

Keyboard Maestro 3 Documentation

[Overview](#)

[What's New](#)

[Features](#)

[Purchase](#)

[Screenshots](#)

[Tour](#)

[Movies](#)

[Links](#)

[Quick Start](#)

[How do I ...?](#)

- [How do I install Keyboard Maestro?](#)
- [How do I upgrade Keyboard Maestro?](#)
- [How do I purchase Keyboard Maestro?](#)
- [How do I register Keyboard Maestro?](#)
- [How do I get started?](#)
- [How do I create a new Macro?](#)
- [How do I Insert Styled/Colored Text or Images?](#)
- [How do I Insert the Current Date?](#)
- [How do I configure the program switcher?](#)
- [How do I uninstall Keyboard Maestro?](#)
- [How do I revert to Keyboard Maestro 2?](#)

[Macros](#)

- [Overview](#)
- [Macro Groups](#)
- [Macros](#)

[Macro Actions](#)

- [Overview](#)
- [Process Control](#)
- [Interface Control](#)
- [iTunes Control](#)
- [System Control](#)
- [Open](#)
- [Execute](#)
- [Clipboard](#)
- [Activate Switcher](#)
- [Record Quick Macro](#)
- [Pause](#)

[Macro Triggers](#)

- [Overview](#)
- [Hot Keys](#)
- [Application](#)
- [Login](#)
- [Macro Palette](#)
- [Public Web](#)
- [Status Menu](#)
- [Time](#)
- [Wake](#)

[Recording](#)

[Macro Examples](#)

- [Launch Your Most Used Applications](#)
- [Open Your Most Used Documents](#)
- [Insert Text Templates](#)
- [Use Hot Keys to Open Financial Accounts](#)
- [Use Hot Keys to connect to SSH or FTP sites.](#)
- [Simulate Bookmarks](#)
- [Remap Command Keys](#)
- [Simulate Missing Features](#)
- [Swap Characters](#)
- [Save a Text Clipping](#)
- [Delayed Click](#)
- [Insert Boilerplate Text](#)
- [Apply Text Conversions](#)
- [Simulate Workspaces](#)
- [Setup an Application When Launched](#)
- [Clean Up After Using an Application](#)
- [Feedback During Macro Execution](#)
- [Rakesh Kumar's PC Switcher's Pack](#)

[Program Switcher](#)

[Window Switcher](#)

[Clipboard Switcher](#)

[Clipboard History Switcher](#)

[Preferences](#)

- [Other Hidden Preferences](#)

[Scripting](#)

[Windows](#)

- [Macros Window Pane](#)
- [Macro Group Editor Sheet](#)
- [Macro Editor Window](#)
- [Recording Window](#)
- [Program Switcher Window](#)
- [Window Switcher Window](#)
- [Clipboards](#)
- [Clipboard Switcher Window](#)
- [Clipboard History Switcher Window](#)
- [Preferences Window Pane](#)
- [About Window Pane](#)

[Menus](#)

- [Keyboard Maestro](#)
- [File](#)
- [Edit](#)
- [Window](#)
- [Help](#)

[Tips](#)

[Troubleshooting](#)

[Glossary](#)

[Administrative Details](#)

- [Requirements](#)
- [Distribution](#)
- [History](#)
- [Credits](#)
- [Warranty](#)
- [Licenses](#)
- [Fine Print](#)

Overview

Keyboard Maestro will take your Macintosh experience to a new level in "Ease of Use". With Keyboard Maestro you can design or record your own macro shortcuts and activate them at any time, you can navigate through running applications and open windows, and you can work with an unlimited number of clipboards - all by pressing simple keystrokes.

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 or iTunes; 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 using a web browser to access Keyboard Maestro's built in web server.

Using Keyboard Maestro's [Program Switcher](#) and [Window Switcher](#) you can cycle through applications or windows, closing, hiding, launching, and more. Keyboard Maestro can help you regain control of your crowded screen by letting you close or hide multiple windows or applications.

Using Keyboard Maestro's [Clipboard Switcher](#), you can select from a number of named clipboards, storing information away for later retrieval. Store the details you are working on and access them by name again and again.

Using Keyboard Maestro's [Clipboard History Switcher](#), you can access past clipboards, so you will never lose your clipboard again. It also makes it easy to copy and paste multiple items such as username and password or serial number without having to switch back and forth between applications for each item.

You may try Keyboard Maestro for a few weeks, but then you must [purchase a license](#) to continue using it.

What's New

Keyboard Maestro 3 adds new ways to create macros, including recording, new ways to trigger macros, including using an iPhone or iPod touch or via the web, new actions, simplifications and an improved user interface.

New in 3.5

- Keyboard Maestro Control for iPhone or iPod touch (available from the App Store).
- Shiny new Macro Palette window.
- New Preferences Window (based on [Matt Ball's MBPreferencesController](#)).
- New About window.
- Actions for Set/Increase/Decrease the iTunes Rating of the current song.

- Actions for Set Volume and Set iTunes Volume.
- Tweaked the look of the vertical list (Window/Program Switcher).
- Only Quit the engine and relaunch if it is not in the right location with the right version.
- Fixed a crashing bug in the Press Button action.
- Report the new version on a manual check even if the current new version was previously ignored.
- Preserve the display/hide state for Excluded & Always Shown applications in Program Switcher.
- Fixed FireFox issue with the web server drop down menus.
- Fixed default nudge macros to be relative to Mouse and not Front Windows.
- Dragging a folder or bundle document to the Action list now creates an Open File action.
- Updated John Gruber's Title Case to Aristotle Pagaltzis refactored version.

New in 3.4

- Typed String trigger - fire off macros when you type a matching string.
- Shiny New Program and Window switcher windows.
- Save Clipboard To Named Clipboard et al save a redundant Clipboard Name for export.
- Display only active macros in Macro Group Palettes.
- Tweaked simulate keystroke action sheet to make it clearer you can type a key.
- Fixed bug where Web Server was not starting at Engine launch.
- Fixed some bugs in the Manipulate Window action editor.
- Fixed some bugs in the Scroll Wheel action editor.
- Fixed a crash displaying help for clipboard data.
- Removed PreserveOldClipboard hidden pref.

New in 3.3

- Status Menu & Status Menu triggered macros.
- Add "Add to Status Menu" option for Record Quick Macro and Macro Groups.
- Added enable/disable for actions in action lists.
- Added an action to Fast User Switch.
- Added Cut/Copy/Paste/Duplicate for macros, triggers and actions.
- Cut/Copy of macros, triggers and actions also copies as text.
- Added preference to Save & Restore Clipboard History.
- Badge the Folder icon for triggered macro groups.
- Enable/Disable Macros/Macro Groups via AppleScript (setMacroEnable).
- Added Title for Alert action.
- Added Comment (does nothing) action.
- Remove restriction on option-dragging macros within a macro group
- Sort window names in the Window Switcher case insensitively.
- Allow Set Clipboard to Text "" (empty).
- Correct title for Clipboard History Switcher window.
- New Trigger/New Action focuses the trigger/action browser.
- Remember the position of the update progress windows.
- Remember the position of the Alert windows.
- Add hidden preference for delay between actions (InterActionDelay),
- Fixed a crash when Execute AppleScript referred to a missing file.
- Window switcher only lists normal windows (not palettes).
- Updated version 1 prefs file incorrectly switched Insert Text to By Typing instead of By Pasting.
- Allow zoom on main editor window.
- unsetenv SHLVL in Keyboard Maestro and Keyboard Maestro Engine.
- Use AppleScript to open Universal Access System Preference pane.
- Use AppleScript to configure Engine login item.
- Use AppleScript to launch Screen Saver in 10.5.
- Stopped Copy Switcher from reselecting Create New Named Clipboard each time.

New in 3.2

- Macro Group options including:
 - Enable contained Macros only immediately after a Hot Key press.
 - Toggle contained Macros with a Hot Key press.
 - Display a floating palette of contained Macros only after a Hot Key press.
 - Toggle display of a floating palette of contained Macros with a Hot Key press.
 - Display a floating palette of contained Macros.
- Added extensive Help to Macro Group Editor
- Added Quick Start help to Macros Pane
- Try Now button tries only selected actions.
- Script pseudo-trigger.
- Wake trigger.
- Login trigger.
- Added Alert action with Stop/Continue dialog.
- Support Delete key in Clipboard History Switcher.
- Allow Macros without any direct triggers.
- Move Clipboard -> Insert Text action menu item to Text -> Insert Text
- Disable editing for Global Macro Group.
- Save & Restore Unix/AppleScript Results Window Size & Position.
- Save Log files in ~/Library/Log/Keyboard Maestro folder.

- Remove Macro Group "Never" - replace with Disabled.

New Clipboard Features in 3.1

- Clipboard History Switcher - never lose your clipboard again.
- Added actions for setting the system clipboard to:
 - specific text including tokens like current date.
 - Named Clipboard.
 - past clipboards from the Clipboard History.
- Added an action to save the system clipboard to a Named Clipboard.
- Added clipboard filters:
 - Remove Styles.
 - Set line endings to Mac, Unix or Windows/DOS.
 - Change case (Uppercase, Lowercase, Capitalize and John Gruber's Title Case).
 - Trim Whitespace or Wrap or Unwrap text.
 - Change quotes to Smart, Dumb or French.
 - Encode and decode HTML entities.
 - Zap control characters.
 - Count characters, words or lines.
- Added an action for applying BBEdit Text Factories to the current clipboard.
- Clipboard Switcher windows can be moved and resized and can stay open if desired.

Other New Action in 3.1

- Manipulate Window can now center the window on its current screen.
- Pause can now pause for fractional seconds (eg 0.75).
- Insert Text can now insert via typing simulated keystrokes.

New Scripting Support in 3.1:

- Keyboard Maestro Engine's **do script** now supports executing any action based on its XML code.
- **do script** now waits for the action to finish before completing.
- You can ask Keyboard Maestro Engine if it is currently executing any macro.

Changes in 3.1:

The system clipboard is no longer restored after actions that set it (such as Insert Text via Paste or Use Named Clipboard). This avoids race conditions related to when the system clipboard is restored. However, you can restore the system clipboard with the Set Clipboard To Past Clipboard action or use the Clipboard History Switcher to choose any past clipboard.

Fixes in 3.1:

- Fixed crash with %CurrentClipboard% token.
- Disabled macros no longer conflict with new macro hot keys.
- Fixed a bug where the recorded click could be relative to the wrong sheet.
- Move "Check For Updates" to Keyboard Maestro menu.
- You can set the Macro Palette icon size with defaults write com.stairways.keyboardmaestro.engine macroPalette IconSize -int N
- Open Third Party Licenses folder from Help menu.
- Fixed bug in Time Trigger minute display.
- Defer Clipboard Switcher Copy/Cut until clipboard selection is made.
- Clicking on the paste clipboard window will re-focus it.
- Delete excluded application did not save change.
- AppleScript result display was not Unicode-savvy.
- Fixed a potetial crash in with Execute AppleScript.
- Fixed an issue with time triggers and sleep.

New integration with KeyCue 4.2

Without even having to upgrade Keyboard Maestro, you can now use the recently released [KeyCue 4.2](#) to display Keyboard Maestro hot key triggered macros at any time by simply holding down the control key.

New in 3.0.1

- Floating palette could disappear off the screen.
- Launch/Quit macro triggers were not working.
- Launch application action could fail in some cases.
- Quit All Applications was not quitting the current application.
- You can now add custom styles to the web server.
- Repeating hot key triggers continue to repeat with simulated keystrokes.
- Fixed 10.4 displaying all hot keys as ?
- Fixed 10.4 issue with quit/launch/close keys in Program Switcher.
- Fixed name display of unlaunched applications in Program Switcher
- Fixed some cases where excessive CPU usage could occur.
- Fixed some Macro Palette location issues related to multiple screens.

New ways to create and work with macros

- Recording - create macros by recording what you do.
- Record Quick Macros - record and replay without changing applications.
- Enable and disable macros and macro groups.
- Try macros and individual actions as you create them.
- New [PC Switcher Pack from Rakesh Kumar](#).

New ways to trigger macros

- Via AppleScript **do script**.
- Via built in web server.
- Using any hot key (eg unmodified A).

New and improved actions

- Execute inline Unix script text.
- Execute inline AppleScript text.
- Unix and AppleScripts can display their results briefly.
- Execute Automator Workflows.
- Log Out action.
- Activate the Program, Window or Clipboard switchers.
- Record Quick Macros.
- Direct support for inserting extended (ICU) date formats.
- [Horizontal Scroll Wheel](#) support.
- [Manipulate Window](#) resize smaller or to a specific size .

Improved user interface

- Streamlined main window.
- Streamlined macro editor window.
- Automatic (sparklish) updating.
- Exported macros include group information.
- Cleaned up and/or simplified pretty much every action and trigger editor.
- Running Applications are shown in popup menus rather than requiring disk navigation.
- Display an approximation of the time saved.
- Spruced up the switcher windows, including adding program switcher icon grid view.
- Cute little square buttons.
- Added rename buttons for those who can't figure out how to rename.
- Added video tutorials to the documentation.
- All HIView goodness.

Simplified

- Dropped support for PowerMate.
- Dropped support for version 1 preference files.
- Dropped support for Classic processes.
- Dropped support for 10.3.
- Window/Program/Clipboard Switchers are now activated as any other macro.

And in detail

- Hot keys shown without the dash (ie, ^J not ^-J).
- Improved display of hot keys with longer and more key names.
- Fixed up keyboard focus.
- gethotkeys AppleEvent to return current active hot key macros.
- Move preferences to a subfolder.
- Switched to extensible property lists for preference and macro files.
- Improved some icons (thanks Opacity!).
- Remove and reduce many artificial delays.
- Add extra default macros for Quick Macro support.
- Add Check for Update in Help menu.
- New clipboards created with good default name.
- [Macro Palette](#) displays in all spaces.
- Add actions after selection.
- Improve macro deletion messages.
- Set default clipboard text.
- Window switcher no longer brings window to the front just to close it.
- Double click, return, and button to rename.
- Display Number Pad hot keys.
- Display stderr from executed shell scripts.
- Support all AppleScript formats.
- Disable Hot Keys before simulating keys.
- Improved Press Button ability to find buttons.
- Warn about missing access for assistive devices.

Download

If you have seen enough to convince you, [Download Keyboard Maestro Now](#), otherwise continue on to [Features](#) or [ask us](#) whether Keyboard

Maestro can solve your automation needs.

Features

Keyboard Maestro 3 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 [Program 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](#).

Macros

- Create [Macro Groups](#), sets of [Macros](#) targeted at specific applications
- [Macros](#) can be triggered from one or more [Macro Triggers](#) including:
 - [Hot Keys](#) - put [Macros](#) at the press of a key
 - [Applications](#) - on launch, quit, activate, deactivate, or periodically while an application is running
 - [Time](#) - on login, at a particular time of day, or periodically
 - [Macro Palette](#) - with a click on a context sensitive [Macro Palette](#)
 - [Public Web](#) - over the Internet, explicitly to the public, or via authenticated logins
- Create [Macro Actions](#) by recording your actions
- [Macros](#) can execute a sequence of one or more [Macro Actions](#) including:
 - Process Control
 - Switch to Last, Next or a specific application
 - Quit All, Other or a specific application
 - Hide All, Other or the current application
 - Bring the current application's windows to the front
 - Interface Control
 - Manipulate the front or a named window - resize, close, zoom, move, minimize
 - Move or click the mouse
 - Select a specific menu item
 - Press a specific button by name
 - Simulate a keystroke
 - Simulate the scroll wheel
 - Control iTunes
 - Play a specific or random track or a specific Playlist
 - Play or pause
 - Rewind or fast forward
 - Go to the next or previous track
 - Control the System
 - Sleep, Shutdown, Restart or Log Out
 - Activate screen saver
 - Open/close CD tray
 - Control the system volume and brightness
 - Open a file, folder, [URL](#) or system preferences pane
 - Execute an AppleScript or unix script from a file or inline text, optionally displaying the results
 - Execute an Automator workflow
 - Control the Clipboard
 - Simulate Cut, Copy or Paste
 - Insert text, with token expansion for various time and dates and other system information, either by pasting or typing
 - 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 [Named Clipboard](#)
 - Apply a wide variety of filters to the current clipboard
 - Apply a BBEdit Text Factory to the current clipboard
 - Activate the Program, Window, Clipboard or Clipboard History switchers.
 - Record a quick macro without even launching Keyboard Maestro
 - Pause a [Macro Action](#) sequence

Program and Window Switcher

- Switch to any application or window with a keystroke
- Easily select the exact application or window you want
- Launch, hide, or quit any application, or close or minimize any window
- Choose the application ordering you want: alphabetically, by last use, or by launch order
- Optionally hide other applications
- Optionally always hide other applications
- Select from various themes

Clipboard Switcher

- An unlimited number of [Named Clipboards](#)
- Copy, Cut or Paste to/from a [Named Clipboards](#) using a single keystroke
- Clipboards can be saved across logins and restarts

Clipboard History Switcher

- Never lose your clipboard again

- Browse the past clipboards, pasting any previous clipboard item
- Clipboard History is saved across logins and restarts

Purchase

Keyboard Maestro is engineered by Stairways Software Pty Ltd and distributed by [Kagi](#). Keyboard Maestro is licensed on a per workstation basis (individuals can choose to license it on a per user basis).

New customers can purchase Keyboard Maestro for US\$36 by choosing [Purchase Keyboard Maestro](#) from the [Keyboard Maestro menu](#) or you can purchase multiple licenses from <http://purchase.stairways.com/>. A volume discount of 50% is available for purchases of 5 or more copies (the discount will be automatically applied by Kagi after you confirm your payment method).

Keyboard Maestro 3 is a paid upgrade from previous versions. Customers who purchased Keyboard Maestro after 1 November 2007 have been issued a free upgrade to Keyboard Maestro 3. If you have not received your free license, you can claim your free upgrade by looking up your Keyboard Maestro 2 purchase at <http://enquiry.stairways.com/>.

Customers who purchased Keyboard Maestro 2 between July 2004 and October 2007 can upgrade to Keyboard Maestro 3 for US\$18 until 30 June 2008. If you have not received your instructions on how to upgrade, you can find details by looking up your Keyboard Maestro 2 purchase 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 your upgrade license or purchase instructions, let us know and we will look into why you did not receive your email.

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 download from <http://download.stairways.com/>.

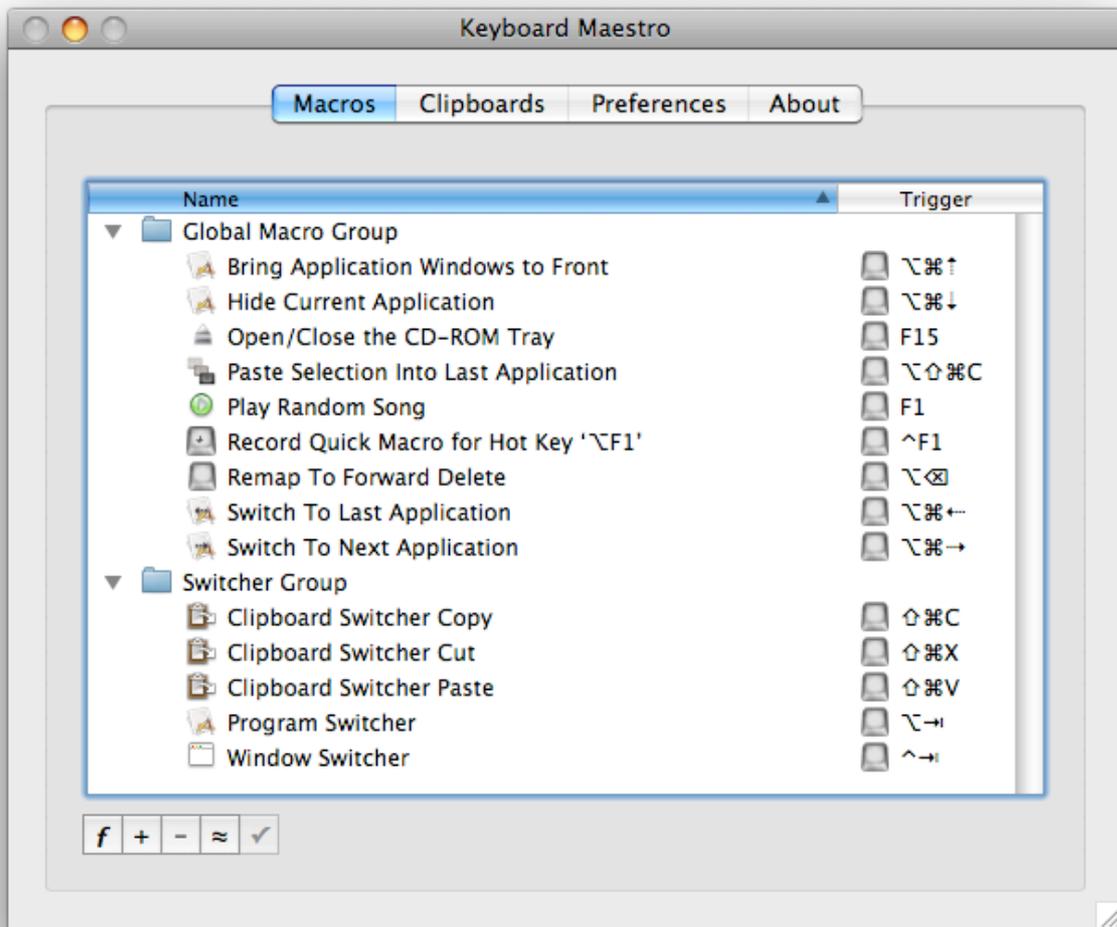
For sales enquires, customer service, technical support, or to contact project management, our current contact information is listed at <http://contact.stairways.com/>.

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

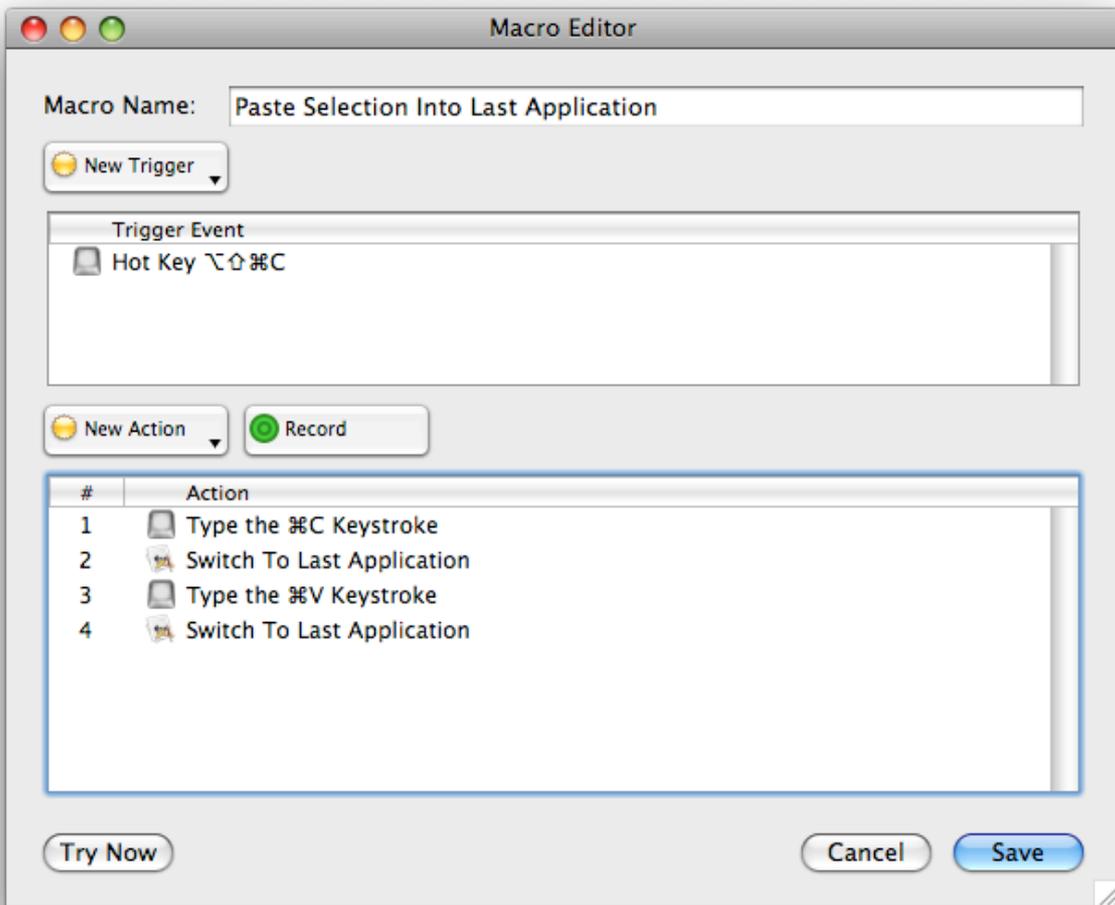
Screenshots

This is a very quick taste of Keyboard Maestro.

Macro Window Pane



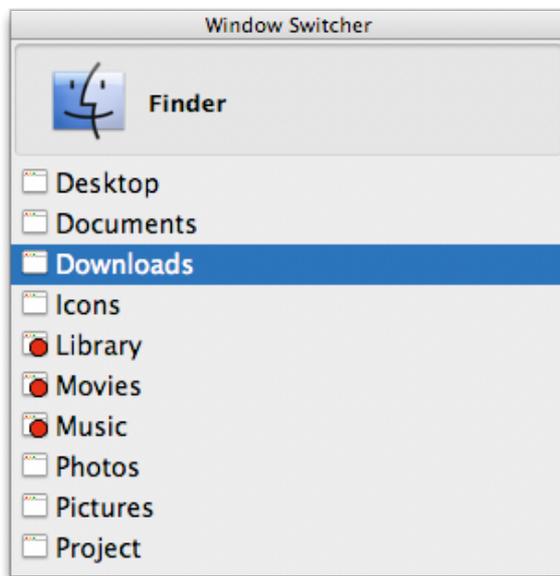
Macro Editor Window



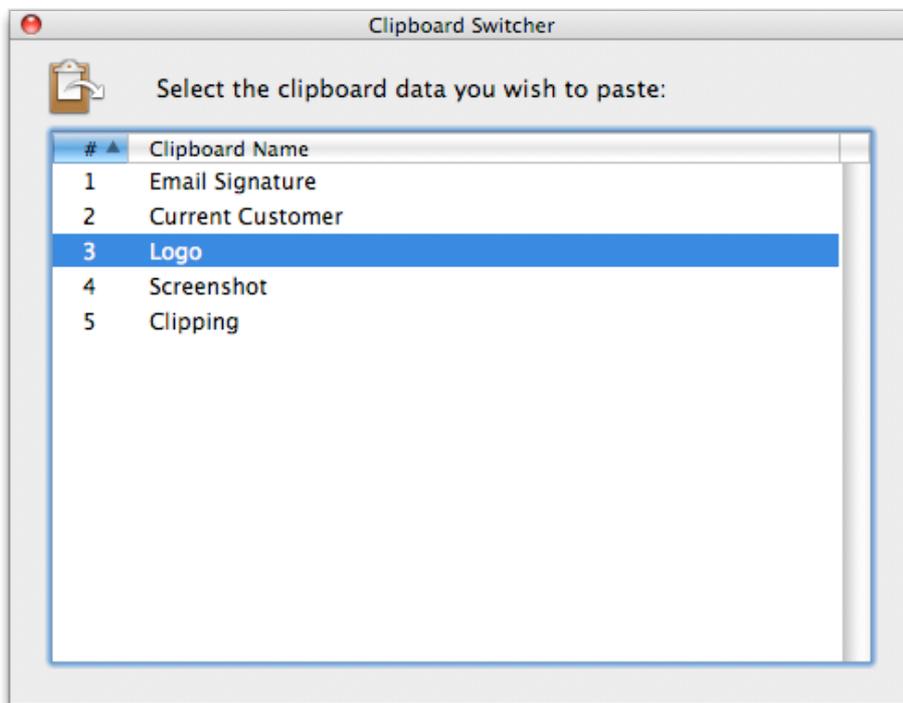
Program Switcher Window



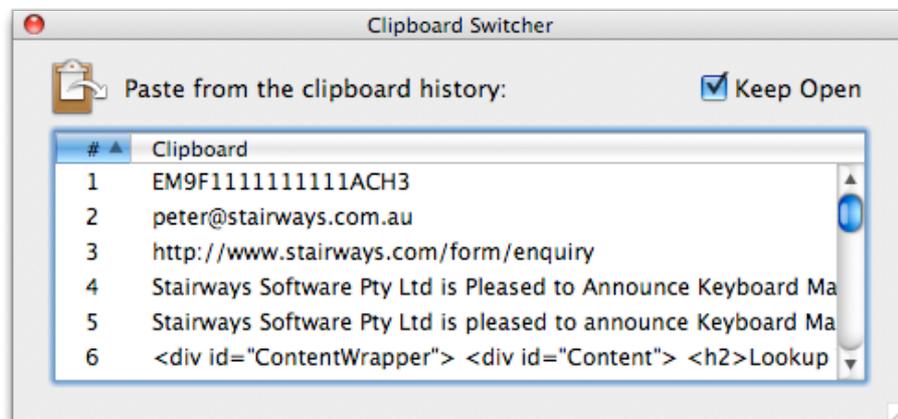
Window Switcher Window



Clipboard Switcher Window



Clipboard History Switcher Window

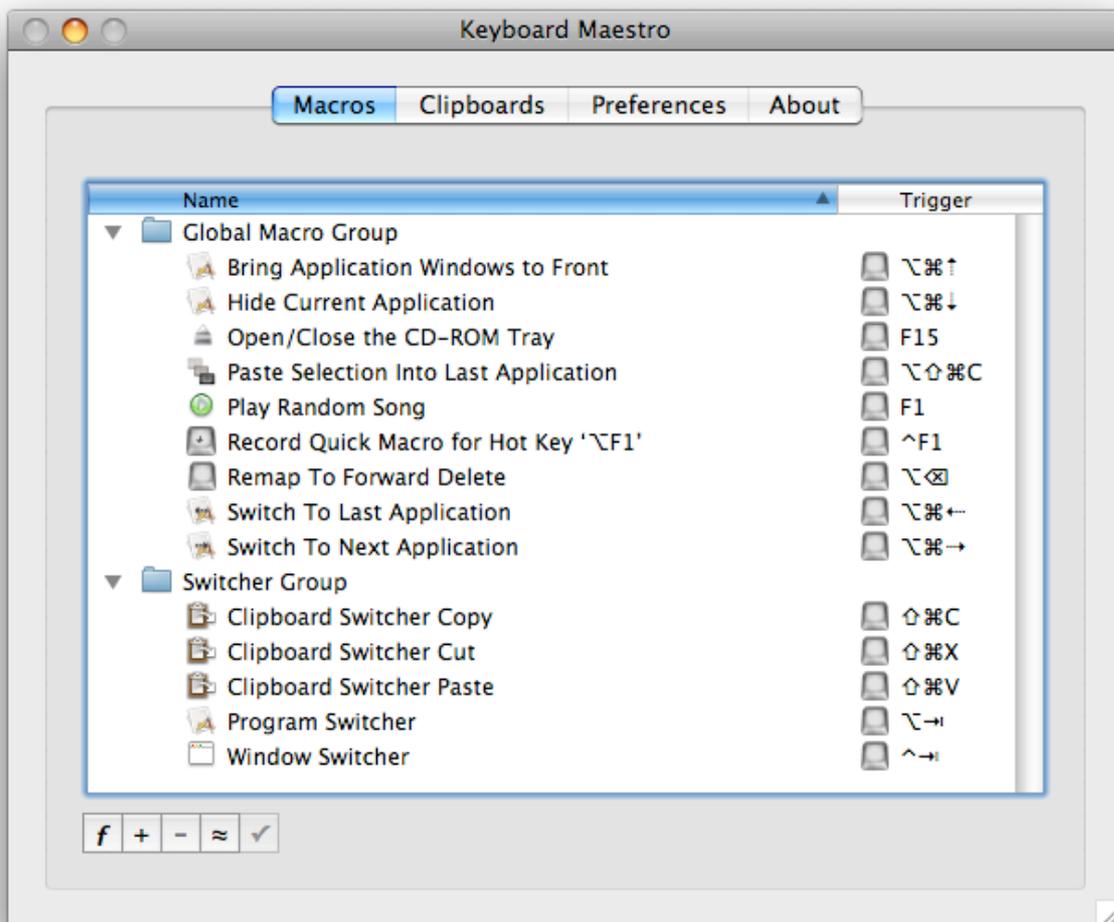


Tour

This demonstration will give you some idea of the power and versatility of Keyboard Maestro.

Getting Started

To start, launch Keyboard Maestro and it will display the [Macros window pane](#).



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

Make New Macro

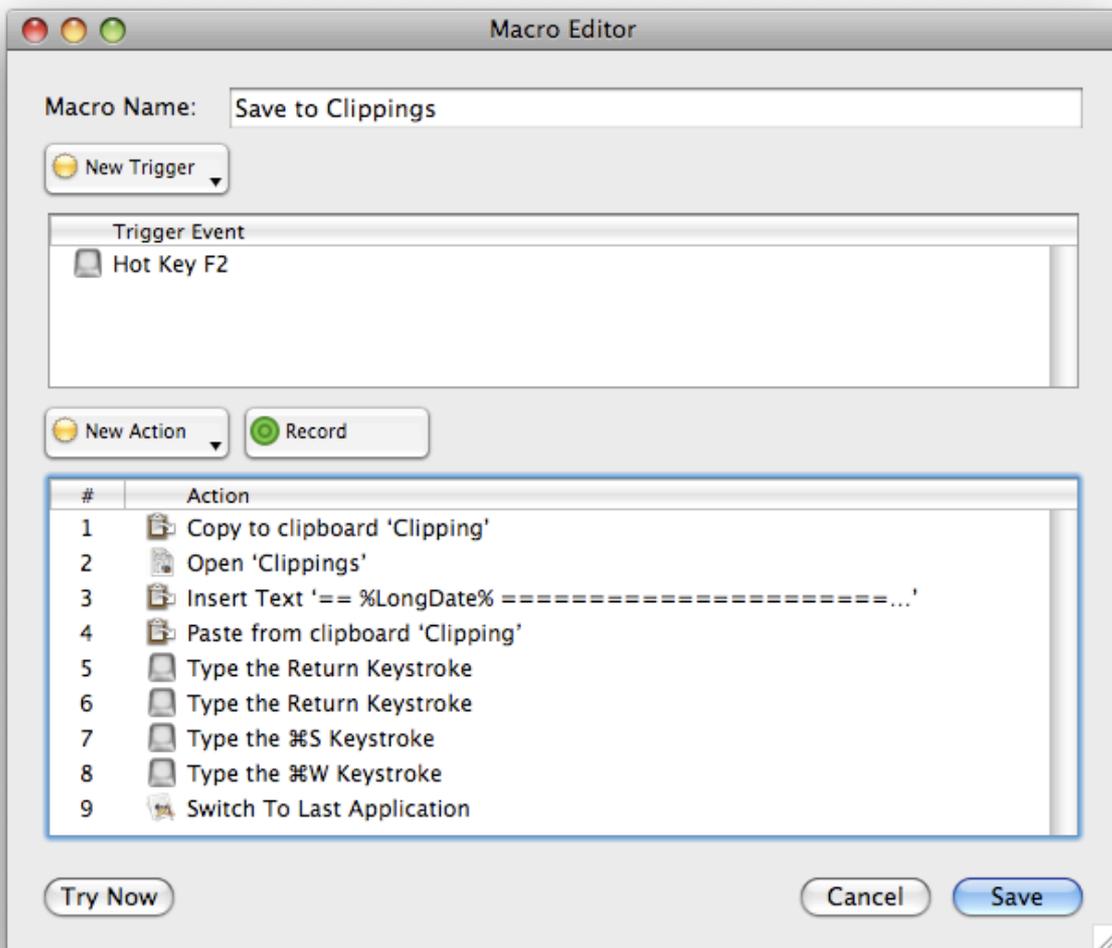
Click the + button to add a macro and display the [Macro Editor window](#). We will now design quite a complex Macro that will allow you to save clippings to a text file. First, launch TextEdit and create a new empty document and save it to **Clippings.rtf** in your **Documents** folder.

Now to add a selection to the **Clippings** file, you would normally do this:

- Press command-C to copy the selection.
- Open your **Documents** folder and then open the **Clippings.rtf** file.
- Press command-V to paste the clipping.
- Press return and type a line of dashes and return to separate the clippings.
- Close the window and save the file.
- Switch back to wherever you were before.

That is all very tedious, and probably explains why you do not bother doing it!

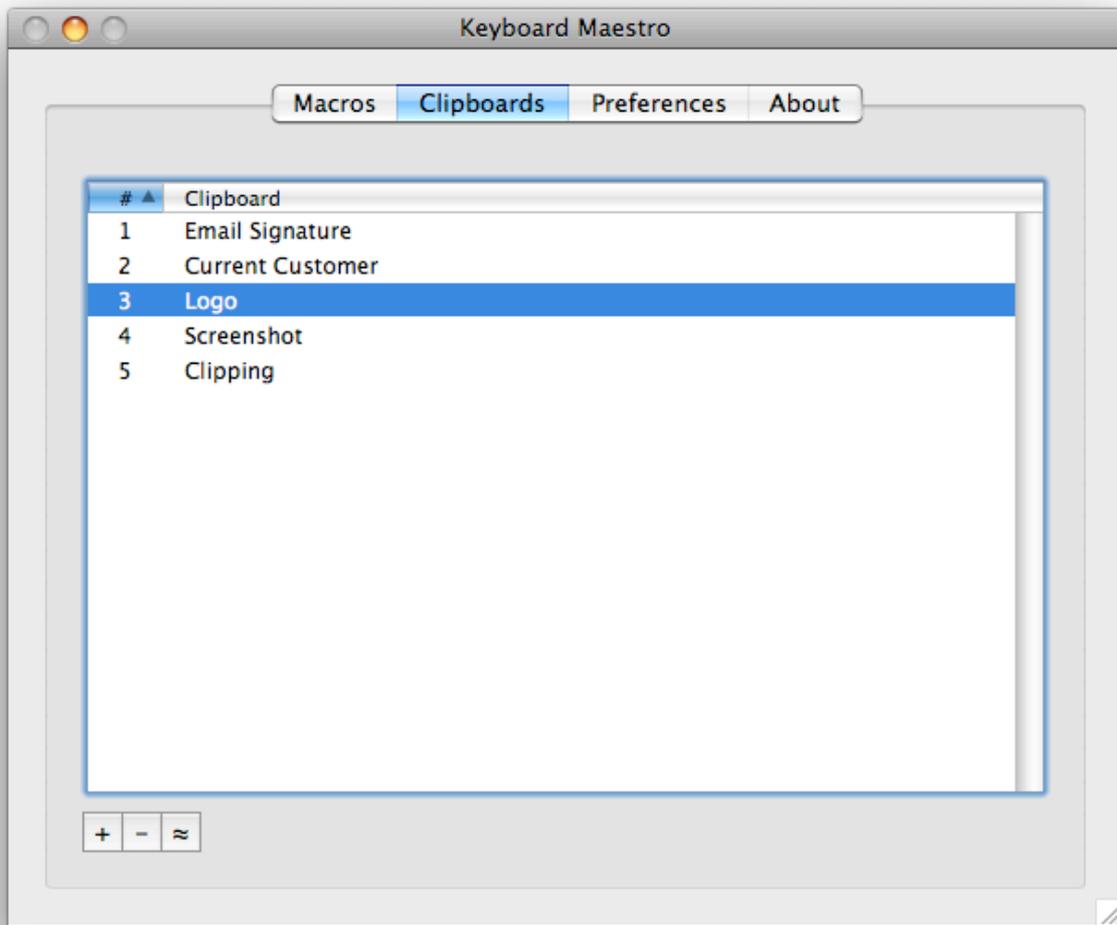
So lets define the whole sequence as a [Macro](#).



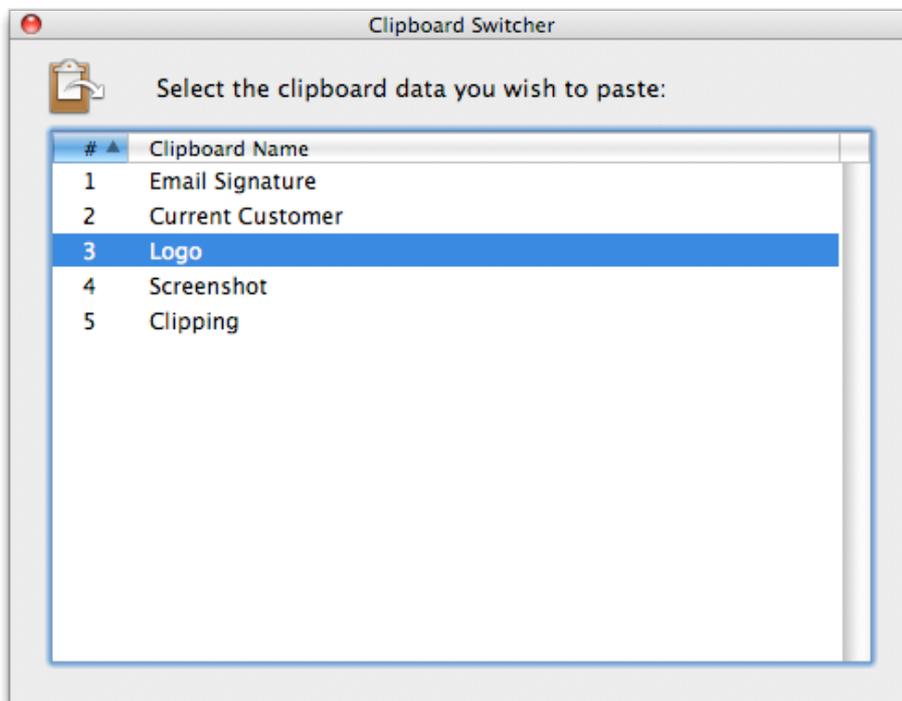
Save the Macro and now any time you have some text you want to save, just select it and press F2!

Named Clipboards

If you were paying attention, you might have noticed that rather than just copy the selection using command-C, we actually copied it to the named clipboard "Clipping". Keyboard Maestro lets you create as many named clipboards as you want using the [Clipboards window pane](#).



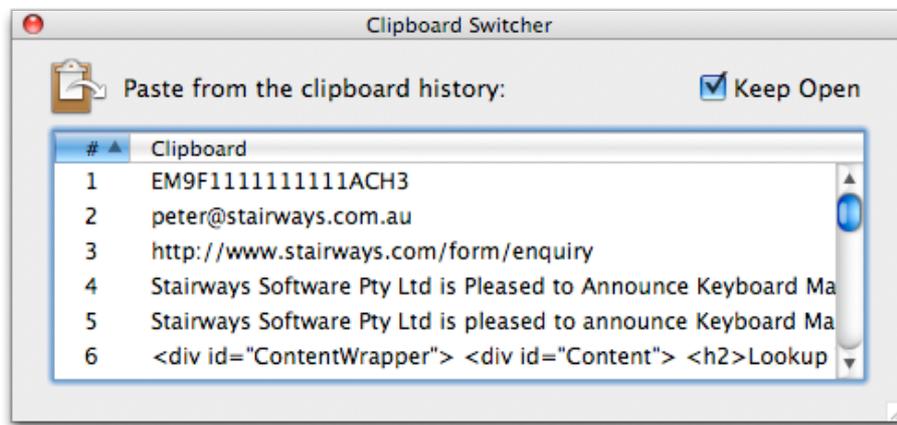
and then you can cut, copy or paste from them using the defined Hot Keys (by default, command-shift X, C and V respectively).



You can save frequently used information like your company logo in named clipboards and then paste them whenever you want, or use a named clipboard to store current information, like the current customer or project and paste it in as required.

Clipboard History

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



If you have a large enough screen, or a specific job that needs it, you can have the window stay open and paste previous clipboards by simply double clicking.

Program Switcher

At the end of the [Macro](#) we defined, we used the Switch To Last Application action to switch back to the program you were using before. Of course, most people use a lot more than one program and need to be able to switch between them. Keyboard Maestro not only lets you switch between applications, but also lets you define applications that always appear when you are switching so you can launch them quickly, and also [Excluded Applications](#) that never appear. As well, you can quit, hide or show applications. As of Keyboard Maestro 3, the Program Switcher is activated like any other macro, so you can find it in the Switcher Group in the [Macros window pane](#). By triggering the [Program Switcher](#) macro (for example, by pressing option-Tab) you can display the [Program Switcher](#) window.

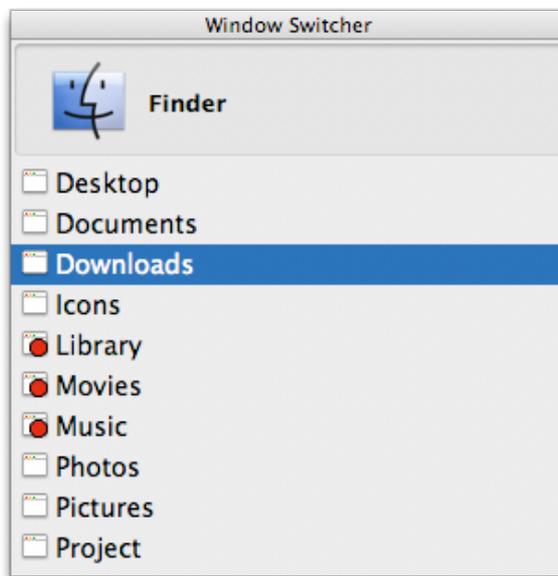


Note the applications set to be launched or quit.

If you want to be able to launch a specific program with a keystroke, you can define a [Macro](#) to do that and assign a [Hot Key Macro Trigger](#) such as F3.

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 the windows in the current application.



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

Conduct Your Mac Like a Pro!

This is just a taste of what Keyboard Maestro can do for you. Isn't it 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



Create A Simple Macro



Configuration



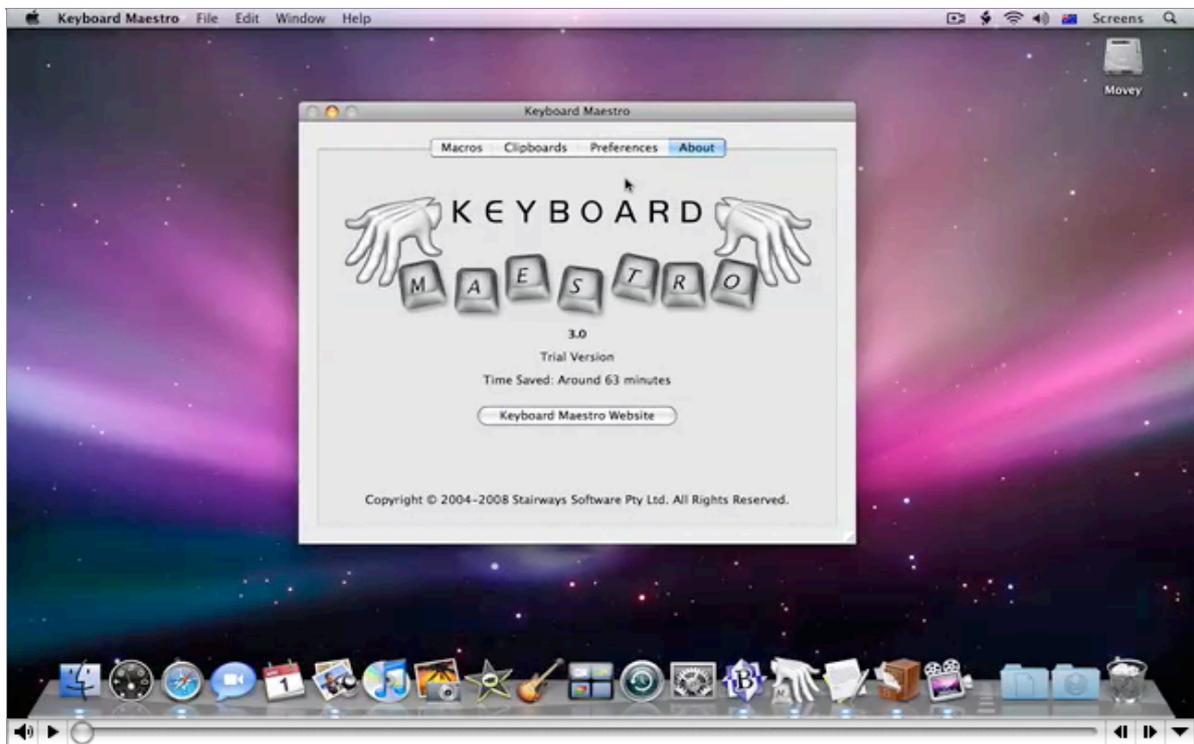
Save to Clippings Tutorial



Setup TextEdit Recording Example



Record Quick Macro Example



Links

Keyboard Maestro can be downloaded 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/>.

You can join the [Keyboard Maestro User Group](http://user.keyboardmaestro.com/) online community consisting of the developers and users of Keyboard Maestro 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 relatively easy to use once you understand how it operates. But it does take a few minutes to 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 when you want to make changes, and then you quit it. It does not need to be running normally. 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).

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 the clipboard, handles remote web requests, and executes your [Macro Actions](#). It should be running at all times, so it is a good idea to enable the "Launch Engine at Login" preference.

Keyboard Maestro organises 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. The [Hot Key](#) press will enable all the contained macros, either for a single execution or until the [Hot Key](#) is pressed again, and can optionally display 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 **f** button in the Macro pane in Keyboard Maestro. You can disable or enable [Macro Groups](#) by clicking the **✓** button. You can configure a [Macro Group](#) by double clicking on it.

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 in the Macro pane in Keyboard Maestro. You can disable or enable [Macro Groups](#) by clicking the **✓** button (remember that a

Macro can only be active when the [Macro Group](#) that contains it is active). You can edit a Macro by double clicking on it.

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. 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. 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. 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 selecting from the New Trigger menu.

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 (Process, Window, Clipboard and Clipboard History Switchers). You can also execute a script (AppleScript, Unix Script or Automator Workflow). You create Actions by creating or editing a Macro and selecting from the New Action menu 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 user 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?](#)
- [How do I upgrade Keyboard Maestro?](#)
- [How do I purchase Keyboard Maestro?](#)
- [How do I register Keyboard Maestro?](#)
- [How do I get started?](#)
- [How do I create a new Macro?](#)
- [How do I Insert Styled/Colored Text or Images?](#)
- [How do I Insert the Current Date?](#)
- [How do I configure the program switcher?](#)
- [How do I uninstall Keyboard Maestro?](#)
- [How do I revert to Keyboard Maestro 2?](#)

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](#), [Program 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 Keyboard Maestro preference "Launch Engine at Login" 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 3 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 move the new version in to the Applications folder, possibly replacing the old one.

Keyboard Maestro 3 will automatically import your version 2 macros, clipboards and preferences. Your old files will be saved in the [~/Library/Preference/Keyboard Maestro/Saved Version 2](#) folder in case you decide not to upgrade to version 3 for any reason.

If you have not done so already, you should consider turning on the Keyboard Maestro preference "Launch Engine at Login" 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 a new single user license of Keyboard Maestro for US\$36 by choosing [Purchase Keyboard Maestro](#) from the [Keyboard Maestro menu](#) or you can purchase multiple licenses from <http://purchase.stairways.com/>. We offer a 50% discount of orders of 5 licenses or more.

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 register Keyboard Maestro?

If you purchase Keyboard Maestro from within Keyboard Maestro by choosing [Purchase Keyboard Maestro](#) from the [Keyboard Maestro menu](#), it will automatically be registered.

Regardless of how you purchase Keyboard Maestro you will immediately 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 [Kagi](#) (click the Track Order link) or 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 3 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 or [Kagi](#)).

How do I get started?

The easiest way to get started is to 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](#) 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.

See also the [Tips](#) section.

How do I create a new Macro?

To create a Macro, launch Keyboard Maestro, select the [Macros window pane](#) and click the **+** button. Give the Macro a name, select at least one action and at least one trigger and click the **Save** button. 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 Clipboard -> 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 -> Insert Named Clipboard set to Paste to 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 date format](#).

See a movie showing how to [create a macro to insert the current date](#).

How do I configure the program switcher?

The [Program Switcher](#) (and all the switchers) are activated by macro actions. By default, Keyboard Maestro creates a "Switcher Group"

[Macro Group](#), and in there creates five macros. Each macro has a hot key trigger and a matching action which activates the appropriate switcher.

So to configure the [Program Switcher](#), launch Keyboard Maestro, switch to the Macros pane, find the Switcher Group, and in it the Program Switcher macro. Double click the macro, and then double click the Program Switcher action. You can then configure the various Program Switcher parameters, such as style and icon size.

How do I uninstall Keyboard Maestro?

Launch Keyboard Maestro and ensure the "Launch Engine at Login" preference 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 Keyboard Maestro 2?

Launch Keyboard Maestro and ensure the "Launch Engine at Login" preference 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.

Open the [~/Library/Preference](#) folder and the folder [~/Library/Preference/Keyboard Maestro/Saved Version 2](#) folder, move the files from the later folder into the former folder. Trash the [~/Library/Preference/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.

Macros

- [Overview](#)
- [Macro Groups](#)
- [Macros](#)

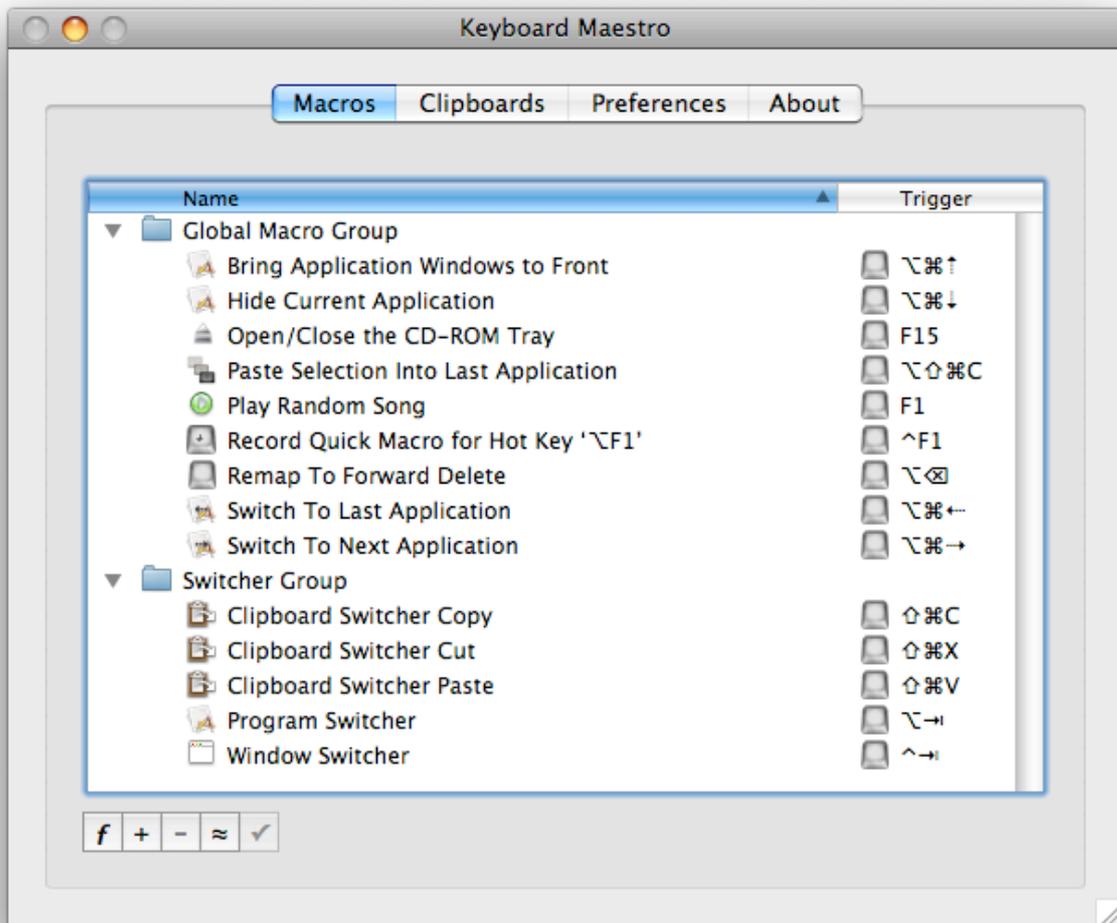
Overview

[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. Each macro consists of a sequence of [Macro Actions](#) to be performed together with one or more possible [Macro Triggers](#) which define when the actions will be performed. Sets of [Macros](#) are collected together in to [Macro Groups](#).

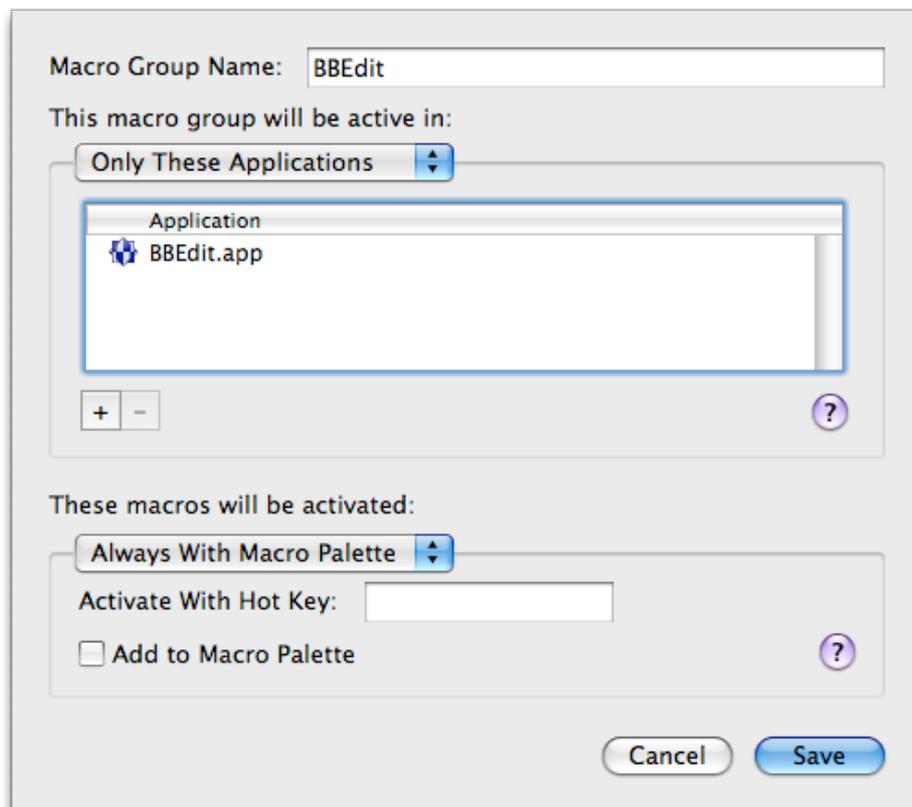
Macro Groups

A [Macro Group](#) contains a set of [Macros](#). Each [Macro Group](#) can be restricted to only a defined set of applications, so you can target particular Macros to particular applications and use the same [Macro Triggers](#) (often [Hot Keys](#)) in different applications to perform different actions.

To create a new [Macro Group](#), first launch Keyboard Maestro and click the [Macros](#) tab to select the [Macros window pane](#).



Now click the *f* button.



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 active 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.

For example you might have a Macro to launch your email client or word processor, or a Macro to open your financial accounts (perhaps one Macro to open your personal accounts and another to open your business accounts).

To have macros only active in specific applications, set the Macro Group to be active in Only These 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 Iris that arrange items or script guides.

To have macros active everywhere **except** specific applications, set the Macro Group to be active in All But These 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 with a Hot Key (either for a one shot macro, or toggled on and off) or display the macros in a floating macro palette.

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

To have macros that are active only after you press a specified Hot Key, set the Macro Group to be activated Once After Hot Key. The macros in the Macro Group will be enabled when you press the Macro Group Hot Key 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 press the group Hot Key. For example, you could have a group of macros to launch various applications and 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).

To have Macros that are active after you press a specified Hot Key, set the Macro Group to be activated by Toggle With Hot Key. The Macros in the Macro Group will be enabled when you press the Macro Group Hot Key and will remain enabled until you press the Macro Group Hot Key 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 after you press a specified Hot Key, set the Macro Group to be activated Once With Macro Palette. The Macros in the Macro Group will be displayed in a floating Macro Palette and enabled when you press the Macro Group Hot Key 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 press a specified Hot Key, set the Macro Group to be activated With Macro Palette. The Macros in the Macro Group will be displayed in a floating Macro Palette and enabled when you press the Macro Group Hot Key and will remain displayed until you press the Macro Group Hot Key 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 program, 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 be activated Always With Macro Palette. The Macros in the Macro Group will be displayed in a floating Macro Palette and enabled. The palette will close when you press the Macro Group Hot Key and will remain closed (and the macros disabled) until you press the Macro Group Hot Key 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 active 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 program, and have the Macro Group active (and hence the palette displayed) only in the CAD program.

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 activate (ie, whenever the palette is displayed).

You can disable or enable a macro group by selecting it and clicking the ✓ button. If a Macro Group is disabled, all its contained Macros will be disabled. You can disable or enable a Macro Group 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 double clicking on it in the Macros window pane.

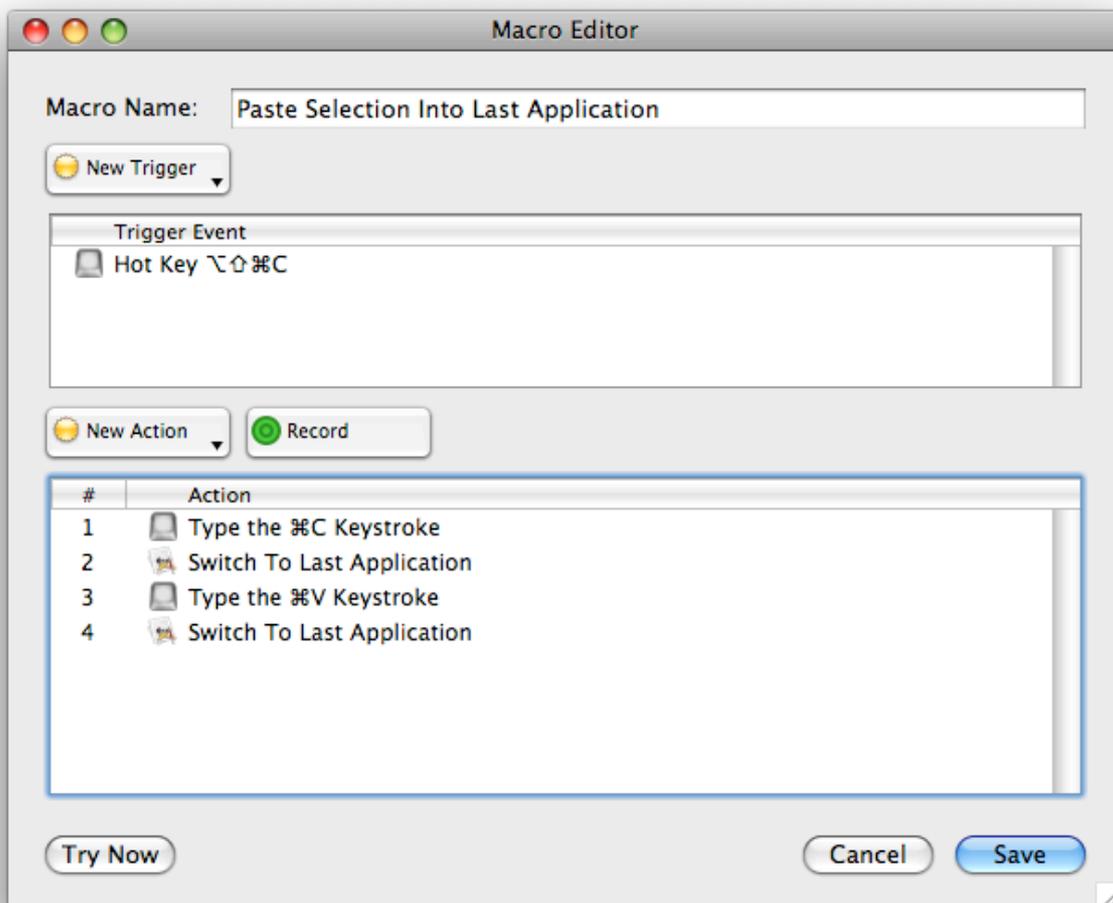
To delete a Macro Group, select the macro group, and either close it or select all of its macros as well, then press the Delete key or click the - button.

You cannot delete or rename the [Global Macro Group](#).

Macros

A Macro consists of a sequence of [Macro Actions](#) to be performed together with one or more possible [Macro Triggers](#) which define when the actions will be performed. A typical Macro consists of a single action, such as "open a particular file" together with a single [Hot Key](#) trigger, such as control-F1.

To create a Macro, first launch Keyboard Maestro and select the [Macros window pane](#) and then click the + button. To edit a Macro, double click it. Either way, the [Macro Editor window](#) will be displayed.



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 from the New Trigger pop-up menu. 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 selecting it and pressing the Delete key.

You can create a new action from the New Action pop-up menu. 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 also drag actions between macros and copy them by option-dragging. You can delete an action by selecting it and pressing the Delete key.

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 action manually.

Once completed, you can try the macro immediately by clicking the **Try Now** button.

Finally click the **Save** button to save the Macro. It will be available immediately (subject to the restrictions of the [Macro Group](#) it is contained in). If you select a [Macro Group](#) before creating a macro, it will be created within that group. By default, new Macros will be created in the [Global Macro Group](#).

You execute a macro's action sequence by:

- triggering any of its triggers:
 - pressing a [Hot Key](#)
 - launching/quitting/activating or deactivating the appropriate application

- waiting for the right time to arrive
- clicking it in the [Macro Palette](#)
- accessing the built in web server and selecting the public macro
- by logging in to the built in web server and selecting any macro
- by using AppleScript's `do script` command to execute a macro by name

Note that the web server needs to be separately enabled in the [Preferences window 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. You can disable or enable a Macro from AppleScript with:

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

\$end tell

See also the [Macro Groups](#), [Macro Actions](#), [Macro Triggers](#) and [Recording](#) sections.

Macro Actions

- [Overview](#)
- [Process Control](#)
- [Interface Control](#)
- [iTunes Control](#)
- [System Control](#)
- [Open](#)
- [Execute](#)
- [Clipboard](#)
- [Activate Switcher](#)
- [Record Quick Macro](#)
- [Pause](#)

Overview

A Macro executes a sequence of [Macro Actions](#) in series. There are many actions to choose from (detailed below), some simple actions, for example Sleep Computer, require no other information and simply do their job, while other more complex actions, for example 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.

Process Control

Process 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 (as defined by the [Program Switcher](#) sorting method).
- Activate Specific Application
launches if necessary and brings the specified application to the front.
- Bring Application Windows Forward
brings the 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 Specific Application
quits the specified application.
- 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.

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. 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.
- 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.

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 Titled Song
 - play a specified song.
- Play Titled Playlist
 - play the songs in a specified Playlist.
- Play Random Song From Library
 - play a random song.
- Play Random Song From Titled Playlist
 - play a random song from a specified Playlist.
- Play/Pause
 - toggle from playing to pausing or vice-versa.
- Pause
 - pause the current song.
- Stop
 - stop playing any song.
- Fast-Forward
 - 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
 - rewind the current song. It will keep rewinding until you do something else or until it reaches the beginning of the song.
- Next Song
 - play the next song.
- Previous Song
 - Go to the previous song or the start of the current song if it is already playing.
- Increase/Decrease Volume
 - Increase or decrease the iTunes volume.

System Control

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

- Sleep Computer
 - puts your Mac to sleep.
- Restart Computer
 - restarts your Mac.
- Shutdown Computer
 - shuts your Mac down.
- 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 if already muted.
- Increase/Decrease System Sound Volume
 - Increases or decreases the sound volume.
- System Beep
 - Play a simple beep (this can be useful for debugging complex Macros).
- Increase/Decrease Screen Brightness
 - Increases or decreases the screen brightness.

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 URL
 - open a URL with the appropriate helper.
- Open System Preference Pane
 - opens a specified System Preference Pane.

Execute

Execute actions let you execute AppleScripts, Unix scripts or Automator Workflows. Unix 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 Unix Script
 - execute a specified Unix script, either from a file or text. For example, "pbpaste | pbcopy".
- Execute Automator Workflow
 - execute a specified Automator Workflow.

If the AppleScript or Unix script generates any results, they can be optionally displayed in a floating window or briefly in a HUD window. For example, you could have a unix script `date` display briefly in a HUD window every hour.

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

Clipboard

Clipboard actions let you manipulate the system clipboard, [Named Clipboards](#) 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.

Insert Text

insert the specified text. Tokens are provided to include such things as the time or date. Text can be inserted by setting the system clipboard and pasting, or by simulating a sequence of keystrokes.

Use Named Clipboard

cut, copy or paste to/from the specified [Named Clipboard](#).

Set Clipboard to Text

set the system clipboard to 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. This can be used to restore the system clipboard after another action sets the clipboard temporarily.

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 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 (see below for a list).

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 "Enjoy, 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 date format](#). For example, we 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.
 */
```

Filter Clipboard can apply a number of possible transforms to the current system clipboard:

- Remove Styles.
- Set line endings to Mac, Unix or Windows/DOS.
- Change case (Uppercase, Lowercase, Capitalize and John Gruber's Title Case).
- Trim Whitespace or Wrap or Unwrap text.
- Change quotes to Smart, Dumb or French.
- Encode and decode HTML entities.
- Delete or Bullet control characters.
- Count characters, words or lines.

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. Alternatively, in the mean time, remember that you can apply scripted filters using a unix or AppleScript, for example the unix 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.

Activate Switcher

Keyboard Maestro includes several powerful switchers, including [Program 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:

Clipboard Switcher Copy

copies the current selection to a selected named clipboard (command-shift-C).

Clipboard Switcher Cut

cuts the current selection to a selected named clipboard (command-shift-X).

Clipboard Switcher Paste

pastes the selected named clipboard into the current system clipboard and current selection (command-shift-V).

Clipboard History Switcher

pastes a previous system clipboard from the clipboard history into the current system clipboard and current selection (command-control-shift-V).

Program Switcher

activates the [Program Switcher](#) (option-Tab).

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.

Unfortunately, the system program switcher uses command-Tab with no way to disable it, so that hot key is not available for your use, which is kind of ironic since Keyboard Maestro and its predecessor, Program Switcher, were providing this functionality long before it was

available as part of the system software. A recent bug report to Apple on this resulted in the not-very-promising response of "At this point, though, Engineering believes the issue behaves as intended."

Record Quick Macro

Recording is a new facility introduced in Keyboard Maestro 3. It 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 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.

Pause

The Pause action lets you pause in the middle of a sequence of actions. You can use this to get a visual break in your sequence of actions (eg for a slideshow), or to allow time for the applications affected by previous actions to "catch up" before continuing.

Macro Triggers

- [Overview](#)
- [Hot Keys](#)
- [Application](#)
- [Login](#)
- [Macro Palette](#)
- [Public Web](#)
- [Status Menu](#)
- [Time](#)
- [Wake](#)

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 create a floating palette of macros, execute macros in response to applications launching or quitting, or continuously while they are running, execute them remotely via an in-built web server, or at a particular time of day, as well as other possible events.

Hot Keys

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. Normally the Macro will not repeat if you hold the key down, but you can enable Key Repeat to have the macro execute continuously while you hold they key combination down.

A common use for [Hot Key](#) triggers is to open applications or documents, insert text templates, or as a way of remapping command keys (for instance you can make command-T execute "Replace and Find Again" in an application that uses a different command key (or no command key) for that menu command).

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.

Login

You can use the Login trigger to execute a macro when you login. You can also execute a macro periodically (for example every ten minutes or every three days). Login triggers can also be restricted to certain days of the week.

You could use a trigger like this to set up your Mac environment when you start your Mac or run periodic maintenance or backup scripts.

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.

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

Public Web

Keyboard Maestro 3 has an inbuilt web server. You can enable it in the [Preferences window 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.

Status Menu

You can have a macro execute when you select it from the Keyboard Maestro Status Menu (on the right hand side of the menu bar). Keyboard Maestro will only display the Status Menu triggered macros that are active, so if your Macros are restricted to particular applications, then they will only appear in those applications. This 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.

Time

You can use the Time trigger to execute a macro at a specific time. You can also execute a macro periodically (for example every ten minutes or every three days). Time triggers can also be 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.

Wake

You can use the 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.

Recording

A major new feature of Keyboard Maestro 3 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 [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 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](#)
- [Open Your Most Used Documents](#)
- [Insert Text Templates](#)
- [Use Hot Keys to Open Financial Accounts](#)
- [Use Hot Keys to connect to SSH or FTP sites.](#)
- [Simulate Bookmarks](#)
- [Remap Command Keys](#)
- [Simulate Missing Features](#)
- [Swap Characters](#)
- [Save a Text Clipping](#)
- [Delayed Click](#)
- [Insert Boilerplate Text](#)
- [Apply Text Conversions](#)
- [Simulate Workspaces](#)
- [Setup an Application When Launched](#)
- [Clean Up After Using an Application](#)
- [Feedback During Macro Execution](#)
- [Rakesh Kumar's PC Switcher's Pack](#)

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 keys 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.

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 inbuilt 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.

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 to Named Clipboard "Clipping"
- Open File "Clippings.rtf"
- Insert Text "-- %LongDate% -----<return>"
- Paste from Named Clipboard "Clipping"
- Simulate Keystroke Return
- Simulate Keystroke Return
- Select Menu Item File » Save
- Manipulate Window Close Front Window
- Switch to Last Application (or Quit Specific Application)

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 unix 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 positions 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 when it is executed, but if you want to do something more, you can use an AppleScript or Unix script to speak text (AppleScript `say "hello"`) or display an alert (AppleScript `display dialog "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.

Program Switcher

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

The [Program Switcher](#) action lets you choose from three themes (vertical list, horizontal icons, or icon grid), icon size, and the sort order.

You can also choose to hide other applications when switching (Keyboard Maestro also has a preference in the [Preferences window 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 [Preferences window pane](#), you can configure various applications to never be displayed.

While the [Program Switcher window](#) is displayed, you can perform various actions:

- Press "q" to mark (or unmark) an application to be quit
- Press "s" to mark (or unmark) an application to be hidden
- Press "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



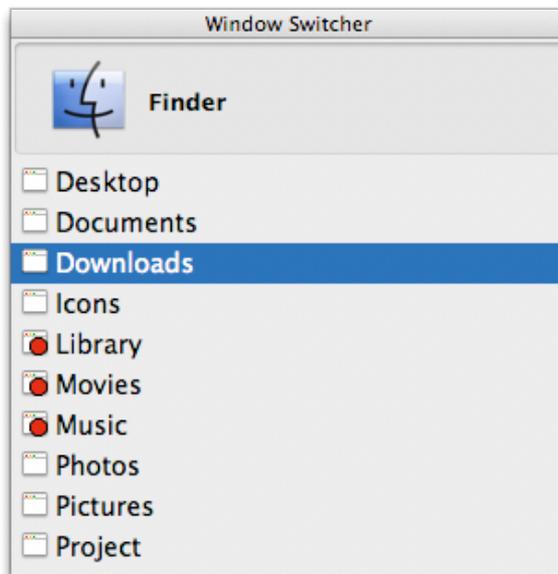
By default, Keyboard Maestro creates a [Program Switcher](#) macro in the "Switcher Group" [Macro Group](#), triggered by option-Tab. You can disable this Macro by selecting it in the [Macros window pane](#) and clicking the ✓ button.

Window Switcher

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:

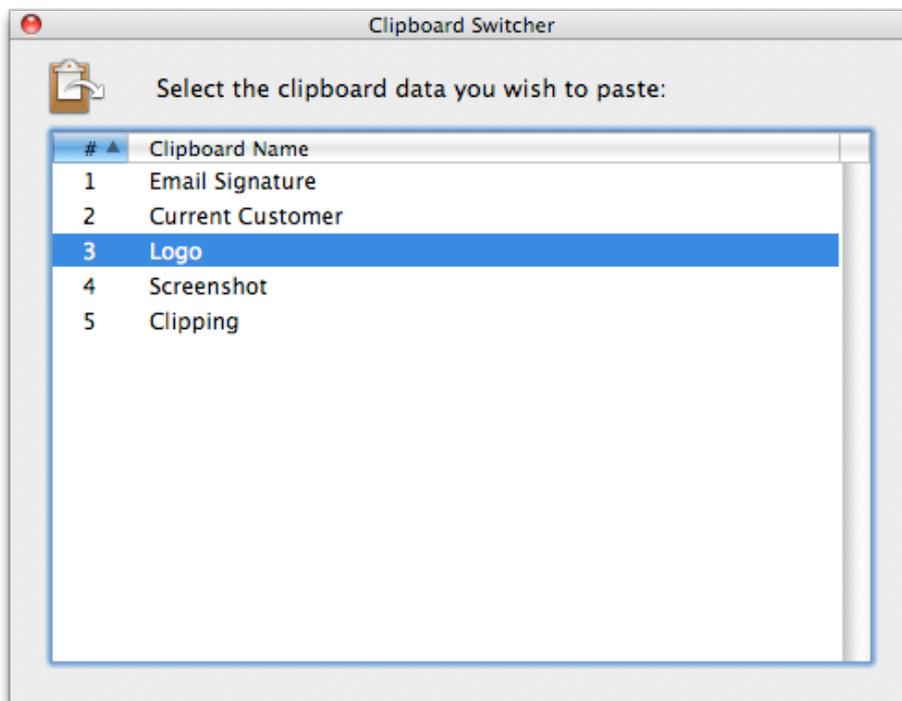
- Press "q" to mark (or unmark) a window to be closed
- Press "s" to mark (or unmark) an window to be minimize
- Press "c" to select the current application



By default, Keyboard Maestro creates a Window Switcher macro in the "Switcher Group" [Macro Group](#), triggered by control-Tab. You can disable this Macro by selecting it in the [Macros window pane](#) and clicking the ✓ button.

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.

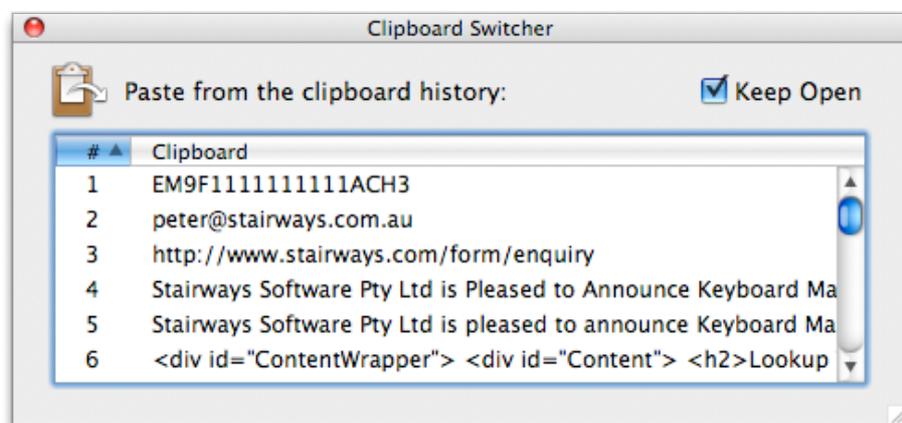


By default, Keyboard Maestro creates three [Clipboard Switcher](#) macros in the "Switcher Group" [Macro Group](#), triggered by command-shift-X, C and V (Cut, Copy and Paste). You can disable the Macros by selecting them in the [Macros window pane](#) and clicking the ✓ button.

You can add or delete named clipboards in the [Clipboards window 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.

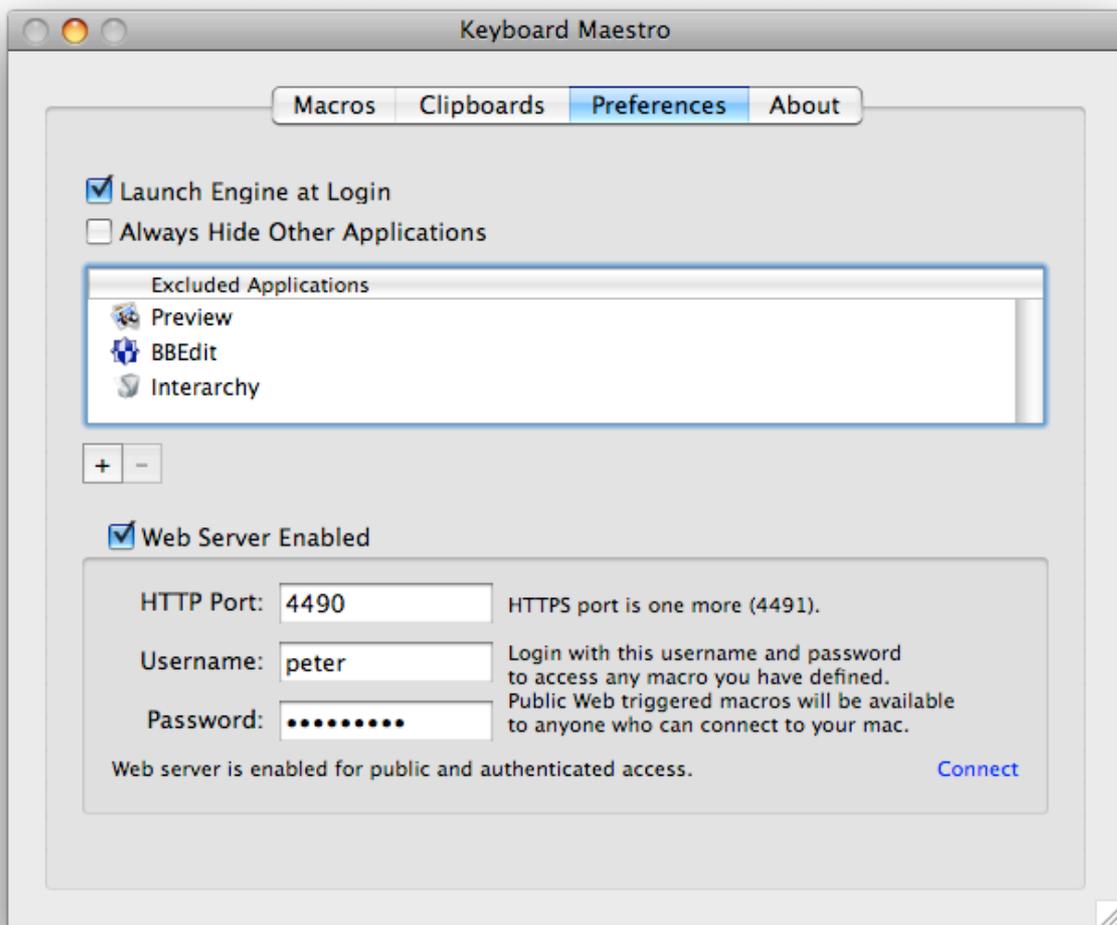


By default, Keyboard Maestro creates a Clipboard History Switcher macro in the "Switcher Group" [Macro Group](#), triggered by command-control-shift-V. You can disable the Macros by selecting them in the [Macros window pane](#) and clicking the ✓ button.

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 them in the [Macros window pane](#) and clicking the ✓ button.

Preferences

To configure Keyboard Maestro, first launch Keyboard Maestro and click the [Preferences](#) tab to select the [Preferences window pane](#).



In the [Preferences window pane](#) you can:

- enable or disable launching the [Keyboard Maestro Engine](#) at Login.
- configure whether to always hide other applications when switching applications.
- configure which applications should be excluded from the Program Switcher list of applications.
- configure Keyboard Maestro's inbuilt web server.

The web server is disabled by default.

If the web server is enabled, then anyone who can connect to your Mac can access any Public Web triggered macros.

If the web server is 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 access any of your macros.

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 more clearly differentiate them.

Other Hidden Preferences

You can configure the icon size for the [Macro Palette](#) using:

```
defaults write com.stairways.keyboardmaestro.engine macroPalette IconSize -int 32
```

You can set the maximum number of items in the clipboard history with:

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

You can set a delay between each action execution with:

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

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 all 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 "Switch to Last Application"
end tell
```

The macro must be defined and currently active.

If there is more than one macro with the same name, they will all be executed, so by preference you can use a UID instead of a name.

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

The easiest way to determine a macro's UID is to export it and then open the exported file and find the UID field.

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
```

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

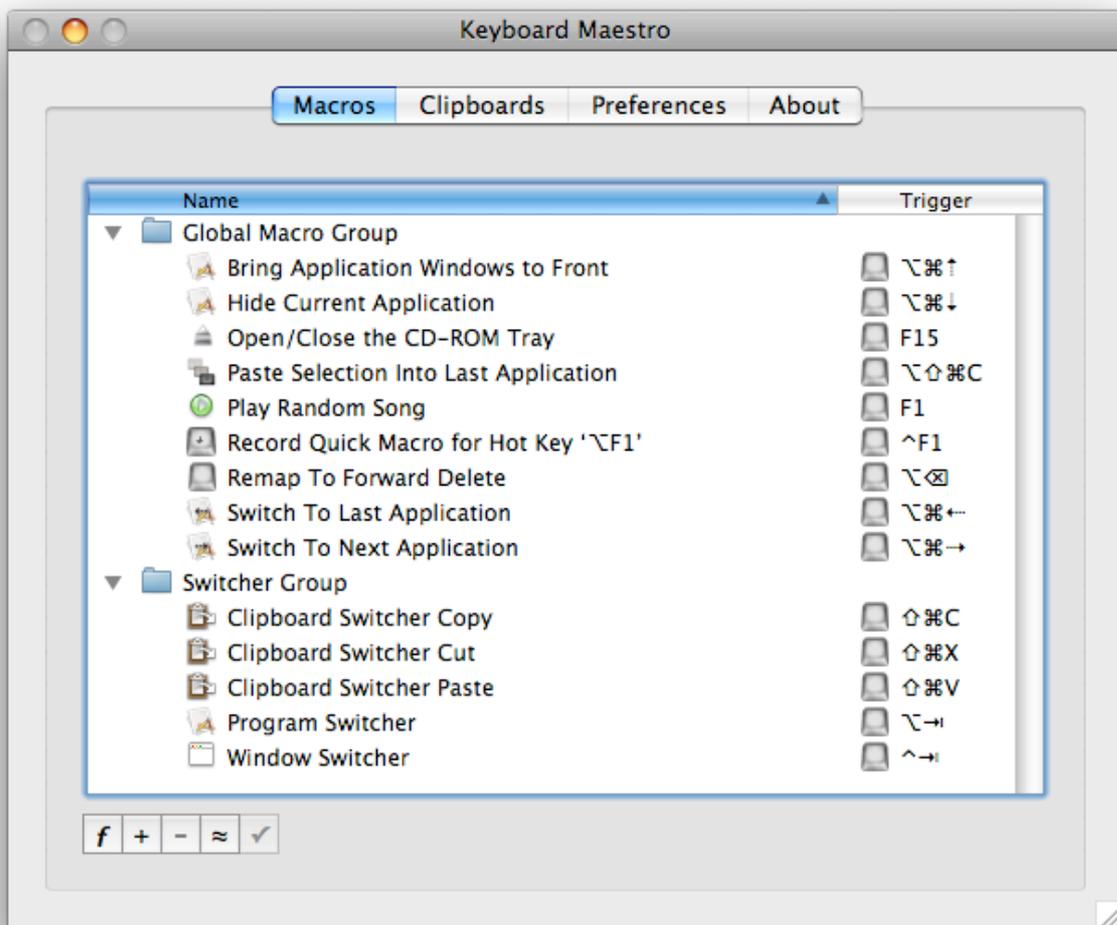
Windows

- [Macros Window Pane](#)
- [Macro Group Editor Sheet](#)
- [Macro Editor Window](#)
- [Recording Window](#)
- [Program Switcher Window](#)
- [Window Switcher Window](#)
- [Clipboards](#)
- [Clipboard Switcher Window](#)
- [Clipboard History Switcher Window](#)
- [Preferences Window Pane](#)
- [About Window Pane](#)

Macros Window Pane

This window pane lets you manipulate Macros and [Macro Groups](#), creating new ones, deleting old ones, enabling and disabling them and so on.

You get this window pane by launching Keyboard Maestro and clicking the **Macros** tab.



The window pane contains a list of [Macro Groups](#) and their associated Macros.

You can create a new [Macro Group](#) by clicking the **f** button. A [Macro Group Editor sheet](#) will appear allowing you to enter the [Macro Group's](#) details.

You can create a new Macro by clicking the **+** button. A [Macro Editor window](#) will appear and allow you to define your new Macro. The new Macro will be created in the currently selected [Macro Group](#) or the Global Macro Group if no item is selected.

You can change an entry (Macro or [Macro Group](#)) by double clicking it. You can rename an entry by clicking the **≈** button. You can delete an entry by clicking the **-** button or pressing the Delete key.

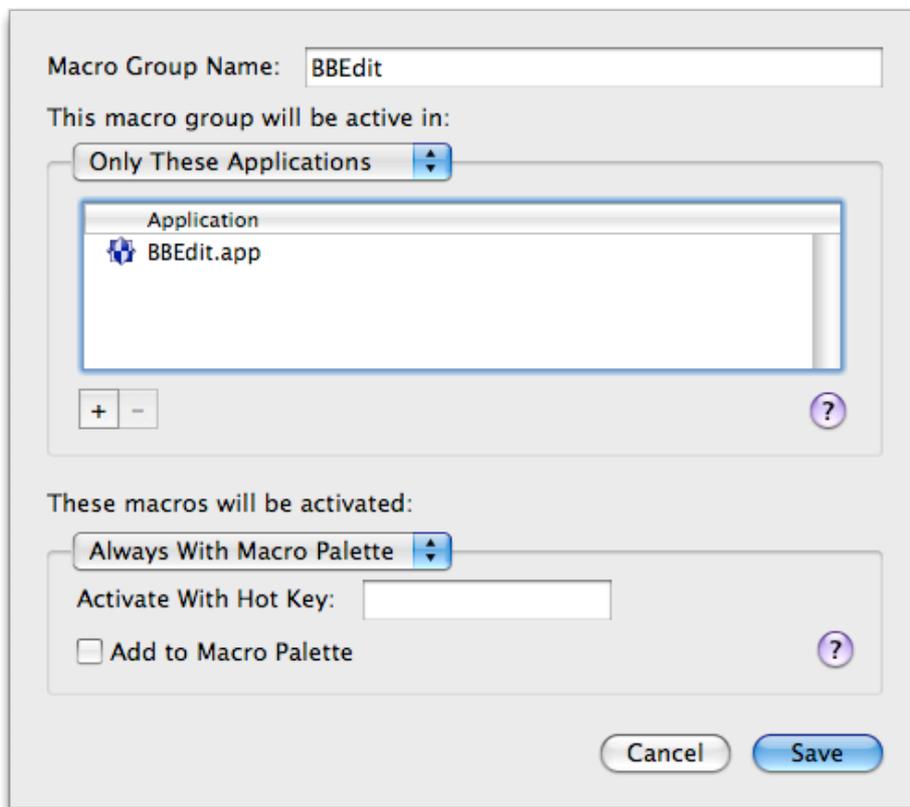
To delete a [Macro Group](#), you must either select it and all the enclosed macros, or close (collapse) the group and then select it.

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

See also the [Macros](#) and [Macro Editor Window](#) sections.

Macro Group Editor Sheet

This sheet allows you to enter the name of the group, choose the applications in which the group's Macros should be active, and how the macros will be activated or displayed.



Typically a group's macros would be active everywhere (All Applications), or it might be specific to a particular application (Only These Applications) in which case you might name the Macro Group after the application.

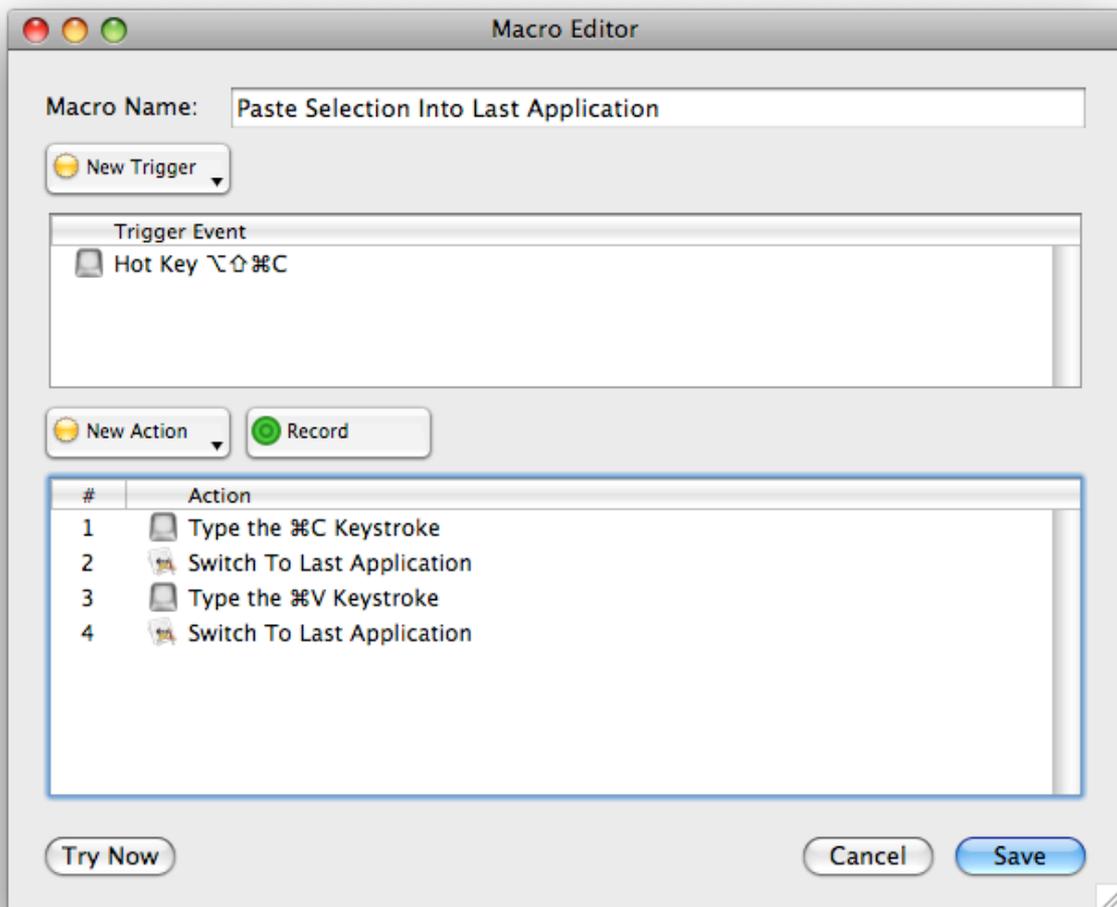
You can also configure the macro group to be activated only after a Hot Key press (either for a single use or toggled on and off), and whether to display the macros in a floating palette.

See also the [Macros](#) section.

Macro Editor Window

This window lets you create or edit a Macro.

You get this window by selecting the [Macros window pane](#) and either clicking the + button to create a new Macro or by double clicking an existing Macro to edit it.



You can give the Macro a name, add, delete or rearrange actions, and add or delete triggers.

You can click the **Record** button to record your actions, or click the **Try Now** button to try out the macro actions.

To learn more about creating or editing Macros, see the [Macros](#) section.

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.

Program Switcher Window

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

You get this window by triggering the [Program Switcher](#) macro.

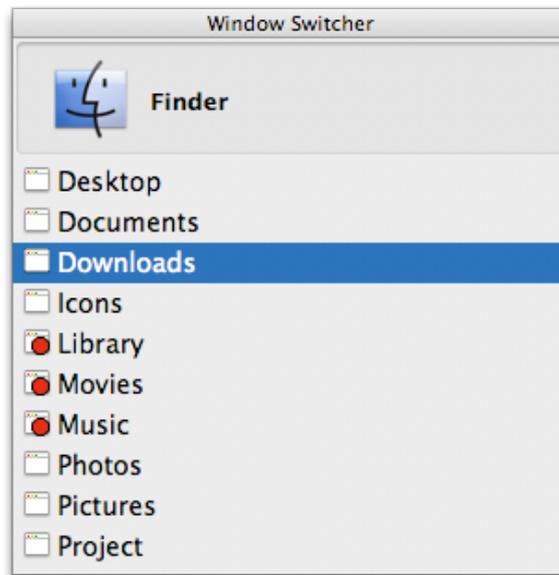


To learn more about the [Program Switcher](#), see the [Program Switcher](#) section.

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 Window Switcher macro.

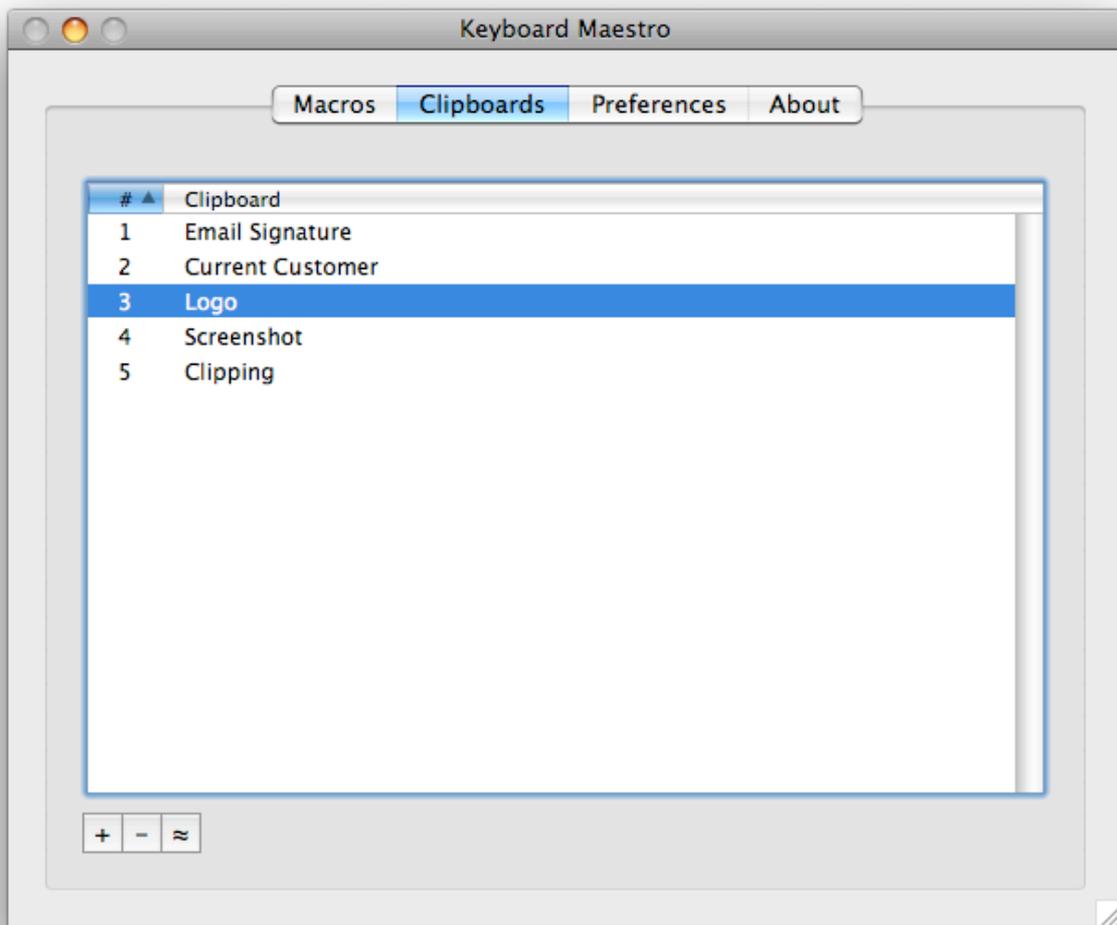


To learn more about the Window Switcher, see the [Window Switcher](#) section.

Clipboards

This window pane lets you create or delete named Clipboards.

You get this window pane by launching Keyboard Maestro and clicking the [Clipboards](#) tab.

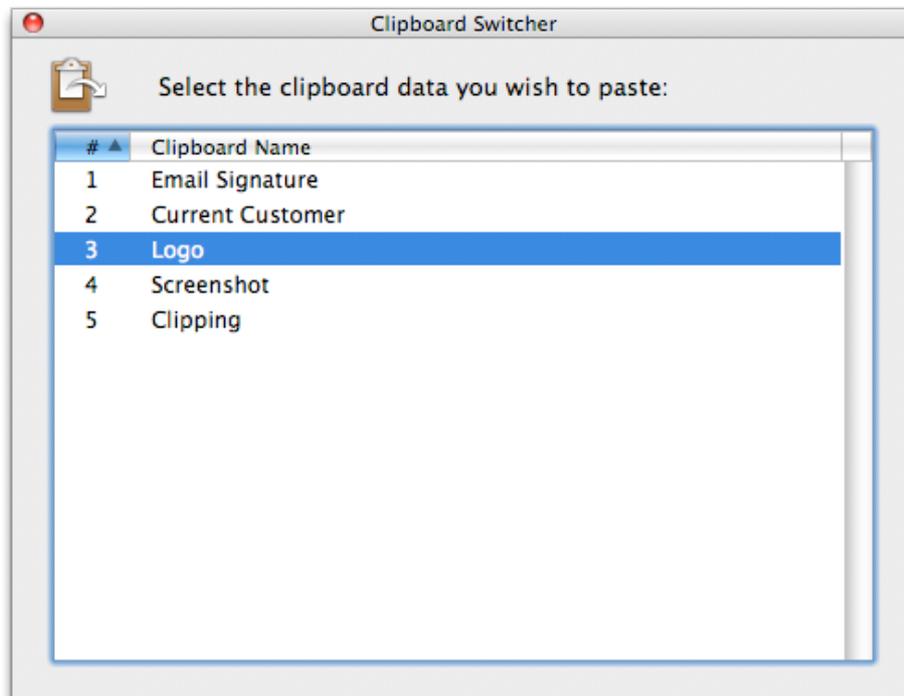


To learn more about the Clipboard Switcher, see the [Clipboard Switcher](#) section.

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 [Clipboard Switcher](#) macros.

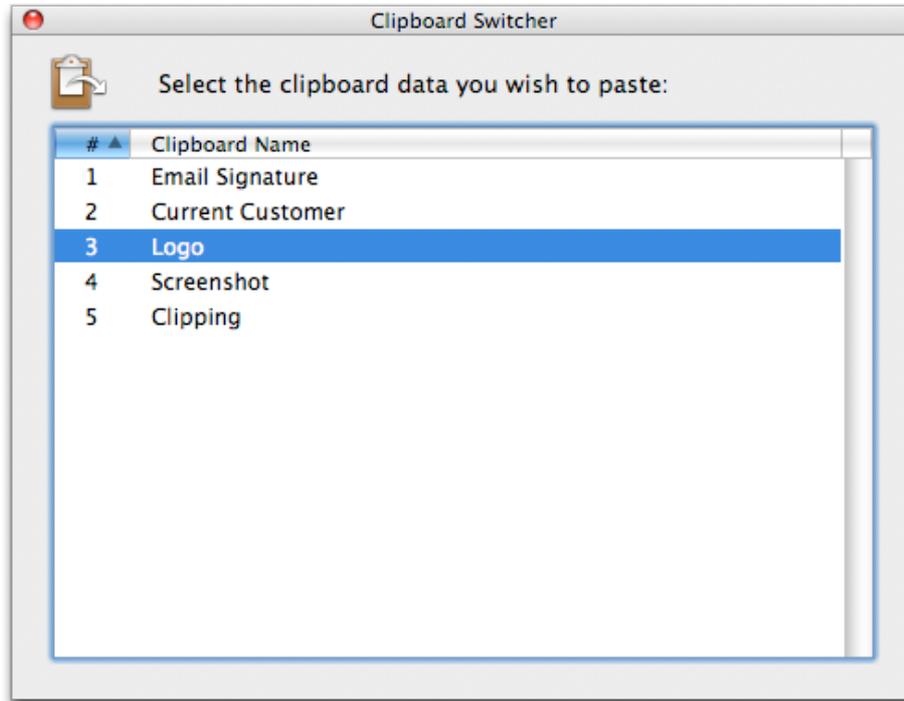


To learn more about the Clipboard Switcher, see the [Clipboard Switcher](#) section.

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.

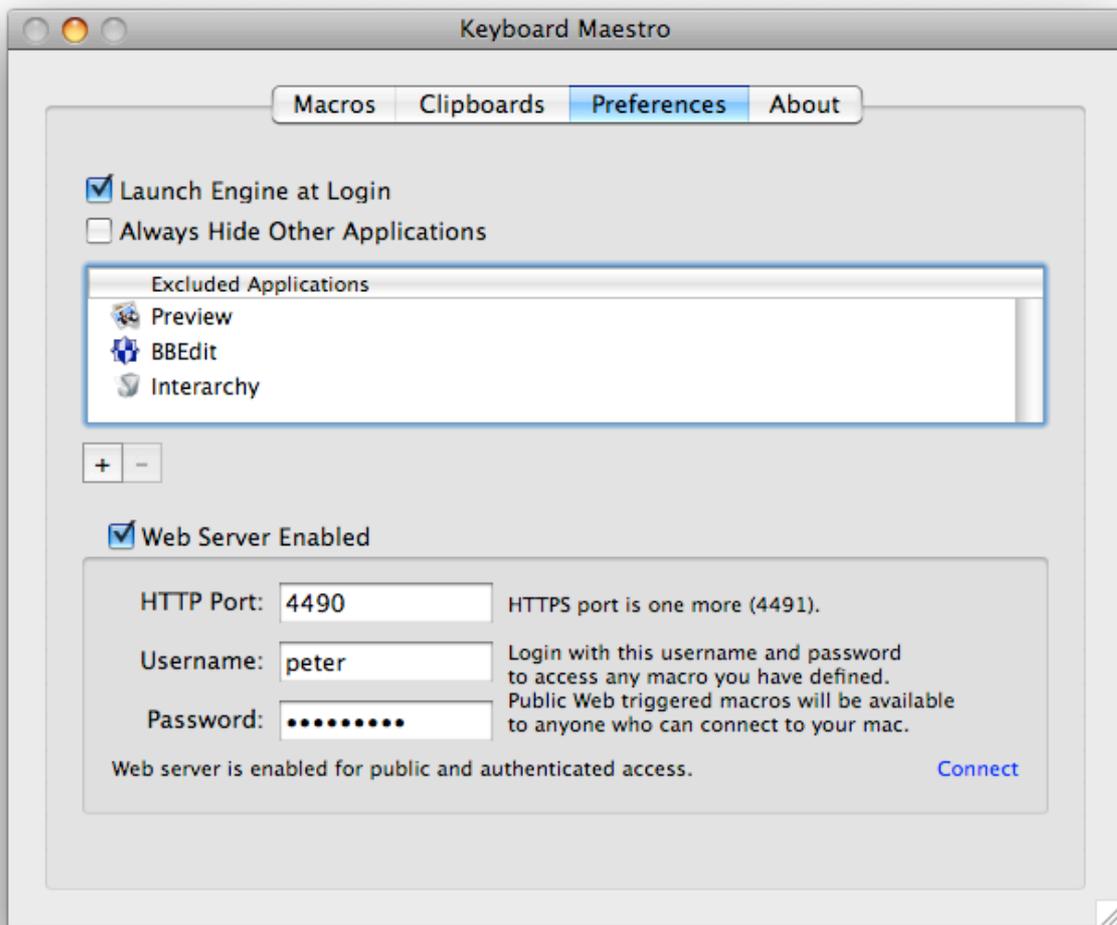


To learn more about the Clipboard History Switcher, see the [Clipboard History Switcher](#) section.

Preferences Window Pane

This window pane lets you configure Keyboard Maestro.

You get this window pane by launching Keyboard Maestro and clicking the **Preferences** tab.



To learn more about the Preferences, see the [Preferences](#) section.

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 clicking the [About](#) tab.



Menus

Keyboard Maestro

- [About Keyboard Maestro](#)
- [Purchase Keyboard Maestro](#)
- [Register Keyboard Maestro](#)
- [Check For Updates](#)
- [Preferences](#)
- [Services](#)
- [Hide Keyboard Maestro](#)
- [Hide Others](#)
- [Show All](#)
- [Quit Keyboard Maestro](#)

File

- [Close Window](#)
- [Export Macros](#)
- [Import Macros](#)
- [Quit Engine](#)
- [Launch Engine](#)

Edit

- [Undo](#)
- [Redo](#)
- [Cut](#)
- [Copy](#)
- [Paste](#)
- [Clear](#)
- [Select All](#)

Window

- [Minimize Window](#)

- [Zoom](#)
- [Bring All to Front](#)

[Help](#)

- [Keyboard Maestro Help](#)
- [Online Documentation](#)
- [Keyboard Maestro Web Site](#)
- [Stairways Software Web Site](#)

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 Updates

The [Check For Updates 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 » Close Window

The [Close Window 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 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.

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 » 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](#), [Program 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.

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](#), [Program 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.

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. Most operations in Keyboard Maestro cannot be undone, so use caution.

Edit » Redo

The [Redo command](#) in the [Edit menu](#) redoes the previous undone command. Most operations in Keyboard Maestro cannot be undone, so use caution.

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 » Paste

The [Paste command](#) in the [Edit menu](#) pastes the current system clipboard into the current selection.

Edit » Clear

The [Clear 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.

Window

The [Window menu](#) contains menu items relating to windows.

Window » Minimize Window

The [Minimize Window command](#) in the [Window menu](#) minimizes the front window.

Window » Zoom

The [Zoom command](#) in the [Window menu](#) zooms the front window.

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 Help

The [Keyboard Maestro Help command](#) in the [Help menu](#) takes you to [Help Viewer](#) and displays the Keyboard Maestro local documentation.

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.

Tips

- [Remembering Macro Hot Keys](#)
- [Use Function Keys for Global Hot Keys](#)
- [Use the Number Pad](#)

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.

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/Switching does not work after I login, what's wrong?](#)
- [My Macros are not working, what's wrong?](#)
- [The Window Switcher shows an empty list, what's wrong?](#)
- [The Program, Window and Clipboard Switcher does not work at all, what's wrong?](#)
- [I want to use command-Tab for the Program Switcher, but the system overrides it, what's wrong?](#)
- [How do I get more help?](#)

Macros/Switching does not work after I login, what's wrong?

The Macros/Switching was 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 [Preferences window 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.

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 Program, Window and Clipboard Switcher does not work at all, what's wrong?

The [Hot Keys](#) for the Program, 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 Program Switcher, but the system overrides it, what's wrong?

Apple have claimed the command-Tab keystroke and refuse to allow third parties to override this or users to disable this. While we disagree with this behavior by Apple, we are not willing to implement a hack solution that may cause compatibility problems, especially in light of this being a deliberate action. We write software that is as robust as we can make it and are not likely to add solutions that attempt to specifically do things Apple does not want us to do.

Under 10.4 (Tiger), you can use [PullTab](#) together with [Unsanity APE](#) to disable the command-Tab facility in the Dock and then you will be able to use Keyboard Maestro with command-Tab. There is no known solution for 10.5 (Leopard).

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!

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 [Program Switcher](#) list, allowing you to hide programs you rarely want to switch to. These programs 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.

Help Viewer
Apple's Help Viewer application is the standard [Mac OS X](#) help system, used for display application and system documentation. Significantly degraded in 10.5 (Leopard), but what can you do?

Hot Key
A keystroke that acts as a [Macro Trigger](#) to start the execution of [Macro Actions](#) in a Macro.

Keyboard Maestro Engine
The process that enables your [Macros](#), [Program Switcher](#), [Window Switcher](#), and [Clipboard Switcher](#) to work even after you quit Keyboard Maestro.

Mac OS X
Apple's operating system versions 10.0 and up. Keyboard Maestro only supports version 10.4 and up, and 10.4.11 and up is recommended.

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 [Program 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(#recording).

Record Quick Macro
the action that when triggered records a [Quick Macro](#). SEE(#recording).

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).

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/".

Administrative Details

- [Requirements](#)
- [Distribution](#)
- [History](#)
- [Credits](#)
- [Warranty](#)
- [Licenses](#)
- [Fine Print](#)

Requirements

Keyboard Maestro 3 requires [Mac OS X](#) 10.4 or later.

Distribution

You may distribute this program 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 [Program 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 Program 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 program 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.

Going forward, we plan to aggressively develop Keyboard Maestro, 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 the many others who have provided input and support over the past decade.

Warranty

This program 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 2008 Stairways Software Pty Ltd. All Rights Reserved. You may use this program for a short trial period and then you must purchase the program or stop using it.

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, keyboardmaestro.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.