

Keyboard Maestro 4 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 a previous version of Keyboard Maestro?](#)

[Macros](#)

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

[Macro Actions](#)

- [Overview](#)
- [Process Control](#)
- [Interface Control](#)
- [iTunes Control](#)
- [QuickTime Player Control](#)
- [System Control](#)
- [MIDI](#)
- [Open](#)
- [Execute](#)
- [Text](#)
- [Clipboard](#)
- [Activate Switcher](#)
- [Record Quick Macro](#)
- [Google](#)
- [Pause](#)
- [Growl](#)
- [System Beep](#)
- [Alert](#)
- [Comment](#)
- [Cancel All Macros](#)

[Macro Triggers](#)

- [Overview](#)
- [Hot Key](#)
- [Typed String](#)
- [Application](#)
- [Wake](#)
- [Login](#)
- [Time](#)
- [While Logged In](#)
- [Macro Palette](#)
- [Status Menu](#)
- [Public Web](#)
- [Device Key](#)
- [MIDI Note](#)

[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](#)

- [General Preferences](#)
- [Web Server Preferences](#)
- [Clipboards Preferences](#)

- [Excluded Preferences](#)
- [Other Hidden Preferences](#)

[Scripting](#)

- [Controlling Keyboard Maestro via Scripting](#)
- [Executing Scripts](#)

[Windows](#)

- [Welcome Window](#)
- [Macros Window](#)
- [Tutorial](#)
- [Macro Group Editor](#)
- [Macro Editor Window](#)
- [Recording Window](#)
- [Program Switcher Window](#)
- [Window Switcher Window](#)
- [Clipboard Switcher Window](#)
- [Clipboard History Switcher Window](#)
- [Preferences Window](#)
- [Preferences General Pane](#)
- [Preferences Web Server Pane](#)
- [Preferences Clipboards Pane](#)
- [Preferences Exclude Pane](#)
- [About Window Pane](#)

[Menus](#)

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

[Tips](#)

[Troubleshooting](#)

[Support](#)

[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, iTunes or QuickTime Player; and more — all with the touch of a key, click of the mouse, periodically or at specific times, when an application is launched or running, or even over the Internet using Keyboard Maestro Control on your iPhone or iPod touch, or 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, or 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 previous 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 free for a few weeks, but then you must [purchase a license](#) to continue using it.

What's New

Keyboard Maestro 4 brings an entirely new and modernized user interface fit for Snow Leopard (while retaining compatibility with Leopard). The new interface draws on familiar Mac OS X applications to quickly get you devising your own macros.

New in 4.3.2

- Improved the look of disabled actions to make them more obviously disabled.
- Added a short delay before saving to coalesce saving, especially when typing.
- Remove trailing line feed from Execute AppleScript
- Remove sorting characters "xx)" from KeyCue display.
- Added configurable command for osascript (specifically to allow 32-bit execution).
- Fixed a crash in Execute Workflow/Execute AppleScript when the file has disappeared.
- Ensure changes to the file/text popup menu in the Execute Script editor sets the modified flag.

New in 4.3.1

- Fixed a potential CPU issue when executing AppleScripts and shell scripts.
- Fixed a potential crash with Device triggers.
- Removed the reset of the front application from the InsertText macro.
- Fixed a cosmetic issue in the Execute Script action.
- Added support for naming a few more X-Key keyboard keys.
- Improved handling of negative numbers in the Move Window action.

New in 4.3

- Device triggers from modifiers, mouse buttons, and programmable keyboards.
- Execute AppleScript & Shell script can insert their results by typing or pasting.

- Shell scripts no longer lock the engine up on timeouts.
- AppleScripts no longer lock the engine up.
- Added command-option-Q/L for Quit/Launch Engine.
- do script only executes a single macro.
- do script properly waits for only the macro it executes.
- Do not uppercase accented characters when displaying command keys/hot keys.
- While Running application trigger is no longer executed every time any macro changed.
- Avoid creating OLE LINK bookmarks in Word for every copy/paste.
- Default again to releasing only keys that have to be released when simulating a keystroke.
- Fixed a crash when deleting a named clipboard while it is editing the name.
- Attempt to select menus even if they are dimmed if "stop on failure" is turned off.
- Close the first duplicate macro palette before displaying a second one.

New in 4.2

- Edit macros from the Status Menu or floating palettes with the option key.
- Execute Macro action.
- MIDI triggers.
- MIDI actions for Note On, Note Off, Control Change.
- Display a palette of macros when a duplicate hot key is pressed.
- Select Menu Item action optionally cancels macro if the menu is not executed.
- Press Button action optionally cancels macro if the button is not pressed.
- Add an option to Activate Application to reopen the initial windows.
- Add an option to Activate Application to switch out, hide or quit if it is already at the front.
- Offer to email helpful tips to new users.
- Allow Typed String triggers with returns in them to work properly.
- Logging for AppleScript errors.
- Edit Macro, bring macros window to the front.
- Fixed unlocalized title of Insert Text and Set Clipboard To Text views.
- Fixed a potentially crashing bug in the Select Menu Item action editor.
- Fixed odd resorting glitch in the macros list.

New in 4.1.1

- Fixed a crashing bug in Select Menu Item
- Added support for editMacro AppleScript command

New in 4.1

- Welcome Page.
- In-app tutorial reminiscent of Apple Guide.
- In-app documentation.
- Help for Macro group activation and targeting.
- Sort macros by name or trigger.
- Improved Select Menu editor to allow selecting from all current menu items.
- Allow "or" comparisons to handle toggle menus (eg Show|Hide).
- Allow .../ellipsis comparisons for buttons (eg Print...).
- Added menu commands for various actions like Edit/Show Actions/Record.
- Added support for 01) prefixes in macro names for sorting status menus, macro palette.
- Icons in floating palette were upside down!
- Switch back to original app on cancel of Google Search.
- Disabled Typed String triggers while entering typed string triggers.
- Disabled Typed String triggers while simulating keystrokes.

New in 4.0.2

- Support importing macros by double clicking/drag & drop.
- Fixed command-Tab handler getting lost.
- Allow command-shift-Tab as a hot key.
- Add "Other" to Hot Key menu to make it clear any key can be used.
- Adjusted the look of the Hot Key view.
- Allow Typed String triggers with only a single character.
- Fixed a bug with ICUDateTime parsing.
- Fixed an issue with Dvorak keyboards.
- Fixed a crash with the Hot Key popup menu.
- Restricted Engine to 32 bit mode in 10.5.
- Fixed crash while recording in 10.5 in 64-bit.
- Fixed sound actions (mute/up/down) in 10.5 in 64-bit.
- Fixed program switcher grid view to display < 8 apps.
- Restrict login trigger macros to macros that are actually enabled.

New in 4.0.1

- Fixed the slow simulating keystrokes issue.
- Fixed a crash involving setLaunchEngineAtLogin on 10.5
- Fixed it so simulating keys will not fire hot key triggers.
- Added 10 second emergency timeout on trigger repeat.
- Set Remap Forward Delete macro to disabled by default.
- Fixed a case where multiple engines could be running.
- Added support for Growling macro execution.
- Added a bit of color and made the text larger in the non-edit mode.
- Fixed typo in Manipulate Window action menu.
- Fixed typo in Preferences window.

New in 4.0

- Entirely new User Interface.
- 64-bit.
- Allow command-Tab as a hot key.
- Execute macros when hot keys are pressed, held down or released.
- Support for [Growl](#) notifications.
- Enhanced and extended clipboard switcher including searching, image and rich text display.
- Enhanced program switcher including switch to an application and hide all others.
- Enhanced Status Menu with grouped/hierarchical items.
- Allow a hot key to execute multiple macros.

Simplified

- 10.5+ only.
- Removed support for version 2 preference files.

Enhanced Actions

- Google search action.
- **Growl** action.
- Control QuickTime Player.
- Show/Hide specific applications.
- Show all applications.

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 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 Key](#) - make [Macros](#) available at the press of a key.
 - [Typed String](#) - type a short string to execute an action.
 - [Applications](#) - on launch, quit, activate, deactivate, or periodically while an application is running or active.
 - [Login](#) - on login.
 - [Time](#) - at a particular time of day, or periodically.
 - [Wake](#) - on system wake.
 - [Macro Palette](#) - with a click on a context sensitive [Macro Palette](#).
 - [Status Menu](#) - select from a global system status menu.
 - [Public Web](#) - over the Internet, explicitly to the public, or via authenticated logins.
 - [Device Key](#) - when any HID device key is pressed or released.
 - [MIDI Note](#) - when a MIDI note is pressed or released.
- Create [Macro Actions](#) by recording your actions.
 - [iPhone or iPod touch](#) - from your iPhone or iPod touch.
- [Macros](#) can execute a sequence of one or more [Macro Actions](#) including:
 - Process Control
 - Switch to Last, Next or a specific application.
 - Quit All, Others or a specific application.
 - Hide All, Others, a specific, or the current application.
 - Show All or a specific application.
 - Bring the current application's windows to the front.
 - Interface Control
 - Manipulate the front or a named window - resize, move, center, close, zoom move, minimize, bring to front.
 - Move or click the mouse with modifiers.
 - 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, pause or stop.
 - Rewind or fast forward.
 - Go to the next or previous track.
 - Increase, decrease or set the volume or the rating of the current song.
 - Control QuickTime Player
 - Play or pause the current movie.
 - Step backward or forward.
 - Increase, decrease or set the volume.
 - Control the System
 - Sleep, Shutdown, Restart or Log Out or Fast User Switch.
 - Activate screen saver.
 - Open/close CD tray.
 - Increase, decrease, set or toggle mute the system volume.
 - Increase or decrease the brightness.
 - Open a file, folder, [URL](#) or system preferences pane.
 - Execute an AppleScript or shell script from a file or inline text, optionally displaying the results.
 - Execute an Automator workflow.
 - Insert Text either by typing or pasting, with token expansion for various time and dates and other system information.
 - Control the Clipboard
 - Simulate Cut, Copy or Paste.
 - Copy, Cut or Paste to/from a permanent [Named Clipboard](#).
 - Set the system clipboard to specific tokenized text, a past copy of the clipboard, or a [Named Clipboard](#).
 - Save the system clipboard to a permanent [Named Clipboard](#).
 - 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.
 - Display an Alert, [Growl](#) or Beep.
 - Cancel all macros.

Program and Window Switcher

- Optionally replace the system command-Tab process switcher.
- Switch to any application or window with a keystroke.
- Switch to an application and hide all others.
- Easily select the exact application or window you want.
- Launch, hide, 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 are saved permanently.

Clipboard History Switcher

- Never lose your clipboard again.
- Browse the past clipboards, pasting any previous clipboard item.
- Clipboards are displayed with rich text and images.
- Clipboard History is optionally 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 4 is a paid upgrade from previous versions. Customers who purchased Keyboard Maestro after 1 April 2009 have been issued a free upgrade to Keyboard Maestro 4. If you have not received your free license, you can claim your free upgrade by looking up your Keyboard Maestro license at <http://enquiry.stairways.com/>.

Customers who purchased Keyboard Maestro prior to April 2009 can upgrade to Keyboard Maestro 4 for US\$18 until 31 January 2010. If you have not received your instructions on how to upgrade, you can find details by looking up your Keyboard Maestro license at <http://enquiry.stairways.com/>.

Customers who have not disabled upgrade emails have been emailed with new license or upgrade instructions as appropriate. If you have not received 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 downloaded from <http://download.stairways.com/>.

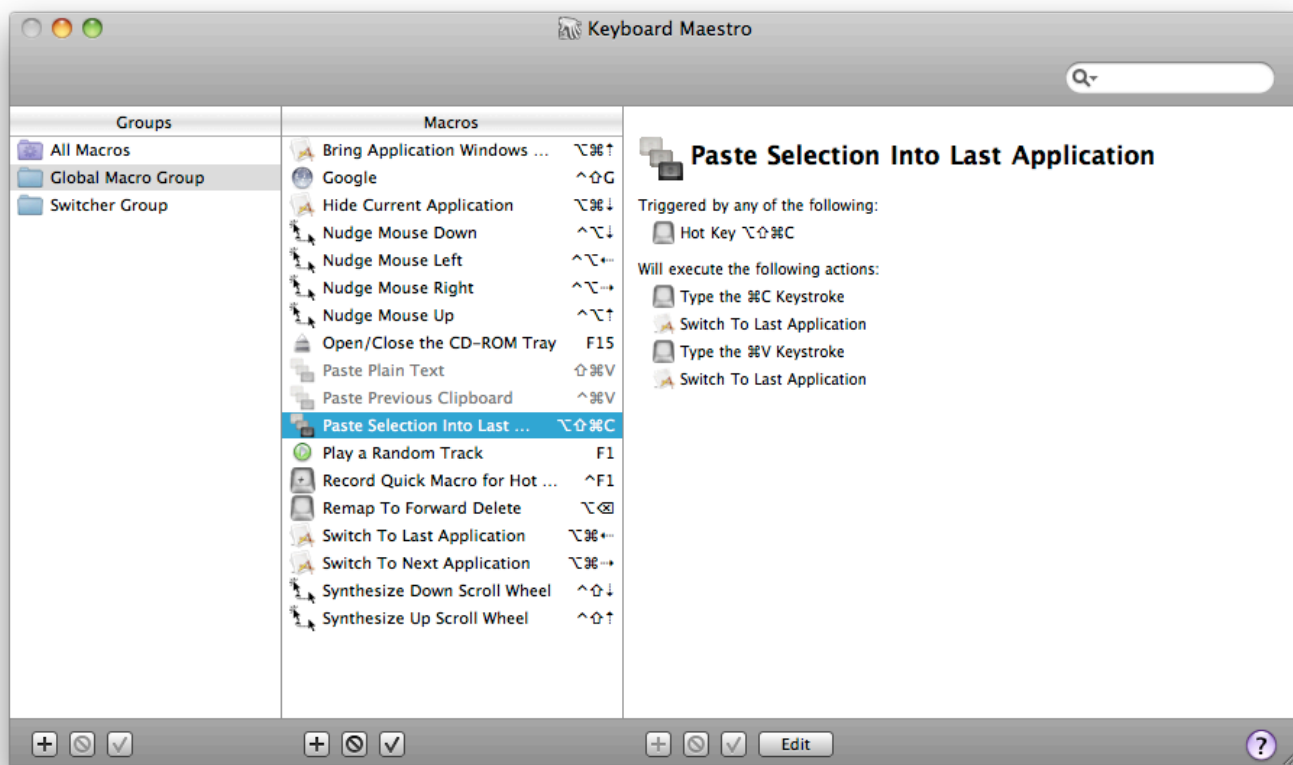
For sales enquires, customer service, technical support, or to contact project management, our current contact information is listed at <http://contact.stairways.com/>.

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

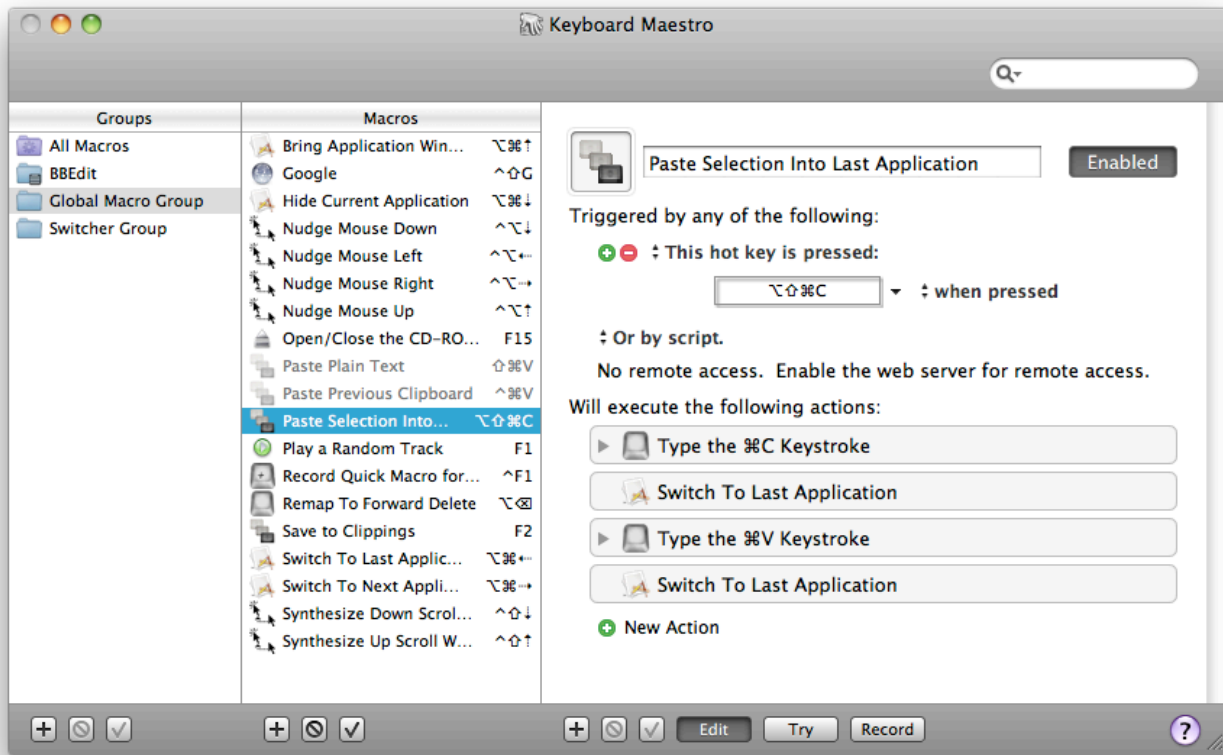
Screenshots

This is a very quick taste of Keyboard Maestro.

Macros Window



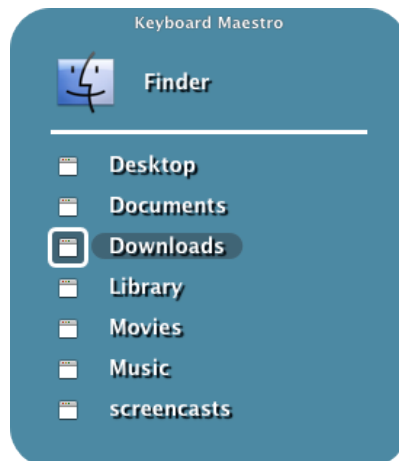
Macro Editor



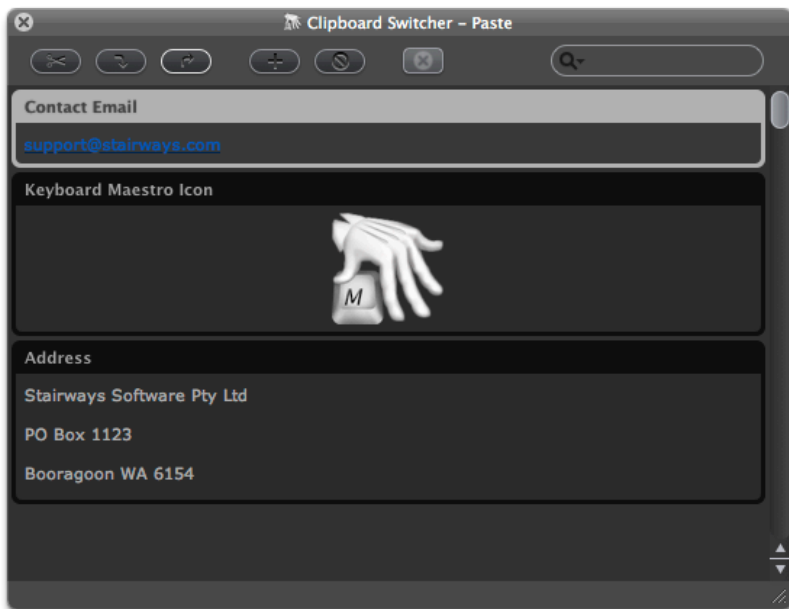
Program Switcher Window



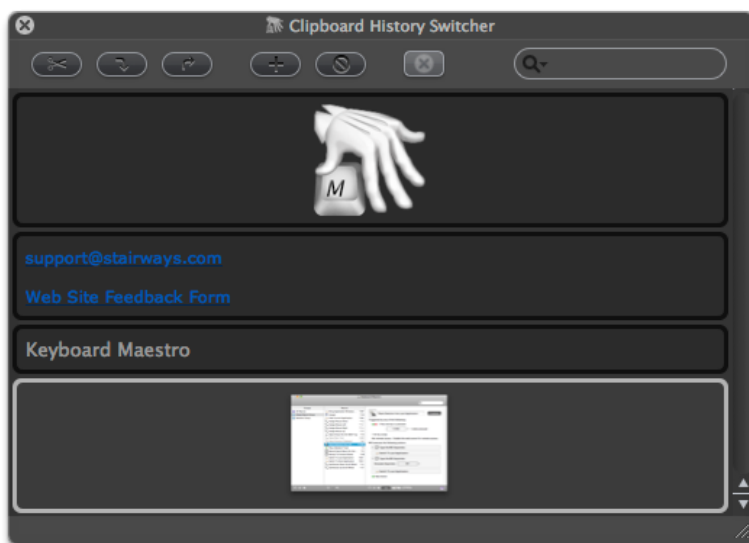
Window Switcher Window



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 [Welcome window](#).

Welcome to Keyboard Maestro



Welcome to Keyboard Maestro 4

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

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

[Start Tutorial](#)

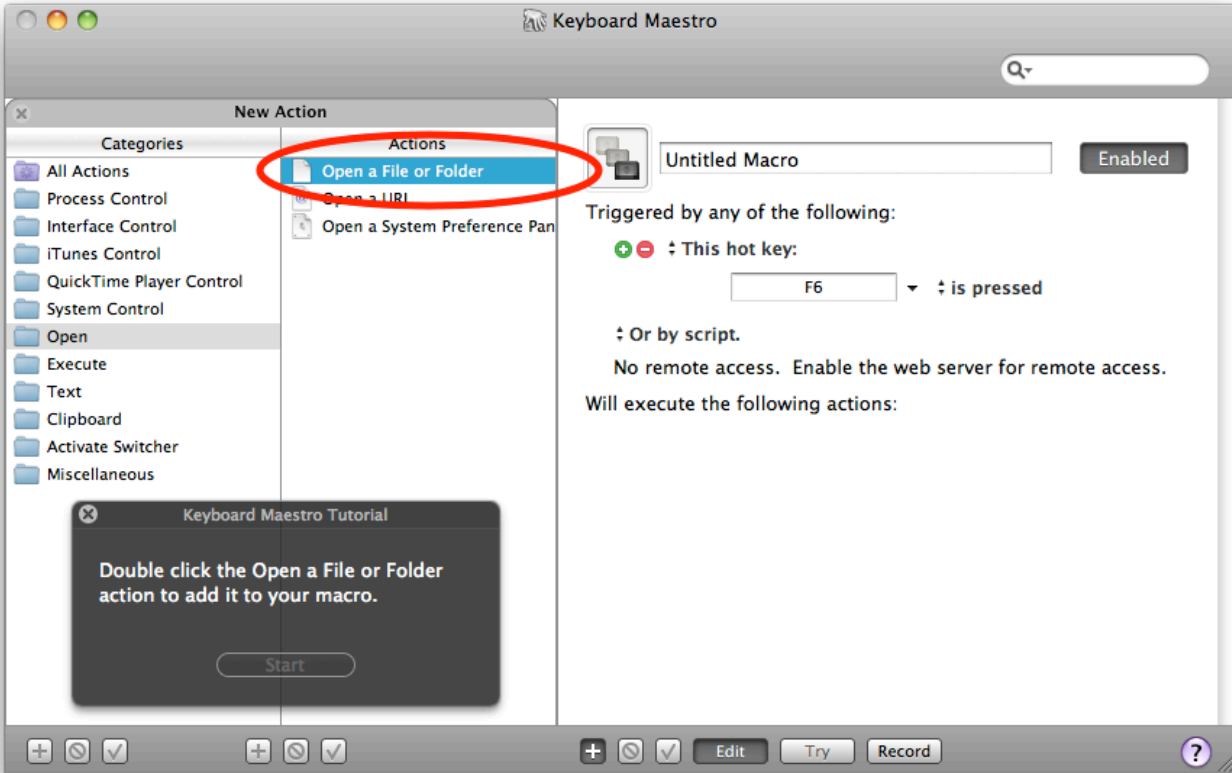
Show this window when Keyboard Maestro opens

Quick Start
Learn about the components of Keyboard Maestro and how they work together.
[Learn more >](#)

Assistance
Learn about the user group, documentation, videos, and customer support options.
[Learn more >](#)

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

Keyboard Maestro



New Action

Categories: All Actions, Process Control, Interface Control, iTunes Control, QuickTime Player Control, System Control, Open, Execute, Text, Clipboard, Activate Switcher, Miscellaneous

Actions

- Open a File or Folder
- Open a URI
- Open a System Preference Panel

Untitled Macro Enabled

Triggered by any of the following:

- This hot key: is pressed
- Or by script.

No remote access. Enable the web server for remote access.

Will execute the following actions:

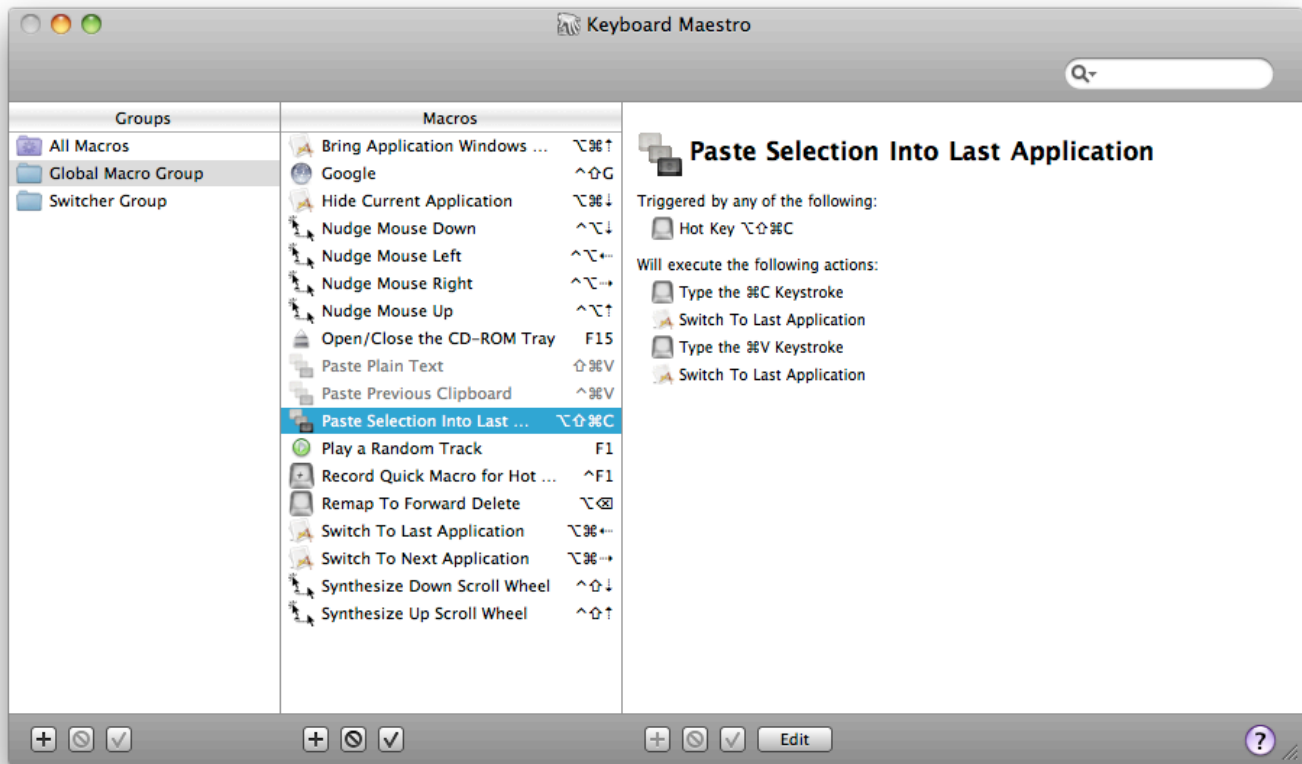
Keyboard Maestro Tutorial

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

[Start](#)

[Edit](#) [Try](#) [Record](#)

Close the Welcome window to display the Macros window.



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

Make New Macro

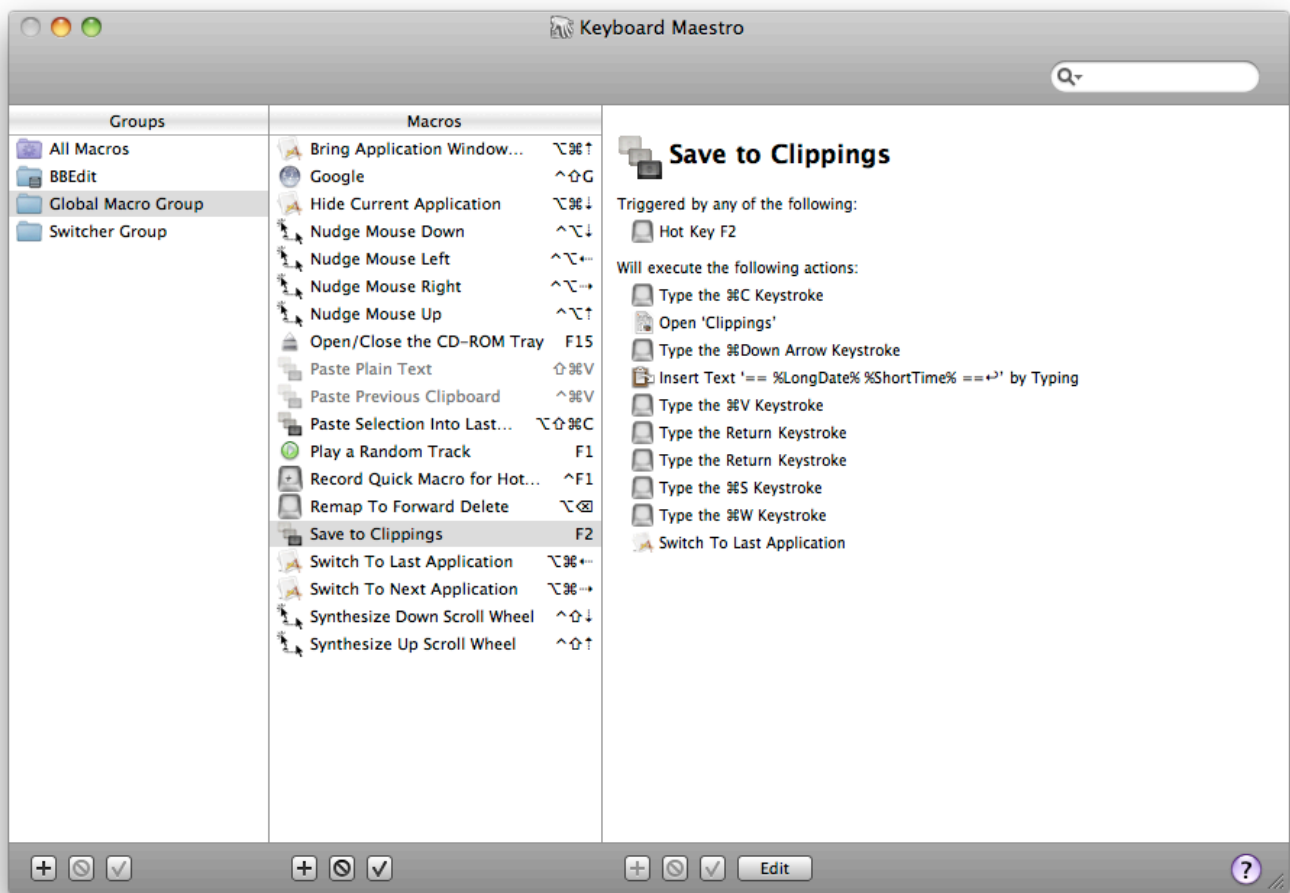
Click the **+** button under the Macro column to add a macro and display the [Macro Editor window](#). We will now design quite a complex Macro that will allow you to save clippings to a text file. First, launch TextEdit and create a new empty document and save it to **Clippings.rtf** in your **Documents** 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.
- Go to the end of the file.
- Press return and type a line of dashes and return to separate the clippings.
- Press command-V to paste the clipping.
- 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!

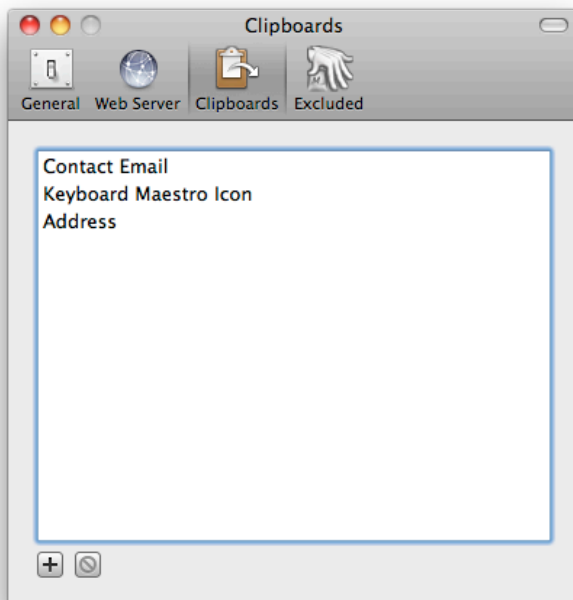
So lets define the whole sequence as a Macro.



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

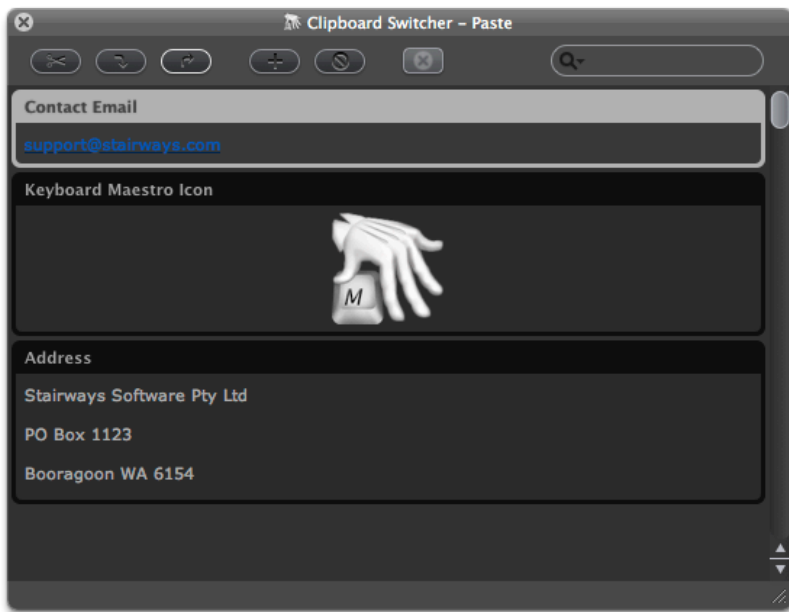
Named Clipboards

Keyboard Maestro lets you create as many named clipboards as you want using the [Clipboards preference pane](#).



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.

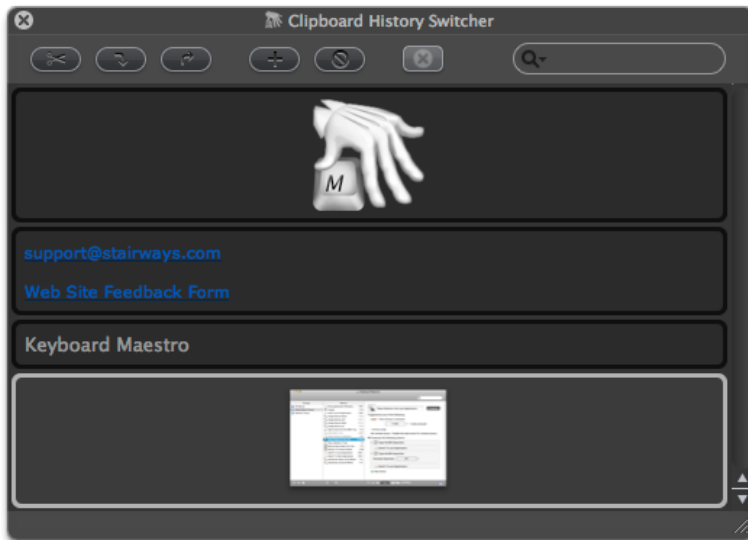
Then you can cut, copy or paste from the named clipboards using the defined [Hot Keys](#) (by default, command-shift X, C and V respectively).



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

Clipboard History

For less permanent information, Keyboard Maestro automatically remembers your clipboard history, saving a copy of each new clipboard 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 (by toggling the **Close After Action** button) and paste previous clipboards by simply double clicking.

You can also have your clipboard history saved across reboots by enabling the "Save Clipboard History" preference in the [General preference pane](#).

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. The Program Switcher is activated like any other macro, so you can find it in the Switcher Group in the [Macros window](#). By triggering the [Program Switcher](#) macro (for example, by pressing option-Tab) you can display the [Program Switcher](#) window.

As of Keyboard Maestro 4, you can use command-Tab as the macro hot key if you like and replace the system program switcher.

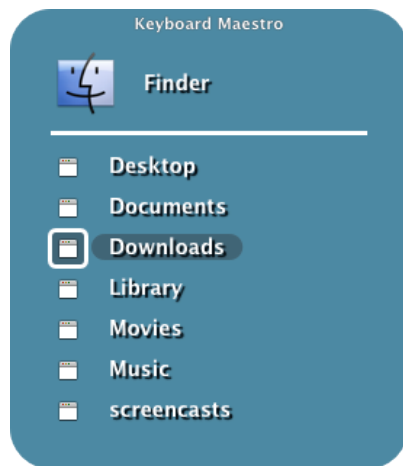


Note the applications set to be launched or quit.

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.

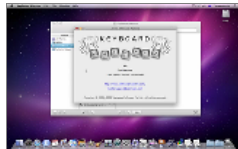


Note the windows set to be closed. This is a very useful way of cleaning up an application with a lot of open windows, especially prevalent in the Finder.

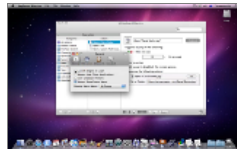
Conduct Your Mac Like a Pro!

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



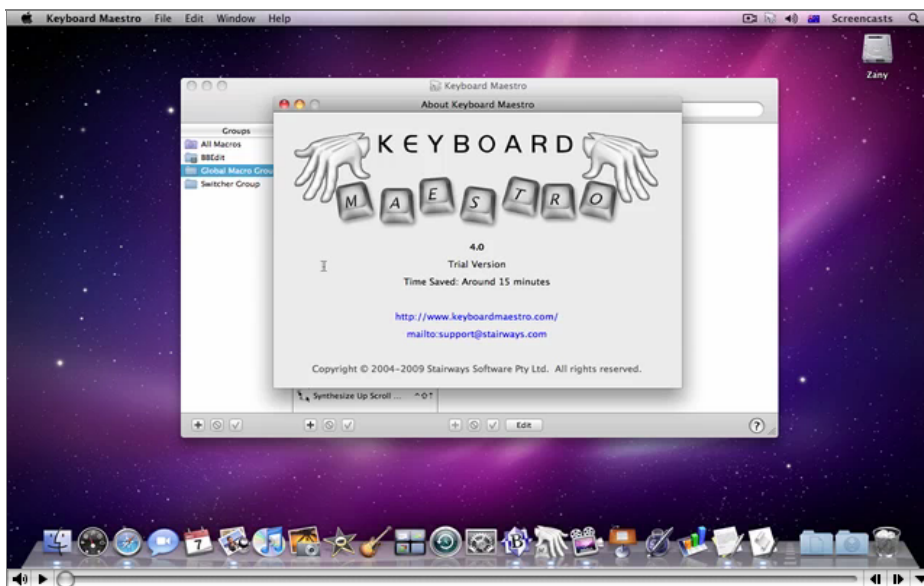
Overview



Configuration



Save to Clippings Tutorial



Thanks to [Dan Benjamin](#) for doing the fantastic voice overs on these tutorial videos.

Links

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

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

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

You can join the [Keyboard Maestro User Group](#) online community consisting of the developers and Keyboard Maestro users at <http://user.keyboardmaestro.com/>.

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

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

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

Quick Start

Keyboard Maestro is 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 always need to be running. Whenever you launch Keyboard Maestro, it also launches the [Keyboard Maestro Engine](#) which continues running until you log out (you can have the [Keyboard Maestro Engine](#) launched automatically when you login by enabling the

“Launch Engine at Login” preference in the [General preference pane](#)).

The [Keyboard Maestro Engine](#) is a background only application that enables all of Keyboard Maestro’s features. It responds to your [Hot Key](#) presses, watches the time, tracks applications and the clipboard, handles remote web and iPhone 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 in the [General preference pane](#).

Keyboard Maestro organizes your macros into [Macro Groups](#) which are like folders of macros. Each [Macro Group](#) controls when the macros it contains are active. A [Macro Group](#) can target or exclude specific applications, which means the macros it contains will only be active in those desired applications. For example, you can have macros which are active only in Mail.app. A [Macro Group](#) can also act as a container for specific-use macros which are enabled only after a [Hot Key](#) press or which are displayed as a palette of the macros. For example, you could create a [Macro Group](#) containing macros that resized or repositioned windows using the arrow keys, but those macros would only be active after the [Hot Key](#) was pressed so that the arrow keys could be used normally at other times. You create a [Macro Group](#) by clicking the + button at the bottom of the [Macro Groups](#) list. You can disable or enable [Macro Groups](#) by clicking the ✓ button. You can configure a [Macro Group](#) by selecting it and clicking the Edit button.

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

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

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

By using these six things (Editor and Engine, [Macro Groups](#) and Macros, Triggers and Actions) together, you can dramatically enhance your Mac 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 a previous version of Keyboard Maestro?](#)

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 “Launch Engine at Login” preference in the [General preference pane](#) to ensure all of Keyboard Maestro’s facilities are available to you as soon as you login or startup your Mac.

How do I upgrade Keyboard Maestro?

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

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

Keyboard Maestro will automatically import your version 3 macros, clipboards and preferences. Your old macros will be saved in the [~/Library/Preference/Keyboard Maestro/Keyboard Maestro Macros Saved Version 3.plist](#) in case you decide not to upgrade to version 4 for

any reason.

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

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

How do I purchase Keyboard Maestro?

New customers can purchase 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 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](#)) and also that your license matches the major version number (eg, a version 4 license will work with version 4.x of Keyboard Maestro).

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](#) or Status Menu trigger.

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

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

See also the [Tips](#) section.

How do I create a new Macro?

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

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

How do I Insert Styled/Colored Text or Images?

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

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

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

How do I Insert the Current Date?

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

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 six 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, select the Switcher Group, and double click the Program Switcher macro.

You can then configure the various Program Switcher parameters, such as style and icon size by configuring the Program Switcher action.

How do I uninstall Keyboard Maestro?

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

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

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

For version 2, open the `~/Library/Preference` folder and the folder `~/Library/Preference/Keyboard Maestro/Saved Version 2` folder. Move the files from the latter 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.

For version 3, open the `~/Library/Preference/Keyboard Maestro` folder. Trash the `Keyboard Maestro Macros.plist` and replace it with the [Keyboard Maestro Macros Saved Version 3.plist](#)

Download Keyboard Maestro 3.5, if necessary, from <http://files.stairways.com/keyboardmaestro/keyboardmaestro-35.zip>. Move Keyboard Maestro 3 to the Applications folder and launch it. Turn on the "Launch Engine at Login" preference if desired.

Macros

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

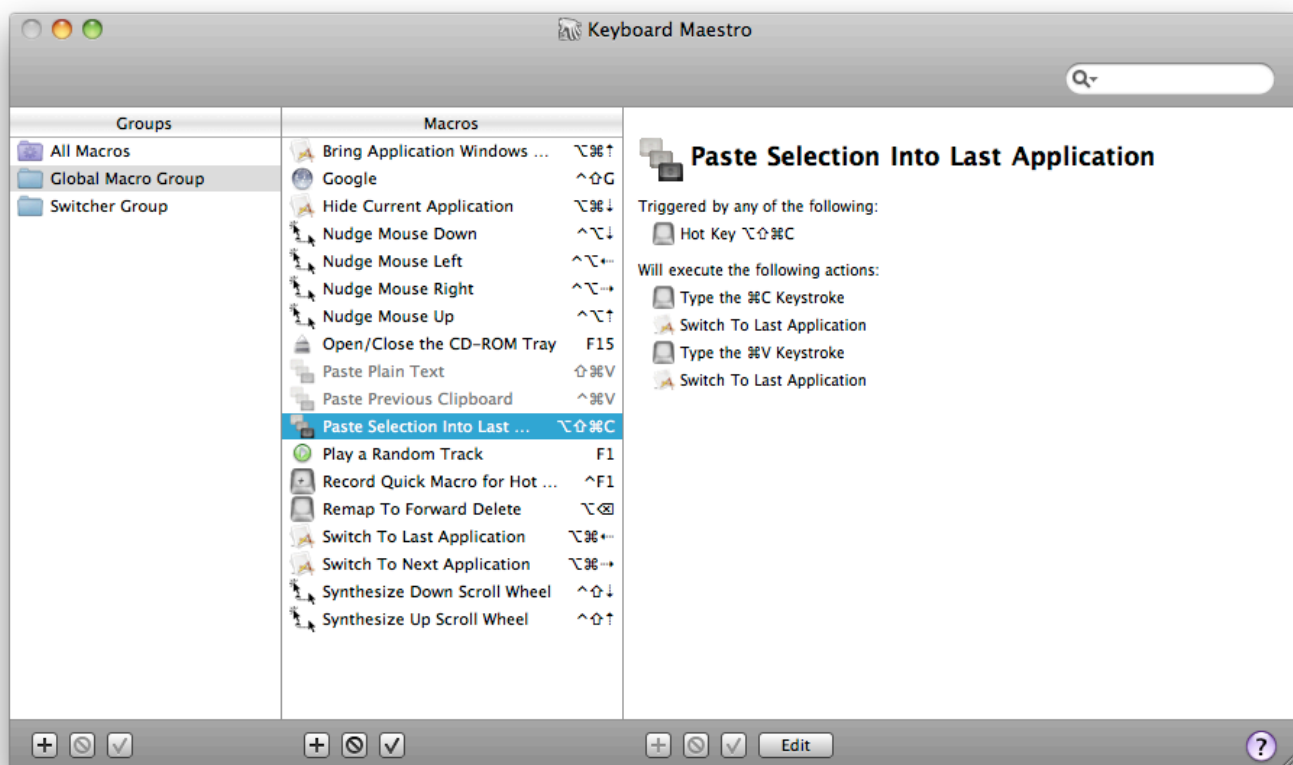
Overview

[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](#).

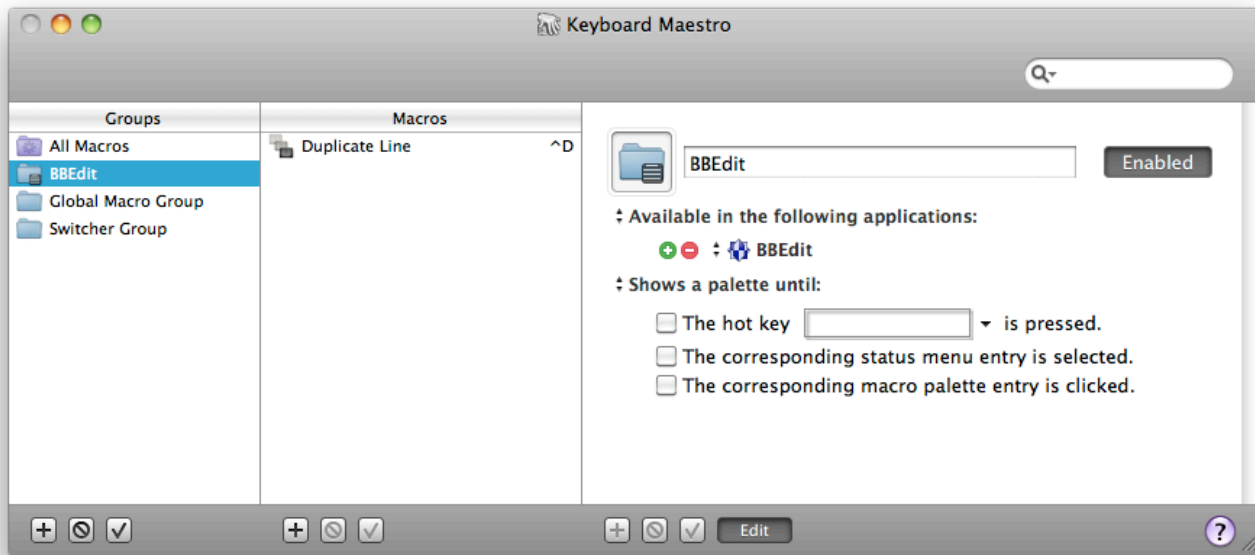
Macro Groups

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.



Now click the + button below the [Macro Groups](#) list.



Enter the name of your new Macro Group.

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

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

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

For example, you could have macros targeted at:

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

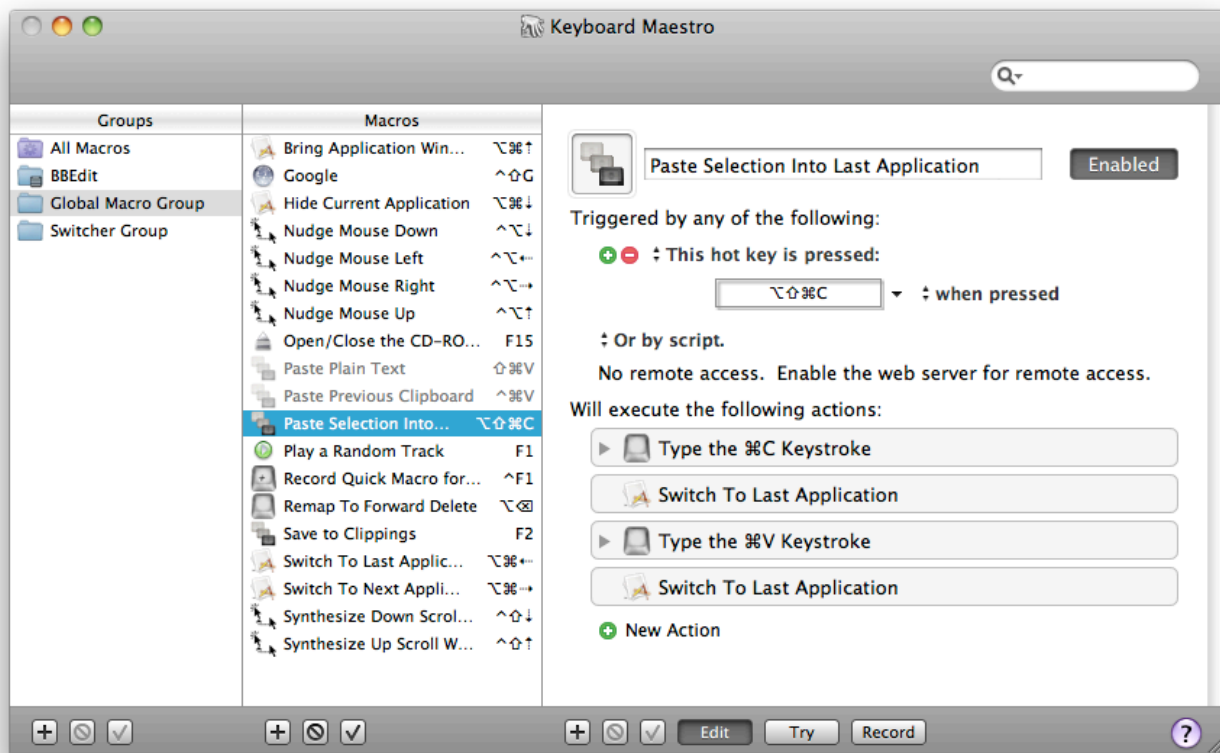
To delete a [Macro Group](#), select the macro group and then press the Delete key or click the ⓧ button at the bottom of the Groups list.

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

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, select the desired Macro Group to contain it, and then click the + button below the Macros list. To edit a Macro, select it and click the [Edit](#) button (or double click it). 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 by clicking on the green ⊕ button. There are several [Macro Triggers](#) to choose from, the most common being the [Hot Key](#) which allows you to execute the Macro at the press of a key. You can define several different triggers and any of the triggers will execute the [Macro Actions](#). You can delete a trigger by clicking the red ⊖ button.

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

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

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

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

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

You execute a macro's action sequence by triggering the macro using any of the [Macro Triggers](#) you have defined.

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

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

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

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

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

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

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

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

Macro Actions

- [Overview](#)
- [Process Control](#)
- [Interface Control](#)
- [iTunes Control](#)
- [QuickTime Player Control](#)
- [System Control](#)
- [MIDI](#)
- [Open](#)
- [Execute](#)
- [Text](#)
- [Clipboard](#)
- [Activate Switcher](#)
- [Record Quick Macro](#)
- [Google](#)
- [Pause](#)
- [Growl](#)
- [System Beep](#)
- [Alert](#)
- [Comment](#)
- [Cancel All Macros](#)

Overview

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

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

Process Control

Process Control actions allow you to switch, quit, or hide applications. The actions are:

Switch to Last Application

switches back to the previous application you were in.

Switch to Next Application

switches to the next application (alphabetically).

Activate a Specific Application

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

Bring Application Windows to Front

brings all the windows of the current application to the front.

Quit All Applications

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

Quit Other Applications

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

Quit a 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.

Hide a Specific Application

hides the specified application.

Show All Applications

shows all foreground applications.

Show a Specific Application

shows the specified 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.

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

The Select Menu and Press Button actions allow you to specify multiple options separated by a vertical bar (eg Show|Hide) to allow for toggling menus. They will also ignore the difference between three dots (...) and an ellipsis so you do not have to worry which one the menu uses.

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

iTunes Control

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

- Play a Specific Track
 - play a specified song.
- Play a Specific Playlist
 - play the songs in a specified Playlist.
- Play a Random Track
 - play a random song.
- Play a Random Track from a Specified Playlist
 - play a random song from a specified Playlist.
- Play/Pause Current Track
 - toggle from playing to pausing or vice-versa.
- Pause Current Track
 - pause the current song.
- Stop Current Track
 - stop playing any song.
- Fast-Forward Current Track
 - fast-forward the current song. It will keep fast-forwarding until you do something else or until it reaches the end of the song.
- Rewind Current Track
 - rewind the current song. It will keep rewinding until you do something else or until it reaches the beginning of the song.
- Next Track
 - play the next song.
- Previous Track
 - Go to the previous song or the start of the current song if it is already playing.
- Increase/Decrease or Set iTunes Volume
 - Increase, decrease or set the iTunes volume.
- Increase/Decrease or Set Rating
 - Increase, decrease or set the rating of the current track.

QuickTime Player Control

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

- Play Current Movie
 - play the current movie.
- Play/Pause Current Movie
 - toggle from playing to pausing or vice-versa.
- Pause Current Movie
 - pause the current movie.
- Step Forward Current Movie
 - step the current movie forward one frame.
- Step Backward Current Movie
 - step the current movie backward one frame.
- Increase/Decrease or Set QuickTime Player Volume
 - Increase, decrease or set the QuickTime Player volume.

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.
- Fast User Switcher
 - Fast User Switcher to a specific 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.
- Increase/Decrease Screen Brightness
 - Increases or decreases the screen brightness.

MIDI

Send a MIDI message:

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

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

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

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

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

Execute Automator Workflow
execute a specified Automator Workflow.

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

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

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

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

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

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

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

Shell scripts are executed in the background and will be abandoned after 20 seconds, allowed to continue running to completion but otherwise ignored.

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

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

Text

The Insert Text action allows you to insert specified text either by setting the system clipboard and pasting, or by simulating a sequence of keystrokes.

Tokens are provided to include such things as the time or date.

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

Insert Text can expand various tokens, including dates in any [ICU 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.
 */
```

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.

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.

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

Set Clipboard to Named Clipboard
set the system clipboard to a specified [Named Clipboard](#).

Save Clipboard to Named Clipboard
set the specified [Named Clipboard](#) to the current system clipboard.

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

Filter Clipboard
apply one of a set of filters on the current system clipboard (see below for a list).

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.
- Trim Whitespace.
- Hard Wrap or Unwrap text.
- Change case (Uppercase, Lowercase, Capitalize and John Gruber's Title Case).
- Change quotes to Smart, Dumb or French.
- Encode and decode HTML entities.

- Create an HTML list.
- 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. In the mean time, remember that you can apply scripted filters using an AppleScript or shell script, for example the shell script:

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

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

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:

- Activate Clipboard Copy Switcher
copies the current selection to a named clipboard you select (command-shift-C).
- Activate Clipboard Cut Switcher
cuts the current selection to a named clipboard you select (command-shift-X).
- Activate Clipboard Paste Switcher
pastes a named clipboard you select into the current system clipboard and current selection (command-shift-V).
- Activate Clipboard History Switcher
pastes a previous system clipboard you select from the clipboard history into the current system clipboard and current selection (command-control-shift-V).
- Activate Program Switcher
activates the [Program Switcher](#) (option-Tab).
- Activate Window Switcher
activates the Window Switcher (control-Tab).

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

You may want to change the Activate Program Switcher trigger to command-Tab to replace the system Program Switcher.

Record Quick Macro

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

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

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

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

For example, say you wanted to quote a dozen different words in a paragraph, you could click in the middle of the first word, press control-F1 to start quick recording, type option-left arrow, quote ("), option-right arrow, quote ("), and then control-F1 again to finish recording. Now click in the middle of each remaining word and press option-F1.

Google

The Google action displays a HUD window you can enter a search query in to and this fires off the query in your web browser.

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.

Growl

The Growl action lets you display a message using the [Growl](#) notification system. The message can include tokens so you can display the current clipboard or the date or various other pieces of information.

System Beep

Play a simple beep. This can be useful for debugging complex Macros or for alerting you when a long running macro is finished.

Alert

The Alert action lets you display a message as a system alert. The message can include tokens so you can display the current clipboard or the date or various other pieces of information. The alert includes two buttons, Continue and Stop. The macro continues when you click Continue, or stops if you click Stop.

Comment

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

Cancel All Macros

The Cancel All Macros action stops all macros currently executing.

Macro Triggers

- [Overview](#)
- [Hot Key](#)
- [Typed String](#)
- [Application](#)
- [Wake](#)
- [Login](#)
- [Time](#)
- [While Logged In](#)
- [Macro Palette](#)
- [Status Menu](#)

- [Public Web](#)
- [Device Key](#)
- [MIDI Note](#)

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, execute them remotely via the in-built web server, or at a particular time of day, as well as other possible events.

Hot Key

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

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

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

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

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

Typed String

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

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

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

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.

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.

Login

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

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

Time

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

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

While Logged In

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

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

Macro Palette

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

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

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

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

Status Menu

You can have a macro execute when you select it from the Keyboard Maestro Status Menu (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.

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

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

Public Web

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

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

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

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

Device Key

This is an advanced trigger—generally you should use a [Hot Key Trigger](#) if possible.

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

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

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

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

MIDI Note

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

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

Recording

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

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

Keyboard Maestro can record the following actions:

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

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

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

While Keyboard Maestro is recording, it will display the [Recording window](#).



You can stop all recording by clicking on the [Recording window](#).

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

The other way to use recording is via a Record Quick Macro action. When triggered, the Record Quick Macro immediately starts recording your actions into a private macro. When you have demonstrated the sequence of actions you want, trigger the Record Quick Macro again.

The sequence can now be executed via the specified [Hot Key](#) or the Status Menu or [Macro Palette](#). For example, if the Record Quick Macro is triggered by pressing control-F1, and the specified [Hot Key](#) is option-F1, then if you typed:

```
control-F1, h, e, l, l, o, control-F1
```

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

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

to

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

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

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

```
option-shift-right arrow, delete, forward delete, command-shift-right arrow, command-X, c, o, n, s, t, <space>, i, n, t, <space>, k,
command-V, =, ", command-V, ", ;
```

Finish with command-left arrow, down arrow to carefully put the cursor at the start of the next line. Now press control-F1 again to finish the recording, and option-F1 twice to translate the next two lines.

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

Macro 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. Also consider using Typed String triggers for these sorts of macros, for example "=em=" for email address and "=addr=" for address.

Use Hot Keys to Open Financial Accounts

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

Use Hot Keys to connect to SSH or FTP sites.

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

Simulate Bookmarks

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

Remap Command Keys

If you find yourself pressing a command key in an application and expecting it to do something but it does not work (for example, command-T for "Replace and Find Again"), use a Macro to make the command key "do the right thing" in that application. Similarly, if you use a function in an application frequently, but it has a convoluted command key or no command key at all, define your own command key

by using a [Hot Key](#) to select the menu item. Keep in mind that you can do some menu key remapping in the System Preferences Keyboard preference.

Simulate Missing Features

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

Swap Characters

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

- Simulate Keystroke shift-Right Arrow
- Cut to Named Clipboard "Temp"
- Simulate Keystroke Left Arrow
- Paste from Named Clipboard "Temp"

Save a Text Clipping

If you often want to save snippets of text, you could create a [Hot Key](#) to save a clipping:

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

Delayed Click

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

Insert Boilerplate Text

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

Apply Text Conversions

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

Simulate Workspaces

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

Setup an Application When Launched

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

Clean Up After Using an Application

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

Feedback During Macro Execution

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

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

Rakesh Kumar's PC Switcher's Pack

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

Download [Rakesh Kumar's PC Switcher Pack](#) and follow the instructions.

Program Switcher

The [Program Switcher](#) is essentially a highly specialized macro action that enables you to launch, switch, hide, and quit applications. By triggering the macro, the 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), as well as configure the icon size, color tint, and the sort order.

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

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

While the [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
- Press "d" to switch directly to the current app and hide other applications.



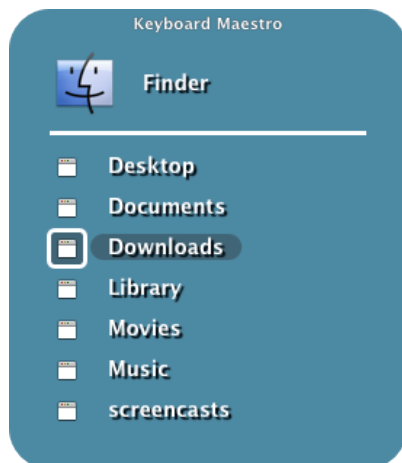
By default, Keyboard Maestro creates a [Program Switcher](#) macro in the "Switcher Group" [Macro Group](#), triggered by option-Tab. You can disable this Macro by selecting the Switcher Group, then selecting the [Program Switcher](#) macro and clicking the ✓ button below the Macros list. You can edit this macro to change the trigger to command-Tab to replace the system program switcher, or you can change it to any other desired [Hot Key](#).

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:

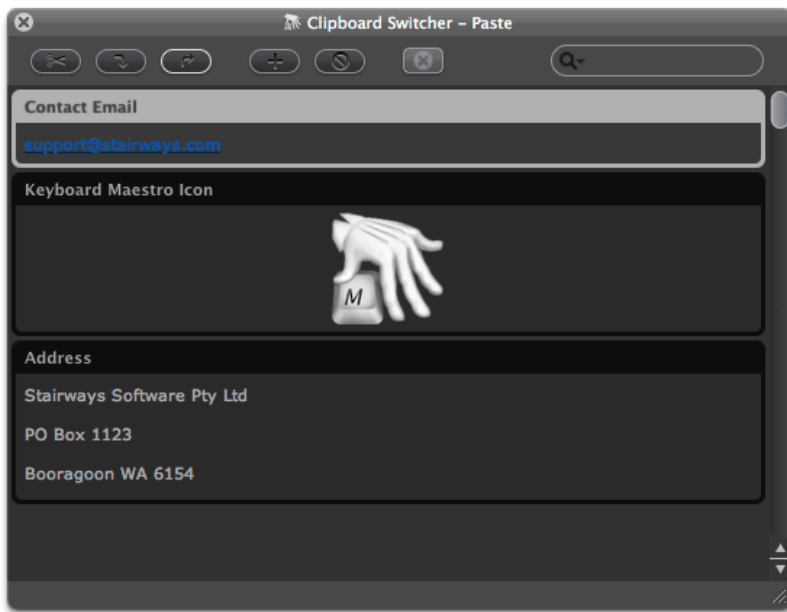
- Press "q" to mark (or unmark) a window to be closed
- Press "s" to mark (or unmark) a window to be minimize



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 the Switcher Group, then selecting the Window Switcher macro and clicking the ✓ button below the Macros list.

Clipboard Switcher

[Clipboard Switcher](#) enables you to define any number of named clipboards which can be used 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.



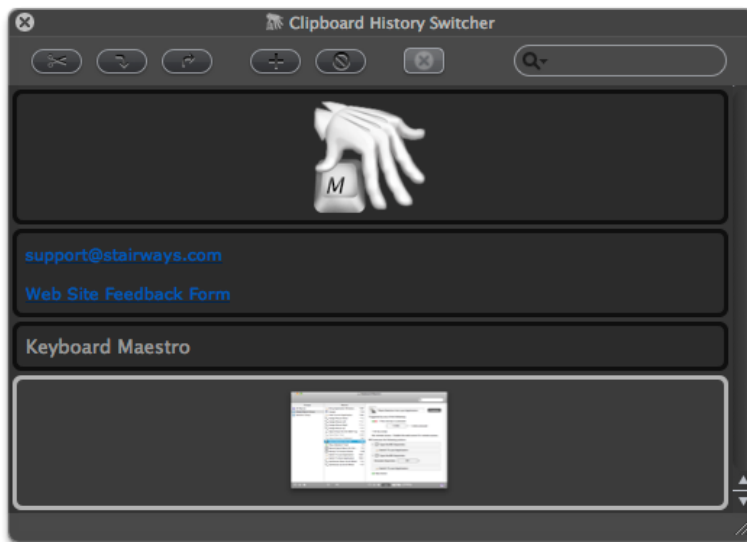
You can use type-ahead to select a named clipboard, and you can use the search field to filter the clipboards.

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

You can add or delete named clipboards in the [Clipboards preference pane](#).

Clipboard History Switcher

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



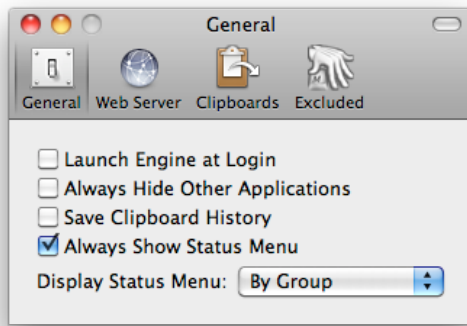
You can use the search field to filter the clipboards.

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

Keyboard Maestro also creates disabled macros for Paste Plain Text (command-shift-V) and Paste Previous Clipboard (command-control-V) in the "[Global Macro Group](#)" [Macro Group](#). You can enable these by selecting the Global Macros Group, then selecting the macros and clicking the ✓ button below the Macros list.

Preferences

To configure Keyboard Maestro, first launch Keyboard Maestro and choose [Preferences](#) from the [Keyboard Maestro menu](#).



The preferences are divided into sections.

General Preferences

In the [General preference pane](#) you can:

- enable or disable launching the [Keyboard Maestro Engine](#) at Login.
- configure whether to always hide other applications when switching applications.
- configure whether to save the clipboard history between logins.
- configure whether to always hide show the status menu icon.

Web Server Preferences

In the [Web Server preference pane](#) you can:

- enable or disable the web server.
- configure the username, password and port of the web server.
- enable or disable web browser access
- enable or disable iPhone access

The web server is disabled by default.

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

If the web server and web browsing 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.

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

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

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

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

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

Clipboards Preferences

In the [Clipboards preference pane](#) you can add, remove and rename [Named Clipboards](#).

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

Excluded Preferences

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

Other Hidden Preferences

You can configure various preferences using defaults write from the Mac OS X Terminal.

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

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

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

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

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

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

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

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

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

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

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

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

Scripting

Controlling Keyboard Maestro via Scripting

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

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

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

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

```
tell application "Keyboard Maestro Engine"
do script "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, you will get an error, so you can use a UID instead of a name.

```
tell application "Keyboard Maestro Engine"
do script "DOC150C7-8A0C-4837-918A-427E2BCFB6B9"
end tell
```

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

The easiest way to determine a macro's UID is to edit the macro and select from the "Or by script" trigger to display some script code which includes the UID.

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

```
tell application "Keyboard Maestro Engine"
do script "<dict><key>MacroActionType</key><string>SwitchToLastApplication</string></dict>"
end tell
```

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

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

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

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

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

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

Executing Scripts

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

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

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

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

Shell scripts are executed in the background and will be abandoned after 20 seconds, allowed to continue running to completion but otherwise ignored.

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

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

Windows

- [Welcome Window](#)
- [Macros Window](#)
- [Tutorial](#)
- [Macro Group Editor](#)
- [Macro Editor Window](#)
- [Recording Window](#)
- [Program Switcher Window](#)
- [Window Switcher Window](#)
- [Clipboard Switcher Window](#)
- [Clipboard History Switcher Window](#)
- [Preferences Window](#)
- [Preferences General Pane](#)
- [Preferences Web Server Pane](#)
- [Preferences Clipboards Pane](#)
- [Preferences Exclude Pane](#)
- [About Window Pane](#)

Welcome Window

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

Welcome to Keyboard Maestro



Welcome to Keyboard Maestro 4

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

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

[Start Tutorial](#)

Show this window when Keyboard Maestro opens

Quick Start
Learn about the components of Keyboard Maestro and how they work together.
[Learn more >](#)

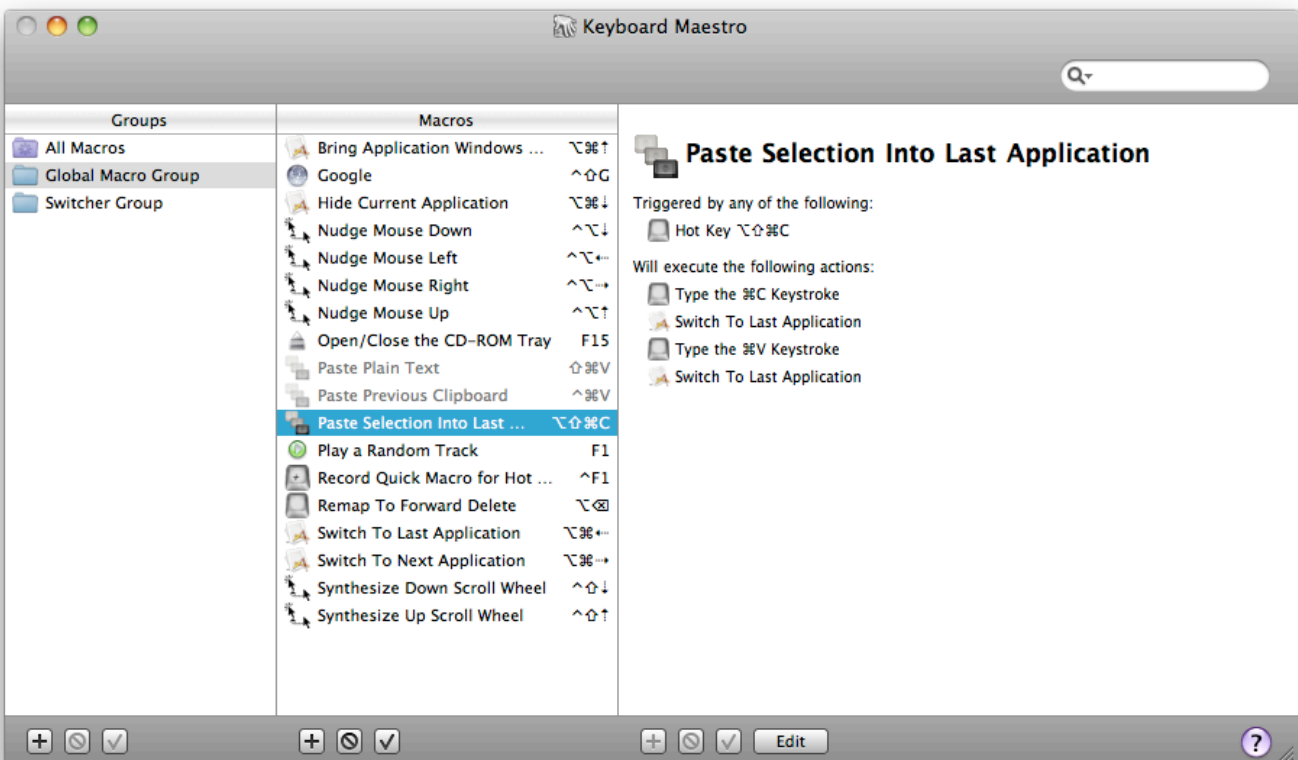
Assistance
Learn about the user group, documentation, videos, and customer support options.
[Learn more >](#)

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

Macros Window

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

You get this window pane by launching Keyboard Maestro.



Groups	Macros	
All Macros	Bring Application Windows ...	⌘⇧↑
Global Macro Group	Google	⌘⇧G
Switcher Group	Hide Current Application	⌘⇧↓
	Nudge Mouse Down	⌘⇧↓
	Nudge Mouse Left	⌘⇧←
	Nudge Mouse Right	⌘⇧→
	Nudge Mouse Up	⌘⇧↑
	Open/Close the CD-ROM Tray	F15
	Paste Plain Text	⇧⌘V
	Paste Previous Clipboard	⇧⌘V
	Paste Selection Into Last ...	⌘⇧⌘C
	Play a Random Track	F1
	Record Quick Macro for Hot ...	⇧F1
	Remap To Forward Delete	⌘⇧
	Switch To Last Application	⌘⇧←
	Switch To Next Application	⌘⇧→
	Synthesize Down Scroll Wheel	⇧⇧↓
	Synthesize Up Scroll Wheel	⇧⇧↑

Paste Selection Into Last Application

Triggered by any of the following:

- Hot Key ⌘⇧⌘C

Will execute the following actions:

- Type the ⌘C Keystroke
- Switch To Last Application
- Type the ⌘V Keystroke
- Switch To Last Application

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

You can create a new [Macro Group](#) by clicking the + button below the Groups list.

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

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

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

You can rename a [Macro Group](#) or Macro by double clicking it and changing the title.

You can delete a [Macro Group](#) by selecting it and clicking the ⓧ button below the Groups list.

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

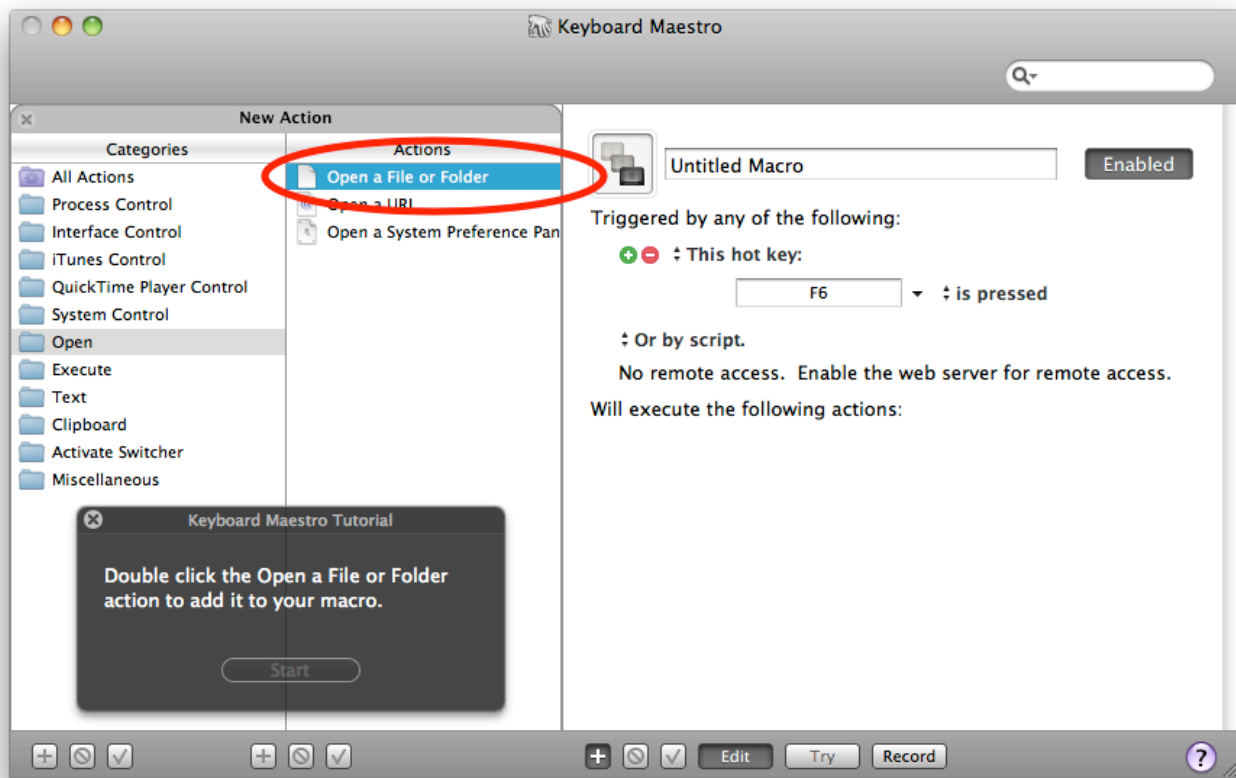
Similarly, you can enable or disable [Macro Groups](#) or Macros by clicking their respective ✓ button.

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

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

Tutorial

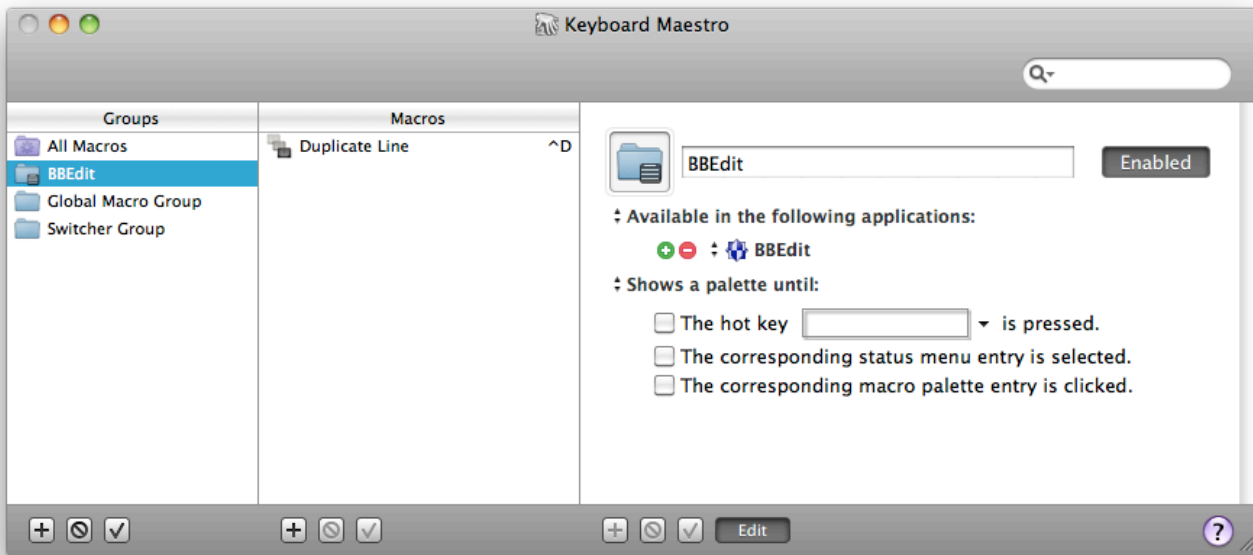
If you are new to Keyboard Maestro, start the tutorial by clicking the [Start Tutorial](#) button in the [Welcome window](#) or by choosing [Tutorial](#) from the [Help menu](#) and Keyboard Maestro will walk you through creating a simple macro.



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

Macro Group Editor

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



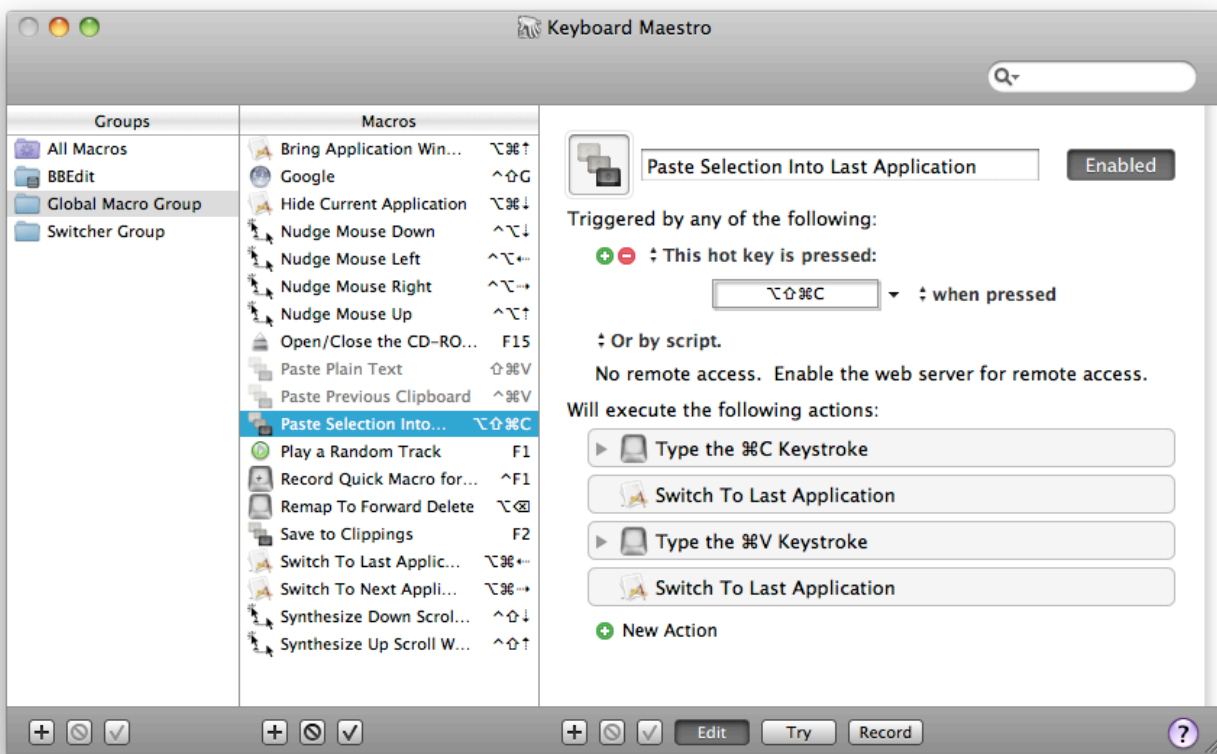
Typically a group's macros would be available everywhere (available in all applications), or it might be specific to a particular application (available in the following applications) in which case you might name the Macro Group after the application.

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

See also the [Macros](#) section.

Macro Editor Window

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



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

To see how to execute this macro via a script, select from the "Or by script" menu.

How you can execute the macro remotely is also displayed.

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

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

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

You can click the **Record** button to record your actions.

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

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.

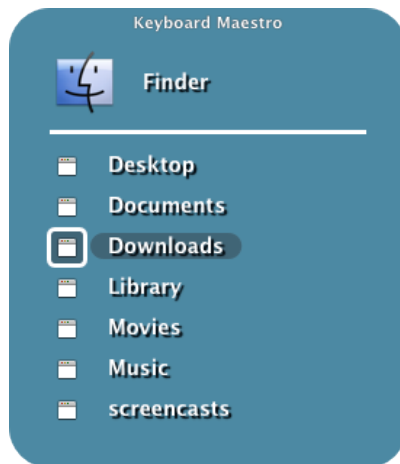


To learn more about the [Program Switcher](#), see the [Program Switcher](#) section.

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.

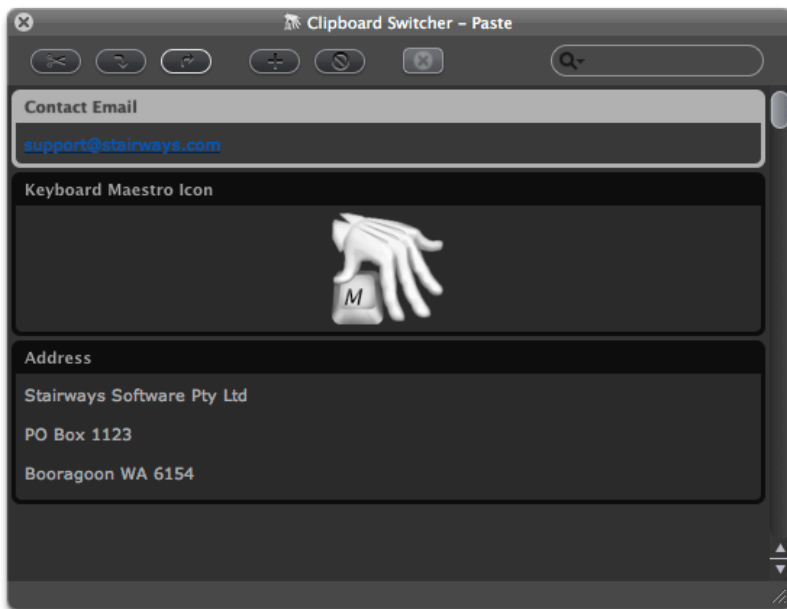


To learn more about the [Window Switcher](#), see the [Window Switcher](#) section.

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.



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

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

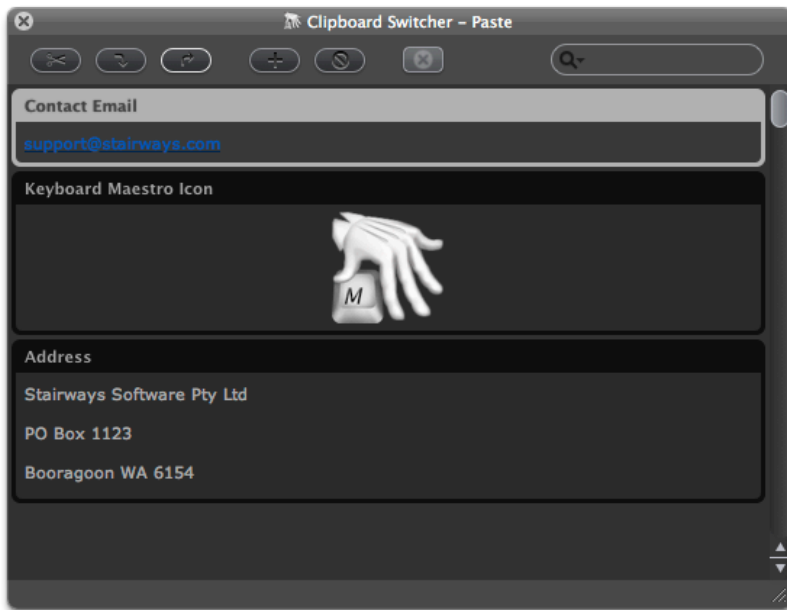
Use the search field to filter the named clipboards.

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

Clipboard History Switcher Window

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

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



Select something cut or copy and it will appear in this list. Select an item and click the **Paste** button to paste a named clipboard. Click the **+** button to copy the current selection. Select an item and click the **⊘** button to delete it (this is useful if you want to delete a password or other sensitive information).

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

Use the search field to filter the named clipboards.

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

Preferences Window

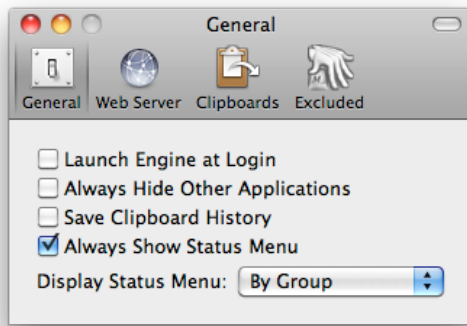
This window lets you configure Keyboard Maestro.

You get this window by launching Keyboard Maestro and choosing [Preferences](#) from the [Keyboard Maestro menu](#).

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

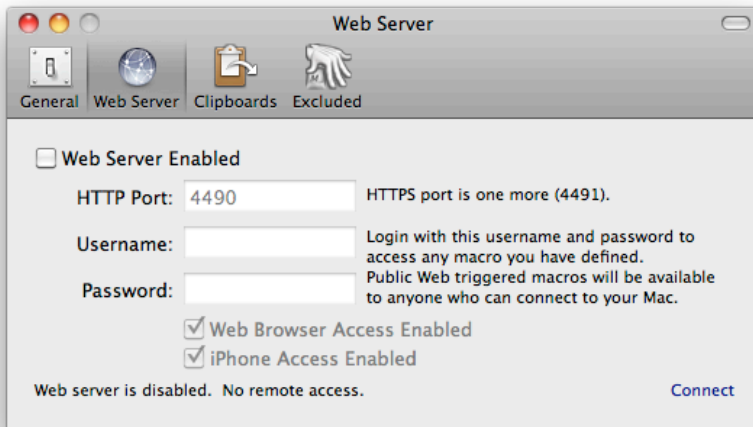
Preferences General Pane

This window pane lets you configure general preferences.



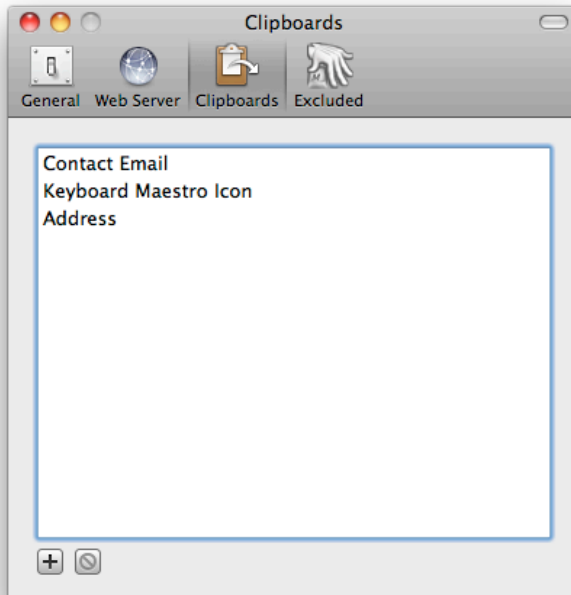
Preferences Web Server Pane

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



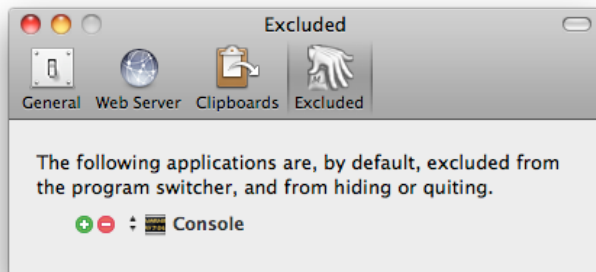
Preferences Clipboards Pane

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



Preferences Exclude Pane

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



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

Edit

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

View

- [Sort by Macro Name](#)
- [Sort by Macro Trigger](#)

- [Edit](#)
- [Show Actions](#)
- [Record](#)

[Window](#)

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

[Help](#)

- [Keyboard Maestro Help](#)
- [Keyboard Maestro Quick Start](#)
- [Welcome to Keyboard Maestro](#)
- [Tutorial](#)
- [Videos](#)
- [Logs](#)
- [Third Party Licenses](#)
- [Online Documentation](#)
- [Keyboard Maestro Web Site](#)
- [Stairways Software Web Site](#)
- [Service and Support](#)

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

The [Close command](#) in the [File menu](#) closes the front window.

File » Export Macros

The [Export Macros command](#) in the [File menu](#) exports the selected macros to a file that you can 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 » 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.

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.

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 » Delete

The [Delete command](#) in the [Edit menu](#) deletes the current selection.

Edit » Select All

The [Select All command](#) in the [Edit menu](#) selects all text or items.

Edit » Duplicate

The [Duplicate command](#) in the [Edit menu](#) selects all text or items.

View

The [View menu](#) contains menu items relating to display and actions.

View » Sort by Macro Name

The [Sort by Macro Name command](#) in the [View menu](#) sorts the macros in the main window by name.

View » Sort by Macro Trigger

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

View » Edit

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

View » Show Actions

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

View » Record

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

Window

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

Window » Minimize

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

Window » Zoom

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

Window » 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](#) (sorry) and displays the Keyboard Maestro local

documentation.

Help » Keyboard Maestro Quick Start

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

Help » Welcome to Keyboard Maestro

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

Help » Tutorial

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

Help » Videos

The [Videos command](#) in the [Help menu](#) displays the videos that can help you learn to use Keyboard Maestro.

Help » Logs

The [Logs command](#) in the [Help menu](#) displays the Keyboard Maestro Logs folder.

Help » Third Party Licenses

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

Help » Online Documentation

The [Online Documentation command](#) in the [Help menu](#) takes you to the Keyboard Maestro web site and displays the documentation.

Help » Keyboard Maestro Web Site

The [Keyboard Maestro Web Site command](#) in the [Help menu](#) takes you to the Keyboard Maestro web site.

Help » Stairways Software Web Site

The [Stairways Software Web Site command](#) in the [Help menu](#) takes you to the Stairways Software web site.

Help » Service and Support

The [Service and Support command](#) in the [Help menu](#) displays the service and support details.

Tips

- [Remembering Macro Hot Keys](#)
- [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.

Keyboard Maestro also interoperates with [KeyCue](#) - if you use both programs and hold the control key down KeyCue will display all your active [Hot Keys](#).

Use Function Keys for Global Hot Keys

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

Use the Number Pad

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

Troubleshooting

- [Macros/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?](#)
- [AppleScript's display dialog does not work, why?](#)
- [I have an X-Key keyboard, are there any issues?](#)
- [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 [General preference pane](#).

My Macros are not working, what's wrong?

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

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

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

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

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

The 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?

By default, Keyboard Maestro uses option-Tab for its Program Switcher to avoid overriding the system Program Switcher, but if you wish to use command-Tab you can do so.

Select the Switchers Group, then select the Program Switcher macro. Click the **Edit** button to switch in to Edit mode if necessary. Click the [Hot Key](#) field and press command-Tab.

AppleScript's display dialog does not work, why?

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

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

See also the [Scripting](#) section.

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

In general, Keyboard Maestro's Device trigger can work with all the [P.I. Engineering's X-Key](#) or other programmable keyboards.

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

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

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

How do I get more help?

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

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

Support

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

You can join the [Keyboard Maestro User Group](#) online community consisting of the developers and Keyboard Maestro users.

For ideas, see the [Macro Examples](#) section.

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

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

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

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

Glossary

Clipboard

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

Clipboard History

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

Clipboard Switcher

is a feature of Keyboard Maestro that allows you to copy or paste to/from a set of [Named Clipboards](#).

Excluded Applications

is the set of applications that should not appear in the [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.

Growl

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

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.

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

Keyboard Maestro Engine
The process that enables your [Macros](#), [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 4 requires [Mac OS X](#) 10.5 or later. 10.6 or later recommended.

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. Keyboard Maestro 3 was released in April 2008 followed by a succession of releases over the rest of 2008.

Development of Keyboard Maestro 4 began in earnest in late 2008 and was released in late 2009. Version 4 was a complete rewrite of the user interface, bringing with it a modern look and feel reminiscent of various modern Apple applications.

Going forward, we plan to develop Keyboard Maestro aggressively, bringing it to new levels of both power and ease of use in the long tradition of both [Mac OS](#) and Stairways Software.

Credits

Thanks to [Michael Kamprath](#) for all his work producing Keyboard Maestro.

Thanks to Alan Gentle for many example Macro ideas.

Thanks to Ken G. Brown for some great beta testing.

Thanks to Rakesh Kumar for the set of Switcher Macros.

Thanks to Sam Stephenson and the Prototype Core Team for the Prototype JavaScript Framework.

Thanks also to:

- [Eric Rescorla](#) for wserver
- [Jerry Krinock](#) for NS(Attributed)String+Geometrics
- John Engelhart for [RegexKitLite](#)
- [John Gruber](#), [Aristotle Pagaltzis](#) for Title Case
- [Matt Gemmell](#) - Magic Aubergine for MGAnimatedView and MGTemplateEngine
- [Michael Ash](#) for MAKVONotificationCenter
- [Rainer Brockhoff](#) for RBSplitView

Thanks to Corentin, Philippe, Stephen and others for their great assistance with beta testing.

Thanks to Andy for great help editing this documentation.

Thanks also to the many others who have provided input and support over the past decade.

Warranty

This 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 2009 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.

Keyboard Maestro is licensed on a per-Mac basis. You must purchase a license for each Mac you use it on. We offer steep volume discounts up to 65% off.

Individuals purchasing for non-commercial purposes may license it on a per-User basis.

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.