

KeyQuencerTM

Version 2.5 Online User Guide

User Guide written by Gary González

Programming by Alessanadro Levi Montalcini

Visit our website: <http://www.binarysoft.com>

Send us email: binary@binarysoft.com

Binary Software, Incorporated
2118 Wilshire Boulevard, Suite 900
Santa Monica, California, 90403
United States of America

© 1997 by Binary Software, Inc.

Protected under the Berne Convention. All rights reserved.

Version 2.0 1996, Version 2.1 1997, Version 2.5 1997

Printed in the United States of America

No part of this publication may be reproduced, transmitted, transcribed, stored in a retrieval system, or translated into any other language or computer language in whole or in part, in any form or by any means, whether it be electronic, mechanical, magnetic, optical, manual or otherwise without the prior consent of Binary Software, Inc..

Except for the limited warranty set forth in the Binary Software License Agreement, the user documentation is provided "as is."

Unauthorized use of the documentation or the accompanying software can result in civil damages or criminal prosecution. Please refer to the Binary Software License Agreement for additional information.

Square One®, Binary Software®, KeyQuencer™, the Binary Software logo, and the phrase "Because it either is Binary Software, or it isn't"™ are trademarks or registered trademarks of Binary Software, Inc..

Power Macintosh™ and Macintosh® are trademarks or registered trademarks of Apple Computer, Inc..

Trademarks of other companies, and other trademarks mentioned in this documentation appear for identification purposes only and are trademarks or registered trademarks of their respective holders.

Contents

Credits	7
1: Welcome!	8
What Is KeyQuencer?	8
What Is KeyQuencer Lite?	8
How Is the Lite Version Different from the Full Version?	9
What You Need to Begin	9
Registration and Support	9
I Need Help!	9
Overview of This Guide	10
Conventions Used	10
2: Installation	11
How Do I Install KeyQuencer?	11
Upgrading from Version 2.0 or 2.1	11
Upgrading from Version 1.2	11
Installing KeyQuencer 2.5	11
What Does KeyQuencer Install?	11
3: Getting Started	13
What Is a Macro?	13
How Do I Execute a Macro?	13
How Do I Execute a Macro with the QuickMacro List?	13
How Do I Execute a Macro with the KeyQuencer Contextual Menu?	14
How Do I Execute a Macro with the KeyQuencer Control Strip Module?	14
How Do I Execute a Macro with the KeyQuencer Launcher?	15
How Do I Control Macro Execution?	15
What Is the Activity Monitor?	15
How Do I Turn KeyQuencer Off?	15
How Do I Stop a Macro While It's Executing?	15
4: Making Simple Macros	16
What Is a Macro Name?	16
What Is a Keystroke?	16
What Is the KeyQuencer Editor?	16
How Do I Open the KeyQuencer Editor?	16
What About the KeyQuencer Panel?	17
What Is a Text Macro?	17
How Do I Automatically Make a Text Macro?	17
How Do I Manually Make a Text Macro?	18

What Is an Open Macro?	18
How Do I Automatically Make an Open Macro?	18
How Do I Use the Contextual Menu to Make an Open Macro?	19
How Do I Manually Make an Open Macro?	20
What Is a Menu Macro?	20
How Do I Automatically Make a Menu Macro?	20
How Do I Manually Make a Menu Macro?	21
What Is a Button Macro?	22
How Do I Automatically Make a Button Macro?	22
How Do I Manually Make a Button Macro for a Named Button?	23
How Do I Manually Make a Button Macro for a Nameless Button?	23
What Is a Click Macro?	25
How Do I Automatically Make a Click Macro?	25
How Do I Make a Click Macro in a Dialog?	25
How Do I Manually Make a Click Macro?	26
What Is a Drag Macro?	27
How Do I Automatically Make a Drag Macro?	27
How Do I Make a Drag Macro in a Dialog?	28
How Do I Manually Make a Drag Macro?	28
What Is a Key Macro?	30
How Do I Automatically Make a Key Macro?	30
How Do I Manually Make a Key Macro?	30
 5: Using the KeyQuencer Editor	 32
Running, Creating, Changing and Deleting Macros	32
How Do I Run a Macro?	32
How Do I Make a New Macro?	32
How Do I Change a Macro's Name?	33
How Do I Change a Macro's Keystroke?	33
How Do I Delete a Macro?	33
What Is the Commands Palette?	34
How Do I Show the Commands Palette?	34
How Do I View a Command's Parameters?	34
How Do I Insert a Command or Parameter into a Macro?	34
How Do I Insert a File's Pathname into a Macro?	34
How Do I Insert a File's Creator Type into a Macro?	34
How Do I Change the Font of the Commands Palette?	34
How Do I Change the Font Size of the Commands Palette?	34
How Do I Change the Layout of the Commands Palette?	35
What is the Command Help Window?	35
How Do I Show the Command Help Window?	35
How Do I See Help on the KeyQuencer Macro Language?	35
What Is a Macro Suitcase?	35

How Do I Open the Installed Macros Suitcase?	36
How Do I Make an Editor Suitcase?	36
How Do I Make a Launcher Suitcase?	36
How Do I Make a Stand-Alone Applet?	36
How Do I Copy a Macro from One Suitcase to Another?	37
How Do I Change a Suitcase's Type?	37
How Do I Merge Two Suitcases?	37
How Do I Backup or Replace the Installed Macros Suitcase?	37
How Do I Customize the Editor?	38
How Do I Change the Font Face in the Editor?	38
How Do I Change the Font Size in the Editor?	38
How Do Turn On or Off Text Wrapping?	38
How Do I Change the Sort Order of My Macros?	38
How Do I Change the Layout of the Editor Window?	38
How Do I Automatically Save Suitcases When I Close Them?	38
How Do I Automatically Save Changes Before Running Macros?	38
Editor Shortcuts	39

6: Writing Macros 40

What Is a Parameter?	40
What Is a Pathname?	41
What Does "Debug" Mean?	41
How Do I Automate Popup Menus?	41
How Do I Find Out a Popup Menu's Index?	42
What Is a Macro Comment?	42
How Do I Add Comments to a Macro?	42
What Is a Variable?	43
What Is an Escape Sequence?	44
How Do I Debug a Macro?	44
Debugging Tips	44
Where to Go from Here	44

7: Advanced Features 45

What Is a KeyQuencer Extension?	45
Where Are the KeyQuencer Extensions?	45
How Do I Add a New KeyQuencer Extension?	45
How Do I Disable a KeyQuencer Extension?	45
Using KeyQuencer with AppleScript	45
What AppleScript Commands Does KeyQuencer Support?	45
How Do I Run a Macro from AppleScript?	46
How Do I Run an Active Macro from AppleScript?	46
How Do I Control the Flow of Execution with AppleScript?	46
Using KeyQuencer in Other Environments	46

KeyQuencer Batcher	46
How Do I Configure the Macros in My Batcher?	47
How Do I Set My Batchers Preferences?	47
How Do I Choose Files to Batch?	48
8: Language Reference	49
What Is a Macro Command Line?	49
How Do I Interpret a Command's Syntax?	49
What Are the Different Types of Parameters?	49
Comments	52
KeyQuencer Command Descriptions	52
How Do I Create a Printable Text File of All KeyQuencer Commands?	52
Appendix A: KeyQuencer License	53
Appendix B: KeyQuencer Lite License	54
Index	56

Credits

KeyQuencer original design, concept and programming by Alessandro Levi Montalcini.

KeyQuencer version 2.5 programming, design and concept by Alessandro Levi Montalcini.

KeyQuencer Installer was created by Gary González using the StuffIt Installer.

KeyQuencer version 2.5 User Guide was written by Gary González using Adobe PageMaker 6.5.

Thanks to all our beta testers, especially Mark Nagata, the best beta-tester in this corner of the galaxy.

Extra special thanks to Wendy Frawley, Dolores González, Thomas Green, Charles Haspel, Deborah Haspel, Lawrence Haspel, Eleanor Haspel-Portner, Nick Lush, Alessandro Levi Montalcini, Nancy Podosin, Robert Podosin, Adam Portner, Marvin Portner, Leonard Rosenthol, jud spencer, Melissa Spencer, Eric Stern, Steven Trilling, Bruce Tyson, and Shelley Tyson.

And, of course, thank you, our users. Your suggestions and support made version 2.5 possible.

1: Welcome!

Thank you for choosing KeyQuencer. We promise that after using KeyQuencer for a couple of weeks you'll wonder how you ever lived without it.

What Is KeyQuencer?

KeyQuencer is a utility that lets you automate any — yes, **any** — repetitive task. It is a *macro* utility. A macro is a set of instructions. You can use KeyQuencer macros to:

- Launch or quit programs, open documents or folders, or switch applications.
- Type text or keystrokes
- Click or drag the mouse
- Press buttons
- Choose from menus.
- Navigate scroll bars, move or select windows, or wait for specific events.
- Copy, move, rename, or delete files or folders.
- Jump directly to any folder or disk from within an **Open** or **Save** file dialog.
- Copy and paste text from up to ten clipboards.
- Compress, decompress or BinHex encode files and folders with the StuffIt Engine.
- Switch printers, mount file servers, unmount volumes, or eject removable disks.
- Change your monitor bit depth or resolution, or your speaker volume.
- Shut Down, Restart, Empty the Trash, or toggle Balloon Help.
- Play or pause an Audio CD.
- Control one or more computers across a network, synchronize the clocks or clipboards.
- Open or close a PPP connection.
- And much more!

What Is KeyQuencer Lite?

KeyQuencer Lite is the shareware version of KeyQuencer. The most important thing about KeyQuencer Lite is...

.....
KeyQuencer Lite is Not Free! If you use and like KeyQuencer Lite, please take the time to show your support and pay the registration fee. Information about how to register can be found in the KeyQuencer folder.
.....

How Is the Lite Version Different from the Full Version?

1. It doesn't include the KeyQuencer Launcher and the KeyQuencer Batchter.
2. It is limited to 50 macros and 50 commands.
3. It has about half the number of commands that the full version has.
4. It cannot create Stand-Alone Applets.
5. It cannot use variables.

If you are using KeyQuencer Lite, you can still use this User Guide. Except for the above, everything in this User Guide is applicable to the Lite version.

What You Need to Begin

- A Macintosh running Mac OS 7.0 or higher is required. Mac OS 7.5 or later recommended.
- A hard disk-drive.
- 4 or more megabytes of RAM.

Registration and Support

If you purchased KeyQuencer directly from Binary Software, then you are automatically registered. Otherwise, please complete the enclosed registration card and mail it to us. Or, if you prefer, call us, fax us, or email us. When you register KeyQuencer you get:

- Unlimited, free, and friendly technical support.
- Free bug-fixes.
- Special offers on new products.

Please register: it's the only way we can keep you up to date.

I Need Help!

We tried our best to make this User Guide as clear and nontechnical as possible. We aren't perfect though, and you may still have some unanswered questions.

If you have any questions at all about KeyQuencer, please check our web site, give us a call, or drop us a line. We would love hear from you:

- Web: www.binarysoft.com
- Phone: (310) 449-1481, Monday through Friday, 10:00 AM to 5:00 PM Pacific Time
- Fax: (310) 449-1473
- Email: support@binarysoft.com
- CompuServe email: 74220,3377
- America Online email: BINARY SW

Overview of This Guide

Chapter 1: Welcome explains what KeyQuencer is, provides an overview of this User Guide, tells how to register your software, and how to get help.

Chapter 2: Installation explains how to install KeyQuencer and what each one of KeyQuencer's files does.

Chapter 3: Getting Started explains all the different ways to execute macros so you can get started fast using KeyQuencer's standard macros

Chapter 4: Making Simple Macros explains step by step how to create macros to type text, open a file, choose a menu command, press a button, or click or drag the mouse.

Chapter 5: Using the KeyQuencer Editor explains how to use the Commands Palette and the Command Help window, how to work with multiple suitcases, and how to customize the Editor to meet your needs.


Chapter 6: Writing Macros explains how to write and debug complex macros.

Chapter 7: Advanced Features tells you how to monitor and control macro execution, how to Manage KeyQuencer Extensions, how to use KeyQuencer with AppleScript, how to control computers across a network, and how to use the Keyquencer Batchter.

Chapter 8: Language Reference is a quick reference guide to KeyQuencer's macro language.

Appendix A: KeyQuencer License and *Appendix B: KeyQuencer Lite License* describe your license rights and usage limitations.

Conventions Used

- **Bold** text refers to **buttons**, **menus**, **checkboxes** and **menu items**.
- *Italicized text* refers to *folder names*, *window names*, *dialog names*, *disk-drive names*, and *file names*. It's also used for *new terms*, *chapter or section references*, and *important ideas*.
- Hierarchical submenus are noted as **Submenu: Submenu Item**. For example, the **by Name...** item of the **Arrange** submenu of the **View** menu would be described in this guide as "Choose **Arrange: by Name...** from the **View** menu."
- "Command" followed by a letter means hold down the Command key while pressing the letter. For example, "Command-V" means hold down the Command key while pressing the "V" key. The Command key has a cloverleaf symbol on it that looks like this: .
- "Shift-click" means hold down the Shift key while clicking the mouse button, "Option-click" means hold down the Option key while clicking the mouse button. The same is true for "Control-click" and "Command-click."
- Warnings, notes, and tips have a rule above and below them:

.....
Always send in your registration cards.
.....

- Sample macro or AppleScript text appears in a different font. For example:

Menu "by Name" submenu

2: Installation

How Do I Install KeyQuencer?

Upgrading from Version 2.0 or 2.1

- Be sure to move all your third party KeyQuencer commands out of the KeyQuencer Extensions before you run the Installer. Otherwise, the Installer will delete them.

Upgrading from Version 1.2

Follow these instructions to save your macros. **Quit the Installer and do this now before proceeding or you will lose your macros!**

1. Open the *Preferences* folder (inside your *System Folder*) and create a new folder there titled *KeyQuencer Preferences*.
2. Open the KeyQuencer control panel.
3. Open the **Edit** menu of the KeyQuencer control panel and choose **Select All**.
4. Open the **File** menu of the KeyQuencer control panel and choose **Export Selected....**
5. In the export dialog that appears, name the exported file “KeyQuencer Macros” and save it in the *KeyQuencer Preferences* folder that you created in step one above.
6. Find and remove the old *Macro Editor FAT* application since the Installer doesn’t take care of it.
7. Now run the Installer as usual. All your macros will be preserved.

Installing KeyQuencer 2.5

1. Disable any virus protection or disk security software before you begin.
2. Double-click the *Installer* icon.
3. Click **Install**.
4. The KeyQuencer Installer will copy all necessary files to the appropriate locations. If you have an earlier version of KeyQuencer, the Installer will update all your old files. After installation is complete, you will see a dialog box with a **Restart** button. Click **Restart**.
5. Enable any virus protection or disk security software that was previously disabled.

That’s it! You are ready to begin using KeyQuencer. You may now skip ahead to *Chapter 3: Getting Started*.



















What Does KeyQuencer Install?



KeyQuencer folder contains all of KeyQuencer’s applications and support files. By default, it is placed on the root level of your startup disk. Any outdated files from earlier versions of KeyQuencer are moved to an *Outdated Files* folder inside the *KeyQuencer* folder and can be removed once you’ve verified that the new setup is functioning.



KeyQuencer Editor is used to create and edit macros.

-  *KeyQuencer Launcher* displays a window with a double-clickable list of your macros (not included in KeyQuencer Lite).
-  *KeyQuencer Batchers* applies a KeyQuencer macro individually and sequentially to each file in a group of files (not included in KeyQuencer Lite).
-  *KeyQuencer 2.5 Read Me* contains the latest information about KeyQuencer.
-  *Macros* is a folder that contains more than one-hundred sample macros, AppleScripts, and sample Batchers. It also contains a macro suitcase with the Standard Macros.
-  *Standard Macros* is a macro suitcase that contains all the standard macros or default macros. Users upgrading from a previous version of KeyQuencer can use this to get access to the standard macros referred to in this User Guide.
-  *Goodies* contains some miscellaneous KeyQuencer components, like Sounds Like KeyQuencer, KeyQuencer Recover, and SuitcaseToText. Documentation for these items can be found in the accompanying *Read Me* files.
-  *Links* contains links that let you use KeyQuencer with third-party programs like Frontier, FileMaker, 4D, Cron and HyperCard. Documentation for these items can be found in the accompanying *Read Me* files.
-  *Documentation* contains this User Guide in Acrobat and/or eDoc format. If you are installing from a floppy disk that does not include the Acrobat version, it can be downloaded from our website at **www.binarysoft.com**.
-  *KeyQuencer Engine* is the system extension that enables all of KeyQuencer's commands. The Engine is vital to KeyQuencer and if it is not loaded, KeyQuencer will not work.
-  *KeyQuencer Panel* can be used to edit and create macros. It is only installed if a previous version of KeyQuencer is detected. Although it is a control panel, it does not load at startup time and is not necessary to use KeyQuencer.
-  *KeyQuencer* folder contains some helpers and macros that help you to make simple macros. It is installed in the *Apple Menu Items* folder.
-  *KeyQuencer CMM* provides support for KeyQuencer's context menu. It is installed in the *Contextual Menu Items* folder in Mac OS 8.
-  *KeyQuencer Strip* is KeyQuencer's control strip item. It is installed in the *Control Strip Modules* folder.
-  *KeyQuencer Links* provides KeyQuencer's AppleScript support. It is installed in the *Scripting Additions* folder.
-  *KeyQuencer Extensions* is a folder in your *System Folder* that contains all your KeyQuencer extensions. Each command in KeyQuencer's macro language corresponds to an extension in the *KeyQuencer Extensions* folder.
-  *KeyQuencer Helpers* is a folder that contains Stand-Alone Applets that help you make KeyQuencer macros. An alias to this folder is placed inside the *Apple Menu Items* folder.
-  *KeyQuencer Macros* is the macro suitcase that contains all your active macros. It is located in the *KeyQuencer Preferences* folder in the *Preferences* folder.
-  *Developer Toolkit* contains information for programmers interested in writing KeyQuencer extensions. The Developer Toolkit is not installed by default. To install it, go to the Custom Install page of the Installer.

3: Getting Started

This chapter explains how to begin using KeyQuencer right away. By the end of this chapter you will be able to use all of KeyQuencer's standard macros. You will probably find many of the standard macros are time savers in and of themselves!

What Is a Macro?

A *macro* is a set of instructions that describes a series of tasks for your computer to perform. So that you can get started right away, we created and installed a set of standard, ready-to-use macros.

How Do I Execute a Macro?

The easiest way to execute a macro is to type it's assigned keystroke. For example, one of the standard macros included with KeyQuencer is *Empty Trash*. You can execute this macro (and empty your Trash) by pressing its assigned keystroke

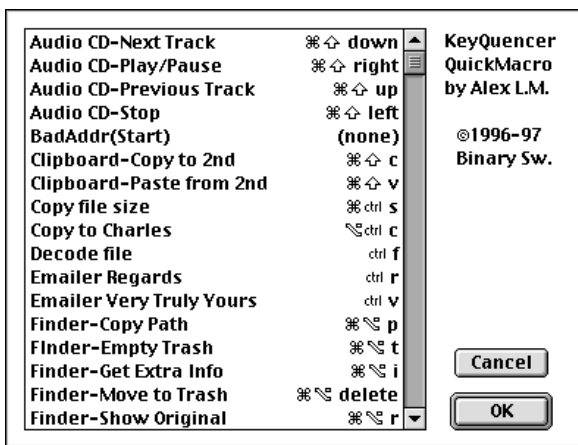
- Press Option-Command-T

Since you may not always remember a macro's keystroke, KeyQuencer gives you other ways to execute macros.

How Do I Execute a Macro with the QuickMacro List?

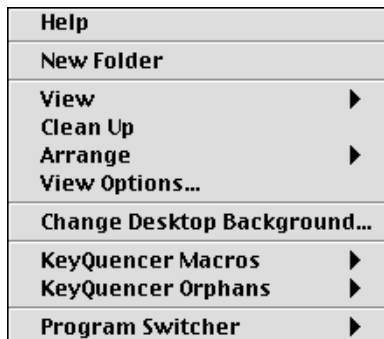
You can use the QuickMacro List any time to execute a macro or see its keystroke.

1. Press Command-Control-; or open the **Apple** menu and choose **KeyQuencer Helpers: QuickMacro List**.
2. Click the macro in the list that you want to execute and press **OK** (or just double-click the macro).



How Do I Execute a Macro with the KeyQuencer Contextual Menu?

1. Control-click in any Mac OS 8 savvy program to see the context menu.
2. Choose the macro you want to execute.
 - *KeyQuencer Macros* lists all active macros.
 - *KeyQuencer Orphans* lists only macros with no assigned keystroke (orphans).
 - *KeyQuencer Actions* appears when you control-click on a file in the Finder. It lists only those macros that act on the current Finder selection.



Regular Contextual Menu



Contextual Menu of Finder Selection

How Do I Execute a Macro with the KeyQuencer Control Strip Module?

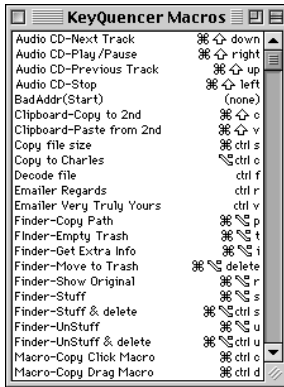
1. Click-hold on the KeyQuencer Strip module in your Control Strip.



2. Choose the macro you want to execute.
 - Control-click-hold to see assigned keystrokes.
 - Command-click to see only macros with no assigned keystroke (orphans).

How Do I Execute a Macro with the KeyQuencer Launcher?

1. Open the **Apple** menu and choose **KeyQuencer Helpers: Open Launcher** or open the *KeyQuencer* folder and double-click *KeyQuencer Launcher*.
2. Double-click any macro in the list to execute it.



How Do I Control Macro Execution?

This section tells you how to use the Activity Monitor, turn KeyQuencer on and off, and how to stop a macro while it's executing.

What Is the Activity Monitor?


When a KeyQuencer macro is executing you will see a small, quick flash in the upper left corner of your screen, near the **Apple** menu.



That's the KeyQuencer activity monitor. The activity monitor tells you when KeyQuencer is busy.

How Do I Turn KeyQuencer Off?

When KeyQuencer is off, KeyQuencer ignores your keystrokes, even if they are assigned to active macros (you can still run macros from the KeyQuencer Editor or the KeyQuencer Launcher). To turn KeyQuencer on or off:

- Press Command-Shift-Option-Return. This will toggle KeyQuencer on and off. The activity monitor  next to the **Apple** menu flashes once when you turn KeyQuencer off, and many times when you turn it on again.

How Do I Stop a Macro While It's Executing?

Sometimes KeyQuencer can get stuck while executing a macro. To stop any macro in any program at any time...

- Press Command-Shift-Return.
- To stop macro execution without displaying a confirmation dialog, press Command-Shift-Control-Return.

4: Making Simple Macros

This chapter explains how to make some simple macros of your own. By the end of this chapter you will be able to make macros to type some text, open any file, choose a menu command, press a button, or click or drag the mouse.

KeyQuencer macros are text based. This means that (for the most part) to make a macro you must write it yourself using KeyQuencer’s macro language. There is no record or “watch-me” feature. Since making text-based macros can be a little difficult at first, we created some macros to assist you in the process.

What Is a Macro Name?

A macro name is the name of a macro. It is used to identify the macro. As you make your macros, you will have to give them names. Try to name the macro something meaningful, so that you will know what it does just by looking at its name. For example, you might name a macro that types your name “Type name.” Or you might name a macro that launches Navigator “Open Navigator.”

What Is a Keystroke?

A keystroke is the keyboard shortcut that you assign to a macro. Each macro has a unique keystroke assigned to it. Whenever you press a keystroke that corresponds to a KeyQuencer macro, that macro will execute.

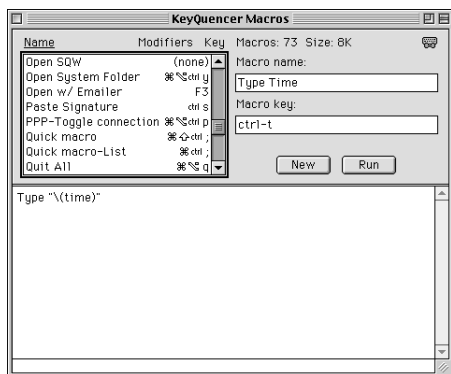
A Precise Definition of a Keystroke: A keystroke is a combination of keys that includes exactly one typing key (for example “a”, “1” or “<”) and any number of modifier keys. There are four modifier keys: Command, Shift, Option and Control. Some examples of valid keystrokes are F2, Command-H, Shift-Option-4, or Control-T.

What Is the KeyQuencer Editor?

The KeyQuencer Editor is the program you use to make, change, or delete macros. Before you can make the simple macros described in this chapter, you may need to open the KeyQuencer Editor.

How Do I Open the KeyQuencer Editor?

- Press Option-Command-K.
- Open the **Apple** menu and choose **KeyQuencer Helpers: Open Editor**.
- Open the *KeyQuencer* folder and double-click *KeyQuencer Editor*.



What About the KeyQuencer Panel?

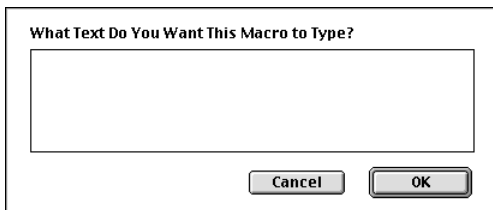
The KeyQuencer panel is like a mini-Editor. It's not as full featured as the KeyQuencer Editor. It has no startup code and is not required to use KeyQuencer. It is not installed by default. If you don't have it, don't worry about it. It is included for backward compatibility with previous versions of KeyQuencer only. We don't suggest that you use it and we will probably remove it from future versions.

What Is a Text Macro?

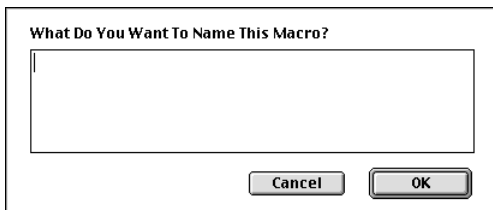
A text macro is a macro that types some text. For example, you might make a text macro that types your name, or "Sincerely yours," or "United States of America". Text macros are very convenient. Make text macros for all of the words and phrases that you must repeatedly type. Text macros are very convenient for chat rooms and online games where often you must type the same text over and over.

How Do I Automatically Make a Text Macro?

1. Open the **Apple** menu and choose **KeyQuencer Helpers: Make Text Macro**.
2. Enter the text that you want your macro to type.



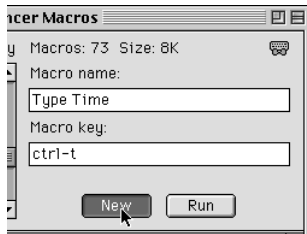
3. Enter the name that you want to give your new macro.



4. Press the keystroke that you want to assign to your macro.
5. The **Make Text Macro** will create your macro and save it automatically.

How Do I Manually Make a Text Macro?

1. Open the KeyQuencer Editor.
2. If you just launched the Editor, the *Macro name* field will already say “New Macro”. If you don’t see “New Macro”, then click the **New** button.



3. Enter the name that you want to give your new macro.



4. Press Tab or click in the *Macro key* field.



5. Press the keystroke that you want to assign to your macro.
6. Press tab or click in the *Macro text* field.
7. Type the following, exactly as it appears below, replacing the text in brackets as appropriate:
Type "[text you want to type here]"
8. Open the **File** menu and choose **Save Suitcase** (Command-S) to save your macro.

What Is an Open Macro?

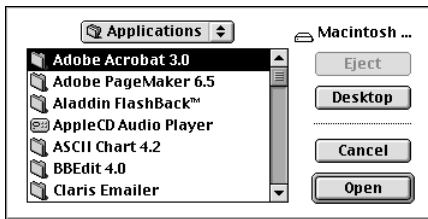
An open macro is a macro that opens an application, document, or folder. For example, you might make an open macro that launches Navigator, opens a FileMaker Pro database, or opens the *Documents* folder. Make open macros for all of the applications, documents, and folders that you want to be able to open with a keystroke.

How Do I Automatically Make an Open Macro?

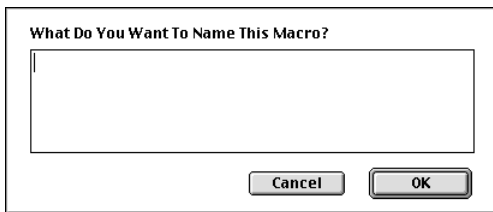
(Not available in KeyQuencer Lite)

1. Open the **Apple** menu and choose **KeyQuencer Helpers: Make Open Macro**.

2. Choose the file that you want your macro to open.



3. Enter the name that you want to give your new macro.

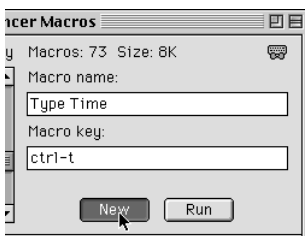


4. Press the keystroke that you want to assign to your macro.
5. The **Make Open Macro** will create your macro and save it automatically.

How Do I Use the Contextual Menu to Make an Open Macro?

(Not available in KeyQuencer Lite)

1. Open the KeyQuencer Editor.
2. If you just launched the Editor, the *Macro name* field will already say “New Macro”. If you don’t see “New Macro”, then click the **New** button.



3. Enter the name that you want to give your new macro.



4. Press Tab or click in the *Macro key* field.



5. Press the keystroke that you want to assign to your macro.

6. Switch to the Finder and control click the file you want to open. Choose **KeyQuencer Actions: Macro-Copy Open Macro** from the contextual menu that appears.
7. Switch to the KeyQuencer Editor, select the *Macro text* field, and **Paste** (Command-V) in the text on the clipboard. It should look something like this:


```
Open "Macintosh HD:Applications:Netscape Navigator™ Folder:Netscape
Navigator™ 4.03"
```
8. Open the **File** menu and choose **Save Suitcase** (Command-S) to save your macro.

How Do I Manually Make an Open Macro?

1. Open the KeyQuencer Editor.
2. If you just launched the Editor, the *Macro name* field will already say “New Macro”. If you don’t see “New Macro”, then click the **New** button.
3. Enter the name that you want to give your new macro.
4. Press Tab or click in the *Macro key* field.
5. Press the keystroke that you want to assign to your macro.
6. Press tab or click in the *Macro text* field.
7. Type the following, exactly as it appears below:


```
Open
```
8. If the file you want to open is an application, open the **Macros** menu and choose **Insert Signature...** (Command-U). For any other kind of file, open the **Macros** menu and choose **Insert Pathname...** (Command-I).
9. Choose the file that you want your macro to open and click **Open**. The signature or pathname will be appended to your macro. It should look something like this for an application:


```
Open 'AD65 '
```


Or like this for any other kind of file:


```
Open "Macintosh HD:Binary:Binary Website:binarysoft web site"
```
10. Open the **File** menu and choose **Save Suitcase** (Command-S) to save your macro.

What Is a Menu Macro?

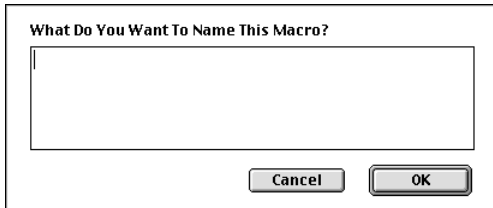
A menu macro is a macro that chooses a menu item. For example, you might make a menu macro that chooses the **View Options...** item from the **View** menu of the Finder. Make menu macros to automatically choose menu commands with a keystroke.

How Do I Automatically Make a Menu Macro?

This only works for standard menu items in named menus. For items in submenus, or in the **Apple** menu, see *How Do I Manually Make a Menu Macro?*.

1. Open the **Apple** menu and choose **KeyQuencer Helpers: Make Menu Macro**.

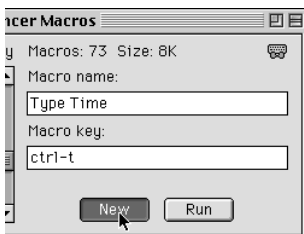
2. Enter the name of the menu that contains the menu item you want to choose. For example, **File**, **Edit**, or **View**.
3. Enter the name of the menu item you want to choose. For example, **Clear**, **Restart**, or **Preferences....** Be sure to include the ellipsis (the three little dots) at the end of the item name if necessary. You can generate the ellipsis character by pressing Option-; (Option-Semicolon).
4. Enter the name that you want to give your new macro.



5. Press the keystroke that you want to assign to your macro.
6. The **Make Menu Macro** will create your macro and save it automatically.

How Do I Manually Make a Menu Macro?

1. Open the KeyQuencer Editor.
2. If you just launched the Editor, the *Macro name* field will already say "New Macro". If you don't see "New Macro", then click the **New** button.



3. Enter the name that you want to give your new macro.



4. Press Tab or click in the *Macro key* field.



5. Press the keystroke that you want to assign to your macro.
6. Press tab or click in the *Macro text* field.
7. Type the following, exactly as it appears below, replacing the text in brackets as appropriate:

Menu "[name of menu]" "[name of menu item]"

For example, a macro that chooses the **Restart** item of the Finder's **Special** menu looks like this:

Menu "Special" "Restart"

Be sure to include the ellipsis (the three little dots) at the end of the item name if necessary. You can generate the ellipsis character by pressing Option-; (Option-Semicolon). For example:

Menu "Edit" "Preferences..."

You can specify an item in the apple menu like this:

Menu apple "Chooser"

To specify an item in a submenu, omit the menu name, and replace it with the *submenu* parameter.. For example, here's a macro that chooses the **by Name** item from the **Arrange** submenu of the Mac OS 8 Finder's **View** menu:

Menu "by Name" submenu

8. Open the **File** menu and choose **Save Suitcase** (Command-S) to save your macro.

What Is a Button Macro?

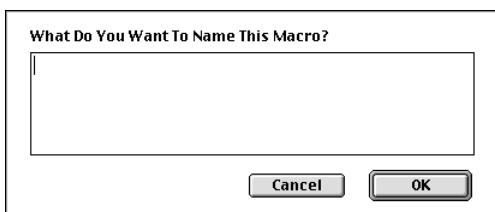
A button macro is a macro that presses a button in a window or dialog. For example, you might make a button macro that presses the **OK** button in a confirmation dialog, or that clicks a radio button, or checkbox. Use button macros to press buttons that have no key equivalents.

.....
Some buttons are nonstandard and tricky. You may find that your button macro doesn't work, even though you followed all the instructions. If this happens, make a click macro instead to click the button.
.....

How Do I Automatically Make a Button Macro?

This only works for standard, named buttons. For nonstandard or nameless buttons, see the manual instructions following this section.

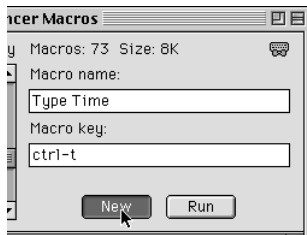
1. Open the **Apple** menu and choose **KeyQuencer Helpers: Make Button Macro**.
2. Enter the name of the button that you want to press. For example, **OK**, **Cancel**, or **Find**. Be sure to include the ellipsis (the three little dots) at the end of the item name if necessary. You can generate the ellipsis character by pressing Option-; (Option-Semicolon).
3. Enter the name that you want to give your new macro.



4. Press the keystroke that you want to assign to your macro.
5. The **Make Button Macro** will create your macro and save it automatically.

How Do I Manually Make a Button Macro for a Named Button?

1. Open the KeyQuencer Editor.
2. If you just launched the Editor, the *Macro name* field will already say “New Macro”. If you don’t see “New Macro”, then click the **New** button.



3. Enter the name that you want to give your new macro.



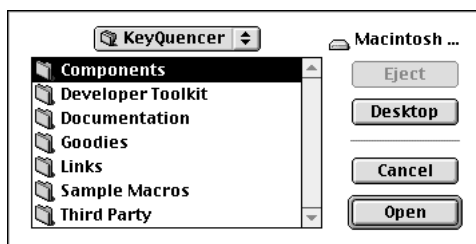
4. Press Tab or click in the *Macro key* field.



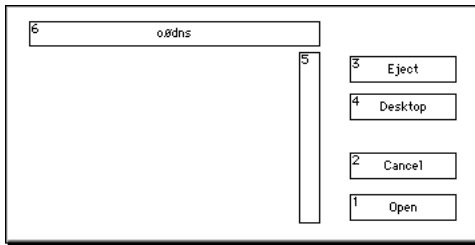
5. Press the keystroke that you want to assign to your macro.
6. Press tab or click in the *Macro text* field.
7. Type the following, exactly as it appears below, replacing the text in brackets as appropriate:
Button "[name of button]"
8. Open the **File** menu and choose **Save Suitcase** (Command-S) to save your macro.

How Do I Manually Make a Button Macro for a Nameless Button?

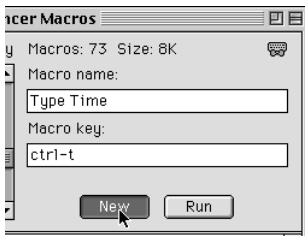
1. Switch to the appropriate application and bring the window or dialog to the front that contains the button, checkbox, or radio button that you want your macro to select.



2. Press Control-Option-Command-7. The window will change appearance, and in place of the button will be an outline and a number. Make a note of that number.



3. Open the KeyQuencer Editor.
4. If you just launched the Editor, the *Macro name* field will already say “New Macro”. If you don’t see “New Macro”, then click the **New** button.



5. Enter the name that you want to give your new macro.



6. Press Tab or click in the *Macro key* field.



7. Press the keystroke that you want to assign to your macro.
8. Press tab or click in the *Macro text* field.
9. Type the following, exactly as it appears below, replacing the text in brackets as appropriate:

Button [button number]

Your completed macro should look something like this:

Button 3

10. Open the **File** menu and choose **Save Suitcase** (Command-S) to save your macro.

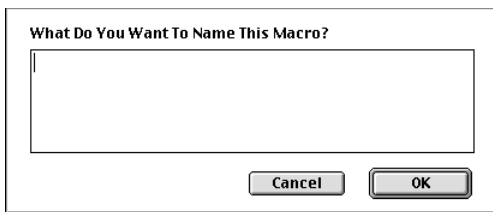
What Is a Click Macro?

A click macro is a macro that clicks the mouse somewhere — anywhere. For example, you might use a click macro to select a field in a database, click a button on a toolbar, or click the collapse box of a window. Make click macros to automate with a keystroke anything that ordinarily requires a mouse click.

How Do I Automatically Make a Click Macro?

(Not available in KeyQuencer Lite)

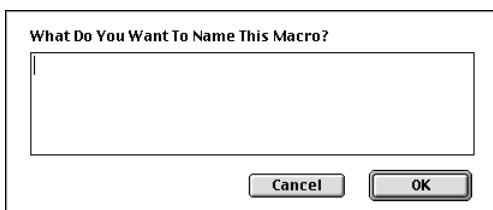
1. Make sure the location you want to click is visible on screen. If the click location is in a dialog, then skip to the next section *How Do I Make a Click Macro in a Dialog?*.
2. Open the **Apple** menu and choose **KeyQuencer Helpers: Make Click Macro**.
3. The mouse will turn into a crosshair. Click the mouse where you want your macro to click.
4. Enter the name that you want to give your new macro.



5. Press the keystroke that you want to assign to your macro.
6. The **Make Click Macro** will create your macro and save it automatically.

How Do I Make a Click Macro in a Dialog?

1. Bring the dialog to the front that contains the location that you want to click.
2. Press Control-Command-C to activate the **Make Click Macro** standard macro.
3. The mouse will turn into a crosshair. Click the mouse at the point in the dialog where you want your macro to click.
4. Dismiss the dialog so that the macro can proceed.
5. Enter the name that you want to give your new macro.



6. Press the keystroke that you want to assign to your macro.
7. The **Make Click Macro** will create your macro and save it automatically.

How Do I Manually Make a Click Macro?

This one is a little tricky, so get a pen and paper handy: you'll need it.

1. Switch to the appropriate application and bring the window or dialog to the front that contains the location that you want your macro to click.
2. Press Control-Option-Command-6. The mouse will turn into a crosshair . Click the mouse where you want your macro to click.
3. A small KeyQuencer window will appear in the top left corner.

Coordinate: 1 2

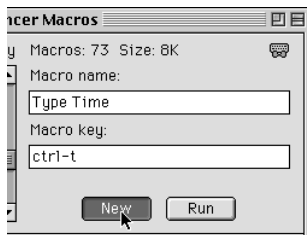
Screen:	336, 296
Window:	336, 296
Width:	0
Height:	0

This window will tell you the screen and window coordinates that correspond to the point you clicked the mouse. Make a note of the screen and window coordinates like this, replacing “xx” with the numbers you see on screen:

Screen1 = xx, Screen2 = xx

Window1 = xx, Window2 = xx

4. Open the KeyQuencer Editor.
5. If you just launched the Editor, the *Macro name* field will already say “New Macro”. If you don’t see “New Macro”, then click the **New** button.



6. Enter the name that you want to give your new macro.

Macro name:
New Macro

7. Press Tab or click in the *Macro key* field.

Macro key:
(none)

8. Press the keystroke that you want to assign to your macro.
9. Press tab or click in the *Macro text* field.

10. If you want your macro to click on the exact same part of the screen every time, then type the following, exactly as it appears below, replacing the text in brackets as appropriate:

Click [Screen1] [Screen2]

Your completed macro should look something like this:

Click 244 30

11. If you want your macro to click on the exact same part of a window or dialog every time, then type the following, exactly as it appears below, replacing the text in brackets as appropriate:

Click [Window1] [Window2] window

Your completed macro should look something like this:

Click 170 13 window

12. Open the **File** menu and choose **Save Suitcase** (Command-S) to save your macro.

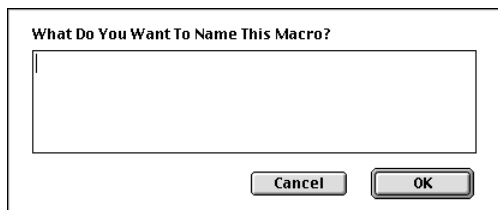
What Is a Drag Macro?

A drag macro is a macro that drags the mouse from one point to another. For example, you might use a drag macro to make a drag to the Trash macro, or to choose a menu item from a floating palette or toolbar. Make drag macros to automate with a keystroke anything that ordinarily requires a mouse drag.

How Do I Automatically Make a Drag Macro?

(Not available in KeyQuencer Lite)

1. Make sure the locations you want to drag from and to are visible on screen. If the drag location is in a dialog, then skip to the next section *How Do I Make a Drag Macro in a Dialog?*.
2. Open the **Apple** menu and choose **KeyQuencer: Make Drag Macro**.
3. The mouse will turn into a crosshair. Move the mouse to the starting point and then Click-drag the mouse to the end point. If you are trying to select a menu item you'll have to estimate, because the menu will not popup during your drag.
4. Enter the name that you want to give your new macro.

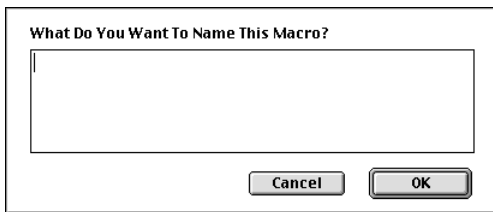


5. Press the keystroke that you want to assign to your macro.
6. The **Make Drag Macro** will create your macro and save it automatically.

How Do I Make a Drag Macro in a Dialog?

(Not available in KeyQuencer Lite)

1. Bring the dialog to the front that contains the location that you want to click.
2. Press Control-Command-D to activate the **Make Drag Macro** standard macro.
3. The mouse will turn into a crosshair. Move the mouse to the starting point and then Click-drag the mouse to the end point. If you are trying to select a menu item you'll have to estimate, because the menu will not popup during your drag.
4. Dismiss the dialog so that the macro can proceed.
5. Enter the name that you want to give your new macro.



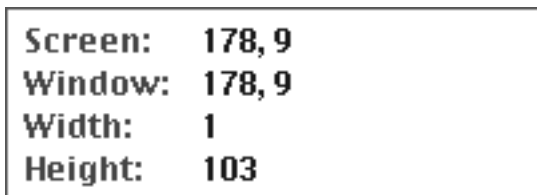
6. Press the keystroke that you want to assign to your macro.
7. The **Make Drag Macro** will create your macro and save it automatically.

How Do I Manually Make a Drag Macro?

Get a pen and paper ready before you begin.

1. Switch to the appropriate application and bring the window or dialog to the front that contains the to and from locations that you want your macro to drag.
2. Press Control-Option-Command-6. The mouse will turn into a crosshair. Move the mouse to the starting point and then Click-drag the mouse to the end point. If you are trying to select a menu item you'll have to estimate, because the menu will not popup during your drag.
3. A small KeyQuencer window will appear in the top left corner.

Coordinate: 1 2



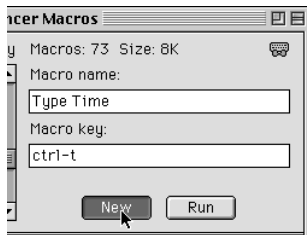
This window will tell you the screen and window coordinates that correspond to your starting point you clicked the mouse and the relative width and height of your drag. Make a note of all the values, like this, replacing “xx” with the numbers you see on screen:

Screen1 = xx, Screen2 = xx

Window1 = xx, Window2 = xx

Width = xx, Height = xx

4. Open the KeyQuencer Editor.
5. If you just launched the Editor, the *Macro name* field will already say “New Macro”. If you don’t see “New Macro”, then click the **New** button.



6. Enter the name that you want to give your new macro.



7. Press Tab or click in the *Macro key* field.



8. Press the keystroke that you want to assign to your macro.
9. Press tab or click in the *Macro text* field.
10. If you want your macro to drag on the exact same parts of the screen every time, then type the following, exactly as it appears below replacing the text in brackets as appropriate:

Drag from [Screen1] [Screen2] to [Width] [Height] relative slower

Your completed macro should look something like this:

Drag from 244 30 to 36 72 relative slower

11. If you want your macro to drag on the exact same parts of a window or dialog every time, then type the following, exactly as it appears below replacing the text in brackets as appropriate:

Drag from [Screen1] [Screen2] to [Width] [Height] relative window slower

Your completed macro should look something like this:

Drag from 158 49 to 0 50 relative window slower

12. Open the **File** menu and choose **Save Suitcase** (Command-S) to save your macro.

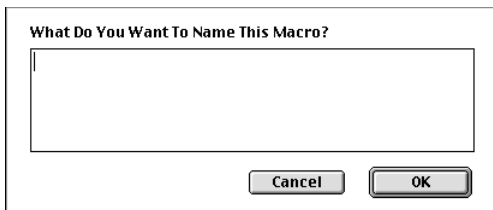
What Is a Key Macro?

A key macro is a macro that presses. For example, you might make a key macro that presses Command-C to copy some text. Make key macros for keystrokes that are hard to press or to select menu commands or buttons.

How Do I Automatically Make a Key Macro?

(Not available in KeyQuencer Lite)

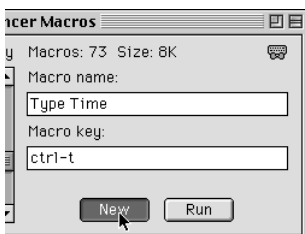
1. Open the **Apple** menu and choose **KeyQuencer Helpers: Make Key Macro**.
2. Press the keystroke that you want your macro to press.
3. Enter the name that you want to give your new macro.



4. Press the keystroke that you want to assign to your macro.
5. The **Make Key Macro** will create your macro and save it automatically.

How Do I Manually Make a Key Macro?

1. Open the KeyQuencer Editor.
2. If you just launched the Editor, the *Macro name* field will already say "New Macro". If you don't see "New Macro", then click the **New** button.



3. Enter the name that you want to give your new macro.



4. Press Tab or click in the *Macro key* field.



5. Press the keystroke that you want to assign to your macro.
6. Press tab or click in the *Macro text* field.

7. Type the following text, depending on the key you want to press. Remember, a key macro presses only one keystroke. Use a type macro for words or phrases.

This is how Shift-Command-C looks:

```
Key shift cmd "c"
```

This is how Control-F1 looks:

```
Key ctrl F1
```

This is how Option-tab looks:

```
Key opt tab
```

8. Open the **File** menu and choose **Save Suitcase** (Command-S) to save your macro.

5: Using the KeyQuencer Editor

This chapter explains in detail how to use the KeyQuencer Editor to:

- Make, change, or delete macros
- Use the Command Palette to speed up macro making
- Use the Command Help window to get onscreen help
- Work with multiple macro suitcases
- Import or copy and paste macros between suitcases
- Create Stand-Alone Applets
- Change the font, layout, text wrap, and save settings of the KeyQuencer Editor.

Running, Creating, Changing and Deleting Macros

All the instructions below assume that the KeyQuencer Editor is open. To open the Editor, open the **Apple** menu and choose **KeyQuencer: Open KeyQuencer Editor**.

How Do I Run a Macro?

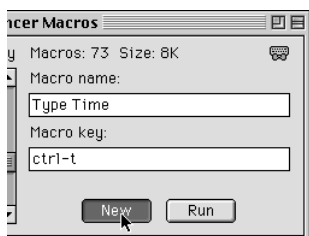
- Select the macro you want to run by clicking it in the list in the upper left part of the *KeyQuencer Macros* window and click **Run**.



- Double-click the macro.

How Do I Make a New Macro?

1. Click **New** or open the **Macro** menu and choose **New Macro** (Command-K).



2. Click in the *Macro name* field and type a name for your new macro.



3. Click in the *Macro key* field. Type the keystroke you want to assign to the macro (or leave it blank to create an “orphan”). Press the Delete key if you make a mistake.



4. Type the text of your macro in the bottom area.
5. Open the **File** menu and choose **Save Suitcase** (Command-S) to save your macro.

.....
You can use the Tab key to change the active item in a suitcase window and cycle through the Macro list, Macro name, Macro key and Macro text fields.
.....

How Do I Change a Macro's Name?

1. Select the macro whose name you want to change by clicking it in the list in the upper left part of the *KeyQuencer Macros* window.
2. Click the *Macro name* field.
3. Type the new name.
4. Open the **File** menu and choose **Save Suitcase** (Command-S) to save your changes.

How Do I Change a Macro's Keystroke?

1. Select the macro whose keystroke you want to change by clicking it in the list in the upper left part of the *KeyQuencer Macros* window.
2. Click the *Macro key* field.
3. Type the new keystroke.
4. Open the **File** menu and choose **Save Suitcase** (Command-S) to save your changes.

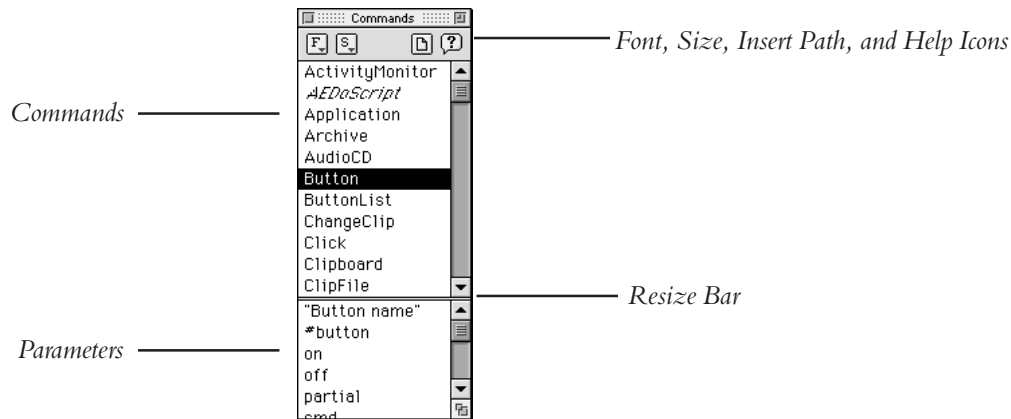
.....
*The Command-S keystroke won't work if the Macro Key field is still selected, since it changes the macro's keystroke instead of invoking the **Save** menu command.*
.....

How Do I Delete a Macro?

1. Select the macro you want to delete by clicking it in the list in the upper left part of the *KeyQuencer Macros* window. Shift-click, Command-click, or Select All (Command-A) to choose more than one macro to delete.
2. Open the **Macros** menu and choose **Delete Macros** (Delete).
3. Click **Delete** to confirm and delete the selected macro(s).

What Is the Commands Palette?

The Commands palette shows you at a glance all available KeyQuencer commands and parameters. It also makes it easier to build macros and get help on a particular command.



How Do I Show the Commands Palette?

The Commands Palette is displayed by default, but if you don't see it:

- Open the **Windows** menu and choose **Show Commands Palette** (Command-Y).


How Do I View a Command's Parameters?

- Click on the command in the Commands Palette. Its parameters will appear below.


How Do I Insert a Command or Parameter into a Macro?

- Double-click on a command or parameter in the Commands Palette to insert it into your macro's text.

How Do I Insert a File's Pathname into a Macro?

1. Click on the document icon  at the top of the Commands Palette.
2. Choose the file whose path you want to insert into your macro's text and click **Open**.


How Do I Insert a File's Creator Type into a Macro?

1. Command-click on the document icon  at the top of the Commands Palette.
2. Choose the file whose signature you want to insert into your macro's text and click **Open**.

How Do I Change the Font of the Commands Palette?

- Click the "F" icon  at the top of the Commands Palette and then choose the display font you want.

How Do I Change the Font Size of the Commands Palette?

- Click the "S" icon  at the top of the Commands Palette and then choose the font size you want.

How Do I Change the Layout of the Commands Palette?


- Move the cursor to the horizontal separating line between the two areas and drag up or down to change the proportional space allotted to commands and parameters.



What is the Command Help Window?

The Command Help window shows you detailed information about the command selected in the Commands Palette.

How Do I Show the Command Help Window?

- Open the **Windows** menu and choose **Show Command Help** (Command-H).
- Click the help balloon icon  in the Commands Palette.

How Do I See Help on the KeyQuencer Macro Language?

To see help text on the KeyQuencer Macro language and using escape sequences (instead of help on a specific command):

- Close the Commands Palette by opening the **Windows** menu and choosing **Hide Commands Palette** (Command-Y). The Command Help window will show the macro language help text.

What Is a Macro Suitcase?

A *macro suitcase* is a file that contains one or more macros. KeyQuencer uses four kinds of macro suitcases: *Installed Macros*, *Editor Suitcase*, *Launcher Suitcase*, and *Stand-Alone Applet*.

- *Installed Macros* are your active macros. This type of suitcase is reserved exclusively for the installed *KeyQuencer Macros* file, which is stored in the *KeyQuencer Preferences* folder in your *Preferences* folder.
- A *Launcher Suitcase* is a collection of macros that, when double-clicked on, opens using the KeyQuencer Launcher. You use Launcher Suitcases for completed, ready-to-use collections of macros.
- An *Editor Suitcase* is a collection of macros that, when double-clicked on, opens the KeyQuencer Editor. You use Editor Suitcases for collections of macros that are in development or frequently changed.

- A *Stand-Alone Applet* is a self-executing macro suitcase containing only one macro. It is an application that, when launched, executes the macro it contains. Stand-Alone Applets are very useful. Put a Stand-Alone Applet in your *Apple Menu Items* folder to add a macro to your **Apple** menu. You can also put a Stand-Alone Applet in the *Startup Items* folder or the *Shutdown Items* folder to execute a KeyQuencer macro whenever you startup or shut down your computer.


Here are step-by-step instructions for working with all the different types of suitcases using the KeyQuencer Editor.

How Do I Open the Installed Macros Suitcase?


The Installed Macros suitcase is titled *KeyQuencer Macros* and opens by default when you launch the Editor. If you ever need to open it though, just...

- Open the **File** menu and choose **Open Installed Macros** (Command-M).


How Do I Make an Editor Suitcase?

1. Open the **File** menu and choose **New Suitcase** (Command-N).
2. Create one or more macros.
3. Open the **Macros** menu and choose **Editor Suitcase** from the **Suitcase Type** submenu. You can also choose **Editor Suitcase** from the popup menu that appears when you click the small KeyQuencer icon  in the suitcase window.
4. Open the File menu and choose **Save Suitcase** (Command-S).
5. Type in a name for your new suitcase, choose a location on your disk-drive, and click **Save**.

How Do I Make a Launcher Suitcase?

1. Open the **File** menu and choose **New Suitcase** (Command-N).
2. Create one or more macros.
3. Open the **Macros** menu and choose **Launcher Suitcase** from the **Suitcase Type** submenu. You can also choose **Launcher Suitcase** from the popup menu that appears when you click the small KeyQuencer icon  in the suitcase window.
4. Open the File menu and choose **Save Suitcase** (Command-S).
5. Type in a name for your new suitcase, choose a location on your disk-drive, and click **Save**.


How Do I Make a Stand-Alone Applet?

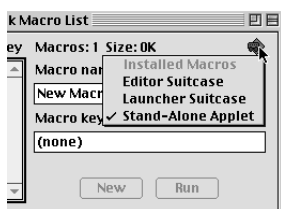
1. Open the **File** menu and choose **New Suitcase** (Command-N).
2. Create one (and only one macro) or paste in a macro from another suitcase.
3. Open the **Macros** menu and choose **Stand-Alone Applet** from the **Suitcase Type** submenu. You can also choose **Stand-Alone Applet** from the popup menu that appears when you click the small KeyQuencer icon  in the suitcase window.
4. Open the **File** menu and choose **Save Suitcase** to save your Stand-Alone Applet.
5. Type in a name for your new suitcase, choose a location on your disk, and click **Save**.

How Do I Copy a Macro from One Suitcase to Another?

1. Open the macro suitcase that contains the macro that you want to copy.
2. Click on the name of the macro that you want to add.
3. Open the **Edit** menu and choose **Copy** (Command-C).
4. Open the macro suitcase into which you want to copy the macro.
5. Click any macro in the macro list (to make that area active)
6. Open the **Edit** menu and choose **Paste** (Command-V).
7. Open the **File** menu and choose **Save Suitcase** (Command-S) to save your changes.

How Do I Change a Suitcase's Type?

1. Open the macro suitcase whose type you want to change.
2. Open the **Macros** menu and choose a new type from the **Suitcase Type** submenu. You can also choose a new type from the popup menu that appears when you click the small KeyQuencer icon  in the suitcase window.



3. Open the **File** menu and choose **Save Suitcase** to save your macro.

.....
Stand-Alone Applets can only have exactly one macro. They become Launcher Suitcases if you add more macros.
.....

How Do I Merge Two Suitcases?

1. Open the suitcase into which you want to import macros, then, open the **File** menu and choose **Import Suitcase....**
2. Select the suitcase whose contents you want to import and click **Open**.

How Do I Backup or Replace the Installed Macros Suitcase?

1. Open the *KeyQuencer Preferences* folder. It is located in the *Preferences* folder in your *System Folder*.
2. Either backup or replace the macro suitcase named *KeyQuencer Macros*.
3. If you replaced the *KeyQuencer Macros* suitcase, open the KeyQuencer Editor to load your new macros into memory and make them active.

How Do I Customize the Editor?

You can change the appearance of the text, set text wrapping, change the macro sort order, alter the layout of the window, and set the save preferences in the KeyQuencer Editor to suit your needs.

How Do I Change the Font Face in the Editor?

- Open the **Options** menu and choose a new font from the **Text Font** submenu.

How Do I Change the Font Size in the Editor?

- Open the **Options** menu and choose a new size from the **Text Size** submenu.

How Do Turn On or Off Text Wrapping?

- When text wrapping is on, a line of text which extends past the right edge of the window will automatically “wrap” to the next line. This is for visual clarity only; KeyQuencer does not insert carriage returns when text wrapping is active.
- Open the **Options** menu and choose **Wrap Text** (Command-T).

How Do I Change the Sort Order of My Macros?

- Click the word **Name** at the top of the list to sort your macros by name.
- Click the word **Modifiers** at the top of the list to sort your macros by modifier key.
- Click the word **Keys** at the top of the list to sort your macros by key.

How Do I Change the Layout of the Editor Window?

- Move the cursor to the horizontal separating line between the two areas and drag up or down to change the proportional space allotted to the header area and the macro text area.



How Do I Automatically Save Suitcases When I Close Them?

- Open the **Options** menu and choose **Auto Save Suitcases**.

How Do I Automatically Save Changes Before Running Macros?

- Open the **Options** menu and choose (if necessary) **Don't Save Before Run** so that it is *not* checked.

Editor Shortcuts

- To insert a command or parameter in a macro's text, double-click the command or parameter in the Commands Palette.
- To insert a file's pathname, click the small document icon  in the Commands Palette or open the **Macros** menu and choose **Insert Pathname...** (Command-I).
- To insert a file's signature or creator, command-click the small document icon  in the Commands Palette or open the **Macros** menu and choose **Insert Signature...** (Command-U).
- To comment a block of text select it, then open the **Macros** menu and choose **Insert Comment** (Command-Hyphen {"-"}).
- To remove all the comments in a macro, open the **Macros** menu and choose **Strip Comments**.
- To cycle through the open windows, open the **Windows** menu and choose **Show Next Window** (Command- "=").
- To print a list of your macros, **Select All** (Command-A) the macros in a suitcase, **Copy** (Command-C), then open any text editor and **Paste** (Command-V) the macro list in a new window, and print.

6: Writing Macros

This chapter walks you step by step through the process of writing complex KeyQuencer macros. By the end of this chapter, you will know how to make macros that are capable of performing a sequence of actions. You will also learn how to include comments in your macros, how to debug your macros, and how to use KeyQuencer variables.

Writing macros can be tricky. Don't feel frustrated if at first this chapter seems hopelessly confusing. Just keep writing macros; eventually you'll get the hang of it.

What Is a Parameter?

Let's start with an easy macro: a macro that executes the **Clear** command in the **Edit** menu. This macro is somewhat useful, because **Clear** rarely has an assigned command key. Here it is:

```
Menu "Edit" "Clear"
```

This should look familiar to you. It's a standard menu macro, as described in Chapter 4 of this User Guide. Now let's examine it more closely.

Menu is a KeyQuencer *command*. Commands tell KeyQuencer to do something. In this case, Menu tells KeyQuencer to execute a menu command.

"Edit" and "Clear" are *parameters*. Parameters modify, specify or limit commands. In this case, "Edit" and "Clear" tell KeyQuencer which menu item to execute.

Here's an analogy to help you understand the difference between a command and a parameter. Pretend I'm writing a macro to control you! I want you to get me the television remote control from the living room. Here's my macro:

```
Get "Living Room" "Remote"
```

Here, Get is the command because it's the action, it's what you do. "Living Room" and "Remote" are parameters. They tell you where and what to get.

Let's look at another easy macro. This one closes a window:

```
Click closebox
```

Here, Click is the command and closebox is the parameter. Is this making sense?

Here's one more easy macro. This one restarts your computer:

```
Session restart
```

Session is the command and restart is the parameter.

If you know the right command and parameters, most macros are as simple as the three we just created. Use the Command Help window in the KeyQuencer Editor to get detailed information about each of KeyQuencer's commands. Also be sure to look at the *Command Samples* suitcase in the *Macros* folder for lots of sample macros with detailed comments that show you how to use the different parameters.

What Is a Pathname?

Many KeyQuencer commands use what's called a file's pathname as a parameter. A good example of this is the Open command:

```
Open "Macintosh HD:Applications:SimpleText"
```

The Open command tells KeyQuencer to launch an application or open a document. The parameter "Macintosh HD:Applications:SimpleText" is the *pathname* to SimpleText. A pathname tells KeyQuencer where to look to find a file. This pathname says that *SimpleText* is stored in the *Applications* folder on the disk named *Macintosh HD*.

.....
*Use the **Insert Pathname** command in the **Macros** menu to automatically insert the pathname of any file you choose.*
.....

What Does "Debug" Mean?

Here is a macro which is supposed to open the Numbers control panel. However, it doesn't quite work, as we'll soon see:

```
Open panels "Numbers"
```

The Open command tells KeyQuencer to open a file, the panels parameter tells KeyQuencer to look in the *Control Panels* folder, and "Numbers" tells KeyQuencer the exact name of the file to open.

Go ahead and try this macro. Open the KeyQuencer Editor, type it in and run it. What happens? It doesn't quite work. The Numbers control panel opens, but it doesn't come to the front. It remains hidden behind the KeyQuencer Editor window. Let's change our macro a little to fix this problem. Here is our new, improved macro:

```
SwitchApp "Finder"  
Open panels "Numbers"
```

We added a SwitchApp command to bring the Finder to the front (control panels open into the Finder's application layer). What we just did is called *debugging*. Debugging means removing bugs. A bug is a defect or error which prevents a macro from working as it should. In this case, our bug was that the Numbers control panel was opening in the background. We fixed it, or debugged it, by adding a command to bring the Finder to the front.

How Do I Automate Popup Menus?

Suppose that you frequently need to change your number format to French Canadian because some of your important customers live in Quebec. How would you make a macro to do that?

With the Numbers control panel already open, we need KeyQuencer to open the **Number Format** menu and choose **French Canadian**. We can do this with KeyQuencer's PopUpMenu command.

The PopUpMenu command needs two parameters: which popup menu to open, and which item to choose from it. It would be nice if we could specify both of these parameters by name, like this:

```
PopUpMenu "Number Format" "French Canadian"
```

But if you try it that way, you'll get an error. This is because KeyQuencer expects a popup menu to be specified by its button index, not by its name.

Almost every clickable item in a window has an index, or a unique number assigned to it. To specify a popup menu, we must use its index. But how do we find out an item's index? Fortunately, we can use KeyQuencer's standard macro **Macro-Show Button List** to find out a button's index.

How Do I Find Out a Popup Menu's Index?

1. Bring the window or dialog to the front that contains the popup menu.
2. Press Control-Option-Command-7. The window will change appearance, and in place of the popup will be an outline and a number. That number is the popup menu's index.

If you do this with the Numbers control panel open, you will see that the index of the **Number Format** menu is 5. Armed with this information, our new macro looks like this:

```
SwitchApp "Finder"
Open panels "Numbers"
PopUpMenu button 5 "French Canadian"
```

But when we try our new macro, we are likely to get an error “can't find popup menu button.” This is because of bad timing: KeyQuencer is looking for the popup menu before the Numbers panel is open. To correct this we need to insert a `WaitWindow` command to slow down the macro. This will pause the macro until the Numbers panel is completely opened.

While we're at it, let's clean up after ourselves and close the Numbers panel when we're done. We will do this by using the universal Command-W keystroke to close the window. Here is the finished macro:

```
SwitchApp "Finder"
Open panels "Numbers"
WaitWindow "Numbers"
PopUpMenu button 5 "French Canadian"
Key cmd "w"
```

What Is a Macro Comment?

Comments are bits of explanatory text inserted in your macros. They do not affect the macro and are ignored by KeyQuencer. You can add comments to your macros to help you (and others) understand what they do and how they work.

How Do I Add Comments to a Macro?

- Type two consecutive hyphens (“--”). Everything on a line following a double hyphen is ignored.

The macro in the previous example could have been helped by some comments. Here is the same macro, with comments added:

```
-- Changes number format.
-- To French Canadian
SwitchApp "Finder" -- Switch to Finder
Open panels "Numbers" -- Open Numbers panel
WaitWindow "Numbers" -- Wait till it's open
PopUpMenu button 5 "French Canadian" -- Choose French Can. Format
Key cmd "w" -- Close window
```

.....
Comments do take up memory so use them sparingly.
.....

What Is a Variable?

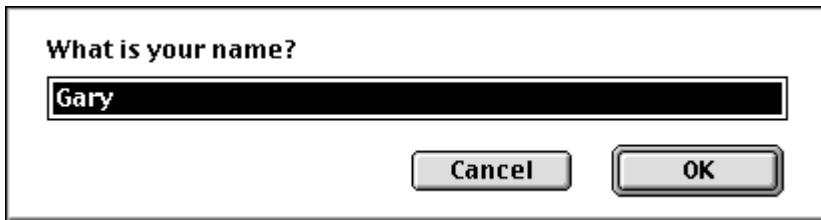
A variable is a holding place for information. A variable contains information, like a number or a word. Variables are so named because the information they contain is variable and can change.

KeyQuencer lets you use variables to create extremely powerful macros (KeyQuencer Lite doesn't support variables). Let's look at an example. Here is a macro that asks for a name and then displays it onscreen:

```
SetVariable name ask 'What is your name?'  
Message $name
```

How does this macro work? Let's step through it.

The first line (`SetVariable name ask 'What is your name?'`) defines a new variable named `name`. This variable will contain the name entered by the user. The `ask` parameter opens a dialog which directs the user to enter some text. `'What is your name?'` is the prompt text that appears in this dialog.



The second line (`Message $name`) uses variable `name` as a parameter for the `Message` command.



The dollar sign (\$) tells KeyQuencer to enclose the contents of `name` with double quotes. So for example, if the user enters the name Gary, then after KeyQuencer makes the substitutions, the second line looks like this:

```
Message "Gary"
```

In addition to the dollar-sign, you can also use the ampersand ("&") to enclose the substituted variable text in single quotes or the percent sign ("%") to substitute the variable text without any quotes.

What Is an Escape Sequence?

Suppose you want to make a macro that types something like “Harry "Happy" Higgins” If you try this:

```
Type "Harry "Happy" Higgins"
```

KeyQuencer will type only “Harry ” and then display an “Unknown keyword” error message. This is because KeyQuencer interprets the double-quote (") before the word `Happy` as the closing quote of the string parameter of the type command. The way to solve this problem is with escape sequences.

An escape sequence is special code, embedded within double or single quotes that stands for something else. Escape sequences are changed at runtime by KeyQuencer into their assigned values, just like a variable. An escape sequence is always preceded by a backslash (“\”) character (which is located directly above the Return key on most keyboards).

The escape sequence “\q” stands for a double quote. We can use this escape sequence to correct our macro:

```
Type "Harry \qHappy\q Higgins"
```

This macro works as we expect, typing “Harry "Happy" Higgins” when executed.

Some other useful escape sequences are \r for return, \s for single quote, and \ (clipboard) for the contents of the clipboard.

You can even embed a variable using escape sequences. Here’s the same macro we made in the previous section, only this time it uses an escape sequence:

```
SetVariable name ask 'What is your name?'  
Message "\[name]"
```

How Do I Debug a Macro?

No matter how expert you become writing KeyQuencer macros, you will still have to debug them. Debugging can be challenging. You have to test your macro, note problems, and creatively solve them. Debugging requires patience, so try not to get frustrated. Debugging yields to patient effort. If you get hopelessly stuck, just call us. We’re happy to help. Here some tips:

Debugging Tips

- Never type a pound sign in a macro; it will always cause an error. Whenever you see a pound sign in a parameter substitute with a number.
- If you get an error trying to push a button, choose a menu, or click on an item, it’s probably due to bad timing. Try using the `WaitWindow` or `WaitApp` command.
- If you see the error message “Unloaded extension or syntax error” it usually means that you mistyped or messed up a command’s parameters somehow. Go over your macro carefully to find your error.
- Use the `Message` command to check the contents of variables or the clipboard.

Where to Go from Here

If you read and understood this chapter, you already know enough to write almost any macro you need. From here, you might want to become a KeyQuencer expert by learning about AppleScript and the KeyQuencer Batchter in *Chapter 7: Advanced Features*. You also might want to learn more about KeyQuencer’s macro language in *Chapter 8: Language Reference*.

7: Advanced Features

This chapter covers some of KeyQuencer's advanced features. It explains how to how to disable and enable KeyQuencer commands, control macro execution, how to use KeyQuencer's third-party links, how to use KeyQuencer with AppleScript, and how to use the KeyQuencer Batchter.

What Is a KeyQuencer Extension?

A KeyQuencer extension is a file loaded at startup time by the KeyQuencer Engine that expands KeyQuencer's functionality. Each command in KeyQuencer's macro language corresponds to a separate KeyQuencer Extension. There are many advantages to KeyQuencer's extensibility.

First, you can easily add new KeyQuencer commands — Just drop in a file and restart. Also, since each KeyQuencer extension takes up a small amount of memory, you can disable unused commands to free up resources. Finally, because KeyQuencer is available in a shareware version, shareware authors are constantly writing new extensions that you can use. Visit our website at **www.binarysoft.com** to download the latest KeyQuencer extensions.

Where Are the KeyQuencer Extensions?

- All KeyQuencer extensions are stored in subfolders, grouped by kind, in the folder named *KeyQuencer Extensions* in the root level of the active *System Folder*.

How Do I Add a New KeyQuencer Extension?

- Move the new extension to the *Third Party* folder in the *KeyQuencer Extensions* folder and restart your computer.

How Do I Disable a KeyQuencer Extension?

- Move the extension that you want to disable from it's enclosing folder in the *KeyQuencer Extensions* folder to the corresponding (*disabled*) folder and restart your computer.

Using KeyQuencer with AppleScript

KeyQuencer includes integrated support of AppleScript. You can run KeyQuencer macros from within AppleScript and use AppleScripts flow control to create more flexible macros. The rest of this chapter assumes a basic familiarity with AppleScript.

What AppleScript Commands Does KeyQuencer Support?

KeyQuencer's AppleScript link adds the following commands to AppleScript:

- `KeyQuencer Do Macro "macro text"` — Executes a KeyQuencer macro passed in as a string. Returns nonzero if error.
- `KeyQuencer is Busy` — Returns true if KeyQuencer is busy executing a macro.
- `KeyQuencer Macro Failed` - Returns true if the last macro failed to execute.
- `KeyQuencer Get Variable "varName"` - Returns the value of the KeyQuencer variable as a string.

- `KeyQuencer Set Variable "varName" to "value"` - Sets the specified KeyQuencer variable to the specified string.

In the *Macros* folder you'll find sample scripts that use all the above commands.

How Do I Run a Macro from AppleScript?

- Enclose the entire macro text in double quotes. Use `\` for double, `'` for single quotes, and two backslashes for escape sequences. For example:

```
KeyQuencer Do Macro "Message \"1st Line\\r2nd Line\""
```

How Do I Run an Active Macro from AppleScript?

- Precede the macro name with the `@` character, like this:

```
KeyQuencer Do Macro "@Open KeyQuencer Editor"
```

How Do I Control the Flow of Execution with AppleScript?

Use the `KeyQuencer is Busy` command to pause an AppleScript until a macro is done executing:

```
repeat while KeyQuencer is Busy
end repeat
```

Using KeyQuencer in Other Environments

You can embed a KeyQuencer Macro within a FileMaker Pro script, a 4th Dimension procedure, a Frontier script, a Cron script, or a HyperCard stack. You'll find all the links, plus documentation, inside the *Links* folder of the *KeyQuencer* folder. Have a look!

KeyQuencer Batcher

The KeyQuencer Batcher is an application that lets you batch process and selectively apply a macro to each of a set of files. Once you set up the macro that you want to use, you drag and drop files, folders, or volumes to the KeyQuencer Batcher to process them.

What does *batch process* mean? Suppose you are baking hundreds of cookies. Once you are done mixing the cookie dough (don't eat any!) you will batch process the cookies by baking them in the oven one cookie sheet at a time. The KeyQuencer Batcher (oven) will batch process files (cookies) by selectively applying a macro to each file (cooking them).

There are three separate macros to set up in the KeyQuencer Batcher: the Begin Batch macro, the Handle Item macro, and the End Batch macro. The Begin Batch macro is executed only once, before batch processing begins. The Handle Item macro is executed once for each file which is processed. The End Batch macro is executed only once, after all files have been processed.

The Handle Item macro defaults to:

```
Message "Item = \"[name]\"rPath = \"[pathname]\""
```

This macro displays a dialog with the name of the item being processed and its path. Notice that this macro uses the variables `name` and `pathname`. The KeyQuencer Batcher recognizes the following, predefined variables. You can use any of these variables in a KeyQuencer Handle Item macro (the value of the variables when the Begin Batch and End Batch macros are run is undefined):

Variable	Description
pathname	Pathname of the item. Example: “Macintosh HD:Documents:Item”.
name	Name of the item. Example: “Item”.
directory	Pathname of the folder enclosing the item. Example: “Macintosh HD:Documents”.
folder	Name of the folder enclosing the item. Example: “Documents”.
disk	Name of the disk that contains the item. Example: “Macintosh HD”.
type	File type of the item. Example: “TEXT”.
creator	Signature of the item. Example: “ttx”.
size	Size in bytes of the item. Example: “12168”
crDate	Date the item was created. Example: “04/08/96”
crTime	Time the item was created. Example: “12:46”
modDate	Date the item was last modified. Example: “04/15/96”
modTime	Time the item was last modified. “08:22”

All of the macro-related and batch-related preference settings are stored within the KeyQuencer Batcher application itself (including any batch list). This is so you can set up separate KeyQuencer Batchers for separate jobs (just duplicate the Batcher). Or, if you prefer, you can also export and import Batcher preference settings.

How Do I Configure the Macros in My Batcher?

1. Open the *KeyQuencer* folder and double-click *KeyQuencer Batcher*.
2. Open the **Macros** menu and choose either **Begin Batch...** (Command-B), **Handle Item...** (Command-M), or **End Batch...** (Command-E).
3. Click the **Macro Name** radio button and enter the name of an active macro or click the **Macro Text** radio button and enter the macro text to execute.
4. Click **OK**.

How Do I Set My Batcher’s Preferences?

- *To scan only top level folders when dropping folders onto the Batcher:* Open the **Batch** menu and choose **Scan Top Level Folders**. Only the top level contents of any folders will be batch processed. If this option is off, only the folders themselves will be processed; their contents are ignored.
- *To scan all nested folders when dropping folders onto the Batcher:* Open the **Batch** menu and choose **Scan Nested Folders**. The contents of all folders and subfolders will be batch processed.
- *To Batch Process the Files in the Batch List on Launch:* Open the **Batch** menu and choose **Start Batch List When Launched**.
- *To only process files of certain types or signatures:* Open the **Batch** menu and choose **Edit File Filters** (Command-F). Change the settings and click **OK**.
- *To export or import all the preference settings (including macros, and the batch list):* Open the **File** menu and choose **Export List & Prefs...** or **Import List & Prefs...**

How Do I Choose Files to Batch?

- Drag and drop any files on the KeyQuencer Batcher to process them. The KeyQuencer Batcher will process the files, scanning, or not scanning the contents of folders and subfolders according to your preference settings.
- Open the **Batch** menu and choose **Show Batch List** (Command-L). Type in the pathnames to the files that you want to batch. Or, open the **Edit** menu and choose **Insert Pathname...** (Command-I) to add a file's pathname to the batch list. When you're done adding files to the batch list, open the **File** menu and choose **Start Batch List** (Command-R).
- Open the **File** menu and choose **Start Batch...** (Command-O). Choose the file you want to batch and click **Open**. To choose a folder, select the folder and click **Select "[folder name]"**.

8: Language Reference

This chapter describes the KeyQuencer command language syntax in detail. It is designed as a quick reference for someone who is already somewhat familiar with the KeyQuencer macro language.

What Is a Macro Command Line?

A KeyQuencer command line begins with a command or comment and is terminated by a carriage return. Any number of blank spaces or tabs can precede a command or comment.

Commands are not case-sensitive, although headline style capitalization is recommended. Most KeyQuencer commands are followed by a series of parameters, separated by spaces. Parameters are also case insensitive and must be included on the same line as the KeyQuencer command. You can also include any number of tabs or spaces between parameters to make command lines easier to read.

You may not include non-breaking spaces (generated by pressing Option-Space Bar). Non-breaking spaces will generate a syntax error.

How Do I Interpret a Command's Syntax?

A command's syntax takes this format in the onscreen help of the KeyQuencer Editor:

```
CommandName parameter1 parameter2 [optionalParameter1]
```

Here's a specific example:

```
Play "Sound name"/([snd] #index)/res #id/beep/all
```

- `Play` is the KeyQuencer command.
- Slashes ("/") are used to show mutually exclusive parameters that can't be used together. In this example, you could use either "Sound name" or beep, but not both.
- Brackets ("[" and "]") are used to show optional parameters.
- Parenthesis (or brackets) indicate either groups of mutually exclusive parameters, or items that must be logically grouped together. In this example, the optional `[snd]` parameter is logically grouped with the `#index` parameter. Note that the parenthesis may be omitted for situations where a keyword is followed by a numeric parameter, as in `res #id` above; these items are always logically grouped together even if no parenthesis is used.

What Are the Different Types of Parameters?

- **Keywords.** A keyword is a specific word understood by KeyQuencer to be a parameter, for example, quiet, continue, or skip.
- **Strings.** Any string of characters enclosed by straight double quotes (""). The quotes must be typed in the macro. Curly double quotes (""") are not valid as string delimiters; they are considered part of the text string if inserted inside a quoted string (as in Message "These "quotes" can be used in a string.").
- **Four-character values.** A four-character value (also known as an OSType, signature, or creator) must be enclosed by straight single quotes ('). For example, 'MACS', 'TEXT'.

- **Single quoted strings.** Some commands, like `GetFileInfo`, accept single quoted format strings as parameters. Format strings may contain both text and formatting information. For example, a format string might look like this: `'Creator: $cr'`. Here, `Creator` is text and `$cr` is formatting information. Some commands also use single quotes to distinguish between different types of strings. For example the macro,


```
File rename "Macintosh HD" to 'Gary's Hard Drive'
```

 encloses the new file name `'Gary's Hard Drive'` in single quotes because the double quotes are already used to specify the original file to rename.
- **Numbers.** Short integers. In the onscreen command help, numbers are specified by a pound sign (“#”) followed by the parameter name. The pound sign is never actually typed in the macro, it is used only to signify that a parameter is a number. For example, `Command #num` might be entered in a macro as `Command 9`. Sometimes, numbers have some parameter text preceding them. For example, `Command num #num` would be entered in a macro as `Command num 9`.
- **Files.** Either a full path to a file or folder enclosed in double quotes, like: `"Disk:Folder:File"`, or a partial path preceded by a keyword, like: `sysdisk "Folder:File"`. File parameters appear in this chapter as `<file>` or `<item>`. Wherever `<file>` or `<item>` is listed as a parameter, the following syntax applies:

Parameter	Description
<code><item></code>	selected/([sysdisk/desktop/trash/system/panels/extensions/apple/preferences/startup] "Disk:folder:...:file")
<code>selected</code>	Acts on the current Finder selection.
<code>sysdisk</code>	Looks for the file in the startup disk (example: <code>sysdisk "Documents:My Numbers"</code>).
<code>desktop</code>	Looks for the file in the startup disk's Desktop folder.
<code>trash</code>	Looks for the file in the Trash.
<code>system</code>	Looks for the file in the active System Folder.
<code>panels</code>	Looks for the file in the Control Panels folder (example: <code>panels "KeyQuencer"</code>).
<code>extensions</code>	Looks for the file in the Extensions folder.
<code>apple</code>	Looks for the file in the Apple Menu folder (example: <code>apple "Chooser"</code>).
<code>preferences</code>	Looks for the file in the Preferences folder.
<code>startup</code>	Looks for the file in the Startup Items folder.
<code>"Disk:folder:...:file"</code>	Full or partial pathname of the item.

- **Variables.** Variable names are case insensitive and may not contain special characters or spaces in their names. Here are some valid names: `userName`, `line1`, `FILENAME` and `x`. Variables may contain any text string, for example `"hello world"`, `"42"`, or `"partial "Finder" quiet"`. KeyQuencer treats variables as if they were regular parameters. Whenever a variable is used, it must be preceded by a single character prefix that tells KeyQuencer how to interpret it as follows:

Prefix	Meaning
\$	Enclose variable value in double quotes. Used to pass a variable as a string or file parameter. Example: \$helloStr is interpreted as "hello world".
&	Enclose variable value in single quotes. Used to pass a variable as a four-character value or format string. Example: &helloStr is interpreted as 'hello world'.
%	Pass each word of the value without quotes. Used to pass a variable as one or more keywords or numerical parameters. Example: %helloStr is interpreted as hello world.

Variables can't be the first word in a command line. A variable passed with the “%” prefix can contain more than one word, quotes or comments. Variables which haven't been previously assigned (undefined variables) are treated as though they contain nothing (as empty variables). See the onscreen help text of `SetVariable` for more information.

- **Escape sequences.** Any single or double quoted string parameter may contain one or more escape sequences embedded within it. Escape sequences will be substituted for the equivalents at runtime. KeyQuencer interprets the characters immediately following a backslash character (“\”) in a quoted string as an escape sequence. The following are valid sequences:

Character Sequences

\r	return
\n	return
\q	double quote (")
\s	single quote (')
\t	tab
\\	backslash (\)
\6E	hexadecimal ASCII code 6E (6E is the “N” character). Valid ASCII codes are from 00 to FF.

Symbolic Sequences

\(clipboard)	The current text (up to 255 characters) of the clipboard.
\(date)	The current date in numeric format.
\(long date)	The current date in long format. For example “Monday, March 25, 1996”.
\(short date)	The current date in short format. For example “Mon, Mar 25, 1996”.
\(time)	The current time without seconds.
\(long time)	The current time with seconds.
\(day)	The current day number.
\(zero day)	The current day number with a leading zero, if necessary, for example “01” instead of “1”.
\(month)	The current month number.
\(zero month)	The current month number with a leading zero, if necessary, for example “01” instead of “1”.

<code>\(year)</code>	The current year as a four digit value.
<code>\(short year)</code>	The current year as a two digit value.
<code>\(hour)</code>	The current hour.
<code>\(zero hour)</code>	The current hour with a leading zero, if necessary, for example “01” instead of “1”.
<code>\(minute)</code>	The current minute.
<code>\(second)</code>	The current second.
<code>\(day name)</code>	The current day name.
<code>\(month name)</code>	The current month name.
<code>\(ticks)</code>	The number of ticks (one tick equals one sixtieth of a second) since the computer last started up.
<code>\(system time)</code>	A four-byte unsigned integer (which gets converted to a signed value) that equals the number of seconds since midnight, January 1, 1904 (the date of antiquity). Only for use by programmers.
<code>\(random)</code>	A random integer from -32768 to 32767.

Variable Sequences

`\[variable]` The current value of the named variable.

- **Error control keywords.** Many commands accept the `quiet` and `continue` parameters. They are used to determine how errors should be handled. If `quiet` is specified, errors encountered while executing that particular command line (runtime errors) will not be reported. If `continue` is specified, macro execution will continue even if an error occurs in that command line. If `continue` is not specified, macro execution will cease. You may specify either `quiet` or `continue`, neither, or both.

Comments

You use two hyphens to indicate that the text that follows, up to the carriage return, should be treated as a comment (ignored). You can start a comment at the beginning or at the end of a command line, for example:

```
--here's a comment
SwitchApp 'Finder'    --here's another comment
```

KeyQuencer Command Descriptions

Descriptions of each KeyQuencer command can be found in the onscreen help of the KeyQuencer Editor. Sample macros for each command can be found in the macro suitcase named *Command Samples* located in the Macros folder (in the *KeyQuencer* folder).

How Do I Create a Printable Text File of All KeyQuencer Commands?

1. Open the KeyQuencer Editor.
2. Open the **File** menu and choose **Export Help Text...**
3. Choose a location and click **Save**.

Appendix A: KeyQuencer License

WARNING

READ THIS LICENSE AGREEMENT BEFORE INSTALLING THE SOFTWARE. BINARY SOFTWARE IS WILLING TO LICENSE THE ENCLOSED SOFTWARE TO YOU ONLY UPON THE CONDITION THAT YOU ACCEPT ALL OF THE TERMS CONTAINED IN THIS LICENSE AGREEMENT. BY INSTALLING THE SOFTWARE, YOU WILL INDICATE YOUR ASSENT TO AND AGREEMENT TO BE BOUND BY THE TERMS OF THIS LICENSE AGREEMENT. IF YOU DO NOT AGREE TO THESE TERMS, THEN BINARY SOFTWARE IS UNWILLING TO LICENSE THE SOFTWARE TO YOU, IN WHICH EVENT YOU SHOULD RETURN THE PACKAGE, TOGETHER WITH ALL ITS ACCOMPANYING MATERIALS AND PROOF OF PAYMENT, TO THE PLACE FROM WHICH IT WAS ACQUIRED FOR A FULL REFUND.

License

You are permitted to use KeyQuencer, version 2.5 (the "Software") on any single computer, provided the Software is in use on only one computer at a time. "Use" means loading the Software into memory or installing the Software onto the hard disk or any other storage device of that computer.

You are not permitted to rent, lease, lend, sublicense or otherwise temporarily transfer the right to use the Software or the user documentation accompanying the Software (the "User Documentation"). You may permanently transfer all of your rights to use the Software to another person, provided that you at the same time transfer all copies of the Software (or destroy all copies not transferred) and the User Documentation.

You are not permitted to reverse engineer, decompile, disassemble, or otherwise attempt to render source code or file format from the Software.

Your right to use the Software and the User Documentation terminates automatically if you fail to comply with the terms of this License.

Copyright Protection

The Software is owned by Binary Software, and is protected by various laws, including United States copyright law and international copyright treaties. Therefore, you must treat the Software like any other copyrighted material, except that you are permitted to make archival copies of the Software for backup purposes, provided that you destroy all archival copies in the event you permanently transfer possession of the Software, or if your continued possession of the Software ceases to be rightful. You may not copy the User Documentation.

Limited Warranty

Binary Software warrants that the media on which the Software is furnished will be free of defects in materials and workmanship for a period of one (1) year from the date of receipt. BINARY SOFTWARE DISCLAIMS ALL OTHER WARRANTIES WITH RESPECT TO THE SOFTWARE, USER DOCUMENTATION AND MEDIA, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. Some states do not allow the exclusion or limitation of implied warranties, so these exceptions and limitations may not apply to you. This warranty gives you specific legal rights, and you may have others which vary from state to state.

Remedies

Binary Software's entire liability and your exclusive remedy for breach of this Limited Warranty shall be, at Binary Software's Option, either (1) refund of the price paid, or (2) replacement of the defective media and the User Documentation. Under no circumstances shall Binary Software be liable to you for any claim for damages (including special, incidental, or consequential damages) resulting from loss of data, business interruption, or other similar event, even if Binary Software has been advised of the possibility of such damages.

U. S. Government Restricted Rights

The Software and User Documentation are licensed to the U. S. Government with RESTRICTED RIGHTS. Use, duplication, or disclosure by the U. S. Government is subject to restrictions as set forth in the Rights in Technical Data and Computer Software clause at DFARS 252.227-7013 and the Commercial Computer Software clause at 48 CFR 52.227-19, as applicable.

Contractor/manufacturer is Binary Software, Inc., 2118 Wilshire Boulevard, Suite 900, Santa Monica, California 90403.

General

This License shall be governed by the laws of the State of California. This License may only be modified in writing signed by you and Binary Software.

Please direct questions concerning this License, in writing, to Binary Software, Inc., 2118 Wilshire Boulevard, Suite 900, Santa Monica, California 90403.

Appendix B: KeyQuencer Lite License

NOTICE TO USER: THIS IS A CONTRACT. BY INSTALLING THIS SOFTWARE YOU ACCEPT ALL THE TERMS AND CONDITIONS OF THIS AGREEMENT. IF YOU ARE UNWILLING TO AGREE TO THESE TERMS, THEN BINARY SOFTWARE IS UNWILLING TO LICENSE THE SOFTWARE TO YOU.

This Binary Software Incorporated ("Binary") License Agreement accompanies a Binary software product and related explanatory written materials ("Software"). The term "Software" shall also include any upgrades, modified versions or updates of the Software licensed to you by Binary. This copy of the Software is licensed to you as the end user. Please read this Agreement carefully.

YOU ARE PERMITTED TO EVALUATE THE SOFTWARE, FREE OF CHARGE FOR 30 DAYS. YOU MUST PAY THE REGISTRATION FEE TO BINARY SOFTWARE TO RIGHTFULLY USE THE SOFTWARE AFTER THE 30-DAY EVALUATION PERIOD. Provided you pay, Binary grants to you a nonexclusive license to use the Software, provided that you agree to the following:

Use of the Software

You may install the Software in a single location on a hard disk or other storage device and make backup copies of the Software.

Copyright

The Software is owned by Binary Software, and is protected by various laws, including United States copyright law and international copyright treaties. Therefore, you must treat the Software like any other copyrighted material, except that you are permitted to make archival copies of the Software for backup purposes, provided that you destroy all archival copies in the event you permanently transfer possession of the Software, or if your continued possession of the Software ceases to be rightful. You may not copy the User Documentation. You agree not to modify, adapt, translate, reverse engineer, decompile, disassemble or otherwise attempt to discover the source code or file format of the Software. This Agreement does not grant you any intellectual property rights in the Software.

Transfer

You may not rent, lease, or sublicense the Software. You may, however, transfer all your rights to use the Software to another person or entity, provided that you transfer this Agreement with the Software.

No Warranty

The Software is being delivered to you AS IS and Binary makes no warranty as to its use or performance. BINARY AND ITS SUPPLIERS DO NOT AND CANNOT WARRANT THE PERFORMANCE OR RESULTS YOU MAY OBTAIN BY USING THE SOFTWARE OR DOCUMENTATION. BINARY AND ITS SUPPLIERS MAKE NO WARRANTIES, EXPRESS OR IMPLIED, AS TO NONINFRINGEMENT OF THIRD PARTY RIGHTS, MERCHANTABILITY, OR FITNESS FOR ANY PARTICULAR PURPOSE. IN NO EVENT WILL BINARY OR ITS SUPPLIERS BE LIABLE TO YOU FOR ANY CONSEQUENTIAL, INCIDENTAL OR SPECIAL DAMAGES, INCLUDING ANY LOST PROFITS OR LOST SAVINGS, EVEN IF A BINARY REPRESENTATIVE HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES, OR FOR ANY CLAIM BY ANY THIRD PARTY. Some states or jurisdictions do not allow the exclusion or limitation of incidental, consequential or special damages, or the exclusion of implied warranties or limitations on how long an implied warranty may last, so the above limitations may not apply to you.

Governing Law and General Provisions

This Agreement will be governed by the laws of the State of California, U.S.A., excluding the application of its conflicts of law rules. This Agreement will not be governed by the United Nations Convention on Contracts for the International Sale of Goods, the application of which is expressly excluded. If any part of this Agreement is found void and unenforceable, it will not affect the validity of the balance of the Agreement, which shall remain valid and enforceable according to its terms. You agree that the Software will not be shipped, transferred or exported into any country or used in any manner prohibited by the United States Export Administration Act or any other export laws, restrictions or regulations. This Agreement shall automatically terminate upon failure by you to comply with its terms. This Agreement may only be modified in writing signed by an authorized officer of Binary.

Notice to Government End Users

If this product is acquired under the terms of a: GSA contract-Use, reproduction or disclosure is subject to the restrictions set forth in the applicable ADP Schedule contract; DoD contract-Use, duplication or disclosure by the Government is subject to restrictions as set forth in subparagraph (c) (1) (ii) of 252.227-7013; Civilian agency contract-Use, reproduction, or disclosure is subject to 252.227-19 (a) through (d) and restrictions set forth in the accompanying end user agreement.

Unpublished-rights reserved under the copyright laws of the United States. Binary Software Incorporated, 2118 Wilshire Blvd 900, Santa Monica, CA 90403.

Index

A

aborting execution of a macro 15
active macros. *See* macro
AppleScript 45–46
application signature. *See* signature, application

B

backing up your macros 37
Batcher, KeyQuencer 46–48
Binary Software 9
button macro, creating 22–24

C

click macro, creating 25–27
commands, KeyQuencer
 explained 40
 in Commands Palette 34–35
 syntax 49
Commands Palette 34–35
comments
 inserting in a macro 39, 42–43
 use of 52

D

drag macro, creating 27–31

E

Editor, KeyQuencer
 customizing appearance of 38
 executing macro with 32
 opening 16
Editor Suitcase 35–36
escape sequences 43, 51–52

F

FileMaker Pro 46
font, changing size of in Editor 38
font, display
 changing in Commands Palette 34

changing in Editor 38
4th Dimension 46

H

help
 contacting Binary Software 9
 onscreen 35

I

importing macros 37
Installed Macros suitcase 35, 36
installing
 description of files installed 11–12
 KeyQuencer 11

K

key macro, creating 30–31
KeyQuencer Batchter. *See* Batchter, KeyQuencer
KeyQuencer commands. *See* commands, KeyQuencer
KeyQuencer Editor. *See* Editor, KeyQuencer
 Opening 16–17
KeyQuencer Launcher. *See* Launcher, KeyQuencer
KeyQuencer Panel 17. *See also* Panel, KeyQuencer
keystroke
 definition of 16
 modifying 33
keystroke, creating a macro to press 30–31
keywords 49

L

Launcher, KeyQuencer 15
Launcher Suitcase 35–36
 creating 36
launching applications, creating a macro to 18–20

M

macro
 copying a sample macro 37
 creating 17–31, 32–33
 definition of 8, 13
 deleting 33
 renaming 33
 suitcases, defined 35–36

- macro name
 - definition of 16
- macro suitcase. *See* suitcase, macro
- macros
 - backing up 37
 - replacing active macros with new suitcase 37
- menu macro, creating 20–22
- mouse click or drag, making a macro to 25–31

O

- open macro, creating 18–20

P

- parameters
 - explained 40
 - types of parameters 49–52
- pathname
 - inserting in a macro 34, 39–55
 - use of as parameter 50
- printing list of macros 39

Q

- QuickMacro list 13

S

- signature, application
 - inserting in a macro 34, 39
 - use of as parameter 49, 50
- Stand Alone Applet
 - creating 36
- Stand-Alone Applet 36–37
- strings 49, 50
- suitcase, macro 35–37
 - changing type of 37
 - defined 35–36
 - Editor Suitcase 35–36, 36
 - importing macros into 37
 - Installed Macros suitcase 35, 36
 - Launcher Suitcase 35–36, 36
 - printing list of 39
 - Stand-Alone Applet 36, 36–37
- syntax, command 49
- system requirements 9

T

text macro, creating 17–18

type text, creating a macro to 17–18

V

variables 43, 50–51