

[Home](#)[Download](#)[Buy](#)[Blog](#)[Forum](#)[Support](#)

Sublime Text 2

Download

The current version of Sublime Text 2 is 2.0.2.

Sublime Text 3 is currently in beta, and contains many improvements over Sublime Text 2.

- [OS X](#) (OS X 10.6 or later is required)
- [Windows](#) - also available as a [portable version](#)
- [Windows 64 bit](#) - also available as a [portable version](#)
- [Linux 32 bit](#)
- [Linux 64 bit](#)

Sublime Text 2 may be downloaded and evaluated for free, however a license must be purchased for continued use. There is currently no enforced time limit for the evaluation.

Please submit feature requests to <http://sublimetext.userecho.com/>. For notification about new versions, follow [sublimehq](#) on twitter.

Version 2.0.2

Release Date: 8 July 2013

- Sublime Text 3 beta is now available from <http://www.sublimetext.com/3>
- Removed expiry date
- Backported various fixes from Sublime Text 3
- Improved minimap click behavior. The old behavior is available via the `minimap_scroll_to_clicked_text` setting
- Added `copy_with_empty_selection` setting, to control the behavior of the copy and cut commands when no text is selected

Version 2.0.1

Release Date: 14 July 2012

- Keyboard input while dragging a selection will cancel the drag
- Improved backspace behavior when `use_tab_stops` and `translate_tabs_to_spaces` are true
- Improved shift+drag behavior
- Improved double click drag select behavior
- About Window shows the license key details
- Fixed a Goto Anything issue where pressing backspace could scroll the overlay incorrectly
- Fixed a crash triggered by double clicking in the Goto Anything overlay
- Fixed incorrect window position when dragging a tab in some scenarios
- Added missing `toggle_preserve_case` command
- `word_wrap` setting accepts "true" and "false" as synonyms to true and false
- OSX: Fixed System Preferences menu not working
- Linux: Added support for mice with more than 5 buttons
- Linux: Fixed an occasional graphical glitch when running under Unity

Version 2.0

Release Date: 26 June 2012

- See the [blog post](#) for a summary.
- OSX: Added Retina display support
- Added "Quick Skip Next", on Command+K, Command+D / Ctrl+K, Ctrl+D
- Added text drag and drop
- Reloading a file maintains the selection
- OSX: Find uses the system find pasteboard
- OSX: Added support for the ODB Editor Suite
- Theme: Added support for the highlight_modified_tabs setting
- Added setting show_full_path, to control if the full path is shown in the window title
- Added setting bold_folder_labels
- OSX: Bundle is signed in preparation for Mountain Lion
- OSX: Added scroll bar support for the "Jump to the spot that's clicked" system preference
- OSX: Added create_window_at_startup setting
- OSX: Folders accessed via symlinks are monitored for changes
- OSX: Fixed Lion press-and-hold inserting two characters
- Windows: Installer is signed
- Windows: Directory scanning makes use of inodes where available
- Windows: Files are opened with the FILE_SHARE_DELETE share mode
- Windows: Fixed a regression in command line handling with network paths
- Windows and Linux: Tweaked command line handling when Sublime Text isn't already running
- Linux: Tweaked handling of the Primary selection
- Linux: GTK is loaded at runtime, removing any dependency on specific libpng versions
- Linux: Fixed occasional jittering with the auto complete window
- Linux: Fixed a directory watching issue
- Linux: Sub-folders within the syntax menu on the status bar work as expected
- Linux: Show Completions has changed from ctrl+space to alt+/, to not interfere with IMEs
- Reworked CSS completions
- Improved Toggle Comment for HTML and CSS
- Added Open in Browser context menu for HTML files
- Saving a file now calls fsync to ensure the data is written to disk
- Added support for file_include_patterns and folder_include_patterns in projects
- Find in Files can exclude directories using "-some_path/" syntax
- Added BOM variants to the Save with Encoding menu
- Added setting preview_on_click
- Improved Indentation Detection
- ASCII control codes are rendered in a different style
- Auto complete and tool tips stay within the screen boundaries
- Double clicking in the tab header area will create a new tab
- Delete Folder will prompt before moving the folder to the trash
- Double clicking on whitespace will select only whitespace
- Theme: Added 'expandable' attribute to tree_rows, enabling folders to be styled differently
- Theme: Tweaked quick panel to better indicate the selected item
- Disabling find result highlighting will also disable find-as-you-type in the Find panel
- Pressing escape in the incremental find panel will set the selection to the search start point
- Extended behavior of home and end keys on word wrapped lines
- Changes in file name case are detected in the side bar
- Fixed save_on_focus_lost causing Goto Anything to not show if the current file has unsaved changes
- Fixed folders not being added to the recent folders list
- Fixed rendering of active indent guides
- Fixed pasting file:line expressions into the Goto Anything panel
- Invalid key names in keymap files will be logged to the console
- Build Systems: Added variants

- Build Systems: Improved debug output of the exec command when the target isn't found
- Build Systems: Added show_panel_on_build setting
- Build Systems: Build times are reported
- Build Systems: Exit code is reported, if it's non-zero
- Build Systems: Improved output panel height serialization
- Build Systems: Canceling a build will terminate rather than kill the process
- Vintage: Added backspace motion (thanks mrannanj)
- Vintage: Improved quote text object (thanks Guillermo)
- Vintage: Added Ctrl+U and Ctrl+D (thanks misfo)
- Vintage: 'vintage_use_clipboard' forces pasting from system clipboard too (thanks Guillermo)
- Vintage: Fixed motions with "0" in them for actions (thanks bengolds)
- Vintage: Visual mode I and A (thanks misfo)
- Vintage: Added vintage_use_clipboard setting to yank to the system clipboard (thanks Guillermo)
- Vintage: Tweaked behavior of cw and cW when the caret is in whitespace (thanks misfo)
- Vintage: Added bindings for moving the group focus up, down, left, right (thanks misfo)
- Vintage: Paste accepts a repeat count (thanks misfo)
- Vintage: Added bindings for u and U in visual mode, to change the case of the selection and exit visual mode (thanks quarnter)
- Vintage: Macro replay accepts a repeat count (thanks zbuc)
- Vintage: r<char> processes lines individually (thanks Guillermo)
- Vintage: Tweaked % behavior (thanks misfo)
- Vintage: Fix bug that caused deletes to erroneously left_delete (thanks misfo)
- Vintage: Fix EOF behavior with linewise changes (thanks misfo)
- Vintage: Ignoring a and i when in visual mode (thanks Guillermo)
- Vintage: cc and S take counts (thanks misfo)
- Vintage: Correctly shrink selections when reversed (thanks misfo)
- Vintage: Added vap (thanks behrends)
- API: Added sublime.message_dialog()
- API: Added sublime.ok_cancel_dialog()
- API: Added Command.is_checked() to support checkbox menu items
- API: Added sublime.log_result_regex()

Build 2181

Release Date: 22 February 2012

- See the [blog post](#) for a summary
- New Icon, by The Iconfactory
- File and Global settings have been combined into a single settings file
- Selecting a word will highlight other occurrences. This can be controlled with the match_selection setting
- Reworked text rendering for improved quality
- Improved automatic indentation. It now uses bracket matching to better determine indentation
- Added setting indent_to_bracket and smart_indent, to customize automatic indentation
- Auto Complete is less intrusive
- Renaming a file via the side bar will rename any open files
- Improved folder scanning performance on huge directory trees
- Untitled files have their name set automatically from the first line of the file. The can be controlled with the set_unsaved_view_name setting
- Ruby: #{} is expanded when # is entered in a string
- Folder names are shown in the window title if no project is open
- Folders in projects are stored with relative paths
- Added Save with Encoding
- Spelling errors are indicated with an underline only, the foreground color of the text is no longer changed
- Added no_bold and no_italic font options

- Added margin setting, to control the size of the gutter
- Key bindings for Goto Anything, Save, etc will work if a panel has input focus
- "Open all with current extension as" handles filenames without extensions
- Fixed an auto complete display issue
- Fixed find_under_expand not setting whole_word correctly
- Fixed Paste and Indent adding unwanted whitespace when pasting within a line
- Fixed a minor compatibility issue with .tmLanguage files
- Fixed a tab dragging issue on OS X and Windows
- OS X: Tweaked sub-pixel positioning to match TextEdit and Terminal. The previous behavior is available via the no_round font option.
- OS X: Open File and Open Folder have been combined into a single Open menu item
- OS X: Control+Option+Left/Right will move by sub words, in the same manner as Control+Left/Right
- Windows: Default font is now Consolas
- Windows: Improved text rendering quality when using Direct Write
- Windows: Added support for --wait
- Windows: Added support for drive letter relative paths passed on the command line
- Windows and Linux: Added key bindings for ctrl+shift+backspace/delete
- Linux: Added a workaround for a glibc bug when MALLOC_CHECK_ is defined
- Linux: Pango is used for font rendering, improving support for CJK text
- Vintage: Escape cancels current input (thanks fjl)
- Vintage: Pasting over visual mode selections works as expected (thanks misfo)
- Vintage: Linewise changes set indentation (thanks misfo)
- Vintage: Linewise motions work as expected (thanks misfo)
- Vintage: c, C, s and S update registers (thanks misfo)
- Vintage: Tweaks to G and gg (thanks guillermooo)
- Vintage: Added zt and zb (thanks djjcast)
- Vintage: Added gq (thanks orutherfurd)
- Plugin initialization time is reported in the console at startup
- API: Tweaked module importing
- API: Added window.find_open_file(file_name)
- API: Added sublime.run_command()
- API: Added view.unfold([regions])
- API: Added view.folded_regions()

Build 2165

Release Date: 12 January 2012

- See the [blog post](#) for a summary
- Theme: New UI theme, by [Mike Rundle](#)
- Added support for bold and italic font styles
- Speed: Improved startup time
- Speed: Optimized memory usage
- Speed: Improved font glyph rasterization speed
- Rearranged the menu, making important features are easier to locate
- Auto complete: Completion menu is shown automatically. This is configurable with the auto_complete file setting
- Auto complete: Tab can be used instead of enter to accept the current completion, by using the auto_complete_commit_on_tab file setting
- Auto complete: Many quality improvements
- Tab Completion: Pressing tab multiple times will cycle through the available completions
- Added reopen_last_file command, bound to Command+Shift+T / Ctrl+Shift+T
- Deleting files via the side bar will send them to the trash, rather than deleting them outright
- Improved behavior when hot_exit and remember_open_files are false
- Color schemes and syntax definitions are reloaded on the fly
- Paste and Indent will now do the right thing in more circumstances
- save_on_focus_lost won't save if the underlying file has been deleted
- Added key binding for replace_next

- slurp_find_string will set the whole_word flag based on if the selection is empty
- Added file setting move_to_limit_on_up_down. This is enabled by default on OS X.
- Shift+Drag will set the active end of the selection as expected
- Clicking on a tab will always give the relevant sheet input focus
- Simplified indent guide settings, and disabled active indent guide drawing by default. The indent_guide_options file setting can be used to change this
- Improved minimap dragging
- increase_font_size command will allow sizes up to 128pt
- Changed how the toggle argument to show_panel is handled
- Files with colons in their names may be opened from the command line
- Fixed handling of selectors with extraneous ',' characters
- File Encoding: Added default_encoding setting
- File Encoding: Added Hexadecimal encoding, for basic editing of binary files. This is controlled with the enable_hexadecimal_encoding global setting
- Projects: Folders may be given a name attribute, to change how they're displayed
- Projects: Exclude patterns may include paths, for example: file_exclude_patterns: ["tmp/*.cc"]
- Projects: .sublime-project files passed on the command line will be opened as projects, without --project being required
- Projects: Folder paths with trailing slashes are supported
- Projects: Fixed folder_exclude_patterns not being serialized correctly in project files
- OS X: Prompting for elevated privileges when saving protected files
- OS X: Double clicking on a .sublime-project file in Finder will open it in Sublime Text
- OS X: Control clicking on a tab will show the context menu
- OS X: Increased default font size
- OS X: Fixed an issue that opened additional empty windows when using the command line helper
- Linux: Fixed an issue with tab dragging
- Vintage: Column is maintained when moving between lines
- Vintage: == reindents current line (thanks Gordin)
- Vintage: Added ZZ (thanks Guillermo00)
- Vintage: Added X command (thanks misfo)
- Vintage: Added _ command (thanks redjohn)
- Vintage: Added ~ (thanks misfo)
- Vintage: r,enter is supported (thanks misfo)
- Vintage: Added various Ctrl+W,<x> bindings for layout control (thanks Gordin)
- Vintage: Added various z,<x> bindings for folding (thanks Gordin)
- Vintage: _ accepts a count (thanks guillermooo)
- Vintage: Ctrl+C is a synonym for escape on OS X(thanks sugarcoded)
- Vintage: dw works as expected at EOL
- Vintage: Fixed an issue with the zz key binding (thanks Bradley Priest)
- API: Plugins have a finite time to respond to events before users will be issued a warning indicating the slow plugin
- API: Added sublime.log_input()
- API: Added view.encoding() and view.set_encoding()
- API: Added view.line_endings() and view.set_line_endings()
- API: Added view.viewport_position() and view.set_viewport_position()
- API: Added view.viewport_extent() and view.layout_extent()
- API: Added view.text_to_layout() and view.layout_to_text()
- API: Added view.line_height() and view.em_width()
- API: Added window.active_view_in_group(group)
- API: Added window.views_in_group(group)
- API: Added window.get_layout() and window.set_layout()
- API: Added window.get_view_index() and window.set_view_index()
- API: settings.has() now works as expected
- API: on_query_completions results may have a tab embedded in their descriptions, to delimit help text

Build 2139

Release Date: 2 November 2011

- See the [blog post](#) for a summary
- Added Replace in Files functionality to the Find in Files panel
- Find in Files panel now has a single "Where" field, combining the previous "In Files" and "In Location" fields
- Added fold buttons to the gutter. They can be customized with the `fold_buttons` and `fade_fold_buttons` file settings
- Added keyboard navigation to the side bar
- Re-added and improved side bar expand / collapse animations. This can be controlled with the `tree_animation_enabled` global setting
- Alt clicking a folder in the side bar will expand it and all its children
- The reindent command can be used to reformat selected lines, via the Edit/Line /Reindent menu item
- Improved startup time in some scenarios
- OSX: Menu items flash when triggered with a key binding
- OSX: Improved open file behavior when no windows are currently open
- OSX: Command+Right Delete will delete to EOL
- OSX: Tweaked click-through behavior
- OSX: Fixed opening an already open file from Finder with `open_files_in_new_window` enabled creating a new window
- Windows and Linux: Ctrl+Scroll Wheel will change the font size
- Ctrl+F will select the find pattern when the find panel is already open
- The `show_panel` command now accepts a `toggle` parameter, which is false by default. The previous behavior was to always toggle
- Pressing escape while dragging a scroll bar will cancel the scroll
- Fixed a Ctrl+D/Super+D regression, where wrapping at EOF wasn't handled correctly
- Fixed a Find in Files regression where unsaved files couldn't be opened from the Find Results buffer
- Fixed a crash that could occur when trying to drag a closed panel
- Fixed a rare crash that could occur after moving a file between windows
- Vintage: Added '=' action to reindent
- Vintage: 'o' in visual mode will reverse the selection direction (by Guillermo)
- Vintage: Added support for the "" key to go to bookmarks (by Guillermo)
- Vintage: Escape will exit visual line mode, instead of changing to visual mode
- Vintage: Added gf (by Guillermo)
- Vintage: Added z. and z,enter (by Guillermo)
- Vintage: Added zz (by Johnfn)
- Vintage: Added "{" and "}" motions (by Johnfn)
- Vintage: Fixed a regression with escape in 2126, causing it not to exit from visual mode
- API: `window.run_command()` will run text commands, dispatching them via input focus
- API: `hide_auto_complete` is safe to call from `on_modified`

Build 2126

Release Date: 28 September 2011

- See the [blog post](#) for a summary
- Hot Exit is now the default, and configurable via the `hot_exit` global setting
- Added Code Folding, see the *Edit/Coding Folding* menu for details. More folding options will be added in the future.
- Find in Files results are colored for better visibility
- Matching tags are highlighted
- Improved word wrapping of source code: extra indentation is now added on wrapped lines
- Added global setting `highlight_modified_tabs` to better indicate tabs with unsaved changes
- Goto Anything: Improved ranking accuracy on repetitive file paths
- Goto Anything: Files are now displayed in most recently used order, and shown by default
- Side Bar: The side bar will auto hide if open files aren't shown, and there are no open folders

- Side Bar: Disabled animation on expand/collapse
- Side Bar: Headings are no longer indented
- Side Bar: Clicking on a folder will expand or collapse it, rather than selecting it
- Side Bar: Open files are now hidden by default
- Side Bar: The current file is highlighted in both the Open Files and Open Folders sections, if present
- Side Bar: Folders will be expanded by default
- The find highlight outline color now has a better default, and is configurable via the highlight key in the color scheme
- *Ctrl+F3* (*Option+Command+G* on OS X) will search by whole words if the selection is empty
- *Ctrl+D* (*Command+D* on OS X) will search by whole words if the selection is initially empty
- Find All no longer disables the "In selection" flag
- Find Preview will not be shown if finding for "." in regex mode
- Find Next Using Selection unsets the reverse flag
- Find in Files accepts relative paths
- Projects: File names in workspaces are stored with relative paths
- Projects: Fixed project specific settings persisting when the project is closed
- Indent Guides: Fixed solid indent guides not rendering crisply with sub-pixel glyph positioning
- Indent Guides: Fixed indent guide rendering and word wrapping in conjunction with Perl syntax highlighting
- *Ctrl+Tab* will only cycle between files in the current group
- Improved undo grouping logic, so the amount of typing undone is more predictable
- Added Expand Selection to Indentation (*Command+Shift+J* on OS X, and *Ctrl+Shift+J* on Windows and Linux)
- Expand Selection to Tag will handle unclosed tags
- OSX: Clicking on a background window will bring it to the front only, and not change the selection
- OSX: Sequential key bindings are displayed in the menu
- OSX: *Ctrl+Delete* will delete by sub words
- OSX: Added workaround for Core Text crashing on invalid unicode code points
- OSX: Killing the subl process while waiting for files to be closed no longer causes problems for the application
- OSX: Fixed dictionary lookups being offset
- Linux: *Ctrl+Q* is bound to quit
- Fixed rot13 tripping up on non-ascii characters
- Vintage: Find no longer selects the found text
- Vintage: Arrow keys, page up/down and home/end act as motions
- Vintage: All Ctrl key bindings are now enabled only if the `vintage_ctrl_keys` setting is true, which is the default for OS X, but not Windows and Linux.
- Vintage: *Ctrl+[* is now an alias for escape (if `vintage_ctrl_keys` is enabled)
- Vintage: Added *Ctrl+E* and *Ctrl+Y*, to scroll up/down by lines, if `vintage_ctrl_keys` is enabled.
- Vintage: *I* will insert at the first non-whitespace character on the line (thanks Trent)
- Vintage: Added *ga* (thanks Guillermo)
- Vintage: Improved *H*, *L* and *M* motions (thanks Guillermo)
- Vintage: *IG* and *G* are distinguished
- Theme: Tabs with modified files now have a "dirty" attributes that can be used in selectors
- Theme: Added key `indent_top_level` for the `sidebar_tree` class
- API: Added `view.fold()` and `view.unfold()`

Build 2111

Release Date: 31 August 2011

- See the [blog post](#) for a summary
- Added indentation guides. These may be disabled with the `draw_indent_guides` setting

- Added syntax, indentation, and spell checking information to the status bar
- Added Vintage mode, for vi key bindings in Sublime Text 2
- Added Expand Selection to Tag (*Command+Shift+A* on OS X, and *Ctrl+Shift+A* on Windows and Linux)
- Added Close Tag (*Command+Alt+.* on OS X, and *Alt+.* on Windows and Linux)
- Projects are now stored in two files: `.sublime-project` (user editable, can be checked into version control) and `.sublime-workspace` (containing the session state). Existing project files will be converted on load.
- Projects: Folders in projects may be given `file_exclude_patterns` and `folder_exclude_patterns`
- Projects: Per-project file settings may be specified in the `.sublime-project` files
- Projects: Build systems may be specified within `.sublime-project` files
- Projects: Open Project and Recent Projects open the project in a new window
- Added File/Open Folder, to open a folder in a new window. These entries will appear in the Open Recent list.
- Passing folders on the command line will open them in a new window by default
- Shift+Tab will always unindent when the caret is at the beginning of the line
- Tweaked Delete Word
- Tweaked behavior of up/down with non-empty selections
- Build is now bound to *Ctrl+B* Windows and Linux (*Command+B* on OS X, as it was previously). F7 is also bound to build on all platforms.
- Toggle Side Bar is now bound to *Command+K*, *Command+B* on OS X, and *Ctrl+K*, *Ctrl+B* on Windows and Linux
- Added the ability to hide the open files section on the side bar, via the View/Side Bar menu
- Added global setting `close_windows_when_empty`, to close windows as soon as the last file is closed. This is enabled by default on OS X
- Find and replace panels operate via commands, and indicate the relevant key bindings on the buttons
- Updated C syntax highlighting
- Updated Bash highlighting
- `fallback_encoding` is respected by Find in Files
- The copy command is now recorded in macros
- Resolved a syntax highlighting issue in `contentName` handling
- Fixed a crash that could occur with replace in selection
- Fixed an issue that could result in an incorrect scroll range when `scroll_past_end` is enabled
- Fixed incorrect tab positions in some circumstances
- Fixed an issue that could erroneous carets in the replace panel
- Middle clicking on tab close buttons is no longer ignored
- OS X: Fixed Quit not quitting until the next input event
- Windows: Fixed delay when opening multiple files from Explorer
- Windows: Fixed Open Containing Folder for files in root directories
- Linux: Hid the window resize grip on Ubuntu
- API: Added `Command.description()`
- API: Added `sublime.score_selector(scope, selector)`
- API: Added `view.score_selector(point, selector)`
- API: Added `view.find_by_selector(selector)`
- API: `view.scope_name()` now returns scope components in the expected order. `view.syntax_name()` is still available for compatibility.
- API: Added `sublime.get_macro()`
- API: Added `sublime.MONOSPACE_FONT` option for the quick panel
- API: If a plugin module has a top level function called `unload_handler`, it'll be called when the plugin is reloaded or unloaded.
- API: Fixed a crash when reloading plugins while threads are active
- API: API functions better handle null windows

Build 2095

Release Date: 2 August 2011

- Improved labeling in Preferences menu
- Added context menu key support (*alt+F2* in OS X)

- Files opened from the command line use the last active window on any virtual desktop, rather than only looking on the current virtual desktop
- OSX: Sub-pixel glyph placement is disabled when rendering without anti-aliasing
- Windows without open files now display the project name
- Fixed a scenario where tabs would be incorrectly positioned
- Fixed a scenario where Jump to Matching Bracket could cause the carets to not update
- Changed Preferences menu mnemonic from 'r' to 'n', to remove a conflict with the find panel
- Dialogs now accept both the normal enter/return key and the one on the keypad
- OSX: Fixed a crash when closing a window in full screen mode
- OSX: Fixed a scenario that could result in carets not blinking after toggling full screen
- OSX: Fixed multi-character key bindings that don't use modifiers
- OSX: Updated keyboard handling to support the 'Dvorak - Qwerty Command' keyboard layout
- OSX: Fixed a rare crash bug crash when redrawing the minimap
- OSX: Updated .dmg file with a symlink to /Applications
- API: Fixed a crash when calling `view.run_command("save")` on a background view
- API: Added `view.command_history()`
- API: Added `Settings.add_on_change()` and `Settings.clear_on_change()`

Build 2091

Release Date: 29 July 2011

- OSX: Text layout uses sub-pixel positions
- OSX: Added support for OS X 10.7 style scroll bars (over-bounce is not yet implemented)
- OSX: Added support for Lion style full screen
- Linux: Sub-pixel anti-aliasing is enabled by default
- Carets blink by default (can be changed with the `caret_style` file setting).
- Files with the same name, but different directories, now have unambiguous labels in the side bar and tabs
- New windows are created with the settings and size of the previous window. This supersedes the `new_window_settings`.
- `duplicate_line` will operate on non-empty selections, duplicating the selection
- Added global setting `show_tab_close_buttons`
- Added *Reveal in Side Bar* context menu item
- Added *Tools/New Snippet* menu item
- Added menu items for syntax specific settings and distraction free settings
- Fixed performance issues with SQL and JavaScript syntax highlighting on files with long lines
- `shellVariables` values specified in `.tmPreferences` files are available to snippets, fixing many Rails snippets.
- OSX: Changed key bindings for Find, Replace and Full Screen to match Lion defaults
- OSX: Files and folders dropped onto the dock icon will be opened in a new window by default
- OSX: `Command+Alt+O` will toggle overwrite mode
- OSX: `Command+,` will open the user file settings
- OSX: Initial support for input methods
- Dropped files will be opened in the pane they're dropped on
- Added global setting `binary_file_patterns`, to exclude files from Goto Anything and Find in Files
- Commands may be passed as arguments via `--command`
- Enabling `scroll_past_end` in `Distraction Free.sublime-settings`
- Line highlights are disabled by default
- Improved scrolling logic when working with cloned files, and resizing word wrapped files
- Improved behavior of page up / page down
- Improved Toggle Comment behavior on empty selections for languages without line comments

- Find in Files: 'Use Buffer' is now selected by default
- "New Folder" menu item can create multiple folders at once
- Panel heights are stored in the session
- Goto Symbol is able to display symbols with forward slashes
- Fixed scenario where characters with a background color could be drawn clipped
- Fixed a scenario that could result in Goto Anything rows appearing in the wrong location
- OSX and Linux: Fixed reading and writing of UTF-16 encoded files
- OSX and Linux: Select Folder dialogs allow multiple selection
- OSX: Fixed not being able to click on the auto complete window
- Fixed Goto Line not updating the status bar
- Fixed Goto Symbol not updating the status bar
- Linux: Improved redraw performance
- Windows: Save As dialog will prompt to overwrite files
- Windows: Fixed a scenario where the update notification dialog could be missed
- Using a bullet, rather than an asterisk, to indicate dirty files
- Theme: Tab labels fade rather than elide, and are better positioned to make use of available space
- Theme: Images are reloaded on the fly
- Theme: Added a 1px border to the right of the side bar
- Theme: Added new classes `sidebar_container`, `overlay_control` and `sheet_container_control`
- Theme: Minimap viewport color is themeable, via the `minimap_control` class
- Theme: Theme specific widget settings are now used, via `Widget - <theme name>.sublime-settings`
- Theme: Buttons now use standard label controls for their text
- Theme: Label controls now have properties `font.face`, `font.size`, `font.bold`, `font.italic`, `shadow_color` and `shadow_offset`
- Theme: Layers now have a repeat property
- Theme: Controls that used to use 'background_color' now use layers instead
- API: Added `window.views()`
- API: More strictly enforcing the main thread only usage restriction
- API: `show_input_panel()` will run its callbacks at a time when they're able to show panels and overlays
- API: Changed handling of `on_query_completion` results with an empty trigger
- API: Added `window.hwnd()` (Windows only)

Build 2076

Release Date: 1 July 2011

- Relabeling from Alpha to Beta
- Added Command Palette (Windows and Linux: *Ctrl+Shift+P*, OSX: *Command+Shift+P*)
- Added Distraction Free mode (accessible via *View/Enter Distraction Free Mode*)
- Selected text is rendered with syntax highlighting, unless the color scheme specifies a `selectionForeground` key
- Added a progress bar when loading files
- OSX: Distributing as a universal binary, with 64 bit and 32 bit support
- Added a key binding to wrap the selection in a tag (Windows and Linux: *Alt+Shift+W*, OSX: *Ctrl+Shift+W*)
- Monokai is now the default color scheme, rather than Monokai Bright
- Added a *File/Close All* menu item
- Folders may now be dropped anywhere on the window, not just on the side bar
- Improved selection and line highlight rendering
- Selection border color can be customized in `.tmTheme` files with the `selectionBorder` key
- Find highlight color will default to yellow if a color scheme doesn't specify one
- Added `encode_html_entities` command (accessible via the Command Palette)
- Added `rot13` command (accessible via the Command Palette)
- Added `shift_tab_unindent` setting, to force shift+tab to unindent even without a multi-line selection

- Added setting `ensure_newline_at_eof_on_save`
- Increased search limit for the Jump to Matching Bracket command
- Improved auto-indent behavior in the face of mixed indentation
- Improved toggle block comment on an empty selection
- Tweaked auto-pairing rules
- Fixed Add Line Before adding indentation on lines where there should be none
- Fixed issue when using Find/Replace with back references and a reversed search direction
- JavaScript: Added support for node.js shebangs
- Improved the HTML completions for `link` and `style`
- Shell Script: Fixed file type associations
- Added Toggle Comment support for YAML
- Fixed shutdown not proceeding while folders are being scanned
- Fixed spell checking for words with non-ascii prefixes or suffixes
- Fixed a quirk when using Goto Anything, typing, and then clearing the text box
- Fixed the minimap initially showing at the wrong size on new files
- Fixed a scenario where the label and close button on a tab would be in the wrong spot
- Fixed a rare crash that could occur after closing a file, then changing layout
- Windows: Improved support for menu hiding
- OSX: Changed the color space used for rendering
- Linux: Fixed an issue that could cause the status bar not to show at startup
- Linux: Changed Python compiler settings, to enable loading of compiled python extensions, and importing the hashlib module
- Build Systems: Current directory when building will default to the file's directory if none is specified in the build system
- Build Systems: `result_line_regex` may specify a column and message (using the 2nd and 3rd capture groups)
- `insert_snippet` command: extra arguments passed to the command will be forwarded onto the snippet
- The `auto_complete` command now takes an optional `disable_auto_insert` argument
- API: Added `window.show_quick_panel()`
- API: Added `sublime.platform()` and `sublime.arch()`
- API: Added `window.focus_view()`
- API: Improved the behavior of tab completion in conjunction with `on_query_completions` returning non-word triggers
- API: Calling `run_command()` from Python will no longer bypass `is_enabled()` checks
- API: Fixed `view.replace()` scrolling the view

Build 2065

Release Date: 15 June 2011

- Improved cold startup time
- Added key bindings to toggle find options, which are shown in the tool tips
- Find: The number of matching occurrences is indicated in the status bar
- Find: Found text is highlighted in a separate color
- Added "Reopen with Encoding" menu
- Updated Scala package
- Added Clojure syntax highlighting
- Added Haml syntax highlighting
- Added `reStructuredText` syntax highlighting
- Syntax menu is now organized by package
- Improved `.tmLanguage` file compatibility with regards to 0 length matches in rules
- Added 'theme' global setting
- Themes are automatically reloaded
- Theme selectors are more powerful, and can now select based on the classes and attributes of parents
- Color Schemes: Added `inactiveSelection` and `inactiveSelectionForeground` keys
- Added `Solarized` color schemes
- OSX: Using a white ibeam cursor on dark backgrounds
- OSX: Added key binding for Wrap Paragraph at Ruler (Command+Option+Q)

- OSX: Added document icons
- OSX: Fixed a scenario that could result in the save panel double adding an extension
- API: Added `Window.num_groups()`, `Window.active_group()`, and `Window.focus_group(group)`
- API: Added `Window.folders()`, to access the currently opened folders
- API: Region rendering will use the foreground and background colors for a given scope if both are specified in the `tmTheme` file
- Fixed UI glitch with the replace panel and empty replacement strings
- Fixed a crash with very large or very small font sizes

Build 2059

Release Date: 28 April 2011

- Added smooth scrolling
- Added spell checking
- Tab Completion is enabled by default. Shift+Tab can be used to insert a literal tab
- Pressing Ctrl+Space after a tab completion will adjust the last completion, rather than starting a new one
- Tab completion is smarter about when to insert a completion, and when to insert a tab
- Auto complete is now case insensitive
- Added global setting, `remember_open_files`
- Disclosure buttons on the tree control have a larger target area
- Tweaked tab header background for improved contrast
- Status bar column display correctly accounts for tabs
- Fixed an issue with scrollbars on small files
- Shift+Mouse wheel scrolls horizontally
- Recent files no longer appear in the Goto Anything list
- Improved startup time
- Updated HTML completions
- Updated path handling in `exec.py`
- OSX: Added support for Help/Search
- OSX: Activating the application when no windows are open creates a new window
- Linux: Fixed a crash that could occur in Save All

Build 2051

Release Date: 19 March 2011

- Bracket matching is smarter, and will ignore brackets in comments and strings
- OSX and Linux: Fixed a bug that could result in spiraling CPU usage
- Added `"::"` pairing in CSS
- Reworked Layout menu
- Scroll bar pucks are rendered with subpixel precision
- Windows and Linux: Moving to the right by words will stop at the end of words, rather than the beginning
- OSX: Improved minimap rendering quality
- OSX: Fixed a regression in 2046 with some keyboard layouts, such as Japanese and Chinese
- OSX: Changed mouse button bindings. Ctrl+Left Mouse will now open a context menu
- Linux: Current window is brought to front when opening a file from the command line
- Linux: Context menu is shown on button press, instead of button release
- "Remove Folder" is now "Remove Folder from Project"
- Fixed ordering of the Default package
- Fixed Wrap Paragraph not correctly accounting for leading whitespace
- API: Added `sublime.version()` and `sublime.channel()`
- API: Fixed `view.set_status()` and `view.erase_status()` not refreshing the status bar

Build 2047

Release Date: 13 March 2011

- Fixed mouse input regression in the license window and the select project window
- OSX: Changed how the project is displayed in the title bar

Build 2046

Release Date: 12 March 2011

- Added a side bar menu, with the ability to do basic file manipulation
- Snippets with punctuation triggers are supported again
- Improved automatic snippet field cancellation
- Added Show Unsaved Changes context menu item
- Reworked keyboard input
- Mouse buttons can be configured via the new *.sublime-mousemap files
- Windows and Linux: Right mouse + scroll wheel will change files
- Mouse buttons 4 and 5 will change files
- Rotating the mouse wheel in the tab header area will change tabs
- Linux: Window position and state are recorded in the session
- Ctrl+W (Command+W on OSX) will close the window if no files are open
- Ctrl+Shift+W (Command+Shift+W on OSX) will close the window
- Quote auto-pairing is smarter, and will do the intended thing more often
- Added find_selected_text file setting
- Packages may add items to menus, including the main menu, context menu, and side bar menu
- Main.sublime-menu is automatically reloaded when changed
- Added New Plugin and New Build System menu items
- Build files may specify environment variables, using the env key
- Added build_env file preference, which will be picked up by the build system
- Add Line Before will insert the correct amount of indentation
- Tweaked status message display
- Tweaked switch_file
- Disabled typing while dragging
- Added diff syntax highlighting
- Fixed regression in 2036 that caused unsaved file indicators to not show in the title bar
- Linux: Fixed locale related bug that could cause multi-pane layouts to not work
- OSX: Fixed Command+{X,C,V,Z,Y,A} not working in save and open panels
- API: Added Command.is_visible(args)

Build 2039

Release Date: 4 March 2011

- Fixed an issue with Linux font rendering
- Show minimap setting is respected when cloning
- Fixed flicker that could occur when showing the auto complete window

Build 2036

Release Date: 3 March 2011

- Tab completion is now off by default, but can be turned back on with the new tab_completion setting
- Added *File/Clone File* menu item
- Showing the project in the title bar
- Populating file types in Open and Save dialogs
- Ctrl+Shift+K will delete the current line (all platforms)
- Improved delete_word command, it will now delete a word and a neighboring space when it makes sense to do so

- Added Wrap Paragraph command (*Alt+Q* on Windows and Linux)
- Updated PHP completions, with thanks to *vostok4*
- Linux: Added support for global menus, as used by Ubuntu 11.04
- Linux: Using `Xdg_CONFIG_HOME` if set
- Windows and Linux: Fixed slow startup when large folders are open
- OSX: *Option+Arrows* will stop at the beginning of words when moving left, and the end of words when moving right
- OSX: Drawing shadows on popup windows
- OSX: Fixed mouse over highlights sometimes remaining on when the mouse leaves the window
- Build systems support a key containing platform specific settings ("windows", "osx" or "linux")
- Build systems support an optional "target" setting, which defaults to "exec". This specifies the command that will be run when the build system is triggered
- Fixed the selector field in .sublime-build files not being interpreted correctly
- Corrected the file_regex in Make.sublime-build
- Fixed a crash bug that could occur when using Ctrl+P with save_on_focus_lost enabled
- Fixed serialization of large numbers in settings files
- Documentation: Added API Reference

Build 2032

Release Date: 26 February 2011

- Opening files no longer does any blocking IO in the main thread; working with remote files is much better now
- Goto Anything gives file previews a short time to load before displaying them, reducing flicker from asynchronous loading
- Reworked snippets and auto complete into a unified system
- Word completions can be inserted by pressing tab
- Added .sublime-completions file format, for pre-defined completions
- Added completion entries for HTML and PHP
- Added new setting, save_on_focus_lost, to automatically save files when switching applications
- Improved font glyph generation speed
- Folder scanning is faster
- Content based syntax detection now takes a lower precedence to the file extension based detection
- next_view / prev_view commands won't update the stack until the modifier keys are released
- In the replace panel, ctrl+alt+enter will perform a replace all
- OSX: Supporting "no_antialias" and "gray_antialias" font rendering settings
- OSX: Not showing the proxy icon when the Goto Anything panel is open, to prevent Cocoa from doing unwarranted blocking IO
- OSX: Restoring the window title when returning from full screen mode
- Windows: Hide Menu is now restored correctly on startup
- Fixed default_line_ending not being applied to empty files
- API: Added auto complete API, `EventListener.on_query_completions()`
- API: Added `on_activated(view)` and `on_deactivated(view)`
- API: save and prompt_save_as commands are accessible from the view

Build 2027

Release Date: 19 February 2011

- Added bookmarks
- Opened folders are kept synchronized with the file system
- Individual folders can be removed via their context menu
- Folders are loaded incrementally. Refresh Folders also works incrementally, and will not reset the expanded/collapsed state of the folder tree
- Goto Anything: the ranking function will try harder to find the best match for each file

- Added for emacs style marks, via commands `set_mark`, `select_to_mark`, `delete_to_mark` and `swap_with_mark`. Marks interact well with multiple selections.
- Added an emacs style kill ring, and a corresponding yank command.
- Added command `show_at_center`, which scrolls the view to center on the selection. This is bound to `Ctrl+K`, `Ctrl+C` on Windows and Linux, or `Ctrl+L` on OSX.
- Added Word Wrap Column menu
- Added Ruler menu
- Linux: Single instance support. This will only be enabled if a recent version of GTK is installed.
- Linux: Support for the `--wait` command line parameter
- Linux: Keypad enter key will insert a newline
- Linux: `WM_CLASS` is set
- OSX: `Ctrl+K` now works as expected when the cursor is at the end of a line
- OSX: `Ctrl+B` is now unbound, so the default action of moving backwards by a character works
- OSX: Option+Shift+Up/Down will work as expected
- OSX: Implemented `centerSelectionInVisibleArea` (`Ctrl+L`), and yank (`Ctrl+Y`) messages
- Any keys bound to move by lines will take effect in Goto Anything and the filter window. Specifically, on OSX `Ctrl+P` and `Ctrl+N` can be used to select the next and previous items.
- Windows: Added fake scrollbars to the window, as required by some belligerent mouse drivers
- Stronger protection against infinitely recursive directory trees, especially for file systems that don't report inodes
- In Global Settings, `file_exclude_patterns` has been separated from `folder_exclude_patterns`
- Changes to `folder_exclude_patterns` and `file_exclude_patterns` are applied on the fly
- Fixed Find in Files not recursing in some scenarios
- `command_mode` now disables character input, as it should
- Automatic Build System selection now prefers Build Systems in the User folder
- Fixed an issue with curly bracket pairing
- Normalizing newlines emitted from exec'd programs
- `delete_word` no longer deletes punctuation and whitespace at the same time
- Fixed a crash when trying to focus a group that doesn't exist
- API: Added `view.show_at_center(point)`
- API: `add_regions()` can be given an icon name
- API: Added `sublime.log_commands(bool)`, to log the commands that are being run

Build 2023

Release Date: 13 February 2011

- Fixed a crash in Goto Anything introduced in the previous version
- Goto Anything file previewing is faster
- Windows: Can now hide the menu, revealing it when alt is pressed
- Multi-part extensions, such as "index.html.erb" work as expected
- Updated syntax highlighting for Ruby and Ruby on Rails
- Syntaxes are always shown in the menu, even if they don't have any associated file types
- Added AppleScript syntax highlighting
- Linux: Tweaked ignored modifier keys
- Fixed window title being set incorrectly in some circumstances

Build 2020

Release Date: 11 February 2011

- New version numbering scheme
- OSX: Using sparkle for automatic updates

- OSX: Added command line helper
- File syntax is detected by looking at the first line of the file
- Added shadows to the text area when the sides are clipped. These can be customized with the shadowWidth and shadow properties in .tmTheme files
- Adding folders is faster
- Added Copy File Path to the context menu
- OSX: Tweaked font rendering
- OSX: Open and save dialogs follow the current directory
- OSX: Improved startup time
- OSX: *Command+Shift+S* will Save As, *Command+Alt+S* will Save All
- OSX: Fixed a scenario where files passed to the application were ignored
- OSX: Fixed a crash when clicking on the select project window
- Find in Files: The global folder_exclude_patterns are respected
- Find in Files: Folder names may be used in the include / exclude patterns
- Linux / OSX: Adding folders with recursive symlinks are handled properly
- Fixed a rendering issue with extremely long tokens
- Fixed a crash bug if the folder list is cleared while it's still loading
- Fixed a crash when pressing Ctrl+Tab with no open files
- Added --project command line parameter
- Tweaked scrolling behavior of the exec command
- Improved symbol detection, especially for C++
- Added new key "gutterForeground" to color schemes
- Updated Haskell.tmLanguage
- Updated Python.tmLanguage
- Updated XML.tmLanguage
- Updated CSS.tmLanguage
- Lua: Highlighting the not keyword
- CSS: Fixed the '{' key binding not working as expected
- Fixed explicitly set file types not being restored correctly from the session
- Fixed Goto Anything displaying erratically when no files are open
- API: Added view.set_syntax_file(file_name)
- API: Implementing the on_close() callback

20110203 Changelog

- Added tab context menu
- Added "draw_minimap_border" setting
- Added subword movement: Alt+Left/Right on Windows and Linux, Ctrl+Left/Right on OSX
- Menu Mnemonics for Windows and Linux
- Added Switch File menu
- Folders passed on the command line are handled correctly
- When closing a file that's currently loading, the load will be canceled immediately
- Syntax definitions are loaded asynchronously
- JavaScript: Better identification of functions for the symbol list
- About window is modal
- Added Ant and Erlang build systems
- Made find highlights and bracket match highlights more visible
- Goto Anything: When multiple folders are open, the folder name will be prefixed to the displayed file name
- Fixed spaces in Goto Anything throwing off the highlighting character
- Linux: Setting the mouse cursor
- Linux: Support for running via a symlink
- Linux: Support for remote files
- Linux: Fixed a problem with drag and drop if the dropped filenames contained spaces
- Linux: Checkboxes only show up on menu items where they should
- Linux and Windows: Alt+<number> will switch to the given tab. Changing layout has moved to Alt+Shift+<number>
- Windows: Improved startup speed
- Windows: "-z arg" may be passed on the command line, and it will be ignored
- OSX: Implemented scrollToBeginningOfDocument: and scrollToEndOfDocument:

- OSX: Fixed Quad Pane layout incorrectly being bound to Option+4 (now Command+Option+4)
- OSX: Fixed a crash that could occur in Select Project
- OSX: Fixed a crash when pasting if the pasteboard contents can't be converted to a string
- API: Fixed an interaction between EventListener plugins and the console panel
- API: API functions validate they're being called on the correct thread

20110130 Changelog

- Fixed a crash that could occur on startup
- Fixed a crash that could occur when changing pane layout
- Added menu items to change font size, in Preferences/Font
- OSX: Adding support for more actions, including moveToBeginningOfParagraph: and moveToEndOfParagraph: deleteToEndOfParagraph:, and more. This enables Ctrl+a, Ctrl+e and Ctrl+k.
- Adding the same folder a second time to the folder list will do nothing
- Added a hover state to the disclosure buttons in the side bar
- Ctrl+[and Ctrl+] will indent / unindent on Windows and Linux
- Implemented the command_mode setting
- Linux: Fixed Ctrl+Shift+<number> key bindings
- Linux: Open file dialog will allow multiple files to be selected
- API: Python TextCommand plugins are triggered by widgets. This enables commands like delete_word (i.e., Ctrl+Backspace).
- API: Added on_query_context
- API: Added sublime.load_settings() and sublime.save_settings()
- API: Added settings.erase()
- API: view.text_point works as expected when passing in negative rows and columns
- API: parser module may be imported on Linux

20110129 Changelog

- Linux: Fixed an incompatibility with non-ubuntu distributions
- Windows: Fixed an issue when running from a path with unicode characters
- OSX: Command+1 will select the first tab, Command+2 the second, and so on
- OSX: Command+M will minimize the window, match brackets is now Ctrl+M
- OSX: Added Zoom to the Window menu
- OSX: scroll_past_end defaults to false
- Ignoring spaces when searching for files in Goto Anything
- Goto Anything: '@' operator will select the current symbol by default
- Goto Anything: ':' operator won't scroll the view until a line number is entered
- Goto Anything: '#' operator won't set the whole word flag on the find panel
- Goto Anything: '#' operator can be used to search for punctuation
- Goto Anything: Fixed a bug where an unexpected file could be selected
- Added syntax highlighting for Go
- Tightened up tab close animation
- Giving more feedback when adding a folder to the sidebar
- Improved find panel status messages
- API: Added sublime.packages_path()
- API: Added sublime.installed_packages_path()

20110127 Changelog

- Initial public version