Keyboard Maestro 5 Documentation

Overview

Keyboard Maestro will take your Macintosh experience to a new level. Keyboard Maestro enables you to create or record custom macro shortcuts that you can activate at any time. For example, your macros could help you navigate runnings applications or work with an unlimited number of clipboards. Best of all, every macro you create is available using simple keystrokes you choose. The only limit to Keyboard Maestro is your imagination!

Using Keyboard Maestro’s powerful Macros, you can control applications, windows, or menus; insert text; open documents, applications or URLs; execute scripts; control the system, iTunes or QuickTime Player; and more — all with the touch of a key, click of the mouse, periodically or at specific times, when an application is launched or running, or even over the Internet using Keyboard Maestro Control on your iPhone or iPod touch, or by using a web browser to access Keyboard Maestro’s built-in web server.

Using Keyboard Maestro’s Application Switcher and Window Switcher you can cycle through applications or windows, closing, hiding, launching, and more. The Application Launcher lets you quickly launch applications. Keyboard Maestro can even help you regain control of your crowded screen by letting you close or hide multiple windows or applications.

Using Keyboard Maestro’s Clipboard History Switcher, you can access previous clipboards. You'll never lose something on your clipboard again! The Clipboard History Switcher also makes it easy to copy and paste multiple items, such as a username and password or a serial number, without having to switch back and forth between applications for each item. You can mark a clipboard as a favorite so it stays forever or send it to another Mac running Keyboard Maestro.

Using Keyboard Maestro’s Clipboard Switcher, you can select from a number of named clipboards, storing information away for later retrieval, or store the details you are working on and access them by name again and again.

Keyboard Maestro requires an Intel Mac running Mac OS X 10.6 or later.

Keyboard Maestro is free to try with no limitations. Once the trial period ends, a license must be purchased to continue using it.

What’s New

Keyboard Maestro 5 takes the modern engine of Keyboard Maestro 3, the modern user interface of Keyboard Maestro 4 and adds to that base an incredible breadth and depth of power with almost no increase in complexity.

Changes for 5.3.2

- Fixed HID device crashing bug particularly prevelent in Mountain Lion.
- Fixed Read File variable editor so you can click in the variable field.
- Fixed calculation operators that can be followed by numbers or letters (eg MOD).
- Fixed "timout" typo.

Changes for 5.3.1

- Resolved an issue with executing Automator Workflows in Mountain Lion.

- Resolved a performance issue with the Button Condition.
- Notify via Growl when a timeout cancels a macro.
- Fixed some memory leaks.
- Fixed the SECONDS() function.
- Fixed a performance issue with getting the Finder's selection.
- Fixed the YEAR/MONTH/DAY/DOW/HOUR/MINUTE/SECOND to be in local timezone.
- Fixed some edge cases in the Hard Wrap filter.
- Allow "APPLICATION" in the menu condition.
- Avoid "no autorelease pool" warnings on launch of engine in Mac App Store version.
- Removed "move to" logging.
- Fixed Edit with TextEdit macro library entry.
- Fixed short form variables with spaces or underscores in their names.
- Fixed triple click action editor to actually triple click.
- Fixed a case where the editor get confused when switching macros.

Changes for 5.3

- Enhanced Mouse Click action:
  - Added support for mouse drag.
  - Click relative to a found image on the screen.
  - Click relative to the center of the window/screen/image.
  - Double and triple click.
  - Right click (and center button and other buttons).
- New Screen Capture action captures one or all screens or a selected window.
- New Image actions:
  - Create New Image.
  - Flip image.
  - Rotate image.
  - Resize Image (Resize image, resize canvas, add or remove margins, crop).
  - Composite (images or styled text) onto Image.
  - Draw Shape (line, rectangle or oval) onto Image.
  - Trim Image.
  - Display Clipboard (image or styled text).
  - Get Image Size.
  - Find Image on Screen.
- New Screen Contains Image condition.
- New Highlight Location action.
- Allow "APPLICATION" in select menu as an alias for the application name.
- Process text tokens in the Clipboard, Script, and Path condition tests.
- Added live reporting of the results of condition tests in action editors.
- Added selectedMacros AppleScript command.
- Allow short form variable tokens (%variable name%).
- Added MidX and MidY to the SCREEN & WINDOW functions.
- Fixed some edge cases in the Unwrap text filter.
- Fixed the Tutorial.
- Fixed a bug where selecting the Variable in the Search and Replace Variable action did not stick.
- Removed the five second delay in displaying applications/utilities.
- Fixed the %WindowPosition%n% token
- Corrected non-edit display of File collection recursive status.
- Fixed a bug that caused expressions with commas to fail in some calculation fields.
- Process text tokens in the Clipboard, Script, and Path condition test.
- Do not process \ characters in the regular expression search fields.
- Added Image and File categories to Export Library.

Changes for 5.2

- Compatibility Note: Tokens being parsed in some actions may mean single percents now need to be doubled (%%)
- Compatibility Note: Tokens now parse for \x characters, so the backslash may need to be doubled (\\)
- New File actions:
  - New Folder.
  - Move (or rename) a file or folder.
  - Copy a file or folder.
  - Duplicate a file or folder.
  - Trash a file or folder.
  - Delete a file.
  - Delete a file or directory (recursively).
  - Read a file to a variable or to the clipboard (can be an image or styled text).
  - Write to a file from a variable or the clipboard (can be an image or styled text).
  - Get/Set File Attributes.
- New Filters:
  - Last Path Component
  - Delete Last Path Component
  - Path Extension
  - Delete Path Extension
  - Base Name
  - Expand Tilde In Path
  - Abbreviate With Tilde In Path
  - Resolve Symlinks
  - Standardize Path
- New Tokens:
  - %WindowFrame%n% (and WindowPosition, WindowSize) token.
  - %WindowName%n% token.
  - %Application%n% token.
  - %NetworkLocation% token.
  - %ICUDateTimePlus% token for displaying offseted formatted (eg now + 3 days).
  - %LineFeed%, %Return%, %Tab%, %Space% tokens.
  - \a,\b,\e,\f,\t,\r,\n characters.
  - %NN% or %NNNN% tokens for arbitrary hex unicode characters.
- New Calculation functions:
  - MOUSEX, MOUSEY, MOUSEBUTTON(n) - mouse coordinates and button states.
  - SCREEN(n, Left|Right|Top|Bottom|Width|Height - screen frame coordinates.
  - WINDOW(n, Left|Right|Top|Bottom|Width|Height) - window frame coordinates.
  - SCREENCOUNT - the number of screens (displays).
  - WINDOWCOUNT - the number of windows in the front application.
  - IDLE - the number of (fractional) seconds the Mac has been idle.
  - MILLISECONDS (or MILLS or MS) - fractional milliseconds since startup.

    - SECONDS - fractional seconds since startup.
    - TIME - return (unix) time in seconds since 1970.
    - JD, MJD, and TIME enhanced to allow specifying year,minth.day,hour,minute,second.
    - TIME2JD, TIME2MJD, JD2TIME, JD2MJD, MJD2TIME, MJD2JD conversion functions.
    - YEAR, MONTH, DAY, DOW, HOUR, MINUTE, SECOND functions which take a unix time in seconds since 1970.
    - APPLICATIONS function that returns the number of running applications.
    - ONLINE function returns whether the Mac is currently connected to the Internet.
    - BATTERY function returns whether the Mac is currently running off battery power.
    - SCREENSAVER function returns whether the Mac is currently displaying the screen saver (or the display is off).
- Added Edit -> Insert Token menu.
- Added Edit -> Insert Function menu.
- Tokens with numbers in them can now be calculations (eg %WindowFrame%3+4%)
- Improved code signing for Snow Leopard.
- Added option click on the File Reference buttons to reveal the target file in the Finder.
- Allow dragging of files/folders to file reference buttons.
- Fixed the Open URL action to actually allow tokens in the URL.
- Fixed a potential crashing bug precachine application icons (particularly prevelent in Mountain Lion)
- Fixed an issue with repeating triggers on daylight saving days with more than 24 hours.
- Avoid com.adobe.illustrator.aicb clipboard flavor to prevent issues with Adobe Illustrator.
- Fixed a bug in window placement when moving windows back on the screen.
- Fixed a potential crash with the MIDI action editor.

Changes for 5.1

- Added For Each action allowing looping over a variety of collections.
- Added Developer ID Signing for Mountain Lion Gatekeeper.
- Added Percent Encode for URL filter (Clipboard & Variable).
- Added text token processing to button names in Prompt For User Input.
- Added text token processing to Open URL and Search & Replace actions.
- Changed the way String Matching/Reg Ex menus are named.
- Changed icon for Display Text and Mouse actions.
- Disable changes to the Global Macro Group which were not allowed anyway.
- Fixed case correct Insert Text by Typing in applications like Screen Sharing.
- Fixed a crash that could happen if a text factory no longer exists.
- Fixed lock up when entering quoted strings in calculations.
- Remember the location of the Recording window (again!).
- Made yet more changes to the HID code to try to handle the system crash.
- Extended DisableAnimation to include the macro palette.
- Left justified the text in the text display window.

Changes for 5.0.3

- Improve the visibility of the clipboard HUD icons.
- Added %LastApplication% token.
- Updated to Growl SDK 1.2.2
- Removed a case where resolving old aliases could lock up for a long period of time.
- Removed "variables" debugging messages.
- Removed dependency on RBSplitView.
- Another attempt to fix HID crashes with wacom tablets.
- Stopped Keyboard Maestro launching (and immediately crashing) on pre-10.6.
- Resolved dragging failure after dragging an action to the Favorites category.
- Fixed a crash that could happen when the first action referenced the current macro.
- Fixed a crash that could happen with the Menu condition when there is no menu bar.
- Fixed drawing glitch in variable popup menus.
- Fixed the display of "inline text" popups in Lion to display in proper location.

Changes for 5.0.2

- Speak String action now processes tokens.
- Truncate window names in the middle instead of at the end.
- Add search/replace text to the non-Edit display.
- Refind lost applications.
- Library Window now remembers its position.
- Changed Share as Macro Library to Export as Macro Library.
- Improved a text insertion case.
- Fixed a long standing drawing glitch.
- Fixed a possible cause of the CPU issue with clipboard syncing.
- Fixed a possible crash when there is no front application.
- Some documentation tweaks.

Changes for 5.0.1

- Added a macro library entry showing how to store passwords in the Keychain.
- Added Set Variable to Clipboard and Clipboard to Variable pseudo actions.
- Allow negative window indexes in Manipulate Window to count from the back.
- Addde Help -> Open Preferences Folder since Lion hides the Library folder.
- Moved the Keyboard Maestro preference folder to the Application Support folder.
- Ensure the log files are created.
- Added Command-Shift-Tab trigger to default Activate Application Switcher.
- Added display of nested actions for Repeat command in non-Edit mode.
- Changed the origin of Screen tokens to top left like other origins in Keyboard Maestro.
- Ignore bogus windows in Manipulate Window when considering indexes.
- Resolve alias files in Open File or Folder action.
- Just include one Quick Macro macro by default.
- Added documentation links for ICU Date Time format.
- Added token processing for most of the condition fields.
- "Save Clipboard History" appears on in the preferences whether it is on or off.
- Handle the case where the preferences folder is readonly.
- Execute Script, Save to Variable, changes to the variable name were not sticking.
- Fixed Text Factory clipboard filtering.
- Fixed a UI glitch for Variable Condition "Is Empty".
- Fixed Swap Characters library entry.
- Fixed some potential problems with clipboard syncronization (including CPU excess usage)
- Fixed an edge case for multiple overlapping typed string triggers.
- Fixed an issue editing macros containing Open URL or Open File or Folder actions.
- Fixed the weird behaviour with multiple Record Quick Macros and duplicate UIDs.
- Fixed a potential crash displaying titles of clipboards.

- Fixed the Pixel Color test condition testing the pixel at the wrong location.
- Fixed a potential crash in the Pixel Color test condition.

Major Changes for 5.0

- Requires 10.6 and Intel.
- Control Flow - Pause Until, If/Then/Else, While, Until and Repeat with extensive conditional tests.
- Variables - permanently stored and accessible in a variety of ways.
- Calculations - based on variables or a variety of functions.
- Macro Library built in with ready-to-use macros.
- Undo/Redo - finally!
- Enhanced Application Switcher with Cover Flow, Force Quit, Get Info and Reveal.
- Enhanced Window Switcher with Cover Flow of Applications and window depth sorting.
- Enhanced Clipboard History with Favorites, Quick Look and Send clipboard to another Mac.
- Application Launcher.

Control Flow & Variables & Calculations

- New Control Flow actions Pause Until, If/Then/Else, While, Until and Repeat.
- Extensive conditional tests (Application, Window, Menu, Button, Modifiers, and many more).
- Permanently stored variables containing text, or arrays of numbers.
- Prompt For User Input action (with text, passwords, popup menus and checkboxes).
- Most action fields can contain calculations (e.g. Pause 60*Time in Minutes).
- Set Variable to Text action with tokens.
- Set Variable to Calculation action (with array access).
- Search & Replace Variable action (with regex and tokens).
- Use Variable action to adjust mouse, windows, applications, and more.
- Filter Variable action.

Enhanced Clipboards

- Added Quick Look to Clipboard History and Named Clipboard Switchers.
- Added Favorites to Clipboard History.
- Send Clipboard to other Macs (appears in clipboard history, ready for access).
- Delete Past Clipboard action.
- Search & Replace Clipboard action (with regex and tokens).

General Improvements

- Favorites category in action list.
- Configurable timeouts for individual actions.
- Macro Revisions.
- Applications listed in Status Menu.
- Use regular expressions in menu or button name matching by prefixing with ^.
- Typed String triggers can ignore case or remember case.
- Insert Text actions can honour case of remember case Typed String triggers.
- Allow triggering on "any" application activations, launches or quits.

New and Improved Actions

- Set Macro Enable action.
- Speak Text action.
- Play Sound action.
- Action to display text.
- Show/Hide the global macros palette action.
- Manipulate Window action can now operate on window with index N.
- Manipulate Window action can now center a window at any point.
- Generalized Search Web action to allow any URL.
- Added Relaunch & Force Quit options to Quit action.
- Added "Current Application" option to Quit action.
- Open Finder Selection action optionally with a specific application.
- Added "with specific application" to Open File or Folder action.
- Added "with specific application" to Open URL action.

New Text Tokens

- Token for variables.
- Token for Named Clipboards.
- Tokens for Current Track name, album, artist, ratingstars, etc.
- Tokens for Current Application, Current Mouse, and Front Window details.
- Tokens for Screen Frame and System Volume.
- Tokens for Address Book Name, Email, etc.
- Tokens for Executing Macro and Macro Group name.

Enhanced Scripting

- Variables can be accessed or written via AppleScript.
- Variables are available to Shell Scripts.
- Execute Script can store results in a variable.
- Support for AppleScript command to calculate expressions.
- Support for AppleScript command to process tokens.
- AppleScript command to reload the macros in the Keyboard Maestro Engine.
- AppleScript command to play sounds.

Enhanced Application & Window Switcher & Application Launcher

- Added new Application Launcher.
- Applications listed in Status Menu (Running, Recent, Applications, Utilities).
- Added Force Quit (press k twice).
- Added Get Info (press i).
- Added Reveal in Finder (press r).
- Added Relaunch (press l). E.g. press k k l to Force Quit & Relaunch.
- New icon badges.
- Mouse Rollover limited to when you change icons.
- Popup contextual menu in Application Switcher & Window Switcher.
- Option to disable mouse rollover in Application Switcher & Window Switcher.
- Control the opacity of Application Switcher & Window Switcher.
- Option to have Window Switcher sort by window depth.

- Allow switching through applications in Window Switcher.
- Added Cover Flow style to Application Switcher.
- Changed to using an alpha colour, removed the alpha slider, and now honours the brightness.

In Detail for 5.0:

- Command-Option-Control-Shift-Click on Status Menu icon to cancel all macros.
- Removed support for Zonic KRM - online purchases only now.
- Private preference (DisableAnimation) to disable all animations.
- Private preference (FavoritesDisplayName) to adjust spelling of Favorites.
- Support org.nspasteboard.TransientType (et al) and have the clipboard history ignore them.
- Support org.nspasteboard.AutoGeneratedType for Insert Text by Pasting.
- Named Clipboard popup New Clipboard leaves the user editing the name.
- Up/Down arrow moves the clipboard selection when clipboard switcher search field is selected.
- Enable/Disable button is properly disabled for Global Macro Group.
- Command-W closes palettes.
- The Recording window preserves its location.
- Import menu is properly grayed out only when it should be.
- Removed the Key Repeat preference from Application/Window Switcher.
- Multiple Quick Macros would conflict because UIDs were not unique.
- Searching for Cut, Copy or Paste in the Action list would not find the actions.
- The number pad keys can now select duplicate macros from the duplicate macro palette.

Download

Download Keyboard Maestro now to try all these great capabilities. Or keep reading for even more details about the Features of Keyboard Maestro.

Alternatively, you can contact us if you have a question about whether Keyboard Maestro can solve your automation needs. We want all our customers to be satisfied, so we are happy to help you understand how Keyboard Maestro can achieve your automation goals.

Features

Keyboard Maestro is a productivity enhancer with four main functions, allowing you to record and design your own macro shortcuts and activate them at any time, navigate through running applications with Application Switcher and open windows with Window Switcher, work with an unlimited number of clipboards using Clipboard Switcher and with clipboard history using Clipboard History Switcher.

Keyboard Maestro organizes your macros into Macro Groups. Think of them as folders of macros. Each Macro Group controls when the macros it contains are active. A Macro is made of two parts: a set of Triggers you choose to determine when the macro is executed and a list of Actions that define what the macro does when it is executed.

Macro Groups

- Create Macro Groups, which contain Macros and control when they are active.
- Macro Groups can be restricted to or excluded from specific applications.
- Macro Groups can be activated or deactivated with hot keys, or via the status menu or palette.
- Macro Groups can display a palette of contained macros.
- Macro Groups can be enabled or disabled.

Triggers

- Macros can be triggered by one or more Macro Triggers using any number of the following:
  - Hot Key - press, hold or release a key.
  - Typed String - type a string of keys.
  - Applications - on launch, quit, activate, deactivate, or periodically while an application is running or active.
  - System Wake - when the system wakes from sleep.
  - Login - when you log in to your Mac.
  - Time - at a particular time of day.
  - While Logged In - periodically while logged in.
  - Macro Palette - with a click on a context sensitive Macro Palette.
  - Status Menu - by selecting from a global system status menu.
  - Public Web - over the Internet, explicitly to the public, or via authenticated log in.
  - Device Key - when any HID device key is pressed, held down or released.
  - MIDI Note - when a MIDI note is pressed or released.
  - iPhone or iPod touch - from your iPhone or iPod touch.
  - By Script - from an AppleScript or a shell script.

Actions

- You can create Macro Actions manually or by recording them.
- Macros can execute a sequence of one or more Macro Actions including:
  - Application Control
    - Switch to Last, Next or a specific application.
    - Bring the current application's windows to the front.
    - Quit All, Other, the current or a specific application.
    - Hide All, Other, the current or a specific application.
    - Show All or a specific application.
  - Interface Control
    - Manipulate a window - resize, move, center, close, zoom, minimize, bring to front.
    - Move or click the mouse with modifiers.
    - Select a specific menu item.
    - Press a button with a specific name.
    - Simulate a keystroke.
    - Simulate moving the scroll wheel.
  - Control the System
    - Sleep, Restart, Shut Down, Fast User Switch or Log Out.
    - Activate screen saver.
    - Open/close the CD tray.
    - Increase, decrease, set or mute/unmute the system volume.
    - Increase or decrease the brightness.
  - Control iTunes
    - Play a specific a specific song, a random song or a specific Playlist.
    - Play, pause or stop.
    - Rewind or fast forward.
    - Go to the next or previous track.
    - Raise, lower or set the volume to a specific level
    - Set the rating of the current song
  - Control QuickTime Player

    - Play or pause the current movie.
    - Step backward or forward.
    - Increase, decrease or set the volume.
- Send a MIDI Note or Control Change.
- Open a file, folder, URL, the current Finder selection or System Preferences pane.
- Execute an AppleScript or shell script and optionally display or store the results.
- Execute an Automator workflow.
- Search the Web.
- Insert or Display Text by typing, pasting or with built-in token expansion (e.g., the current time).
- Control the Clipboard
    - Simulate Cut, Copy or Paste.
    - Copy, Cut or Paste to/from a permanent Named Clipboard.
    - Set the system clipboard to specific tokenized text, a past copy of the clipboard, or a Named Clipboard.
    - Save the system clipboard to a permanent Named Clipboard.
    - Delete a past clipboard entry from the clipboard history.
    - Filter the contents of the current clipboard.
    - Apply a BBEdit Text Factory to the current clipboard.
    - Search and Replace the current clipboard.
- Use Variables
    - Set a variable to specific tokenized text.
    - Set a variable to a calculation.
    - Search and Replace a variable.
    - Use a variable to set the mouse location, adjust window location or size, change the frontmost window or application by name, or set the system volume.
    - Filter variables in a variety of ways.
    - Prompt for variable values (with text, passwords, popup menus and checkboxes).
- Activate the Application Launcher or the Application, Window, Clipboard or Clipboard History switchers.
- Notifications
    - Via Growl.
    - By displaying text.
    - By speaking text, playing a sound or beeping.
    - With an alert.
- Control the Flow and Behavior of a Macro
    - Pause for a number of seconds (possibly calculated).
    - Pause Until conditions are met.
    - Execute a list of actions until conditions are met.
    - While conditions are met, execute a list of actions.
    - Repeat a list of actions N (may be a calculation) times.
    - If conditions are met, execute a list of actions, otherwise execute another list.
    - Execute another macro (like a subroutine).
    - Cancel All Macros or Cancel This Macro.
- Record a quick macro without launching Keyboard Maestro.
- Set (or toggle) whether a Macro Group or Macro is enabled.
- Show or Hide the Global Macro Palette.
- Comment (no action, just for helping you document a macro sequence).

Application Launcher

- Display a Cover Flow view of available applications for quick selection.
- Type ahead selection including abbreviations (eg "a p" for Adobe Photoshop).

Application and Window Switcher

- Optionally replace the system Command-Tab application switcher.
- Customize the switcher to match the look you want.
- Switch to any application or window with a keystroke.
- Switch to an application and hide all others.
- Easily select the exact application or window you want.
- Launch, hide, quit or force quit any application
- Close or minimize any window.
- Quit (or force quit) and relaunch applications.
- Get Info or reveal applications.
- Choose the application ordering you want: alphabetically, by last use, or by launch order.
- Sort windows alphabetically or by window depth order.
- Optionally hide other applications.
- Optionally always hide other applications.

Clipboard Switcher

- An unlimited number of Named Clipboards
- Copy, Cut or Paste to/from a Named Clipboards using a single keystroke.
- Clipboards are saved permanently.

Clipboard History Switcher

- Never lose your clipboard again.
- Browse your past clipboards and paste any previous clipboard item.
- Send clipboard entries to other Macs.
- Clipboards display rich text and images.
- Use Quick Look to view clipboard entries.
- Set clipboard entries as favorites so they are always available in your clipboard history.
- Clipboard History is optionally saved across logins and restarts.

Purchase

Keyboard Maestro is engineered by Stairways Software Pty Ltd and distributed by FastSpring. Keyboard Maestro is licensed on a per-user basis and individual users may use it on up to five Macs.

New customers can purchase Keyboard Maestro for US$36 by choosing Purchase Keyboard Maestro from the Keyboard Maestro menu. Customers with five or more users should contact us for a volume discount quote.

Keyboard Maestro 5 is a paid upgrade for most users of previous versions. Existing users are eligible for a discount.

Customers who purchased Keyboard Maestro after 1 March 2011 have been issued a free upgrade to Keyboard Maestro 5. If you have not received your free license, you can find your free license upgrade at http://enquiry.stairways.com/.

Customers who purchased Keyboard Maestro prior to March 2011 can upgrade to Keyboard Maestro 5 for US$18 until August 31 2011 and for US$25 after that date. If you have not received your instructions on how to upgrade, you can find details by looking up your Keyboard Maestro license at http://enquiry.stairways.com/.

Customers who have not disabled upgrade emails have been emailed with new license or upgrade instructions as appropriate. If you have not received this email, please contact us so we can resolve this promptly.

Keyboard Maestro may become available on the Mac App Store at some date, but that is at Apple's discretion. Please note that Apple does not currently have a mechanism allowing users to transfer a license bought online to or from the Mac App Store version, nor is there any way to receive the discounted price through the Mac App Store when upgrading a previous version of Keyboard Maestro.

It is our informal policy to have a paid major upgrade roughly once a year. This allows us to have a reasonably consistent revenue stream with which to fund development of Keyboard Maestro.

A fully-functional trial version of Keyboard Maestro is available for downloaded from http://download.stairways.com/.

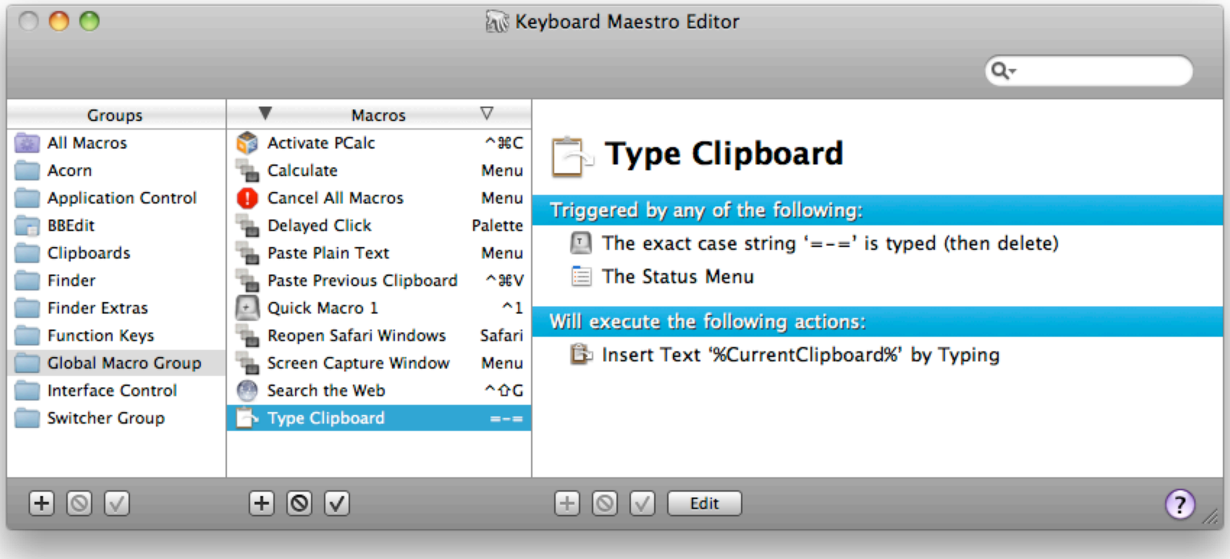
For sales enquires, customer service, technical support, or to contact project management, our current contact information is listed at http://contact.stairways.com/.

For more information about anything to do with Keyboard Maestro visit http://www.keyboardmaestro.com/.

Screenshots

Here is a quick taste of what Keyboard Maestro offers.

Macros Window



Keyboard Maestro Editor

Groups
All Macros
Acorn
Application Control
BBEdit
Clipboards
Finder
Finder Extras
Function Keys
Global Macro Group
Interface Control
Switcher Group

Macros	
Activate PCalc	^⌘C
Calculate	Menu
Cancel All Macros	Menu
Delayed Click	Palette
Paste Plain Text	Menu
Paste Previous Clipboard	^⌘V
Quick Macro 1	^1
Reopen Safari Windows	Safari
Screen Capture Window	Menu
Search the Web	^⇧G
Type Clipboard	=-=

Type Clipboard

Triggered by any of the following:

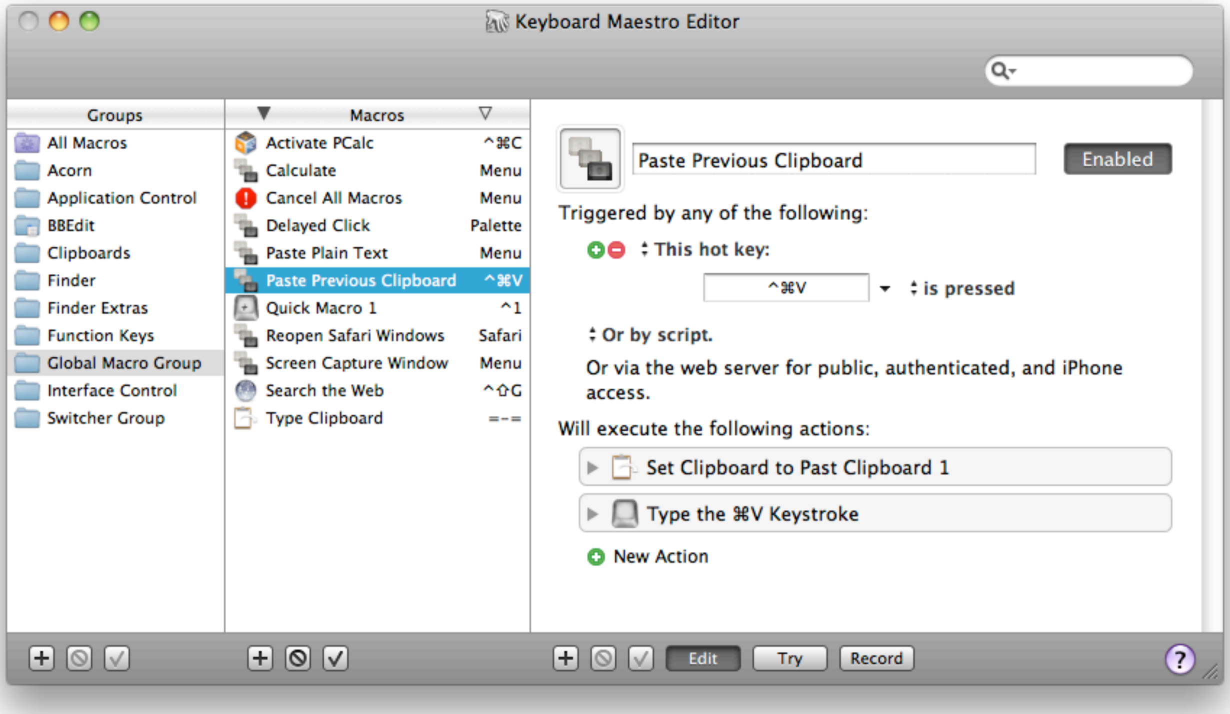
The exact case string '=-=' is typed (then delete)

The Status Menu

Will execute the following actions:

Insert Text '%CurrentClipboard%' by Typing

Macro Editor



Keyboard Maestro Editor

Groups
All Macros
Acorn
Application Control
BBEdit
Clipboards
Finder
Finder Extras
Function Keys
Global Macro Group
Interface Control
Switcher Group

Macros	
Activate PCalc	^⌘C
Calculate	Menu
Cancel All Macros	Menu
Delayed Click	Palette
Paste Plain Text	Menu
Paste Previous Clipboard	^⌘V
Quick Macro 1	^1
Reopen Safari Windows	Safari
Screen Capture Window	Menu
Search the Web	^⇧G
Type Clipboard	=-=

Paste Previous Clipboard Enabled

Triggered by any of the following:

This hot key: ^⌘V is pressed

Or by script.

Or via the web server for public, authenticated, and iPhone access.

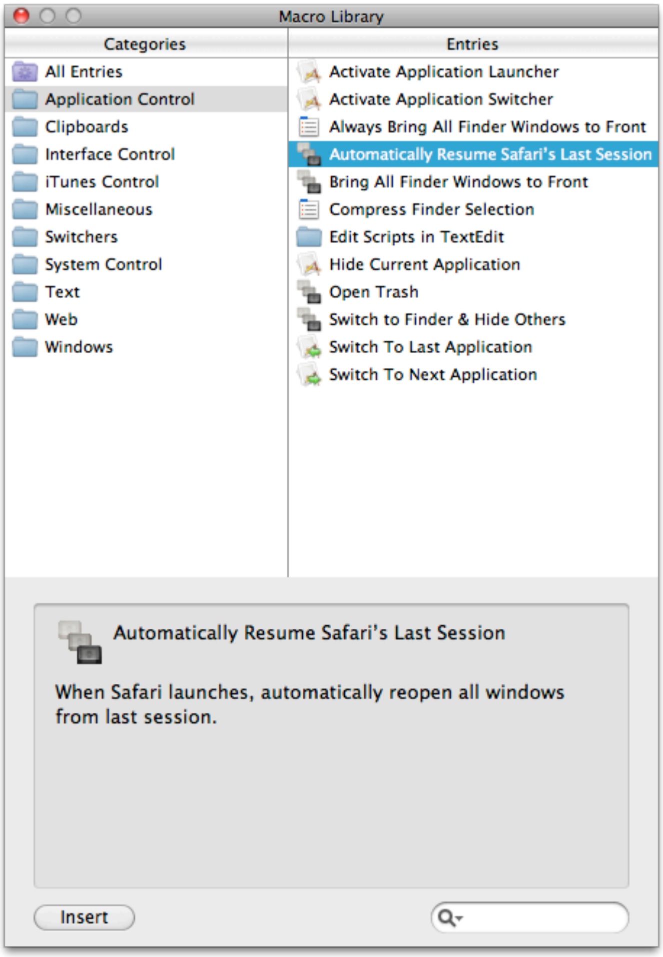
Will execute the following actions:

Set Clipboard to Past Clipboard 1

Type the ⌘V Keystroke

New Action

Macro Library



Categories

- All Entries
- Application Control
- Clipboards
- Interface Control
- iTunes Control
- Miscellaneous
- Switchers
- System Control
- Text
- Web
- Windows

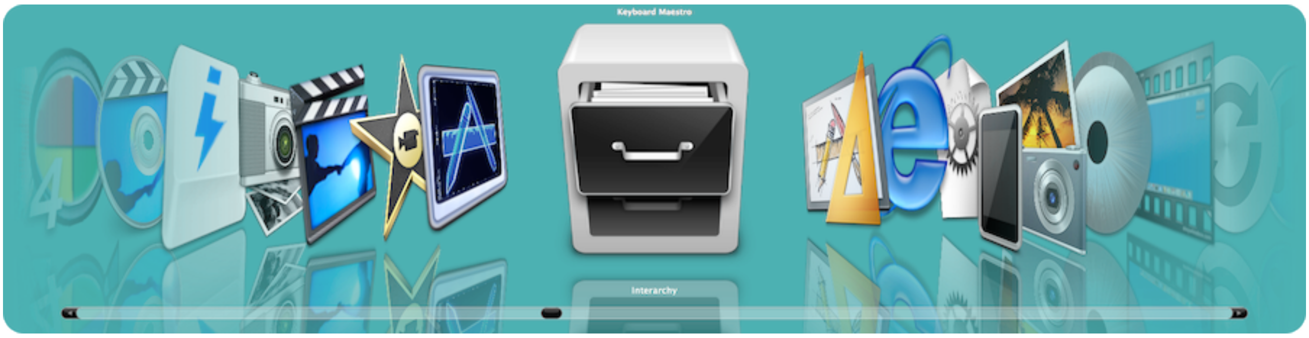
Entries

- Activate Application Launcher
- Activate Application Switcher
- Always Bring All Finder Windows to Front
- Automatically Resume Safari's Last Session
- Bring All Finder Windows to Front
- Compress Finder Selection
- Edit Scripts in TextEdit
- Hide Current Application
- Open Trash
- Switch to Finder & Hide Others
- Switch To Last Application
- Switch To Next Application

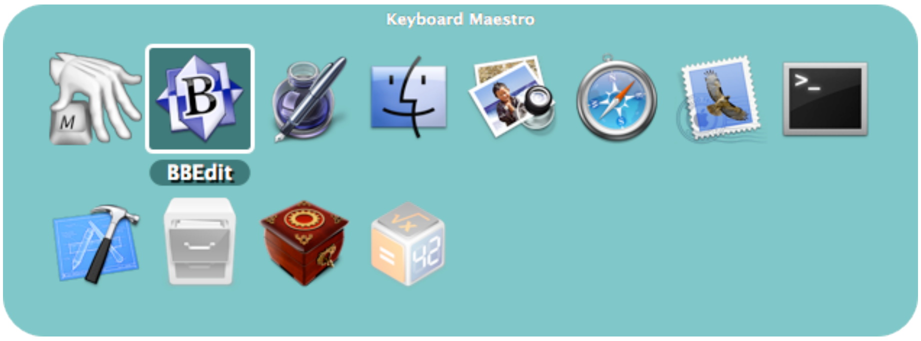
Automatically Resume Safari's Last Session

When Safari launches, automatically reopen all windows from last session.

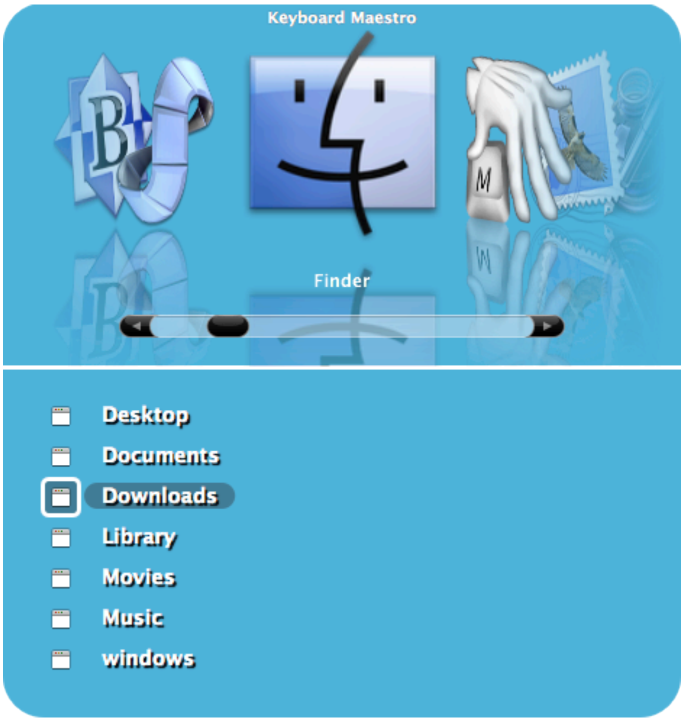
Application Launcher Window



Application Switcher Window

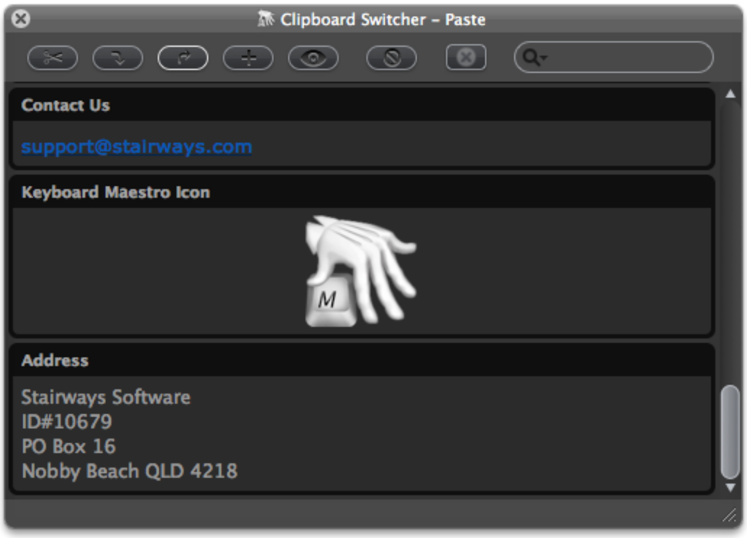


Window Switcher Window



Desktop
Documents
Downloads
Library
Movies
Music
windows

Clipboard Switcher Window



Contact Us

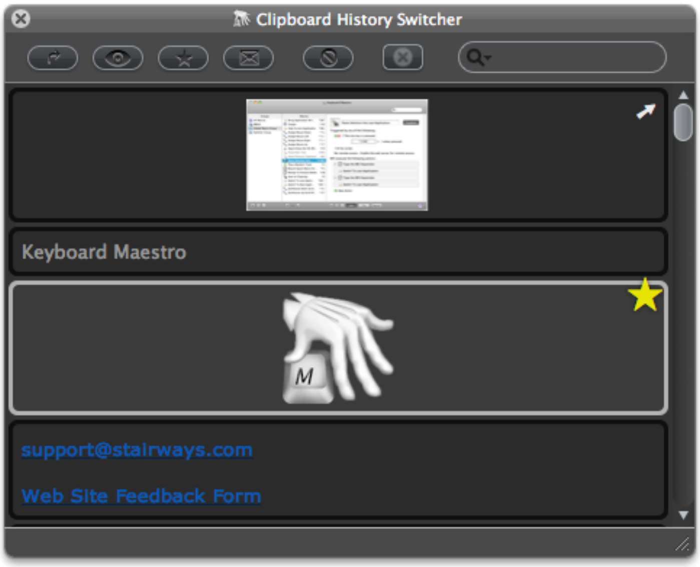
support@stairways.com

Keyboard Maestro Icon

Address

Stairways Software
ID#10679
PO Box 16
Nobby Beach QLD 4218

Clipboard History Switcher Window



Keyboard Maestro

support@stairways.com

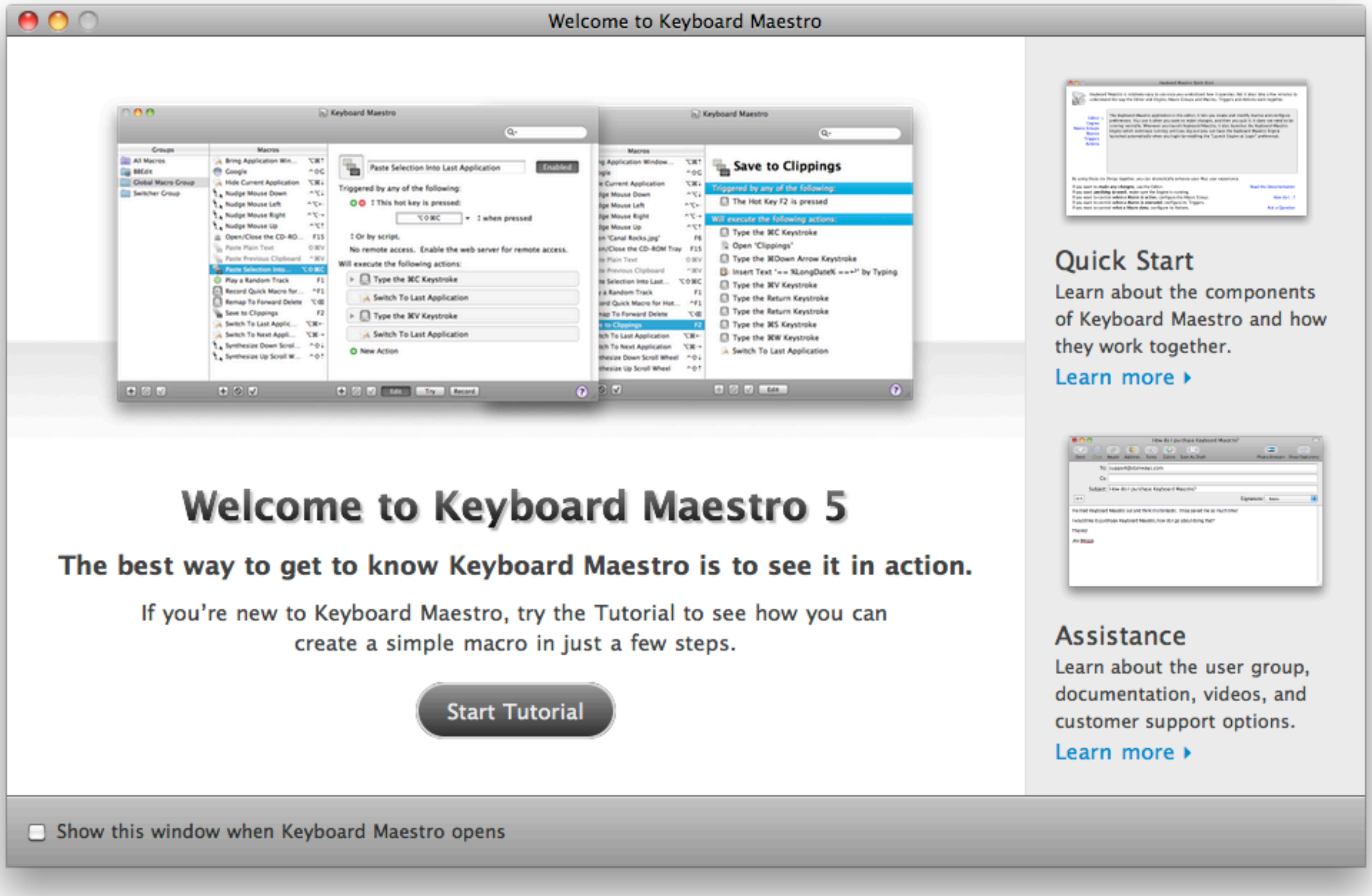
Web Site Feedback Form

Tour

This demonstration will begin to show you the power and versatility of Keyboard Maestro.

Getting Started

To start, launch Keyboard Maestro and it will display the Welcome window.



Welcome to Keyboard Maestro

Welcome to Keyboard Maestro 5

The best way to get to know Keyboard Maestro is to see it in action.

If you're new to Keyboard Maestro, try the Tutorial to see how you can create a simple macro in just a few steps.

Start Tutorial

Quick Start

Learn about the components of Keyboard Maestro and how they work together.

Learn more ▸

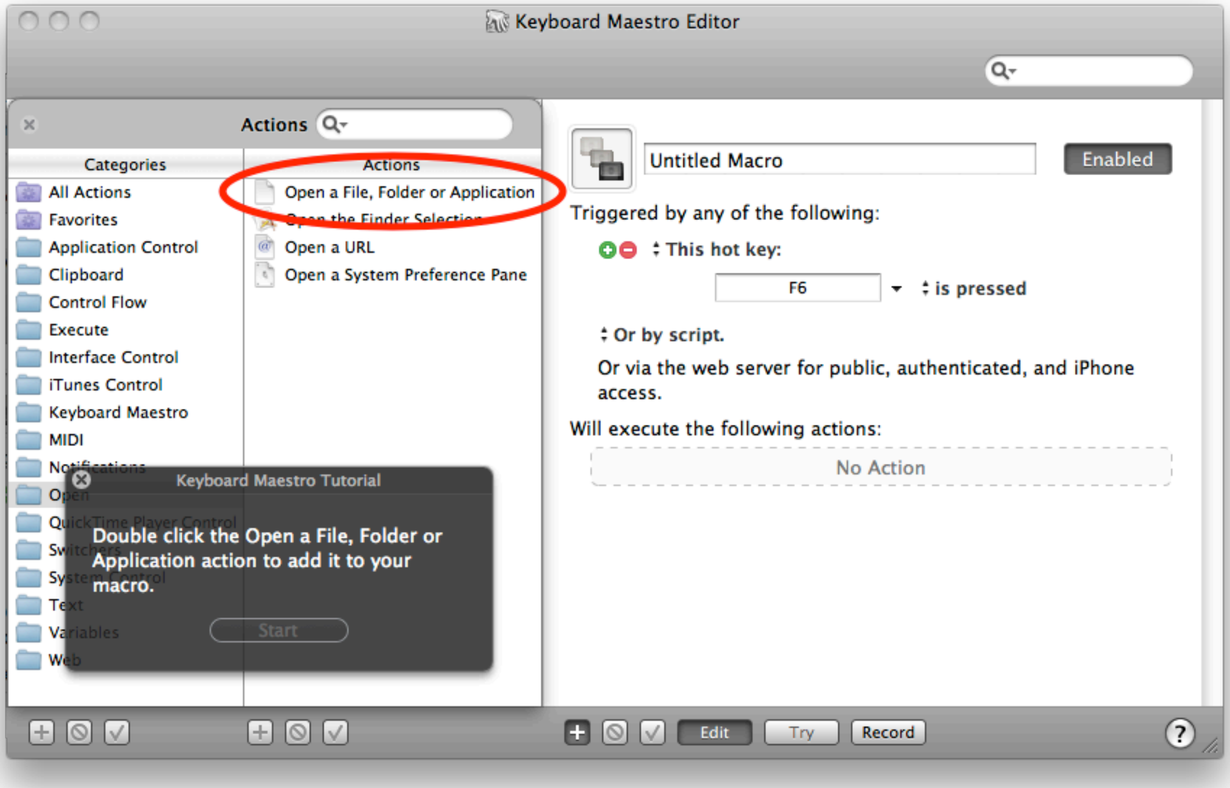
Assistance

Learn about the user group, documentation, videos, and customer support options.

Learn more ▸

☐ Show this window when Keyboard Maestro opens

If you are new to Keyboard Maestro, start the tutorial and Keyboard Maestro will show you how easy it is to create a macro.



Keyboard Maestro Editor

Keyboard Maestro Tutorial

Double click the Open a File, Folder or Application action to add it to your macro.

Untitled Macro — Enabled

Triggered by any of the following:

This hot key: F6 is pressed

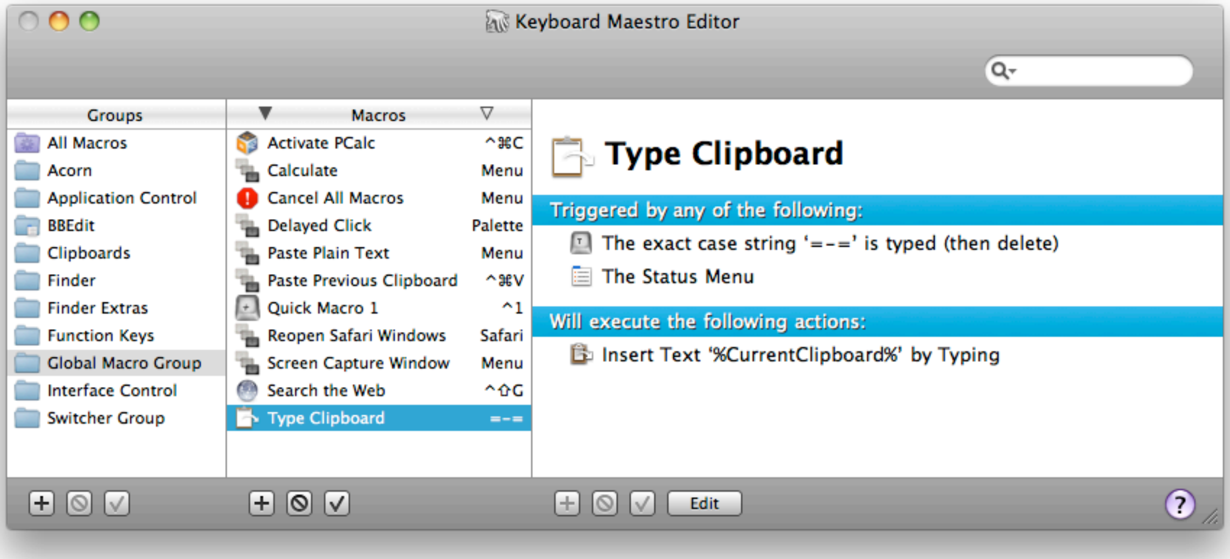
Or by script.

Or via the web server for public, authenticated, and iPhone access.

Will execute the following actions:

No Action

Close the Welcome window to display the Macros window.



Groups
All Macros
Acorn
Application Control
BBEdit
Clipboards
Finder
Finder Extras
Function Keys
Global Macro Group
Interface Control
Switcher Group

Macros	
Activate PCalc	^⌘C
Calculate	Menu
Cancel All Macros	Menu
Delayed Click	Palette
Paste Plain Text	Menu
Paste Previous Clipboard	^⌘V
Quick Macro 1	^1
Reopen Safari Windows	Safari
Screen Capture Window	Menu
Search the Web	^⇧G
Type Clipboard	=-=

Type Clipboard

Triggered by any of the following:

The exact case string '=-=' is typed (then delete)

The Status Menu

Will execute the following actions:

Insert Text '%CurrentClipboard%' by Typing

You can see some example Macros we have included for you.

Make New Macro

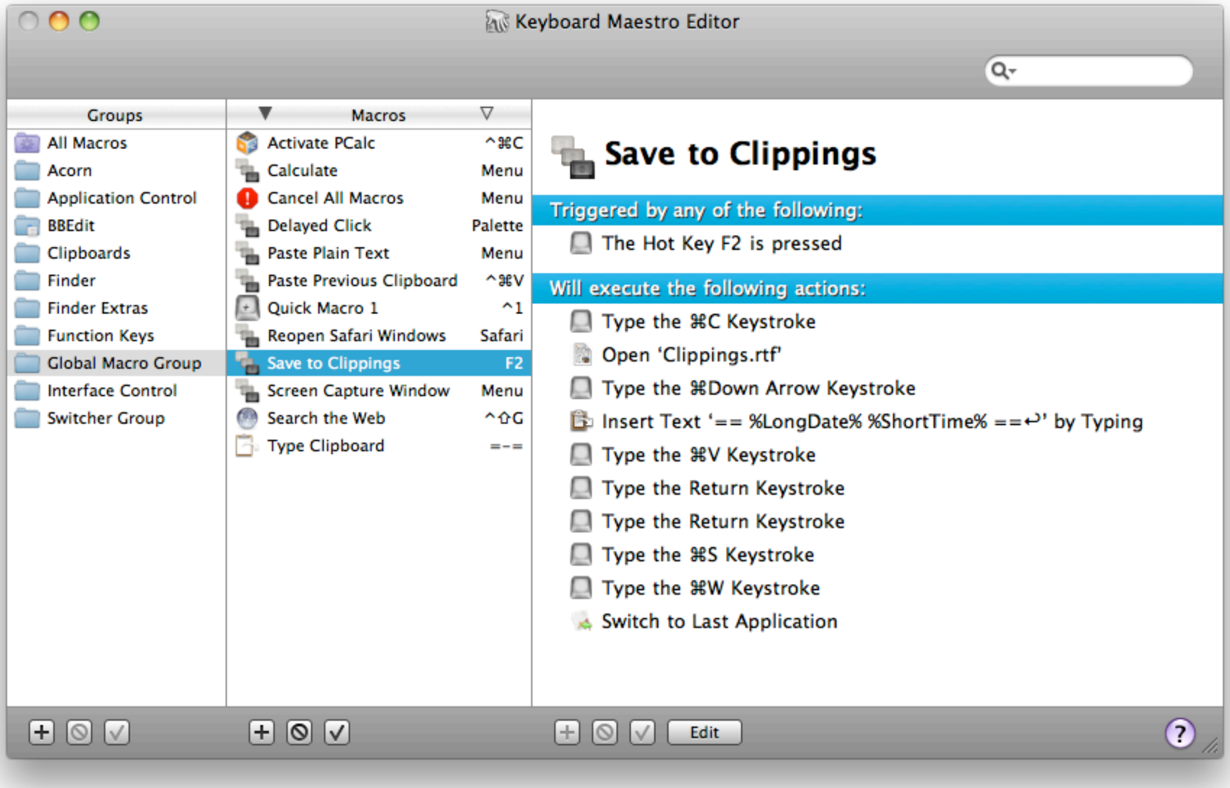
Click the + button under the Macro column to add a macro and display the Macro Editor window. We will now design a complex Macro enabling you to save clippings to a text file. First, launch TextEdit and create a new empty document. Save this blank document as Clippings.rtf in your Documents folder.

Normally, to add a selection to the Clippings.rtf file, you would have to do all this:

- Press Command-C to copy the selection in an application.
- Got to the Finder, open your Documents folder, then open the Clippings.rtf file.
- Go to the end of the file in TextEdit.
- Press the Return key and type a line of dashes and return to separate the clippings.
- Press Command-V to paste the selection you made before.
- Save the file and close the document.
- Switch back to the application where you originally selected some text or a picture.

That is all very tedious, and probably explains why most people never even bother!

So lets define the whole sequence as a Macro.



Groups
All Macros
Acorn
Application Control
BBEdit
Clipboards
Finder
Finder Extras
Function Keys
Global Macro Group
Interface Control
Switcher Group

Macros	
Activate PCalc	^⌘C
Calculate	Menu
Cancel All Macros	Menu
Delayed Click	Palette
Paste Plain Text	Menu
Paste Previous Clipboard	^⌘V
Quick Macro 1	^1
Reopen Safari Windows	Safari
Save to Clippings	F2
Screen Capture Window	Menu
Search the Web	^⇧G
Type Clipboard	=-=

Save to Clippings

Triggered by any of the following:

The Hot Key F2 is pressed

Will execute the following actions:

Type the ⌘C Keystroke

Open 'Clippings.rtf'

Type the ⌘Down Arrow Keystroke

Insert Text '== %LongDate% %ShortTime% ==↩' by Typing

Type the ⌘V Keystroke

Type the Return Keystroke

Type the Return Keystroke

Type the ⌘S Keystroke

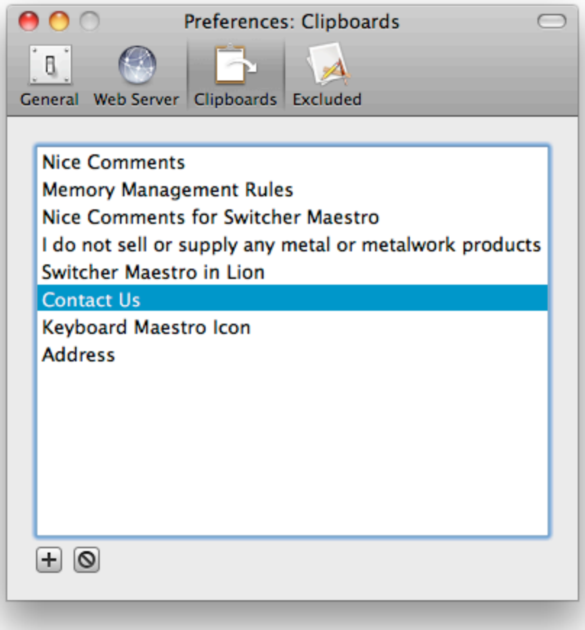
Type the ⌘W Keystroke

Switch to Last Application

Save the Macro and now any time you have some text you want to save, just select it and press F2. What used to be too much hassle to bother with is now done in seconds!

Named Clipboards

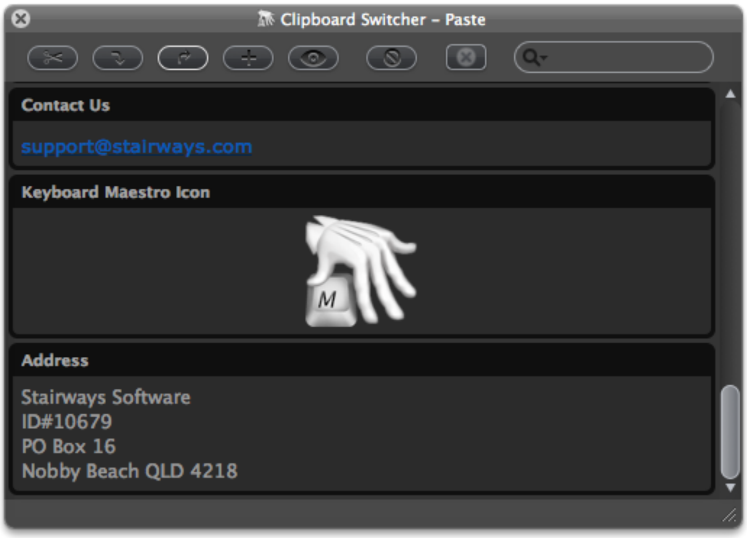
Keyboard Maestro lets you create as many named clipboards as you want using the Clipboards preference pane.



Nice Comments
Memory Management Rules
Nice Comments for Switcher Maestro
I do not sell or supply any metal or metalwork products
Switcher Maestro in Lion
Contact Us
Keyboard Maestro Icon
Address

Named clipboards let you save frequently used information, like your company logo, timely information like a customer's address, or your address so you never have to type it again. This allows you to paste the saved information whenever you want, wherever you want.

Then you can cut, copy or paste to/from the named clipboards using the defined Hot Keys (by default, Command-Shift-X, C and V respectively, but you can change them, too).



Contact Us

support@stairways.com

Keyboard Maestro Icon

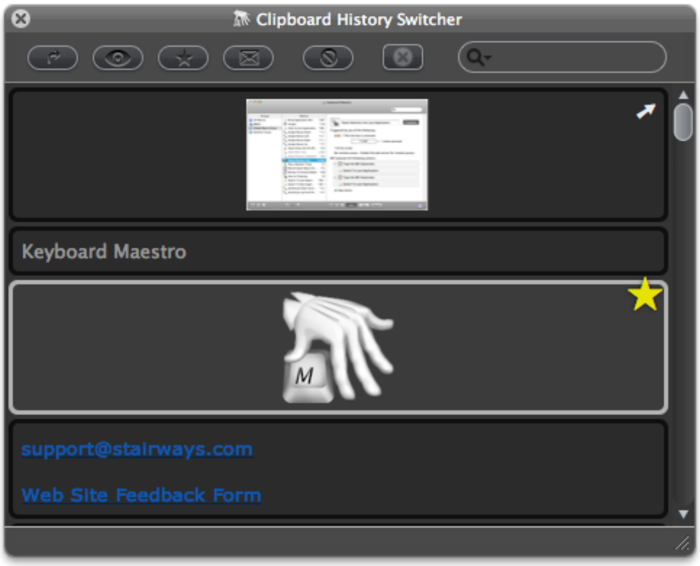
Address

Stairways Software
ID#10679
PO Box 16
Nobby Beach QLD 4218

If you have a large screen, or a specific job that needs it, you can have the window stay open (by toggling the Close After Action button) and copy or paste named clipboards easily.

Clipboard History

For less permanent information, Keyboard Maestro automatically remembers your clipboard history, saving a copy of each new clipboard item as you copy it. You can then paste any previous clipboard using the defined Hot Key (by default, Command-Control-Shift-V).



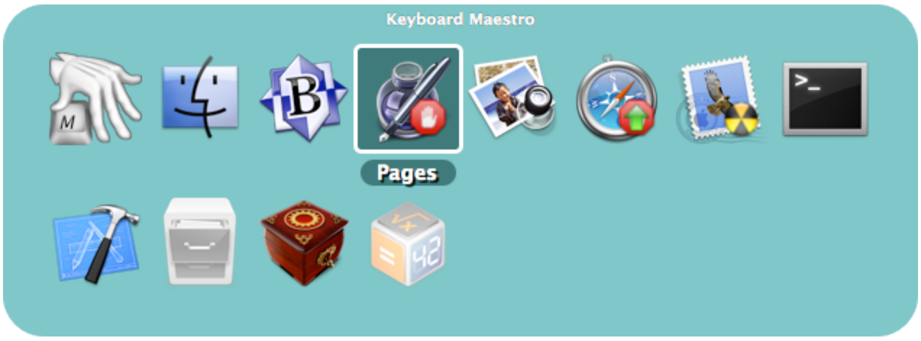
Another advantage of the Clipboard History is you can leave the window open (by toggling the Close After Action button) and paste previous clipboards by simply double clicking the desired item. This is particularly helpful when you need to copy many items in many places to be pasted into one application.

Keyboard Maestro can even save your clipboard history across restarts and log outs. Just enable the “Save Clipboard History” preference in the General preference pane.

Application Switcher

At the end of the Macro we defined, we used the Switch To Last Application action to switch back to the application you were using before. But most of us use a lot more than one application and we need to be able to switch between them, whether or not the application is already open. You can do more than just switch to between applications with Keyboard Maestro, it lets you define applications that always appear when you are switching so you can launch them quickly, and also Excluded Applications that you never want to see. As well, you can quit, hide or show applications. The Application Switcher is activated like any other macro, so you can find it in the Switcher Group in the Macros window. By triggering the Application Switcher macro (for example, by pressing Option-Tab) you can display the Application Switcher window.

As of Keyboard Maestro 5, the default setting is to use Command-Tab to replace the system switcher, but you can change this if you'd like.

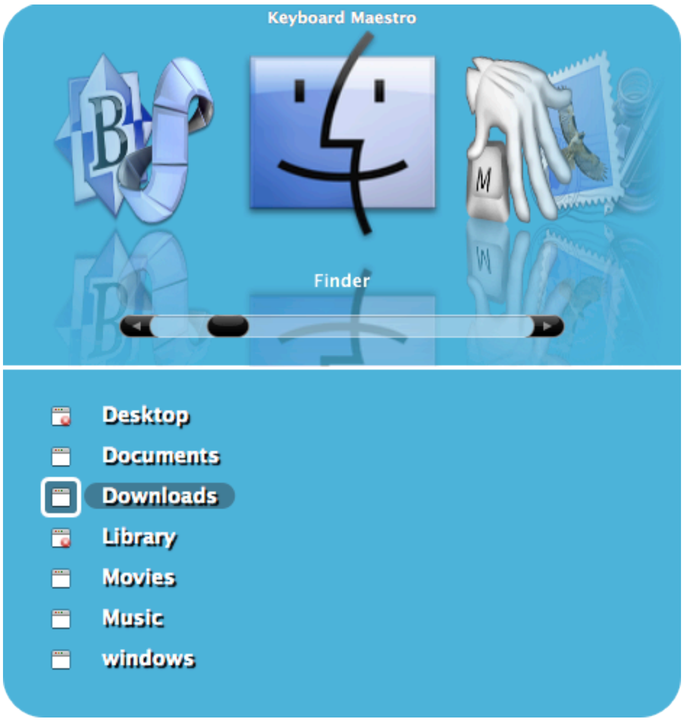


Note the applications set to be launched or quit.

If you want a specific keystroke to always launch a certain application, you can define a Macro to do that. For example, you could create a Macro with a Hot Key Trigger of F3 so it always activates Safari.

Window Switcher

Keyboard Maestro also includes a Window Switcher. By triggering the Window Switcher macro (for example, by pressing Control-Tab), the Window Switcher window appears showing all the windows of current application (in the current space).

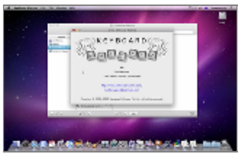


Note the windows set to be closed. This is a very useful way of cleaning up an application with a lot of open windows.

Conduct Your Mac Like a Pro!

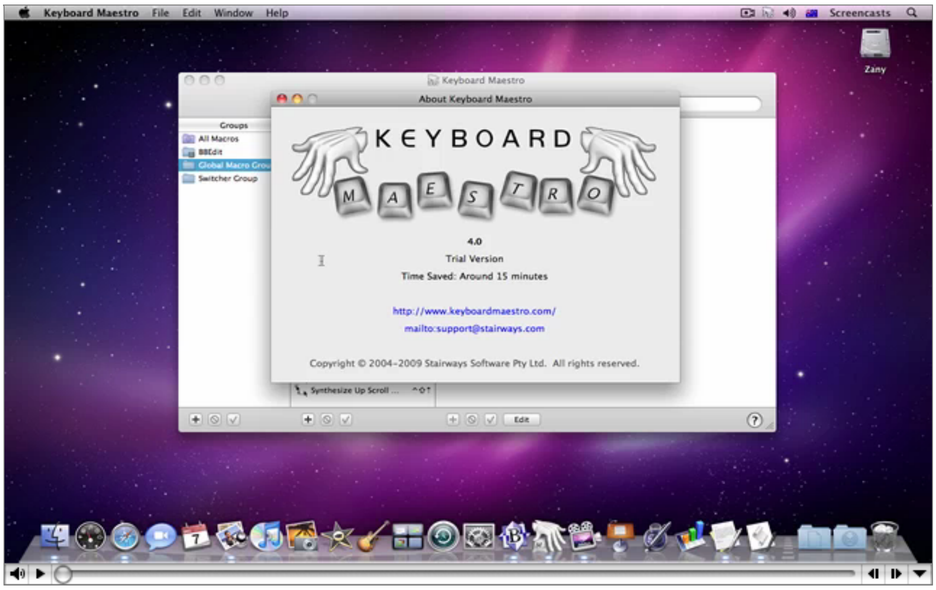
This is just a taste of all that Keyboard Maestro can do for you. It's time you started getting the most from your Mac? Download Keyboard Maestro today and you can be working faster and smarter in no time.

Movies


Overview

Configuration

Save to Clippings Tutorial



KEYBOARD MAESTRO

4.0

Trial Version

Time Saved: Around 15 minutes

http://www.keyboardmaestro.com/

mailto:support@stairways.com

Copyright © 2004-2009 Stairways Software Pty Ltd. All rights reserved.

Thanks to Dan Benjamin for doing the fantastic voice overs on these tutorial videos.

Links

Download Keyboard Maestro from http://download.stairways.com/.

Purchase Keyboard Maestro at http://purchase.stairways.com/.

Look up your current or previous license status and serial numbers, and get information about discounted upgrades from http://enquiry.stairways.com/.

Join the Keyboard Maestro User Group online community consisting of the developers and Keyboard Maestro users at http://user.keyboardmaestro.com/.

Documentation describing Keyboard Maestro is available at http://documentation.keyboardmaestro.com/.

For sales enquires, customer service, technical support, or to contact project management, our current contact information is listed at http://contact.stairways.com/.

For more information about anything to do with Keyboard Maestro visit http://www.keyboardmaestro.com/.

Quick Start

Keyboard Maestro is easy to use once you understand the way the Editor and Engine, Macro Groups and Macros, Triggers and Actions work together.

The Keyboard Maestro application is the editor. It lets you create and modify macros and configure preferences. You use it only when you want to make changes and then you quit it. It does not always need to be running. Whenever you launch Keyboard Maestro, it also launches the Keyboard Maestro Engine which continues running until you log out (you can have the Keyboard Maestro Engine launched automatically when you login by enabling the "Launch Engine at Login" preference in the General preference pane).

The Keyboard Maestro Engine is a background only application that enables all of Keyboard Maestro's features. It responds to your Hot Key presses, watches the time, tracks applications and maintains your clipboard history, handles remote web, iPhone requests and receiving clipboards, and, of course, executes your Macro Actions. It should be running at all times, so we recommend you enable the "Launch Engine at Login" preference in the General preference pane.

Keyboard Maestro organizes your macros into Macro Groups which are like folders of macros. Each Macro Group controls when the macros it contains are active. A Macro Group can target or exclude specific applications, which means the macros it contains will only be active in those desired applications. For example, you can have macros which are active only in Mail.app.

A Macro Group can also act as a container for specific-use macros which are enabled only after a Hot Key press or which are displayed as a palette of the macros. For example, you could create a Macro Group containing macros that resized or repositioned windows using the arrow keys, but those macros would only be active after the Hot Key was pressed so that the arrow keys could be used normally at other times.

You create a Macro Group by clicking the + button at the bottom of the Macro Groups list. You can disable or enable Macro Groups by clicking the ✓ button. You can configure a Macro Group by selecting it and clicking the Edit button.

Keyboard Maestro's main purpose is to execute Macros. A Macro lives in a Macro Group and consists of a set of Triggers that determine when the macro is executed, together with a list of Actions that define what the macro does when it is executed. You create a Macro by clicking the + button at the bottom of the Macros list. You can disable or enable Macro Groups by clicking the ✓ button. Keep in mind that a Macro can only be active when the Macro Group that contains it is active. You can edit a Macro by selecting it and clicking the Edit button.

A Trigger defines when a macro will be executed. There are a variety of Triggers available, the most common is the Hot Key trigger which executes the macro when a specified Hot Key is pressed. Similarly, you can use a typed trigger to execute a macro when you type some text (for example =addr=). Another common trigger is the Macro Palette which lets you trigger a macro by clicking on a context (front application) sensitive floating palette of macros. The Status Menu trigger displays the macro in the Status Menu. You can also trigger a macro when you login or when your Mac wakes from sleep, at a specific time or on a specific day, when an application launches, activates or quits, by executing a script, or remotely using a web browser or iPhone. A Trigger will only execute the macro if the Macro Group and Macro are enabled and currently active. You create Triggers by creating or editing a Macro and clicking the + button at the bottom of the macro detail view.

When a Macro is Triggered it executes a list of Actions. Keyboard Maestro performs each of the Actions in order. There are a wide variety of Actions allowing you to control applications, simulate user interface events like key presses, mouse clicks and menu selections, open files, control your Mac or the clipboard, or display a variety of powerful switchers (Application, Window, Clipboard and Clipboard History Switchers). You can also execute a script (AppleScript, Shell Script or Automator Workflow). You create Actions by creating or editing a Macro and clicking the + button to display available actions or by clicking on the Record button and performing the action while Keyboard Maestro records your actions to your Macro.

By using these six things (Editor and Engine, Macro Groups and Macros, Triggers and Actions) together, you can dramatically enhance your Mac experience.

- If you want to **make changes**, use the Editor.
- If you want anything to work, make sure the Engine is running.
- If you want to control **when a Macro is active**, configure the Macro Group.
- If you want to control **when a Macro is executed**, configure its Triggers.
- If you want to control **what a Macro does**, configure its Actions.

How do I …?

How do I install Keyboard Maestro?

To install Keyboard Maestro, simply copy it to your Mac's Applications folder (or anywhere you like).

When you launch Keyboard Maestro it launches an invisible "Keyboard Maestro Engine" that continues to run even after you quit Keyboard Maestro. The engine is the process that enables your Macros, Application Switcher, and Clipboard Switcher to work. This means that they will continue to work after you quit Keyboard Maestro, as long as the engine is still running.

You can quit or launch the engine manually using the File menu.

You should consider turning on the "Launch Engine at Login" preference in the General preference pane to ensure all of Keyboard Maestro's facilities are available to you as soon as you login or startup your Mac.

How do I upgrade Keyboard Maestro?

Keyboard Maestro includes an automatic upgrade mechanism, so to upgrade Keyboard Maestro once you are running version 3.0 or later, simply click the Install Update button when prompted.

To upgrade Keyboard Maestro manually, or from any version before 3.0, simply replace the Keyboard Maestro application in your Applications folder with the new one. Because the Keyboard Maestro Engine is probably running, the Finder may not let you replace the existing application with the new version, so the simplest process is to launch the new version (which quits any old engine and starts the new one), then choose Quit Engine command in the File menu and quit Keyboard Maestro. Now move the new version in to the Applications folder, possibly replacing the old one. Finally, launch Keyboard Maestro to restart the engine.

Keyboard Maestro will automatically import your version 3 or 4 macros, clipboards and preferences. Your old macros will be saved in the ~/Library/Preferences/Keyboard Maestro/Keyboard Maestro Macros Saved Version 4.plist in case you decide not to upgrade to version 5 for any reason.

If you are upgrading directly from a much older version, you will get better results by upgrading to the last of each major version in turn, ie, run 2.1.3, then 3.5, then 4.3.2, then the current version.

If you have not done so already, you should consider turning on the "Launch Engine at Login" preference in the General preference pane to ensure all of Keyboard Maestro's facilities are available to you as soon as you login or startup your Mac.

How do I purchase Keyboard Maestro?

New customers can purchase Keyboard Maestro for US$36 by choosing Purchase Keyboard Maestro from the Keyboard Maestro menu. Customers with five or more users should contact us for a volume discount quote.

You can look up your current or previous license status and serial numbers, and get information about discounted upgrades from http://enquiry.stairways.com/.

Thanks for supporting us and enabling us to continue work on Keyboard Maestro.

See also the Purchase section.

How do I purchase Keyboard Maestro from the Mac App Store?

We will submit Keyboard Maestro to the Mac App Store, but when or if it appears there is up to Apple and largely beyond our control. When (or if) it does, you can purchase it as normal from the Mac App Store.

See also the Purchase section.

How do I register Keyboard Maestro?

When you purchase Keyboard Maestro you will be given a serial number, and will also promptly be emailed your username (email address) and serial number in the “Thanks For Your Purchase” email. Although you can retrieve this information from us at any time in the future, it is a good idea to keep this safe.

If you do not receive your serial number promptly after purchasing, it may be that the email has not reached you, possibly due to spam filtering on your email service. In this case, try looking up your purchase at http://enquiry.stairways.com/ (although that will email you your serial number which might again be lost to over-zealous spam filters).

Once you have your username (email address) and serial number, launch Keyboard Maestro and either immediately click the Use Existing License button or choose Register Keyboard Maestro from the Keyboard Maestro menu. Now enter the username (email address) exactly as shown and the serial number exactly as shown and click the OK button. If you have any problems, recheck that the email address and serial number you are entering is exactly as shown (the serial number’s email address does not change even if you have changed your email address with us) and also that your license matches the major version number (eg, a version 5 license will work with version 5.x of Keyboard Maestro).

How do I get started?

The first thing to do is to read the Quick Start. You may also want to subscribe to our Getting Started emails (Keyboard Maestro will ask you to subscribe).

After that, use your Mac normally and keep an eye out for things you do repetitively. Things like:

- launch or switch to a particular application.
- open a particular document.
- type a specific string of text (eg your name, address, etc).

When you notice something, consider making a Macro to do it and assigning it to a Hot Key or a Macro Palette or Status Menu trigger.

Try to be consistent with your Hot Keys, for example you might have a set of applications you open, using a function key for each, and a set of documents you open, using a Control-Function Key combination for each of them, and a bunch of snippets of text your type, using Control-Letter for each of them (the letter could be a mnemonic to help you remember which one is which). Consider putting a sticker on your keyboard across the top of your function keys to help you remember which ones do which function. Remember that you can use the number pad keys as well.

Also, keep in mind common command keys and system defined hot keys and try to avoid conflicting with them.

See also the Tips section.

How do I create a new Macro?

To create a Macro, launch Keyboard Maestro and click the + button under the Macros list. Give the Macro a name, add one or more triggers, and one or more actions. The Macro is immediately active.

An easy way to generate macro actions is to turn on recording and proceed to show Keyboard Maestro what you want to do. Then turn recording back off and look through the actions — chances are you will want to delete or adjust some of the recorded actions to make a robust macro, but this will be much quicker than creating each macro manually.

How do I Insert Styled/Colored Text or Images?

You can insert plain text using the Text -> Insert Text action. If you want to insert styled text or images you can do so using a Named Clipboard. Here is how to create a Macro to insert styled text when you press a Hot Key (the same method would work for an image with minor changes).

- Create a macro (see the How do I create a new Macro? section) with the action Clipboard -> Use Named Clipboard set to Paste from the named clipboard “My Styled Text”.
- Double check the Activate Switcher macro group contains the Clipboard Switcher Copy macro with Command-Shift-C.
- Switch to your word processor and create the styled text if necessary. Then select the styled text and press Command-Shift-C. The Clipboard Switcher window will appear. Double-click the “My Styled Text” entry to copy your selection permanently in to that named clipboard.

Now whenever you trigger the macro you just created, your styled text or image will be pasted in.

How do I Insert the Current Date?

You can create an action Clipboard -> Insert Text which includes tokens. There are some basic date format tokens, or you can use any ICU data format.

How do I configure the application switcher?

The Application Switcher (and all the switchers) are activated by macro actions. By default, Keyboard Maestro creates a “Switcher Group” Macro Group, and in there creates six macros. Each macro has a hot key trigger and a matching action which activates the appropriate switcher.

So to configure the Application Switcher, launch Keyboard Maestro, select the Switcher Group, and double click the Activate Application Switcher macro. You can then configure the various Application Switcher parameters, such as style and icon size by configuring the Application Switcher action. You can also configure the hot key used to activate the switcher, or disable the switcher.

How do I uninstall Keyboard Maestro?

Launch Keyboard Maestro and ensure the “Launch Engine at Login” preference in the General preference pane is turned off. Then choose Quit Engine from the File menu to quit the engine, and then choose Quit Keyboard Maestro from the Keyboard Maestro menu to quit the application. You can then trash the Keyboard Maestro application from your Applications folder.

How do I revert to a previous version of Keyboard Maestro?

Launch Keyboard Maestro and ensure the “Launch Engine at Login” preference in the General preference pane is turned off. Then choose Quit Engine from the File menu to quit the engine, and then choose Quit Keyboard Maestro from the Keyboard Maestro menu to quit the application. You can then trash the Keyboard Maestro application from your Applications folder.

For version 2, open the ~/Library/Preferences folder and the folder ~/Library/Preferences/Keyboard Maestro/Saved Version 2 folder. Move the files from the latter folder into the former folder. Trash the ~/Library/Preferences/Keyboard Maestro folder. Download Keyboard Maestro 2.1.3, if necessary, from http://files.stairways.com/keyboardmaestro/keyboardmaestro-213.dmg. Move Keyboard Maestro 2 to the Applications folder and launch it. Turn on the "Launch Engine at Login" preference if desired.

For version 3, open the ~/Library/Preferences/Keyboard Maestro folder. Trash the Keyboard Maestro Macros.plist and replace it with the Keyboard Maestro Macros Saved Version 3.plist. Download Keyboard Maestro 3.5, if necessary, from http://files.stairways.com/keyboardmaestro/keyboardmaestro-35.zip. Move Keyboard Maestro 3 to the Applications folder and launch it. Turn on the "Launch Engine at Login" preference if desired.

For version 4, open the ~/Library/Preferences/Keyboard Maestro folder. Trash the Keyboard Maestro Macros.plist and replace it with the Keyboard Maestro Macros Saved Version 4.plist. Download Keyboard Maestro 4.3.2, if necessary, from http://files.stairways.com/keyboardmaestro/keyboardmaestro-432.zip. Move Keyboard Maestro 4 to the Applications folder and launch it. Turn on the "Launch Engine at Login" preference if desired.

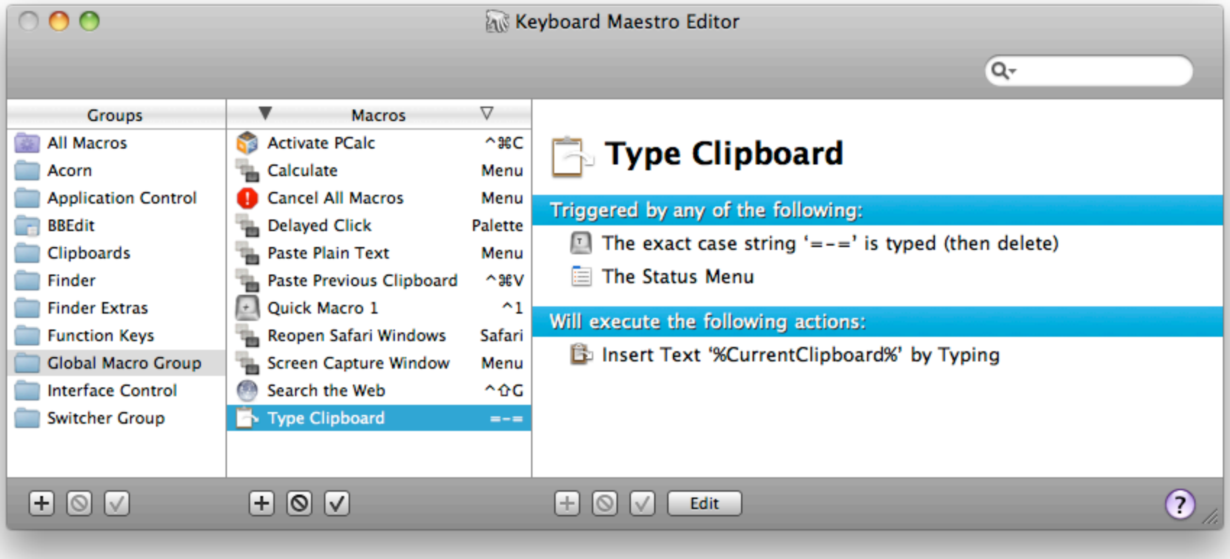
Macro Groups

Keyboard Maestro organizes your macros into Macro Groups which are like folders of macros. Each Macro Group contains a number of macros and controls when those macros are active.

A Macro Group can target or exclude specific applications, which means the macros it contains will only be active in those desired applications. For example, you can have macros which are active only in Mail.app.

A Macro Group can also act as a container for specific-use macros which are enabled only after a Hot Key press or which are displayed as a palette of the macros. For example, you could create a Macro Group containing macros that resized or repositioned windows using the arrow keys, but those macros would only be active after the Hot Key was pressed so that the arrow keys could be used normally at other times.

To create a new Macro Group, first launch Keyboard Maestro.



Groups	Macros	
All Macros	Activate PCalc	^⌘C
Acorn	Calculate	Menu
Application Control	Cancel All Macros	Menu
BBEdit	Delayed Click	Palette
Clipboards	Paste Plain Text	Menu
Finder	Paste Previous Clipboard	^⌘V
Finder Extras	Quick Macro 1	^1
Function Keys	Reopen Safari Windows	Safari
Global Macro Group	Screen Capture Window	Menu
Interface Control	Search the Web	^⇧G
Switcher Group	Type Clipboard	=-=

Type Clipboard

Triggered by any of the following:

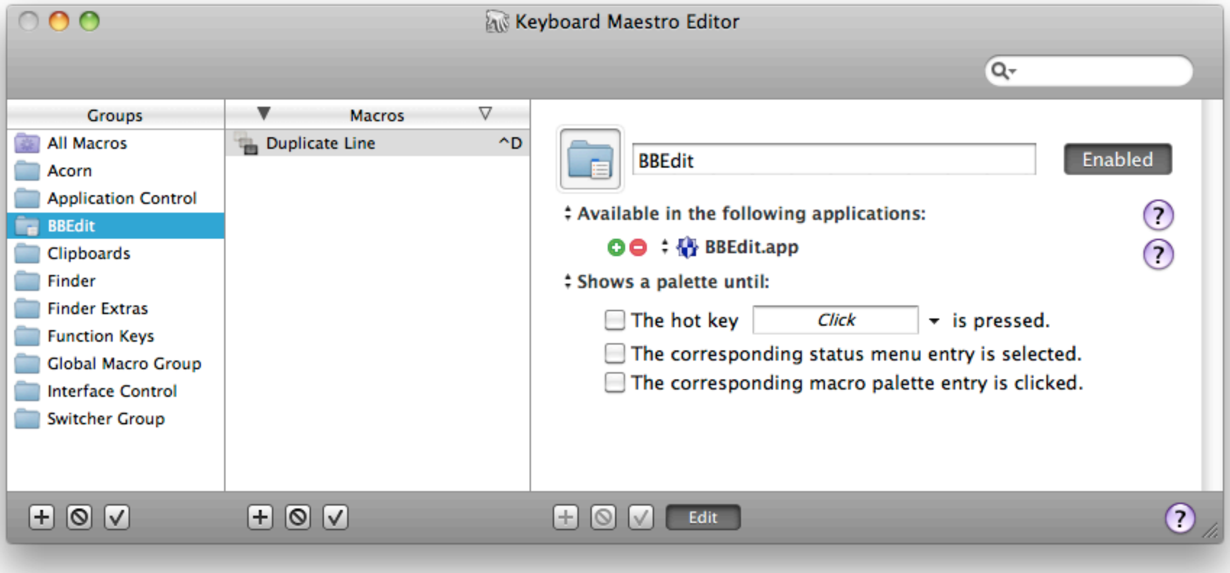
The exact case string '=-=' is typed (then delete)

The Status Menu

Will execute the following actions:

Insert Text '%CurrentClipboard%' by Typing

Now click the + button below the Macro Groups list.



Groups	Macros	
All Macros	Duplicate Line	^D
Acorn		
Application Control		
BBEdit		
Clipboards		
Finder		
Finder Extras		
Function Keys		
Global Macro Group		
Interface Control		
Switcher Group		

BBEdit

Available in the following applications:

BBEdit.app

Shows a palette until:

□ The hot key Click is pressed.

□ The corresponding status menu entry is selected.

□ The corresponding macro palette entry is clicked.

Enter the name of your new Macro Group.

You can choose to target the macros in your Macro Group at specific applications.

By default, Macro Groups and their Macros are available in all applications, that is they are always ready to be triggered. These are especially useful for Macros that give you instant access to applications or documents, or type in globally applicable text. For example you might have a Macro to launch your email client or word processor, a Macro to open your financial accounts, and a macro to type your name or email address.

To have macros only active in specific applications, set the Macro Group to be “Available in the following applications” and add the desired applications to the list.

For example, you could have macros targeted at:

- Mail.app that inserts common text messages.
- BBEdit and Xcode that inserts code chunks or duplicate lines or add #include headers.
- Safari that configure windows or enter information.
- Photoshop or Acorn that arrange items or script guides.

To have macros active everywhere **except** specific applications, set the Macro Group to be “Available except in the following applications” and add the desired applications to the list. For example, you could exclude macros from triggering in games. Also, if you have an application that uses lots of function keys for crucial tasks, you could exclude that application to allow you to use the functions keys for macros elsewhere without conflicting with that application.

You can also choose to activate the macros manually (either for a single execution, or toggle the macro group on and off) or display the macros in a floating macro palette. The options are:

- Always activated.
- Activated for one action when:
- Activated/deactivated when:
- Shows a palette for one action when:
- Shows/hides a palette when:
- Shows a palette until:

To have the macros in a Macro Group always ready to be triggered, set the Macro Group to be “Always activated”.

To have macros that are active only immediately after you trigger the Macro Group, set the Macro Group to be activated “Activated for one action when”. The macros in the Macro Group will be enabled when you trigger the Macro Group and will remain enabled until either any macro is triggered or you press any other key. You could use this to create a set of related actions with easily remembered hot keys that will not conflict with normal use because they are not activated until you trigger the group. For example, you could have a group of macros to launch various applications so that Command-Control-L activated the group, and then a single letter press launched the application (eg M for Mail, S for Safari, F for Finder).

As with each of the following options, you can trigger the macro group by pressing a Hot Key, by selecting from the Status Menu, or by clicking on the global floating palette.

To have Macros that are active after you trigger the Macro Group, set the Macro Group to be “Activated/deactivated when”. The Macros in the Macro Group will be enabled when you trigger the Macro Group and will remain enabled until you trigger the Macro Group again. You could use this to create a set of related actions with easily remembered hot keys that will not conflict with normal use because they are not activated until you press the group Hot Key. For example, you could have a group of macros to move and resize windows and have Command-Control-W activate the group. Then a single arrow key press moves the front window. When the window is positioned, press Command-Control-W a second time to disable the macros.

To have Macros that are active and displayed only immediately after you trigger the Macro Group, set the Macro Group to “Shows a palette for one action when”. The Macros in the Macro Group will be displayed in a floating Macro Palette and enabled when you trigger the Macro Group and will remain displayed until either any macro is triggered or you press any other key. You could use this to create a set of related actions that do not even need a Hot Key. For example, you could have a group of macros to launch various applications like Mail, Safari and the Finder so when you press Command-Control-L, a palette of these macros is displayed and a single click will launch the application.

To have Macros that are active and displayed after you trigger the Macro Group, set the Macro Group to “Shows/hides a palette when”. The Macros in the Macro Group will be displayed in a floating Macro Palette and enabled when you trigger the Macro Group and will remain displayed until you trigger the Macro Group again. You could use this to create a set of related actions that do not even need a Hot Key. For example, you could have a group of macros to align objects in a CAD application, so when you press Command-Control-A, a palette of these macros is displayed and you can click various alignment options (distribute left-right, align top edges) and then close the palette by pressing Command-Control-A a second time.

To have Macros that are always active and displayed in a palette, set the Macro Group to “Shows a palette until”. The Macros in the Macro Group will be displayed in a floating Macro Palette and enabled. The palette will close when (if) you trigger the Macro Group and will remain closed (and the macros disabled) until you trigger the Macro Group again. You could use this to create a set of related actions that do not even need a Hot Key and that are available in a Macro Palette. You could make the Macro Group available only in a specific application so that it appears only in that application. For example, you could have a group of macros to align objects in a CAD application, and have the Macro Group available (and hence the palette displayed) only in the CAD application.

Even if the macros are displayed in a Palette, the macros can still have Hot Key (or any other kind of) triggers which will be available whenever the macro group is active (in this case, whenever the palette is displayed).

You can disable or enable a macro group by selecting it and clicking the ✓ button at the bottom of the Groups list. If a Macro Group is disabled, all its contained Macros will be disabled. You can disable or enable a Macro Group using the Set Macro Enable action, or from AppleScript with:

```
tell application "Keyboard Maestro"
  setMacroEnable "Macro Group Name or UID" with/without enable
end tell
```

You can change the targeting of an existing group by selecting it in the Groups list and clicking the Edit button or by double clicking it.

You can start editing a Macro or Macro Group from AppleScript with:

```
tell application "Keyboard Maestro"
  editMacro "Macro Group Name or UID"
end tell
```

You can get the list of currently selected macros or macro groups with:

```
tell application "Keyboard Maestro"
  selectedMacro
end tell
```

To delete a Macro Group, select the macro group and then press the Delete key or click the ⊘ button at the bottom of the Groups list.

You cannot delete or rename or disable the Global Macro Group - if you want to control when some of the macros within the Global Macro Group are active, make a new macro group and drag those macros to it.

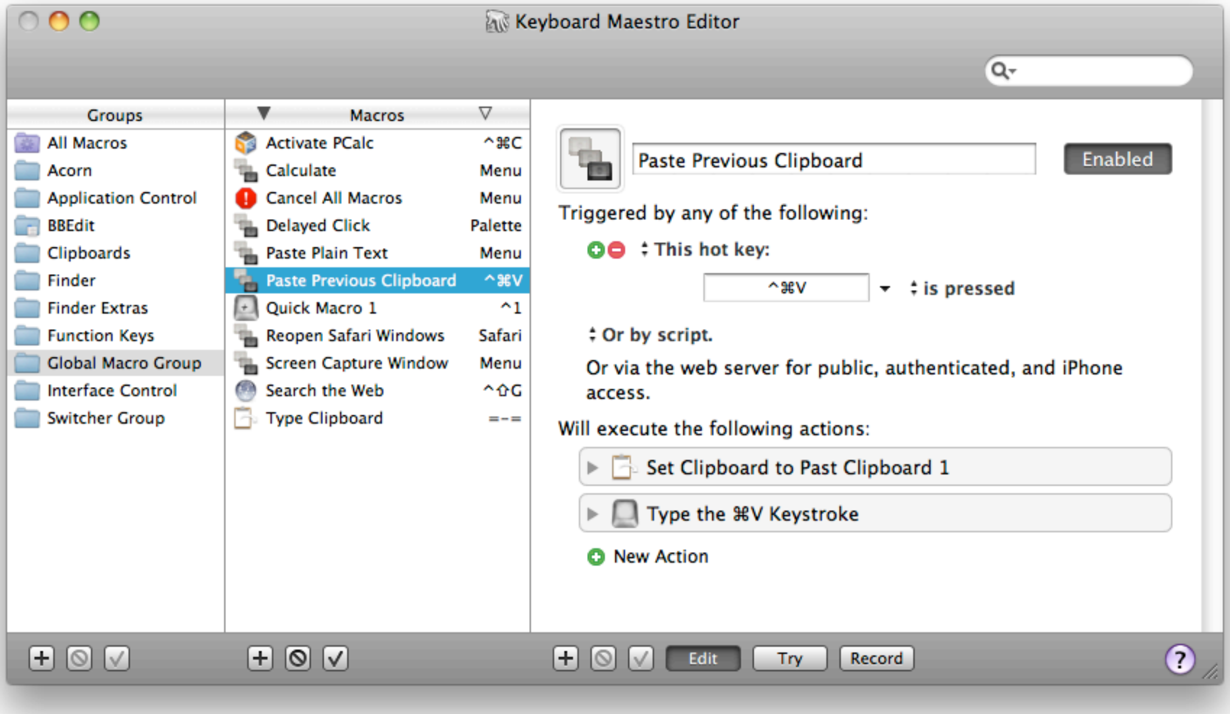
Macros

Macros, also known as Shortcuts, are a way of improving your productivity by allowing you to more quickly perform repetitive or frequently required actions, tailoring your Mac to your usage patterns.

A macro consists of one or more possible Macro Triggers which define when the macro will be executed together with a list of Macro Actions to be performed. Sets of Macros are collected together in to Macro Groups.

A typical simple Macro consists of a single Hot Key trigger, such as Control-A together with a single action, such as “type my address” .

To create a Macro, first launch Keyboard Maestro, select the desired Macro Group to contain it, and then click the + button below the Macros list. To edit a Macro, double click it, or select it the Edit button. The Macro Editor window will be displayed.



Paste Previous Clipboard

Triggered by any of the following:

This hot key: ^⌘V is pressed

Or by script.

Or via the web server for public, authenticated, and iPhone access.

Will execute the following actions:

Set Clipboard to Past Clipboard 1

Type the ⌘V Keystroke

Enter the name of your new Macro (you can skip this and it will be named for you based on the action you select).

You can create a new trigger by clicking on the green ⊕ button. There are several Macro Triggers to choose from, the most common being the Hot Key which allows you to execute the Macro at the press of a key. You can define several different triggers and any of the triggers will execute the Macro Actions. You can delete a trigger by clicking the red ⊖ button.

You can add a new action by clicking the New Action button or the + button at the bottom of the macro detail view to display the list of actions. There are many Macro Actions to choose from. The actions you include will be executed in order. You can reorder the actions by dragging them around. You can copy actions by option-dragging. You can also Copy and Paste actions. You can delete an action by selecting it and pressing the Delete key. You can enable or disable actions by selecting them and clicking the ✓ button at the bottom of the macro detail view.

An easy way to generate macro actions is to turn on recording by clicking the Record button at the bottom of the macro detail view and proceed to show Keyboard Maestro what you want to do. Then turn recording back off and look through the actions - chances are you will need to delete or adjust some of the recorded actions to make a robust macro, but this will be much quicker than creating each macro action manually.

You can display more or less detail about some actions by clicking the disclosure triangle. While disclosed, you can try the action immediately by clicking the Try button at the top right of the action. You can disclose all the actions simultaneously by option clicking on the disclosure triangle.

You can try the actions immediately by selecting some or all of them and clicking the Try button at the bottom of the macro detail view.

Macros are continuously saved, so the macro is live as soon as it is created. It will be available immediately (subject to the restrictions of the Macro Group it is contained in).

You execute a macro’s action sequence by triggering the macro using any of the Macro Triggers you have defined.

If the web server is enabled for remote access, you can trigger a macro remotely after logging in using a web browser or the Keyboard Maestro Control iPhone application.

You can also trigger a macro using AppleScript or another scripting language (select the “Or by script” entry to display script code in various languages such as AppleScript or Perl).

Note that the web server needs to be separately enabled in the Web Server preference pane, and all macros are subject to the restrictions of the Macro Group they are contained in.

You can disable or enable a macro by selecting it and clicking the ✓ button at the bottom of the Macros list. You can disable or enable a Macro using the Set Macro Enable action, or from AppleScript with:

```
tell application "Keyboard Maestro"
  setMacroEnable "Macro Name or UID" with/without enable
end tell
```

You can start editing a Macro or Macro Group from AppleScript with:

```
tell application "Keyboard Maestro"
  editMacro "Macro Group Name or UID"
end tell
```

See also the Macro Groups, Macro Actions, Macro Triggers and Recording sections.

Macro Triggers

- Overview
- Hot Key
- Typed String
- Application
- System Wake
- Login
- Time
- While Logged In
- Macro Palette
- Status Menu
- Public Web
- Device Key
- MIDI Note
- By Script

Overview

A Macro is executed when any of its Macro Triggers is activated. There are several triggers to choose from (detailed below), the most common being a Hot key, that is a Macro is executed in response to a keystroke, usually in conjunction with one or more modifier keys. You can also trigger a macro by typing a string. Or you can create a floating palette of macros, execute macros in response to applications launching or quitting, execute them remotely via the in-built web server, or at a particular time of day, as well as other possible events.

Hot Key

The most common Macro Trigger is the traditional Hot Key. You execute a Macro by pressing a key combination - a keyboard key like a letter, number, symbol or function key, often in combination with one or more modifiers (shift, control, option and/or command). Almost any key can be a trigger, and keep in mind the number pad counts as different keys to the numeric keys on the main keyboard.

The macro can execute when the hot key is pressed, released or continuously while it is held down. This allows you to do things like have a macro execute when the key is pressed, and then a second macro execute when the key is released, for example to toggle a setting on and then off again.

A common use for Hot Key triggers is to open applications or documents, insert text templates, or as a way of remapping command keys (although you can remap command keys in the System Preferences Keyboard preference settings).

Hot Keys suffer from the drawback that you need to remember a cryptic keystroke. This can be mitigated by selecting consistent keystrokes (such as Control-Letter to mean insert text and Control-Option-Letter to mean launch an application). You can also use a tool like KeyCue to display command keys and macro hot keys.

If multiple macros are executed with the same hot key, the duplicate macros are displayed in a palette allowing you to select the desired macro. You could use this feature to allow a single hot key to do multiple user-selected actions. Pressing a number key will allow you to execute a macro without using the mouse, or press escape or take any other action to cancel.

Typed String

The Typed String trigger lets you execute a macro in response to a sequence of keys.

Typed String triggers allow you to use more verbose (and hence descriptive) sequences of keys to trigger a macro. Because the keys first go through to the current application, the keys are usually deleted prior to executing the macro, although with this disabled you can use it in an application that largely ignores keys. For this reason, the allowed keys in a Typed String macro are limited to plain 7-bit ASCII characters.

To avoid macros firing unexpectedly it is a good idea to include a consistent prefix and/or suffix to your strings. For example, to insert your email address, rather than use just “em” (which would fire if you typed “them”, use something like “=em=” which you will not type accidentally.

You can optionally allow typed string triggers to work regardless of case, or to remember the case of the typed string and mimic the case in any Insert Text actions.

Note that the system will not allow Keyboard Maestro to see keys typed in password fields, so Typed String triggers will not fire if you type them in most password fields - you can use Insert Text by Typing to type in to password fields (although this is a serious security concern), but you cannot use Typed String triggers while inside a password field.

Application

You can have a Macro execute in response to an application event, such as when the specified application launches, quits, activates or deactivates. You can also have the Macro run periodically while an application is running or while it is at the front.

You could use a trigger like this to simulate workspaces by automatically setting up an application the way you want when you launch it, or you could clean up after an application when you quit.

As with all triggers, the trigger will fire only if the Macro Group that contains it is active, which is based on the current foreground application **before** the specified application launches, or **after** the specified application is deactivated or quits. In practice, this means the Macro Groups that contain this trigger should be targeted at All Applications.

System Wake

You can use the System Wake trigger to execute a macro when your Mac wakes from sleep.

You could use a trigger like this to set up your Mac environment, first determining your location and then taking appropriate action.

Login

You can use the Login trigger to execute a macro when you login.

You could use a trigger like this to set up your Mac environment when you start your Mac.

Time

You can use the Time trigger to execute a macro at a specific time, optionally restricted to certain days of the week.

You could use a trigger like this to set up your Mac environment before arriving at work, run periodic maintenance or backup scripts late on the weekend, or launch iChat for your weekly video conference.

While Logged In

You can use the While Logged In trigger to repeatedly execute a macro during a portion of the day, optionally restricted to certain days of the week.

You could use a trigger like this to run periodic maintenance or track changes.

Macro Palette

You can have a macro execute when you click on it in a floating Macro Palette. Keyboard Maestro will only display the palette when there are active Macros, so if your Macros are restricted to particular applications, then the Macro Palette will only appear in those applications. This is particularly useful for less frequently used macros whose Hot Key you might forget, or for complex apps like Final Cut Pro that has so many different but frequently used keyboard shortcuts that you cannot memorize them all.

You could use a trigger like this to add an action palette to an application that does not have such a facility.

You can control the sorting order of macros by adding two characters and a closing braket (eg “01)My Macro”). The prefix will be removed before displaying in the macro palette.

You can edit a macro by holding down the option key and selecting it from the macro palette.

Status Menu

You can have a macro execute when you select it from the Keyboard Maestro Status Menu (the Keyboard Maestro icon on the right hand side of the menu bar). You add a macro to the Status Menu by including the Status Menu trigger as a Macro trigger.

Keyboard Maestro will only display the Status Menu triggered macros that are active, so if your Macro is restricted to particular applications, then it will only appear in in the Status Menu in those applications.

The Status Menu is particularly useful for less frequently used macros whose Hot Key you might forget.

You could use a trigger like this to add custom facilities to applications, such as to open specific common files or set up windows in specific ways.

You can control the sorting order of macro groups and macros by adding two characters and a closing braket (eg “01)My Macro”). The prefix will be removed before displaying in the status menu.

You can edit a macro by holding down the option key and selecting it from the status menu.

Public Web

Keyboard Maestro has an in-built web server. You can enable it in the Web Server preference pane. If enabled, and if you configure a username and password, you can connect to your Keyboard Maestro’s web server and login and then execute any macro you have defined. Also, if the web server is enabled, and if you have configured any Macro with a Public Web trigger, then anyone on the Internet can connect to your Mac and trigger Public Web macros.

Macros are only available if they are currently active (ie, they must not be disabled or in a Macro Group that is disabled, and their Macro Group must be currently active which depends on the current application).

For example, if you are running some sort of process on your Mac that occasionally fails, you could write a script to restart it and make it available as a Public Web triggered Macro, which you (or anyone else) could then execute to restart the process.

Clearly there are some serious security issues with this, so you should use a lot of caution when you allow any macro to be executed with a Public Web trigger.

Device Key

This is an advanced trigger—generally you should use a Hot Key Trigger if possible.

Keyboard Maestro can trigger a macro when any device key is pressed—this includes modifier keys, mouse buttons, programable keyboards like P.I. Engineering’s X-Key, and even the brightness buttons on USB connected monitors.

The macro can be executed when the key is pressed, released or periodically while it is held down. The trigger can also optionally be restricted to when certain modifier keys are pressed.

For example, a macro could fire every five seconds while Control-Mouse Button 3 is held down.

This trigger watches input devices at a low level, but it does not affect them, so any key presses continue to have their normal operation as well as triggering the macro. This is fine for modifiers, unused mouse buttons, programable keyboards and other unused buttons, but would likely be problematic for normal keystrokes which will continue to have some other probably unwanted affect.

MIDI Note

Keyboard Maestro can trigger a macro when it recieves a MIDI note. You execute a Macro by pressing a key on a MIDI device like an electronic Keyboard.

The macro can execute when the MIDI note is pressed (note on), released (note off) or continuously while it is held down. This allows you to do things like have a macro execute when the key is pressed, and then a second macro execute when the key is released, for example to toggle a setting on and then off again.

By Script

Keyboard Maestro can be trigger from an AppleScript or shell script.

```
tell application "Keyboard Maestro Engine"
  do script "Name or UID of Your Macro"
end tell
```

or

```
osascript -e 'tell app "Keyboard Maestro Engine" to do script "Name or UID of Your Macro"'
```

The macro must be defined and currently active.

Macro Actions

Overview

A Macro executes a sequence of Macro Actions in order. There are many actions to choose from (detailed below), some simple actions, such as Sleep Computer, require no other information and simply do their job, while other more complex actions, such as Select Menu Item, require you to specify more information, such as a target application or menu name.

There are many powerful Macro Actions available for your use, and you can sequence them together to perform complex tasks.

To see all actions, select the All Actions category. To select just your favorite actions, select the Favorites. You can drag actions into your Favorites category.

Application Control

Application Control actions allow your to switch, quit, or hide applications. The actions are:

Switch to Last Application
    switches back to the previous application you were in.

Switch to Next Application
switches to the next application (alphabetically).

Activate a Specific Application
launches if necessary and brings the specified application to the front, optionally reopening the initial windows. If the application is already at the front, the action can optionally switch out, hide or quit it, allowing you to toggle an application, very useful for utilities like PCalc.

Bring Application Windows to Front
brings all the windows of the current application to the front.

Quit All Applications
quits all foreground applications, optionally honoring the excluded application list.

Quit Other Applications
quits all foreground applications except the current one, optionally honoring the excluded application list.

Quit Current Application
quits the current application, optionally force quitting or relaunching.

Quit a Specific Application
quits the specified application, optionally force quitting or relaunching.

Hide All Applications
hides all foreground applications, optionally honoring the excluded application list.

Hide Other Applications
hides all foreground applications except the current one, optionally honoring the excluded application list.

Hide Current Application
hides the current application.

Hide a Specific Application
hides the specified application.

Show All Applications
shows all foreground applications.

Show a Specific Application
shows the specified application.

Clipboard

Clipboard actions let you manipulate the system clipboard, Named Clipboards, the clipboard history and cut, copy or paste clipboard items. The actions are:

Cut, Copy, Paste
simulate Command-X, Command-C, or Command-V keystrokes to Cut, Copy or Paste to/from the system clipboard.

Set Clipboard to Text
set the system clipboard to text. Text Tokens are provided to include such things as the time or date.

Set Clipboard to Past Clipboard
set the system clipboard to some item that was previously copied to the system clipboard.

Delete Past Clipboard
delete a past system clipboard. This can be used to restore the system clipboard after another action sets the clipboard temporarily.

Use a Named Clipboard
cut, copy or paste to/from the specified Named Clipboard.

Set Clipboard to Named Clipboard
set the system clipboard to a specified Named Clipboard.

Save Clipboard to Named Clipboard
set the specified Named Clipboard to the current system clipboard.

Apply a BBEdit Text Factory
apply any saved BBEdit Text Factory to the current system clipboard.

Filter Clipboard
apply one of a set of filters on the current system clipboard.

Search and Replace Clipboard
search and replace the current system clipboard, optionally using regular expressions, and allowing Text Tokens. The replacement can also include $1 tokens for regular expression replacements.

The Filter Clipboard action includes a number of filters you can apply to the clipboard (or correspondingly, the Filter Variable action can apply them to variables).

The filters are:

- Remove Styles (ie, make the clipboard plain text - not applicable to variables).
- Set line endings to Mac, Unix or Windows/DOS.
- Trim Whitespace.
- Hard wrap or unwrap paragraphs.
- Lowercase (all characters), Lowercase First (just the first character).
- Uppercase (all characters), Uppercase First (just the first character).
- Capitalize (all words) or Title Case (inteligently uppercase certain first letters).
- Change quotes to Smart, Dumb or French quotation marks.
- Encode HTML or non-ASCII HTML entities.
- Decode HTML entities.
- Generate an HTML list.
- Percent Encode for URL.
- Get or delete the last path component or the path extension.
- Get the basename of the path (ie the name without directory or extension).
- Expand tilde (~) paths, or abbreviate with a tilde.
- Resolve symlinks, or standardize the path.
- Delete or bullet (•) control characters.
- Calculate an expression and return the result, see the Calculations section.
- Process Text Tokens and return the result, see the Text Tokens section.
- Count the characters, words or lines and return the result.

We will likely expand the list of possible filters, so if you have specific filtering needs that you think might be of general interest, please let us know. In the mean time, remember that you can apply scripted filters using an AppleScript or shell script, for example the shell script:

```
pbpaste | perl -pe 'tr/A-Z/a-z/' | pbcopy
```

is roughly equivalent to the Lowercase filter, except that it only works with ASCII characters.

Control Flow

You can control the flow of a macro execution using a variety of actions, from a simple Pause for a number of seconds, through a complex nesting of If/Then/Else and looping.

The actions to control the flow of a macro are:

Pause
pause for a number (which may be a calculation) of seconds.

Pause Until
pause until conditions are met.

Until
execute a list of actions until conditions are met.

While
while conditions are met, execute a list of actions.

Repeat
repeat a list of actions a number (which may be a calculation) of times.

For Each
loop over a collection of values.

If Then Else
if conditions are met, execute a list of actions, otherwise execute another list.

Execute a Macro

execute another macro (like a subroutine).

Cancel All Macros
: Cancel all macros that Keyboard Maestro Engine is currently executing.

Cancel This Macro
: Cancel this macros (including any macro that executed this macro).

You will often need to use the Pause action to slow down replaying of an action sequence to allow the system time to catch up—especially after you have changed application or if you want to use a mouse click.

The condition clause of the flow control actions can be any of:

- Any of the following are true - at least one condition must be true.
- All of the following are true - every condition must be true.
- None of the following are true - no condition is true.
- Not all of the following are true - at least one condition must be false.

If there are no conditions in the set at all, the action will not execute anything except the Until action which will execute the actions once. Neither side of the If Then Else will execute.

There is over a dozen different kinds of conditions, from testing what the current application is, through examining variables or the clipboard, and on to check your network location or testing a pixel on your screen, see the Conditions section.

The For Each action loops over a set of collection items, setting a variable to each value and executing the contained actions. The collections can include:

- A number range (eg 1 to 10).
- The running applications.
- The files in a directory (optionally recursively).
- The files or folders selected in the Finder.
- The currently mounted volumes.
- The lines in a clipboard, variable or file.
- The clipboard history.

The control flow actions include a nested list of their own actions to execute, and that nested list can include further control flow actions—go wild! But keep in mind, there may be a time when a shell or AppleScript is a more useful way of describing your solution.

See also the Conditions, Variables and Calculations sections.

Execute

Execute actions let you execute AppleScripts, Shell scripts, Automator Workflows or another macro. Shell scripts can be any kind of script, sh, zsh, tcsh, perl, python, and so on. The actions are:

Execute AppleScript
: execute a specified AppleScript, either from a file or text. For example, say "hello".

Execute Automator Workflow
: execute a specified Automator Workflow.

Execute a Shell Script
: execute a specified shell script, either from a file or text. For example, pbpaste | pbcopy.

Execute a Macro
: execute a specified macro. This allows you to create subroutines of instructions. The current macro waits until the submacro finishes.

The results of an AppleScript or shell script can be ignored, or you can:

- Display it in a floating window.
- Display it briefly in a HUD window (via Growl if available).
- Type it in to the current selection.
- Paste it in to the current selection.
- Save it in to a variable.

For example, you could have a shell script date display briefly in a HUD window every hour, or use a hot key to type the results directly into your text editor.

These powerful actions allow you to add any new facilities we have not provided for, stringing them together with other actions as desired.

You can also use the clipboard by piping from pbpaste and to pbcopy.

Shell scripts are executed in the background and can access variables by using environment variables, see the Variables section.

AppleScripts are executed in the background via osascript. This means they are not allowed to do user interaction. You can work around this by asking an application like System Events to do the user interaction for you, for example:

```
tell application "System Events"
  activate
  display dialog "Hello"
end tell
```

AppleScripts can access variables by talking to the Keyboard Maestro Engine, see the Variables section.

File

File actions allow you to interact with the file system, moving, copying, duplicating, trashing or deleting files or folders. The actions are:

- Create a new folder.
- Move (or rename) a file or folder.
- Copy a file or folder.
- Duplicate a file or folder.
- Trash a file or folder.
- Delete a file.
- Delete a file or directory (recursively). This is potentially very dangerous.
- Read a file to a variable or to the clipboard (can be an image or styled text).
- Write to a file from a variable or the clipboard (can be an image or styled text).
- Append text to a file from a variable or the clipboard.
- Get/Set file attributes.

The source path must be an absolute path (or a home relative ~ path). The destination path must either be a simple single path component (not ~ or /) or an absolute path. If it is a simple path component, then it is relative to the parent directory of the source path. This is particularly useful in the Move or Rename action, as you can, for example, rename from ~/Folder/Old Name to New Name.

You can read or write files in a variety of formats, including PNG, TIFF, JPEG, as well as HTML, Web Archive, Word Document and more.

Image

Image actions allow you to manipulate images, such as flipping, rotating, or capturing the screen. The actions are:

- Screen Capture - capture a window, screen or all screens.
- Create New Image - with size and color.
- Flip Image - horizontally or vertically.
- Rotate Image - by 90°, 180°, 270° or an arbitrary angle.
- Resize Image - resize image, resize canvas, add or remove margins or crop.
- Composite images or styled text onto an Image.
- Draw a Shape (line, rectangle or oval) onto an Image.
- Trim Image - trim off transparent areas from the edge of the image.
- Display Clipboard - display a clipboard image or styled text.
- Get Image Size - read the image size into a variable.
- Find Image on Screen - find the location of an image.

Image actions typically act on an image in the clipboard or in a named clipboard. You can read or write images files using the File actions.

Interface Control

Interface Control actions allow you to interact with the user interface, selecting menus, clicking buttons, simulating keystrokes and so forth. The actions are:

Manipulate Window
: resize, move, center, bring to front, close, zoom, or minimize the front window or a specified window.

Move or Click Mouse
: move, click or move and click the mouse using any mouse button, and optionally dragging. You can specify modifiers such as the shift key, as well as the position, relative to any corner of the front window, the main screen, or the current mouse location, or a found image on the screen.

Select Menu Item
: select a specified menu item in the front or a specified application.

Press Button
: press a named button in the front window.

Simulate Keystroke
: simulate pressing a specified keystroke.

Simulate Scroll Wheel
: simulate scrolling the mouse wheel up or down, left or right.

The various size and location fields can be Calculations.

The Select Menu Item action lets you choose from all currently running applications and their menus.

The Select Menu and Press Button actions allow you to specify multiple options separated by a vertical bar (eg Show|Hide) to allow for toggling menus. They will also ignore the difference between three dots (...) and an ellipsis so you do not have to worry which one the menu uses. Alternatively, you can start the name with an ^ and use a regular expression to match the menu or button name. Also, Select Menu will translate the word "APPLICATION" (all capitals) into the current application name, allowing menu selections like APPLICATION -> About APPLICATION.

The Select Menu and Press Button actions will, by default, abort the macro if the action is not successful (for example if the button or menu cannot be found or is disabled). The action can be configured to allow the macro to continue if the menu or button is not essential (such as “Mark As Read” which might be disabled if the item is already marked as read).

iTunes Control

iTunes Control actions allow you to interact with iTunes, playing songs, stopping or pausing, rewinding or fast-forwarding. All actions will launch iTunes if it is not already running. The actions are:

Play a Specific Track
: play a specified song.

Play a Specific Playlist
: play the songs in a specified Playlist.

Play a Random Track
: play a random song.

Play a Random Track from a Specified Playlist
: play a random song from a specified Playlist.

Play/Pause Current Track
: toggle from playing to pausing or vice-versa.

Pause Current Track
: pause the current song.

Stop Current Track
: stop playing any song.

Fast-Forward Current Track
: fast-forward the current song. It will keep fast-forwarding until you do something else or until it reaches the end of the song.

Rewind Current Track
: rewind the current song. It will keep rewinding until you do something else or until it reaches the beginning of the song.

Next Track
: play the next song.

Previous Track
: Go to the previous song or the start of the current song if it is already playing.

Increase/Decrease or Set iTunes Volume
: Increase, decrease or set the iTunes volume.

Increase/Decrease or Set Rating
: Increase, decrease or set the rating of the current track.

MIDI

Send a MIDI message:

Send MIDI Note On
: sends a MIDI Note On message, specifying the note, velocity and channel.

Send MIDI Note Off
: sends a MIDI Note Off message, specifying the note, velocity (usually 0) and channel.

Send MIDI Control Change
: sends a MIDI Control Change message, specifying the control, value and channel.

The MIDI messages come from a device named “Keyboard Maestro” which will be created as soon as you add any MIDI action.

Notifications

Keyboard Maestro can notify you in a variety of ways:

Growl
: display a message via Growl (or in a popup HUD window if Growl is not installed).

Display Text
: display text (which may contain Text Tokens).

Alert
: display an alert (which may contain Text Tokens).

Prompt For User Input
: display a dialog requesting information, see the Variables section.

System Beep
: play a standard system beep.

Play Sound
: play a sound, optionally through a specific output device.

Speak Text

speak text (which may contain Text Tokens) in any desired voice.

Highlight Location
: highlight a location on the screen.

Open

Open actions allow you to open files, folders, URLs or System Preference Panes. The actions are:

Open File or Folder
: opens a specified file or folder. Don’t underestimate the power of this command as you can open bookmarks or other action documents to perform a lot of customized actions.

Open the Finder Selection
: opens the files or folders that are currently selected in the Finder.

Open URL
: open a URL with the appropriate helper.

Open System Preference Pane
: opens a specified System Preference Pane.

Files, folders, the Finder selection and URLs can all open either in their default applications, or a specific application. So you could, for example, create a macro that opens the Finder selection in BBEdit.

QuickTime Player Control

QuickTime Player Control actions allow you to interact with QuickTime Player, playing movies, stopping or pausing, stepping forward or backward, or adjusting the volume. All actions will launch QuickTime Player if it is not already running. The actions are:

Play Current Movie
: play the current movie.

Play/Pause Current Movie
: toggle from playing to pausing or vice-versa.

Pause Current Movie
: pause the current movie.

Step Forward Current Movie
: step the current movie forward one frame.

Step Backward Current Movie
: step the current movie backward one frame.

Increase/Decrease or Set QuickTime Player Volume
: Increase, decrease or set the QuickTime Player volume.

Switchers

Keyboard Maestro includes several powerful switchers, including Application Launcher, Application Switcher, Window Switcher, and Copy, Cut and Paste Clipboard Switchers and Clipboard History Switcher.

Each of these switchers is actually just a macro action, triggered like any other macro. By default Keyboard Maestro creates a Macro Group called Switcher Group which includes the following macros:

Activate Application Launcher
: activates the Application Launcher (Command-Control-Tab).

Activate Application Switcher
: activates the Application Switcher (Command-Tab).

Activate Clipboard Copy Switcher
: copies the current selection to a named clipboard you select (Command-Shift-C).

Activate Clipboard Cut Switcher
: cuts the current selection to a named clipboard you select (Command-Shift-X).

Activate Clipboard Paste Switcher
: pastes a named clipboard you select into the current system clipboard and current selection (Command-Shift-V).

Activate Clipboard History Switcher
: pastes a previous system clipboard you select from the clipboard history into the current system clipboard and current selection (Command-Control-Shift-V).

Activate Window Switcher
: activates the Window Switcher (Control-Tab).

You can create your own macros using these actions or adjust their triggers within the Switcher Group.

By default, older versions of Keyboard Maestro used Option-Tab for its Application Switcher to avoid overriding the system Application Switcher, but if you wish to use Command-Tab you can do so. Similarly, if you wish to use the system switcher with Command-Tab, you can do so by changing the hot key trigger or by disabling Keyboard Maestro’s macro.

System Control

System Control allows you to control your Mac. Most are pretty self explanatory. The actions are:

Put Computer to Sleep
: puts your Mac to sleep.

Restart Computer
: restarts your Mac.

Shut Down Computer
: shuts down your Mac.

Fast User Switcher
: Fast User Switcher to a specific user.

Log Out
: Log Out the current user.

Activate Screen Saver
: starts the screen saver (could be especially useful if the screen saver requires a password to unlock it).

Open/Close CD Tray
: Opens the CD tray if it is closed, or closes it if open (especially useful for keyboards that do not have an eject key).

Toggle System Sound Mute
: Mutes the sound if sound is currently on, unmutes the sound is already muted.

Increase/Decrease System Sound Volume
: Increases or decreases the sound volume.

Set System Sound Volume
: Sets the sound volume to a specific amount (can be a calculation).

Increase/Decrease Screen Brightness
: Increases or decreases the screen brightness.

Text

The Insert Text action allows you to insert specified text either by setting the system clipboard and pasting, or by simulating a sequence of keystrokes. The Display Text action allows you to display the resulting text in a floating window or briefly in a HUD window (via Growl if available).

Tokens are provided to include variables, calculations, the time or date, or a large variety of other system information.

The Insert Text action can be very useful to insert standard text templates, such as your signature, address, copyright or other boilerplate text, and so on. If you ever get email from Peter, you will probably notice that many of his emails end with “Thanks for your kind words, Peter.” - with the amount of email he deals with, you don’t think he types that in every time do you?

Insert Text can expand various tokens, including dates in any ICU data format. For example, you can use an Insert Text action to insert the copyright message at the top of code files:

```
/*
```

```
 *  Created by %UserName% on %ICUDateTime%EEE d MMM yyyy%.
 *  Copyright (c) %ICUDateTime%yyyy% Stairways Software. All rights reserved.
 */
```

See also the Text Tokens section.

Variables

Keyboard Maestro includes permanently stored variables that you can use in a number of special purpose actions, as well as in Calculations or Text Tokens which can be used in almost any field in Keyboard Maestro.

These specific actions let you manipulate variables:

Set Variable to Text
    set a variable to text. Text Tokens are provided to include such things as the value of other variables or the time or date.
Set Variable to Calculation
    set a variable to the result of a calculation, see the Calculations section.
Filter Variable
    apply one of a set of filters to a variable, see the Clipboard section.
Search and Replace Variable
    search and replace the value of a variable, optionally using regular expressions, and allowing Text Tokens. The replacement can also include $1 tokens for regular expression replacements.
Use Variable
    Use a variable to set a variety of system values like mouse location or front window. See below.
Prompt For User Input
    display a dialog asking for a variety of user input. See below.

The Use Variable action can perform the following actions:

- Set the mouse location.
- Set the front window position, size or frame.
- Set the front window by name.
- Set the front application by name.
- Set the system volume.

Most of these have analogs in the Text Tokens, so for example you might do:

```
Set variable ‘Temp’ To Text ‘%FrontWindowFrame%’
Set variable ‘Temp’ To Calculation ‘Temp[1]+Temp[3]/2,Temp[2]+Temp[4]/2’
Use ‘Temp’ to Set the Mouse Location
```

Which will result in the mouse being placed at the center of the front window.

Prompt For User Input will allow you to ask for user input, storing any results in variables as designed. If the variable is a password variable, it will be displayed in a password field.

If the default value you specify consists of choices separated by a bar (|), then a popup menu will be used (the first value will be the default value, and can be repeated later if a different location is desired). For example “Better|Good|Better|Best” would result in a popup menu with Good, Better, and Best, with Better pre-selected.

If the popup menu would contain only "0" and "1", then a checkbox is used (so for example “0|1" or "1|0", depending on the desired default).

You can include one or more buttons, which may optionally cancel the macro. You can also include / to configure command keys for the buttons (/. means escape, / by itself means the default button). If there are no text fields in the dialog, the command key is not needed and the letter by itself will complete the dialog. The name of the button pressed will be stored in the Result Button variable.

See also the Variables, Text Tokens and Calculations sections.

Web

This action displays a HUD window where you can enter a search query and then sends the query in your web browser. You can customize the URL (by default the query is to search Google).

Record Quick Macro

Recording allows Keyboard Maestro to watch you as you perform a task and create the actions to produce a similar result.

Record Quick Macro is a variant of this that you can activate at any time without even launching Keyboard Maestro. You trigger the recording, and then perform a series of actions, and then turn off recording. Then press the associated hot key to replay the recording.

By default, Keyboard Maestro creates a macro triggered by Control-F1 which executes the Record Quick Macro action with a hot key of Option-F1. So for example, if you press Control-F1 to start quick recording, type “hello” and then press Control-F1 again to finish recording, then Keyboard Maestro will type “hello” each time you press Option-F1.

Because you cannot see or edit the recorded actions, it is best to keep them very simple. A good rule of thumb would be not to touch the mouse, just use the keyboard.

For example, say you wanted to quote a dozen different words in a paragraph, you could click in the middle of the first word, press Control-F1 to start quick recording, type Option-Left Arrow, quote ("), Option-Right Arrow, quote ("), and then Control-F1 again to finish recording. Now click in the middle of each remaining word and press Option-F1.

Set Macro or Group Enable

The Set Macro or Group Enable action allows you to enable, disable or toggle the enable of a macro group or macro. The Keyboard Maestro editor does not need to be running, but it will see the enable state when it is next launched (or immediately if it is already running).

Comment

The Comment action lets you store a comment in your Macro. It does nothing when executed. This can be useful for documenting complex macros.

Show/Hide Macro Palette

These macros will show or hide the global macro palette. It will remain hidden until shown. You can also hide the global macro palette by clicking the close icon, so you may wish to include a global macro, perhaps with a status bar trigger, which uses the Show Macro Palette to redisplay it.

Variables

Keyboard Maestro includes permanently stored variables that you can use in a number of special purpose actions, as well as in Calculations or Text Tokens which can be used in almost any field in Keyboard Maestro.

Variable names must start with a letter, and then can contain letters, numbers, spaces, or underscores. Variable names are case insensitive, but their case is remembered.

Variable names that starts or ends with “Password” or “PW” are considered passwords - there values will not be stored (except in memory) and they cannot be read by shell or AppleScripts. The Prompt For User Input dialog will display such variables in a password field.

Variable values are text, but they can contain comma separated numbers, and can then be accessed as arrays (eg Variable Name[1]). Mouse positions, window frames and the like can then be stored and manipulated in variables.

Variables can also be accessed from shell scripts via the environment variables in the form $KMVAR_Variable_Name where KMVAR_ is prefixed, and spaces are converted to underscores.

AppleScript can read variables with:

```
tell application "Keyboard Maestro Engine"
  set myVar to make variable with properties {name:"Calculation Result"}
  set theValue to value of myVar
end tell
```

and write variables with:

```
tell application "Keyboard Maestro Engine"
  make variable with properties {name:"My Variable", value:"New Value"}
end tell
```

and delete variables with:

```
tell application "Keyboard Maestro Engine"
  try
    set myVar to make variable with properties {name:"Old Variable"}
    delete myVar
  end try
end tell
```

You can also set a variable value to %Delete% to avoid having it show up in variable popup menus.

Text Tokens

Text Tokens allow you to add information to text fields. You can insert text tokens into a field using the Insert Token popup near the field.

The available text tokens include the following:

%Variable%Variable Name%
the value of the variable, see the Variables section.

%Calculate%1+2%
perform a calculation and use the result, see the Calculations section.

%CurrentClipboard%
the text of the current clipboard.

%PastClipboard%1%
the text of a past clipboard.

%NamedClipboard%Clipboard Name%
the text of a named clipboard.

%LongDate%
the current date in long format.

%ShortDate%
the current date in short format.

%NumberDate%
the current date in numeric format.

%LongTime%
the current time with seconds.

%ShortTime%
the current time without seconds.

%ICUDateTime%EEE, MMM d, yyyy%
the current date and time in any ICU data format.

%ICUDateTimePlus%3*6%Hours%EEE, MMM d, yyyy%
an offseted date and time in any ICU data format.

%UserName%
the current user name.

%UserLoginID%
the current user login id.

%MacName%
the current Mac name.

%MacIPAddress%
the current Mac IP Address.

%AddressBook%Name%
your AddressBook name.

%AddressBook%First%
your AddressBook first name.

%AddressBook%Last%
your AddressBook last name.

%AddressBook%Nickname%
your AddressBook nickname.

%AddressBook%Organization%
your AddressBook organization.

%Application%n%
the name of the n’th application.

%FrontWindowPosition%
the position of the front window.

%FrontWindowSize%
the size of the front window.

%FrontWindowFrame%
the frame of the front window.

%FrontWindowName%
the name of the front window.

%WindowPosition%n%
the position of the n’th window.

%WindowSize%n%
the size of the n’th window.

%WindowFrame%n%
the frame of the n’th window.

%WindowName%n%
the name of the n’th window.

%CurrentMouse%
the current mouse location.

%Screen%Main%
the frame of the main screen.

%Screen%2%
the frame of the second screen.

%SystemVolume%
the current system volume.

%NetworkLocation%
the current network location name.

%CurrentTrack%name%
the current iTunes track name.

%CurrentTrack%artist%
the current iTunes track artist.

%CurrentTrack%album%
the current iTunes track album.

%CurrentTrack%ratingstars%

the current iTunes song rating.
%ExecutingMacro%
the name of the currently executing macro.
%ExecutingMacroGroup%
the name of the macro group containing the currently executing macro.
%Tab%
the tab (0x08) character.
%Space%
the space character.
%LineFeed%
the linefeed (0x0A) character.
%Return%
the carriage return (0x0D) character.
%NN% or %NNNN%
arbitrary hex unicode characters (eg %41% is an A).
\a,\b,\e,\f,\t,\r,\n
characters (bell,backspace,escape,form feed,tab,return,line feed).

You can also use a short form of just %Variable Name% to include variables as long as there is no corresponding text token.

The index for windows and applications is from front to back when positive, or from back to front when negative.

To include a percent in your text, simply double the percent (%%). To include a backslash \ in your text, double the backslash (\\).

Calculations

Keyboard Maestro supports calculations in almost any numeric field. For example, you can Pause for 60*Time in Minutes. Calculations can also use comma separated lists of numbers as arrays, and can return such arrays, so you can operate on frames and points, for example:

```
Set variable ‘Temp’ To Text ‘%FrontWindowFrame%’
Set variable ‘Temp’ To Calculation ‘Temp[1]+Temp[3]/2,Temp[2]+Temp[4]/2’
Use ‘Temp’ to Set the Mouse Location
```

will result in the mouse being placed at the center of the front window.

Unfortunately, because of this you must use commas for this purpose, and full stops (.) for decimal numbers, regardless of your desired language.

Keyboard Maestro’s expressions include precedence, nested bracketed expressions, many inbuilt functions, various numeric bases, so you should be able to write most expressions you might like to use, as well as use it as a general purpose calculator if desired.

Operators based on precedence from lowest to highest are:

array separator (,)
separates elements of an array.
ternary operator (?)
a==b ? 3 : 4.
bitwise or (|), bitwise and (&) and bitwise xor
operators.
comparison operators (<, ≤, =, >, ≥, ≠)
compare for (in)equality and return 0 or 1.
shift operators («, »)
shift a number left or right.
addition operators (+, -)
add or subtract.
multiplication operators (*, /, MOD)
multiply, divide or mod.
power operator (^)
exponentiation.
unary prefix operators (√, -, brackets)
square root, negation, sub-expressions.
functions
a variety of functions.
numbers and variables or array accesses (5,$50,0x50,8#007,Variable,Variable[5])
identifiers and values.
unary postfix operators (!,%,°)
factorial, percent, degrees.

To minimize conflict with variables, operators and functions must be in uppercase.

Supported functions include:

SIN, COS, TAN, ASIN, ACOS, ATAN, ATAN2
trigonometry functions
SINH, COSH, TANH, ASINH, ACOSH, ATANH
hyperbolic trigonometry functions
EXP
exponentiation of e.
LOG or LN
logarithm base e.
LOG2, LOG10
logarithm base 2 or 10.
ABS
absolute value.
CEIL, FLOOR
integer ceiling or floor.
TRUNC, ROUND
truncate or round.
MIN, MAX
Minimum or maximum.
RANDOM
random real number from 0 to, but not including, N (defaults to 1).
RAND
random integer number from 0 to, but not including, N.
MICROSECONDS or MICROS
microseconds since startup.
MILLISECONDS or MILLS or MS
fractional milliseconds since startup.
SECONDS
fractional seconds since startup.
NOW
current time in seconds since the start of 1970.
TIME
time in seconds since the start of 1970.
JULIANDATE or JD
days (and fractions of days) since January 1, 4713 BC Greenwich noon.
MJD
days since the start of November 17, 1858.
TIME2JD
convert time from unix TIME to JULIANDATE.

TIME2MJD
convert time from unix TIME to MJD.

JD2MJD
convert time from JULIANDATE to MJD.

JD2TIME
convert time from JULIANDATE to TIME.

MJD2TIME
convert time from MJD to TIME.

MJD2JD
convert time from MJD to JULIANDATE.

YEAR
get the year component from a unix time (seconds since the start of 1970).

MONTH
get the month component from a unix time.

DAY
get the day component from a unix time.

DOW
get the day of the week (1 = Sunday, 7=Saturday) component from a unix time.

HOUR
get the hour component from a unix time.

MINUTE
get the minute component from a unix time.

SECOND
get the second component from a unix time.

MOUSEX
the x coordinate of the current mouse location.

MOUSEY
the y coordinate of the current mouse location.

MOUSEBUTTON
1 if the corresponding button is pressed, 0 otherwise.

SCREEN(n, Left|Right|Top|Bottom|Width|Height|MidX|MidY)
screen frame coordinates.

WINDOW(n, Left|Right|Top|Bottom|Width|Height|MidX|MidY)
window frame coordinates.

SCREENCOUNT
the number of screens (displays).

WINDOWCOUNT
the number of windows in the front application.

APPLICATIONS
the number of running applications.

IDLE
the number of (fractional) seconds the Mac has been idle.

ONLINE
whether the Mac is currently connected to the Internet.

BATTERY
whether the Mac is currently running off battery power.

SCREENSAVER
whether the Mac is currently displaying the screen saver (or the display is off).

The JD, MJD, and TIME functions return the current date and time, or can take the date (year, month, day) or date and time (year, month, day, hour, minute, seconds) to return.

The YEAR, MONTH, DAY, DOW, HOUR, MINUTE, SECOND functions return the relevant component of the current time or the can take the a unix time (seconds since the start of 1970).

The SCREEN index can be 0 for the main screen, and then 1 through n are the screens in orientation order.

The WINDOW index can be 0 for the main focussed window, and 1 through n are the screens in Z-order (1 is usually the main window), or -1 through -n in reverse order.

IDLE time is based on the Human Interface (HID) system, and so notices only HID device activity like mouse movement or keyboard presses, not things like disk access or movies playing.

Some example functions might be:

```
Amount in Dollars * 100
MJD() > 55928
NOW() > TIME(2012,3,23,12,2,1)
DOW(2012,4,4) == 4
Radius*SIN(20°),Radius*COS(20°)
Window Frame[1]+Window Frame[3]/2,Window Frame[2]+Window Frame[4]/2
MOUSEBUTTON() + 2 * MOUSEBUTTON(4)
```

Conditions

Keyboard Maestro includes a variety of Control Flow Actions which perform actions depending on a set of conditions.

The condition clause of the flow control actions can be any of:

- Any of the following are true - at least one condition must be true.
- All of the following are true - every condition must be true.
- None of the following are true - no condition is true.
- Not all of the following are true - at least one condition must be false.

This is followed by a set of specific conditions. **If there are no conditions in the set at all, the action will not execute anything** except the Until action which will execute the actions once. Neither side of the If Then Else will execute.

The available conditions include:

Application
test if an application is running (or not) or at the front (or not).

Front Window
test if the front window exists or has a desired title (or not).

Button
test if a button exists (or not) or is enabled (or not).

Menu
test if a menu item exists (or not) or is enabled (or not).

Modifiers
test if particular modifiers are pressed or not pressed.

Key
test if a particular key is down or up.

Typed String
test if a typed string trigger with remember case was a particular case.

Disk
test if a disk is mounted or not.

Path
test if something, a file or folder exists at a particular path (or now).

Clipboard
test if the clipboard contains text or an image (or not) or specific text (or not).

Variable
test if a variable exists or is empty or contains specific text (or not).

Calculation

test if a calculation returns non zero.

Environment Variable
: test if an envirnment variable exists or contains specific text (or not).

Pixel
: test if a pixel on the screen is or is brighter/darker, more or less blue, etc that a particular color.

Location
: test if the network name is or contains specific text (or not).

Script
: test if a script succeeds or returns particular status or specific text (or not).

Generally, a pixel condition should be considered a last resort, but there are certain cases where it may be useful - keep in mind that you can use expresions for calculating the location of the pixel too.

See also the Control Flow, Variables and Calculations sections.

Recording

A major feature of Keyboard Maestro is the ability to create macro action sequences by recording your actions.

For example, to create a macro that simulates keystrokes, rather than create each macro action individually, you can enable recording and then simply type the keystrokes.

Keyboard Maestro can record the following actions:

- Moving a window
- Resizing a window
- Miniaturizing a window
- Clicking the mouse
- Typing a Keystroke
- Moving the scroll wheel
- Selecting a menu
- Activating an application
- Quitting an application

There are two ways you can use recording: when creating or editing a macro; or via a Record Quick Macro action.

When you are creating or editing a macro, with the Macro Editor window displayed, simply click the Record button to start recording. Then demonstrate the task you would like to perform using any of the above actions and Keyboard Maestro will record your actions as a sequence of Macro Actions. When you are finished, click the Record button again to stop recording.

While Keyboard Maestro is recording, it will display the Recording window.

You can stop all recording by clicking on the Recording window.

Typically you will need to make a few adjustments to the Macro Actions to ensure the macro will operate robustly when used. For example, you will likely have to switch applications immediately after clicking the Record button, and you may not want that recorded action. Generally, use recording to create a base sequence of actions and then adjust as necessary.

The other way to use recording is via a Record Quick Macro action. When triggered, the Record Quick Macro immediately starts recording your actions into a private macro. When you have demonstrated the sequence of actions you want, trigger the Record Quick Macro again. The sequence can now be executed via the specified Hot Key or the Status Menu or Macro Palette. For example, if the Record Quick Macro is triggered by pressing Control-F1, and the specified Hot Key is Option-F1, then if you typed:

Control-F1, h, e, l, l, o, Control-F1

Then each time you press Option-F1, Keyboard Maestro will type “hello” for you. One common use for this is if you want to adjust a sequence of lines in a systematic way. For example, if you had a list of colors, and wanted to change them in to a list of constants, say from this:

```
color Red
color Green
color Blue
```

to

```
const int kRed = "Red";
const int kGreen = "Green";
const int kBlue = "kBlue";
```

You could do this with grep and regular expression, replacing “color (.*)” with “const int k\1 = "\1";”, which is fine if you can remember how to do grep with regular expressions, whether it is \1 or $1, and whether the application you are in supports regular expressions or not. But perhaps a simpler way is to just show Keyboard Maestro how to do the first line and then let it do the others with a single keystroke each.

So move the cursor to the start of the first line, press Control-F1, then the sequence:

Option-Shift-Right Arrow, Delete, Forward Delete, Command-Shift-Right Arrow, Command-X, c, o, n, s, t, <space>, i, n, t, <space>, k, Command-V, =, ", Command-V, ", ;.

Finish with Command-Left Arrow, Down Arrow to carefully put the cursor at the start of the next line. Now press Control-F1 again to finish the recording, and Option-F1 twice to translate the next two lines.

Record Quick Macros can record the same set of actions that normal recording can, however because you cannot see or edit the recorded actions it is wise to keep them simple, preferably just a sequence of keystrokes. Typically, recorded Quick Macros will be used immediately and not reused, but they are saved and remain available until you record over them.

Macro Library

The macro library is a place where we can provide you with a large variety of optional macros. You can download new potential macros from us or from friends or colleagues. You can also share your own macros with out Keyboard Maestro users by exporting your clever macros as a library item.

Keep in mind that macros can do practically anything on your Mac, including cause a huge amount of damage, so you should never execute a macro without verifying the source and better yet, checking exactly what it does.

To use the library, choose Macro Library from the Window menu to display the macro library. You can then look through the available macros and insert any you'd like to use into your macros. You can then use them as is, or configure the new macros, perhaps changing the hot keys or adjusting the macros to your liking.

Each macro comes with a short description to tell you what it does, so scroll through them to see all the possibilities, and click on them to get more details.

Macro Examples

Here are a number of example and suggestions for Macros to give you some ideas of how you can get the most out of Keyboard Maestro and your Mac. For tips on how to remember which Hot Key executes which action, see the Remembering Macro Hot Keys section.

Launch Your Most Used Applications

Use function keys to launch or switch to your most used applications. For example, you probably often switch to the Finder, your Email client, your Web Browser, your Word Processor. Consider putting these and other frequently used applications on function keys.

Open Your Most Used Documents

Use Control-Function Keys to open your most used documents. For example, you might have a documentation file or financial details file that you access frequently, consider putting these on Control-Function Keys.

Insert Text Templates

Use Control-Letter and the Insert Text action to type in text for you, such as your name, address, phone number, and so on. Consider restricting these to just the appropriate applications like your Email client or Word Processor by creating a Macro Group for them. Also consider using Typed String triggers for these sorts of macros, for example “=em=” for email address and “=addr=” for address.

Use Hot Keys to Open Financial Accounts

If you keep your finances on your computer, then you probably need to open a document every time you enter a bill or receive a statement. By creating a Hot Key to open the document for you, you can save a few seconds every time - at least it might make receiving a bill slightly less unpleasant! If you have multiple accounts (eg personal, business, association) then this can be even more useful.

Use Hot Keys to connect to SSH or FTP sites.

You could use Hot Keys to connect to your common servers. You might need to create a Bookmark file for the site and use the Open File Macro Action.

Simulate Bookmarks

You could Use Hot Keys to Insert Text, Simulate Tab Press, Insert Text, Simulate Return Press to make effective bookmarks for applications that do not have an in-built bookmark system or to enter form data. While you could use this to enter usernames and passwords I would advise you use a tool like Web Confidential for such things as it will provide secure storage for your password.

Remap Command Keys

If you find yourself pressing a command key in an application and expecting it to do something but it does not work (for example, Command-T for “Replace and Find Again”), use a Macro to make the command key “do the right thing” in that application. Similarly, if you use a function in an application frequently, but it has a convoluted command key or no command key at all, define your own command key by using a Hot Key to select the menu item. Keep in mind that you can do some menu key remapping in the System Preferences Keyboard preference.

Simulate Missing Features

If you find yourself missing a feature in one application that you are used to in another application (perhaps you switched email clients and a feature is missing), see if you can simulate the feature with a sequence of commands and then use a Hot Key for that. For example, Close Window, Down Arrow, Return to move to next email message, or Command-Left Arrow, Shift-Down Arrow, Command-C, Down Arrow, Command-V to duplicate a line.

Swap Characters

If you often type characters out of order, use a Hot Key to swap them by first placing the cursor between them and then executing:

- Simulate Keystroke Shift-Right Arrow
- Cut to Named Clipboard “Temp”
- Simulate Keystroke Left Arrow
- Paste from Named Clipboard “Temp”

Save a Text Clipping

If you often want to save snippets of text, you could create a Hot Key to save a clipping:

- Copy
- Open File “Clippings.rtf”
- Simulate Keystroke Command-Down Arrow
- Insert Text “== %LongDate% %ShortTime% ==<return>” by Typing
- Paste
- Simulate Keystroke Return
- Simulate Keystroke Return
- Select Menu Item File » Save
- Manipulate Window Close Front Window
- Switch to Last Application (or Quit Specific Application or Command-Q)

Delayed Click

Setup a macro which simply pauses for twenty seconds and then clicks the mouse. Then when you need to print on to an envelope, go all the way through the process, position the mouse over the Print button, execute the Macro, walk over to the printer, insert an envelope and then take the printed envelope back with you.

Insert Boilerplate Text

If you regularly need to insert boilerplate text (eg copyright or file creation text), use an Insert Text macro to insert the text quickly and easily. It can even expand tokens to insert the date or other information.

Apply Text Conversions

If you are regularly translating text from one format to another in an automatic process, perhaps you can automate the whole thing with a Keyboard Maestro macro. For example, converting header functions into function entries. Using a shell script with pbcopy and pbpaste may also be useful for this purpose.

Simulate Workspaces

Create a macro to setup an application to your liking. For example, create multiple tabs in Terminal, each in its own directory, or open multiple documents in TextEdit, each positioned and sized appropriately.

Setup an Application When Launched

If you always do a set of things every time you launch an application (eg arrange the windows in a particular way), use an application Macro Trigger to execute a Macro when you launch the application, then have the Macro do the work for you.

Clean Up After Using an Application

If you always do something after quitting an application (eg unmount a server or disconnect from the Internet), use an application Macro Trigger to execute a Macro when you quit the application. You might need to do a little AppleScripting to perform the action and then use the Execute AppleScript action.

Feedback During Macro Execution

A Macro can play a System Beep, but an alternative is to use an AppleScript or shell script to speak text (AppleScript say "hello").

You can also use the Alert action to display a window with specified text. This also allows you to stop the macro if you decide not to proceed.

Rakesh Kumar’s PC Switcher’s Pack

Rakesh Kumar has created a pack to make life easier for Windows to Mac Switchers. It includes a set of macros for Mail, Microsoft Word and Microsoft PowerPoint as well as macros to map control keys to command keys for various common actions like Cut/Copy/Paste. It also includes a DefaultKeyBinding.dict for Mail to make it work more like Windows users expect.

Download Rakesh Kumar’s PC Switcher Pack and follow the instructions.

Application Launcher

The Activate Application Launcher action is essentially a highly specialized macro action that enables you to launch applications. By triggering the macro, the Application Launcher enables you to launch any applications in your Applications or Utilities folder, as well as any recently running applications. Once the launching window appears, you may select the application to launch, and Application Launcher will launch it for you.

The applications are also listed in the Status Menu, so you can launch applications that way if you prefer.

By default, Keyboard Maestro creates a Activate Application Launcher macro in the “Switcher Group” Macro Group, triggered by Command-Control-Tab. You can disable this Macro by selecting the Switcher Group, then selecting the Activate Application Switcher macro and clicking the ✓ button below the Macros list. You can edit this macro to change the trigger to any other desired Hot Key.

Application Switcher

The Activate Application Switcher action is essentially a highly specialized macro action that enables you to launch, switch, hide, and quit applications. By triggering the macro, the Application Switcher enables you to switch between all running applications. Once the switching window appears, you may select the application to activate, and Application Switcher will take you to it.

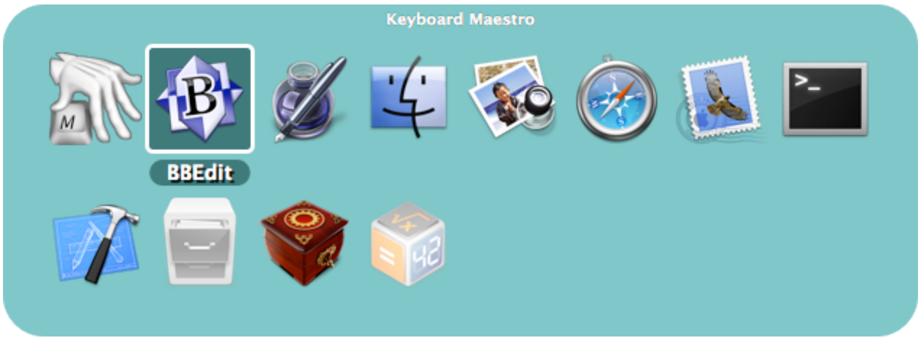
The Activate Application Switcher action lets you choose from three themes (vertical list, horizontal icons, or icon grid), as well as configure the icon size, color tint, and the sort order.

You can also choose to hide other applications when switching (Keyboard Maestro also has a preference in the General preference pane to always hide other applications when switching).

You can select various applications to always be displayed, even if they are not currently running, perfect for launching frequently used applications. In the Excluded preference pane, you can configure various applications to never be displayed.

While the Application Switcher window is displayed, you can perform various actions:

- Press “q” to mark (or unmark) an application to be quit.
- Press “k” twice to mark an application to be force quit.
- Press “s” or “h” to mark (or unmark) an application to be hidden.
- Press “l” or “z” to mark (or unmark) an application to be launched.
- Press “a” to hide (or show) “always included” applications.
- Press “e” to show (or hide) “always ignored” applications.
- Press “c” to select the current application.
- Press “f” to select the Finder.
- Press “d” to switch directly to the current app and hide other applications.



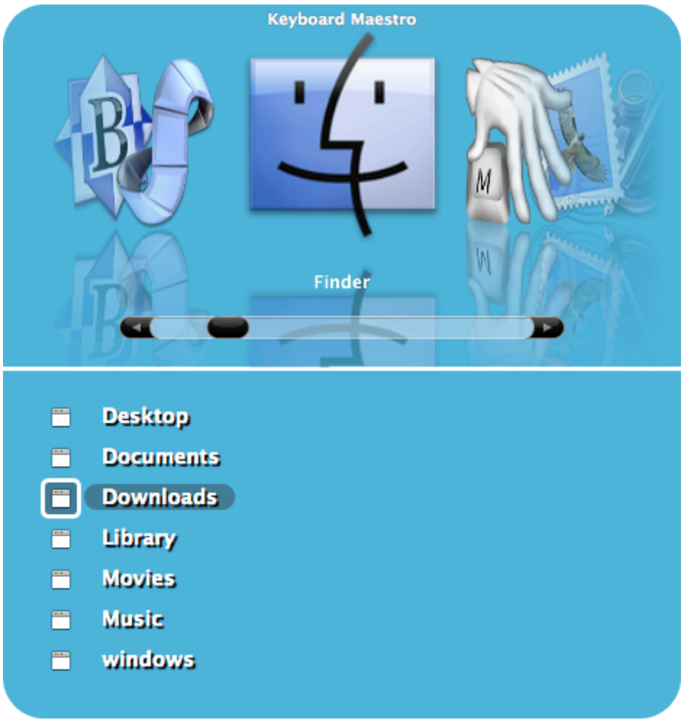
By default, Keyboard Maestro creates a Activate Application Switcher macro in the “Switcher Group” Macro Group, triggered by Command-Tab. You can disable this Macro by selecting the Switcher Group, then selecting the Activate Application Switcher macro and clicking the ✓ button below the Macros list. You can edit this macro to change the trigger to to any other desired Hot Key avoid replacing the system application switcher.

Window Switcher

Activate Window Switcher is essentially a highly specialized macro action that enables you to show, hide, and minimize windows. By triggering the macro, Window Switcher enables you to switch between all open windows in the current application. Once the switching window appears, you may select the window to activate, and Window Switcher will bring it to the front.

While the Window Switcher window is displayed, you can perform various actions:

- Press "q" to mark (or unmark) a window to be closed.
- Press "s" to mark (or unmark) an window to be minimize.
- Press left or right arrow keys to scroll through applications.



Keyboard Maestro

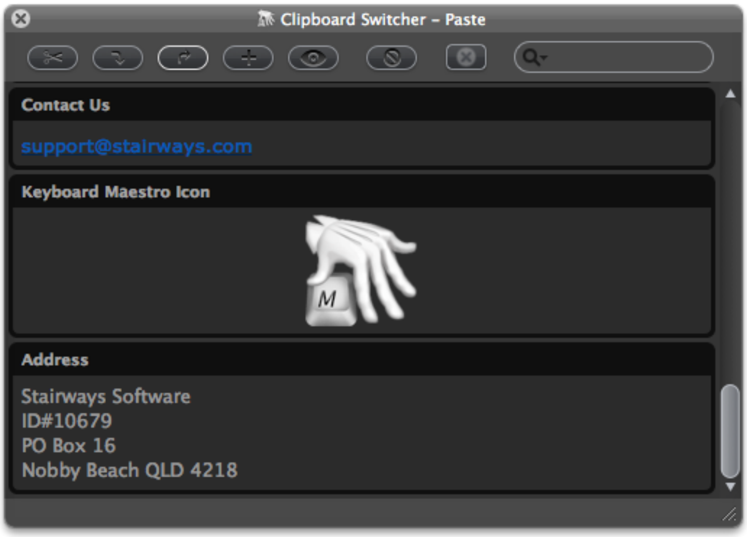
Finder

Desktop
Documents
Downloads
Library
Movies
Music
windows

By default, Keyboard Maestro creates a Activate Window Switcher macro in the "Switcher Group" Macro Group, triggered by Control-Tab. You can disable this Macro by selecting the Switcher Group, then selecting the Window Switcher macro and and clicking the ✓ button below the Macros list.

Clipboard Switcher

Clipboard Switcher enables you to define any number of named clipboards which can be use to Cut or Copy into and Paste from in any application. To use Clipboard Switcher you simply trigger the Clipboard Switcher macro. Clipboard Switcher will present you with a window allowing you to select the named clipboard to use.



Clipboard Switcher – Paste

Contact Us

support@stairways.com

Keyboard Maestro Icon

Address

Stairways Software
ID#10679
PO Box 16
Nobby Beach QLD 4218

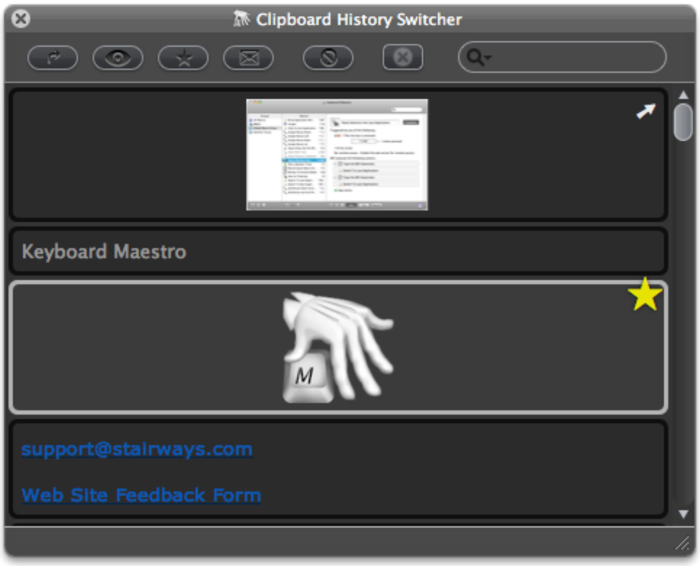
You can use type-ahead to select a named clipboard, and you can use the search field to filter the clipboards.

By default, Keyboard Maestro creates three Clipboard Switcher macros in the "Switcher Group" Macro Group, triggered by Command-Shift-X, C and V (Cut, Copy and Paste). You can disable the Macros by selecting the Switcher Group, then selecting the macros and and clicking the ✓ button below the Macros list.

You can add or delete named clipboards in the Clipboards preference pane.

Clipboard History Switcher

Clipboard History Switcher saves a copy of each clipboard every time you copy something. You can then paste any previous system clipboard by triggering the Clipboard History Switcher macro. Clipboard History Switcher will present you with a window allowing you to select any of the past clipboard and paste them.



You can press arrow keys to scroll through the clipboard entries, and return/enter to paste in an entry.

You can use the search field to filter the clipboards.

You can mark entries as favorites, in which case they will never be removed from the clipboard history.

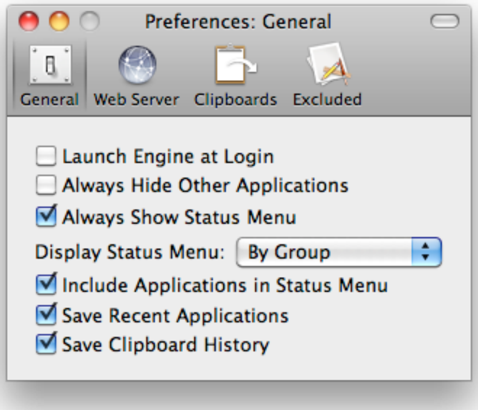
You can send clipboards to another Mac running Keyboard Maestro. The received clipboard entries will appear in the clipboard history (assuming the Keyboard Maestro Web Server is enabled on the destination Mac).

By default, Keyboard Maestro creates a Clipboard History Switcher macro in the “Switcher Group” Macro Group, triggered by Command-Control-Shift-V. You can disable the Macros by selecting them in the Macros window pane and clicking the ✓ button.

Keyboard Maestro also creates disabled macros for Paste Plain Text (Command-Shift-V) and Paste Previous Clipboard (Command-Control-V) in the “Global Macro Group” Macro Group. You can enable these by selecting the Global Macros Group, then selecting the macros and and clicking the ✓ button below the Macros list.

Preferences

To configure Keyboard Maestro, first launch Keyboard Maestro and choose Preferences from the Keyboard Maestro menu.



□ Launch Engine at Login
□ Always Hide Other Applications
☑ Always Show Status Menu
Display Status Menu: By Group
☑ Include Applications in Status Menu
☑ Save Recent Applications
☑ Save Clipboard History

The preferences are divided into sections.

General Preferences

In the General preference pane you can:

- enable or disable launching the Keyboard Maestro Engine at Login.
- configure whether to always hide other applications when switching applications.
- configure whether to always show the status menu.
- configure the style of the status menu.
- configure whether to include applications in the status menu.
- configure whether to save the recent applications between logins.
- configure whether to save the clipboard history between logins.

Web Server Preferences

In the Web Server preference pane you can:

- enable or disable the web server.
- configure the username, password and port of the web server.
- enable or disable web browser access.
- enable or disable iPhone access.
- enable or disable receiving clipboard entries.
- enable or disable replacing the current clipboard.
- access the web server in your default browser by clicking on Connect.

The web server is disabled by default.

If the web server and web browsing are enabled, then anyone who can connect to your Mac can execute any macro that has a Public Web trigger.

If the web server and web browsing are enabled, and if you have configured a username and password, then anyone who can connect to your Mac and login with the specified username and password can execute any of your macros.

If the web server and iPhone access are enabled, and if you have configured a username and password, then anyone who can connect to your Mac from an iPhone and login with the specified username and password can execute any of your macros.

If the web server and receiving clipboard are enabled, then anyone can send you clipboards which will appear in your clipboard history. By default they do not overwrite the current clipboard, but you can enable that to allow the current clipboard to be directly written remotely.

Macros are still only available when they are enabled and their containing macro group is enabled and active.

You can configure custom styles using the defaults write command to add a custom style, for example:

```
defaults write com.stairways.keyboardmaestro.editor WebServerCustomStyles -string 'body { background: green; }'
```

This might be useful if you are controlling multiple Macs and want to differentiate them more clearly.

Clipboards Preferences

In the Clipboards preference pane you can add, remove and rename Named Clipboards.

Named Clipboards store snippets or text or images (or anything the clipboard can hold) and you can copy or paste from them using the Clipboard Switcher or using appropriate macro actions.

Excluded Preferences

In the Excluded preference pane you can add and remove applications from the global excluded applications list. Excluded applications will not be shown in the Application Switcher, and are (optionally) not hidden or quit by the Hide All Applications or Quit All Applications actions.

Other Hidden Preferences

You can configure various preferences using defaults write from the Mac OS X Terminal. Some preferences will take effect immediately, but others may require the engine to be relaunched.

You can set the maximum number of items in the clipboard history (default 100)

```
defaults write com.stairways.keyboardmaestro.engine MaxClipboardHistory -int 100
```

You can set a delay between each action execution (default 0.0)

```
defaults write com.stairways.keyboardmaestro.engine InterActionDelay -float 0.5
```

You can set the delay to wait after pasting before continuing to the next action (default 0.2)

```
defaults write com.stairways.keyboardmaestro.engine ClipboardDelay -float 0.2
```

You can set the delay to wait after simulating a keystroke before continuing to the next action (default 0.01)

```
defaults write com.stairways.keyboardmaestro.engine SimulateKeystrokeDelay -float 0.01
```

If the command key is down, the delay is stored in the SimulateKeystrokeCommandKeyDelay preference (default 0.15)

```
defaults write com.stairways.keyboardmaestro.engine SimulateKeystrokeCommandKeyDelay -float 0.15
```

You can set the command line tool that is used to execute AppleScripts using:

```
defaults write com.stairways.keyboardmaestro.engine OSAScriptCommand "/usr/bin/arch -i386 /usr/bin/osascript"
```

You can disable all animation in the editor with:

```
defaults write com.stairways.keyboardmaestro.editor DisableAnimation -bool YES
```

or the engine with:

```
defaults write com.stairways.keyboardmaestro.engine DisableAnimation -bool YES
```

You can adjust the spelling of Favorites with:

```
defaults write com.stairways.keyboardmaestro.editor FavoritesDisplayName "Favourites"
```

Note: The Mac App Store engine uses bundle ID com.stairways.keyboardmaestro.editor.engine instead of com.stairways.keyboardmaestro.engine for reasons only Apple can explain.

Scripting

Controlling Keyboard Maestro via Scripting

The primary scripting interface to Keyboard Maestro is the Keyboard Maestro Engine's do script support. You can ask Keyboard Maestro to:

- execute a macro by name
- execute a macro by unique ID
- execute an action given its XML code

Note in most cases you must ask "Keyboard Maestro Engine", not "Keyboard Maestro".

The easiest way is to use the name, for example:

```
tell application "Keyboard Maestro Engine"
  do script "Name of Your Macro"
end tell
```

The macro must be defined and enabled, and the macro group must be enabled and currently active.

If there is more than one macro with the same name, you will get an error, so you can use a UID instead of a name.

```
tell application "Keyboard Maestro Engine"
  do script "D0C150C7-8A0C-4837-918A-427E2BCFB6B9"
end tell
```

The do script will not return until the macro is finished executing.

You can determine a macro's UID by selecting it and choosing Copy UID command in the Edit menu.

An even more powerful way to script Keyboard Maestro is to execute specific actions based on their XML code. This allows you to construct any action, including changing the action on the fly, without having to create a macro first. A simple example would be:

```
tell application "Keyboard Maestro Engine"
```

```
  do script "<dict><key>MacroActionType</key><string>SwitchToLastApplication</string></dict>"
end tell
```

The easiest way to determine the appropriate XML is to create an example action in an example macro and then export the macro.

You can disable or enable a Macro or Macro Group from AppleScript with:

```
tell application "Keyboard Maestro"
  setMacroEnable "Macro/Macro Group Group Name or UID" with/without enable
end tell
```

This actually asks the editor to disable or enable the macro or macro group, so the change is both visible and permanent.

Alternatively you can use the Set Macro Enable action.

You can start editing a Macro or Macro Group from AppleScript with:

```
tell application "Keyboard Maestro"
  editMacro "Macro/Macro Group Name or UID"
end tell
```

You can ask Keyboard Maestro Engine to reload the macros with:

```
tell application "Keyboard Maestro Engine"
  reload
end tell
```

Executing Scripts

You can execute shells scripts, AppleScripts, Automator Workflows, or filter the clipboard using BBEdit Text Factories.

AppleScripts and shell scripts give you a powerful way of adding new facilities we have not specifically provided for, as well as controlling other applications.

The results of AppleScripts and shell scripts can be displayed, or they can be typed or pasted in to the current selection, or saved into a variable. This allows you to insert text that depends on many factors, such as date calculations, file listings, SQL queries, web pages, or anything else you can imagine.

You can also use the clipboard to pass data between actions. For example, a script can use pbpaste to read the current clipboard, and pbcopy to set the current clipboard. You can use the Delete Current Clipboard action to restore the clipboard afterwards.

Shell scripts can access variables as environment variables with names like $KMVAR_Variable_Name (KMVAR_ is prefixed, and spaces are converted to underscores).

AppleScript can read or write variables with:

```
tell application "Keyboard Maestro Engine"
  set kmVarRef to make variable with properties {name:"Calculation Result"}
  set oldValue to value of kmVarRef
  set value of kmVarRef to 10
end tell
```

You can delete a variable with:

```
tell application "Keyboard Maestro Engine"
  delete variable "Calculation Result"
end tell
```

AppleScripts are executed in the background via osascript. This means they are not allowed to do user interaction. You can work around this by asking an application like System Events to do the user interaction for you, for example:

```
tell application "System Events"
  activate
  display dialog "Hello"
end tell
```

The osascript tool will execute in 64-bit mode if available, which may be a problem if you have old versions of AppleScript extensions installed. However, you can set the command line tool that is used to execute AppleScripts as described in Other Hidden Preferences.

See also the Variables section.

Enhancing AppleScript

Keyboard Maestro Engine makes several of its facilities available to AppleScript.

You can ask it to play a sound with:

```
tell application "Keyboard Maestro Engine"
  play sound alias "Harddisk:System:Library:Sounds:Glass.aiff"
end tell
```

You can ask Keyboard Maestro Engine to perform a calculation for you with:

```
tell application "Keyboard Maestro Engine"
  set n to calculate "JULIANDATE()"
end tell
```

You can ask Keyboard Maestro Engine to process tokens for you with:

```
tell application "Keyboard Maestro Engine"
  set clip to process tokens "%PastClipboard%3%"
end tell
```

See also the Calculations and Text Tokens sections.

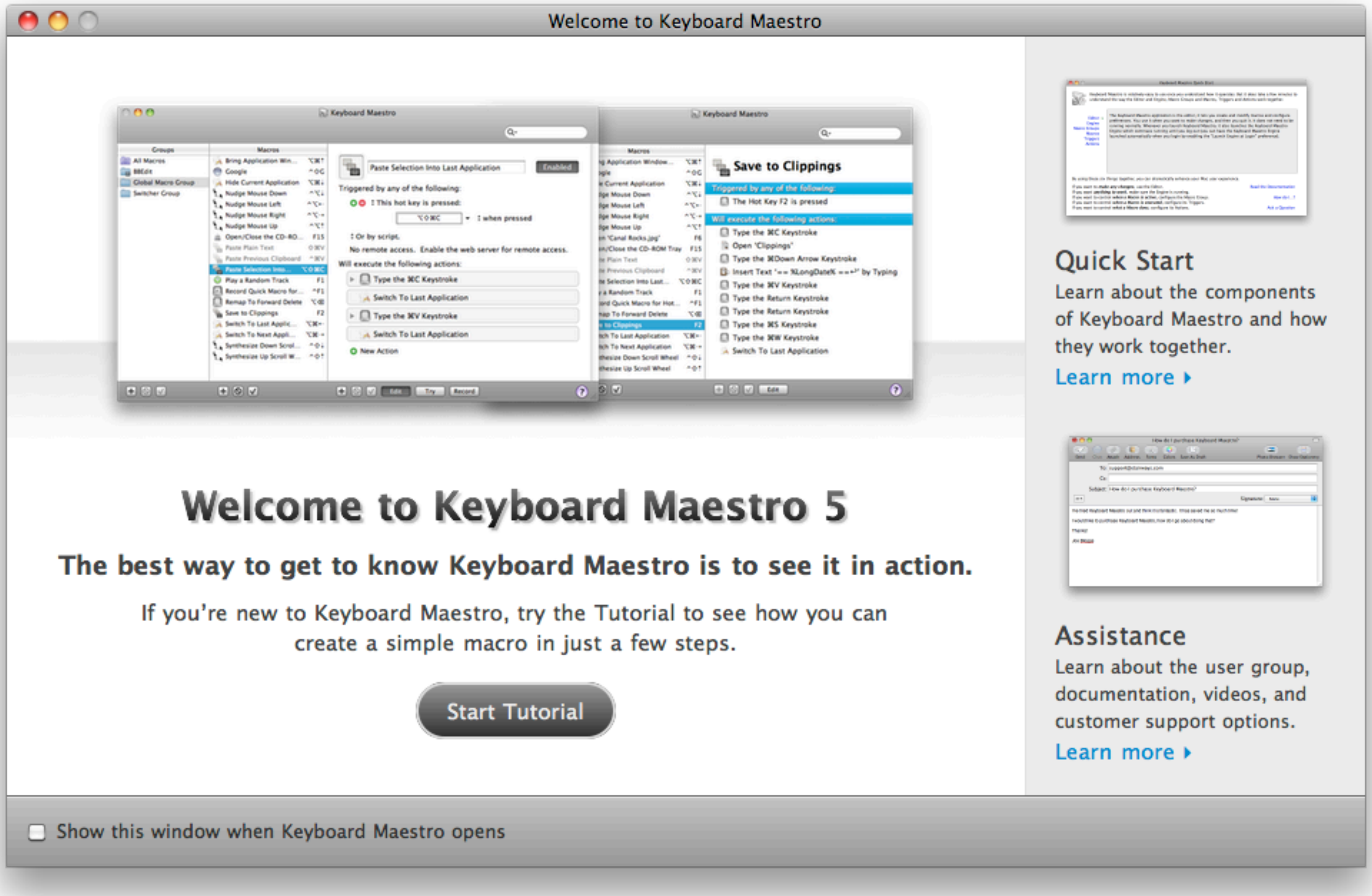
Windows

- Welcome Window
- Macros Window
- Tutorial
- Macro Group Editor
- Macro Editor Window
- Macro Library Window
- Recording Window
- Application Launcher Window
- Application Switcher Window
- Window Switcher Window
- Clipboard Switcher Window
- Clipboard History Switcher Window
- Preferences Window
- Preferences General Pane
- Preferences Web Server Pane

- Preferences Clipboards Pane
- Preferences Exclude Pane
- About Window Pane

Welcome Window

This window welcomes new users and gives you some options for learning about Keyboard Maestro.



Welcome to Keyboard Maestro

Welcome to Keyboard Maestro 5

The best way to get to know Keyboard Maestro is to see it in action.

If you're new to Keyboard Maestro, try the Tutorial to see how you can create a simple macro in just a few steps.

Start Tutorial

Quick Start

Learn about the components of Keyboard Maestro and how they work together.

Learn more ▸

Assistance

Learn about the user group, documentation, videos, and customer support options.

Learn more ▸

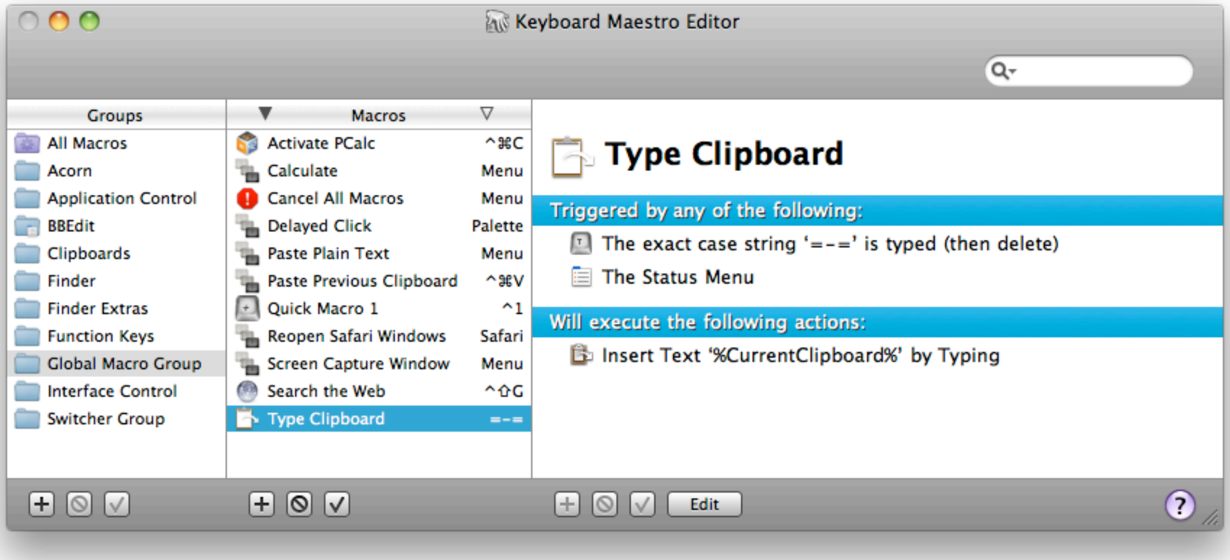
□ Show this window when Keyboard Maestro opens

If you are new to Keyboard Maestro, start the tutorial and Keyboard Maestro will walk you through creating a simple macro.

Macros Window

This window lets you manipulate Macros and Macro Groups, creating new ones, deleting old ones, enabling and disabling them, editing them and so on.

You get this window pane by launching Keyboard Maestro.



Keyboard Maestro Editor

Groups	Macros	
All Macros	Activate PCalc	^⌘C
Acorn	Calculate	Menu
Application Control	Cancel All Macros	Menu
BBEdit	Delayed Click	Palette
Clipboards	Paste Plain Text	Menu
Finder	Paste Previous Clipboard	^⌘V
Finder Extras	Quick Macro 1	^1
Function Keys	Reopen Safari Windows	Safari
Global Macro Group	Screen Capture Window	Menu
Interface Control	Search the Web	^⇧G
Switcher Group	Type Clipboard	=-=

Type Clipboard

Triggered by any of the following:

The exact case string '=-=' is typed (then delete)

The Status Menu

Will execute the following actions:

Insert Text '%CurrentClipboard%' by Typing

The window contains a list of Macro Groups and their associated Macros.

You can create a new Macro Group by clicking the + button below the Groups list.

You can create a new Macro by selecting a macro group and then clicking the + button below the Macros list.

You can see the selected Macro Group or Macro in the right hand column, and edit it by clicking the Edit button.

You can select the All Macros meta-Group to show all Macros, and you can use the search field to filter down the list of macros.

You can rename a Macro Group or Macro by double clicking it and changing the title.

You can delete a Macro Group by selecting it and clicking the ⊘ button below the Groups list.

You can delete a Macro by selecting it and clicking the ⊘ button below the Macros list.

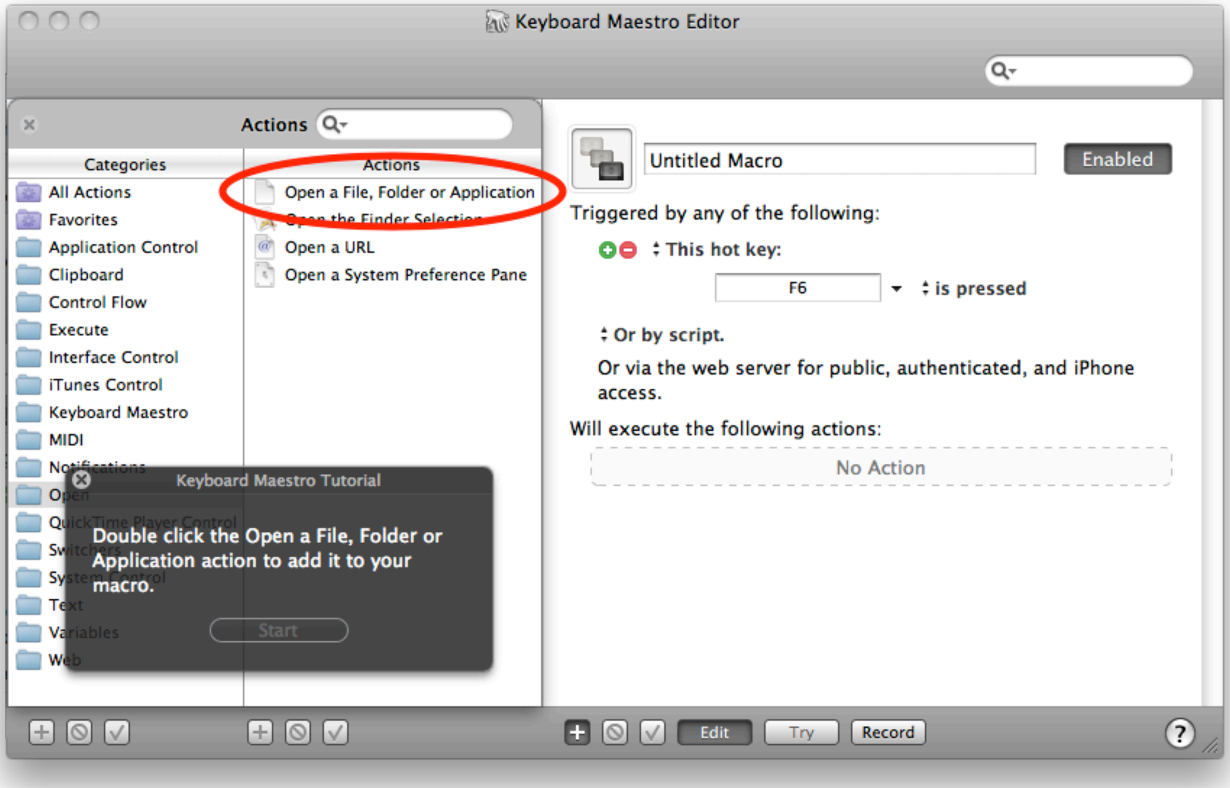
Similarly, you can enable or disable Macro Groups or Macros by clicking their respective ✓ button.

You cannot delete, rename or modify the Global Macro Group.

See also the Macro Groups, Macros and Macro Editor Window sections.

Tutorial

If you are new to Keyboard Maestro, start the tutorial by clicking the Start Tutorial button in the Welcome window or by choosing Tutorial from the Help menu and Keyboard Maestro will walk you through creating a simple macro.



Untitled Macro

Triggered by any of the following:

This hot key:

F6 is pressed

Or by script.

Or via the web server for public, authenticated, and iPhone access.

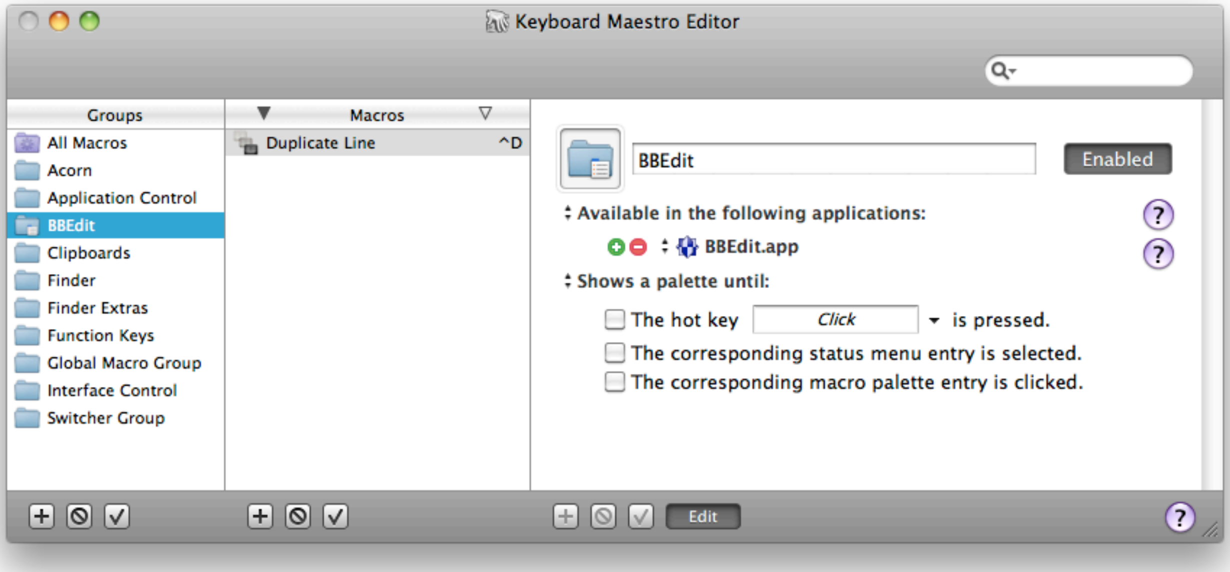
Will execute the following actions:

Double click the Open a File, Folder or Application action to add it to your macro.

Follow the instructions. Keyboard Maestro will hilight the location of the various buttons to help you quickly create a macro. You can even use the tutorial as a wizard to create a hot key triggered macro to perform any of Keyboard Maestro’s many actions.

Macro Group Editor

To edit a Macro Group, select it and click the Edit button. Its details will be shown in the right hand column. You can edit its name, control which applications it is available in, and how it will be activated.



Groups: All Macros, Acorn, Application Control, BBEdit, Clipboards, Finder, Finder Extras, Function Keys, Global Macro Group, Interface Control, Switcher Group

Macros: Duplicate Line ^D

BBEdit

Available in the following applications:

BBEdit.app

Shows a palette until:

□ The hot key Click is pressed.

□ The corresponding status menu entry is selected.

□ The corresponding macro palette entry is clicked.

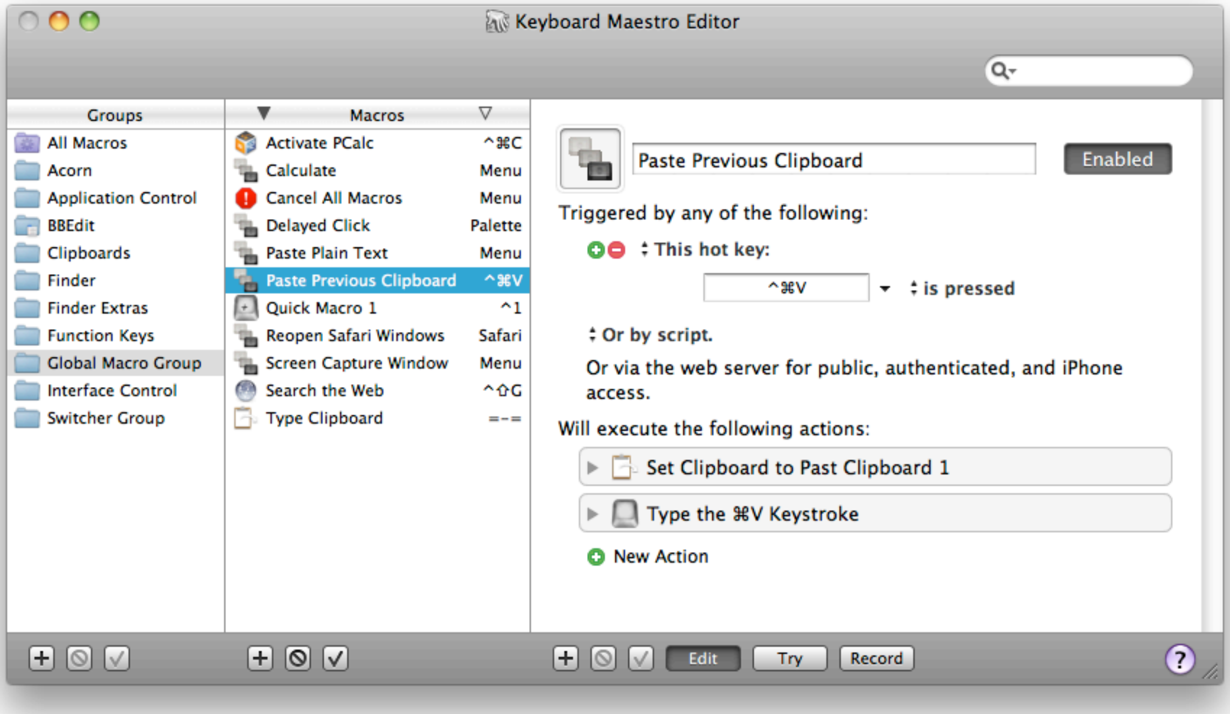
Typically a group's macros would be available everywhere (available in all applications), or it might be specific to a particular application (available in the following applications) in which case you might name the Macro Group after the application.

You can also configure the macro group to be activated only after a Hot Key press (either for a single use or toggled on and off), and whether to display the macros in a floating palette.

See also the Macros section.

Macro Editor Window

To edit a Macro, select it and click the Edit button. Its details will be shown in the right hand column. You can edit its name, add or remove triggers, and configure its action list.



Groups: All Macros, Acorn, Application Control, BBEdit, Clipboards, Finder, Finder Extras, Function Keys, Global Macro Group, Interface Control, Switcher Group

Macros	
Activate PCalc	^⌘C
Calculate	Menu
Cancel All Macros	Menu
Delayed Click	Palette
Paste Plain Text	Menu
Paste Previous Clipboard	^⌘V
Quick Macro 1	^1
Reopen Safari Windows	Safari
Screen Capture Window	Menu
Search the Web	^⇧G
Type Clipboard	=-=

Paste Previous Clipboard

Triggered by any of the following:

This hot key:

^⌘V is pressed

Or by script.

Or via the web server for public, authenticated, and iPhone access.

Will execute the following actions:

Set Clipboard to Past Clipboard 1

Type the ⌘V Keystroke

To add a trigger, click the green ⊕ button and select the type of trigger. To remove a trigger, click the red ⊖ button.

To see how to execute this macro via a script, select from the "Or by script"' menu. How you can execute the macro remotely is also displayed.

To add an action, click the New Action button, or equivalently the + button below the detail view. This will show the lists of possible actions. Double click one or more of them to add actions to the action list for this macro.

You can also Copy and Paste actions, as well as drag them around to rearrange them. Use the ⃠ button and ✓ button to delete or enable/disable the selected actions.

You can try the selected actions by clicking the Try button.

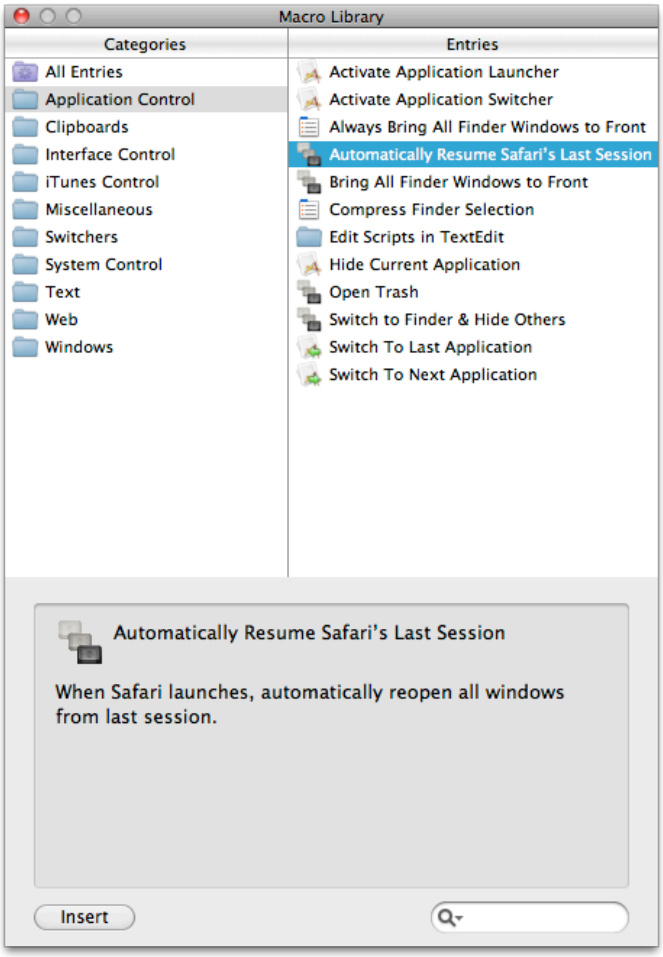
You can click the Record button to record your actions.

To learn more about creating or editing Macros, see the Macros section.

Macro Library Window

This window contains examplke and template macros you can add to your macro collection. You can use the macros as is, or edit them to customize them for your particular needs.

You get this window by choosing Macro Library from the Window menu.



Categories	Entries
All Entries	Activate Application Launcher
Application Control	Activate Application Switcher
Clipboards	Always Bring All Finder Windows to Front
Interface Control	Automatically Resume Safari's Last Session
iTunes Control	Bring All Finder Windows to Front
Miscellaneous	Compress Finder Selection
Switchers	Edit Scripts in TextEdit
System Control	Hide Current Application
Text	Open Trash
Web	Switch to Finder & Hide Others
Windows	Switch To Last Application
	Switch To Next Application

Automatically Resume Safari's Last Session

When Safari launches, automatically reopen all windows from last session.

Each entry represents one or more macros, usually in a single macro group, but occasionally in more than one macro group. You can learn about them by selecting them, and then you can insert them into your macros by clicking the Insert button or by dragging them to a particular macro group (dragging is not available if the library entry represents more than one macro group as you can't drag to two macro groups).

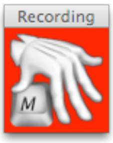
You can add macros to your library by using the File -> Export as Macro Library and selecting the Add to Library checkbox. You can get Macro Library entries from us or from other Keyboard Maestro users and add them to you library by double clicking them or by choosing Import to Macro Library from the File menu.

Remember to use caution when installing a macro or macro library from anyone - macros can potentially do a lot of damage and compromise the security of your Mac, so only install macros from trusted sources.

Recording Window

This window shows you when Keyboard Maestro is recording your actions.

You get this window by clicking the Record button in the Macro Editor window or by triggering a Record Quick Macro action.



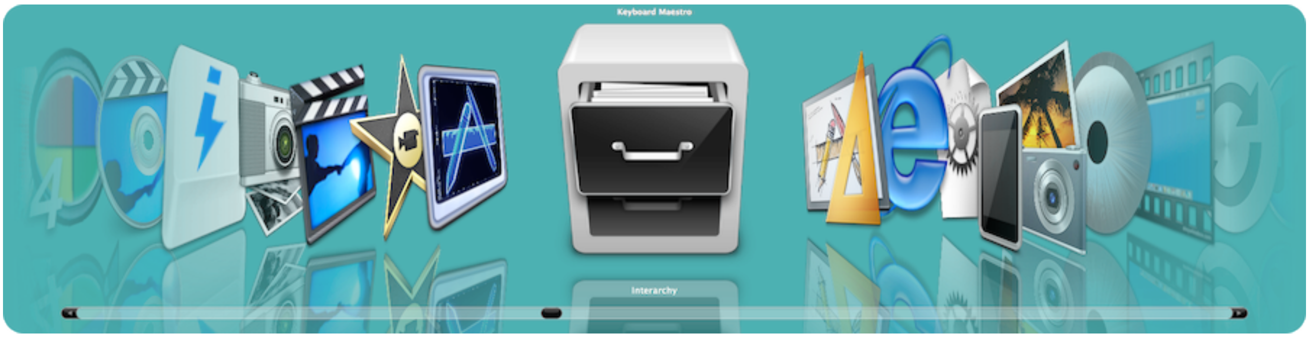
Clicking on this window will stop all recording.

To learn more about recording, see the Recording section.

Application Launcher Window

This window lets you launcher applications.

You get this window by triggering the Activate Application Launcher macro.

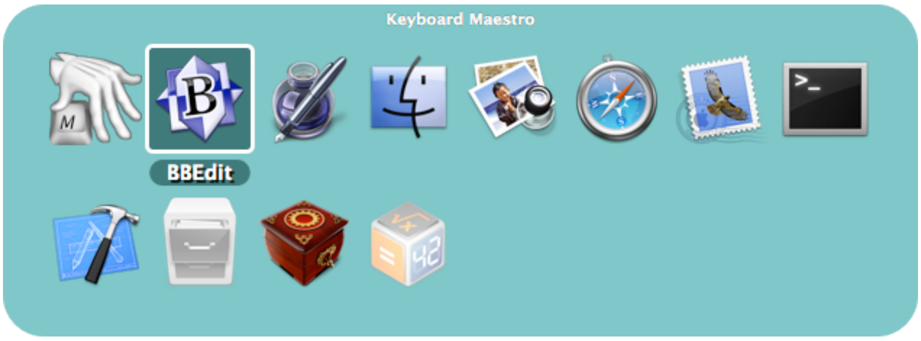


To learn more about the Application Launcher, see the Application Launcher section.

Application Switcher Window

This window lets you switch between active applications, as well as quit or hide applications or quickly launch frequently used applications.

You get this window by triggering the Activate Application Switcher macro.

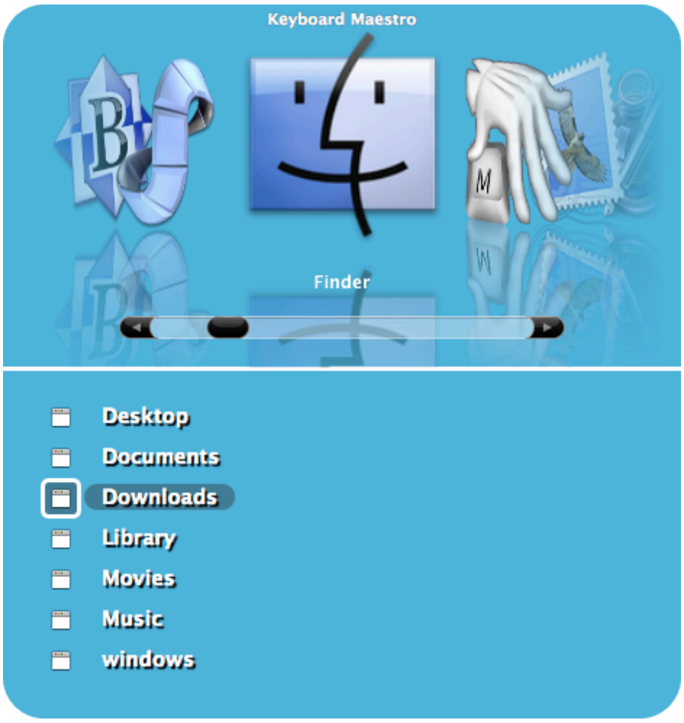


To learn more about the Application Switcher, see the Application Switcher section.

Window Switcher Window

This window lets you switch between windows in the current application, as well as close or minimize windows.

You get this window by triggering the Activate Window Switcher macro.

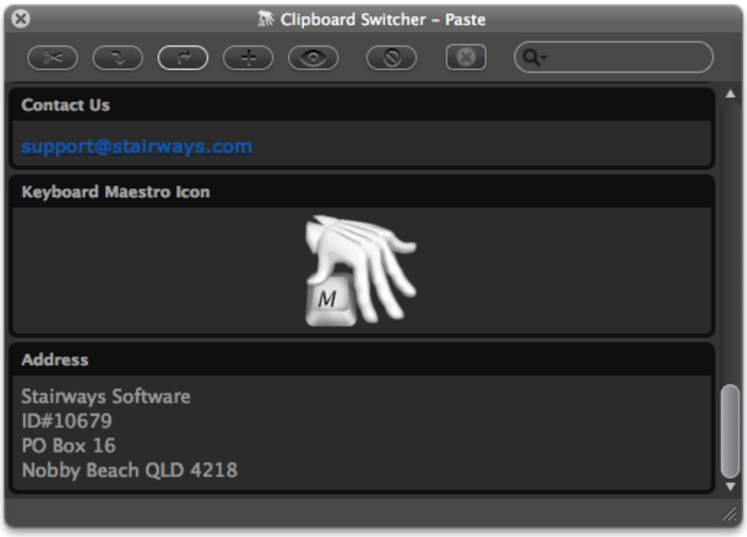


To learn more about the Window Switcher, see the Window Switcher section.

Clipboard Switcher Window

This window lets you select between named clipboards to Cut, Copy or Paste to/from.

You get this window by triggering one of the Activate Clipboard Switcher macros.



Contact Us

support@stairways.com

Keyboard Maestro Icon

Address

Stairways Software
ID#10679
PO Box 16
Nobby Beach QLD 4218

Select something and select a named clipboard and click the Cut button or Copy button to cut/copy to a named clipboard. Select a named clipboard and click the Paste button button to paste a named clipboard. Click the + button to create a new named clipboard. Select a named clipboard and click the Quick Look button to view it. Select a named clipboard and click the ⊘ button to delete it.

Click the x button to toggle whether the window should close after an action.

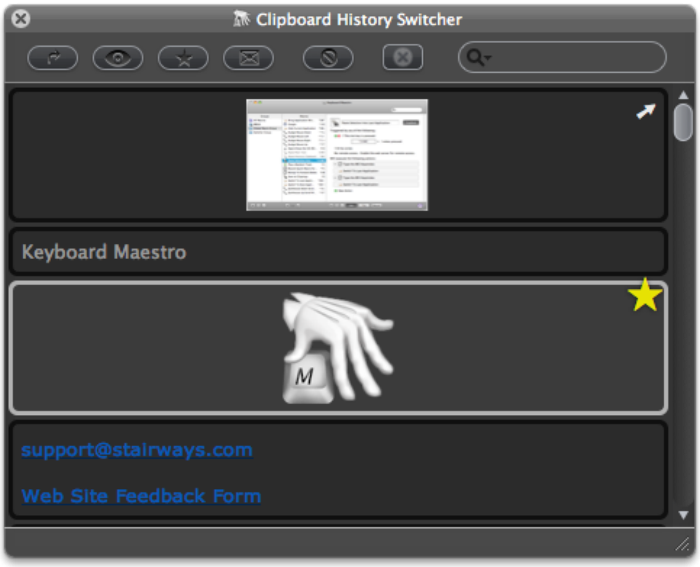
Use the search field to filter the named clipboards.

To learn more about the Clipboard Switcher, see the Clipboard Switcher section.

Clipboard History Switcher Window

This window lets you paste from your clipboard history of items that you have previously cut or copied.

You get this window by triggering the Clipboard History Switcher macro.



Keyboard Maestro

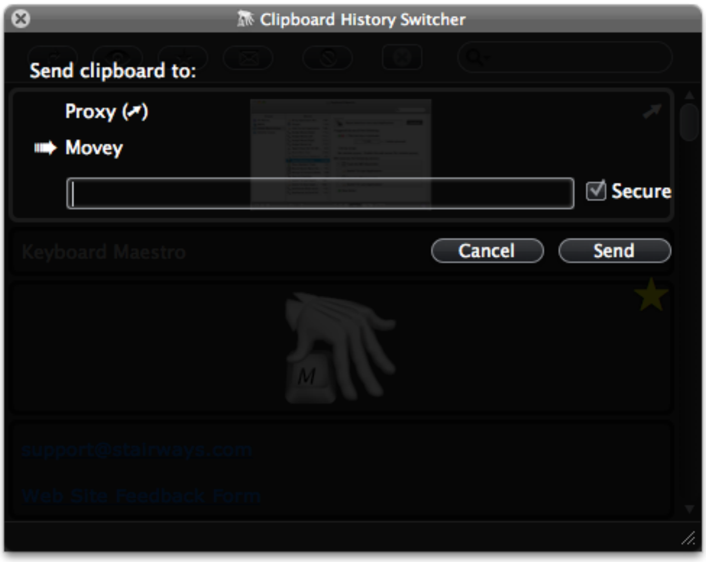
support@stairways.com

Web Site Feedback Form

Cut or copy something and it will appear in this list. Select an item and click the Paste button button to paste a named clipboard. Select an item and click the ★ button (or press Command-L) to mark it as a favorite, or click the ✉ button (or press Command-S) to send it to another Mac. Select an item and click the ⊘ button to delete it (this is useful if you want to delete a password or other sensitive information).

To view an item more fully, select it and click the Quick Look button (or press Space) to display the Quick Look window.

To send a clipboard to another Mac running Keyboard Maestro, click the ✉ button to display the sending window.



Select the desired local destination, or type a host:port destination to send it to a remote Mac, and click the send button. Keyboard Maestro will keep trying to send to the Mac even if it can't connect right now, so as long as both Macs are connected to the Internet eventually, the clipboard should get through. While a clipboard is being sent, it is marked with a progress icon. If a clipboard is marked to be sent in the future, it is marked with a ➶. Once it has been successfully sent, it is marked with a ↗. If it fails to be sent, and Keyboard Maestro has given up, it is marked with a !. Clipboards that have been received are marked with a ✉.

Click the x button to toggle whether the window should close after an action.

Use the search field to filter the named clipboards.

To learn more about the Clipboard History Switcher, see the Clipboard History Switcher section.

Preferences Window

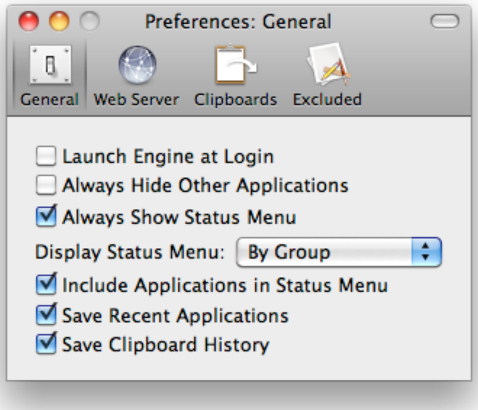
This window lets you configure Keyboard Maestro.

You get this window by launching Keyboard Maestro and choosing Preferences from the Keyboard Maestro menu.

To learn more about the Preferences, see the Preferences section.

Preferences General Pane

This window pane lets you configure general preferences.

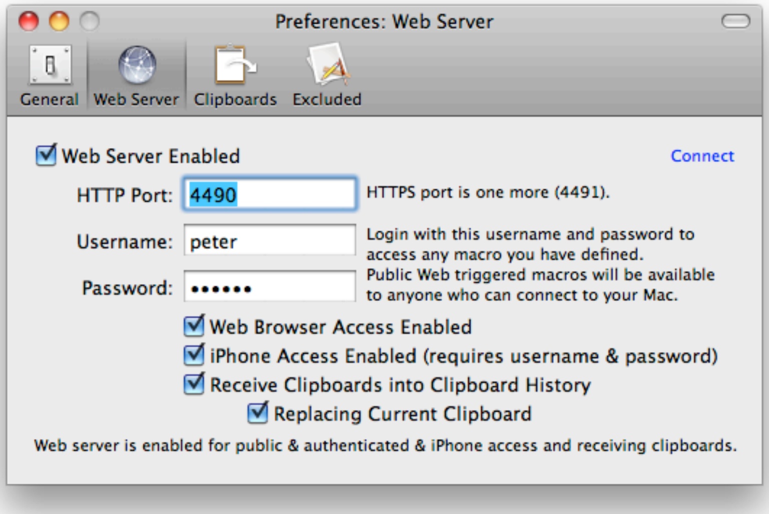


Preferences: General

- ☐ Launch Engine at Login
- ☐ Always Hide Other Applications
- ☑ Always Show Status Menu
- Display Status Menu: By Group
- ☑ Include Applications in Status Menu
- ☑ Save Recent Applications
- ☑ Save Clipboard History

Preferences Web Server Pane

This window pane lets you configure the built-in web server which enables remote execution of macros.



Preferences: Web Server

☑ Web Server Enabled

HTTP Port: 4490 HTTPS port is one more (4491).

Username: peter

Password: ••••••

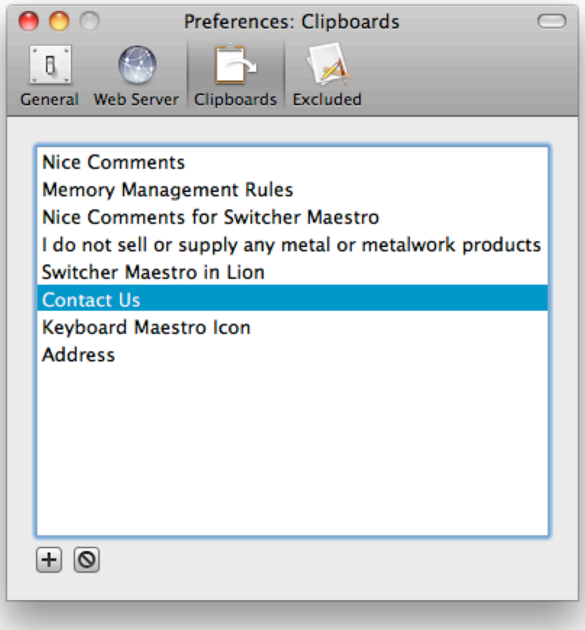
Login with this username and password to access any macro you have defined. Public Web triggered macros will be available to anyone who can connect to your Mac.

- ☑ Web Browser Access Enabled
- ☑ iPhone Access Enabled (requires username & password)
- ☑ Receive Clipboards into Clipboard History
  - ☑ Replacing Current Clipboard

Web server is enabled for public & authenticated & iPhone access and receiving clipboards.

Preferences Clipboards Pane

This window pane lets you add, remove and rename Named Clipboards.

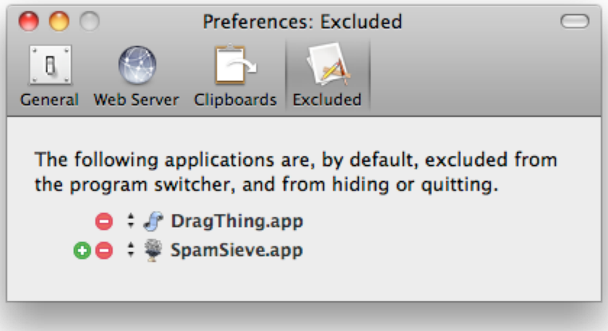


Preferences: Clipboards

Nice Comments
Memory Management Rules
Nice Comments for Switcher Maestro
I do not sell or supply any metal or metalwork products
Switcher Maestro in Lion
Contact Us
Keyboard Maestro Icon
Address

Preferences Exclude Pane

This window pane lets you add and remove applications from the global Excluded Applications list.



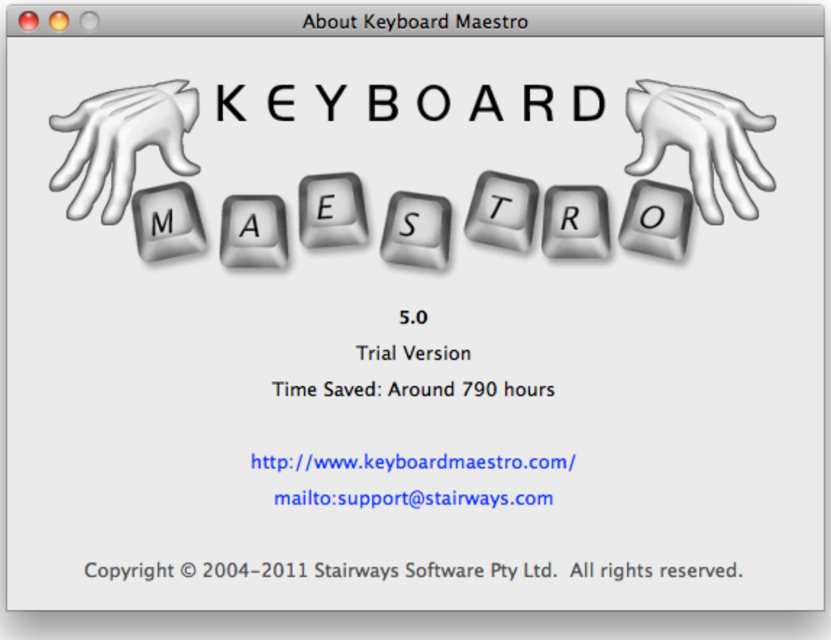
The following applications are, by default, excluded from the program switcher, and from hiding or quitting.

- DragThing.app
- SpamSieve.app

About Window Pane

This window shows you the version of this copy of Keyboard Maestro, to whom it is registered, and allows you to visit the web site.

You get this window by launching Keyboard Maestro and choosing About Keyboard Maestro from the Keyboard Maestro menu.



5.0

Trial Version

Time Saved: Around 790 hours

http://www.keyboardmaestro.com/

mailto:support@stairways.com

Copyright © 2004-2011 Stairways Software Pty Ltd. All rights reserved.

Menus

- Keyboard Maestro
  - About Keyboard Maestro
  - Purchase Keyboard Maestro
  - Register Keyboard Maestro
  - Check For Update
  - Preferences
  - Services
  - Hide Keyboard Maestro
  - Hide Others
  - Show All
  - Quit Keyboard Maestro
- File
  - New Macro Group
  - New Macro
  - Close
  - Export Macros
  - Export as Macro Library
  - Import Macros
  - Import to Macro Library
  - Revert Macros
  - Launch Engine
  - Quit Engine
- Edit
  - Undo
  - Redo
  - Cut
  - Copy
  - Copy UID
  - Paste

Keyboard Maestro

The Keyboard Maestro menu contains menu items relating to the Keyboard Maestro application as a whole.

Keyboard Maestro » About Keyboard Maestro

The About Keyboard Maestro command in the Keyboard Maestro menu displays the About Keyboard Maestro window.

Keyboard Maestro » Purchase Keyboard Maestro

The Purchase Keyboard Maestro command in the Keyboard Maestro menu lets you purchase Keyboard Maestro online.

Keyboard Maestro » Register Keyboard Maestro

The Register Keyboard Maestro command in the Keyboard Maestro menu displays the serial number entry window allowing you to enter your username (email address) and serial number. Make sure you enter them exactly as sent to you.

Keyboard Maestro » Check For Update

The Check For Update command in the Keyboard Maestro menu checks to see if there are any updates to Keyboard Maestro and offers to download and install them if there are.

Keyboard Maestro » Preferences

The Preferences command in the Keyboard Maestro menu displays the Preferences window.

Keyboard Maestro » Services

The Services command in the Keyboard Maestro menu is used to perform Mac OS X Services which are shared functions available across multiple applications. You can learn more about Mac OS X Services from your Mac OS X documentation, and you can install new services which will work with Keyboard Maestro. Keyboard Maestro includes full support for Services, so relevant Services on your system are available in Keyboard Maestro.

Keyboard Maestro » Hide Keyboard Maestro

The Hide Keyboard Maestro command in the Keyboard Maestro menu will hide the Keyboard Maestro application and all its windows. Click on Keyboard Maestro's Dock icon or choose Show All to show Keyboard Maestro again.

Keyboard Maestro » Hide Others

The Hide Others command in the Keyboard Maestro menu will hide all other applications. Choose Show All to show them again.

Keyboard Maestro » Show All

The Show All command in the Keyboard Maestro menu will show all hidden applications.

Keyboard Maestro » Quit Keyboard Maestro

The Quit Keyboard Maestro command in the Keyboard Maestro menu will Quit Keyboard Maestro. the Keyboard Maestro Engine will remain running and all enabled Keyboard Maestro features will continue to operate (unless you have specifically quit the Keyboard Maestro Engine).

File

The File menu is where you import or export Macros or launch or quit the Keyboard Maestro Engine.

File » New Macro Group

The New Macro Group command in the File menu creates and starts editing a new macro group.

File » New Macro

The New Macro command in the File menu creates and starts editing a new macro.

File » Close

The Close command in the File menu closes the front window.

File » Export Macros

The Export Macros command in the File menu exports the selected macros to a file that you can import on another Mac. This is one way to transfered macros from one Mac to another.

File » Export as Macro Library

The Export as Macro Library command in the File menu exports the selected macros to a library file that you can share with others. If you create any interesting macros please consider sending them to us and we will make them available on our web site or in a future version of Keyboard Maestro.

See also the Macro Library section.

File » Import Macros

The Import Macros command in the File menu lets you select a saved macro file and imports the macros it contains.

File » Import to Macro Library

The Import to Macro Library command in the File menu lets you import a shared macro library file into your macro library. Macros in your library are not active, but can be added into one or more macro groups to become active.

See also the Macro Library section.

File » Revert Macros

The Revert Macros command in the File menu lets you revert to a previous version of your macros. If you find you have really messed up your macros, you can revert to how they were when you first launched Keyboard Maestro, or how they were yesterday or even several days ago.

File » Launch Engine

The Launch Engine command in the File menu lets you start the Keyboard Maestro Engine manually. The Keyboard Maestro Engine performs all the Macro, Application Switcher, Window Switcher and Clipboard Switcher functions even while Keyboard Maestro itself is not running. It is launched automatically as a Startup Item when you login (assuming you have not disabled that in the Preferences window) or any time you launch Keyboard Maestro. If it is not running for any reason you can start it manually with this command. This menu item only exists while the Keyboard Maestro Engine is not running.

File » Quit Engine

The Quit Engine command in the File menu lets you quit the Keyboard Maestro Engine. The Keyboard Maestro Engine performs all the Macro, Application Switcher, Window Switcher and Clipboard Switcher functions even while Keyboard Maestro itself is not running. It is launched automatically as a Startup Item when you login (assuming you have enabled that in the Preferences window). If you quit the Keyboard Maestro Engine these functions will no longer operate. This menu item only exists while the Keyboard Maestro Engine is running.

Edit

The Edit menu contains menu items relating to text and selections.

Edit » Undo

The Undo command in the Edit menu undoes the previous command.

Edit » Redo

The Redo command in the Edit menu redoes the previous undone command.

Edit » Cut

The Cut command in the Edit menu copies the current selection to the system clipboard and then deletes the selection.

Edit » Copy

The Copy command in the Edit menu copies the current selection to the system clipboard.

Edit » Copy UID

The Copy UID command in the Edit menu copies the current selected macro or macro group’s UID.

Edit » Paste

The Paste command in the Edit menu pastes the current system clipboard into the current selection.

Edit » Delete

The Delete command in the Edit menu deletes the current selection.

Edit » Select All

The Select All command in the Edit menu selects all text or items.

Edit » Duplicate

The Duplicate command in the Edit menu duplicates the selected items.

View

The View menu contains menu items relating to display and actions.

View » Sort by Macro Name

The Sort by Macro Name command in the View menu sorts the macros in the main window by name.

View » Sort by Macro Trigger

The Sort by Macro Trigger command in the View menu sorts the macros in the main window by trigger. This is useful to see what hot keys are available, or to group all Typed String triggers together.

View » Edit

The Edit command in the View menu toggles Edit mode on and off. Edit mode allows you to modify macros and macro groups. You may prefer to leave it on permanently, or you may like to turn it off when you are not making changes to give a more concise and visually appealing view of the macros and macro groups.

View » Toggle Enable

The Toggle Enable command in the View menu toggles the enable of the selected items.

View » Show Actions

The Show Actions command in the View menu shows the action list. It is available while editing a macro. Double click or drag actions from the action list to add them to your macro. This menu toggles to Hide Actions when the action list is already showing.

View » Set Action Time Out

The Set Action Time Out command in the View menu sets the time out time and behaviour for the selected actions. You can configure how long an action is allowed to execute for before it is aborted, and whether the macro should continue or be canceled if the action times out.

View » Try

The Try command in the View menu tries the selected actions. It is available while editing a macro.

View » Record

The Record command in the View menu toggles recording on and off. It is available while editing a macro.

Window

The Window menu contains menu items relating to windows.

Window » Minimize

The Minimize command in the Window menu minimizes the front window.

Window » Zoom

The Zoom command in the Window menu zooms the front window.

Window » Keyboard Maestro Editor

The Keyboard Maestro Editor command in the Window menu brings the main Keyboard Maestro macro editing window to the front.

Window » Macro Library

The Macro Library command in the Window menu shows or hides the macro library.

See also the Macro Library section.

Window » Bring All to Front

The Bring All to Front command in the Window menu brings all Keyboard Maestro windows to the front.

Help

The Help menu contains menu items relating to Help.

Help » Keyboard Maestro Documentation

The Keyboard Maestro Documentation command in the Help menu displays the Keyboard Maestro documentation.

Help » Keyboard Maestro Quick Start

The Keyboard Maestro Quick Start command in the Help menu displays the Keyboard Maestro quick start help, which quickly gets you up to speed in using Keyboard Maestro.

Help » Welcome to Keyboard Maestro

The Welcome to Keyboard Maestro command in the Help menu displays the Welcome message, giving you a quick overview of what support resources are available for helping you get started using Keyboard Maestro.

Help » Tutorial

The Tutorial command in the Help menu starts the in-application tutorial. The tutorial will walk you through creating a simple macro. By varying the actions slightly, you can create a variety of macros that are triggered by hot keys and that open various documents.

Help » Videos

The Videos command in the Help menu displays the videos that can help you learn to use Keyboard Maestro.

Help » Logs

The Logs command in the Help menu displays the Keyboard Maestro Logs folder.

Help » Third Party Licenses

The Third Party Licenses command in the Help menu displays the licenses folder containing the third party licenses for code used in Keyboard Maestro.

Help » Online Documentation

The Online Documentation command in the Help menu takes you to the Keyboard Maestro web site and displays the documentation.

Help » Keyboard Maestro Web Site

The Keyboard Maestro Web Site command in the Help menu takes you to the Keyboard Maestro web site.

Help » Stairways Software Web Site

The Stairways Software Web Site command in the Help menu takes you to the Stairways Software web site.

Help » Service and Support

The Service and Support command in the Help menu displays the service and support details.

Tips

Remembering Macro Hot Keys

Hot Key Macros are only useful if you can remember which key does what.

Consider using mnemonic Macros. For example, in your email client, you might define a set of Macros to Insert Text, so use Control-A for your Address, Control-S for your Signature, Control-N for your Name, and so on.

Be consistent in your choice of Hot Keys. For example, use function keys to launch applications, Control-Function Keys to open documents, Control-Letter to Insert Text, and so on.

Keyboard Maestro also interoperates with KeyCue - if you use both applications and hold the control key down KeyCue will display all your active Hot Keys.

Use Function Keys for Global Hot Keys

It is quite hard to come up with global Hot Keys that will not conflict with those keys used by any application (a conflict is not really a problem, the Macro Hot Key will simply override the application, but this is not always desirable). It is best to use function keys, especially in conjunction with modifiers, as global Hot Keys since they tend not to be used by most applications.

Use the Number Pad

Remember that the number pad is available (and distinct from the numbers on the main keyboard).

Troubleshooting

Macros/Switchers do not work after I login, what’s wrong?

The Macros and Switchers were all working fine, but then I restarted or logged out and back in, and now they are not working any more, what’s wrong?

The actions are all enabled by the Keyboard Maestro Engine. You can start it by launching the Keyboard Maestro application, or have the engine start automatically by enabling the “Launch Engine at Login” preference in the General preference pane.

My Macros are not working, what’s wrong?

In order for macros that use actions like Select Menu Item or Manipulate Window to work you must enable access for assistive devices in the Universal Access system preference.

Also, check that the macro group that contains the macros is enabled for the application you are testing with.

And finally, check that the Keyboard Maestro Engine is running (the Keyboard Maestro icon should be in the menu bar unless you have turned that off or the engine is not running).

The Window Switcher shows an empty list, what’s wrong?

In order for the Window Switcher to work you must enable access for assistive devices in the Universal Access system preference.

The Window Switcher shows only some of my windows, what’s wrong?

The Windows Switcher can only see windows in the current Space, so if you are using Spaces, you will not see windows in another Space.

The Application, Window and Clipboard Switcher do not work at all, what’s wrong?

The Hot Keys for the Application, Window, and Clipboard Switcher are all Macros in the Switcher Group, so if you have disabled this group (or restricted it to certain applications), that will affect the switcher macros as well.

Also, ensure that the Switcher Group and the macros are enabled.

I want to use Command-Tab for the Application Switcher?

By default, older versions of Keyboard Maestro used Option-Tab for their Application Switcher to avoid overriding the system Application Switcher, but if you wish to use Command-Tab you can do so.

Select the Switchers Group, then select the Application Switcher macro. Click the Edit button to switch to Edit mode if necessary. Click the Hot Key field and select your desired hot key, Command-Tab or Option-Tab or any other hot key.

AppleScript’s display dialog does not work, why?

AppleScripts are executed via osascript, so they run in the background, do not lock up the engine, and cannot crash the engine. However this means that they cannot perform user interaction. If you need user interaction (such as display dialog), you can work around this by asking another application (usually System Events is a good choice) to perform the request. For example:

```
tell application "System Events"
  activate
  display dialog "Hello"
end tell
```

See also the Scripting section.

I have an X-Key keyboard, are there any issues?

In general, Keyboard Maestro’s Device trigger can work with all the P.I. Engineering’s X-Key or other programable keyboards.

To use a key, the key must act like a button, that is it must be a single switch that is pressed on/off. You cannot use jogs, shuttles, proportional joysticks or (generally) sliders as trigger buttons.

To detect a key, Keyboard Maestro must be able to see an individual key turn on and then turn off. If you are using a double key on your device, you must remove it and configure either of the single keys and then you can replace it - otherwise it will seem like you are pressing two keys and Keyboard Maestro will not know which to use.

Currently, Keyboard Maestro cannot recognise the left most column of the MWII Jog & Shuttle, the top two rows of the LCD No Reader, and the left most column of the 128-Key. We are working with P.I. Engineering to resolve these issues.

How do I get more help?

For more information about a specific Keyboard Maestro feature consult the Keyboard Maestro Documentation, post a question to the Keyboard Maestro User Group, visit the Keyboard Maestro web site or contact us.

We always respond to email, however email is no longer a guaranteed medium and spam filters can delete your message to us or our message to you. Messages sent using the feedback form will always get to us, emails sent to us will pretty much always get to us, but if you do not receive a response within one business day check your spam filters to see if they have trapped our reply. If you use the feedback form and want a reply, make sure you enter your email address!

Support

For sales enquires, customer service, technical support, or to contact project management, email us at support@stairways.com or use our Web Site Feedback Form.

You can join the Keyboard Maestro User Group online community consisting of the developers and Keyboard Maestro users.

For ideas, see the Macro Examples section.

You can download Keyboard Maestro from http://download.stairways.com/.

You can purchase Keyboard Maestro at http://purchase.stairways.com/.

You can look up your current or previous license status and serial numbers, and get information about discounted upgrades from http://enquiry.stairways.com/.

For more information about anything to do with Keyboard Maestro visit http://www.keyboardmaestro.com/.

Glossary

Clipboard
: The system clipboard is where you store items when you Copy and Paste. When you Copy an item, it is temporarily stored in the Clipboard and when you Paste, the item is copied from the Clipboard into your currently selection.

Clipboard History
: Normally the system stores only one clipboard. Keyboard Maestro keeps a history of your system clipboard, ensuring you never lose data on the clipboard and allowing you to copy and paste multiple items,

Clipboard Switcher
: is a feature of Keyboard Maestro that allows you to copy or paste to/from a set of Named Clipboards.

Excluded Applications
: is the set of applications that should not appear in the Application Switcher list, allowing you to hide applications you rarely want to switch to. These applications are also ignored when hiding other applications.

Global Macro Group
: a predefined Macro Group that always exists and is the default location for new Macros.

Growl
: a system extension that lets Mac OS X applications unobtrusively tell you when things happen (more info).

Hot Key
: A keystroke that acts as a Macro Trigger to start the execution of Macro Actions in a Macro.

KeyCue
: software from ergonis that displays command keys and can also display Keyboard Maestro Hot Keys (more info).

Keyboard Maestro Engine
: The process that enables your Macros, Application Switcher, Window Switcher, Clipboard Switcher and web server to work even after you quit Keyboard Maestro.

Mac OS X
: Apple’s operating system versions 10.0 and up.

Mac OS
: Apple’s operating system we’ve all come to know and mostly love.

Macro
: a set of Macro Triggers together with a sequence of Macro Actions. Any one of the triggers will begin the execution of the sequence of actions.

Macro Action
: an action you wish to perform, such as opening a file, typing some text, controlling iTunes, and so on.

Macro Group
: a set of Macros which can be restricted to only a defined set of applications.

Macro Palette
: a floating palette containing any active Macros that have a Macro Palette trigger. The palette only appears in applications with at least once active Macro Palette triggered Macro.

Macro Trigger
: an event, such as a Hot Key, application launch, time of day, that starts the execution of a Macro.

Michael Kamprath
: the original developer of Application Switcher and Keyboard Maestro.

Named Clipboard
: Keyboard Maestro provides a set of named clipboards where you can permanently store information (text, logos, graphics, etc).

Program Switcher
: the premier application management utility for Classic Mac OS, written by Michael Kamprath it was in part the inspiration for Keyboard Maestro and forms one of the components of Keyboard Maestro.

Quick Macro
: a macro recorded on the fly in another application. see the Recording section.

Record Quick Macro
: the action that when triggered records a Quick Macro. see the Recording section.

Shortcut
: a Shortcut is another name for a Macro (it is also another name for an Alias but that is a different context to the normal Keyboard Maestro Macro context).

Text Tokens
: text tokens allow you to insert dynamic text, such as the current date or time, into various text fields, see the Text Tokens section.

Tokenized
: see Text Tokens.

UID
: a unique identifier that remains the same even if you rename or modify a macro or macro group.

URL
: an acronym for Uniform Resource Locator, this is the standard way of specifying a location on the Internet. It starts with a protocol, followed by a colon and then the location. For example, “http://www.stairways.com/”.

Z-order
: refers to the order of windows from frontmost to furthest back.

Administrative Details

Requirements

Keyboard Maestro 5 requires and Intel Mac and Mac OS X 10.6 or later.

Distribution

You may distribute this application in any way you wish as long as you only distribute the unmodified Keyboard Maestro package, as downloaded from www.stairways.com. You may not break Keyboard Maestro up into its component files and distribute parts of it separately.

History

Following on the success of Application Switcher for Classic Mac OS, Michael Kamprath wrote Keyboard Maestro for Mac OS X and released it in early 2002. Incorporating an impressively powerful hot key macro facility, as well as Application and Clipboard Switching facilities, it rapidly became an indispensable tool for many Mac OS X users, including us here at Stairways Software.

Development continued on version 1 through the end of 2002, and then work began on version 2. The first beta of 2.0 was released in early 2003 and development continued until the 2.0b6 beta released in May 2003. After that, life and work got in the way. Keyboard Maestro languished for over a year as Michael found that he did not have the time or energy to continue development.

Around May 2004, we contacted Michael as a concerned user to query the long delay in the eagerly awaited 2.0 release. When we learned that Michael was considering abandoning the application we offered to purchase it from him to ensure that we would not lose this valuable tool, as well as to continue the fine tradition that he had started.

On June 30, 2004 the deal was struck and Stairways Software acquired all the rights to Keyboard Maestro. Our aim was to resolve the outstanding issues with Keyboard Maestro and release 2.0 as soon as possible, which we did in September 2004. Keyboard Maestro 2 introduced many new Macro Triggers (such as Application, Time of Day, and so on), Macro Groups to allow easy control over when macros are active, and many new actions.

Development of Keyboard Maestro competed for resources with development of Interarchy until the latter was sold to lead developer Matthew Drayton in early 2007. After a short break, development on Keyboard Maestro 3 started in earnest and resulted in many new features, including improved and streamlined user interface, recording, new triggers, built-in web server, new actions, and numerous minor enhancements. Keyboard Maestro 3 was released in April 2008 followed by a succession of releases over the rest of 2008.

Development of Keyboard Maestro 4 began in late 2008 and was released in late 2009. Version 4 was a complete rewrite of the user interface, bringing with it a modern look and feel reminiscent of various modern Apple applications. A further either minor releases were made through 2010, followed by the initial release of Keyboard Maestro's baby brother Switcher Maestro and the Mac App Store version in January 2011.

By that point, development of Keyboard Maestro 5 was well under way. Keyboard Maestro 5 built on the solid user interface of version 4 (indeed, the look is almost identical), but added depth and breadth of power with almost no addition of complexity. Keyboard Maestro added such powerful features as control flow, conditions, variables, and calculations as well as many new actions, and enhancements to the application and clipboard history switchers.

Going forward, we plan to develop Keyboard Maestro aggressively, bringing it to new levels of both power and ease of use in the long tradition of both Mac OS and Stairways Software.

Credits

Thanks to Michael Kamprath for all his work producing Keyboard Maestro.

Thanks to Alan Gentle for many example Macro ideas.

Thanks to Ken G. Brown for some great beta testing.

Thanks to Rakesh Kumar for the set of Switcher Macros.

Thanks to Sam Stephenson and the Prototype Core Team for the Prototype JavaScript Framework.

Thanks also to:

- Eric Rescorla for wserver.
- Jerry Krinock for NS(Attributed)String+Geometrics.
- John Engelhart for RegexKitLite.
- John Gruber, Aristotle Pagaltzis for Title Case.
- Kelan Champagne for the YRKSpinningProgressIndicator.
- Matt Gemmell - Magic Aubergine for MGAnimatedView and MGTemplateEngine.
- Michael Ash for MAKVONotificationCenter.
- Matthew Ball for MBCoverFlow.
- Rainer Brockerhoff for RBSplitView.

Thanks to Corentin, Philippe, Stephen, Brad and others for their great assistance with beta testing.

Thanks to Andy for great help editing this documentation.

Thanks also to the many others who have provided input and support over the past decade.

Warranty

This application should do what we have described in this document. If it does not, you can simply stop using it. If you purchase it, and within 30 days find that it does not do what we have described here, then you can request a refund and your money will be refunded and we will cancel your license.

Licenses

Keyboard Maestro is copyright 2011 Stairways Software Pty Ltd. All Rights Reserved. You may use this application for a short trial period and then you must purchase the application or stop using it.

Keyboard Maestro is licensed on a per user basis and individual users may use it on up to five Macs. You must purchase a license for each user using Keyboard Maestro.

Trademarks owned by Third Parties such as Mac, Mac OS X, and BBEdit, are owned by their respective owners and no license is granted for their use.

Fine Print

Keyboard Maestro, Switcher Maestro, keyboardmaestro.com, switchermaestro.com and stairways.com are the property of Stairways Software Pty Ltd. Stairways Software Pty Ltd hereby disclaims all warranties relating to this software, whether express or implied, including without limitation any implied warranties of merchantability or fitness for a particular purpose. Stairways Software Pty Ltd will not be liable for any special, incidental, consequential, indirect or similar damages due to loss of data or any other reason, even if Stairways Software Pty Ltd or an agent of theirs has been advised of the possibility of such damages. In no event shall Stairways Software Pty Ltd be liable for any damages, regardless of the form of the claim. The person using the software bears all risk as to the quality and performance of the software.