79
 - file selection, 75
 - filtered list, 79-81
 - inputbox, 73
 - messagebox, 71-73
 - option, 81
 - textbox, 76
- display settings
 - caret, 100
 - indentation guides, 107
 - long lines, 106
 - matching braces, 107
 - mouse cursor, 104
 - scrollbars, 102
 - selections, 101
 - whitespace, 101
 - window, 108
 - wrapped lines, 105
 - zoom, 105
- downloading Textadept, 2
- dropdown dialog, 77-79

E

- encodings
 - converting between, 28
 - for files, 15, 16
 - of filesystem, 10
 - supported, list of, 15

- end of lines, 90
- environment variables, 3
- events
 - autocompletion list, 65
 - buffer and view, 13
 - call tip, 68
 - compile and run, 87
 - connecting to, 11
 - CSI, 129
 - double click, 129
 - dwll, 129
 - emitting, 11
 - error, 129
 - Find & Replace, 44
 - focus, 129
 - indicator, 58
 - initialized, 130
 - input and output, 17
 - interactive list, 65
 - keypress, 129
 - lexer, 125
 - margin, 49
 - mouse, 130
 - movement, 21
 - no command line
 - arguments, 128
 - quit, 130
 - reset, 130
 - resume, 130
 - suspend, 130
 - tab click, 130
 - text, 30
 - update, 130
 - user list, 65
 - zoom, 130

F

- file encodings, 15
- file filters, 15
- file information, 16
- file operations, 13-16
- file selection dialog, 75
- file types, 109
- filesystem encoding, 10

- filtered list dialog, 79-81
- filtering text through shell
 - commands, 26
- Find & Replace Pane, 42
 - Regex syntax for, 38-40
 - search flags for, 43
 - searching and replacing
 - with, 43
 - searching in files
 - with, 43
- finding text (see searching for text)
- fold markers, 51
- folding lines, 68
- fonts and font sizes, 94, 105

H

- hiding lines, 69
- highlighting words, 58, 99

I

- image formats
 - RGBA, 134
 - XPM, 133
- incremental searching, 44
- indentation, 26, 46, 90
- indentation guides, 107
- indicators, 55-58, 98
- init.lua, 4
- input, prompting for (see dialogs)
- inputbox dialog, 73
- inserting text, 23
- installing Textadept, 2
- interactive lists (see autocompletion list; user list)
- internationalizing
 - messages, 9

K

- key bindings
 - configuring, 113
 - modifier keys, list of, 113

- special keys, list of, 113
- terminology, 110-112

L

- language modules, location of, 5
- lexers
 - changing, 109
 - code folding, 123
 - defining, 117
 - embedding, 124
 - fold points, 123
 - information, 110
 - location of, 6
 - patterns, 118-121
 - properties for, 125
 - rules, 122
 - styles, 122
 - tokens, 122
- line annotations, 54
- line endings, 90
- line indentation, 26, 46, 90
- line information, 46
- line margins, 47-49, 96
- line markers, 50-54, 97
- line wrapping, 104
- lines
 - annotations, 54
 - bookmarking, 54
 - endings for, 90
 - folding, 68
 - hiding, 69
 - indentation for, 26, 46, 90
 - information for, 46, 69
 - joining, 27
 - long, 99, 106
 - marking, 50-54, 97
 - moving between, 19
 - moving up or down, 27
 - moving within, 18
 - splitting, 27
 - transposing, 26
 - wrapping, 104

- locale, 3, 5
- localizing messages, 9
- long lines, 99, 106
- Lua commands, issuing, 8, 85
- Lua pattern syntax, 38-40

M

- manipulating text
 - clipboard, using the, 29
 - converting between encodings, 28
 - deleting, 25
 - inserting, 23
 - replacing, 24
 - retrieving, 22
 - setting, 23
 - transforming, 26
- margins, 47-49, 96
- marking lines, 50-54, 97
- marking text, 55-58, 98
- matching braces, 98, 107
- measurements, 47
- messagebox dialog, 71-73
- modules, location of, 5
- moving around
 - between bookmarks, 21
 - between buffers, 20
 - between lines, 19
 - between pages, 20
 - between paragraphs, 21
 - between views, 20
 - selecting and, 32, 35
 - within lines, 18
- multiple selections, 34, 37

O

- option dialog, 81
- overtyping mode, toggling, 128

P

- pages, moving between, 20

- paragraphs, moving
 - between, 21
- pipelining text through shell commands, 26
- pixmap, 133
- position information, 45
- printing messages, 23
- processes, spawning
 - of, 87-89

Q

- quitting, 128

R

- rectangular selections, 35-37
- replacing text, 24, 41
- resetting, 128
- retrieving text, 22
- RGBA image format, 134
- running code, 85-87
- running Textadept, 8

S

- scrolling, 70, 103
- search flags, 38, 43
- searching for text
 - Find & Replace Pane, using the, 42-44
 - in files, 43
 - incrementally, 44
 - regular expression syntax for, 38-40
 - replacing and, 41
 - search flags for, 38, 43
 - simple search, 40
- selecting text
 - modal selection, 33
 - multiple selection, 34
 - rectangular selection, 35
 - simple selection, 31
 - while moving, 32
- selections, 36, 96, 101
- sessions, 8, 18

- setting text, 23
- snippets
 - configuring, 116
 - inserting, 23
 - special characters, list
 - of, 116
 - terminology, 114
- spawning processes, 87-89
- split views, 12
- style information, 127
- styles, 94
- styling text, 125-127
 - (see also lexers)
- switching buffers, 20
- switching views, 20
- syntax highlighting, 109, 125
 - (see also lexers)

T

- target ranges, 24, 27, 41
- text indicators, 55-58, 98
- text manipulations (see
 - manipulating text)
- text selections (see selecting
 - text; selections)

Textadept

- configuring (see
 - configuring
 - Textadept)
- downloading, 2
- installing, 2
- running, 8
- user data directory of, 3,
 - 8, 10

- textbox dialog, 76, 77
- theme (see color theme)
- transforming text, 26
- transposing characters and
 - lines, 26
- typeover characters, 25

U

- undo and redo actions, 28
- user data directory, 3, 8, 10

user list

- configuring, 61
- displaying, 61
- images in,
 - displaying, 62-64
- information, 65

V

- variables, 9, 10
- views
 - information, 12
 - list of open, 9
 - moving between, 20
 - scrolling, 70
 - splitting, 12
 - unsplitting, 12

W

- window, 108
- wrapping lines, 104

X

- XPM image format, 133

Z

- zooming, 105