Quicksilver User Manual

by Howard Melman

Version 0.15 (for Quicksilver version β54)

Jan 10, 2009

Please provide feedback on this manual at the Quicksilver forum in this thread:
http://groups.google.com/group/blacktree-quicksilver/browse_thread/thread/ef99bcac45de5ca7#

© Copyright 2006-2008 Howard Melman

This work is licensed under the Creative Commons Attribution-NonCommercial-ShareAlike 3.0 Unported License. To view a copy of this license, or send a letter to the following address or visit http://creativecommons.org/licenses/by-nc-sa/3.0/

Creative Commons
543 Howard Street, 5th Floor
San Francisco, California, 94105
USA

Table of Contents

How To Read this Manual 5

Part I - Introduction *6*

Introduction 6
Concepts and Terminology 7
Installation 10
Invoking Quicksilver 12
Activating Quicksilver ... 12
Selecting Items ... 13
Combining Activation and Selection ... 14
Immediate Execution ... 15
The Comma Trick ... 15
Helping the Matching Algorithm ... 16
TextMate Ranker ... 17

Part II - Configuration *18*

Configuration 18
Preferences ... 19
Interfaces ... 23
Plug-ins ... 26
Catalog ... 27
Actions ... 31
Handlers ... 32
Notifications ... 32
Triggers 34
Trigger Basics ... 34
HotKey Triggers ... 36
Proxy Objects ... 38
More on Triggers ... 40
Using the Mouse 41
Drag and Drop ... 41
Mouse Triggers ... 42
Abracadabra Mouse Gestures ... 43
Constellation Radial Menus ... 44
Command Objects and Droplets ... 47

Part III - Features *49*

Files and Folders 49
Basic File Actions ... 51
File Tagging and Spotlight ... 54
File Compression and Disk Images ... 56
Sending Files ... 57
File Transfer Programs ... 58
Path Finder ... 59
Deviant ... 59
Applications and Processes 59
Basic Application Actions ... 60
Advanced Process Actions ... 62
Dynamically Accessing a Menu ... 63
iTunes 64
Browsing ... 65
Controlling iTunes and Triggers ... 67
Preferences ... 67
Contacts 68
Address Book ... 68
Entourage ... 72
Now Contact ... 73
BuddyPop ... 73
Mail 74
Mail.app ... 76
Entourage ... 77
Gmail ... 77
Other Mail Applications ... 78
Teleflip ... 78
Chat 79
iChat ... 80
Adium ... 80
Calendar 81
iCal ... 81
Google Calendar Module ... 83
Web 83
Web Browsers ... 84
Online Services ... 86
Web Pages ... 89
Web Searches ... 90
Adding Web Searches Manually ... 94
NetNewsWire ... 96
Text 96
Basic Text Actions ... 96
Dictionary ... 99
Spotlight ... 100
Text Files ... 100
Specially Formatted Text ... 102
Clipboard and Shelf 103
Clipboard History ... 103
Clipboard Storage ... 105
Shelf ... 105
Comparing Clipboard History, Storage and Shelf ... 106
Note Taking Applications 107
VoodooPad ... 107
Yojimbo ... 108
WikityWidget ... 109
Images 109
iPhoto ... 109
Slideshow ... 110
Desktop Picture Action ... 110
Screen Capture ... 110
Image Manipulation ... 110
Flickr Upload ... 111

System **111**
Devices....111
Networking....113
Configuration.... 113
Exposé....114
Extra Scripts.... 115
Development **116**
Automator.... 116
Services....116
Terminal.... 116
CLIX....118
Command Line Tool.... 118
Developer Documentation.... 119
Subversion....119
Eclipse.... 120
Remote Desktop....120

Part IV - Appendix ***120***
FAQ **120**
Leopard Specific Things.... 122
Troubleshooting **123**
Extending Quicksilver **124**
Random Notes **124**
Results List.... 124

How To Read this Manual

Quicksilver's strength is that it integrates with so many different aspects of your Mac. Rather than cover each of the 100 or so plug-ins in some arbitrary order (e.g., alphabetically), the bulk of this manual is organized around topics such as working with files, contacts, email, or web browsers (roughly the Plug-in Preferences categories). Grouping similar plug-in information together avoids duplication and makes it easier to compare capabilities.

Each section includes general usage and configuration instructions, as well as troubleshooting hints and tips for using features more efficiently using advanced Quicksilver features. New users should be able to start using the features quickly and experienced users will probably find something new about many features. The following sections should help in finding information about a topic whether basic or advanced.

		New Users	Experienced Users
Part I	Introduction, installation, basic usage	Read thoroughly	Skim
Part II	Preferences and configuration	Skim, refer back to when needed	Read thoroughly
Part III	Plug-ins grouped by categories	Read sections of interest. Look up unfamiliar things in Part I and II	Read sections of interest
Part IV	Troubleshooting and extending	Read as needed	Read as needed

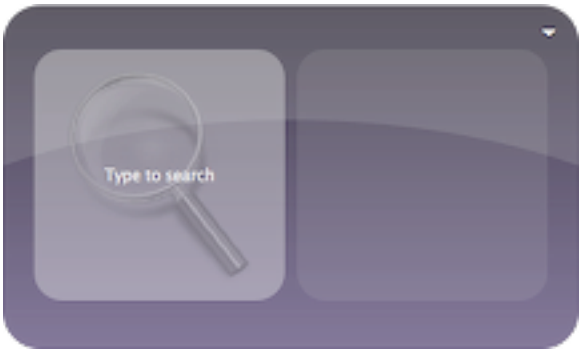
Part I - Introduction

Introduction

When I want to do something on my Mac, my first reaction is to use Quicksilver. Whether I want to send an email, search for something on the web, open a bookmark, open a file, start an application, send an IM, revisit a web page I looked at yesterday, pause iTunes or find a song, or virtually anything else, I type ^-space and activate Quicksilver to do it. That makes Quicksilver very powerful, but also a little difficult to explain. I'll use an example of sending email to my friend Ashish.

First, I could open Mail, type ⌘N to open a new message, type enough of `Ashish` to have her address appear in the To: field and then `tab` to the Subject and continuing writing the message. I could instead open Address Book, search for Ashish's card, and ^-click (or right-click) on the email address and choose Send Email.

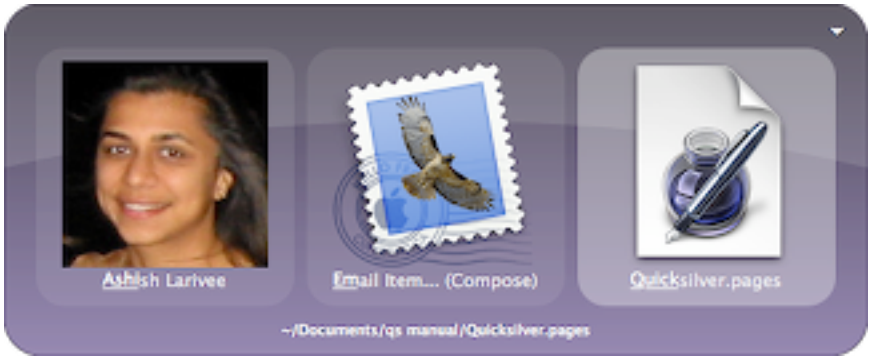
This is how I do it using Quicksilver. I type ^-space, `A`, `Return`. That's it. Let's walk through it. ^-space at any time, in any application, activates Quicksilver bringing up a two pane window shown here.



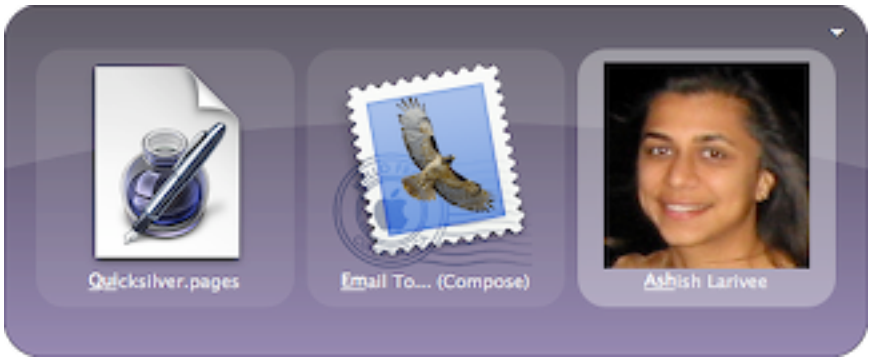
When I type `A`, Ashish appears in the first pane because I often send her email. Her picture appears because I have her picture in her Address Book entry (mostly because Quicksilver makes such good use of it). Also another window appears with other choices. If I kept typing, those would change to be some other choice, but since Ashish is what I want, I'm done. The second pane shows the **Compose Email** action which is what I want to do. Again this appears because it's the most common thing I do when I call up Ashish. Typing `return` performs the selected action so I see a new Compose Message window appear from Mail.app with Ashish's address filled in.

You might think using Quicksilver doesn't seem that much easier than the other methods I described, but the Quicksilver method is done entirely via keystrokes. There's no mousing to the Dock to open a particular icon or having to select a specific field, just type 3 keystrokes (one shifted). You might also think the comparison is unfair because I said some of the choices appeared so easily since I do this often, but that's one of the advantages of Quicksilver. It learns

what you do and makes your most frequent tasks easier. The other methods don't learn and don't get any easier.



Now let's say I wanted to send a document to Ashish. Instead of choosing the default **Compose Email** action, I can `tab` to the second pane, type `email i` to choose **Email item...(Compose)**, this opens a third pane which I `tab` to and type `~/Q` to choose the document in my home directory. Now the message window is opened with the attachment all set up. Say Ashish was expecting this document and I didn't need to include any text in the message, just the attachment. I could choose the action **Email item...(Send)** and then the message window does not open but the message is sent in the background without ever disturbing what I was doing before. The subject is set to the name of the attachment and a short sentence says the file is attached to the message.



The above works if I'm thinking, "I want to send Ashish this document". Say instead I thought "I need to send this document to Ashish". I can do this as well. Select the document in the first pane, choose the **Email to...(Compose)** action and then choose Ashish. Say she is online in iChat. There are actions to open a chat window with her or to send the file to her via iChat. I find these options the real strength of Quicksilver. It lets me easily do what I want, how I want. I don't need to change my thinking to how Quicksilver wants me to do things, and it learns from me and gets easier to use over time. The amazing thing about Quicksilver is how flexible it is. Via a wide variety of plug-ins, Quicksilver can select just about anything on your Mac as an object and do potentially hundreds of different things to it. Of course, it can also do a lot more, get used to reading that.

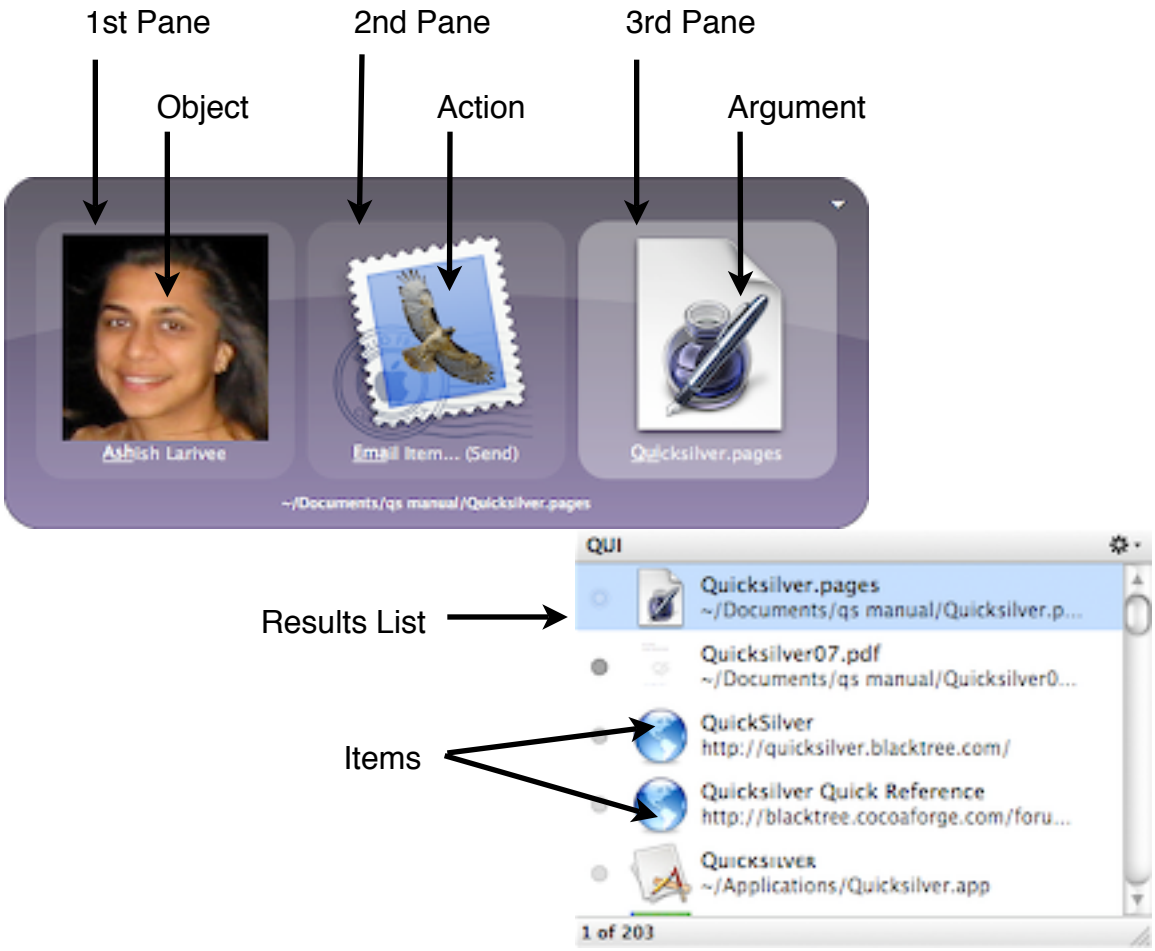
Concepts and Terminology

Quicksilver is a very modular application. This manual is organized around the things that Quicksilver can manipulate such as files, text, music, etc. Various Quicksilver concepts and facilities are introduced here so that the explanations in each following section can make use of them. A term introduced and defined is written like this.

To the right are the modifier key abbreviations used. Combinations are achieved by holding down one or more modifier keys and typing another key such as a letter, number or punctuation character. E.g., ⌘;, ⌥⌘S, etc. The arrow keys are shown as: →, ←, ↑, and ↓.

⌘	Command
⌥	Option
^	Control
⇧	Shift

Quicksilver runs as an application in the background at all times. Typically there is no menu, dock icon, menu bar icon or other indication of Quicksilver on screen until you activate it with the hot key (by default ^-space), though there are preferences to change this. Preferences are available by typing ⌘, when Quicksilver is activated.



Quicksilver commands are entered via two or three panes containing respectively an object, an action and if a third pane is needed, an argument. These are the terms used in this manual, but unfortunately other terms are used in some places in Quicksilver and in the forums and other articles. The primer interface labels the panes Subject, Action, and Object. Some forum posts use Subject, Verb, and Object; and others use Direct Object, Verb, and Indirect Object. This manual uses the terms in the diagram above. As you type into one of the panes, Quicksilver determines what items match and puts the top choice in the pane and additional, lower ranked matching items are shown in the results list. This happens for each pane so the results list will contain objects, actions, or arguments depending on which pane is selected. The results list above is showing the possible arguments that match the entered text `Qui`. The term item is used for something appearing in the results list, regardless of whether they are objects, actions or arguments.

Plug-ins are optional modules that are installed which can add objects, actions or other capabilities to Quicksilver. You manage plug-ins entirely from within Quicksilver Preferences including finding, installing, updating, enabling, configuring and removing them.

The Catalog is the list of objects that you can directly bring up in Quicksilver. Objects are not entered individually, instead you configure sources (in the Catalog Preferences) and Quicksilver scans and indexes them at regular intervals (by default every 10 minutes). For example, there is a source for Safari and from it, all your bookmarks and history are indexed into the catalog. Each of your bookmarks are objects to Quicksilver. So are your files and folders from your home directory, your contacts from Address Book, all the apps from your Applications directory, your albums from iPhoto, the songs from iTunes, and many other things (provided the appropriate plug-ins are installed). Note that some plug-ins allow Quicksilver to index things stored on web servers (such as del.icio.us bookmarks and backpack pages) and not merely things on the hard drive.

A Preference pane shows all the actions that can be performed. All actions work on an object and actions are allowed based on the type of object selected. E.g., the **Open URL** action is only available for objects that are URLs. Actions that require an argument typically end in "…" and arguments are expected to be of a certain type. E.g., the **Email to…** actions expect the argument to be an email address or contact.

Some actions have a complementary action that reverse the object and argument type. E.g., you can enter the command "*file*, **Email To…(Compose)**, *address*" or the command "*address*, **Email Item… (Compose)**, *file*". Notice the names of the actions are slightly different (To vs Item). Many (but not all) action names hint at the type of argument they take. **Email To….** wants an address to follow. **Email Item…** wants some kind of item to send (text or a file). These email actions are so similarly named you might not have noticed they are different. In other cases actions are so differently named you might not notice they are related. E.g., a pair of actions is used for searching web sites (like Google). You can enter a command as "*site*, **Search For…**, *query*" or "*query*, **Find With…**, *site*". Complementary action pairs allow Quicksilver to adapt to how you think of commands. Then there are commands like **Make New…** which have no complement and don't hint at what their argument type is. Hopefully this manual helps explain them.

Interfaces are configurable skins that affect the appearance of Quicksilver. They are installed as plug-ins and can be configured from the Appearance Preference pane. The default interface is called Primer. A very popular interface called Bezel is used throughout this manual.

When you use Quicksilver you type to select the objects, actions and arguments of commands. As you type, your choices are filtered down from all possible choices to only those that match what you've typed (which are shown in the results list). The matching method is one of the strengths of Quicksilver. Matches can be by the beginning of the phrase (e.g., type `Deskt` for "Desktop") or by the first letter of each word of the choice (e.g., `BA` for "Browse Artists" which is an iTunes object). The more you use specific items (objects and actions) the more quickly they will be matched. After a few times, once you type `Des` Quicksilver will start guessing "Desktop" and if you do it enough, you might be able to choose it just by typing `D`.

Some items have sub-items, e.g., folders have files, contacts have phone numbers and addresses, musical artists have albums, etc. Usually when an item has sub-items you'll see a > on the right side of the item in the results list. The way you select the sub-items is by typing → or `/`. The object then changes to the top sub-item and the list shows the sub-item's peers. You can type to match one of them or use the arrows to scroll up and down. This is an easy way to navigate folder hierarchies, or any other hierarchies.

Quicksilver's strength is that it can do so many different things and the matching algorithm and navigation techniques make getting at all those things easy. However, for those commands that you do all the time, there's an even faster way. You can configure triggers to execute specific commands when you type a HotKey or make a mouse gesture (a click, drag, moving to a corner of the screen, and various other things). Have a bookmark or a search you do all the time? Triggers are perfect for them. There are built-in triggers to control iTunes. Triggers can be invoked at anytime, without having to first activate Quicksilver (with ^-space).

There is a type of object known as a proxy object which stands for other objects. E.g., the Current Web Page proxy object stands for the URL of the web page currently shown in Safari (though it should be the default browser). The Current Selection proxy object stands for the current selection in the currently running application. Proxy Objects allow triggers to become dynamic, e.g., pasting the URL of the web page currently shown in the browser.

Installation

Download Quicksilver from the web site http://quicksilver.blacktree.com/ or directly from http://getqs.com/. If you're adventurous you can download pre-release development builds from http://getqs.com/dev/. Open the .dmg file and it will mount the Quicksilver volume and automatically extract the Quicksilver.app. Drag it where you want, I put mine in ~/Applications/.

Double-click Quicksilver.app to start it. When run the first time a setup process is run. You can rerun it later with the Run Setup button from the Application Preferences. After a welcome screen and a license screen, Quicksilver will scan your computer for applications and based on

what applications it finds, will recommend plug-ins to install. Check the plug-ins for all the applications you use regularly and click Continue. Next you can choose the HotKey to activate the Quicksilver command window and how often Quicksilver will scan your computer to keep its catalog updated (the default is every 10 minutes and is a fine choice). This is followed by some basic usage instructions and then the setup closes and a Quicksilver command window (using the primer interface) appears.

In spite of the instructions, first time users might be confused. While Quicksilver is running, there is no menu for it, no Dock icon and no indication it's there aside from the command window. If you close it and didn't read the instructions you might be a little lost.

The Quicksilver.app is usually in /Applications/ or ~/Applications/. When first used, the following per user files and folders are created:

- ~/Library/Application Support/Quicksilver/
  - Actions.plist - list of installed actions
  - Catalog.plist - the configured catalog sources
  - Mnemonics.plist - learned inputs, defaults and abbreviations
  - PlugIns.plist - the list of available plug-ins and how they are configured
  - Triggers.plist - the configured triggers
  - Caches/ - another cache folder
  - Indexes/ - folder of indexed items
  - PlugIns/ - installed plug-ins
  - Shelves/ - where items on the Shelf and clipboards are stored
  - Actions/ - not installed but create this folder to add scripts that implement actions
  - Templates/ - not installed but create this folder to use with the **Make New…** action
- ~/Library/Preferences/com.blacktree.Quicksilver.plist - stores various preferences and internal state
- ~/Library/Caches/Quicksilver/ - stores various state in binary files

To do a complete uninstall use the Uninstall Quicksilver button in the Application preferences, or remove the above files and folders. It can also be useful to move or rename these while trouble-

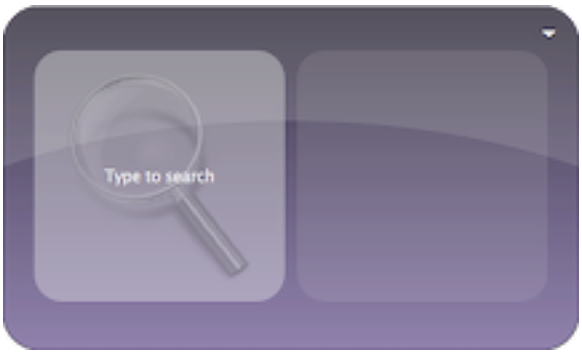
shooting a problem. If you are having trouble installing plug-ins check that the owner of ~/Library/Application Support/Quicksilver/ is the current user and not System. If it's not, change it and restart Quicksilver.

When Quicksilver starts it contacts hrwebservices.net to check for new versions. If you use something that monitors outgoing connections (e.g., Little Snitch) it might warn you about this. It's perfectly normal and benign. If you're curious hrwebservices, is HostRocket Web Services which hosts blacktree.com.

Invoking Quicksilver

Activating Quicksilver

Now that Quicksilver is installed and running, how do you use it? The basic way is to activate the command window with the HotKey ^-space. When you type this the command window appears, the first pane is active and you can type to select some item in your catalog.



The HotKey key is configurable in the Command Preferences Pane under HotKey activation. Quicksilver defaulted to ⌘-space for a long time and then in 10.4, Apple chose to use ⌘-space for Spotllght. So Quicksilver changed its default to ^-space. If you look at older posts on the Web you'll often see ⌘-space used to activate Quicksilver. Since I use Quicksilver a lot more often than Spotlight (and anything without ⌘ conflicts with Emacs), I use ⌘-space for Quicksilver and ⌥⌘-space for Spotlight (set in the Spotlight System Preference pane of OS X). Another advantage is that many hot keys used after activating Quicksilver use ⌘, so using the same modifier key is easier. For example, ⌘, will open up Quicksilver's Preferences (as it will in most applications). I find it easier to type ⌘-space ⌘, than ^-space ⌘, because I don't have to change modifier keys.

Selecting Items

Once you activate Quicksilver with ^-space, you type some characters of what you're looking for and it will appear in the first pane. If I'm looking for the Address Book entry for my friend Ashish Larivee typing any of the following might select it: `ashish`, `ash`, `al`, `alari`, or many other combinations of the letters. In fact Quicksilver keeps track of what you type and what you select and uses that past history to guess what you're looking for now. This means if I type `a` and pick Ashish's entry Quicksilver will start to guess that more often for `a` rather than "Address Book" or "Adium" or other things that begin with "a". Quicksilver uses what's known as a matching algorithm to order its choices, below it's discussed in some detail including what you can do to help it guess more accurately. Note it's only used for items, actions are chosen from a fixed ranking you can configure in the Actions preferences.

In order to keep the top level catalog manageable, you cannot select any item merely by typing after activating Quicksilver (this is known as "at the top level"). E.g, you can't reach a song title from your music or the phone number of a contact. That's not to say Quicksilver can't get to these, you must navigate to these items by typing → (or equivalently /) into the appropriate object (in these examples iTunes and the contact). Most items that you can → *into* have a > at the far right in the results list.

By default Quicksilver will show its other guesses for what you typed after a few seconds, you can then scroll through them by typing ↓. This behavior can be changed in the Command Preferences under "Show other matches" where you can choose: Never, Immediately, or Delayed (the default). You can also configure the spacebar to behave like ↓. Of course the ↑ keys works too but you can also navigate this list with ^N (next) and ^P (previous) and ^V (scroll down a screen) and ⌥V (scroll up a screen).

You might not have realized this but Quicksilver shows what matches what you typed in two different ways. When first activated, Quicksilver will filter the whole catalog down to just the things that match what you typed. To be clear, the things that don't match what you typed, are removed from the results list. However if at some point you type → or / to "go into" an object, if you type you'll find that Quicksilver moves to (snaps to) the first item that matches what you typed alphabetically. The non-matching items are not removed. This is most apparent when browsing the file system or your iTunes library. It's also evident when browsing a contact's information, if you type to find their work phone number, the other information is still displayed in the results list (which is convenient). Sometimes these behaviors are known as selection as opposed to narrowing. In a results list there's a gear menu in the top right of the window, from there you can choose these behaviors via the Search Modes sub-menu:

- **Filter Results** - Filters the current results list.
- **Filter Catalog** - Filters, but also includes the entire contents of the top-level catalog. Lasts until you type `esc` or activate Quicksilver again (`^-space`).

- **Snap to Best** - scrolls the results list to the best match but doesn't remove non-matching items.

If you mistype something, you can type the `delete` key or ⌘X to clear the pane and start over. If you navigate to something via → or / then these keys will still keep you in the current position but clear the search. To reset entirely so that you're using the top-level global catalog use the `esc` key. There's no way to delete just the last letter typed.

In the Command preferences you can configure the "Reset search after" time. This is how long between keystrokes Quicksilver waits before reseting the search. So if it's set to .85 seconds and it's been 1 second since you've typed a key, if you type another one it will start the search over with that key.

Combining Activation and Selection

There are faster ways to use Quicksilver too. You can activate it and put something in one or more panes at the same time. Triggers let you define key sequences, mouse clicks, and mouse gestures that will activate Quicksilver and fill in some panes and even execute commands immediately. These are discussed in more detail below but it's worth describing two triggers now. If you look in the Triggers Preferences under Quicksilver you'll find two that have been predefined **Command Window with Selection** and **Command Window in Text Mode**. The first opens a command window with the current application's selection in the first pane. This is typically text, though for the Finder this is one or more selected files or folders. Some other applications like iPhoto and iTunes also allow their non-text selection to be brought into Quicksilver. The second trigger will bring the selection into the first pane but will put that pane into text-mode so you can edit it.

Quicksilver installs two services available in other applications (look in the application's menu under Services and you'll see Quicksilver). The two services are **Get Current Selection (Internal)** and **Send to Quicksilver**. The fact that there are two of them is a quirk of the implementation, you can use either, but **Send to Quicksilver** is the one meant to be used from the menu. It activates Quicksilver and puts the application's current selection (typically text) into the first pane. The other service is the internal mechanism that enables ⌘-esc and the Current Selection proxy object (used in triggers). These basically accomplish the same thing as **Send to Quicksilver**. Apple's Front Row uses ⌘-esc and this conflicts with the Quicksilver service. You can change the Quicksilver key binding in Quicksilver's Preferences, under Command, using the Alternate Services Menu Key setting. Or if you prefer, the Front Row shortcut can be changed in the Keyboard & Mouse system preferences in the Keyboard Shortcuts tab. Even if you uncheck it, Front Row still works with the Apple Remote. I have the **Command Window with Selection** trigger mentioned above assigned to ⌘-esc. TODO: It's not clear if this is needed but if ⌘-esc isn't working for you, make sure you have enabled advanced features and in the Catalog under Quicksilver check that Internal Commands is enabled.

Yet another way to put the selection into the object pane is to activate Quicksilver and then type ⌘G. This is slower than the other methods since you have to activate Quicksilver first, but since I learned it first, it's in my fingers, so I use it a lot. By default ⌘G only puts the Finder's selection into the first pane. If you have advanced features enabled in the Application Preferences, then in the Extras Preferences checking "Pull selection from front application instead of Finder" will make ⌘G behave just like ⌘-esc, using the same mechanism.

If Quicksilver is already running, clicking on the Quicksilver.app icon will bring up the command window. If you drag and drop something onto the icon it will be selected in the first pane. This is convenient if Quicksilver is in your Dock, but people also put it in the Finder's sidebar or toolbar.

This is particularly useful if you hide your Dock and use Finder windows often (I find the more I use Quicksilver, the less I use the Finder).

There are other, less obvious ways to select something in a Quicksilver command window pane. As discussed above you can type ⌘G to choose the Finder's selection (or possibly the selection of the frontmost application). You can also paste something into the pane with the standard key binding ⌘V and you can drag and drop something into a pane as well (you need to start the drag before invoking Quicksilver).

You can select the last object used by typing ⌘[after activating Quicksilver. In fact it recalls the entire last command, that is the object, action and argument used. Unfortunately as of B51 it toggles between the last command and the current one and you can't select earlier ones by typing ⌘[again.

Immediate Execution

You can create triggers that perform an entire command, like open an application or anything else. Triggers are great but they require you to pre-configure them. You can also perform a command by holding down the key you use to select a subject or action. For example, if you type `ad` to select Adium.app and hold down the `d` then Quicksilver will execute the default actioni which should be **Open**. As described above, for me typing `a` will default to the Address Book contact for my friend Ashish. I can activate Quicksilver and hold down `a` to open an email message (the default action for a contact) addressed to her.

This trick also works for actions. If you hold down the last key you type to select an action it will execute it. So I can select Ashish in the first pane, tab to the second and hold down `e` to use the **Edit Contact** action. This method is a little risky if you don't know what the action will be, but if you do it's a little faster and there's no trigger configuration needed. It's less risky when you realize you can configure the ranking of actions in the Actions preference pane.

To make it faster you can type the letter with the ⇧⌘ modifiers from the first pane. So to edit Ashish's contact entry I would activate Quicksilver, type `a` to bring up her entry and then type ⇧⌘E to have Quicksilver execute the command. This only allows you to use one letter to identify the action and has the same risk that you have to know what action will be run, but if you do, it can be convenient.

TODO: holding down Tab in first pane will execute the command. Holding down right-arrow or / when navigating will too.

The Comma Trick

A less obvious feature is known as the comma trick. Using the comma key, you can select multiple items in the object pane (or the third pane) and then have a single action operate on all of them at the same time. Using this you can move several files to the trash at once or address a new mail message to several different people. Activate Quicksilver, select an item, type **,** (comma) and then select another item. You'll see a collection of smaller icons accumulate at the bottom or left of the object pane depending on the interface used (the menu interface does not support the comma trick). You can select as many items as you want. When you have them all, `tab` to the action pane and continue as normal. If you have several collected you can type the `delete` key to remove the last item collected (i.e., the last small icon).

Another way to select multiple items is with ⌘A. It will select all the items in the results list as if you had used the comma trick for all of them. If your results list is long this can take a long time. It's most useful if you navigate to a folder with a small number of items that you want to operate on (e.g., deleting, moving, or tagging them all).

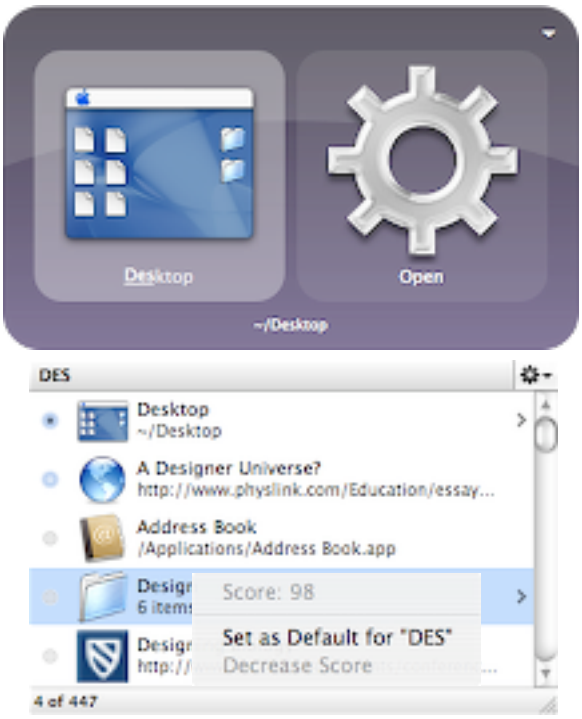
Helping the Matching Algorithm

Quicksilver learns what you do (and makes that easier to do) because of its matching algorithm. It remembers what commands you execute and what you did to call them up, do them a few times and they get easier to call up. However if you do things that are similarly named it confuses Quicksilver and it won't guess as quickly as it could. Fortunately you can teach it. Triggers, described above, are the fastest way to do things in Quicksilver, but there are only so many keystrokes you can remember. Here are some other techniques you can use.

If you activate Quicksilver and start typing into the object pane you'll see a results list appear of items that match what you've typed. The order of the items is known as their rank. The first item is ranked 1 the second is ranked 2, etc. For objects, the rank is determined based on the score of the items. Score is computed based on how well an item matches what you typed.

Theoretically, items move to second place rank the first time they are used to match some input. The second time they are used they get ranked first. The circle to the left of the item indicates how well the item matches the typed input. The darker the circle, the stronger the match (i.e., the higher the score).

You can manually adjust the score of an item by ^-clicking on it to bring up a menu of choices. If you choose "Set as Default for…" then the selected item will match if you type the same input again. You can undo an abbreviation by choosing Decrease Score in the menu.



The user ^-clicked on the 4th item in the results list after typing `Des`*. The pop-up menu shows the item's score, 95 and the option to set it as the default for* `Des`*.*

Setting a default works for what you've typed, but say you want to set as a default something that is mid-way in a very long list that you don't want to scroll through. As an example say you want `z` to bring up your Amazon bookmark and that if you type z, Amazon is further down the list than you want to look. In that case, use the **Assign Abbreviation** action (you need to have advanced features enabled). Bring up the object you want in the first pane (www.amazon.com), choose **Assign Abbreviation** as the action, and in the third pane enter the abbreviation you want via text mode (`z`). Now if you type that abbreviation in either the first or third pane, your choice will be ranked first. This happens because exactly matching an abbreviation causes the item to have a very high score. Note abbreviations will only work if they match the input string in some way.

You can't set `irc` to be an abbreviation for Colloquy.app because if you typed `irc`, Colloquy would not appear in the results list. Say you don't use Activity Monitor often and when you want it you can never remember its name but you think of the word "processes". Quicksilver doesn't help here since it doesn't relate "processes" to Activity Monitor, and as a result Abbreviations don't work for this. TODO: (this doesn't work) But you can use Finder aliases. Find the Activity Monitor.app in /Applications/Utilities/ and create an alias for it called Processes.app. Put the alias someplace that Quicksilver will index it. Now if you type `processes` quicksilver will suggest your new alias.

The above is all for objects in the first and third panes. Actions in the second pane work a little differently. Actions have no score, their rank is statically determined by their order in the Actions Preferences. You can manually change the order by dragging and dropping them (for this to work you must sort by the Rank column).

TextMate Ranker

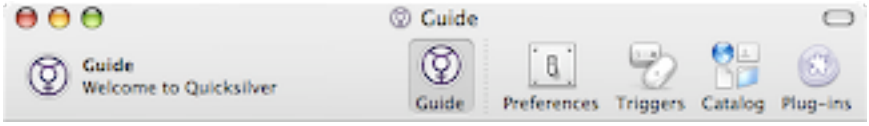
You can also choose a slightly different matching algorithm by installing the TextMate Ranker (+) plug-in (which requires advanced features to be enabled) and selecting it as the String Ranker handler in Preferences.

It makes two changes to the algorithm. The first is that it favors more beginning of word letters that are touched. E.g., if you type `AM` it will match Activity Monitor over Adium since two beginning of word letters are matched instead of just one. The second difference is that it favors input that matches a higher percentage of the words. So the input `AM` will favor Activity Monitor over Audio MIDI Setup since 2/2 (100%) of words are matched instead of 2/3 (67%).

As a result, if you're the type to favor typing acronyms over continuous letters in the name of something, you'll probably prefer the TextMate Ranker. However there are reported problems with matching things in the third pane such as when you use the **Open With…** action or the **Email to…** action. You may find that no results match in the third pane.

Part II - Configuration

Configuration



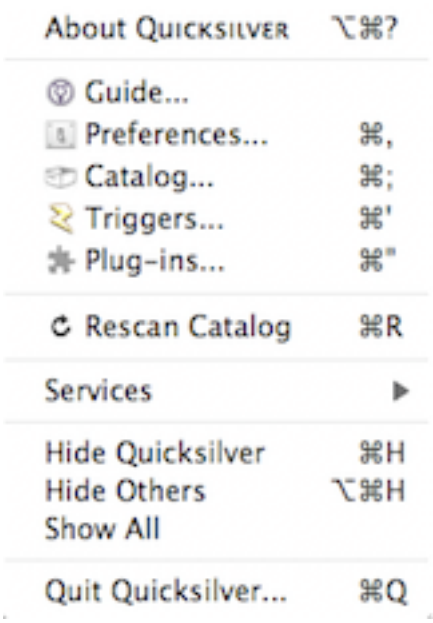
Quicksilver has a configuration window that provides access to the Guide, Preferences, Triggers, Catalog and Plug-Ins. These can be accessed several ways:

1. This is the way I usually do it, after activating Quicksilver (with ^-space) use the standard shortcut ⌘, to bring up preferences or use one of these other shortcuts:

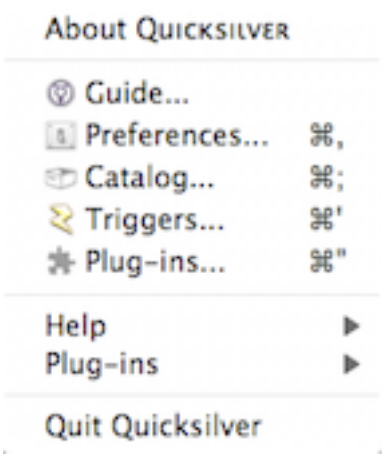
⌘?	Open Guide
⌘,	Open Preferences
⌘;	Open Catalog
⌘'	Open Triggers
⌘"	Open Plug-ins

2. In the main command window select one of the Internal Commands: QS Preferences, QS Triggers, QS Catalog and use the **Run** action.

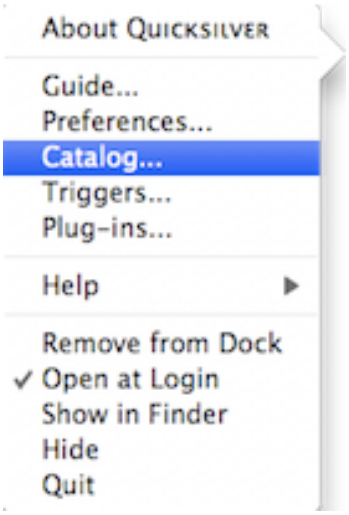
3. Click the Quicksilver Dock icon and Quicksilver's menu appears, look under Quicksilver



4. If the menu bar icon is enabled, a menu is available by clicking on it



5. Right-click the Quicksilver icon in the Dock

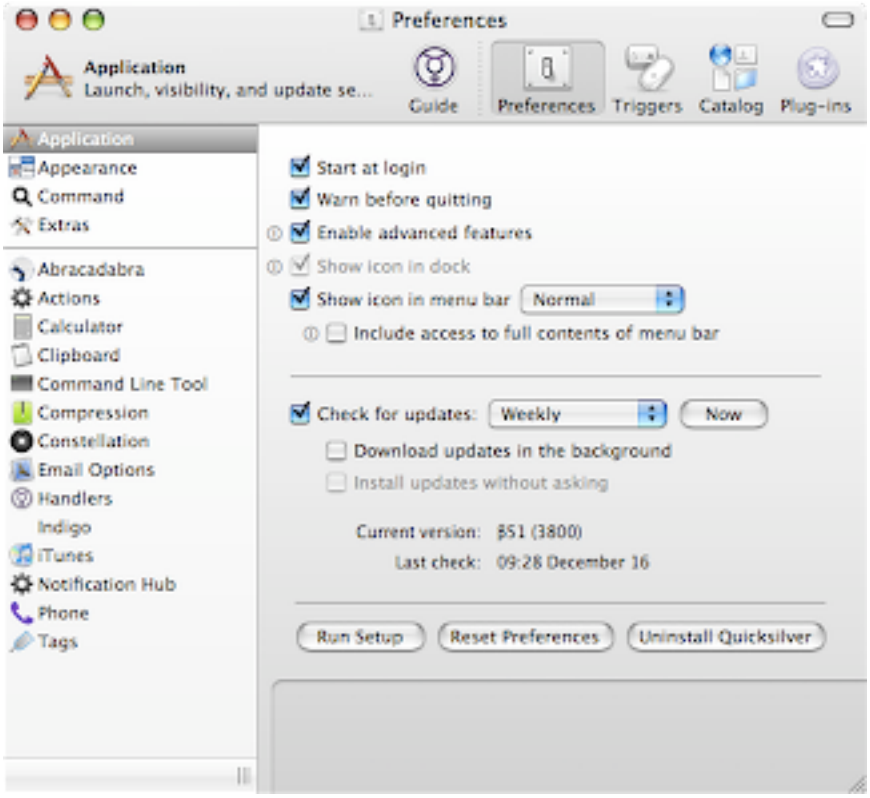


The following sections describe these preferences. First global preferences that affect how Quicksilver is activated and appears and then sections on Plug-Ins, the Catalog, Actions, Handlers, and Notifications. Triggers are a large topic and are covered in their own entire section immediately after this one.

Adding new Quicksilver features is usually done by installing the relevant plug-in. Sometimes there are additional things to configure such as enabling new catalog sources, new actions (under Preferences, Actions) or relevant Handlers (under Preferences, Handers). Finally some plug-ins add their own preference panels listed in the lower left area of the Preferences window.

Preferences

Quicksilver's Preferences are divided into sets at the left of the window. Those listed at the top are built-in to Quicksilver and those in the bottom section are enabled by installed plug-ins (except for Actions and Handlers which really should be in the top).



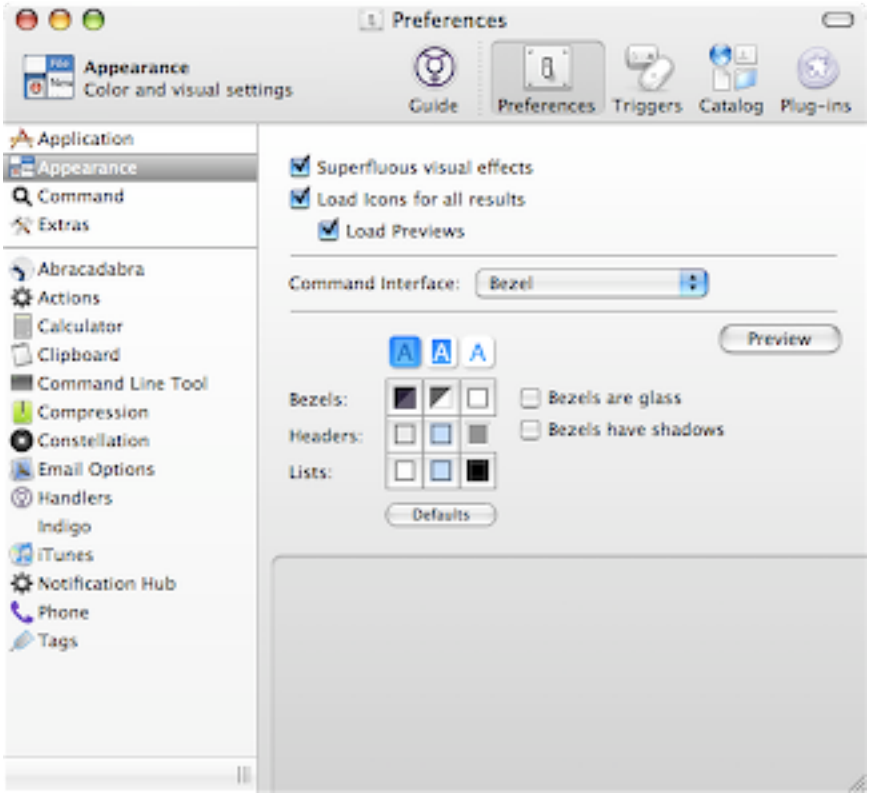
The Application Preferences control the most fundamental aspects of Quicksilver as an application. "Start at login" and "Warn before quitting" are recommended. The next two preferences require a restart before they take effect, this is what the small circle with a line icon indicates. The easy way to restart is to activate Quicksilver (^-space) and then type ^⌘Q

There are many features described in this manual that say you must enable advanced features. To do so check the third preference shown. After restart you'll see the "Extras" Preferences set shown in the image and described below. I recommend you enable Advanced features as a lot of useful plug-ins require it (those are indicated with a + symbol in their name). These aren't necessarily unstable (as those plug-ins marked β usually are).

The "Show icon in dock" preference will let you not waste Dock space with Quicksilver. If you disable this, Quicksilver runs as a background process. This means it won't appear in the Dock or when switching apps with ⌘-tab and the main Quicksilver menu will not appear while the preference window is open. I tend to keep it in the Dock so that it's easy to restart if Quicksilver crashes. The "Show icon in menu bar" preference puts an indicator in the top right part of the screen, some people like this, particularly if they don't configure a Dock icon or have their Dock hidden. It provides a simplified menu to access the various configuration panes of Quicksilver. If you want it to provide the full menu options (via more sub-menus) set the option "Include access to full contents of menu bar". The popup allows you to configure the menubar icon on the left, right side of the menubar or "Normal" lets you configure it as any other menubar item,

The bottom half allows you to configure if Quicksilver checks its website automatically for updates. You can specify if it should automatically download updates and whether or not they should be installed automatically. Some betas have had some problems if installed this way. Usually a manual download of a new version from the Quicksilver website will solve any problems.

The three buttons at the bottom will rerun the installation setup described in the Installation section, Reset Quicksilver's preferences, and completely Uninstall Quicksilver.

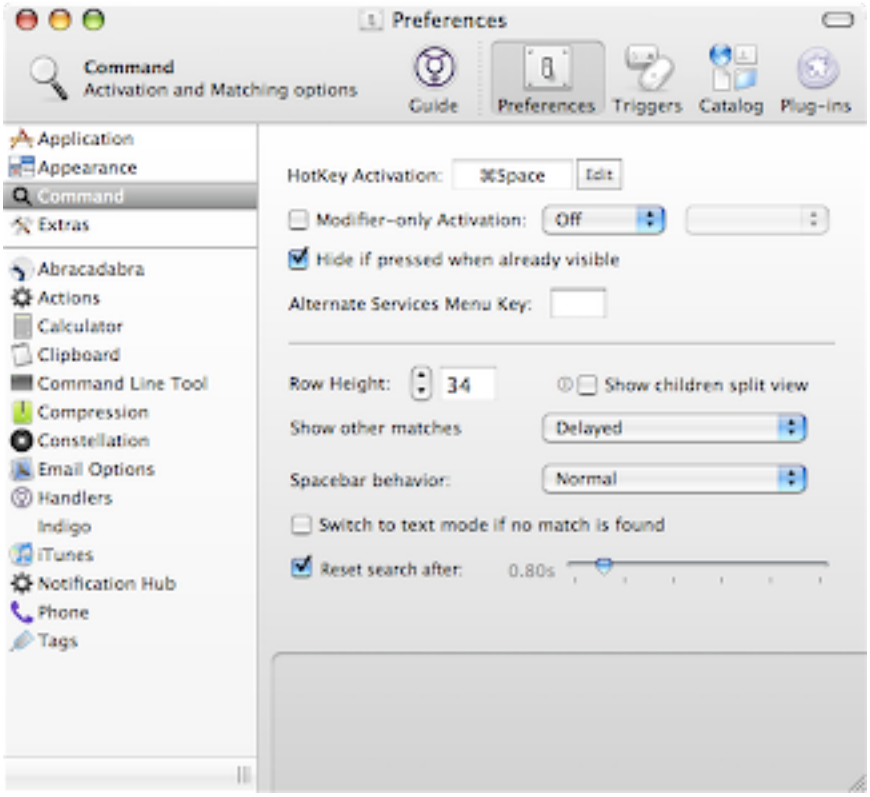


The Appearance Preferences control how Quicksilver looks. The first three options all have the trade off of being a little prettier at the cost of some performance. Enable them if your machine can handle it, if you're finding Quicksilver a bit sluggish try disabling them.

The Superfluous visual effects vary per interface but include the command window unzipping open and fading closed.

Normally Quicksilver doesn't load all the icons of items in the results sets, the second option changes this behavior. The "Load Previews" option is similar to the "Show icon preview" in the Finder's View Options. It will display a thumbnail image of a file or other item in a pane in the command window.

The Command Interface preference is how you choose the interface (aka skin or theme) Quicksilver uses for the main Command window. The popup is filled with choices based on the Interface plug-ins installed. See the Interface section below for details. The default interface is Primer and it shows a little more context information than other interfaces so it's recommended for new users. The Bezel interface is very popular and is shown throughout this manual. The next section shows all the available interfaces. The various controls below the interface selection allow you to further customize the appearance of the interface and vary per interface. The column headers have tooltips showing what they represent: Background, Selection & Accents, and Text colors. Clicking on the nine colors in the grid will bring up a color picker.



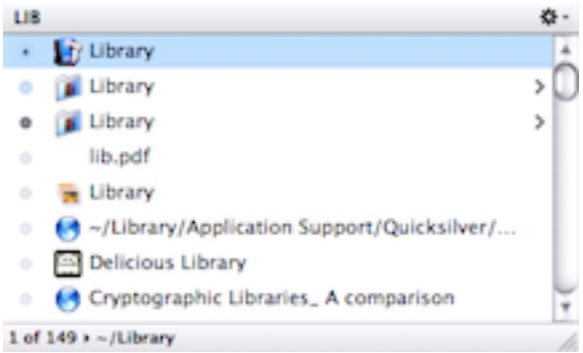
The Command Preferences allows configuration of the command window's behavior.

The top of the pane affects how you activate Quicksilver. The HotKey Activation is the basic way to activate Quicksilver. The default is to type ^-space, though as shown a common alternative (described in Invoking Quicksilver) is to use ⌘-space and to remap Spotlight to something else via the Spotlight System Preferences.

"Modifier-only Activation" allows you to activate Quicksilver by single or double presses of modifier keys such as ⌘, ⌥, ^, ⇧ or even `fn` on a laptop. Some applications can call for secure text entry and this can interfere with modifier only activiation. If you use Visor and a double shift modifer isn't working for you, open Terminal.app and in the File menu enable "Secure Keyboard Entry". There have also been reports that Quicken Scheduler app that runs in the background as a login item to do scheduled downloads can also interfere with modifier only activation.

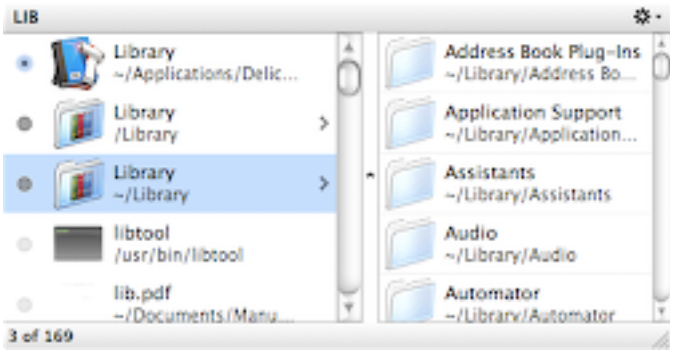
"Hide if pressed when already visible" affects what happens if you activate Quicksilver when it's already active. Regardless of the setting of this option, typing `esc` will close the command window.

The "Alternative Services Menu Key" is how to change the shortcut assigned to the Quicksilver service Get Current Selection. The default value is ⌘-`esc`, so you might want to change it if you use Front Row as described above in Invoking Quicksilver.



The bottom of the Command Preferences pane affects how the results list window appears. You can specify the Row Height in pixels. A higher value will show a larger image on each row. A lower value will result in a single line per entry. The info displayed in the 2nd line for the selected entry will be shown in the bottom of the window.

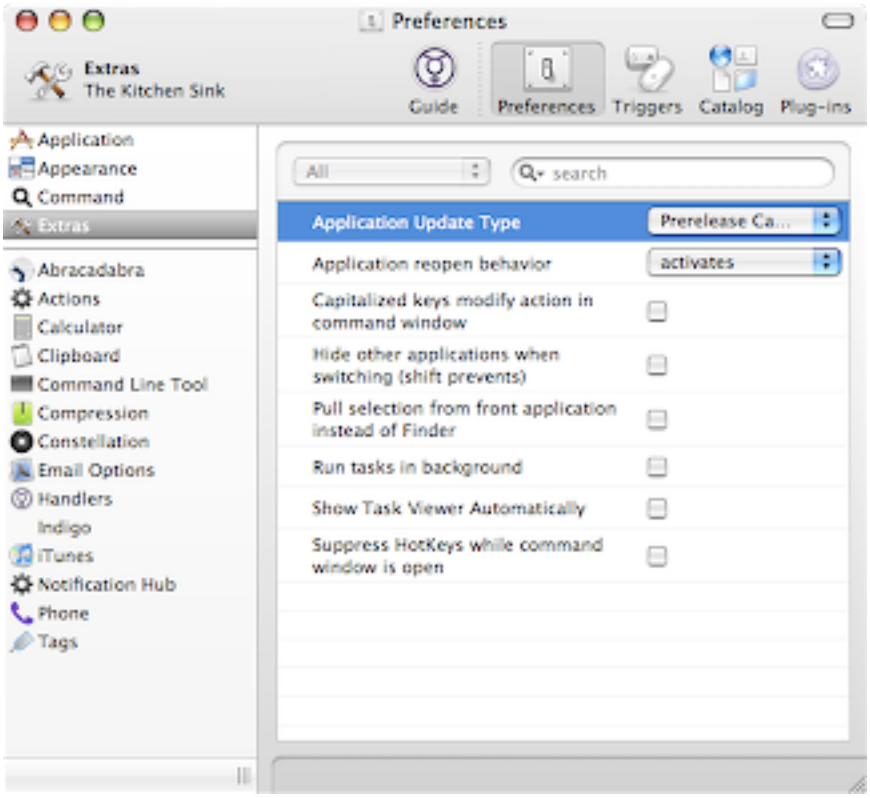
Selecting “Show children split view” will split the results list vertically into two columns, the right one showing the contents of the selected item in the left one (much like the column view of the Finder). You must restart Quicksilver for changes to this setting to take effect. When first enabled you’ll only see one column, but notice the small dot in the middle of the right edge, you can click on and drag that to the left to reveal the second column.



“Show other matches” controls if the results list appears after you type into the command window immediately, after a short delay or not until you type ↓. By virtue of Quicksilver’s matching algorithm, the spacebar isn’t particularly useful in the command window. The “Spacebar behavior” preference lets you reclaim that key to make Quicksilver a little faster to use. You can configure it to:

- Select Next Result - select the next result in the results list, like typing ↓
- Jump to Argument List - change to the third pane
- Switch to text mode - change the current pane to text mode entry, like typing . or ‘
- Show Item’s Contents - goes into the item, like typing → or /

“Switch to text mode if no match is found” will allow you to save typing . or ‘ to enter text mode in a pane. “Reset search after” lets you configure how much of a pause in typing into the command will start a new search as opposed to appending to the existing search.



The Extras Preferences are only available if Advanced features are enabled in the Application Preferences.

“Application Update Type” allows you to choose between Prerelease Candidates and Final Releases when Quicksilver checks for updates as configured in the Application Preferences. Sometimes after installation this is set to a blank value and that has been observed to cause problems, make sure it’s set to some value in the popup.

“Application reopen behavior” configures what happens if you use the **Open** action on an application that is already running.

You can have it activate the application, show the front window or show all windows.

Since the matching algorithm is case-insensitive the shift key is available for some use in Quicksilver. If you check “Capitalize Key modifies action in command window“ then shifted letters are used to select the action, eliminating the need to tab to the second pane. Once moved to the action pane, unshifted letters don’t change the first pane, all typing counts for the second pane. If you also hold down the ⌘ key the action will be performed immediately, no need to type `return`. E.g., ⇧I selects the action for i (perhaps **IM**) while ⇧⌘I performs the action for i. Use caution as it’s not always clear which action is invoked.

If you prefer concentrating on one thing at a time, check “Hide other applications when switching” so that Quicksilver will do a “Hide Others” after switching applications. You can prevent the hiding behavior by holding down the shift key when completing the command. There have been problems reported with this feature that applications no longer appear when doing ⌘-`tab`.

Normally ⌘G will take the Finder’s selection and put it into Quicksilver. By checking the “Pull selection from front application instead of Finder” option, it will use the same mechanism as ⌘-esc and will pull the selection from whatever the active application is. See more details at Invoking Quicksilver.

Checking “Run tasks in background” will do just that. E.g., if you use the **Compress** action it does the compressing in the background allowing you to do other things with Quicksilver immediately. However as of B51 some have reported some instability from using this option. E.g., some have reported that running the **Compress** action via a trigger with this option checked causes Quicksilver to hang during the compression.

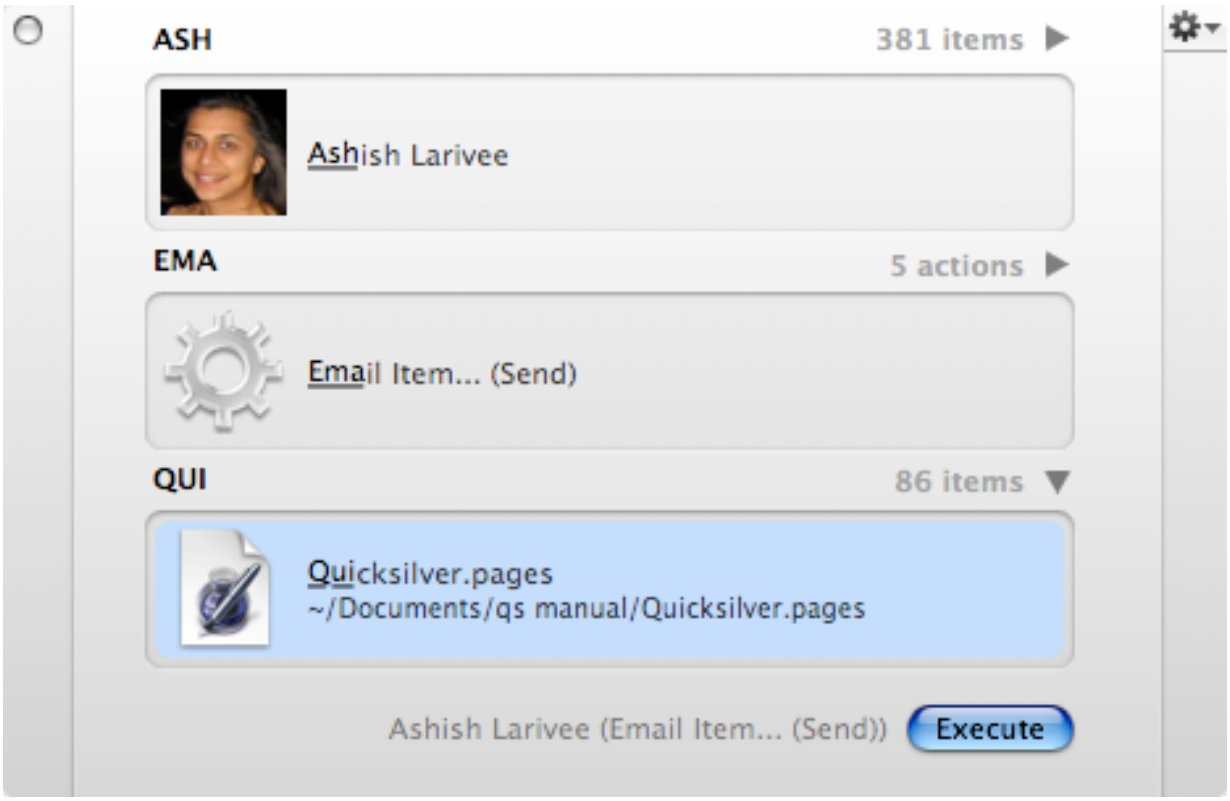
Quicksilver has a small window that shows current status known as the Task Viewer. You can bring it up by activating Quicksilver (^-space) and typing ⌘K. Most people don’t use it. You can also check the “Show Task Viewer Automatically” option to have it appear when Quicksilver is performing tasks such as refreshing the catalog. There’s an image of it in the Catalog section below.

If you enable “Suppress HotKeys while command window is open” then HotKey triggers will not work while Quicksilver is activated (i.e., the command window is open). I have no idea why you would want to enable this.

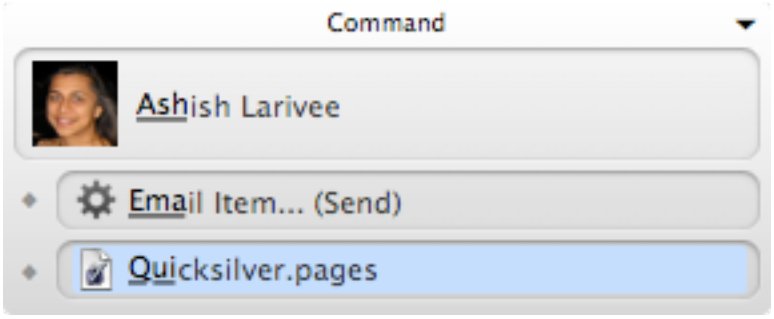
Interfaces

Interface plug-ins change the look of Quicksilver’s command window. Install them just like other plug-ins from the Plug-Ins Preferences. Then select one of the installed interfaces in the Appearance Preference pane using the Command Interface popup. If you use a program like VNC to connect to different machines, you might want to configure them with different interfaces or change the colors on the interface to distinguish them.

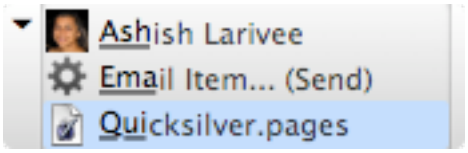
The following screenshots show each of the six interfaces at full size so for comparison purposes. These are all the official interfaces available from the Plug-Ins Preferences. Julius Eckert has created several Quicksilver interfaces of his own that many people like (I had some stability problems when I tried them but that might be corrected now.) They are available here, download the plugin and double-click them to install them, the select them from the Appearance Preference pane like any other interface.



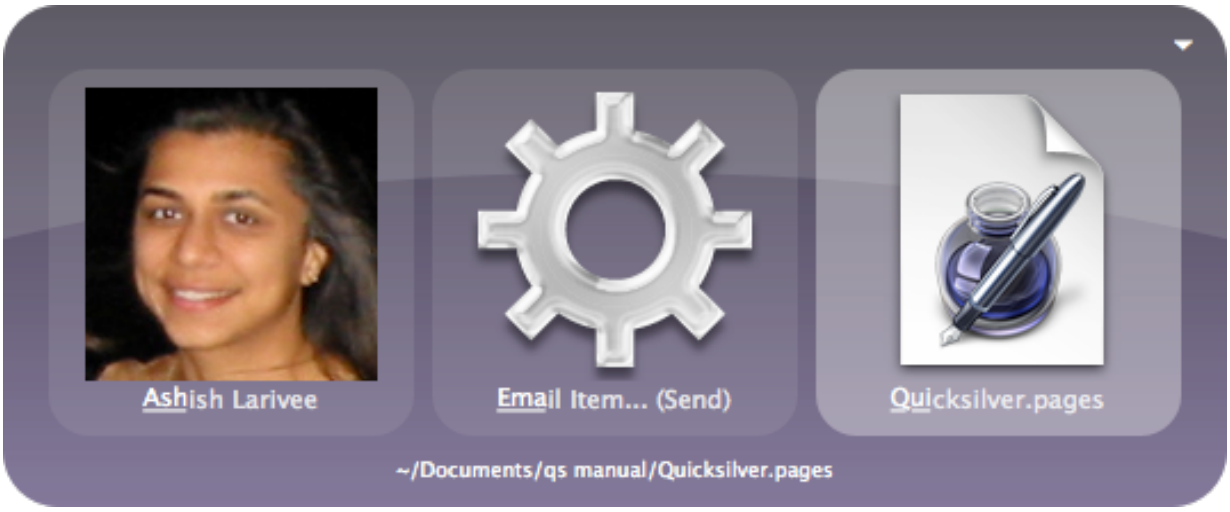
- **Primer** - The default interface is large and provides more info to help you learn



- **Window** - Like Primer but smaller and with less info

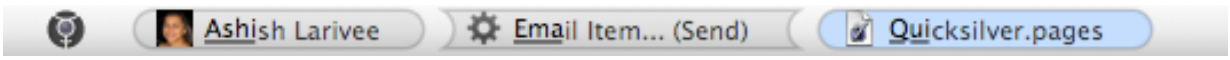


- **Mini** - Like Primer and Window but even smaller

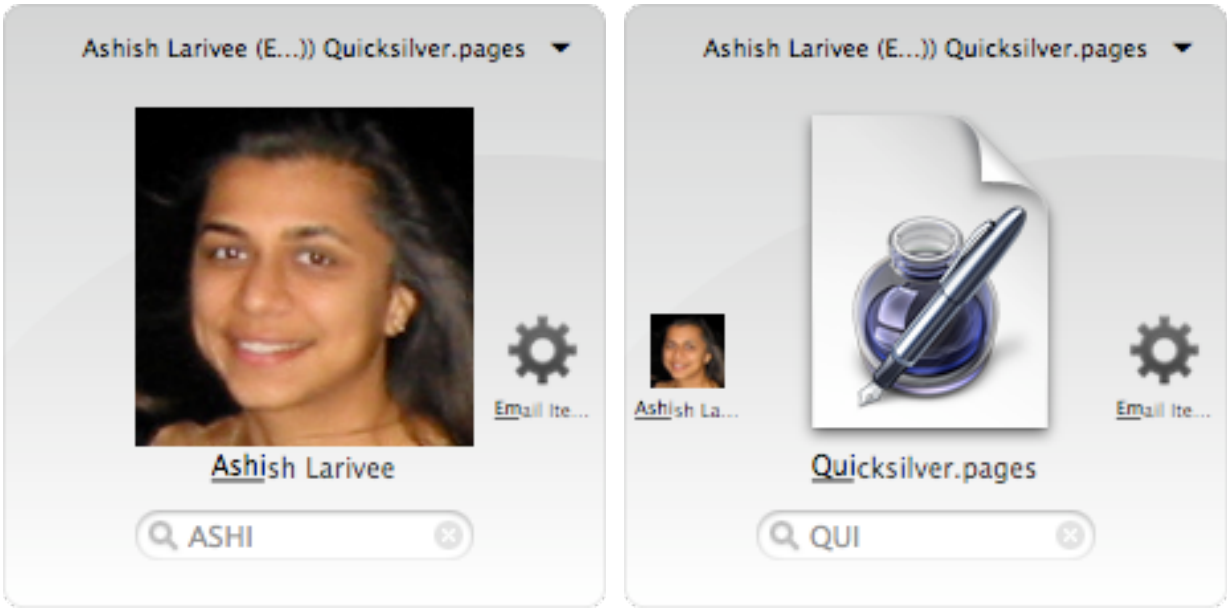


- **Bezel** - Grey, horizontal panes. Very popular (used in this manual at reduced size)

- **Flashlight** - Looks like Spotlight and appears in top left corner of the screen



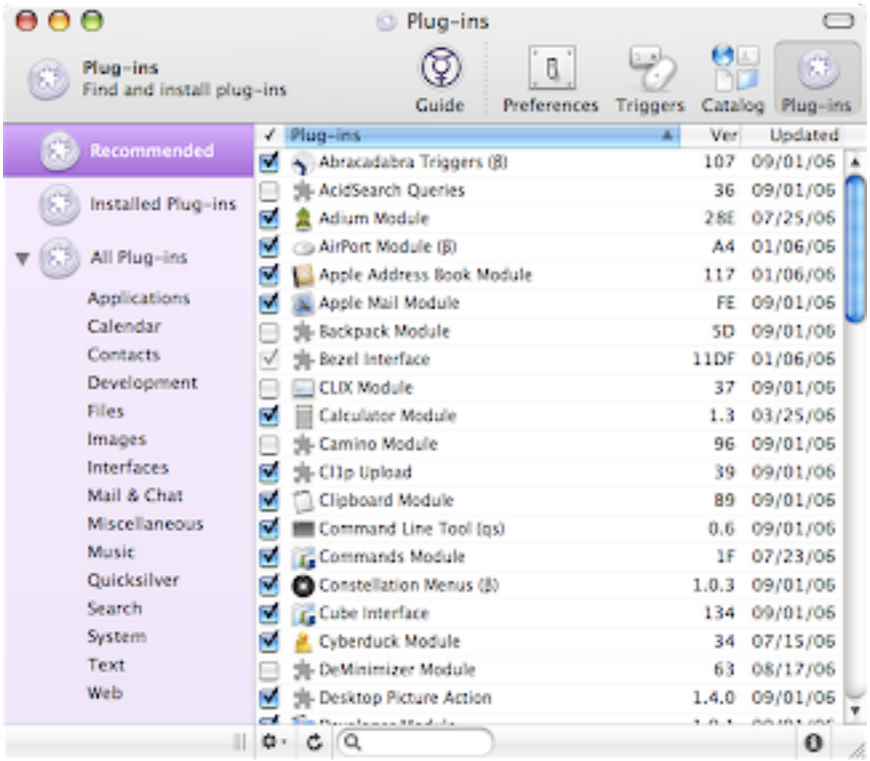
- **Menu** - Fills the OS X menu bar with input panes, doesn't support [the comma trick](#).



- **Cube** - A single square pane that rotates as you tab between the panes. Shown above are the first and third sides of the cube on this one command. Note that the order of the command parts (object, action, argument) changes on each side. It's very slick seeing it rotate.

Plug-ins

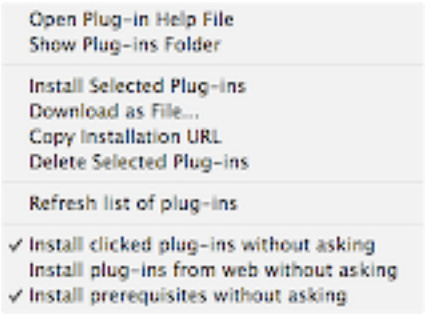
Quicksilver is designed with a central core that implements some basic functionality but most of the features are implemented in plug-ins. You can pick and choose which plug-in functionality you want, but must install the plug-ins before that functionality is available. Managing the plug-ins including finding, installing, enabling, and removing is done entirely from within Quicksilver in the Plug-ins Preferences (though additional configuration in Preferences or the Catalog may be useful or necessary). You can bring it up from the menu or by activating Quicksilver and typing ⌘”.



The plug-ins are shown in the right pane, with a checkbox showing if they are enabled, a column showing the version number (which follow no consistent pattern) and the date the plug-in was last updated. The left panel shows sets of plug-ins:

- **Recommended** - recommended for you based on a scan of what applications are installed in the machine.
- **Installed Plug-ins** - those plug-ins that are installed in ~/Library/Application Support/Quicksilver/PlugIns/. Only those that are checked are enabled, those not checked are installed, but disabled.
- **All Plug-ins** - all available plug-ins from the quicksilver.blacktree.com server. Checked plug-ins are installed and enabled, unchecked plug-ins are not installed. ⌥-click on this to show some hidden built-in plug-ins like Core Support and E-mail Support.
- **Categories** - shows plug-ins collected into relevant categories by feature. Plug-ins can be in more than one category. Otherwise behaves like All Plug-ins, checked plug-ins are installed and enabled, unchecked plug-ins are not installed.

Checking a plug-in will enable it, downloading and installing it if necessary. Selecting a plug-in and clicking the ⓘ button will open a drawer with (usually limited) information about what the plug-in does. Selecting one or more plug-ins and clicking the + button opens a pop-up menu of things you can do to plug-ins including install, download, copy, and delete. Delete is the only one you'll typically use from this menu. The last three items in the menu are options you can enable or not. The ↻ button will refresh the list of plug-ins from quicksilver.blacktree.com.



To disable a plug-in, uncheck it. Its features will no longer be available but its code will still be loaded into memory. To clean up this memory (possibly fixing stability issues), restart Quicksilver. The plug-in is still installed on disk (so it will appear unchecked in the Installed Plug-ins view) until it is deleted. So, to uninstall a plug-in, select it from the list and choose the Delete Selected Plug-ins option from the menu.

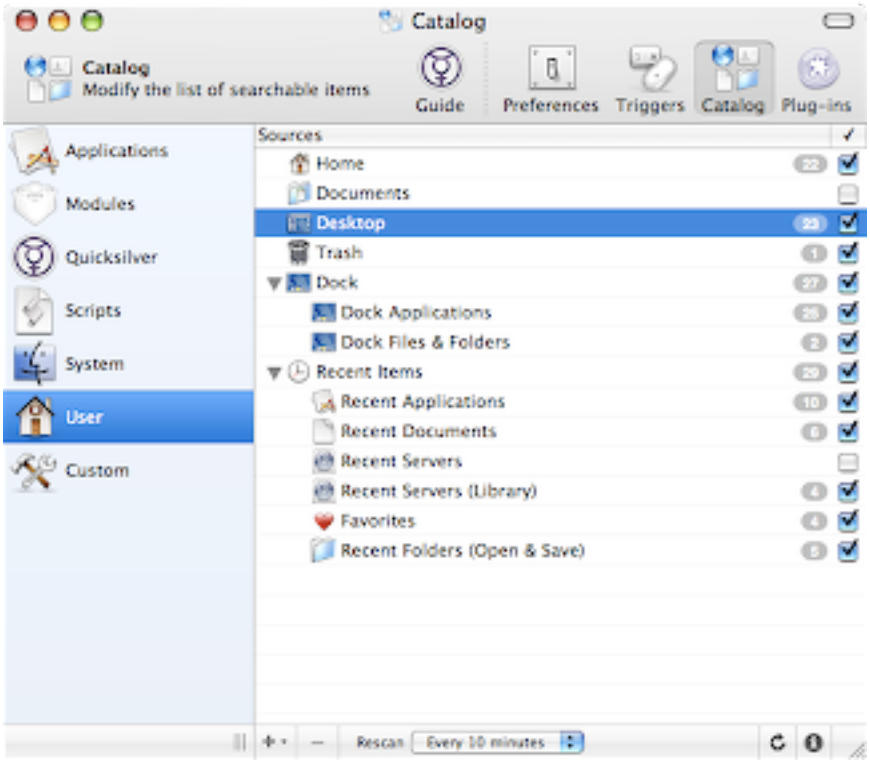
If you expect to see a plug-in in the list and don't, try refreshing the list of plug-ins. Note that some plug-ins are only visible if Advanced features are enabled. Usually these plug-ins have names that end in (+). Plug-ins that end in (β) should be considered to be of beta quality.

If you're having problems installing plug-ins check the ownership and permissions on ~/Library/Application Support/Quicksilver/PlugIns/ and its parent directory. Use the Finder's Get Info command (from the File menu) to see the Ownership & Permissions of a folder. It should be owned by you and you should have permission to read and write it. Usually quitting Quicksilver and removing (or moving) the PlugIns folder or its parent Quicksilver folder and restarting Quicksilver (allowing it to recreate the folder) will solve any problems. Of course removing the Quicksilver folder will remove any customizations you've made.

Some people have reported that using the Little Snitch network monitor interferes with Quicksilver's ability to download plug-ins. If your plug-in list is empty and you run Little Snitch, try disabling it.

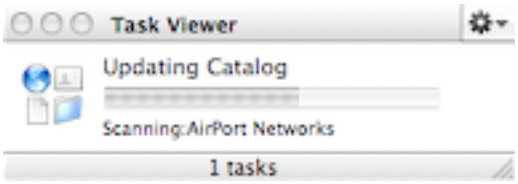
Catalog

The Catalog is the collection of items indexed by Quicksilver during its periodic scans. You populate it by configuring catalog sources which Quicksilver periodically indexes. This is done in the Catalog panel. You can bring it up from the menu or by activating Quicksilver and typing ⌘;.

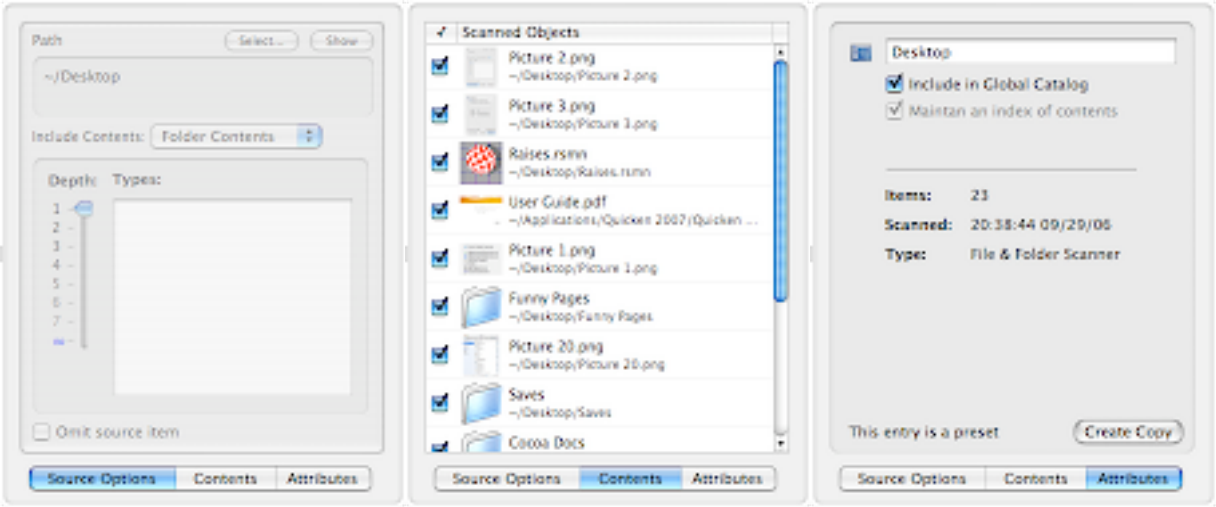


Catalog sources are configured into sets shown in the left panel. Most are pre-configured by various plug-ins and many plug-ins add sources to the Modules set. The checkbox enables or disables the source. Enabled sources that have been indexed show the number of items found. Note that some sources (such as Proxy Objects in the Quicksilver set) require advanced features to be enabled in the Application Preferences and not all sources are enabled by default after plug-ins are installed.

At the bottom of the window you can configure how frequently Quicksilver rescans the catalog sources to find (and remove) items. The default is 10 minutes. The ↻ button will manually rescan a selected catalog source. You can manually rescan the whole catalog with the Rescan Catalog command in the Quicksilver menu (only if you have the Dock icon visible) or more conveniently by typing ⌘R after activating Quicksilver. Some interfaces (e.g., Bezel) will show a progress icon while rescanning the catalog. You can also configure the Task Viewer to show automatically during a catalog rescan. Show the Task Viewer by selecting it from the Window menu (if the Quicksilver Dock icon is enabled) or by typing ⌘K after activating Quicksilver. The gear menu in the top right (which is there even if it's not visible in some interfaces) will let you configure two options: Show Automatically and Resize Automatically. The Task Viewer isn't all that useful as things usually just work. It could be used to notice when Quicksilver is doing rescans or perhaps to help troubleshoot a slow catalog source (if you've configured a file scan to be too deep).



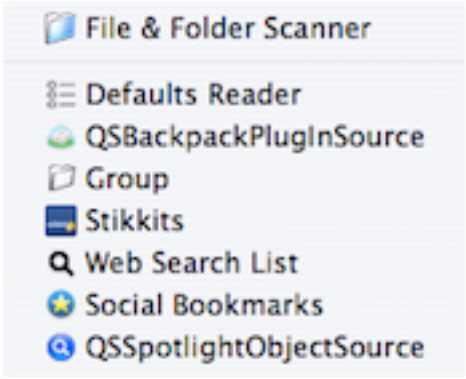
With a source selected click on the ⓘ button to show a drawer with three tabs. Shown below are the tabs for the Desktop source selected in the above image.:



- Source Options - varies based on the kind of source. A file source (shown) allows you to configure depth of scan and file types to be included, see below for details. This is shown greyed out since the Desktop source is not configurable.
- Contents - the list of all items found by indexing this source. You can remove a specific item by unchecking it
- Attributes - provides some info about the source and allows you to enable it and change the name. Include in Global Catalog is whether the source is enabled or not and is the same as the checkbox in the main Catalog panel. Predefined sources have a Create Copy button that will duplicate the source in the Custom catalog set, allowing you to change the source options.

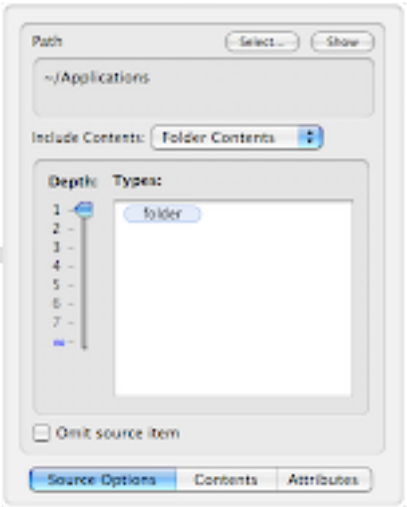
To find out what catalog source an object comes from, bring up the object in the first pane and use the **Show Source in Catalog** action; the Catalog window will open with the source of the object selected. If the catalog contains unwanted items, this is a way to track down what sources you want to modify or remove.

Using the + and - buttons in the Custom set you can add and remove additional catalog sources. You can also create custom sources via the Create Copy button in the Attributes tab of pre-defined sources. The + button will show a drop down menu of various source types (aka scanners) which varies based on the plug-ins installed. Shown at the right are all available scanners. Note that some plug-ins install catalog sources in the Modules set and others install new scanner types. When trying to determine what a new plug-in does, remember to check both places (or read this manual).



The File & Folder Scanner lets you add folders to scan into the catalog. You can also just drag folders into the main catalog pane to add a source. A common thing to want to do is to scan the ~/Documents/ folder deeper than the depth of 2 the default source uses (in the User set). To do this, select the Users set and the Documents source, open the drawer and select the Attributes tab. Click on Create Copy to create a new source in Custom named Documents. The difference is that the Source Options tab of it is editable. Change the depth slider to what you want, I have mine at 3. Many people just select infinity but that's probably not what you want. Your indexes will take longer and your catalog will be huge which will slow Quicksilver down. Also if you have many extra items in your catalog your searches are more likely to have extraneous results. Remember that you can always navigate to any file in Quicksilver (see the Files and Folders section) so all you need in your catalog are your more commonly used files. Also you probably don't want to just index your entire home directory. ~/Library has many preference and cache files that you don't want indexed and the Music and Pictures folders are better served by the iTunes and iPhoto plug-ins and their catalog sources. The File & Folder Scanner can also scan the contents of files to add to the catalog. The scanner can recognize either HTML links or text lines depending on the setting of the Include Contents popup.

The Applications set has 4 sources defined. The Applications (User) source indexes ~/Applications/ to a depth of 3 but also only finds the applications, not the intervening folders. I want those in the catalog so that I can easily move an application into them (e.g., ~/Applications/Games/ and ~/Application/Browsers/). The Types field in the Source Options tab is what's useful for this. If you type file extensions (e.g., .txt, .c, .doc, etc.) into it you can filter the kinds of files indexed. You can also enter Mac Type Codes surrounded by single quotes and type `tab` to have it interpreted. So, to get applications entered as it is in the preconfigured source, enter `'APPL'` (including the single quotes). In this example I want folders, so I enter `'fold'` and type `tab` and it's replaced with "folder". I choose a depth of 1 to include my Games and Browsers folders but not folders that some applications come with (or the Contents folders inside the .app packages) . TODO: Some Type Codes are listed here.



Note that the Find All Applications source under the Applications set will search the whole system for application packages. It also indexes any external drives connected. Some people complain that when connecting drives to their machine the drives spin up. To prevent Quicksilver from causing this, uncheck the Find All Applications catalog source.

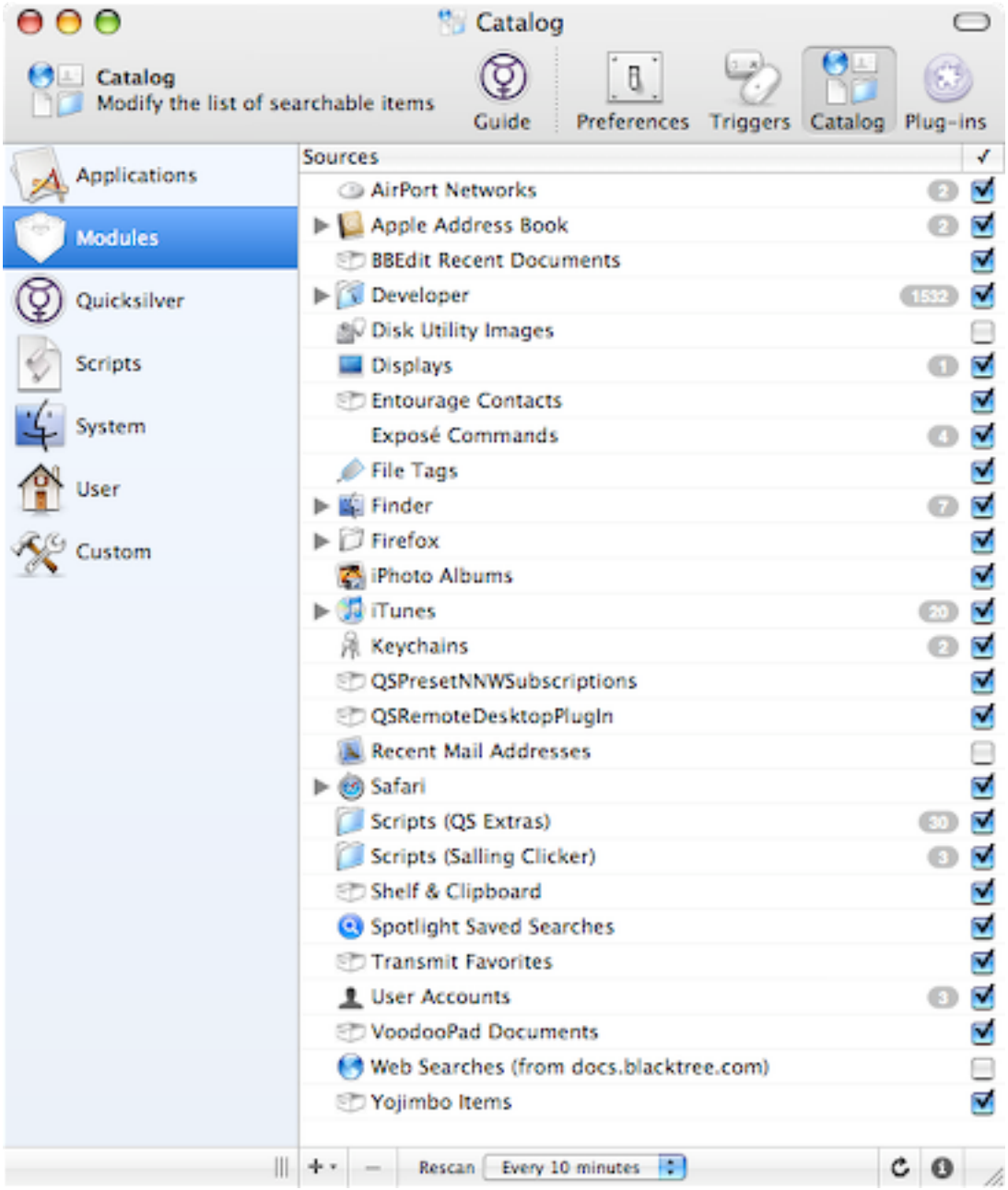
The QSSpotlightObjectSource allows you to specify a Spotlight query to find items to add to the catalog. See Apple's documentation for Spotlight's Query Expression Syntax and Common Metadata Attribute Keys. Unfortunately in B51 this is buggy. It's difficult to get one of these sources to scan and it sometimes goes into a loop continuously rescanning and using lots of CPU.

The Defaults Reader allows you to index some keys from preferences files (.plist), though it's a little flakey (doesn't deal well with paths), is difficult to configure (you must manually specify each key) and doesn't let you change the plist file, so it doesn't seem useful.

The Group type is just a a folder for custom catalog sources to be able to group many custom sources in the catalog preferences to make them easier to read.

QSBackpackPluginSource, Stikkits, Web Search List, and Social Bookmarks are described in other places in this manual.

If Advanced features are enabled, in the Quicksilver set there is a source called Quicksilver Catalog Entries. If enabled, an item is added to the catalog for each source configured. These items have names that end with "(Catalog)". You can select one of these items in the first pane and then type → to navigate through just that source. If you want to do this often, create a trigger. E.g., the trigger Applications (Catalog) (**Show Contents**) lets you search through just the applications in your catalog (well those that are found via this catalog source). While you can often do this just by typing → into an object, if you have custom sources configured (say for files of a particular project) this can be very handy.



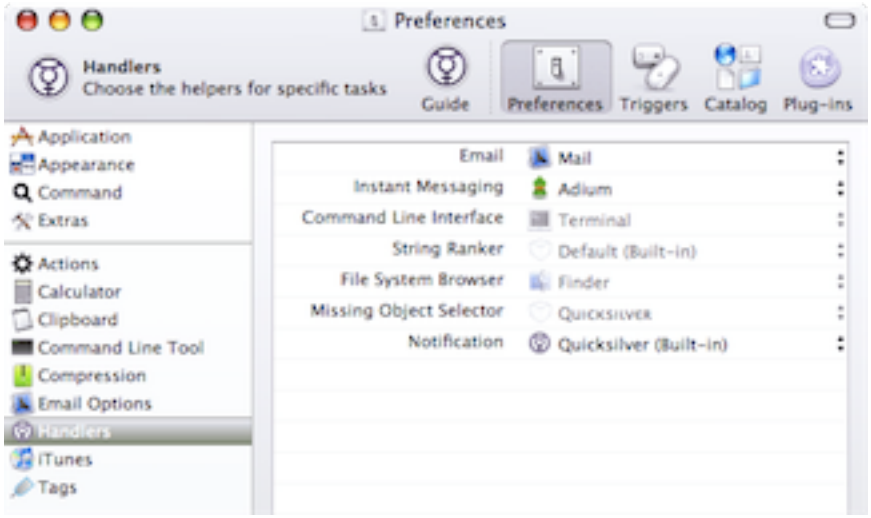
Above is a screenshot showing all the catalog sources installed under Modules with all plug-ins installed in B51.

Actions

TODO

Handlers

Quicksilver not only makes it easier to work with a variety of applications, it sometimes uses other applications to perform commands. These are configurable in the Handlers Preferences pane. The list of handlers varies based on which plugins are installed. Shown here are all the handlers available as of B51.



- **Email** - Mail program to use for Email actions (Mail, Gmail, Entourage, or Mailsmith).
- **Instant Messaging** - IM program to use for IM and chat actions (iChat or Adium).
- **Command Line Interface** - Terminal program to use (either Terminal or iTerm) for Terminal actions .
- **String Ranker** - The TextMate Ranker plug-in installs a different matching algorithm that can be selected here. See the TextMate Ranker section for details. As of B51 it's known to cause problems selecting things in the 3rd pane.
- **File System Browser** - The Path Finder plug-in installs a handler that can be selected here to have Quicksilver use Path Finder instead of Finder for some file system actions like **Reveal**.
- **Missing Object Selector** - This is installed by the Constellation plug-in. It should be set to Quicksilver and not Constellation Menus.
- **Notification** - Display notificiations with Quicksilver, Console or Growl. See the next section.

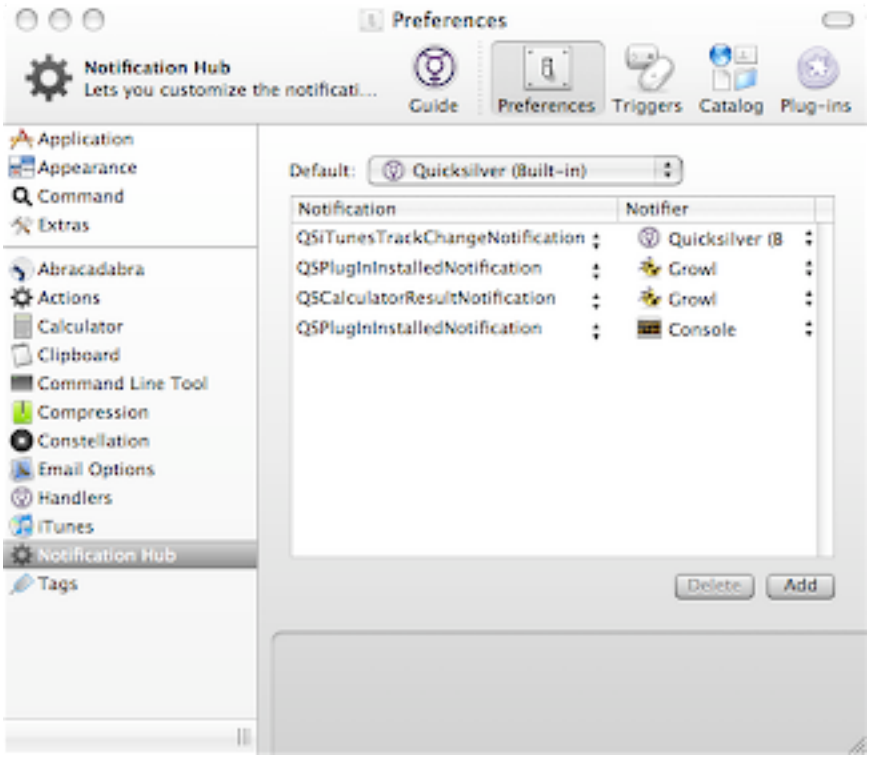
Notifications

Quicksilver supports 3 different notification events:

- iTunes Track Changes
- Calculator Results
- Plug-in Installation

These are typically displayed using Quicksilver's built-in notification system which pops up a small window (shown at left for an iTunes track). In Handlers preferences under Notifications you can change to have notifications sent to the Console. If you install the Growl Notifier plug-in you can instead choose to have Growl handle notifications. One limitation is that iTunes notifications generated by Quicksilver and displayed via Growl will not show the track's rating. Ratings are shown when using Quicksilver's built-in notifications.

The Notifications Hub plug-in allows finer granularity in configuring notifications. It adds a new preference pane called Notification Hub. There you can configure a default notifier as well as configure specific notifiers for specific events. E.g., iTunes notifications go to the Quicksilver built-in notification notifier while Plug-in Installation notifications go to Growl. You can even send the same notification to multiple handlers by adding multiple lines for the same Notification and with different Notifiers. As of B54, The plug-in is Intel-only and will not work on a PowerPC Mac.

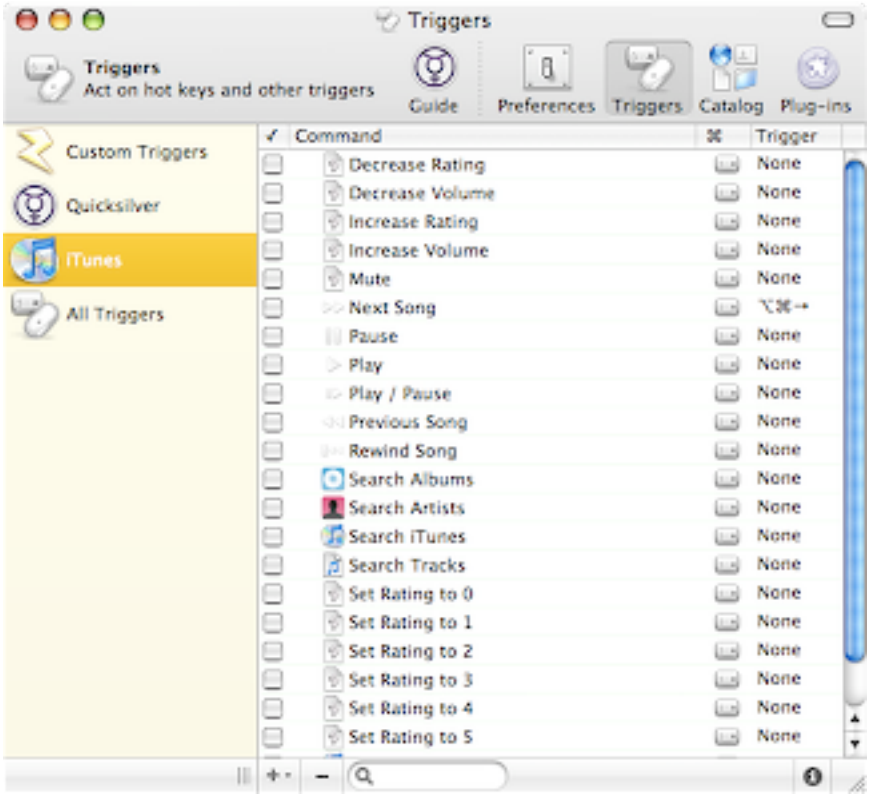


Triggers

Trigger Basics

Triggers allow you to execute Quicksilver commands without having to use the command window. Quicksilver supports executing triggers by typing a HotKey, clicking or dragging the mouse, or (if you have the Abracadabra plug-in installed) making a mouse gesture. Triggers are available whenever Quicksilver is running and you do not have to activate Quicksilver (with ^-space) to use them. Triggers make it easier to control iTunes, launch applications, perform web searches, or do anything else Quicksilver can do. You can create as many as you want but you probably want them only for the operations you do frequently.

Your trigger configuration is stored in ~/Library/Application Support/Quicksilver/Triggers.plist. This is useful to know if you want to have the same configuration on several machines, just copy the file between the machines while Quicksilver is not running.

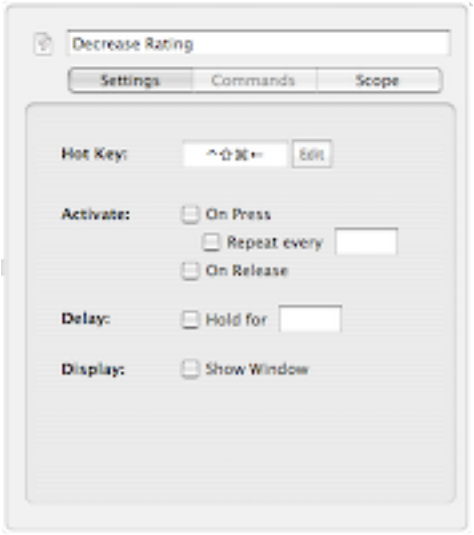


You define triggers in the Triggers Preference pane. You can go to this pane directly by activating Quicksilver and typing ⌘'. You'll see several sets in the left side of the preference pane. Some triggers are predefined by Quicksilver itself or various plug-ins, e.g., you'll find several in iTunes, and two in Quicksilver. Triggers you create will be in the Custom Triggers set.

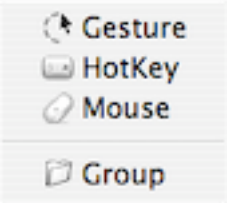
Prior to B51 the Constellation plug-in installed its own trigger set with three triggers defined, but it no longer does so..

Each trigger is shown by its name which is usually a combination of the object and action to be invoked. The checkbox shows if the trigger is active. The ⌘ column shows the type of trigger, in this case they are all activated by a

HotKey which is shown in the Trigger column (some are not assigned to keys). If you select a trigger and click on the ⓘ button at the bottom a drawer is revealed with various settings for the trigger. The settings tab looks different for each type of trigger (Gesture, HotKey, or Mouse). Shown here is the one for a HotKey. You set the HotKey by clicking the Edit button and then typing whatever key you want. You can change the name of the trigger in the top text field. There are various other options to tailor the keystroke to behave exactly the way you want. These are described below.

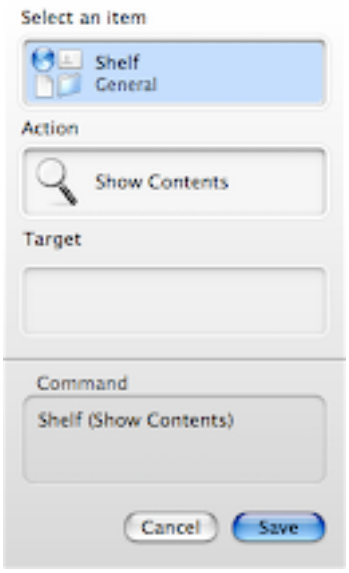


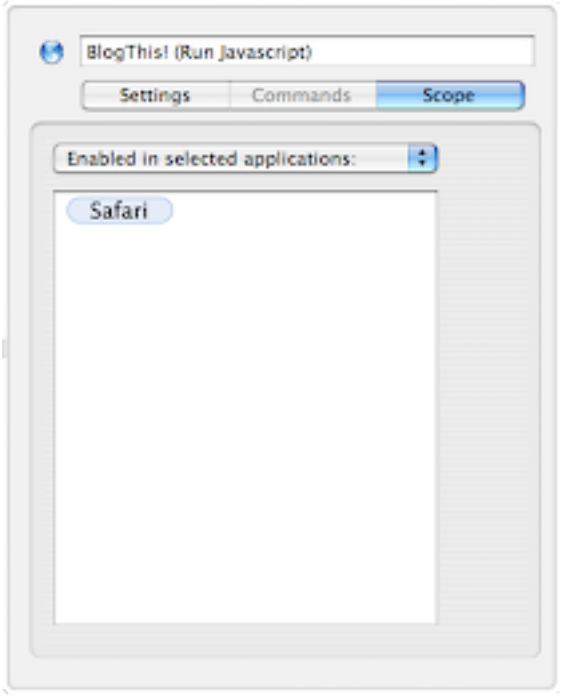
You can change some options by clicking in the main trigger window. If you click on the command name you can get an edit field to change the name of the trigger. If you click in the Trigger column you can set or change the HotKey. If you click on the icon in the ⌘ column you can add another means (Gesture, Mouse) to activate the trigger. Though once you create an additional way, there doesn't seem to be a way to delete it without deleting the entire trigger.



You create a new trigger by clicking the + button at the bottom and selecting a trigger type from the pop-up menu (Gesture, HotKey, or Mouse, depending on installed plug-ins). The Group option is just a way to collect triggers in a group or folder. They don't perform any function other than helping you organize a lot of triggers. You can't activate all the triggers in a group at once. Create a Group from the + menu and drag triggers into it.

Regardless of which type you choose, a special command window appears (populated with the last command you performed) to let you define the command for the trigger. Enter the command and click Save. Then open the drawer to the Settings tab to assign a HotKey, mouse click, or gesture to use to activate the trigger. If you're creating a HotKey trigger, just click in the Trigger column for this new trigger, the drawer will open and you can just type the HotKey as if you had clicked the Edit button.





By default, triggers are available whenever Quicksilver is running, regardless of what application is active. They can also be limited to function only when a certain application is active or when a certain application is not active. You do this by opening the drawer for a trigger and choosing the Scope tab. The default is “Enabled in all applications” but you can also choose from the popup “Enabled in selected applications” or “Disabled in selected applications”. For the latter two you can type the name of the application into the large box and type `tab` after the name to have it turn into a blue button. You can then enter another application if you choose. Unfortunately as of B51, you cannot drag and drop an application into the pane.

One place I use this is for a BlogThis bookmarklet in Safari. Like any bookmarklet it is javascript, which in this case opens a small browser window with a blog entry form and the current URL filled in. Since Safari treats it as a bookmark I can’t easily assign a shortcut to it. So I have a Quicksilver HotKey trigger which opens that bookmark (it’s in the Catalog with the other Safari bookmarks) and I have the scope set to work only when Safari is active because it’s not useful in any other application. This way if I accidently type the HotKey, nothing will happen. You should be able to configure different commands on the same HotKey scoped to different applications but in B51 it doesn’t seem to work.

Note if you have triggers that use certain features, be cautious about deleting the plug-ins that supply those features. In particular if you have triggers using mouse gestures from Abracadabra (see below) and if you remove the Abracadabra plug-in, the trigger panel may display oddly with some blank lines and missing icons. To correct this, reinstall the plug-in (Abracadabra in this case), remove the triggers, and uninstall the plug-in.

The following sections discuss each type of trigger with many examples. The commands used will come from many different plug-ins not covered here. If you want more information on them, look in the relevant sections in Part III of this manual. While you have a lot of freedom in defining triggers anyway you want, some commands are better suited to certain trigger types.

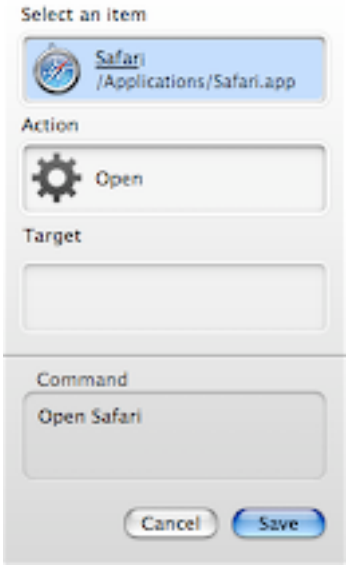
HotKey Triggers are the most commonly used and are described first. Proxy objects make triggers far more powerful and are described next. Mouse triggers are covered next along with Abracadabra gestures. Related to using the mouse are Constellation menus and regular context menus, so they are described net. Finally using and saving command objects, and droplets is described.

HotKey Triggers

If you enter a complete command into the trigger, like choosing an application and the **Open** action, it will be run when you execute the trigger. If you choose just a partial command, like a web-search and the **Search For…** action, when you execute the trigger a command window will appear with as much filled in as your trigger defines. You can’t have gaps (e.g., you can’t leave

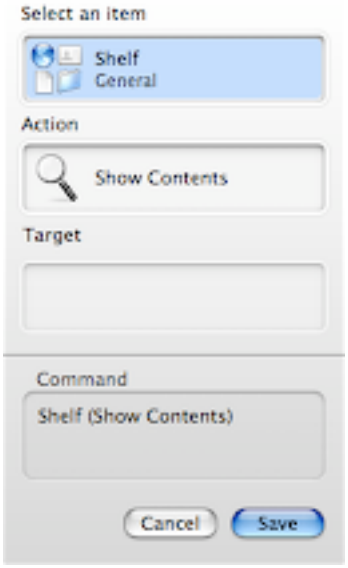
the first pane blank and use the **Get Info** action). This is discussed more below. Triggers are given a default name based on the command they represent. Here are some examples:

Safari (**Open**)



The most common simple trigger created is to open an application. Shown here is a trigger to open Safari, it's just the Safari application in the first pane and the **Open** action in the second. The argument is blank. Notice that when the action is **Open** the default name for the trigger is Open followed by the application. Usually the default name is the object followed by the action in parenthesis. **Open** is special, probably because it's so common. If in the Extras Preferences (available when you enable advanced features) you have Application reopen behavior set to activates, then using this trigger when Safari is already running will make it the active application.

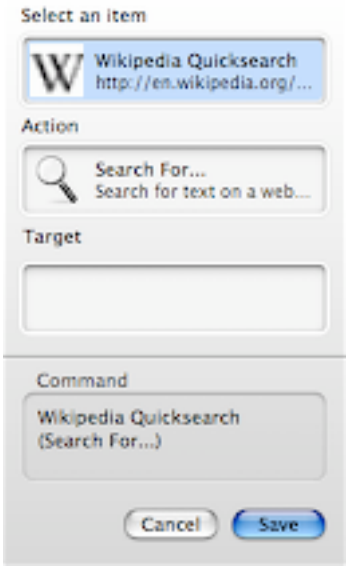
You can obviously make similar triggers to open your favorite applications. Some people create other triggers like Safari (**Quit**) or Safari (**Hide Others**) and put them on the same keys but with different modifiers (e.g., ⌥ or ⇧). Others just use the standard OS X shortcuts of ⌘Q and ⌥⌘H to run these on the current application.



Shelf (**Show Contents**)
Clipboard History (**Show Contents**)

The Shelf and Clipboard History are enabled with the Shelf Module plug-in and the Clipboard Module plug-in respectively. Each enables a window that you typically access with the mouse (see Clipboard and Shelf for details). To get keyboard access to these, first enable the objects in the Catalog under Quicksilver and Shelf & Clipboard (check the box and click the rescan button). If you select these objects in the first pane, you can type → or / to open a new results list with their contents. The **Show Contents** action is the equivalent of this and since it's an action, unlike using →, you can use it in a trigger. So this trigger will open the shelf in a results list, with the first item selected, and let you type to select any item on the shelf (or the Clipboard History if you use that object). You can also use the **Search Contents** action which is very similar but won't select the first time and won't show a results list at first. I prefer **Show Contents** because it has more feedback. You can also use this technique to create a trigger to anything you can type → into, such as Address Book, iTunes, your Documents folder, etc.

Wikipedia Quicksearch (**Search For…**)



Some actions take an argument in the third pane and triggers can use these too. The **Search For…** action will search some web site for the text entered as an argument. See the Web Searches section for the details of using this action. A trigger for a commonly used web search, such as Wikipedia, is very useful. If you specified the argument in the trigger it would search for the same text each time it's run. However, if you leave the third pane blank, then when the trigger is run, Quicksilver will open a command window with the first two panes filled in (in this case with Wikipedia Quicksearch and **Search For…**), and the third pane selected, ready for you to type the query. Quicksilver is also smart enough to realize that the **Search For...** action wants a text argument and puts the third pane in text mode for you. It even fills in the default text from the OSX Shared Find Clipboard (which you can set in many Cocoa applications with ⇧⌘E). Note, if the third pane isn't empty when you create the trigger and you want it to be, you can type ⌘X to cut out whatever is there.

I have this trigger configured on the HotKey ^⌘W. I type ^⌘W, the query and `return` and the browser opens with the search results. If I want to search for some text in a document, I can select the text, copy it with ⌘C, type ^⌘W and then paste the text with ⌘V and type `return`. That's pretty easy, but in the next section you'll see how a proxy objects reduce this to one step.

Proxy Objects

For this section make sure advanced features are enabled in the Application Preferences and also make sure Proxy Objects are enabled in the Catalog under Quicksilver. Proxy objects are a special kind of object that stands for (or proxies) something else that changes. Having an unchanging object represent something that changes allows you to use it when you don't know what it will be ahead of time. Here are some more example triggers to make that clearer

Current Web Page (**Paste**)

This example lets you paste the URL of the page Safari is displaying into any other application without switching applications and without having to do a copy step. I have it bound to ^⌘F and use it all the time to paste an URL into mail messages, chats, documents, iCal events, etc. The magic is because of the Current Web Page proxy object used in the first pane. It acts like a placeholder for the URL Safari will be displaying when I use this command. When the command is run, this proxy object asks Safari what it's currently displaying and uses that in the command. Proxy objects allow your triggers to be dynamic and are one of the most powerful features of Quicksilver.

Note the Current Web Page proxy object is poorly named. It should be called Current Safari Page because it doesn't use the default browser but only Safari (it's defined in the Safari Module plug-in so install that if you don't see it). If you use OmniWeb you can use the OmniWeb Active Page proxy object. Unfortunately as of B51 there are no equivalent proxy objects for other browsers.

Current Selection (**Find With…**) Wikipedia Quicksearch

The Current Selection proxy object gets the selection from the frontmost application. It uses a service internally so it only works with Cocoa applications (e.g., Safari, Mail, Preview, iWork, etc.). That's most modern Mac apps but the notable exceptions are MS Office and Firefox (try Camino instead). Still, it's remarkably useful.

It should be possible to use this proxy object as the argument in the Wikipedia search trigger above, so that you could select text in some Cocoa application, invoke the trigger, and have the search run. However in B51 I can't get Current Selection into the third pane (even with pasting). But the **Search For…** action has a complementary action called **Find With…** that takes the query text as the object and the search site as the argument. I have a trigger of the Current Selection proxy object with the **Find With…** action and the Wikipedia Quicksearch argument bound to the HotKey ⇧^⌘W.

It's useful to assign triggers to HotKeys in some pattern so they are easier to remember. I use ^⌘W to search Wikipedia for text I enter, and I add ⇧ to the sequence to search for selected text. I have similar pairs of triggers for other quick searches such as Google (^⌘G), IMDB (^⌘M), and Amazon (^⌘Z). These triggers are some of the most frequent ways I use Quicksilver.

If you're now convinced that proxy objects are amazing, you probably want to know what other proxy objects are available. Open the Catalog, select the Quicksilver set and then select Proxy Objects source, click the ⓘ button at the bottom to open the drawer and look at the contents tab. Or you can look at the table below which organizes them into similar categories. Note that some third party plug-ins might install additional proxy objects.

iTunes	Finder	iPhoto
• Album Now Playing • Artist Now Playing • Current Playlist • Current iTunes Selection • Selected Playlist • Track Now Playing	• Finder Selection • Mounted Disks • Network Disks • Removeable Disks	• Current iPhoto Selection • Selected iPhoto Album
Quicksilver	**Mouse**	**Applications**
• Clipboard Contents • Current Selection • Current Web Page • Last Command • Last Object • Quicksilver Selection	• Mouse Trigger Dragged Object • Droplet Item	• Current Application • Previous Application • HIdden Applications • Visible Applications

It's obvious from the name what most of them do, with a few exceptions. The Current Web Page proxy object really should be called Current Safari Page. The Mouse ones are covered below. The Quicksilver Selection proxy object represents what is in the object pane (aka first pane). This allows you to assign HotKeys to particular actions (or actions and arguments). E.g., create the trigger Quicksilver Selection (**Get Info**) and assign it to ^⌘I. Now you can activate Quicksilver, select a file object in the catalog or navigate to one in the filesystem and type ^⌘I to bring up the Finder's Info pane of that file. You don't even have to tab to the action pane. You might think you could scope this trigger to only work in Quicksilver but that's probably not what you want. If you do so, the trigger will only work if Quicksilver is active and frontmost, with its menu bar displayed. When you activate it with ^-space, this isn't usually the case. If you're in some other application and activate Quicksilver, that other application is still considered active so the trigger isn't available. Therefore you probably don't want the Get Info trigger on the Finder's binding of ⌘I since that with interfere with every application which usually defines that for toggling italics.

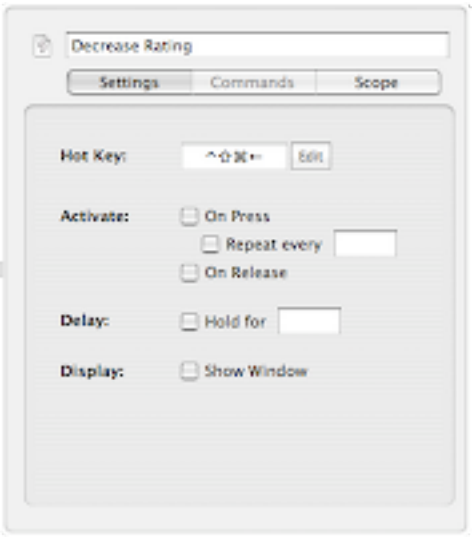
Sometimes when creating triggers with proxy objects, the action you want doesn't appear in the list. As you've probably noticed, Quicksilver normally narrows down the action list to only those that are relevant. Sometimes proxy objects confuse it because they represent something else. If this happens you can work around it with copy and paste. It's not obvious but if you bring up a regular command window (with ^-space), when you tab to the action pane you can type ⌘C to copy the action. You can then paste it into the second pane of the trigger definition window with ⌘V. You can also cut an item out with ⌘X. This is useful when creating commands that do a web search with the **Search For…** action and you want a blank third pane. There are many other proxy objects giving you access to interesting things to use in triggers such as Track Now Playing (in iTunes), Current Application, Previous Application, Finder Selection, etc. Experiment to see what you find useful.

More on Triggers

The possibilities for triggers seem endless. Here are some triggers people have posted on the Quicksilver forums as their favorites:

- Documents (**Open**) - the Finder doesn't have a shortcut for this
- Finder Selection (**Go To Directory in Terminal**)
- Finder Selection (**Rename…**) - Cocoa text editing bindings are available
- Finder Selection (**Scale Image…**) 400
- Finder Selection (**Open With…**) TextMate
- Current Application (**Menu Bar...**)
- Current Application (**Menu Bar Items...**)
- Current Application (**Show Menu Items**)
- Current Application (**Hide Others**) - though ⌥⌘H already works in most every application
- Previous Application (Open) - go back to what you were doing last, toggle between two apps
- Artist Now Playing (**Show Contents**)
- Browse Albums (**Search Contents**)
- `killall Dock` (**Run Command in Shell**) - or any other frequent shell command
- pbpaste | pbcopy (**Run Command in Shell**) - copies the clipboard without formatting
- `date` (**Run Command in Shell**) - acts like a clock
- Current Web Page (**Open With…**) Camino - or any other browser
- Removeable Disks (**Eject**) - great to clean up mounted disk images
- Network Disks (**Eject**)

There are some limits to triggers that you can create. You can't save a trigger with a dynamic action. Dynamic actions are those that ask the application at runtime for their content. So the Current Application (**Menu Bar…**) trigger above works, but you can't do Current Application (**Menu Bar…**) Help. Also actions that appear only for particular applications can't be saved as triggers, so you can't do Mail (**Get New Mail**). For some of these you can physically create the trigger but they won't save properly and will be broken on Quicksilver relaunch.



The Settings tab in the drawer lets you set various options for a HotKey trigger. "Hot Key" lets you set the key. "Activate" lets you determine if the trigger activates when the key is pressed or released, and lets you set it to repeat if the key is held down for a period of time (useful for a fast-forward script). With a "Delay" set, the trigger won't activate until the key has been held down for the number of specified seconds. The "Display" "Show Window" option shows a command window when the trigger is activated which is useful to see feedback that a trigger has executed.

A delay can be useful for some dangerous commands that you don't want to execute if you accidentally type the key. E.g., if you had a HotKey to Run the Quit All Visible Applications script from the Extra Scripts plug-in, you might configure it to only execute if you hold the key down for 2 seconds.

TODO: You can define combinations of triggers to do one operation on immediate press and a second on a press an hold. E.g., you can define ^⌘I to open your Inbox and a second ^⌘I trigger to Get New Mail if you hold it down for 2 seconds. (one key for mail, holding it checks for new mail and then opens it) You can also configure triggers that use two scripts, one on start, one on finish.

If you create a lot of HotKey triggers it's probably best to use some system to remember them. Some people put applications on their function keys, others use mnemonic keys like ^⌘S for Safari. In examples above ^⌘W did a Wikipedia search and adding a ⇧ to the HotKey did the same thing with the current selection. Someone posted on the forums a scheme using ⌥⌘*letter* and ^⌥⌘*letter* for Applications, ⇧⌥⌘*letter* for Folders , ⇧^⌘*letter* for iChat buddies. The important thing is to find something that works for you. It can be difficult to work around keys used by applications you use (particularly if you use TextMate or Emacs), but it can be done. The next sections describe using triggers with the mouse instead of HotKeys.

Using the Mouse

Drag and Drop

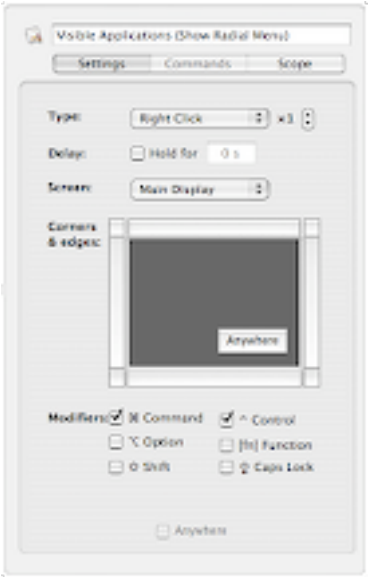
While it does involve switching between keyboard and mouse, the Quicksilver command window can be a convenient drag & drop source or target. E.g., with an Open File dialog up from some application, rather than navigating in that window you can activate Quicksilver, type to select the file you want to open and then drag it from the first pane into the Open File dialog. To make this a little more seamless there is an AppleScript Action you can download from the Quicksilver site that add an action to send the file to an open or save dialog, so you can avoid

the dragging. To move a file from the desktop to a folder, activate Quicksilver and bring up the folder in the first pane; then just drag the file from the desktop to the folder in the first pane.

You can bring up an email address and drag it into the To: or CC: field of a mail message. Mail.app doesn't recognize nicknames in it's matching so using Quicksilver for this might be convenient. Note you have to drag an email address for this to work, not a contact. You can also drag a file into a message as an attachment.

If the application you're dropping into doesn't accept the item your dragging from Quicksilver you'll probably get the name of the item dropped in. E.g., if you drag a file into a web form you'll find the file name with its full path entered into the form. Do the same thing with a contact to make sure you get a difficult name spelled correctly. Drag a postal address from a contact into a document or a field in an online mapping service (note, the newlines might prevent a web field from getting more than the street address). With proxy objects you can drag the Current Web Page URL or the iTunes Track Now Playing.

Mouse Triggers



With the Mouse Triggers plug-in installed triggers can also be assigned to the Mouse. The Settings tab of the drawer for the trigger is different for a mouse trigger. You can assign triggers to a number of mouse clicks, with or without modifier keys. Note that the Anywhere button in the desktop drawing is actually a button. If you want a trigger to work if you ⌘-right-click anywhere in the window you have to click the Anywhere button for it to work. Also as of B51 it seems the Anywhere checkbox at the bottom is always greyed out.

You can also activate a trigger when the mouse enters or exits an edge or corner of the screen or if you drag something to an edge or corner of the screen. If you have multiple monitors connected to your Mac you can choose if the trigger will work on all displays or on a particular one. I have Current Application (**Hide**) set when I mouse to the lower left corner. It gives me an easy mouse-based way to hide an application.

Dragging triggers work well with the Mouse Trigger Dragged Object proxy object. E.g., configure these when dragging to a corner or edge for your commonly used application or folder:

- Mouse Trigger Dragged Object (**Open With…**) *application*
- Mouse Trigger Dragged Object (**Move to…**) *folder*

Here are some more advanced ones that might require additional plug-ins

- Mouse Trigger Dragged Object (**Email to…(Compose)**)
- Mouse Trigger Dragged Object (**Compress (Create Archive)**)
- Mouse Trigger Dragged Object (**Set Desktop Picture**)
- Mouse Trigger Dragged Object (**Add Tags...**) *tags*
- Mouse Trigger Dragged Object (**Upload to Site...**) *Transmit favorite*

Note that if you configure a trigger to activate when mousing or dragging to an edge it might interfere with the shelf or clipboard windows if you have them on an edge. In such a case the trig-

ger will win and you'll need to use some other means like a key stroke to open the shelf or clipboard windows.

One thing might not be obvious, you can configure the same trigger to activate by multiple means. E.g., if you have an Open Safari trigger assigned to the HotKey ^⌘S you can also assign it to having the mouse enter the left edge of the screen, and both will work. You do this by clicking the icon of the type of trigger in the ⌘ column of the Triggers preference pane. A pop-up menu appears of the other choices and you can select another one and configure it. Now both work. This can be useful to have two ways to invoke a trigger, one when your hands are on the keyboard and another when one hand is on the mouse. I don't see a way to remove only one method from a trigger. The - button at the bottom of the screen deletes the whole trigger. For a HotKey you can always change it to not be bound to a specific key by clicking the Edit button and then the delete key. For a mouse trigger you can achieve the same effect by choosing Mouse Entered and selecting no edge or corner.

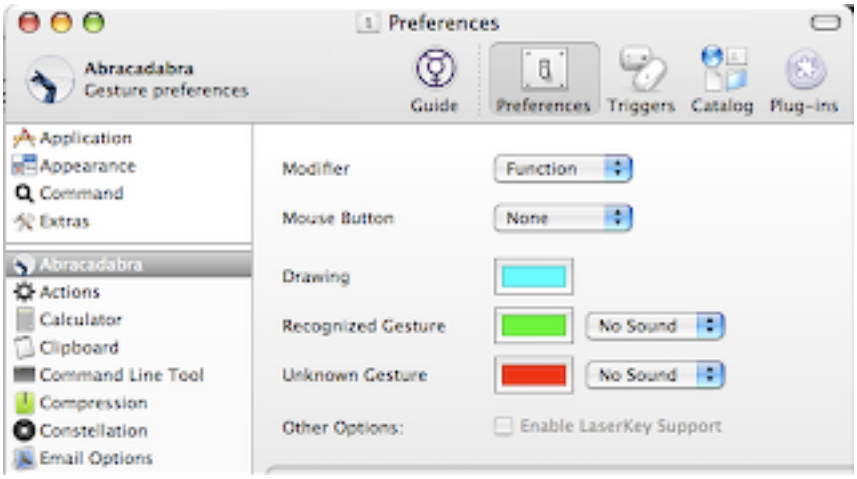
Most useful on mouse triggers are the **Show Menu**, **Show Contents Menu**, and **Show Action Menu** actions. These bring up a context menu for the object. **Show Contents Menu** creates a menu for the items that would appear if you typed → to go into the object, it's like **Show Contents** but in menu form. Items have sub-menus showing the actions you can perform on them and any sub-items they have. **Show Action Menu** shows a menu of the actions you can perform on an object. **Show Menu** shows a menu that combines the contents and actions you can perform. Mouse triggers for Shelf (**Show Contents Menu**) or Clipboard History (**Show Contents Menu**) are very useful.

Abracadabra Mouse Gestures

For this section you'll need to have Advanced features enabled. Then install the Abracadabra Triggers plug-in. This enables a new trigger type known as a Gesture to go along with HotKey and Mouse triggers. It also installs an Abracadabra preference pane.

Abracadabra lets you invoke a trigger by drawing a shape with the mouse. E.g., you can configure Quicksilver to activate the Open Safari trigger by drawing an S on the screen. You configure your shape by drawing with the mouse button down in the pane in the Settings tab of the trigger's drawer. The "Show zooming trigger window" will show a bezel window that enlarges when a gesture trigger is recognized and executed (like the Display: Show Window option of a HotKey trigger).

In the Abracadabra Preference pane you configure how you invoke all gestures. That is what mouse button you hold down and/or what modifier key you hold down while drawing any gesture. If you have a mouse with extra buttons it's probably most convenient to use one of them for this as you'll only need one hand to invoke the gesture. You can also choose the colors used to draw the shape on the screen as you draw and after you finish for a recognized gesture and an unrecognized gesture, as well as sounds to play.



There's also an option to Enable LaserKey Support which is a virtual keyboard device made by Cellulon. With it you can make gestures with your finger. Wherever I say mouse gestures here, LaserKey gestures is implied if you're one of the lucky few to have one of these devices.

Simple gestures are best as you'll have an easier time remembering them and Quicksilver will have an easier time recognizing them. You can create a gesture for any Quicksilver command (that is trigger). E.g., skipping to the next track in iTunes by drawing a line from left to right. Try a gesture for the **Quit** or **Hide** actions using the Current Application proxy object.

Gestures also go well with limiting the scope of a trigger and using the **Menu Bar Items…** action. This way you can make a trigger to invoke Safari's Back command or Mail's Reply command. This usually works best for things that don't need the keyboard, e.g., navigating in Safari. Using a gesture to bring up a Find dialog probably doesn't make much sense since you'll need to type the text to find. Expose actions work well with gestures.

An interesting idea is using the Current Application proxy object and the **Menu Bar Items…** action to invoke an action common in all applications, e.g., Undo or Close. The problem is that many applications dynamically update their menus (e.g., "Undo Typing") or slightly rename these commands (e.g., Close Window and Close Tab) so the trigger won't work in all applications. Also there's no way to select a menu command that is generic and not specific to an application, so this isn't possible.

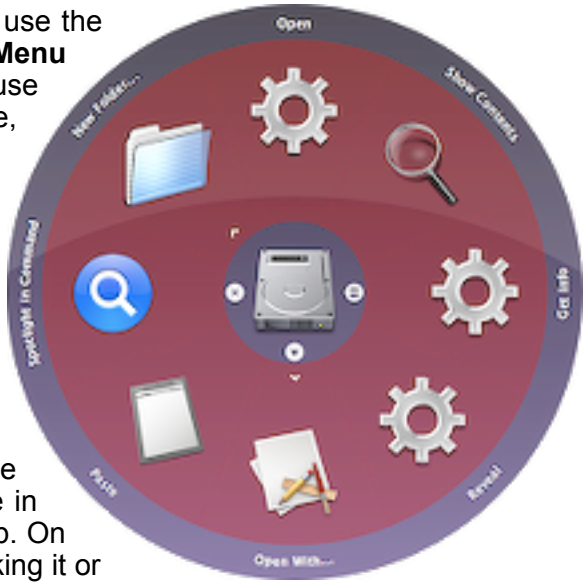
Constellation Radial Menus

Most menus on computers look pretty similar. You click on something and a rectangle appears with commands listed vertically. Move the mouse over the command you want and click and you're done. The one inefficient part is that if you want to choose the last item you have to move the mouse much farther than if you want the first item. Radial menus take a different approach. They pop up a circle around the mouse and divide the circle into pie pieces, one for each command. In this configuration choosing any item is just a different direction to move the mouse, but not any extra distance. Of course Quicksilver can use radial menus, and they look very very pretty.

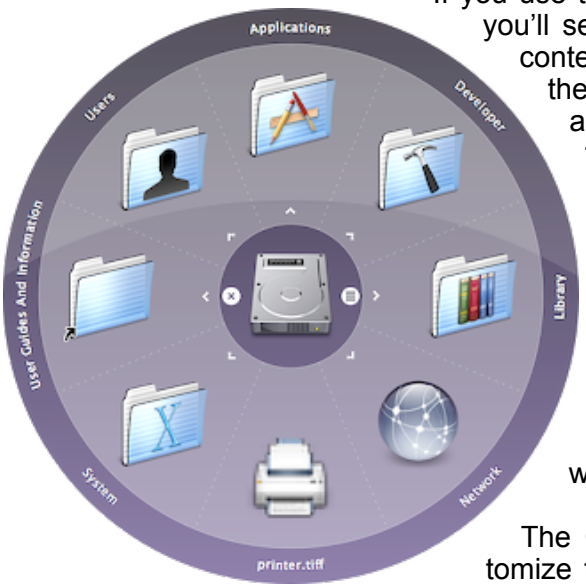
For this section you'll need to have Advanced Features enabled. You'll also need to check "Enable access for assistive devices" in the Universal Access System Preferences. In addition to Constellation you'll need to install the Mouse Triggers and User Interface Access plug-ins. Note that the plug-in installs a Missing Object Selector Handler (in Preferences, Handlers) that you can configure, but don't do that, leave it set to Quicksilver (see note below when discussing the Constellation triggers). The Constellation plug-in installs two actions, **Show Radial Menu** and **Show Radial Action Menu**. It also installs a preference pane called Constellation listed in Gen-

eral . Up until B51 it installed a new trigger set on the left of the Triggers preference pane with 3 triggers predefined.

Let's start with the **Show Radial Action Menu** action. Activate Quicksilver and bring up a folder (these examples use the hard disk) and choose the **Show Radial Action Menu** action. You'll see a menu appear around your mouse pointer with items for the actions you could choose, as if you had brought up the folder in the first pane and looked at the available actions in the second pane. Moving the mouse into one of the pie slices highlights it, click to perform the action.



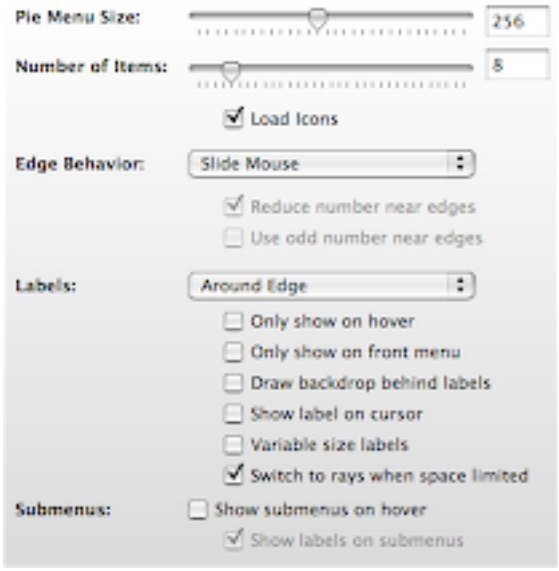
In the center to the left is an x in a circle, clicking that will close the menu. To the right is a button with 3 horizontal lines, clicking it will bring up a traditional menu of the items shown in the radial menu. If there are more items to display than fit in the pie slices a down-arrow button appears at the bottom of the center circle. Clicking it or anywhere in the bottom half of the center icon will bring them up. On additional menus an up arrow is displayed and clicking it or anywhere in the top half of the icon will go back.



If you use the **Show Radial Menu** action for the same folder you'll see not the actions you can do to it but rather the contents of the folder appear in the menu. By clicking the items you can navigate into the folders bringing up additional radial menus of the contents of the sub-folders. The little arrows at the center of the pie slices correspond to >'s in a results list, it shows that there are sub-items. If you click on one without an arrow, a menu opens as if you had done **Show Radial Action Menu** on that object. If you control-click (or right-click) on an item with an arrow in it you will pop up an action menu for that item. If you double-click on an item you will perform the default action for that item immediately. So double-clicking on a folder will open a Finder window of that folder.

The Constellation preference pane allows you to customize the look of the menu. You can set the size of the

menu and the number of slices in the pie. The Edge Behavior setting controls how the menu is displayed when the mouse is too close to an edge to show the whole menu. You can crop the menu (Slice Edge), move the mouse (Slide Mouse), or move the whole screen temporarily (Slide Screen). There are also various options for showing the labels (the above image is using "Around Edge") and whether submenus are shown when hovering.



TODO: The trigger set is not installed in B51, will it come back? Should I remove the following 3 paragraphs?

The Constellation plug-in installs a trigger set called, Constellation. There are three pre-configured triggers: Show Radial Menu for iTunes (on F6), Show Menu Bar Contents (Change Helper) (on F7) and Email Dragged Items (on mouse drag and drop into the lower-right corner). The first shows choices as if you had typed → into iTunes (Recent Tracks, Browse Artists, Genres, Albums, etc.). It's just the iTunes object with the **Show Radial Menu** action. You can click on these to open other radial menus to browse your music library. With album art showing it's very pretty. Control-clicking brings up an action menu with Play and Party Shuffle actions.

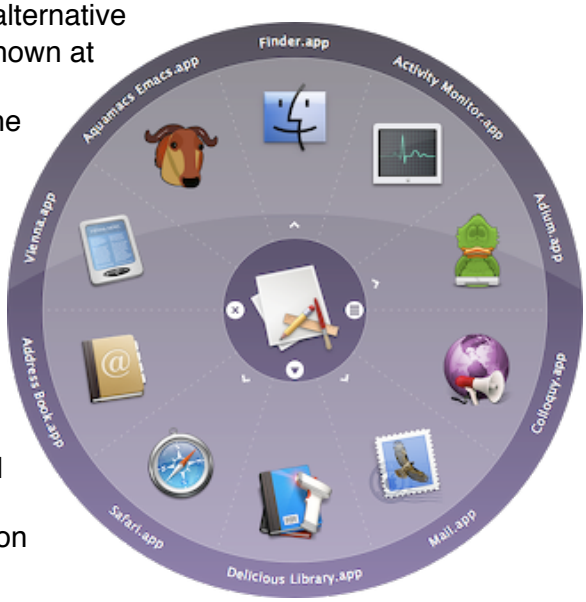
The second pre-configured trigger pops up a menu that shows the current application's menu items (like File, Edit, etc.). If you like the idea of radial menus you'll like this. It's somewhat special in that there seems to be no way to create this yourself. It's like the **Show Menu Items** action but in a radial menu. Note: this trigger only works if the Missing Object Selection Handler is set to Constellation. The problem is that setting it to Constellation can (as of B51) cause Quicksilver to crash on some occasions (see Bug 1017). In particular I have problems when using Web Search triggers with a blank third pane. My understanding is that this handler was originally a hack and I suspect it will be removed in a future version of Quicksilver.

With the third pre-configured trigger you drag text or a file to the lower right corner and it pops up a radial menu of Contacts to send to. It's populated the same as the results list in the third pane of a command with the **Email to…(Compose)** action so it's a list of commonly used contacts. The trigger is the proxy object Mouse Trigger Dragged Object with the action **Email to…(Compose)** but it pops up this radial menu (which with pictures in your Address Book entries is very pretty).

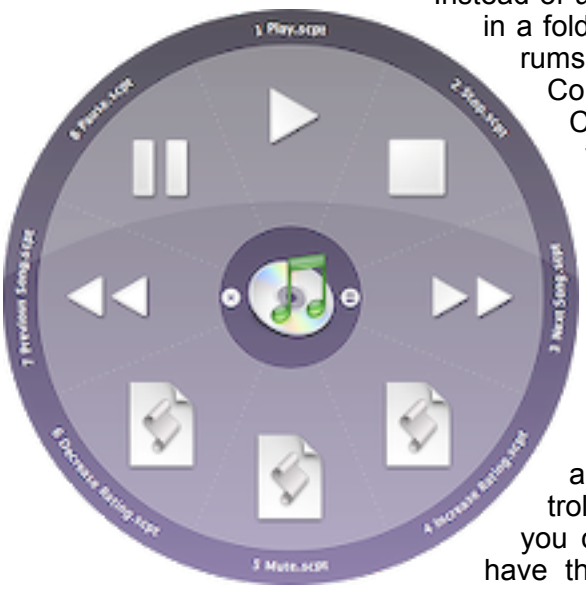
Those are the basics. You can of course use radial menus for lots of things and create your own triggers for commonly used ones (these will be in the Custom trigger set, not the Constellation trigger set). If you create triggers you probably want them bound to the mouse or at least to HotKeys easily typed with your non-mouse hand. What I think works well for radial menus are things that don't change position (which isn't so easy to arrange for) so you can learn them and things with clearly differentiated icons so you can recognize them quickly.

Here are some examples of objects to use the **Show Radial Menu** action on. Many of these are proxy objects which you must enable in the catalog under Quicksilver, Proxy Objects.

- Visible Applications - this is a cool mouse-based alternative to the ⌘-tab method of switching applications. (shown at right).
- Track Now Playing - Shows a four part menu of the album, artist, genre, and composer of the current track, ready for easy browsing.
- Artist Now Playing - this shows all the albums by this artist you have in iTunes (with album art)
- Clipboard History
- Shelf
- Finder Sidebar Items
- Desktop
- Comma trick - create a trigger collecting several things you use frequently via the comma trick and use **Show Radial Menu** to display them. Works well for contacts, bookmarks, playlists, presentation documents, etc.



I haven't thought of any good uses of **Show Radial Action Menu** as a trigger.

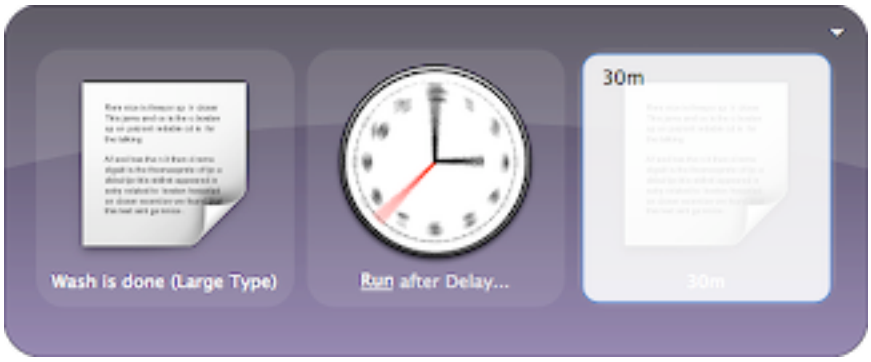


Instead of using the comma trick you can collect useful things in a folder. Here's a fun idea I got from the Quicksilver forums. Create a folder and call it something like iTunes Controls. Do a Get Info on it and on the iTunes.app. Copy the iTunes icon to the new folder. Now activate Quicksilver and bring up the Play.scpt that controls iTunes and use the **Reveal** action on it. This will open a finder window inside the iTunes Module.qsplugin showing the scripts. Now copy them to the iTunes Controls you created. Remove the ones you don't want and rename the remaining ones putting number at the beginning of the name to order them as you want. Now bring up the iTunes Control folder and use the **Show Radial Menu** action to bring up the controls as shown. Save this as a trigger to make these controls easily accessible. If you want to be more creative you can add your own icons for the scripts that don't have them. Jacob Rus has put some icons and extra scripts together you can download here.

Command Objects and Droplets

Commands in Quicksilver can be represented as objects in Quicksilver. To create a command object, activate Quicksilver and enter a command as you normally do, but instead of typing `return` to execute it, instead type `^-return`. You'll see a new command window appear in which the object is a new command object of the command you entered and a default action which is probably **Run**.

Other actions that work on commands include **Run**, **Run after Delay...**, and **Run at Time...**. The first, **Run**, is obvious and not particularly useful since just executing the command without creat-



ing a command object is easier. The other two are useful as alarms when used with actions like **Large Text** or **Display Dialog**. A delay period is entered as a decimal number followed by an `h`, `m` or `s` to indicate hours, minutes or seconds. E.g., `2h`, `10m`, `90s`, or `1.5h`. A number without a suffix is interpreted as minutes. You can combine these separated by spaces such as `2m 30s`. You can also enter a time format such as *minutes*`:`*seconds* or *hours*:*minutes*:*seconds*. For example `9:35` or `2:11:30`. I believe you can also enter decimal numbers with colons like `1.5:10:00` for 100 minutes but that is probably not useful.

Times for **Run at TIme...** are typically entered as a digital clock, *hh*`:`*mm*`:`*ss* although it uses OS X's natural language date parser so things like `3pm December 15, 2008` or `next Tuesday at dinner` are supposed to work.

There's also an **Add Trigger** action but in B51 all it seems to do it open the Triggers preference pane. It actually does add a HotKey trigger representing the command but it doesn't show up in the Triggers list until you restart Quicksilver. You can then assign a HotKey to it.

The **Save Command to File…** action takes a command object and saves it in a file. You enter a folder name as the argument and Quicksilver saves a file there with the extension `.qscommand`. If you open the resulting file (e.g., by double-clicking it), it will run the command. Quicksilver makes running commands simple and triggers make running command even easier. But for commands that you might run a few times but not enough to bother creating a trigger for, the **Save Command to File…** action can be useful. E.g., if your working on creating a document and you want to send several drafts to a group of people (using the comma trick) it's easy to save this (complicated) command in a file to rerun.

With a saved command on your Desktop you might want to be able to drag a file onto the command and have it run on that file. This can be done, and this type of saved command is known as a Droplet. To create a droplet, use the Droplet Item proxy object in the command and save it using the **Save Command to File…** action. The command file will have a special icon and a `.app` extension. Unfortunately in B51 these don't seem to always work and sometimes create running Droplet processes that have to be quit before you can delete the files.

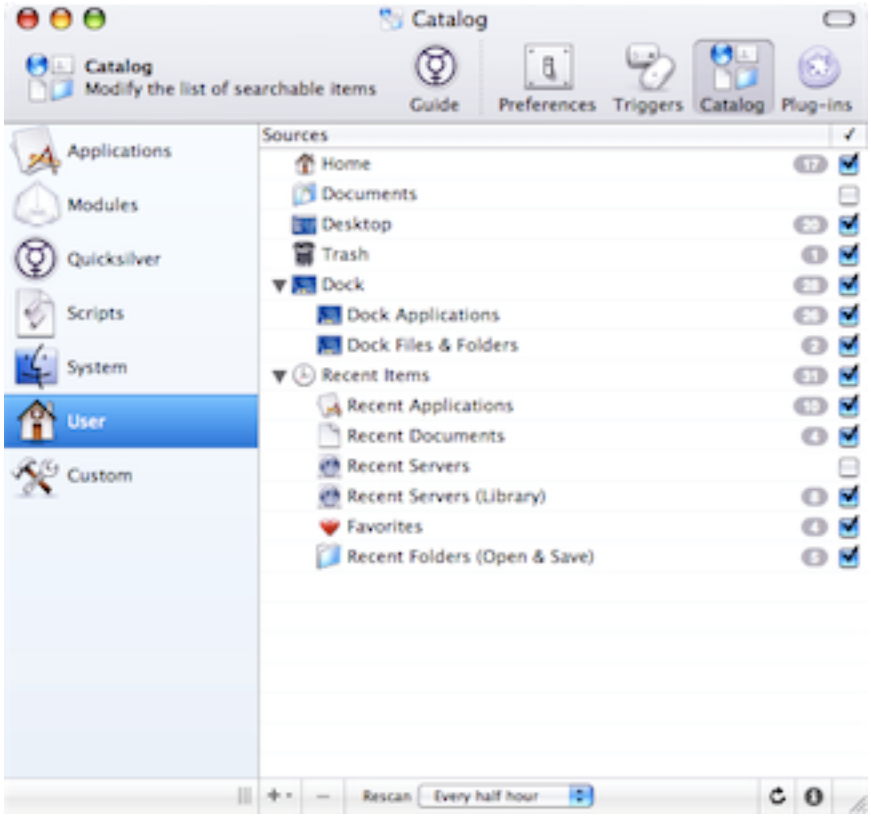
Part III - Features

Files and Folders

The most commonly used items on your computer are files and folders. Any file or folder can be selected as an object in Quicksilver. As I write this manual, the file is called Quicksilver.pages and it's on my Desktop. Since my Desktop is in the catalog I can select it in the first pane by just typing its name. As I type `Qui` it appears with the default action of **Open**. If I type `return`, Pages will open the file. I can select the Desktop itself by typing `Desktop` (or as much of it as I need to match it) and since it's really a folder, the default action is **Open**.

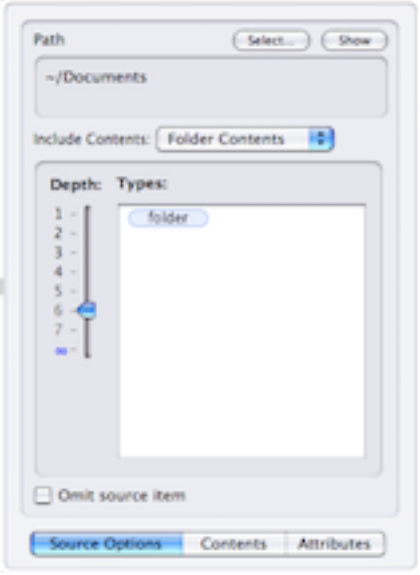
Quicksilver isn't limited to only what's in the catalog. If a folder is in the object pane, use → to navigate into it. Then the object pane changes to a results list of all the child files and folders. Since it's common to separate folders with / it's an equivalent key to → to navigate. Hold down the ⌥ key when navigating to make the results list include hidden files (those that begin with a period). This also enables browsing into packages (like the Finder's Show Package Contents command).

Quicksilver also understands the tilde (~) as the Unix shortcut for the home directory. Since the Desktop is just a folder in the home directory I could navigate to this manual's file (at `~/Desktop/Quicksilver.pages`) by typing `~/De/Qui`. Quicksilver also interprets a backquote (`) as a tilde (on US keyboards the tilde is a shifted backquote). Back up a folder hierarchy by typing ← or ? (i.e., a shifted / on a US keyboard). This can backup all the way to the root directory. Type / and hold it down for a second to select the root directory. Disks are also included in the catalog so type their name to select them. As of B55 Quicksilver has some problems with folders with periods in their name. If you are experiencing problems try removing the period.



Several different folders are listed in the the catalog by default to find files. Under the User set are catalog sources for the Home, Documents and Desktop folders. Under Dock are sources for Dock Files & Folders and under Recent Items is a source for Recent Documents, Favorites and Recent Folders (Open & Save).

Since there are probably a lot of files on the machine, to avoid using a lot of memory and CPU Quicksilver doesn't scan them all into the catalog. Instead the folders only scan to a configured depth. The default Home source is only indexed one level down and the default Documents source only 2 levels. These sources are not configurable themselves, but an editable copy can be made. In the info pane for the source, go to the attributes and click on the Create Copy button; this will make an editable copy of the source in the Custom section of the catalog. See the Catalog section for instructions on doing this.



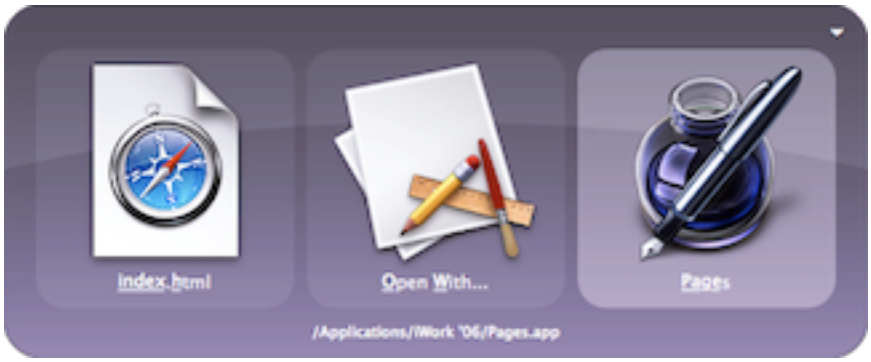
Do not just create copies of the sources that scan to infinite depth. This will probably cause Quicksilver to use a lot of memory and a large percentage of the CPU to scan the entire drive every few minutes. Instead only scan the folders that are really used. Don't scan the Home folder more than one level down. ~/Library/ will probably just clog the Catalog with many unused items, and ~/Music/ and ~/Pictures/ are better scanned with the iTunes and iPhoto plug-ins.

Since Quicksilver doesn't allow file exclusions to sources rearranging some files to move unused files out of a hierarchy might make the scanning more efficient. Another trick is to scan only folders deeply but not all the files, this allows easy access to the folders and then allows → to be used to navigate. I have a copy of my ~/Documents/ source set to scan 3 levels down (which finds 2000 files) and another source set to find only folders under ~/Documents/ set to scan 6 levels down (which finds 250 folders). If rearranging files is an option, it might help to arrange different folder hierarchies to be at the same level in an ancestor folder.

As of B51 Quicksilver has some incompatibilities with Apple's FileVault. People have reported problems with using various features described below including the **Move to…** action, ⌘G to grab files from FileVault protected folders as well as using the File Tagging plug-in. There isn't a known solution so at this point Quicksilver and FileVault are incompatible for those functions.

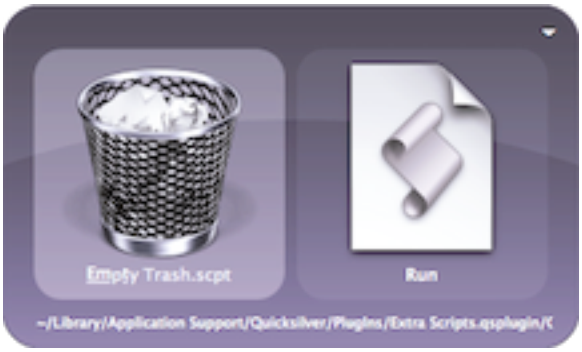
Basic File Actions

The default file action **Open,** opens a file in its default application; the same as double-clicking on the file's icon. The default action for a folder is also **Open**, and just like double-clicking on its icon, it opens the folder in a Finder window. There are also other actions available. Use **Open With...** to open the file with a different application specified in the third pane. The third pane will have a results list of applications to choose from (if it doesn't, disable the TextMate Ranker plug-in if it's installed, it's known to break **Open With…**). If the desired application is not there, navigate to it starting with / or ~ or by selecting the application in the Finder and using ⌘G. If the third pane is in text-mode, exit text mode by typing `esc` and navigate to the desired application.



Another way to use **Open With…** is to select the file (or files) in the Finder, then activate Quicksilver and select the application to use in the first pane. Then instead of tabbing to the action pane, just type ⌥⌘G. The command window will change so that the file is in the first pane, the action is **Open With…** and the application is in the third pane. It's a little convoluted but it is quick.

The default action for a file object is **Open**. For scripts however this could be unclear, should they be opened in an editor or executed? Quicksilver solves this by having the **Open** action open them in an editor and having several **Run** actions. More about them in the section on the Terminal but as an example, the Extra Scripts plug-in includes an Empty Trash script (in the Catalog under Modules, Scripts) to be used with the **Run** action.

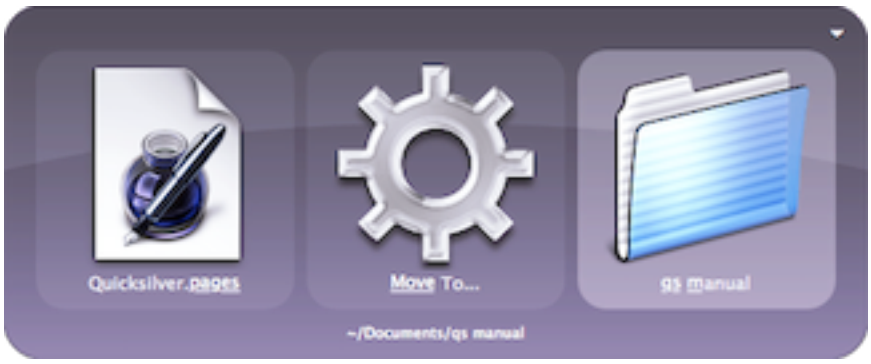


It might be more common to use with an application, but the **Open at Login** action can also set a file to open automatically at login time. The item is added to the list in the Account System Preferences under the Login Items tab. All items listed in that pane are started at login, the checkbox only indicates if they are hidden when started. The **Do Not Open at Login** action removes something from the list. Both of these actions are disabled by default, enabled them in the Action preferences.

The **Reveal** action opens the containing folder and selects the item. This is a very easy way to find what folder an item is in, particularly when finding the item directly via the catalog instead of

navigating down to it. I do this often to find applications (which are in my catalog) and might be in /Applications or ~/Applications or a couple of levels down. In fact it's so useful Quicksilver has a built-in key binding for it. With a file in the first pane and the action set to **Open** (it's probably the default), typing ⌘-`return` performs the **Reveal** action.

Quicksilver includes three methods that take a file and return a form of the file's name. All return an absolute path on the machine to the file. **Get File Location** returns a mac-style path with colons**. Get FIle Path** returns a unix-style path with directories separated by slashes. **Get FIle URL** returns a file:// URL to the file. The last two actions are disabled by default, enabled them in the Action preferences. All three actions show the path in a new command window. Use this to copy and paste a file's path somewhere. The Terminal section describes using the **Go To Directory in Terminal** and **Go To Directory in iTerm** actions to open a terminal shell in the folder specified in the first pane as the current working directory.



The actions **Rename...**, **Copy to...**, and **Move to…** are very commonly used. For all of them, the third pane opens to enter the new name or location. Note that the **Move to…** action when used to move a file between drives internally does two actions, a copy and a delete of the original. In some cases, such as having read access but not write access to the file, it might behave like a copy, because it can't delete the source file after the copy.

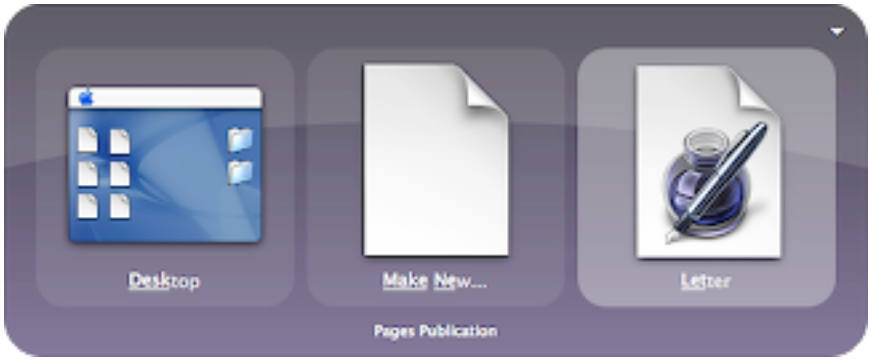
The ⌥⌘G trick also works with the **Move to…** action. In the Finder select the file or files to move, activate Quicksilver and bring up the destination folder in the first pane. Then instead of tabbing to the action pane, just type ⌥⌘G. The command window will change so that the files are in the first pane, the action is **Move to…** and the destination folder is the argument. Type `return` to execute the command. Mouse users might prefer to just drag the file from the Finder to the folder when it's in the first pane.

Before 10.5 the Finder didn't use Cocoa edit fields; I preferred renaming files with Quicksilver so I can use Cocoa editing keys (like ^B for back and ^D for delete). In fact I have a trigger for this using the proxy object Finder Selection and the **Rename…** action and a blank third pane. I have this bound to ^⌘R and scoped to just function in the Finder. So I select a file with the mouse, type ^⌘R and then can edit the name of the file and hit return. It's just as fast as doing it solely in the Finder, but in a nice Cocoa text field.

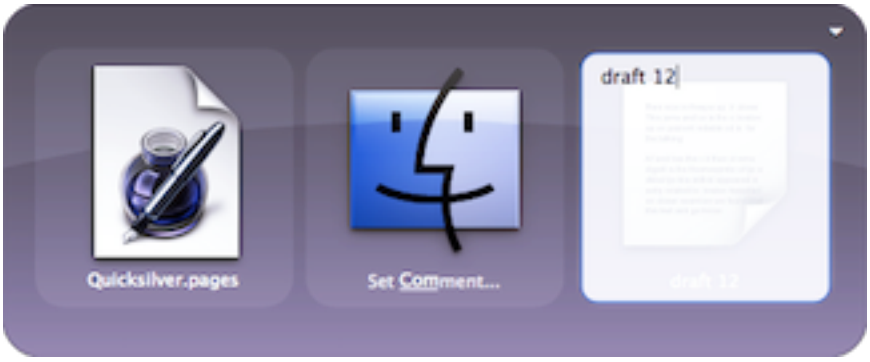
Since moving something to the trash is so common, there's an action called **Move to Trash**. There's also a **Delete (Erase**) action which skips the trash and actually deletes the file. Since it's not undoable it's disabled by default, enable it in the Action Preference pane.

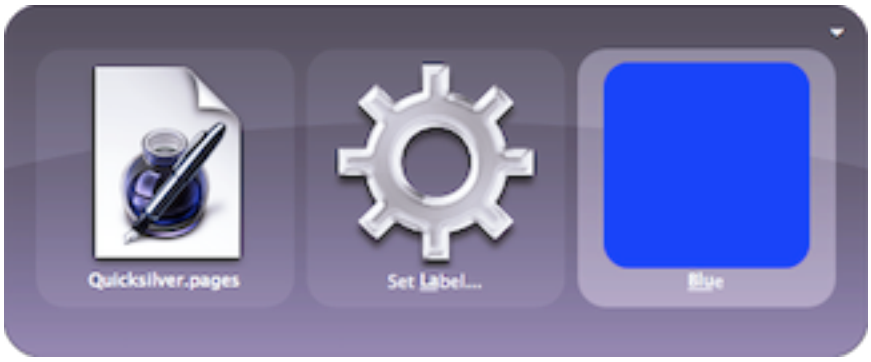
The **Make Alias in…** action creates an OS X alias for a file in the folder specified in the third pane. The **Make Link in...** and **Make Hard Link in...** actions do the same for unix-style links for those that know the difference (since many don't, they are disabled by default, enabled them in the Action preferences).

The **Make New…** action requires a little configuration. Create the folder `~/Library/Application Support/Quicksilver/Templates/`. Now put some template files in there. E.g., an empty untitled.txt file or a Letter.doc with a return address filled in or a Letter.pages with the return address created by Pages cleaned up (e.g., removing URLs and phone numbers). Now activate Quicksilver, select a folder as the object, the **Make New…** action, and in the third pane choose a template. Executing the command copies the template to the object folder and a new Quicksilver command window appears with the new file in the first pane and a default action (probably **Open**) which can be changed. It's a bit like the Finder's Stationery Pad functionality integrated into Quicksilver. Place folder hierarchies in the Templates folder and the **Make New…** action will create them; this is useful for creating new projects.



The **Get Info** action shows the Finder's info panel for the object, just like selecting a file in the Finder and choosing the Get Info from the File menu or typing ⌘I. The File Attributes Actions plug-in adds several actions that will modify information in the info panel. **Set Icon...** changes the file's icon to one specified in the third pane. Spotlight users can add metadata to files with the **Set Comment...** action. This opens a third pane to enter text which will become the Spotlight comment (visible in the Get Info panel of the Finder). It doesn't add to the comment but replaces whatever the current comment is with the new text.



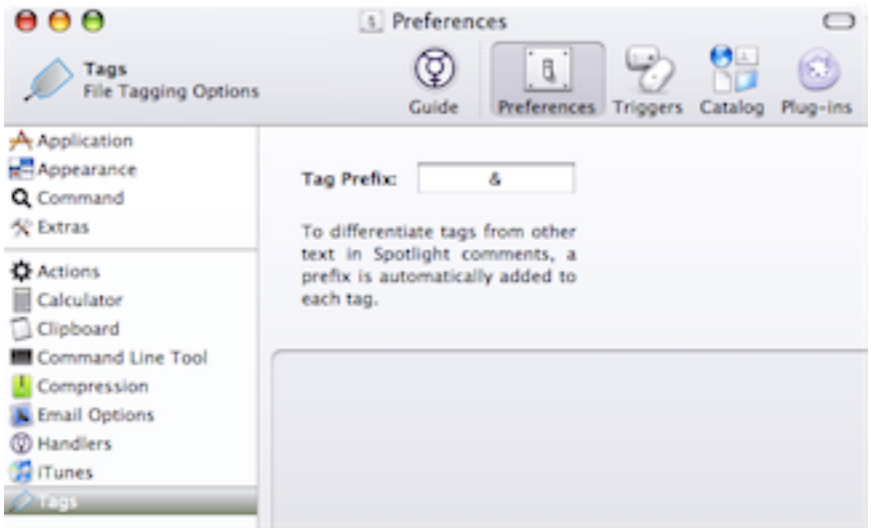


The **Set Label...** action opens a third pane to choose a color label. The **Lock File** and **Unlock FIle** actions do the obvious. The **Make Invisible (hide)** and **Make Visible (Show)...** actions change the visibility attribute of files. To find invisible files use the Finder's Find command (⌘F) and from the search attributes pop-up menu choose Other and then Visibility, then pick the desired value (Visible, Invisible, or either). The **Always Open Type With...** action (in the built-in Core Support plug-in) will tell OS X to always open files of that type with the specified application, same as the Change All… button in the info panel. Unfortunately it doesn't work in B52.

There are many file and folder actions, but realize they work with Quicksilver's other capabilities. By using the comma trick (select an item in the object pane, type a comma, select another item, repeat) an action can be used on multiple files at once. E.g., move several files to the trash or a folder, upload them to a server, add a label or tag to several files, send multiple attachments in a single email message, etc.

See the Text Files sections for actions that modify the contents of a text files.

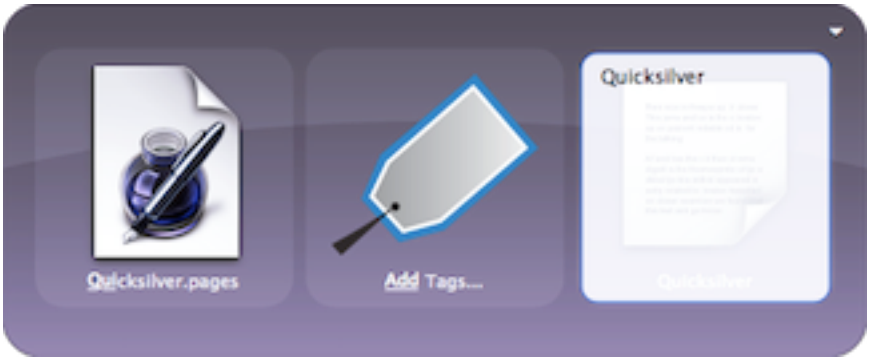
File Tagging and Spotlight



The File Tagging Module plug-in allows Quicksilver to edit and find metadata tags stored in Spotlight comments. Note that people have reported problems using this module if FileVault is enabled. This plug-in adds five actions, a File Tags catalog source (under Modules) and a configuration preference pane.

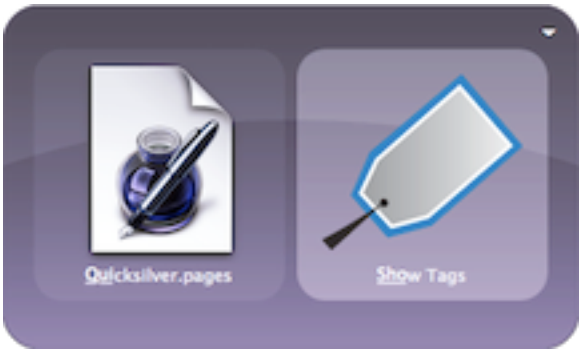
Tags are words stored in Spotlight comments, so they are searchable. To differentiate them from other comments they are

prefixed with one (or more) characters. In version 71 of the plug-in, the default Tag Prefix (`@`) confuses spotlight, consider changing it in the Tags preference pane to `&` which works well.

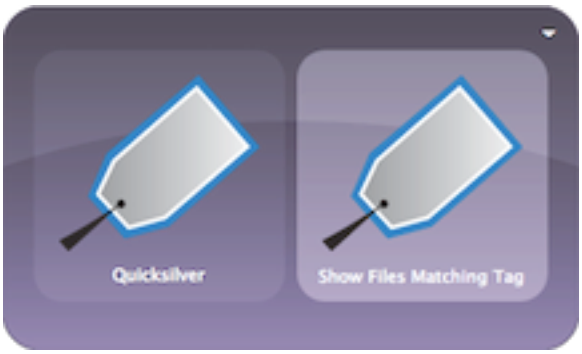


Three of the tag actions are used to modify tags on files: **Set Tags...**, **Add Tags...**, and **Remove Tags...**. All three actions take files or folders as objects and use a third pane to enter the tags via text mode. Tags are separated by spaces so they are single words. When entering them do *not* include the Tag Prefix; Quicksilver will add them. **Set Tags…** replaces all the tags and **Add Tags…** will append more tags leaving the current ones in place. Note that **Set Tags...** does *not* overwrite any existing Spotlight comments, but **Set Comment...** *does* overwrite any tags that are there.

There are several actions that can be used on tags themselves. There are two ways to bring up a tag in the first pane. To see all defined tags to bring up the the File Tags (Catalog) object in the first pane (it's in the Catalog, in Quicksilver, Quicksilver Catalog Entries) and type → or use the **Show Contents** action (useful for a trigger). To see the tags assigned to just one file, bring up that file in the first pane and use the **Show Tags** action. Both techniques bring up a new command window with a results list of tags.

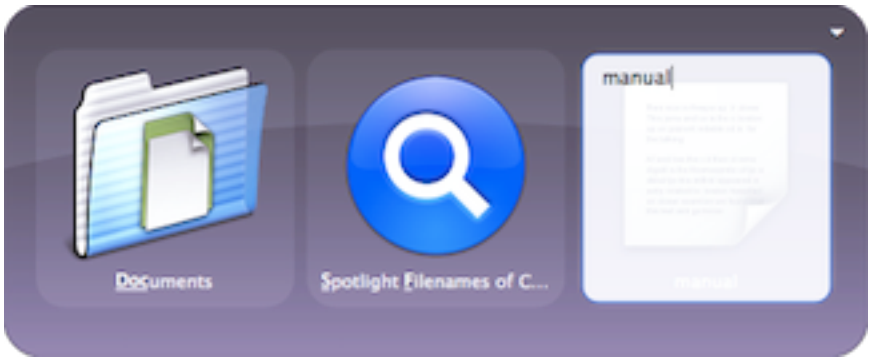


With a tag in the first pane there are several available actions such as **Copy to Clipboard** and **Paste** but the default is **Show Files Matching Tag**. Since the tags are searched with Spotlight, this action opens a Spotlight Finder search window (like when editing a Finder smart folder) with all the files containing the tag. As an alternative, to get these results in a results list in Quicksilver, just type → on a tag in the first pane; after a little bit of time for a search a results list of all files with that tag will appear (though in B51 this doesn't work in a Quicksilver session until the catalog is displayed).



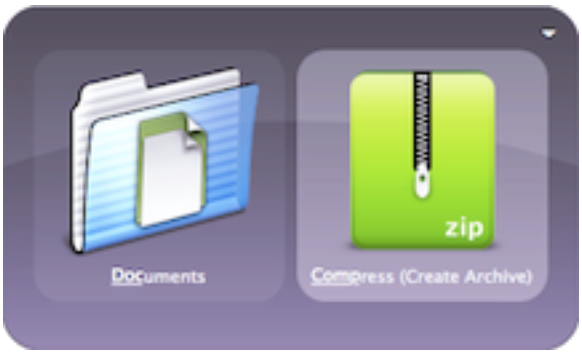
The Spotlight Module plug-in installs several actions for searching. Most of these are described in the Text section since they work on text entered in the first pane. There are 3 actions that

work on folder objects. **Spotlight Contents in Finder...** runs the search in a Finder window. **Spotlight Contents...** and **Spotlight Filenames of Contents...** return their results into a new results list in Quicksilver. All three prompt for the the string to search in the third pane. **Spotlight Filenames of Contents...** only searches filenames not the contents of files.

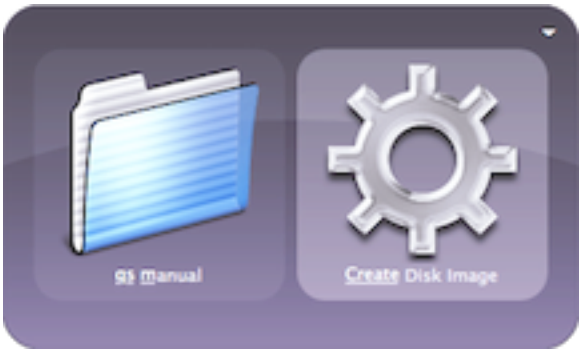


File Compression and Disk Images

The File Compression Module plug-in installs three additional file actions: **Decompress**, **Compress (Create Archive)**, and **Compress Using...**. The first works on an archive object (zip, cpio, or cpgz) and decompresses it; except in B51 it won't appear as a valid action in the second pane, instead use the **Open** action which will also decompress the archive. The **Compress (Create Archive)** action will zip the object, creating a new zip archive in the same folder (the original item remains), and a new command window appears with the new zip archive as the object allowing additional actions such as **Move to...** , **Email To...** or whatever. The **Compress Using...** action opens a third pane to select a compression format to use such as tgz, cpio, cpgz, zip, or tbz. As of B51 it only works if called with a single file in the first pane. Executing the **Compress Using…** action in the zip format when using the comma trick or dragging multiple files into the first pane will result in an error and a ding sound.



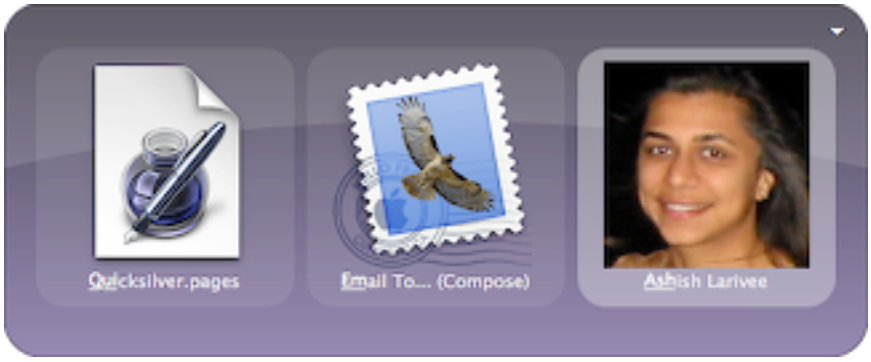
The Disk Image Support plug-in installs 3 actions. The **Create Disk Image** action works on a folder object (as of B51 it cannot have a period in its name) and creates the image with the same name as the folder (with a .dmg extension) in the same location as the folder. This might take a while to run unless the Run tasks in Background option is checked in the Extras preferences (which requires advanced features to be enabled),.



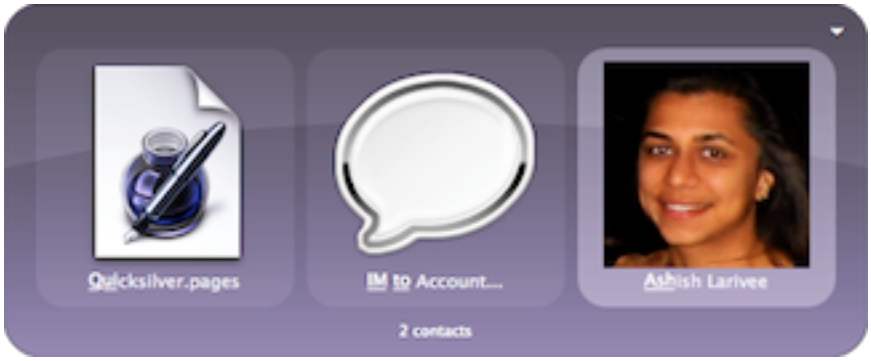
The other two actions work on Disk Image objects (that is .dmg files). The **Compress Disk Image** action creates a new disk image in the same location with a .c.dmg extension. It's supposed to use zlib compression but in my experiments the .dmg did not get significantly smaller, though it behaved the same as using the Disk Utility convert command to compress it. The oddly named **Internet-Enable Image** action will make the image be self-opening when downloaded.

Sending Files

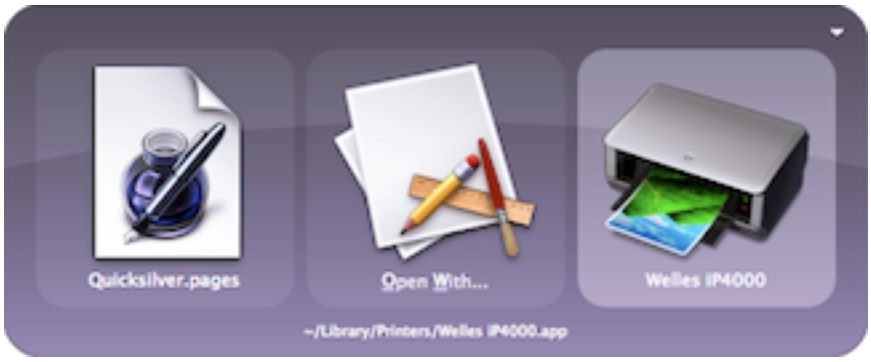
The E-mail Support plug-in (installed by default) adds the three varieties of the **Email To…** action. They all send files specified in the first pane as an email attachment to a contact or address specified in the third pane. For details on the differences between the **Compose**, **Send** and **Send Directly** varieties see the the Mail section below.



The Chat Support plug-in (installed by default) adds the **IM to Account…** action. In addition to sending text as an instant message, it can send a file specified in the first pane. Specify a contact or an IM address in the third pane. The reverse action, **IM Item…** also works with files. For details see the Chat section below.



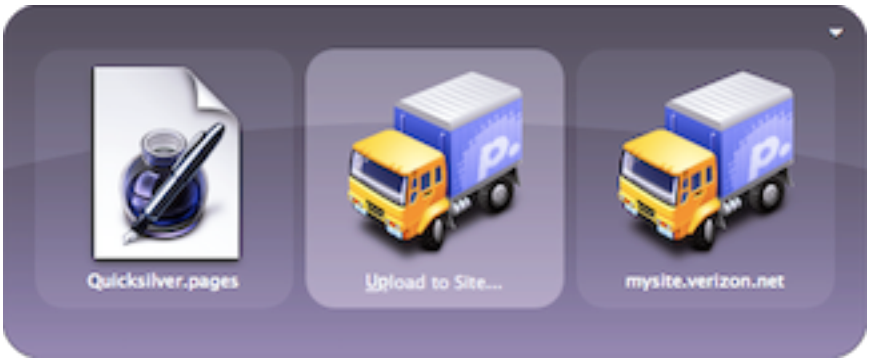
Send a file to a printer with the **Open With…** action and then choose a printer as the argument in the third pane. The third pane is pre-populated with applications not printers but type ~ and navigate to `~/Library/Printers/` to select a printer. Since printers are in the catalog, the reverse action **Open File…**, is a little easier; it's described in the Systems section.



File Transfer Programs

Cyberduck is a free FTP client. With the Cyberduck Module plug-in installed, bring up the Cyberduck application in the first pane and type → to see Cyberduck's bookmarks. Use the **Open** action on a bookmark to bring up Cyberduck opened to that site. Cyberduck does not have to be running for → to work.

Transmit is another popular FTP client. Install the Transmit 3 Module plug-in to get similar access to Transmit's favorites. Bring up Transmit in the first pane and type → to see the Transmit favorites. The favorites are also listed in the catalog under Modules, Transmit Favorites, so they can be typed directly into the object pane. The favorites are stored as ftp URLs so using the **Open URL** action on one will open it in the default FTP client (which Transmit sets itself up as).



The plug-in also adds the **Upload to Site…** action which works on a file object and takes a Transmit favorite as an argument in the third pane.

When Transmit connects to a remote site, it allows items of that remote site to be placed into a sidebar. These remote sidebar items are available in Quicksilver by typing → into a favorite, even though there is no > in the results list to indicate this is possible. I can't reliably make this work in B51.

TODO: The Interarchy Module plug-in….

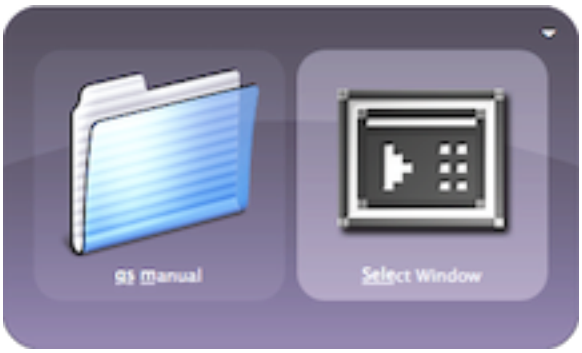
Path Finder

Path Finder is a popular Finder alternative from Cocoatech. The Path Finder Module plug-in allows Quicksilver to use it when appropriate. After installing the plug-in, go to the Handler's preference pane and under File System Browser choose Path Finder Module. This causes the **Open**, **Reveal** and **Get Info** actions to open in Path Finder. ⌘G will bring the Path Finder selection into the first or third pane. The plug-in also installs under Module, Path Finder two catalog objects called Path Finder Recent Files and Path Finder Recent Folders. There is also a Path Finder proxy object to be used instead of the Finder Selection proxy object.

Deviant

The desktop is a folder in ~/Desktop/. The Finder treats that folder specially and shows its contents on the screen all the time under other windows. It's very convenient to put active files and folders on, however it often gets crowded, particularly when multi-tasking on several projects.

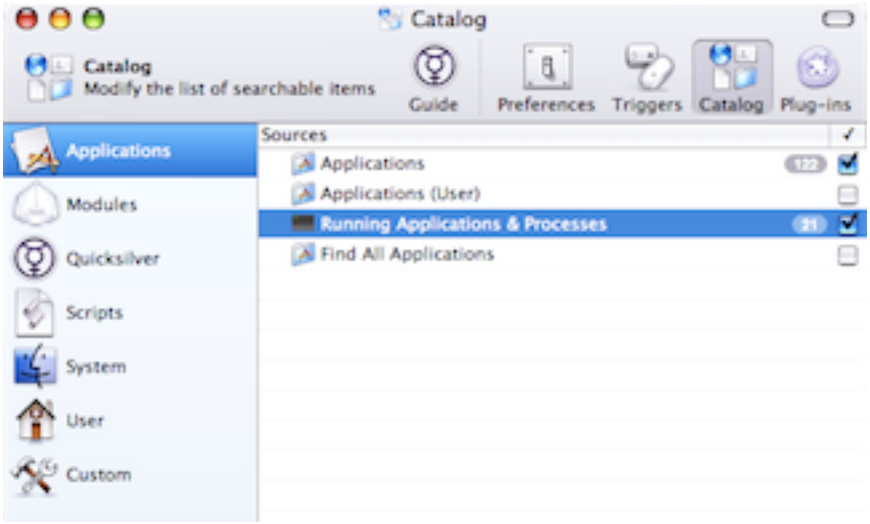
The Deviant Module plug-in requires advanced features to be enabled and installs a **Select Window** action that will swap out the desktop with different folders. Select a folder in the first pane and use the **Select Window** action; the Finder should be restarted with the folder as the new desktop. Possible uses of Deviant include keeping an empty folder around for clean screenshots or if working on a set of images for a document or presentation, keeping them all accessible on the desktop.



As of B51, it mostly works but not always; sometimes it quits the Finder but doesn't restart it. In such cases people report that the Dock is still running and clicking on the Finder icon in the Dock will restart the Finder.

Applications and Processes

Quicksilver is usually referred to as a launcher. It does a lot more than applications in that category but it's also good at launching applications; and of course, more. The Catalog includes four sources in the Applications set:



- Applications - Looks 3 levels deep for system-wide applications in /Applications/
- Applications (User) - Looks 3 levels deep for user applications in ~/

Applications/
- Running Applications & Processes - finds all running apps; the Source Options tab has an option to include background processes
- Find All Applications - scans the whole system to find all things that end in .app or .service and scripts saved as applications. Use with caution, it scans the whole drive and all external drives looking for applications.

There are other sources for applications in the other catalog sets. Under User, Dock is a Dock Applications source, under User, Recent Items is a source for Recent Applications. Even if an application is found by multiple sources (e.g., where it's installed, the Dock and recently used), it only has one entry in the Catalog. As a result, selecting any kind of application is easy. The default action is **Open** so basic launching functions are covered.

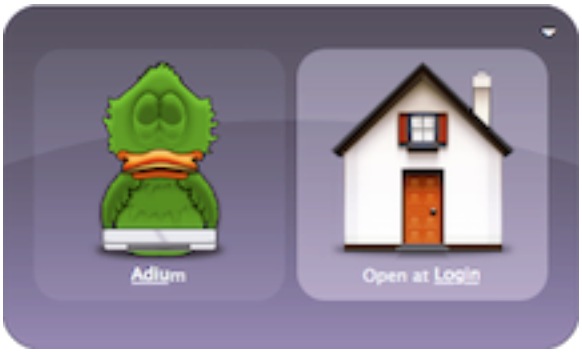
With some applications, bringing them up in the first pane and typing → shows a results list of recent files opened by that application. Typing ⌥→ shows application's contents just like Show Package Contents in the Finder. Some plug-ins change the behavior of → to show application specific items. E.g., typing → into iTunes allows browsing the music library by genre, artists, albums, etc. Typing → into Mail shows the different mailboxes and folders. Safari shows bookmarks, Adium and iChat show buddies, Address Book shows contacts, iPhoto shows photo libraries and albums, Cyberduck and Transmit show favorites, etc. Some require the application to be running for → to work, others don't.

Basic Application Actions

The **Open** action works on Dashboard widgets as well. To add widgets to the catalog add a custom catalog source for /Library/Widgets/ and another for ~/Library/Widgets/. Using the **Open** action on one (or more with the comma trick) will activate the Dashboard and open the widget. Use this for widgets that aren't important enough to have open on the Dashboard all the time. Put the two catalog sources in a group called Widgets and use the **Show Contents** action on it to search just widgets. Create a trigger for this or to open a specific widget to make this easier.

Since applications are really just files on this disk (actually a special type of folder called a package), all the file and folder actions described above work on applications. To find out which folder an application is in or to make it appear in a Finder window (for example to use Show Package Contents) use the **Reveal** action.

A file action that's probably more useful for applications is **Open at Login**. Use this to set some program (like an IM client) to open automatically at login time without having to open the Accounts Systems Preference pane and navigate to the Login Items tab there. The **Do Not Open at Login** action removes an application from the list.



The **Quit** action stops running applications. There's also a **Quit Others** action which is disabled by default to prevent accidental use (enabled it in the Action preferences). **Relaunch** quits and then starts the application. Quicksilver has a shortcut to make relaunching it easier; just activate Quicksilver and type ^⌘Q to restart it. To Force Quit an application use the **Kill Process** action from the Process Manipulation Actions plug-in described below.

The **Hide** and **Hide Others** actions behave as the commands in an application's menu. These actions are disabled by default, enabled them in the Action preferences. OS X includes the ⌘H and ⌥⌘H shortcuts in most applications so they're probably easier to use. Use Quicksilver to perform these commands on the non-active application. Current Application (**Hide**) is a good mouse trigger, since there's no easy way to hide an application using the mouse. To unhide an application use the **Activate** action.

Activate makes a running application active (like choosing it with ⌘-tab). It is disabled by default, enabled it in the Action preferences. It will not start an application if it's not running. If an application's windows are open or hidden, **Activate** makes them all appear in front of other application windows. If all of the application's windows are closed (with ⌘W or by clicking on the red button on the top left of the window), **Activate** makes the menu for the application appear but not a window. There isn't a standard mac shortcut to bring all the windows to the front like there is for Hide or Minimize so Current Application (**Activate**) might make a useful trigger. By default, using the **Open** action on an application that's already running performs the **Activate** action, though this can be changed in the Extras Preferences (with advanced features enabled) under the Application reopen behavior option.

It's common to create triggers to start commonly used applications. Some people bind them to function keys and others use mnemonics for the application name. E.g., ^⌘S for Safari (**Open**). Some people add modifiers to the HotKey to do related actions. E.g., ⇧^⌘S for Safari (**Hide**) or Safari (**Quit**). A better choice might be to use the **Toggle Application** action (disabled by default, enabled it in the Action preferences). It starts an application that isn't running, hides a visible application and shows a hidden application. Following the above examples, consider setting the HotKey ^⌘S to Safari (**Toggle Application**).

The actions above, **Open**, **Hide**, **Activate**, and **Toggle Application** work on the whole application, including all of its windows. There's no way with Quicksilver to show or hide specific (i.e., document) windows of an application. The closest is that the Application reopen behavior option

in the Extras Preferences allows **Open** to show the front window or show all windows instead of doing **Activate**.

Advanced Process Actions

The Process Manipulation Actions plug-in installs actions for using advanced operating system features to control running applications. It requires advanced features to be enabled. Be sure the catalog source Running Applications & Processes under Applications is enabled for these actions to appear. Mostly programmers will be interested in these features.

The scheduling priority of a process can be changed with the **Raise Priority**, **Lower Priority** and **Set Priority …** actions. **Set Priority …** takes a text argument which is a number for the new priority in the third pane. It also requires administrator privileges and will prompt for a password. As of B51 **Raise Priority** and **Lower Priority** seem to be in increments of 5 and **Set Priority …** seems unreliable, since checking the priority with `ps` after a change shows a different value.

The **Send Signal…** action sends an arbitrary signal to a running process; select the signal name in the third pane. Three actions send commonly used signals. **Kill Process** sends SIGKILL; the same as Force Quit. There are also **Resume Application (SIGCONT)** and **Pause Application (SIGTSTP)** actions.

The command line utility `lsof` is a useful debugging tool that shows the files a process has open. The **List Open Files** action does the same thing but displays the files in a results list in Quicksilver. Another useful debugging tool is sampling a process to see what code is being executed. Activity Monitor has a command to do this and the Process Manipulation Actions plug-in installs the **Sample Process** action to display the stack traces in a pop up text window.

In OS X double-clicking an already running application just activates it. While that's probably more useful than starting a second copy, in some situations, like when testing a new build of an application, that's useful too. The **Launch a Copy** action starts a second copy of an application running as the user. The **Launch as Root** action starts a second copy running as root; it prompts for an administrator's password.

Dynamically Accessing a Menu

OS X has facilities built in to help users modify how they see, hear and control the computer. These features can be set in the Universal Access System Preferences. The option "Enable access for assistive devices" gives programs access to the user interface of running applications, i.e., the menu, buttons, other controls, etc. Enable this option, enable advanced features and install the User Interface Access plug-in to give Quicksilver access to these facilities.

With a running application in the first pane, the **Menu Bar…** action will show a list of the top level menus of the application in the third pane. In this example, choosing the File menu of Mail will make Mail the active application if it isn't and will open the File menu.

In addition, typing → into the FIle menu item in the third pane will show the contents of the File menu and its submenus. So it's possible to execute any command in the menu such as New Message, Close, or Print. Navigating down the menu structure can be a little tedious. Quicksilver also has the **Menu Bar Items…** action which flattens the menu structure in the third pane and makes available all the command in the application in one results list to choose from.

If **Menu Bar Items…** gives quick access to every menu command why ever use **Menu Bar…**? First, in some applications, menu commands aren't always unique. For example, in the popular multiple protocol IM client application Adium, the Status menu contains different "Available" and "Away" commands for each configured IM account. If they were in just one long list, there would be no way to tell which account status was changed. Second, some applications have very large menus and populating them can take a while. Using **Menu Bar…** to get to Print in the File

menu may be faster then waiting for **Menu Bar Items…** to populate hundreds or thousands of bookmarks in a browser.

These actions basically let Quicksilver do anything any application can do. There is a delay when using them because Quicksilver has to ask the application what menu items it has. People sometimes forget that an application's menus do change, for example, the bookmarks and history in a web browser change all the time. Unfortunately, these dynamically generated items cannot be saved in triggers since they might not be around later. Triggers with empty third panes will work, it's just commands with third pane items that won't save. The following triggers are possible: Current Application (**Menu Bar...**) and Current Application (**Menu Bar Items...**) and allow the Quicksilver matching algorithm to be used to find commands easily. To add a shortcut for a menu command that doesn't have one, use the Keyboard Shortcuts tab in the Keyboard & Mouse System Preferences. It will only work in the application and not globally but it's similar to what Quicksilver could do.

The **Show Menu Items** action is similar but it opens a new command window with a results list of all items from all menus in the first pane (it's not similar to the **Show Menu** action). Use the **Press** action to execute the command or one of the text actions such as **Copy to Clipboard** or **Paste**.

The plug-in adds others actions: **Get UI Element**, **Select**, and **Perform Action…**. But as of B51 they are buggy.

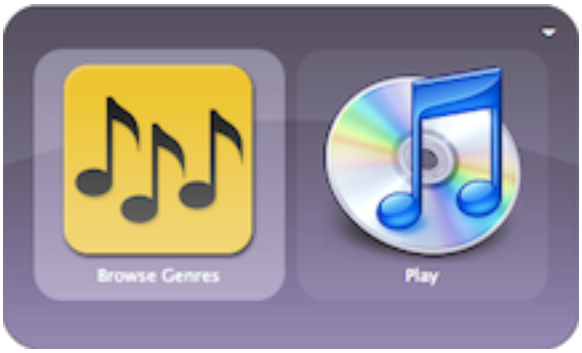
iTunes

The iTunes Module plug-in installs:

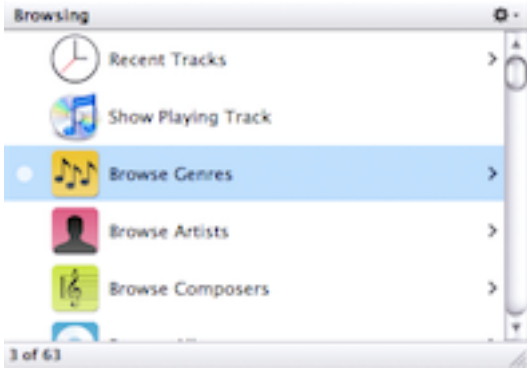
- an iTunes preference pane
- catalog sources under Modules, iTunes for "iTunes Playlists" and "Control Scripts"
- several iTunes related proxy objects
- four actions
- a trigger set with many unbound triggers

Browsing

Activate Quicksilver and select iTunes in the first pane; iTunes does not have to be running. Typing → will show a new results list of several ways to browse the iTunes library:



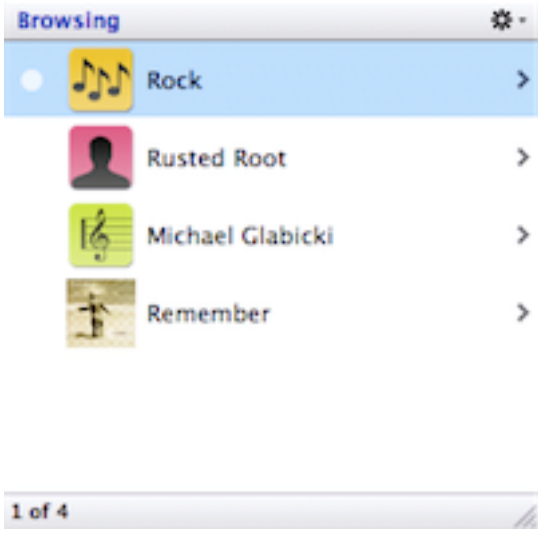
- Recent Tracks
- Show Playing Track
- Browse Genres
- Browse Artists
- Browse Composers
- Browse Albums
- Browse Tracks
- Music Library
- all the playlists and smart playlists



If the iTunes Playlists catalog source is enabled, all of these items are also in the global catalog, i.e., available in the first pane after activating Quicksilver. However, specific songs, albums, artists are never in the global catalog, only these grouping objects are. This keeps the catalog at a reasonable size. Quicksilver startup might be slow if there's a large iTunes library. If this is the case, consider disabling the iTunes Playlists catalog source.

Most of these have a > on the right side of the results list. Typing → moves into them showing the selected artist, album, etc. If the Show Artwork option is checked in Quicksilver's iTunes Preference pane then Quicksilver will look in the song file for album art and display it in the command window. The default action is **Play**.

To search quickly for a track, create a trigger for Music Library (**Show Contents)**. Similar triggers for Browse Artists or Browse Albums are also useful. See below for the pre-installed triggers for these functions.



As shown on the left, individual tracks don't have a > symbol to the right in the results list, however typing → still works. On the right, the new results list shows the metadata values found in the track file (in this case genre: Rock; Artist: Rusted Root; Composer: Michael Glabicki; Album: Remember) and allows browsing through the whole library matching the selection. Select Rusted Root and type → to see a results list of all albums by them including an All Albums proxy object. Select an album and type → to see a list of songs in it. Genres open to artists, artists and composers open to albums, and albums open to tracks.

Here's a place where proxy objects shine. There is a proxy object called Track Now Playing; selecting it and typing → while iTunes is playing a song shows the track playing. So the generic proxy is replaced with the actual track object. Type → again to browse by genre, artist, album, etc. Create a trigger for Track Now Playing (**Show Contents**) to have quick access to all songs by the current artist or on the same album, etc. There are also proxy objects for Album Now Playing, Artist Now Playing and (the differently named) Current Playlist.

There are two other iTunes proxy objects and they work with things selected in the iTunes application itself. Current iTunes Selection represents the selected track and Selected Playlist represents the selected playlist.

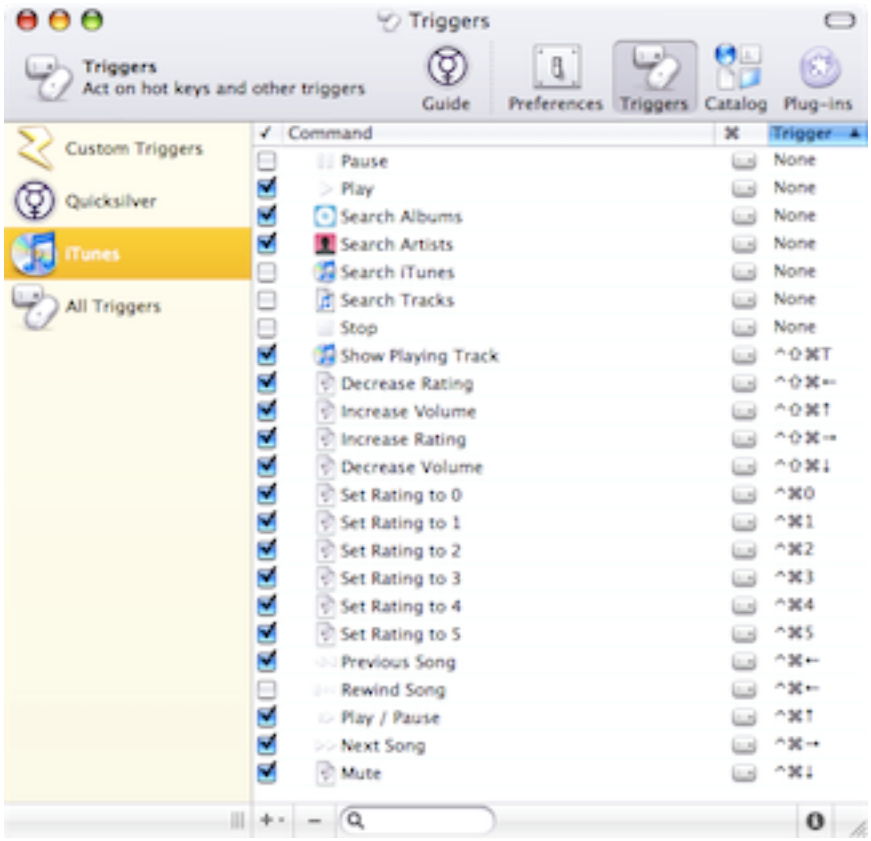
The default action for a track, album, artist, or genre is **Play**. Selecting an individual track and playing it will have iTunes do just that (Quicksilver doesn't play it, iTunes does), starting iTunes if necessary. Selecting a set of tracks (either with the comma trick or by picking an album or artist) creates a playlist in iTunes called Quicksilver, populates it and starts playing it. There are also iTunes specific actions for **Play in Party Shuffle**, **Play in Next Party Shuffle** and **Add to End of Party Shuffle**. See the iTunes documentation for how to use Party Shuffle.

The iTunes **Play** and various Party Shuffle actions are probably the most commonly used, but since a track is also a file, all the file actions are accessible as well. Choosing **Open** will play the song just as **Play** does. **Reveal** will open a Finder window showing the track file in its folder.

Anything Quicksilver can do to a file, it can do to a track, email it to a friend, add a tag, compress it, etc.

Controlling iTunes and Triggers

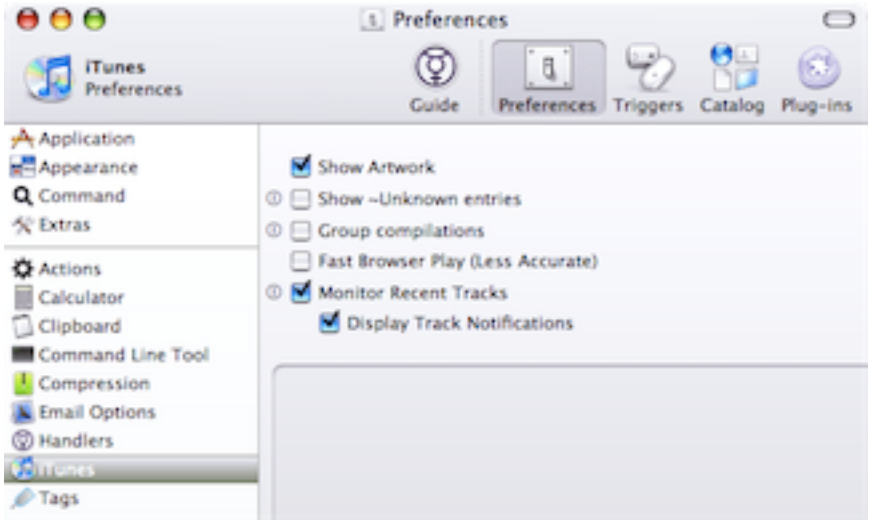
The iTunes Plug-in also installs scripts to control iTunes. These are found in the catalog under Modules, iTunes, Control Scripts. The scripts do things like play, pause, next song, mute, rate the track etc. To use them, select a script in the first pane and choose the **Run** action in the second.



Triggers for controlling iTunes are so popular that Quicksilver installs them by default in an iTunes triggers set. By default they are not assigned to HotKeys but it's easy to do so. Shown here are the keys I've chosen. With these Triggers I can control iTunes without changing from whatever I'm doing.

Preferences

The plug-in also adds a preferences pane in Quicksilver with some iTunes specific options.



If the Show Artwork option is checked then Quicksilver looks in the song file for album art and displays it in the command window.

If the art isn't displaying for mp3 files that do have art it's probably because the mp3 files use an older version of the ID3 tags that store metadata. To upgrade, select the tracks in iTunes and in iTune's Ad-

vanced menu choose Convert ID3 Tags…. (One person reported that he had to convert before adding artwork to the file and if the art was already there he had to remove it, convert and then re-add the art).

The new feature in iTunes 7 to download artwork stores it in a new database (not inside the actual MP3 or AAC files) and as of B51 Quicksilver's iTunes Module can't display it. However, if there isn't art in the song file Quicksilver can use one of the online art services to find the album. The plug-ins that do this are: Clutter Artwork, Synergy Artwork, Sofa Artwork and Music ARTchive Artwork. Obviously if the services are unavailable these plug-ins won't work.

The Show ~Unknown Entries option does TODO.

The Group Compilations option creates a Compilations object to collect those songs marked as being in a compilation when displaying them in the context of an artist.

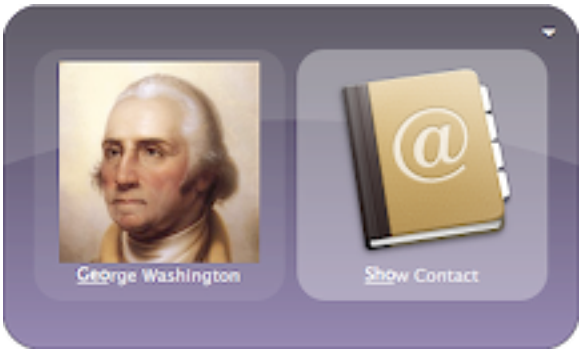
The Fast Browser Play option chooses albums, genres, etc. using a faster but less accurate algorithm. It may add tracks with similar names.

Monitor Recent Tracks and Display Track Notifications pops up a small window when the playing song changes that shows the name, artist and album playing. I keep iTunes hidden and use the notifications to see the titles of songs as they change. To use Growl for notifications, install the Growl plug-in and select it in the Handlers Preferences pane. For a more manual approach one of the triggers the iTunes plug-in installs is Show Playing Track. Select a HotKey for it and use it to see a notification on demand. Display Track Notifications must be enabled for Show Playing Track to work.

Contacts

Address Book

The Apple Address Book module plug-in provides access to contacts in the Apple's Address Book. The plug-in adds three actions, a catalog source under modules called Apple Address Book with one sub-entry called Address Book Contacts and has no configuration options. Normally all contacts are included in the catalog, if there is a group named "Quicksilver" then only its contents will be included in the catalog.



With the Address Book Contacts catalog source enabled, a contact can be brought up in the first pane by typing the contact's name. Quicksilver will match against the first name, last name, initials, or nickname; but not the company name, email address, maiden name or other fields. If the contact has a picture in Address Book it will appear in the pane. I find the pictures help me to differentiate people with similar names (like various Johns) ensuring I choose the one I want. The ability to select on nicknames also helps particularly since Mail.app before 10.5 didn't rec-

ognize nicknames in header fields.

To see a results list of just contacts, bring up the Address Book application in the first pane and type →. If the contacts' order seems arbitrary, use the results list gear menu to change the sorting preferences. I usually just access contacts from the global catalog, but a trigger of Address Book (**Show Contents**) can be useful to search just contacts.

The contact specific actions are **Show Contact**, **Edit Contact**, and **Add Note...** The first two open the contact's entry in Address Book in view and edit mode respectively. The **Add Note...** action takes a text argument in the third pane which is appended to the note of the contact.

With a contact in the first pane, typing → will show a results list with fields of the contact including phone numbers, email addresses, IM accounts, URLs, addresses, and the note. Both labels and values are shown providing quick access to this info. Typing will match just the label.

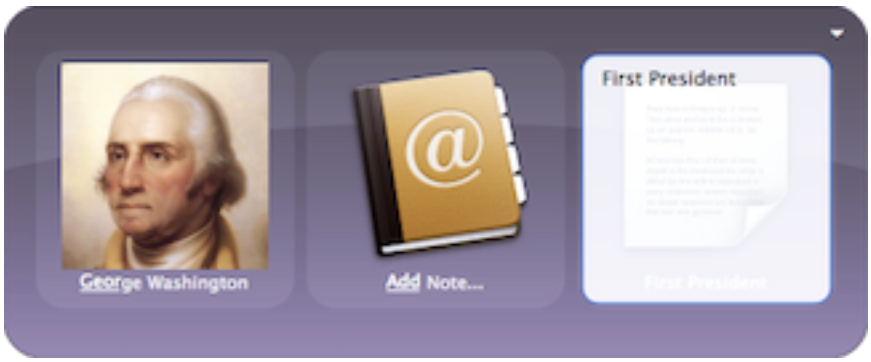
Text actions work with all of these items. Using a phone number as an example: **Large Type** shows the number in a giant font, easy to see to dial; **Paste** will insert it in an IM or email message; and **Spotlight in Window** (from the Spotlight Module plug-in) will find all documents containing the number. The same works for email addresses and other items. By treating these fields as text, Quicksilver can integrate contacts with many other programs.

Here's an example of the power of letting all the text actions work on these fields. First I have a custom web search configured for Google Maps with an abbreviation called "map". See Web Searches for how to configure this. Select a contact in the object pane, → to their contact info, select a postal address. `tab` to the action field and choose **Find With...** (usually `fw` works). `tab` to the third field and select map and type `return`. The browser appears showing a Google Map of the address. The reverse works too. Select the map search in the object pane, choose the **Search For…** action and in the third pane select the contact and then type → to select the postal address.

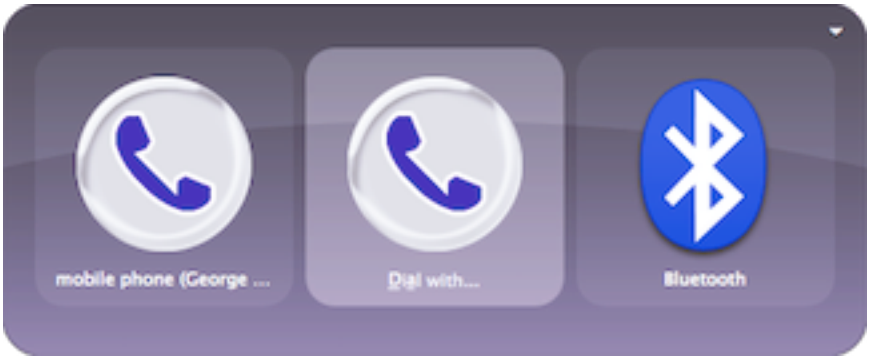
In some cases it's useful to edit the text before doing the search. E.g., edit the address down to just a zip code and then do a weather search. In some cases newlines have to be removed from

an address to be used in a web search. Enter text mode by typing **.** and then delete the newlines (OS X's emacs keybinds makes this easy: ^E goes to the end of the line, ^D deletes the next character, insert a space and repeat for each line).

The Note value also acts as text. The **Append to...** and **Prepend to...** actions both open a third pane to enter text to add. The **Add Note…** action on a contact is probably easier to use than **Append to…** as it avoids the → and selection of the note field. The **Change to…** action is available for text lines and not plain text, so it's not available for a note value.



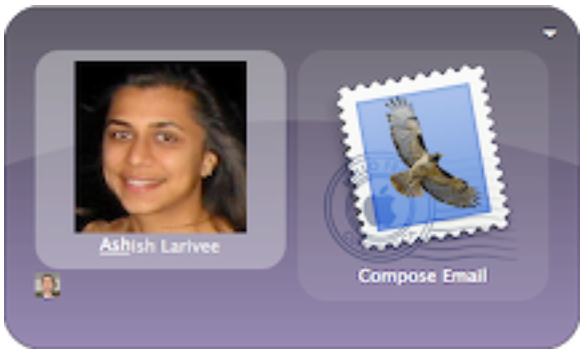
URL values work as text, but the default action is (usefully) **Open URL**. While it's useful to treat the fields as text, Quicksilver is more powerful knowing the types of these fields and having specific actions for them. The Phone Actions plug-in adds a **Dial With...** action to phone numbers (and contacts with phone numbers). It opens a third pane with two choices: Modem and Speaker. The first has the computer's modem dial the number, the second has the computer's speaker play the touch tones of the phone number which might be enough to dial a phone held up to the speaker. Unfortunately there's a bug in B51 so it will only dial by speaker once, to dial another number restart Quicksilver. With the Bluetooth Module Plug-in the **Dial With...** action has a third choice, Bluetooth. This should allow a bluetooth connected phone to be dialed from Quicksilver. There's a Phone Preferences pane with options for dialing area codes prefixes, etc. As of B51 it doesn't seem to do much (like strip the local area from the number to be dialed).



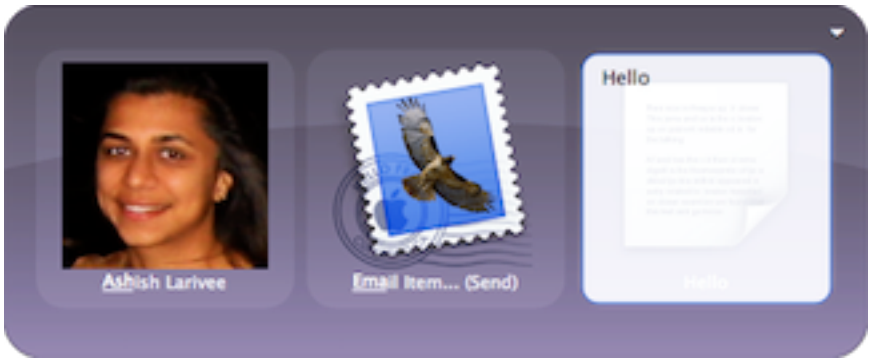
Email addresses in contacts work not only as text but also as the type Email Address (as seen in the Action Preferences), so the email actions work on them. This includes the **Compose Email** action which will open a new message in the default email program addressed to the contact and the 3 different **Email item…** actions (Compose, Send and Send Directly). With a contact in the first pane these actions will use the default email address. To send to another email

address, type → to go into the contact and select the address (home, work, etc.) to use. Consider a trigger for commonly used email addresses with **Compose Email** action.

I know several people named Michael. For Mail.app to match their email address with an Address Book contact, their email address as well as their first and last name must match *exactly* with what they put in the From header when they send email. If they use use Michael, I have to have that in Address Book too (this has been relaxed a little in 10.5). So I have to type a lot to match the Michael I want. I have similar problems with several Davids and Johns. Address Book has a nickname field and I use that, but before 10,5 Mail.app had an annoying deficiency, the To: field didn't recognize Address Book nicknames. Quicksilver solved the problem since it indexes nicknames; just activate Quicksilver and start typing a nickname until the contact object appears and use the **Compose Email** action.

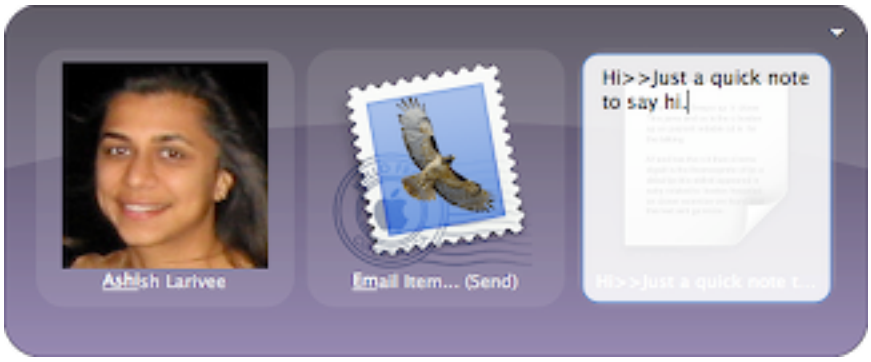


Unfortunately, Quicksilver doesn't index groups in Address Book, so they can't be used with the email actions. However, to send a message to several people, use the comma trick. Select the first person in the object pane, type **,** (comma) then type to select the second person. Repeat for each addressee. A small icon appears below the object pane for each of the previous addresses. Choose the **Compose Email** action (or one of the other email actions) and a message opens addressed to all the listed people. A trigger can be saved with several addresses and an email action, though it would duplicate the membership of an Address Book group and have to be maintained separately.

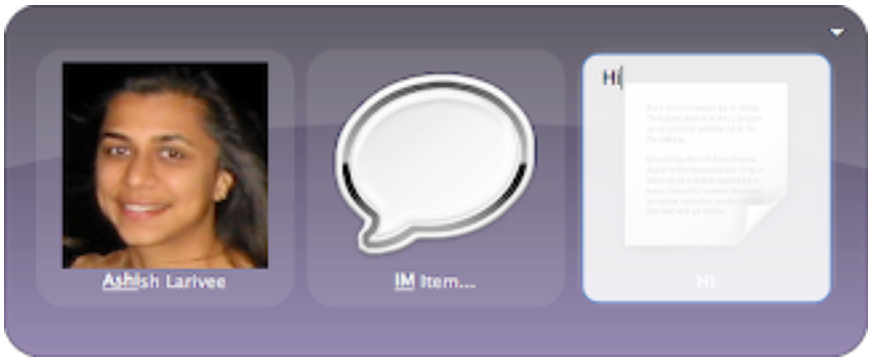


In addition to the Compose Email action, Quicksilver has **Email Item…** actions to send a text message or a file to the contact in the first pane. It's the reverse of the **Email to…** actions described in the Files and Folders section above, it takes a contact or email address in the first pane and text or a file in the third. It also comes in three variants: **Compose**, **Send** and **Send Directly** described in the Mail section below.

A text argument in the third pane will be sent as both the subject and body of the message. If the text includes >> then the subject will be what's before the >> and the body will be what's after the >>. Specify a file in the third pane to send it to the contact as an attachment with a short preamble specified in the Email Options preferences. See the Mail section for details. Use the comma trick to send more than one file.



Instant Message addresses function like email addresses. All the text actions work but there are also two IM Account actions. **IM** opens an IM chat window in the IM program. **IM Item...** takes text or a file in the third pane to be sent via the IM program. The IM program to use is specified in the Handlers Preference pane based on plug-ins installed. Currently there are plug-ins for iChat, and Adium.



Quicksilver chooses default actions based on the type of object in the first pane. Described above is how useful this is for contact fields to be treated as text so those actions are available. Quicksilver also treats contacts as other field types (Email Addresses, IM Accounts, but not URLs) if and only if they are set in the Address Book. So if a contact has an email address sent, **Compose Email** might be the default action if it's ranked higher than **Show Contact** in the Actions Preferences. But if the contact selected in the first pane doesn't have an email address set the action might be **Show Contact** (assuming it's the default).

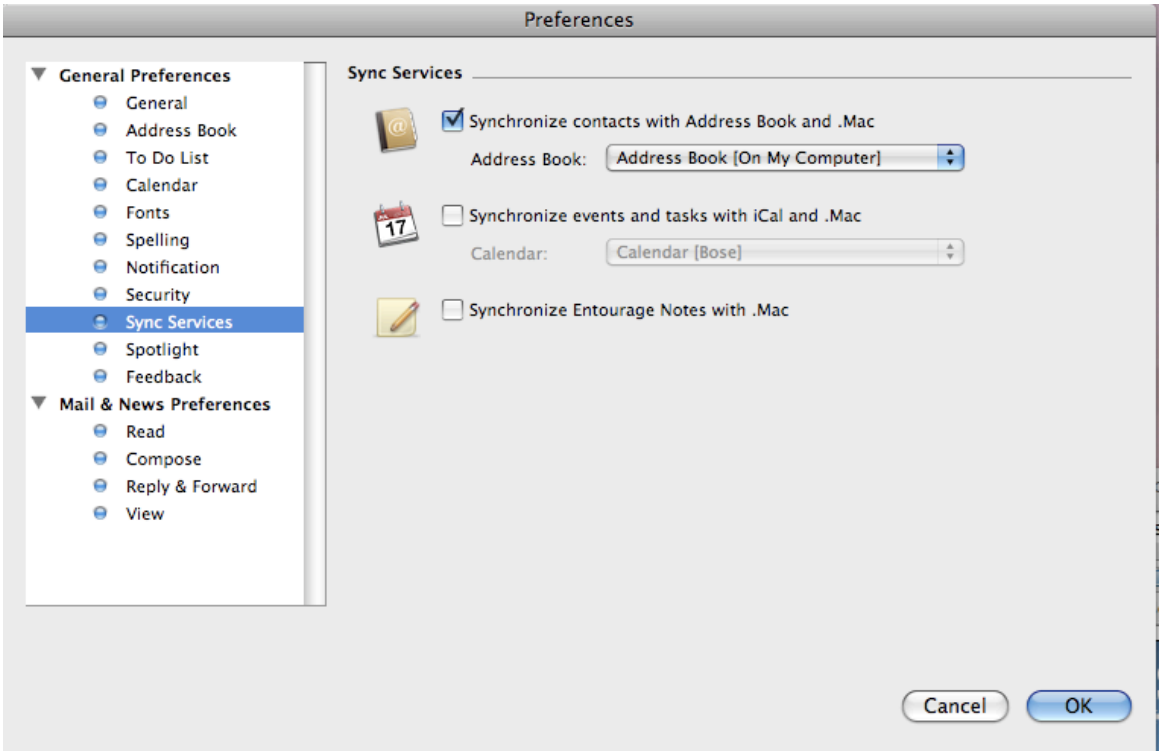
Entourage

The Entourage 2004 Module Plug-in provides support for contacts and email actions. The contact support is more limited than the Apple Address Book plug-in provides. See the email section for details about sending email from Quicksilver using Entourage. See below for using Entourage 2008 with Quicksilver.

Install the Entourage 2004 Module plug-in, start Entourage and bring up the preferences. Under General Preferences, Security make sure "Warn before allowing an external application to access the address book" is unchecked. Open the Quicksilver Catalog and under Modules make sure Entourage Contacts is checked and selected and click the ↻ (Rescan Source) button at the bottom. The Entourage contacts should now be in the catalog, though in B51 it does seem to be a bit buggy. If the contacts don't appear, rescan the catalog.

Activate Quicksilver and start typing the name of a contact to bring it up in the first pane. There is a **Show Contact** action that will bring it up in Entourage. At least in B51 typing → will not show the details of the contact. There is no **Edit Contact** action for Entourage. Other actions that do Email and IM should work for contacts as well. See the email section for details about sending email from Quicksilver using Entourage.

Entourage 2008 changed some things and Quicksilver is no longer able to catalog contacts stored in it. However, Entourage 2008 now has the ability to sync with the Address Book and Quicksilver's Apple Address Book Module can be used to find contacts in it. In Entourage 2008 Preferences under General Preferences in Sync Services, check the option to "Synchronize contacts with Address Book and .Mac" and select Address Book (On My Computer) from the drop down.



Now Contact

The Now Contact Module plug-in allows Quicksilver to scan for contacts from the Now Contact address book. Contacts found can be used as described above, in fact the Now Contact Module adds no actions of its own. Now Contact must be running for Quicksilver to index it. As of B51 there have been reports that this plug-in doesn't work very well or forgets about contacts it has scanned. Check the Quicksilver forums for help with any problems.

BuddyPop

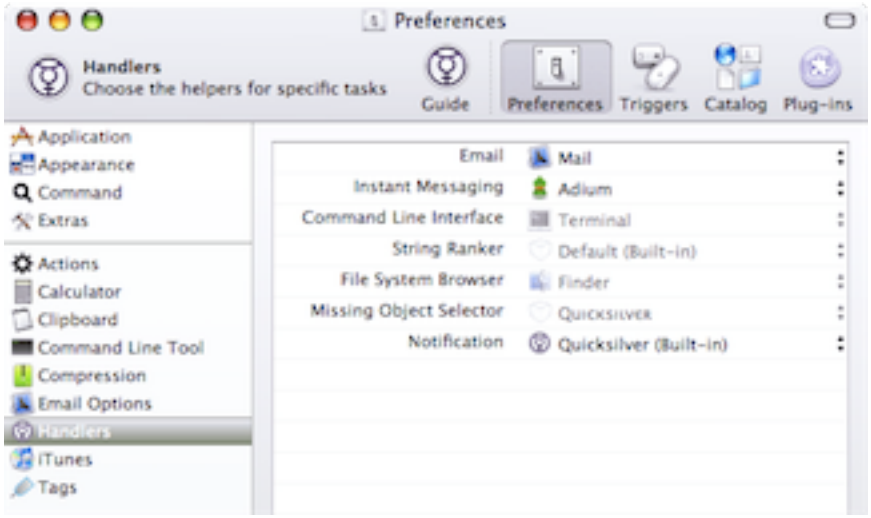
The BuddyPop Module plug-in to add the **Show in BuddyPop** action for contacts. This action will bring up the contact in BuddyPop 2.0 or greater.

Mail

Quicksilver has several common email actions (**Compose Email**, **Email to…**, **Email Item…**) that are described in detail in this section and also mentioned in Files and Folders and Contacts. They are defined in the Email Support plug-in which is normally hidden but can be seen by ⌥-clicking on the All Plug-ins set in the Plug-in preferences.

Quicksilver supports specific email applications via the Apple Mail Module, Gmail Module, Entourage 2004 Module and Mailsmith Module plug-ins. These plug-ins should all install the Email Support plug-in if needed. If the email actions don't appear in the action pane, check that they are enabled in the Actions preferences. If they aren't listed at all or the Email handler described below doesn't appear in the Handlers Preferenece, install the Apple Mail Module plug-in which will certainly install it.

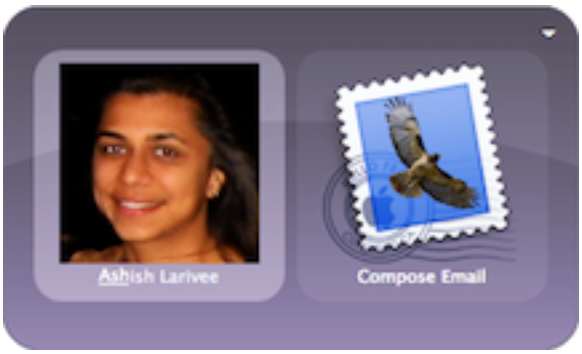
After installing the plug-in for the desired application, set the Email handler in Quicksilver's Handler preferences to the application (e.g., Mail or Entourage).



For basic Quicksilver support of other mail programs, choose "Default". This uses the system default mail application set in Mail.app Preferences in the General tab under Default Email Reader. This refers to what application handles mailto: URLs.

If a mail message is already being composed and the mail application allows file attachments to be dragged into the message, use Quicksilver to bring up the file quickly and **Paste** or drag directly from Quicksilver's first pane into the mail message.

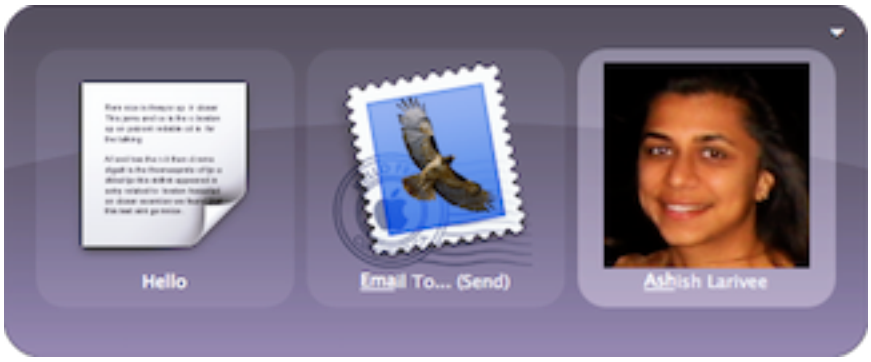
Bring up a contact or email address in the first pane and use the **Compose Email** action to open a new blank email message with the To: field filled out in the mail application specified as the Email handler. If the contact has more than one email address, the first one is used. To select a different address, bring up the Address Book contact, type → to see a results list of the contact's information and select the desired email address. This may not work with contact plugins other than the Apple Address Book Module.



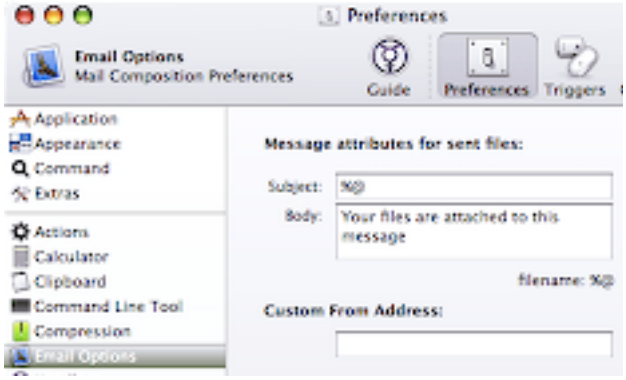
The **Compose Email** action opens a new blank message to be filled out, but Quicksilver also has the **Email Item…** and **Email To…** actions which send text and file attachments directly from

the command window. Both of these actions have three variants. They are: **Email Item... (Compose)**, **Email Item... (Send)**, and **Email Item... (Send Directly)**, and there are similar ones for **Email To…**. The **Compose** variant uses the default email client (specified in the Email Handler) and opens a message compose window to be filled out. The **Send** variant also uses the default email client but doesn't open a window; it sends the mail automatically. Both the **Compose** and **Send** variants will start the default email client if it's not running. The **Send Directly** variant doesn't use the default email client to send the message but does so directly from Quicksilver. As a result it's primary benefit is that the default email client does not need to be running. Note that if the default email client keeps a Sent folder (like Mail.app does) the outgoing message is not stored there. The **Send Directly** variant does get the SMTP configuration from the first mail account listed in Mail.app, so make sure that is setup correctly.

The **Email Item…** actions use a contact or email address in the first pane and a file attachment or text to send in third pane. The **Email To…** actions are the same but entered in reverse; a file or text in the first pane, and a contact or email address in the third. The email address can be selected by typing → into a contact or by entering text mode and typing or pasting an address.

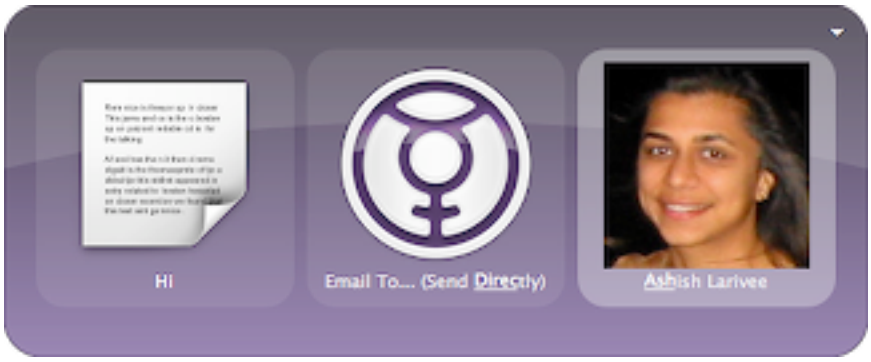


If text is being sent it's used as both the Subject and body of the message. To use a different subject use >> to separate the subject and body using the form *subject>>message body*. If files are being sent, the subject and body are set according to the Email Options Preference pane. Use the characters `%@` in the template to have them replaced with the name of the file. If more than one file is sent using the comma trick they are replaced with some variation of "# Files in Folder" such as "2 Files on Desktop" or "2 [PDF Document] in Documents".



The icons for the Email actions change based on what mail program is used to send the message. In the previous image, Mail.app is configured as the Email handler so it's icon is dis-

played for the **Email To… (Send)** action. In this image, the Quicksilver icon for the **Email To… (Send Directly)** action indicates Quicksilver is sending the mail, not Mail.app (because that's what Send Directly does). However there's a bug in B51, if there is only one mail handler installed, all the mail actions show a generic gear icon. I generated the screenshots in this section by having the Apple Mail Module plug-in installed and selected as the handler and by also having the Gmail Module plug-in installed, though even this doesn't always display correctly for me.



Mail.app

The Apple Mail Module plug-in installs actions to manipulate Apple Mail mailboxes and messages; including browsing through mailboxes in a results list. It installs a handler to select Apple Mail as the program to handle some mail actions. Note that the actions to send files and text as email are part of another plug-in called E-mail Support which is installed automatically when the Apple Mail Module plug-in is installed. For the various **Email To…** and **Email Item…** actions to work, make sure that the first mail account listed in the Maill.app account preferences has its Outgoing Mail Server (SMTP) configured correctly to send mail.

Activate Quicksilver and select Mail.app. Since it's an application the default action is to open it. Notice there is a > in the results list next to Mail.app. If Advanced Features are enabled, type → and the object pane changes to Inbox and the results list shows all local mailboxes, e.g., Inbox, Sent, and Trash as well as all custom mailboxes. IMAP mailboxes will not appear. To use them in Quicksilver drag them from Mail.app into the Catalog (e.g., under Custom). They won't appear after typing → into Mail.app but they are selectable in the object pane.

Even though there is no > next to the mailboxes, typing → will move into the mailbox and show a results list of all the messages in that mailbox. The messages are listed in reverse chronological order, i.e., newest messages on top, for easy browsing.

The Apple Mail Module plug-in installs three actions for messages: **Open**, **Delete**, and **Move to Mailbox...**. **Open** shows the message in a it's own window, **Delete** moves the message to Mail's Trash and **Move to Mailbox…** takes an argument in the third pane to specify the destination mailbox. For a mailbox object, the only Mail action available is **Open**.

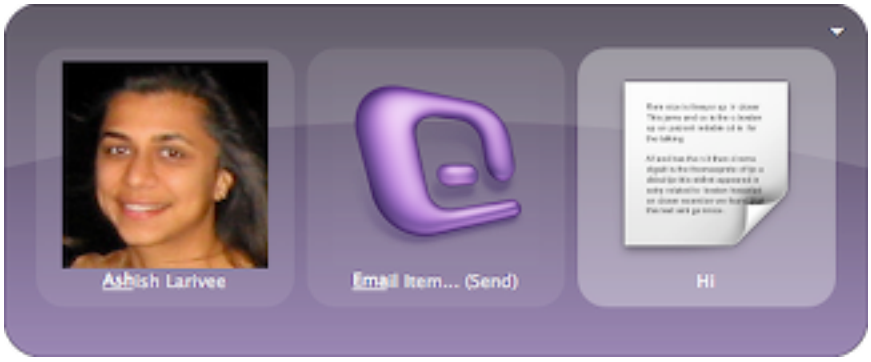
Using the **Open** action on Mail.app will open the last mailbox that was open. To easily get to the Inbox, create a trigger, in the first pane select Mail and then type → to select the Inbox, then choose **Open** as the action and bind this to a HotKey. I have ^⌘i set to this to open my inbox.

There are two actions that are only available if Mail.app is in the object pane. One is **Get New Mail** which will get new mail in all accounts. If Mail is not running it will be started. If it is hidden it will still get new mail, but Mail will remain hidden. If there is no new mail, Mail will beep, even if hidden. The other action available on Mail.app is **Open New Mail**. It doesn't get new mail but if there are unread messages it will open a new window to show the message. If there is no new mail it will just beep. As of B51 triggers using these actions can't be saved.

Mail.app keeps a list of email addresses that have been sent to. The list is viewable by choosing Previous Recipients from the Window menu. Quicksilver's Apple Mail Module plug-in also adds a catalog source under Modules called Recent Mail Addresses that accesses this list. It's useful to get to email addresses that aren't in the Address Book or other contact manager.

Entourage

Quicksilver can't navigate email in Microsoft's Entourage but it can send email using the standard email actions: **Compose Email**, **Email Item…**, and **Email To…**.



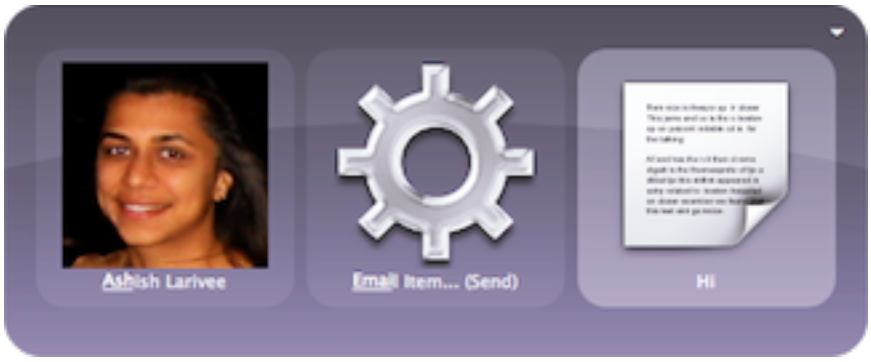
Install the Entourage 2004 Module plug-in. In Quicksilver's Preferences under Handlers choose Microsoft Entourage for Email. You may also want to set the default Mail application for the system and in OS X this is counterintuitively set in the General tab of the Mail.app preferences. At least one person has the email actions working with Entourage 2008 using the Entourage 2004 Module plug-in.

To use contacts stored in Entourage see the Entourage section under Contacts.

Gmail

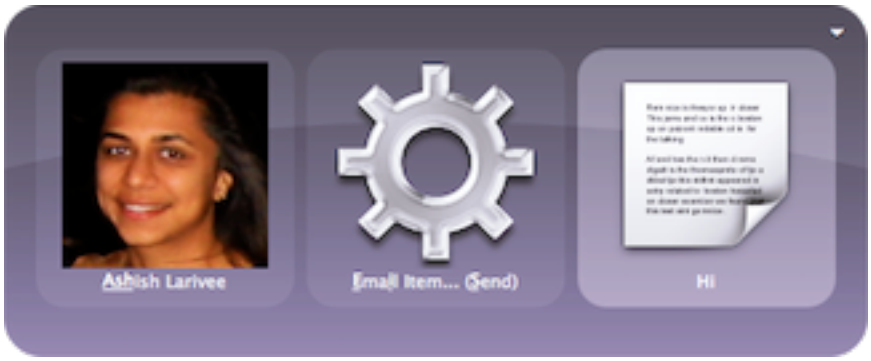
The only mail support Quicksilver has for Gmail is to use it to send email via the **Compose Email**, **Email Item…**, and **Email To…** actions. To configure this, first install the Apple Mail Module plug-in and then install the Gmail Module plug-in. As of B51, the Gmail plug-in doesn't install the Email Support Module on its own, so that's why I suggest installing the Apple Mail plug-in first. Then in Quicksilver's Preferences under Handlers choose Gmail Module. Note that the Gmail Module has nothing to do with Google Notifier.

Quicksilver will open the browser on a Gmail compose mail page, with the addresses filled in the To: field and the text "Hi" as both the subject and body of the message. As of version B51 there is no way to send an attachment with Gmail via Quicksilver. There is also no way to read mail via Gmail or browse mailboxes as there is with Mail.app.



MailSmith

The only mail support Quicksilver has for MailSmith is to use it to send email via the **Compose Email**, **Email Item…**, and **Email To…** actions. To configure this, install the MailSmith Module plug-in then in Quicksilver's Preferences under Handlers choose MailSmith.



Other Mail Applications

In Quicksilver's Preferences under Handlers there is an option “Default”. This refers to the system default mail program which can be set in the Mail.app Preferences in the General tab under Default Email Reader. This sets the **Compose Email** action to use whatever application handles mailto: URLs. TODO: do the Email to… and Email item… actions work? Does the comma trick work?

Teleflip

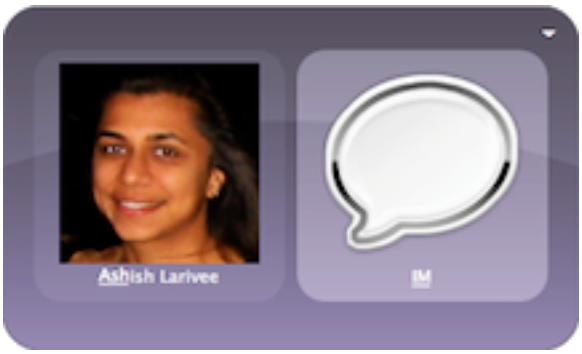
Teleflip is an online service that forwards email messages to United States cell phones as SMS messages. Currently the service is free up to 100 messages a month. Send email to an address of the form ###-###-####@teleflip.com filling in the cell phone number. The subject and body are sent, there is a 160 character limit. The Teleflip Module plug-in makes it easy to send such messages from Quicksilver by constructing the address from a contact’s phone number. It in-

stalls two actions. With a phone number selected as an object use the **SMS Text via Teleflip…** action to send the text entered in the third pane via Teleflip. The reverse action is **SMS via Teleflip…**, use it to send text in the first pane to a phone number selected in the third pane.

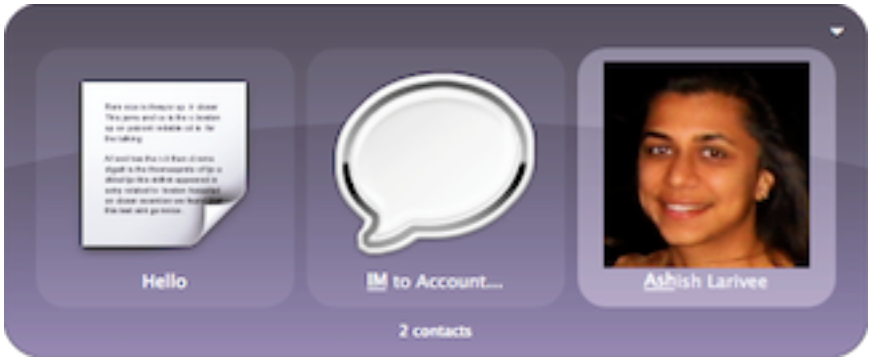
Note that a phone number object can also treated as text, so the **SMS via Teleflip…** action appears for phone numbers too. It will send the number as a message which can be convenient if that's what's desired. Bring up a phone number, type **.** to enter text mode and add more text if desired. Since this is probably less common, make sure **SMS Text via Teleflip…** is above (i.e., ranked higher than) **SMS via Teleflip…** in the action preferences.

Chat

The basic actions, **IM**, **IM Item...**, and **IM to Account...** are all part of the Chat Support default plug-in. There is a Handler called Instant Messaging which defines which application to use for the basic actions. Currently there is support for iChat and Adium, each via their own plug-ins. Install one or both of these plugins and then in Quicksilver Preferences under Handlers set the Instant Messaging handler to one of them.



The **IM** and **IM Item…** actions work on an IM address or a contact (if at least one IM address is entered, the first one is used). **IM** opens a new chat window connected to the contact. Send text or a file directly from Quicksilver by using the **IM Item…** action. In the third pane specify a file to transfer or enter text mode by typing **.** or **'** and then text to send as a message. **IM to Account…** is the reverse of this, select a file or text in the first pane and the IM account or contact in the third. This is convenient if using ⌘-esc to bring selected items into Quicksilver's first pane.



The IM actions work with contacts from the Address Book (or other contacts plug-in) that have IM addresses entered from the global catalog. Both the iChat and Adium plug-ins allow Quicksilver to show a results list of just online buddies by

typing → into the iChat or Adium applications in the first pane. Quicksilver's matching algorithm is a great way to quickly find a buddy in a large buddy list.

Some people hide the client list window of their IM application and just use Quicksilver when they want to start a chat session. Personally I keep a small buddy list of just the people I chat with often and care if they are online. For other people I keep their IM addresses in Address Book and if I want to IM them I can bring up their contact or one of their IM accounts and use the **IM** action.

iChat

The iChat Module plug-in allows the IM actions to work with iChat. Bring up iChat in the first pane and notice the > in the results list. If iChat is running, typing → into it will show a results list of all online (including idle) buddies. Quicksilver will not start iChat as a result of typing →. Begin a chat by using the **IM** or **IM Item…** actions as described above. The **IM to Account...** action also works.

The iChat module plug-in also supports the **Audio Chat** and **Video Chat** actions. These will start an audio or video chat as if the Invite to Audio Chat or Invite to Video Chat commands were selected from iChat's Buddies menu.

The iChat module plug-in has one action specific to it, **Set iChat Status** which uses the text of the object pane as the away message (although in version 0.5.0 of the plug-in this seems buggy).

Adium

The original Adium plug-in for Quicksilver only worked with Adium up to version 0.89.1. Now that Adium is past version 1.0, that plug-in has been removed. No Adium plug-in appears in the plug-in list, but a new plug-in that works with Adium 1.0 is under development and is available from this forum thread: http://blacktree.cocoaforge.com/forums/viewtopic.php?t=6395. Version 30b2 is described here. Download the plug-in, double-click it to install it in Quicksilver and in Quicksilver's Preferences under Handler, set Instant Messager to be `Adium`.

The Adium Module plug-in allows the IM actions to work with Adium. Bring up Adium in the first pane and notice the > in the results list. If Adium is running, typing → into it will show a results list of all buddies including offline buddies and entries representing groups. The first item listed is Online Contacts, type → into that to see just the online (including idle) buddies. Similarly → will work in other groups to show their members but only the online ones. Quicksilver will not start Adium as a result of typing →.

Begin a chat by using the **IM** or **IM Item…** actions as described above. The **IM to Account...** action also works. Using these actions on a group will open separate chats with each of the members. The **Audio Chat** and **Video Chat** actions are not supported because Adium 1.x does not support these features.

The Adium plug-in also provides the ability to set the status from Quicksilver. Enter text in the first pane and then use either the **Set Available Status** or **Set Away Status** actions. The text becomes the status and depending on the action the status is set to available (green) or away (red). Consider using triggers for commonly set statuses.

Calendar

Quicksilver has some limited support for creating calendar events and to-do's in iCal and Google Calendar. As of B51 the Entourage Module plug-in only supports contacts and email but not calendars. Entourage 2008 can be configured to sync with iCal, see the Entourage section in Contacts, the option is in the same place as the option to sync with Address Book. Another option is to manage a text file of todos. See the Text section for the description of the Text Manipulation Actions plug-in for some tips.

iCal

The iCal Module plug-in requires Advanced features be enabled and installs two actions: **Create iCal Event** and **Create iCal To-Do**. Using either of these actions will open iCal if it's not already running though won't open the window if it's hidden or closed. As of B51 the features enabled by the plug-in are limited. There's no way to see or interact with existing events and to-dos, typing → to move into iCal, does nothing.

To create a new to-do, activate Quicksilver, enter text-mode by typing `.` or `'` and enter the to-do text. Tab to the second pane and choose the **Create iCal To-Do** action, tab to the third pane and choose the calendar for the to-do.

A specially formatted to-do text allows a priority and a due date to be entered. Precede the text with zero to three exclamation marks to set the priority in iCal. Zero !'s sets a priority of None, ! is Not important, !! is Important, and !!! is Very important. A due date is set by preceding the text (and any priority !'s) with a date and two hyphens. The date is somewhat flexible, allowing `8/1`, `8/1/2006`, `tue`, and `next wed`. In the example image I use the text: `8/1--!Upload User Manual`. As of B51 it seems it always uses a US-centric date format, *month*/*date*, even if the system is set to a European convention of *date*/*month*. The only work around for this is to use month names such as `Feb`.

Events are similar to to-dos but have no priority and can have a time as well as a date. Creating a new event is similar to creating a to-do; put text in the first pane, use the **Create iCal Event** action and specify a calendar in the third pane. The format of the text string is: *datetime*`--`*text*, that is a date time string separated from the description of the event by two hyphens. If no date or time is specified the event is created now. If just a date is entered, the event is created at noon on the specified day. Times are entered as `1pm` (`1p` is not valid), `1:00` (which is am), or `1:05pm`, or `22:30`. All created events are 1 hour long and Quicksilver provides no way to set a

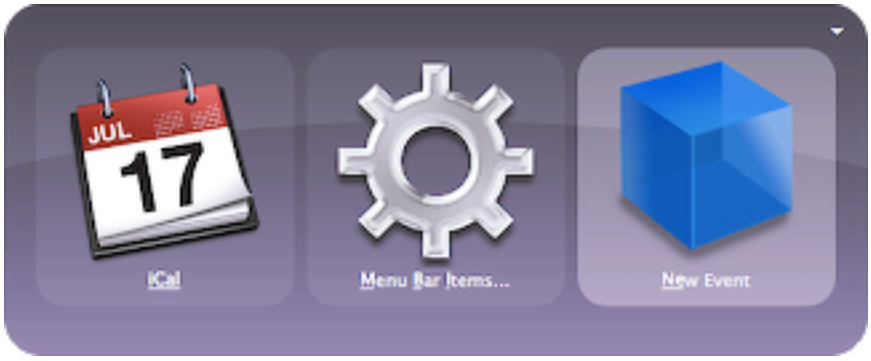
different duration. The created events use the whole string as the event name, so if `wed 6pm--dinner` is entered, that's also the name of the event.

The above is the supported syntax but it seems that by not using `--` the date and time can be put anywhere in the string and it will still work. E.g., `dinner - wed 6pm` or `wed dinner 6pm`, and the event is created at the right time and with the string at the front which is a little friendlier if it gets cropped in the iCal display.

With this plug-in there's no way to specify attendees, repeats, alarms or other iCal fields. There are a few scripts on the Quicksilver site that can be installed that will prompt for more info, but they are under-documented and a little complicated to get setup. Rather than enter event info into an AppleScript dialog I think it's easier to just enter the info into iCal.

If entering information directly into iCal seems easier than remembering a text syntax that supports only some of the fields, then try these commands which require the User Interface Access plug-in be installed and iCal already be running:

- iCal (**Menu Bar Items…**) New Event
- iCal (**Menu Bar Items…**) New To Do



They will make iCal be the active application, opening the window and bringing it to the front if hidden or minimized. The new event or to-do is created and selected, ready to have the name changed and other information entered. Making triggers for the above would be very convenient but as of B51 triggers using the **Menu Bar Items…** action and the third pane filled in don't save. Hopefully this will be fixed in a future version.

For a little more flexibility with creating events, some people like the free OSX service Calendar Creator. This installs two Services, Add Calendar Event and Add To Do. See Calendar Creator's documentation for what it can parse. With the Quicksilver Services Menu Module plug-in installed any text can be sent to these services via Quicksilver.

Google Calendar Module

For those that eschew iCal and prefer Google's Calendar, there's the Google Calendar Module plug-in. It adds one action **Google Calendar Event** that takes text as the object in the first pane and sends it to Google Calendar which can parse text like "`Dinner with Michael 7pm tomorrow`".

If needed, Quicksilver prompts for Google Calendar login info. In the event of login problems, use Keychain Access to check the keychain entry for calendar.google.com.

As of B51, people have reported various problems with this plug-in. The plug-in does not use Google Calendar's Quick Add function so the parsing of information isn't always correct. Some people report the wrong date being entered. Also the entire string entered is used as the text of the event and newly created events don't inherit the default reminder setting.

Web

The simplest way to use Quicksilver with the web is to use text mode to enter a URL in the first pane and use the **Open URL** action to open the page in the default browser. It should be the default action for URLs (if not raise its rank in the Action preferences).

There is also an **Open URL With…** action which allows another browser (or any application) to be selected in the third pane. This action is disabled by default, enable it in the Actions preferences by checking its box.

Since URLs are just text, there are several easier ways to get one into the first pane. Copy one from some application and paste it into the first pane with ⌘V. With ⌘-esc configured correctly (see Combining Activation and Selection), just select text in a Cocoa application and type ⌘-esc to activate Quicksilver and automatically paste the URL into the first pane. The comma trick also works, so multiple URLs can be entered separated by commas and opened with just one command.

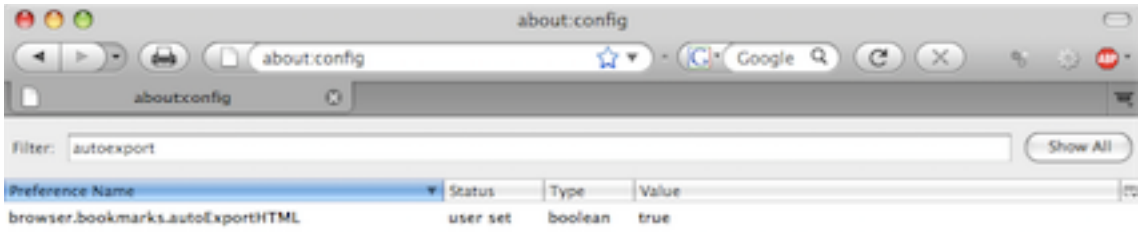
URLs opened from Quicksilver will appear in a new browser tab or window depending on how the browser is configured. Safari will use a new window by default. To change it to use tabs open Safari's preferences and in the General tab for "Open links from applications" choose "in a new tab in the current window". In Firefox go to preferences, Tab Browsing and for "Load external links in" choose "New Tab". In Camino open preferences, Tabs, and for "Link from other application" choose "Opens in a new tab in the frontmost window".

Those are the basics but Quicksilver integrates with the web in several more ways. Browser plug-ins make bookmarks available to the catalog for easy opening and some provide more features. The Web Search Module plug-in lets web searches be entered directly into Quicksilver with the results shown in the browser. Several online services have plug-ins of their own making their data accessible to Quicksilver instead of using a web form. Combine these with triggers and proxy objets and Quicksilver blurs the line between native and web applications.

Web Browsers

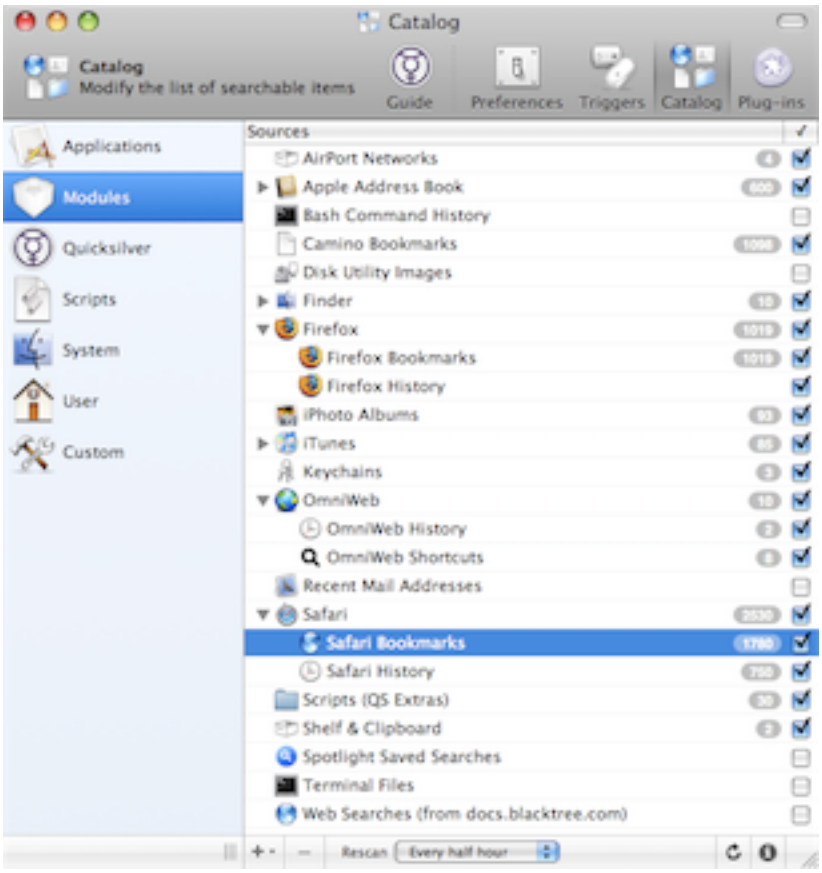
There are plug-ins for four web browsers: Safari, Firefox, Camino and OmniWeb. These all provide access to bookmarks stored in the browser and some number of other features. Install the plug-ins only for frequently used browsers. It's fine to install more than one, but be cautious of the catalog getting too large if there are a lot of bookmarks in each (perhaps duplicated). Consider using a social bookmarking site (like del.icio.us) to store bookmarks and see the next section for how to use one with Quicksilver. The image below shows all the browser bookmarks enabled merely for demonstration purposes.

Note that the Firefox plug-in was written for Firefox 2 and looks for bookmarks in the html file `~/Library/Application Support/Firefox/Profiles/bookmarks.html`. Firefox 2 allows this file to be in different locations (which allowed multiple profiles) and if someplace else, Quicksilver can't find it. To solve this create a copy of the catalog source as described in the Catalog section and select the correct location in Source Options. Firefox 3 stores bookmarks and history in a SQLite database and not in a file, so the plug-in can't find them at all. However, Firefox 3 can periodically save the bookmarks (not history) to a `bookmarks.html` file. In the



Firefox location bar enter `about:config` and hit `return`. In the Filter field that appears type `autoexport` and `browser.bookmarks.autoExportHTML` should appear. Double-click it so the value column changes to `true` (or right-click it and select `Toggle`). Finally restart Firefox and Quicksilver and there should be a `bookmarks.html` file and Quicksilver should be able to index the bookmarks. Firefox 3 will write out the `bookmarks.html` file whenever it is quit. Note that several other browsers are based on Firefox and if they use the same `bookmarks.html` file or `autoExportHTML` option the Firefox plug-in should work with them provided the correct path to the file is given.

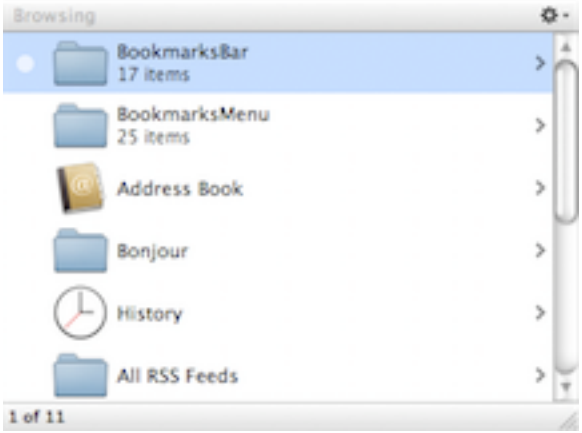
With a browser plug-in installed, a catalog source is added to the modules set putting bookmarks into the catalog. Make sure the box is checked and a number appears to the left of it (if not click the ↻ button at the bottom of the window to rescan the source.



The Firefox, OmniWeb and Safari plug-ins also enable catalog sources for the browser's history (the Camino one does not). Accessing the browser history via Quicksilver can be a remarkably convenient way to find a recently visited web site by just typing in part of its name. If the browser allows configuration of the size of the history, this can allow visiting sites from several days ago, but be cautious of letting the catalog get too large and slowing down Quicksilver.

With URLs from either bookmarks or history the **Open URL** and **Open URL With…** actions work as described above. None of browser plug-ins provide their own actions, so there are no actions to add bookmarks to a browser (maybe AppleScripts could be written to do this if the browser supports it). Unfortunately email actions are not available for URLs. However, URLs in the first pane can be changed into text by typing **.** or **'**. Now that the URL is plain text, tab to the second pane and the **Email To…** actions are available. Some browsers like Camino and Opera install system services such as **Open URL in Camino** and **Open URL in Opera**. The Service Menu Module plug-in will turn these into Quicksilver actions which may be faster to use than **Open URL With...**.

Safari organizes bookmarks into various collections. These can be seen by choosing the Show All Bookmarks command in the Bookmarks menu or using the HotKey ⌘B. In Quicksilver, with the Safari application selected in the first pane, typing → will show a results list with all these collections. Navigate through them with →. This works if Safari is running or not, but does not work for any of the other browsers (it used to work for Camino but it doesn't seem to with Camino 1.6.1).



Safari's Address Book collection is useful if the URL fields in contacts are used. Quicksilver

doesn't index these directly in the catalog, instead, bring up the contact and → into it to select an URL. For faster access to these URLs, create a trigger with Safari's Address Book collection and the **Show Contents** action, a results list appears with all the URLs in Address Book, ready to be selected.

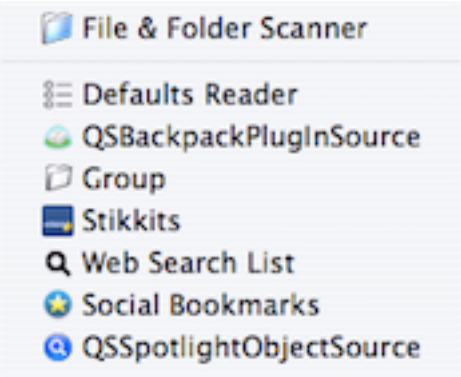
This table summarizes Quicksilver's support for web browsers. Safari is clearly the best supported and this is one of the reasons I use it as my browser instead of the alternatives.

	Safari	Firefox	Camino	OmniWeb
Bookmarks	Yes	Yes	Yes	Yes
History	Yes	Yes	No	Yes
→ into .app	Yes, collections	No	No (used to be Yes, but flat)	No
Proxy Object for Current URL	Current Web Page	None	None	OmniWeb Active Page
Other			**Open URL in Camino** action via Services Menu Module	Enables web searches from OmniWeb shortcuts.
Current Selection Proxy Object Supported	Yes	No	Yes	Yes

Online Services

The Social Bookmarks plug-in (which replaced the del.icio.us Bookmarks plug-in) makes del.icio.us and/or ma.gnolia bookmarks available in the Catalog. There is no catalog entries installed in the Catalog under Modules, instead in the Catalog under Custom create sources for

del.icio.us and ma.gnolia accounts. In the Catalog, click the + button at the bottom and choose Social Bookmarks from the menu that appears. Select the newly created entry and click on the

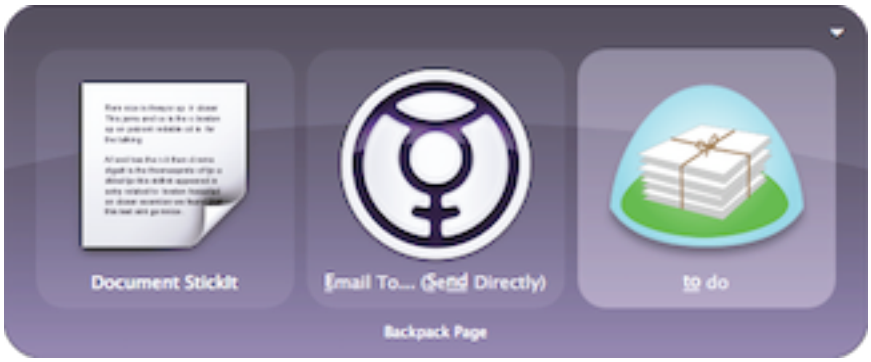


button to open the drawer. In the Source Options tab choose the service and enter the Name and Password of the account. Note the plug-in only works with passwords that are 8 characters or less and people have reported problems with non-alphanumeric (e.g., # and ^) characters in the password. Click the (Rescan Source) button and the bookmarks should appear. Unfortunately as of B54, this plugin is broken for del.icio.us accounts since their switch to delicious.com. The option to Include Tags puts the tags in the catalog as well. Typing → into one shows a results list of all bookmarks with that tag. Quicksilver has no actions to add a bookmark to to a del.icio.us or ma.gnolia account, instead use the standard mechanisms for that. A trigger for a post to del.icio.us bookmarklet and the **Run Javascript** action could be convenient.

Social bookmarks services provide an Internet-based service to centralize bookmarks so that different browsers (on different machines) have the same bookmarks. However, in the single machine case, Quicksilver can be used to centralize bookmarks used by several web browers (and perhaps a social bookmark account as well). Create a custom catalog folder and put all the related catalog sources together (for safari bookmarks create a copy of the preset - go to the attributes tab of the info drawer for the button). Now browse into the catalog entry for the Bookmarks folder (it will have a "(catalog)" after its name, make sure Quicksilver Catalog Entries are enabled in the catalog under the Quicksilver set and advanced features are enabled.) and with the **Show Contents** action search all the bookmarks at once.

Backpack is an online service allowing the simple creation and editing of web pages (similar to a wiki). The Backpack Module plug-in lets Quicksilver index pages from a Backpack account in the catalog. Similar to the Social Bookmarks plug-in, it creates a new type of catalog source. In the Catalog, click on the + button and choose QSBackpackPluginSource from the menu. Select the newly created entry and click on the button to open the drawer. In the Source Options tab enter the Backpack account Name and API Key (aka password). Click the (Rescan Source) button to see the pages added to the catalog. With the QSBackpackPlugInSource (Catalog) object selected in the first pane, typing → shows a results list of all the Backpack pages. Use the **Open URL** action to open the page in a browser. As of B51, the Backpack Module plug-in does not support https so disable it in the Backpack Settings page (which will be less secure).

Backpack pages have an email address associated with them. Messages sent to that address are added to the page. Quicksilver adds these addresses to the catalog automatically and shows them as Backpack pages. Use the regular Quicksilver **Email To…** and **Email Item…** actions to upload text, images or files by just sending them to the Backpack page. Add Backpack dated notes and to-dos, by using Backpack's specially formatted subjects and message bodies. To add a note to a page, select the page, choose the **Email item…**



(Send Directly) action and enter text-mode for the argument. Enter the note as follows: `note:`*`note-title>>note-body`*, which uses the *`subject>>message-body`* syntax that the Email actions understand.

Stikkit was another online site similar to Backpack with a Quicksilver plug-in, however it shut-down in December 2008.

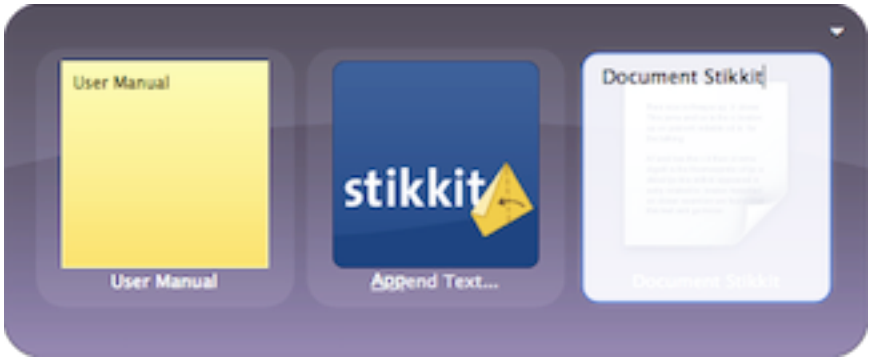
TODO: (Remove this) Web pages (called stikkits) can be edited and have email addresses as well. With the the Stikkit plug-in, Quicksilver can communicate with Stikkit. Similar to the plug-ins above, it creates a new type of catalog source. In the Catalog, click on the + button and choose Stikkits from the menu. Select the newly created entry and click on the ❶ button to open the drawer. In the Source Options tab enter the Stikkit Email and Password. Click the ↻ (Rescan Source) button to add Stikkit pages added to the catalog.

Bring up the the Stikkits (Catalog) object in the first pane and type → for a results list of all stikkits.. Consider creating a trigger for Stikkits (Catalog) (**Show Contents**) to make this easy. The Source Options tab of the catalog source also includes an option to have Stikkit tags included in the catalog. With this enabled, bring up a tag in the first pane and type → to see a results list of all stikkits with that tag.

To create a new stikkit, enter a name in text mode in the first pane and use the **Create Stikkit** action. To rename a stikkit, select it in the first pane and use the **Edit…** action, entering the new name in the third pane with text mode. The new name won't be in the catalog until after a rescan.

With a stikkit in the first pane use the **Open URL** action to open the stikkit in the default browser. The **Open URL With…** action works as well.

The Stikkit plug-in adds new versions of the **Append Text…** and **Prepend Text…** actions that have the Stikkit icon. Use these to add text to a stikkit by choosing a stikkit in the first pane and entering the text in the third pane. Unfortunately there are no reverse versions of these actions allowing the text to be in the first pane and the stikkit in the third. However, it should be possible to create triggers with the stikkit in the first pane and the Current Selection proxy object in the third pane, similar to as described in the Web Searches section below. Quicksilver gives a notification whenever a stikkit is updated.



The Stikkit actions are entered with a low default rank. If frequently used, consider increasing their rank in the Actions preferences.

cl1p.net is a free online service for temporarily posting any text to their web pages. One use is uploading error messages and posting just the URL to a forum or irc channel instead of the full text. The Cl1p Upload plug-in adds a **Post to Cl1p…** action that makes this easy. Select a file or enter text into the first pane and choose the **Post to Cl1p…** action. Type `return` to upload it to an anonymous cl1p page or Tab to the third pane to enter a page name in text-mode. After the command executes, the default browser will open to the newly created page. To save the page, click the Save button.

Web Pages

Here's another hidden feature in Quicksilver. Bring up a web page in the first pane, that is a bookmark, URL or HTML page from the filesystem, and type → to see a results list of all the links on that web page. Quicksilver actually gets the page (over the network if necessary), parses through it to find the links and presents them in a results list.

It can be a little slow if it has to get the page from a remote machine, particularly if the page is large. While this is happening, some interfaces present a "working" indication such as a spinning gear, others do not. Typing ⌘. (thats a period), will supposedly stop a lookup that's in progress but in B51 it doesn't seem to work.

For each link in the page there are two lines shown. The first is the text of the link and the second is the URL. If the link is on an image, there is no title but instead a very scaled down version of the image is shown. Favicons (aka website icons) are not shown.

This seems like a very cool feature in search of a good use case. In the example here I'm browsing the Quicksilver online documentation. I can get to links marked Tutorials and Overview, but there is a lot of other stuff in there and it's a little awkward to browse. Still if I know I'm looking for Tutorials I can just type it in and Quicksilver's matching algorithm will find it.

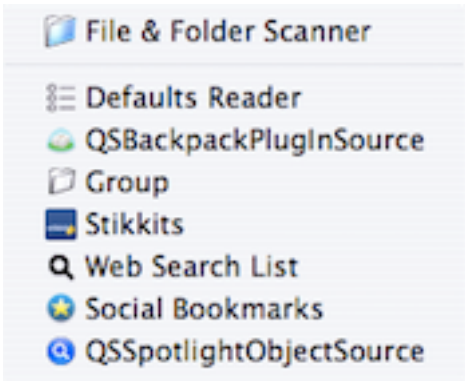
I have a bookmark for the root of the Quicksilver Google Group. If I want to send someone the URL of the FAQ I know is on that page, I could bring up the bookmark in the first pane, type → to see the links, type `faq` to bring up that link and use the **Paste** or one of the **Email to…** actions. While it's a contrived example it is almost convenient enough to use.

TODO: what plugin enables this? is it safari? is it browser specific? are advanced features required?

Web Searches

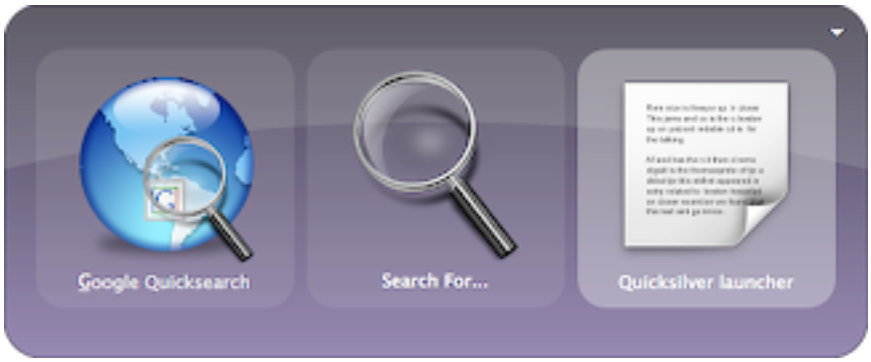
With the Web Search Module plug-in, Quicksilver can send search terms to search engines and have the results shown in the default browser. This is probably my favorite Quicksilver plug-in. I use it to search Google, the Internet Movie Database, Wikipedia, Amazon and many other sites. Rather visiting the sites first (even via a bookmark and Quicksilver) and clicking in a field and typing , it's easier to just bring up Quicksilver, type the string and type return. With triggers and proxy objects this can be much faster.

Quicksilver uses specially formated URLs that include the web site to search and a placeholder where the search terms are substituted. It refers to these special URLs as Web Searches and can store them in the catalog. They are shown as a regular URL icon with a magnifying glass. I'll explain how to use, find and creates these Web Searches below.



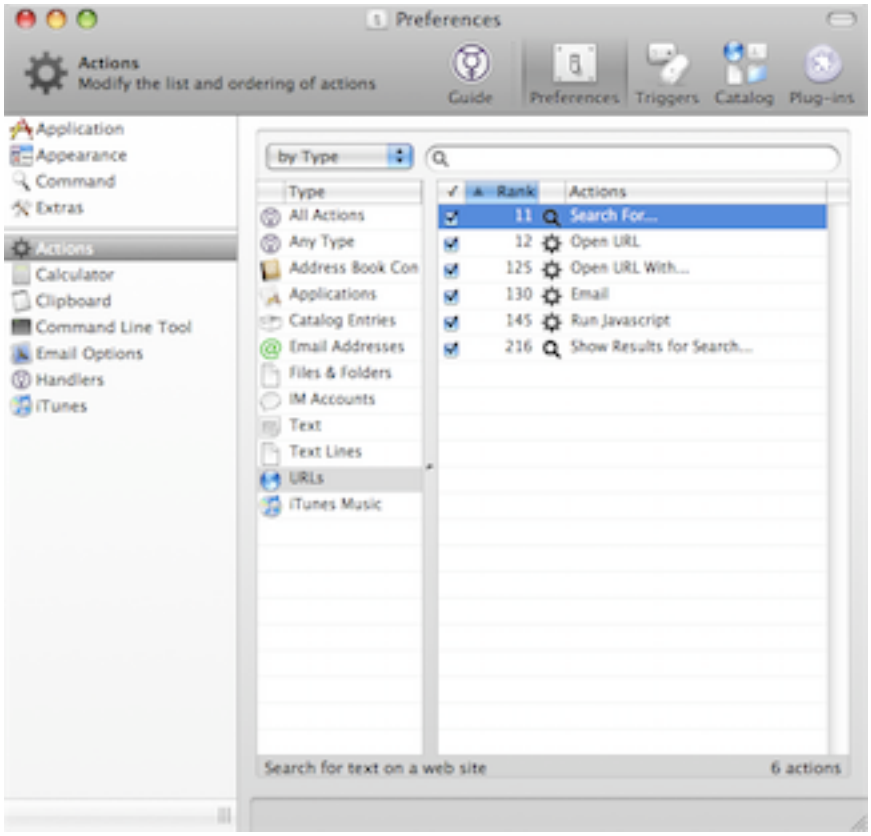
The Web Search Module plug-in adds three actions to Quicksilver: **Search For...**, **Find With...** and **Show Results for Search...**. The plug-in has no configuration options. It adds one catalog source under modules, “Web Searches (from docs.blacktree.com)”. Check the box to add all the listed web searches to the catalog from the blacktree site, there are currently over 600 of them, so it may take a while (and an Internet connection is required). That list of web searches can be found at this web page. Also the Add Custom Source drop down (the + button at the bottom of the catalog) gets an option to add a custom Web Search List to the catalog (Quicksilver must be restarted after the Web Searches Module plug-in is installed for this choice to appear).

I assume the Web Searches (from docs.blacktree.com) catalog source is checked or there are some manually configured custom web searches (described below). To perform a Web Search select the search by name (e.g., “google” or “amazon”) in the first pane. The action should default to **Search For…**, so type `return` (or `tab` twice) to get to the third pane (it will already be in text mode) and type the search terms. Type `return` to perform the search; the results are displayed in the default web browser.



While not well known, this is a standard, system-wide alternative clipboard that many applications automatically populate with text entered in search fields. In those applications that support it, selecting text and typing ⌘E (or sometimes ⇧⌘E) will explicitly copy text into this Shared Find Clipboard instead of the regular clipboard done with ⌘C. When using Quicksilver's **Search For…** action, the third pane defaults to whatever is in the Shared Find Clipboard. The result is that Quicksilver sometimes just knows what it is you want to search for. In B54 this only sometimes works, sometimes the field is blank.

If **Search For...** isn't the default action for a web search URL, make it so by finding it in the results list (while in the second pane) and ^-clicking (or right-clicking) and choosing "Make Default"; or change the ranking in the Action Preferences. The Actions Preferences pane doesn't differentiate between regular URLs and web search URLs (those with *** or %s or beginning with qss- or qssp-) however Quicksilver does. I have **Search For...** ranked 11 and **Open URL** ranked 12. If I bring up a regular URL the default action is appropriately **Open URL** and if I bring up a web search URL the default action is appropriately **Search For...**.



The **Find With...** action is the reverse of **Search For...**, the search terms go in the first pane and the web search site in the third, This is convenient when using ⌘-esc (see Combining Activation and Selection) to bring the selection into the first pane.

The third action, **Show Results for Search...** is similar to **Search For...** but instead of the results showing in the default browser, they are shown in a results list in a new Quicksilver command window (as described above in Web Pages). Unfortunately in B51, **Show Results for Search…** seems to be broken, it hangs when run and it is disabled by default in the Actions preferences.

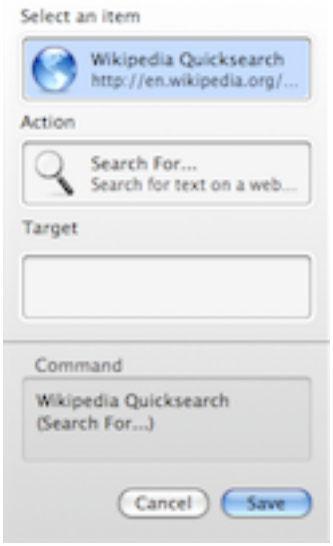
I found I used some searches ("Google", "Wikipedia", "Amazon", and "IMDB") very often and some other searches frequently but not as often and some other ones rarely. I used different Quicksilver techniques to appropriately speed up using them.

The rarely used ones I just keep in the catalog and when I want to use one I just activate Quicksilver and type to select it in the first pane and use Search For… (or correspondingly in the third pane when using Find With…). For those I use a little more frequently I use techniques described above in Helping the Matching Algorithm. E.g., activate Quicksilver, type the letter to use

for the web search site (e.g., g for google), find the web search site in the results list, ^-click (or right-click) and choose “Set as Default for “letter””. In some cases the list is too long and I'm not willing to scroll far enough to find what I want. E.g., I use z to perform an amazon search and since it's not the first letter, it's not high up in the results list. This is what the the Assign Abbreviation action is for. Bring up the web search site in the first pane (in this case amazon), choose the Assign Abbreviation action in the second pane, and enter the desired letter in the third pane (in this example z). This only works if the item does appear somewhere in the results list when the abbreviation is typed. It won't work to assign book to an amazon search. To do that see Adding Web Searches Manually below and create a new search called book with the amazon URL.

^⌘G	Google
^⌘W	Wikipedia
^⌘Z	Amazon
^⌘F	IMDb
^⌘S	Image Search
^⌘E	Google Maps

Triggers are used to provide really fast access to searches and I use these for my most frequently used searches. I have two triggers for each; the first is the web search site, the **Search For…** action and a blank third pane. E.g., on ^⌘W I have a Wikipedia search and the **Search For…** action. It's important to have a blank third pane in the trigger definition so that Quicksilver will open a command window so the search terms can be entered. If there's something in the third pane, Quicksilver will just search for it. To assure the third pane is blank when configuring the trigger, tab to the third pane and type ⌘X, this literally cuts out anything that's there. With this configured, I type ^⌘W at any time, enter text and hit return and the Wikipedia page pops up.



Of course instead of typing into the third pane, paste with ⌘V works just well and and is a little faster if the search terms have been copied from some place. In that case it's just three key strokes to do the search, ^⌘W, ⌘V and return. But we can get it down to just one using proxy objects and another trigger.

The Current Selection proxy object (which requires advanced features to be enabled), represents the highlighted text in the current application; in other words, the current selection. I have Current Selection, **Find With…**, Wikipedia web search configured on ^⇧⌘W. I highlight text in any cocoa application (like Mail or Safari but unfortunately not Word), type ^⇧W⌘ and the Wikipedia page for that search term appears in the default browser. This is amazingly convenient to use, but there's a trick to create it.

If the first pane has the Current Selection proxy object, the **Find With…** action won't appear in the second pane (because the selection might not be text). To get around this, first activate Quicksilver (using ^-space), type **.** to go into text mode, type some text, `tab` to the second pane and select **Find With…** and while the second pane is active type ⌘C to copy the **Find With…** action (yes that works!). Now go into some application and select some text (do not copy it, just select it!). This prevents the Current Selection proxy object from going into a beeping loop when it's used (if that happens, just wait for 8 beeps and hopefully Quicksilver doesn’t crash). Now create the trigger, open the trigger pane, create a new trigger, select the Current Selection proxy object in the first pane, `tab` to the second pane and

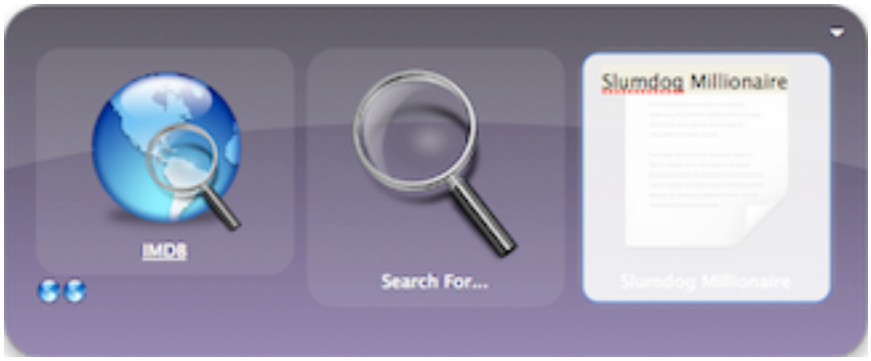
type ⌘V to paste the **Find With…** action, `tab` to the third pane and choose the web search site. Click Save and assign a hotkey. As of B51 it doesn't seem possible to create this trigger using the **Search For…** action because the third pane only wants text and won't accept the Current Selection proxy object (even by pasting it).

I use the above two triggers frequently. ^⌘W to enter search terms for Wikipedia and by adding a ⇧ to that I search for some highlighted text. I have similar pairs of triggers for the others listed on the right (all have a corresponding version with ⇧ using the current selection for the search). The maps one with a ⇧ included is great for highlighting an address on a web page or in an email message and hitting one key (^⇧⌘E) to get a map of the address. If it's multiple lines it might work or Google might ask to narrow the search; either way it's faster than cutting and pasting or retyping the address.

For less frequently used searches I use ⌘-esc and **Find With…** action (it's my default action for text) and then tab to the third field and select a web search site. For many I've configured a single key abbreviation using **Assign Abbreviation…** to make this fast. If someone sends me a movie recommendation I'll select the title of the film, ⌘-esc, tab to the third pane and type n to bring up a Netflix search. I hit return and the browser opens showing me the film and letting me add it to my rental queue with one click.

The important thing is to configure the searches that you do most often, regardless of what they are. If you search for weather info, air port codes, videos, genes, politicians, laws, astronomical objects, or whatever, setup web searches with similar triggers and you'll be working faster than ever.

With web searches, the standard Quicksilver features of ⌘-esc, triggers, and proxy objects are really useful. Another Quicksilver feature is also useful, the comma trick. With it Quicksilver can search for the same terms on several web sites at the same time. Activate Quicksilver and in the first pane select a search (e.g., Google), type `comma`, select another search (e.g., Wikipedia) and continue for all the desired web search types. Then tab to the second pane and select the **Search For…** action and then tab to the third pane and enter the search terms. Type `return` to have all the searches open at once in the default browser (in different tabs or windows depending on how it's configured). The comma trick also works in the third pane when using the **Find With…** action.



Quicksilver isn't the only application to try to make web searching easier. Most browsers now have a search box in the toolbar that can be configured to search different sites, usually by clicking on a button to drop down a menu of sites. Some browsers support keywords on bookmarks that can can do searches from the address bar by entering a keyword (like g for google or w for wikipedia) followed by the search terms. This is easy to use though Quicksilver has some advantages since it works when in any application, not just in the browser.

Safari does not support keyword searches natively but several add-ons exist to do it. There are Quicksilver plugins for three of these: Acid Search, SafariStand, and Sogudi that allow these

keyword searches to be used with the **Search For…** and **Find With…** actions. Some of these plug-ins haven't kept up with changes in the Safari extensions. If they aren't working, see the Quicksilver forums for help. As of B51 the Acid Search plug-in is known to be broken.

TODO: The OmniWeb plug-in also enables web searches based on OmniWeb shortcuts.

An advantage of using Quicksilver web searches for this is that the web search list is independent of the browser. For those that change browsers often, there's no need to keep the keyword searches unique across several browsers. It doesn't work across multiple machines, so in that case find another means to keep keyword searches in sync (e.g., a social bookmark site).

Adding Web Searches Manually

The built-in web search list is pretty large, but it can be extended if needed. The list is at this web page, Quicksilver reads this web page when it updates the catalog. There is a link at the top of that page to add a web search to that list by editing the wiki page. Unfortunately due to vandalism that list is now read-only and can't be edited (though check if this changes).

In addition to or instead of that list, a custom catalog source of web searches can be configured. I use one instead of the default web list of 600 items to keep my catalog smaller. Another advantage is that it will still work even if the Quicksilver site is down or without internet access, which can be useful if the searches are for a local (e.g., corporate) network

To understand how to add custom web searches, it's necessary to understand how search forms and URLs work on the web. When search terms are entered in a form, the browser sends a request to the search engine usually as a specially formed URL which appears in the browser's address bar on the search results page. Here are two examples of searching for "Salma Hayek", the first is for the Internet Movie Database and the second is for Google News.

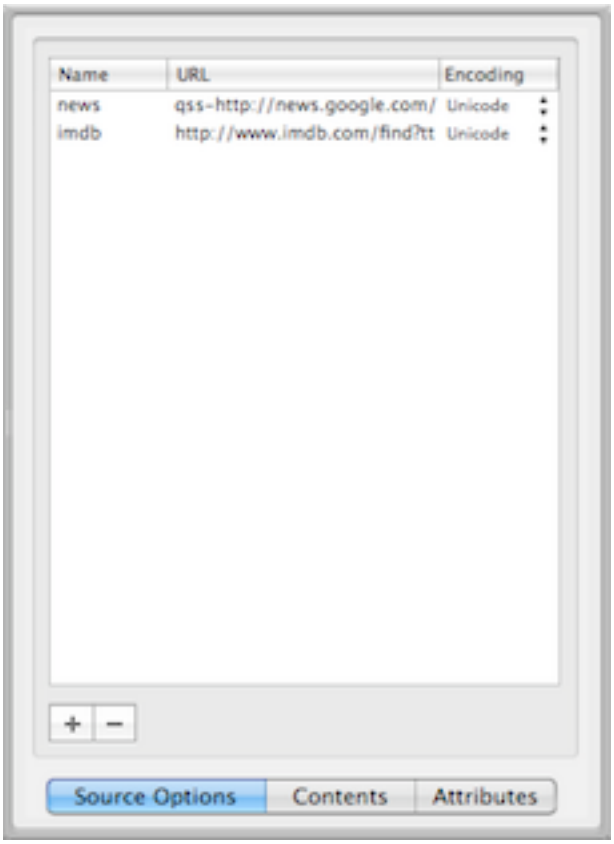
> http://www.imdb.com/find?tt=on;nm=on;mx=20;q=Salma+Hayek
>
> http://news.google.com/news?hl=en&edition=us&q=Salma+Hayek&btnG=Search+News

They're different, but the search term "Salma+Hayek" in both of them. Notice also that the space in the search terms was converted to a +. Quicksilver web searches work by changing the URL to include a special indicator for where Quicksilver should substitute the search terms (and do conversions like space into +). This indicator is three asterisks (`%s` works too). Here are the above URLs suitable for use as a Quicksilver web search:

> http://www.imdb.com/find?tt=on;nm=on;mx=20;q=***
>
> http://news.google.com/news?hl=en&edition=us&q=***&btnG=Search+News

An easy way to generate these is to go the web page of interest and search for `***` and then look in the address bar of the results page and copy the URL. Some search pages hide this info, but if the URL looks similar to the above it's usable. To add it to Quicksilver, open the Catalog, choose Custom and click the + button at the bottom. From the drop down list that appears, select a new Web Search List (it might say the ugly text "QSWebSearchSource" instead). A new

catalog source called Web Search List appears. Select it and if the info drawer is not open, click the button on the bottom to open it. In the Source Options tab click + to add a new item and then enter a name for the web search in the first column and the specially formatted URL in the second column. For the above two examples I use the names “imdb” and “news”. Finally, click the (Rescan source) button and make sure a number appears to the right of the new web search source in the catalog. The web search should now be usable with the **Find With...** and **Search For…** actions.



In addition, Quicksilver registers itself to handle URLs that begin with qss-http: and qssp-http:. If the URL begins with one of these they will open in Quicksilver instead of the browser. If it has three asterisks in it, it will be treated as a web search. The second variety (with the p) does the search using the HTTP POST method instead of the default GET method; some web sites require this. It’s best to convert all manually added web search to use one of the qss- style URLs but not required. Note that any valid URL scheme should work in a web search.

Here’s another example. Google has the ability to limit a search to a specific site.

qss-http://www.google.com/search?q=site%3A***&ie=UTF-8&oe=UTF-8

Using the above web search, enter as a search term the site to search (e.g., `abc.com`) followed by a space and the term to search for (e.g., `Salma Hayek`). Of course, this works with either the **Search For…** or **Find With…** action.

There’s also another way to keep track of URLs and web searches in Quicksilver. Add a custom Catalog source that uses the FIle & Folder scanner and select an .html file as the source. For the Include Contents popup choose HTML Links. Quicksilver will scan the .html file and find all the links and include them in the catalog. For the cost of having to edit an HTML file, this method keeps all the links (instead of in Quicksilver's preferences) in a simple file that's easy to backup or sync between multiple machines.

NetNewsWire

The NetNewsWire Module (β) plug-in works with the NetNewsWire newsreader (not the Lite edition). It adds a new catalog source under Modules called QSPresetNNWSubscriptions which has one item called New Headlines. Bring it up in the first pane and then type → (or use the **Show Contents** action) to see a results list of the new headlines and use the **Open URL** action on them. Typing → on the NetNewsWire.app itself brings up the New Headlines item too. Unfortunately as of B51 I find this unstable and it always crashes Quicksilver when I type →. As of B54 Quicksilver doesn't crash but headlines don't show up either.

Text

Most of the time typing in Quicksilver's first or third pane selects an object such as a file, contact, song, etc. Some actions like **Search For…** or **Rename...** take an argument in the third pane which is arbitrary text. This section is about entering arbitrary text in the first pane. Type a **.** (period) or ' (single quote) to enter text mode. In B51, some people with non-US or Qwerty keyboards report problems using one of these, but often the other works. This is a regular Cocoa text field so all the text editing key bindings work. E.g., ^D deletes a character, ^F moves forward a character and ^B moves back a character.

Formatting also works; pasting formatted text into the first pane preserves the formatting (fonts, colors, alignment) of the source. Furthermore, this stays in effect in future text panes until differently formatted text is pasted in or it's explicitly changed. The standard editing keys work though some only work if the Quicksilver menu is visible, which usually only happens when the Preferences window is open. Change fonts from the Font palette available via ⌘T when the menu is visible. Change colors with the color palette but this only comes up via the menu since the ⇧⌘C key binding is interpreted as the action associated with C, probably **Copy to Clipboard**. Changing alignment works even if the menu isn't visible; use ⌘{ to make the text aligned on the left side.

A common way to bring text into Quicksilver is to use the Send to Quicksilver Service or the predefined triggers Command Window in Text Mode or Command Window with Selection, commonly bound to ⌘-esc. Unfortunately Front Row started using that key binding and now Quicksilver doesn't assign a shortcut by default. See Combining Activation and Selection for more information. These mechanisms work with Cocoa applications, but note that Microsoft Office and Firefox are currently *not* Cocoa applications.

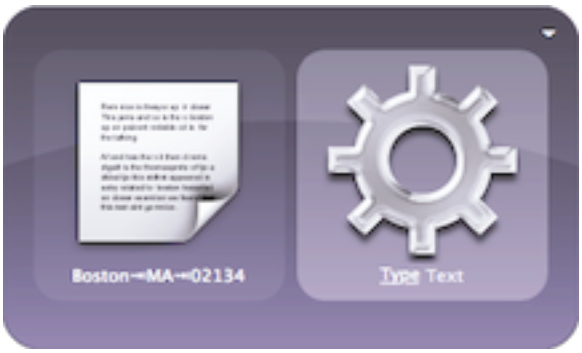
Basic Text Actions

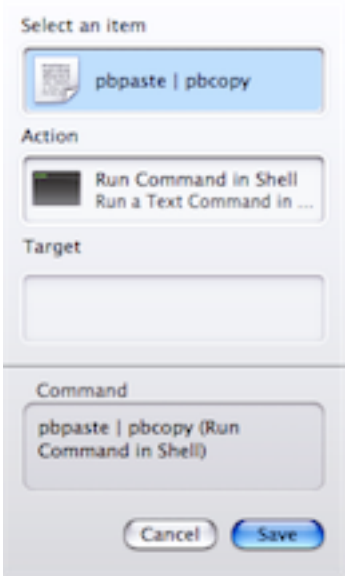
When dealing with text the two actions that come to mind first are **Copy to Clipboard** and **Paste**. However since ⌘C and ⌘V perform these actions in just about every application, I rarely use them in Quicksilver. The exception is when I look up something in Quicksilver like an address, phone number or filename and then **Paste** it somewhere. After finding the item in the first pane, just type . to change the selected item to text mode, then use the **Paste** action or any

other text action. E.g., look up a friend's contact info, tab and select their home address. Then type . and edit the address to just the zip code. Now **Paste** it into a weather web site, or better yet, use the **Find With…** action to call up a weather search on the zip code.

For commonly pasted text, like a signature for a Mail message or a long phrase, consider a trigger with the text and the **Paste** action. The Shelf as described in Clipboard & Shelf is also a good way to do this.

The **Type Text** action is similar to the **Paste** action, but will not put a copy on the clipboard. Also, more than just pasting the text, the application will treat the text as if the user had actually typed it, meaning it might perform commands based on the text. Use ⌥-return and ⌥-tab enter those special characters as text. E.g., the text `Boston→MA→02134` (those are tab characters) with the **Type Text** action completes three different fields in a web form all at once. Using the **Paste** action would put the whole thing, including the tab characters, into the first field.





Most copy and paste commands on the mac preserve the formatting of the source, but this isn't always wanted. A clever way around this is to use the command line tools `pbpaste` and `pbcopy` to take what's already on the clipboard and copy it again but without the formatting. These are used from a Terminal shell but the Terminal Module's **Run Command in Shell** action can be used instead; which means a trigger can be created for it. Create a trigger with "pbpaste | pbcopy" in the first pane and **Run Command in Shell** as the action. I have this bound to ^⇧⌘C. I also renamed the trigger to be Remove Copy Formatting.

In addition to **Run Command in Shell**, small scripts can be entered in the first pane and run in a new Terminal window with the **Run a Text Command in Terminal** action. This is an easy way to run the `top` command and if used often makes a good trigger. Enter multiple commands to run by separating them with a new newline entered with ⌥-return. For AppleScripts instead of shell scripts use the **Run as AppleScript** action. Use these with ⌘-esc to easily run scripts from email messages or posted on forums (but of course, use with caution).

The **Large Type** action will show the text in a giant window visible from across the room. Click anywhere to

close it. It's great when looking up a phone number. The more text there is to display, the smaller it gets. Some use it to view lyrics highlighted on a web page.

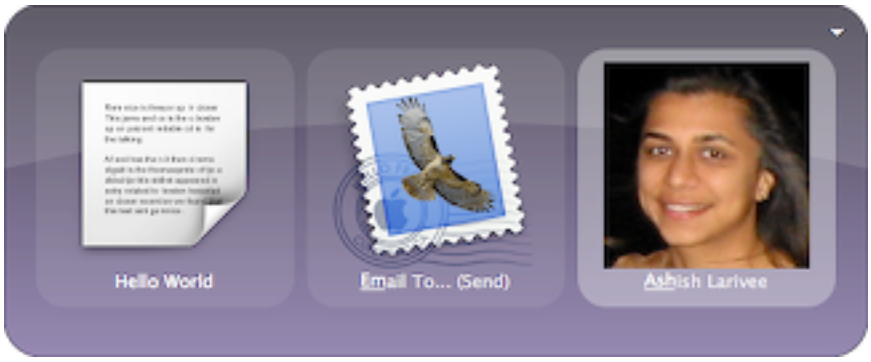
The **Display Dialog** action is similar to the Large Type action but it uses a standard dialog box with an OK button. The advantage is that it can be moved around and it will say visible until the OK button is clicked. It is disabled by default, enabled it in the Action preferences.

The **Speak Text (Say)** action uses OS X's speech synthesizer to speak the text out loud. It is disabled by default, enabled it in the Action preferences. The voice used is set in the Speech System Preferences in the Text to Speech tab. Try it with the **Run after Delay…** action to create a voice alarm. See the Command Objects and Droplets section for more details. I thought it might be useful for pronouncing foreign words but my attempts at Japanese and French with B51 didn't work very well.

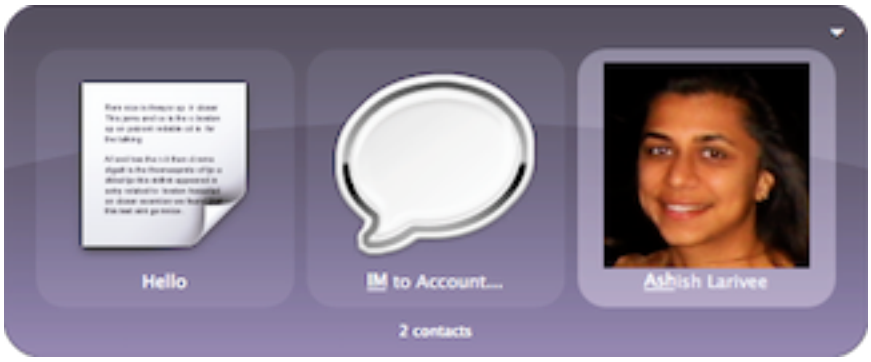
System Services are some of the more underutilized features of OS X. Applications register features that are available system-wide in every application's menu. Many services are only enabled while text is selected. The Services Menu Module plug-in makes all the system Services available in Quicksilver. E.g., Sticky Notes users will like the **Make New Sticky Note** action.

The Web Searches section describes using the Web Search Module plug-in. With text in the first pane, use the **Find With...** action and choose a saved web search in the third pane. The search results appear in the default browser. See the Web Searches section for configuration details and how to make a trigger with the Current Selection proxy object.

The Mail section describes various email actions from the Mail Support plug-in. The three **Email To…** actions (in **Compose**, **Send** and **Send Directly** variations) send text in the first pane to a contact or email address in the third pane. See the Mail section for details on configuration and the differences in the three variations.

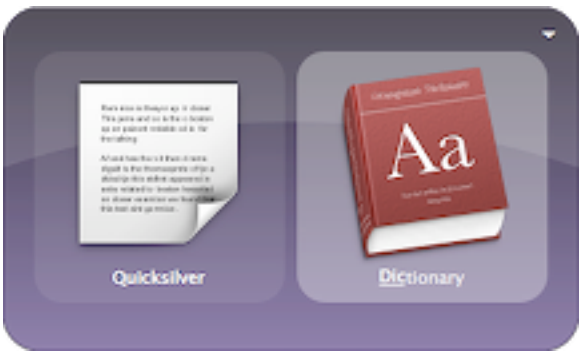


The Chat section describes the instant messager actions. **IM to Account...** sends the text in the first pane as an instant message to a contact or IM account in the third pane. The iChat plug-in supports setting the away message to text in the first pane with the **Set iChat Status** action.



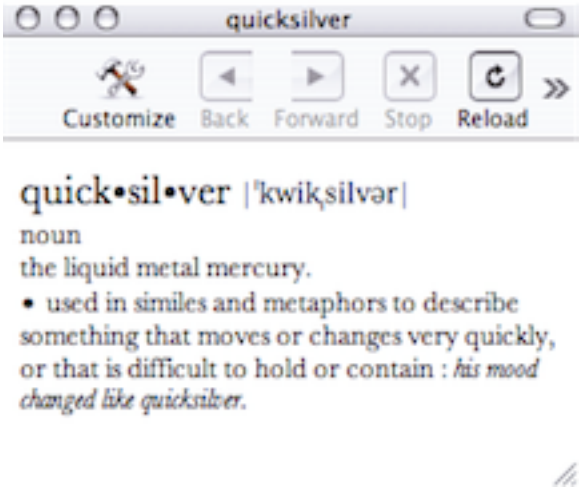
The Calendar section describes creating calendar events and to-dos in iCal and Google Calendar. These use specially formatted text to not only name the event or to-do but to set dates, times and priorities. **Create iCal Event** and **Create iCal To-Do** take a calendar as an argument in the third pane.

Dictionary



The Dictionary Module plug-in provides access to the OS X built-in dictionary. Enter a word in the object pane and use either the **Dictionary** or **Thesaurus** actions to access OS X's dictionary. Regardless of how the Dictionary preferences are configured, the **Dictionary** action shows the definition in a smaller panel (shown here).

The Services Menu Module plug-in makes **Look Up in Dictionary** available and is another way to do the above. **Look Up in Dictionary** opens the Dictionary application with the full window. Unfortunately there is a bug in 10.4 so that if the Dictionary application isn't already running, it's opened but the word to lookup is not entered automatically. It works fine if the Dictionary is already running and the **Dictionary** action seems unaffected by this bug. I have a trigger for the proxy object Current Selection with the **Look Up in Dictionary** action bound to ⇧^⌘D for those applications (e.g., Preview) that don't support the Cocoa key binding of ^⌘D.



The Dictionary Module also adds the **Define with dict.org** action which is similar but aggregates definitions from several web sites and displays them in a simple HTML viewer built-in to Quicksilver. It usually provides the most complete definition but the formatting isn't as nice.

Spotlight

When OS X 10.4 shipped with Spotlight there was a lot of confusion as to what it would mean for Quicksilver. As is probably clear by now the two are quite different. Quicksilver can't search inside files but can do much more than just open things. One of the biggest criticisms of Spotlight is that it starts searching immediately as the query is typed. Quicksilver solves that problem. The Spotlight Module plug-in installs three actions for text objects. **Spotlight in Window** runs the search in a Spotlight window. **Spotlight in Finder** runs it in a Finder window, like when editing a Finder smart folder. **Spotlight in Command** returns the results in an object pane results list in a new command window. None of these searches start until the command is executed. Spotlight saved searches, can be brought into the first pane and executed with the **Run Search with Spotlight** action.

Text Files

The **Create File...** action makes a new text file with the object text as the contents (not the name). It prompts for the new filename (and location) with a standard apple save dialog, saves the file, and then shows a new command window with the new file as the object.

The Text Manipulation Actions plug-in installs actions to manipulate the contents of plain text files (which must end in a .txt extension). These are great for managing a to-do.txt file. With text in the first pane, use the **Append To…** action and

specify a file in the third pane to add the text to the of the file. The **Prepend To…** action works the same way but adds the line to the beginning of the file.

The reverse commands are **Append Text...** and **Prepend Text...** which take the file in the first pane and the text in the third pane. Use these to create triggers to easily add things to a to-do.txt file.

I have a trigger defined for NextAction.txt (**Append Text...**). Whenever I think of stuff to do, I type ^⌘A and type the idea into the third pane and continue with what I was doing. Then I review the file regularly in my text editor. I have another trigger on ⇧^⌘A using the Current Selection proxy object that lets me select text in any Cocoa application and append it to my NextAction.txt file. It's defined as: Current Selection (**Append To...**) NextAction.txt. To use it for several files, leave the third pane blank and enter the file after invoking the trigger.

The Text Manipulation Actions plug-in also allows Quicksilver to access the individual lines of a text file by typing → into one. With a text file in the first pane, even though there is no > next to it in the results list, typing → or / will open a new results list with the lines of the file. Even though the lines have a > next to them, typing → won't do anything. With the command preference Show children split view enabled, the right-hand results list pane acts as a (slow) preview window showing the contents of a text file. Note that for .html files instead of each line being shown, instead each link is shown. See the Web Pages section for more details of .html files.

With text lines selected in the first pane, all the text actions are available and a few more. The ominously named **Delete Line (UNSTABLE)** will remove the line from the file. Yes, be careful about using it and it doesn't always work. Less risky is **Change To…** which takes new text in the third pane and will replace the text in the file with it. In addition, the **Prepend Text…** and **Append Text…** actions can work on text line references. Their argument is added as a new line before or after the line respectively. I had a problem with line-endings when using **Prepend Text…** on the first line in a file.

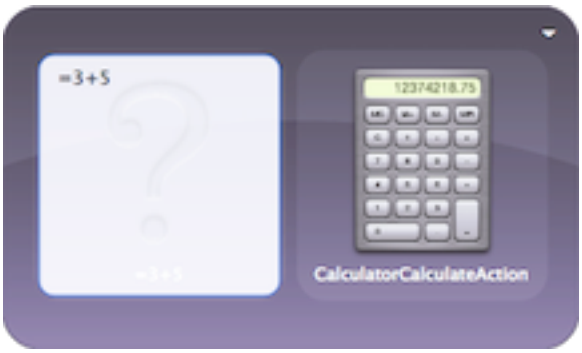
The BBEdit Module plug-in adds one action which works on text lines. Use **Select in BBEdit** to open the file in BBEdit on that line. It also adds a catalog source under Modules called BBEdit Recent Documents but as of B51 it doesn't seem to work.

Specially Formatted Text

The above actions work for all text, but Quicksilver recognizes some text forms and provides more actions for them and often chooses a special default action. E.g., if the text in the first pane looks like an email address (i.e., *text@text.text*, with at least one @ and on period) then the **Email item…** actions are available. If the text looks like an URL, then the **Open URL** action is available. Depending on how the actions are ranked in the Actions Preferences these might become the default actions. In these cases the background of the pane changes to include a faint version of the email or URL icons. These are the only two special forms that Quicksilver recognizes that I know of, but see below for how to perform math calculations.

Math Calculations

The Calculator Module plug-in adds two actions called **CalculatorCalculateAction** and a preference pane called Calculator. It allows math expressions to be entered as text in the object pane and Quicksilver will compute the result. By default it uses the unix utility bc to compute the expressions but this can be changed in the Calculator Preference pane to dc for an RPN calculator. bc understands the basic math expression including addition with +, subtraction with –, multiplication with `*`, and division with `/`. Parentheses specify the order the operations. There's much more it can do, to get the details look in the documentation (which for unix commands is a man page). In the Calculator Preference pane there is a button to bring up the man page in a Terminal window.

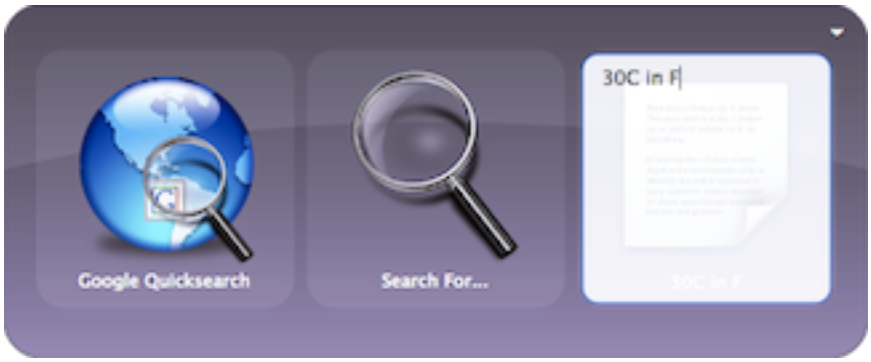


To perform a calculation, activate Quicksilver, type = to enter text mode, followed by an expression, e.g., `3+5`. Type `return` and the command window is replaced with a new one, with the result (in this case 8) in the first pane. Using = to enter text mode isn't required, but it tells Quicksilver to default the action to **CalculatorCalculateAction**.

The Calculator Preference pane allows configuration of the behavior when the result is computed. By opening a new command window with the result it's easy to choose an action like **Copy** or **Paste** to do with the result. But the default can be configured to display with the **Large Type** action or in a notification window like the iTunes module uses (or with Growl if it's configured).

The **Run as AppleScript** action can also perform basic math calculations with +, -, *, and /. In fact, since text mode is a Cocoa field, the Services menu in available. Script Editor has a service called Get Result of Apple Script and it's bound to ⌘* by default. So another way to evaluate simple expressions is to activate Quicksilver, type . to enter text mode, type some simple expression, type ⌘A to select it all and then type ⌘* to have the expression replaced with the result.

The Unit Conversion Module plug-in installs the action **Convert to Units…** which does conversions using the unix units program. In the first pane enter a number and in the third pane enter the units to convert to. E.g., `10 m` **Convert to Units…** `ft`. The result is shown in a new command window. The syntax is a little tricky so see the plug-in's documentation in the info pane or the units(1) man page for more details. In particular there must be a space between the number and the units.

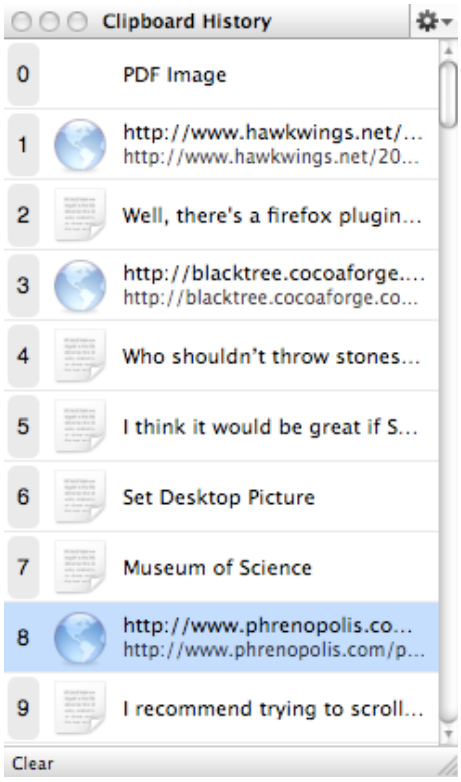


For more advanced calculations and conversions use Google's calculator. It can do temperature conversions, understands scientific notation and many many constants. Enter the expression in the first pane and use the **Search With Google** action (from the Services Menu Module plug-in and the Safari service) or create a trigger for Google Search (**Search For…**). Compared to the other methods above, this one requires a network connection and shows the results in a browser but can do a lot more. Syntax details are described at http://www.google.com/help/calculator.html.

Clipboard and Shelf

Clipboard History

For this section you'll need the Clipboard Module plug-in installed. It doesn't install any actions but does add several catalog objects in the Quicksilver set. Inside Shelf & Clipboard you'll find two items, Shelf, and Clipboard History. Make sure the latter is enabled for this section. (You may need to install the Shelf Module plug-in to see the Shelf & Clipboard catalog source and click refresh after checking it.) Under Internal Commands you'll find several related objects, some of which behave like actions but are scripts (such as Clip Store Copy 2) that you use with the **Run** action. Also under Proxy Objects there's a Clipboard Contents proxy object.



One limitation of copy and paste on the Mac is that the clipboard only remembers the last thing you copied. With Quicksilver's Clipboard Module you can extend this to remember the last several items you've copied. In the Clipboard Preference pane you can configure the number of items to be saved in the clipboard.

A large number like 100 is useful to be able to remember things you copied several days ago but it uses more memory. If you put large images on the clipboard consider a number a like 10.

You can bring up a window showing the Clipboard History by activating Quicksilver and typing the built-in HotKey ⌘L. You can also choose the Clipboard History object and the Show action. This action is actually a toggle and will close it if it's visible. If you use it often you can create a trigger. Note that some have reported that having the TextExtras Input Manager installed breaks the ⌘L HotKey.

If a clipboard item is text, the first few words are included in the list to help you identify the correct item. If the item is an image, the icon (usually) shows a thumbnail of it. Personally I use Clipboard History with the **Show Contents** action as a trigger on ^⌘C so that I get a new results list that I can use the matching algorithm to search through. However this doesn't work too well for image items, you have to scroll to see them.

⌘L is a toggle, so typing it again will make the window disappear. In the Clipboard preference pane you can enable the "Hide after pasting" option. If you position the window with one side on an edge of the screen then it will always disappear when idle (even if "Hide after pasting" is unchecked). You can then bring it back by moving the mouse to that same edge of the screen and it will appear again. This is similar to using the screen corners in the Dashboard & Expose System Preference.

For the first 10 items in the history you can paste them into the current application by typing the number of the item while the Clipboard History window is open. For other items you'll need to double-click on them to paste them.

Items remain in the Clipboard History until it fills up, in which case the oldest items are removed when new items are added (i.e., FIFO). The Clipboard History survives Quicksilver restarts and system reboots. The clear button at the bottom of the Clipboard History window will remove all but the most recent item from the history. You can remove a single item from the history by selecting it with single click and then typing the delete key. You can't delete the first (number 0) item. Instead copy something else onto the history to move it down and then delete it.

The Clipboard Contents proxy object represents the 0'th or most recent item in the history. Using it with the **Paste** action is the same as using Paste in whatever application you are using. It will include the formating of the text, use the **Type Text** action to paste without the original formating, matching the style of the destination. However you can use it with other actions such as **Open URL…**, **Look Up in Dictionary…**, **Email To…**, etc. I find it easier to use ⌘-esc to bring the selection into the object pane but if your item is already on the clipboard this could be useful.

There have been some reports of pasting from the clipboard not working or crashing Quicksilver B51 when a non-US keyboard is in use. To solve this, you don't have to switch to a US keyboard, but you do have to enable the "U.S." keyboard in the Input Menu tab of the International System Preferences. That is, the U.S. keyboard doesn't need to be in use, just enabled.

TODO: Show Clipboard Cache (pNew) and Show Clipboard Cache (pOld)

Clipboard Storage

Since cut and copy are such common operations on your Mac, items in the Clipboard History change frequently. If you want to save something for longer you can use an alternate clipboard called Clipboard Storage. The contents only last as long as Quicksilver is running. If you restart Quicksilver (or obviously if you logout or reboot) the Clipboard Storage will be lost. The Clipboard Storage is also installed with the Clipboard Module plug-in and is accessed via the same window. You can toggle the window between showing the Clipboard Storage and the Clipboard History by using the drop down in the top of the window. You can also show the Clipboard Storage by using the Show Clipboard Storage object with the **Run** action. That's something you may want to create a trigger for.

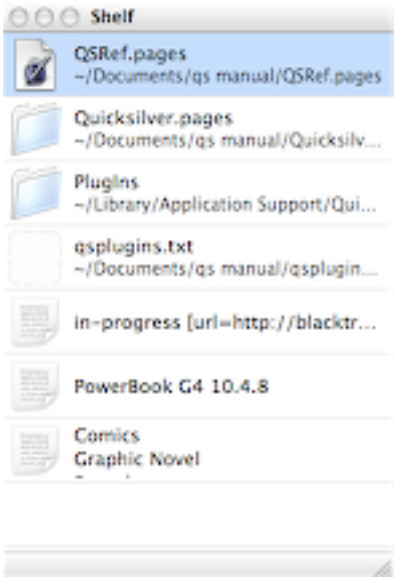
Unfortunately showing the Clipboard Storage seems to toggle the state of the window but there's no way to toggle it back to showing the Clipboard History without using the mouse to choose it from the drop down in the top right of the window. The command Show Clipboard, **Run** will just bring up the window in its last state. The command Clipboard History, **Show** behaves the same way. However the trigger I use, Clipboard History (**Show Contents)**, continues to show the history in the results list regardless of what the window is showing.

You use the Clipboard Storage the same way you use the Clipboard History. Double-clicking or typing a digit will paste that item into the current application. The Storage always has 10 numbered slots regardless of the number of history items you have configured. The only way to put an item into the Clipboard Storage is to drag it into a slot. You can use any slot regardless if higher slots are filled or not. Dragging an item into an already filled slot will replace the item in that slot. You can remove an item by selecting it and typing the delete key. The clear button will remove all items from the Storage, it doesn't leave the latest item as it does in History. There are objects of the form Clip Store Copy # and Clip Store Paste # which would seem very useful for frequent access to non-temporary items but they don't seem to work in B51.

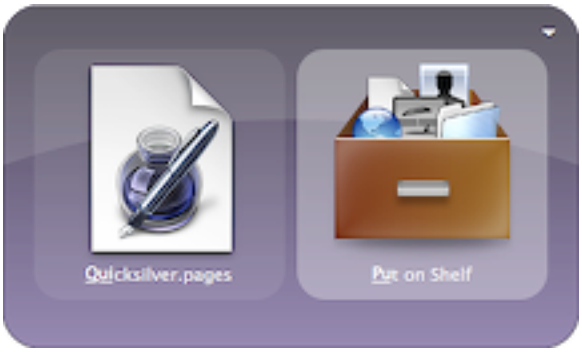
Shelf

For this section you'll need the Shelf Module plug-in installed. The Shelf is in the catalog under Quicksilver under Shelf & Clipboard. Make sure it's enabled (in B51 it's not by default) by selecting Shelf & Clipboard, typing the button to open the drawer, going to the Contents tab and checking the Shelf object. You might need to run the command Shelf & Clipboard (Catalog), **Rescan Catalog Entry**, after the first time you add something to shelf to have it scanned by Quicksilver. The plug-in installs two actions **Put on Shelf** and **Show**. The **Put on Shelf** action is not enabled by default, so find it in the Actions Preferences and check it.

The shelf is a place for long term storage frequently used items. It's like the Clipboard Storage described above but it lasts across Quicksilver sessions. People use it for: email addresses, images, code fragments and templates, favorite hex colors, lorem ipsum text, passwords (though this isn't secure), etc.. You can also put documents, folders and, applications on it and treat it like an alternate Dock.



When Quicksilver is activated, you can use the built-in HotKey ⌥⌘S to toggle the shelf. The Shelf object with the **Show** action will also toggle the shelf. If you use it frequently consider creating a trigger for this, it will work faster than ⌥⌘S since you won't first have to activate Quicksilver. Just like with the Clipboard, if you position the shelf window with one side on an edge of the screen then it will disappear when idle. You can then bring it back by moving the mouse to that same edge of the screen and it will appear again. This is similar to using the screen corners in the Dashboard & Expose System Preference.



The typical way to add something to the shelf is using the **Put on Shelf** action. It adds items to the top of the shelf as if it were a stack. You can also drag items to the shelf, into any contiguous location. The Current Selection proxy object with the **Put on Shelf** action makes a good trigger. You can also use ⌘V to paste things onto the shelf. If nothing in the shelf is selected the item will be pasted to the top, otherwise it's added above the selected shelf object.

You can bring up the shelf in the object pane and then type → to bring its contents into a results list. You can also drag and drop from the shelf with the mouse. Unlike the Clipboard, there are no numbers to select items with. While the shelf otherwise behaves as a stack, there's no easy way (such as a Pop From action or an analog for the Clipboard Contents proxy object) to get the first item into the object pane.

You can delete an item by selecting it with the mouse and then typing the delete key. Like on a stack, the lower items move up. There's no clear button to empty the shelf, but just select the top item and type `delete` repeatedly for each item on the shelf.

I use the shelf as follows. I have it on the edge of the screen so if I am using a mouse I can conveniently drag things to or from it. I have a trigger for the Shelf with the **Show Contents** action so I can quickly call up items using just the keyboard.

TODO: Right click shelf items to show a menu.

Comparing Clipboard History, Storage and Shelf

	History	Storage	Shelf
Window	Shared with Storage	Shared with History	Separate

	History	Storage	Shelf
Max Items	Configurable in Preferences	Unlimited?	Unlimited
Items Added	On top with every Cut or Copy	Anywhere	On top or above selected item with ⌘V
Items Accessed	Number keys, Results list, mouse	Number keys, mouse	Results list, mouse
Persistence	Survives system restarts but oldest items removed when full	Empties when Quicksilver restarts	Permanent until removed, survives restarts
Right Click Menu	No	No	Yes

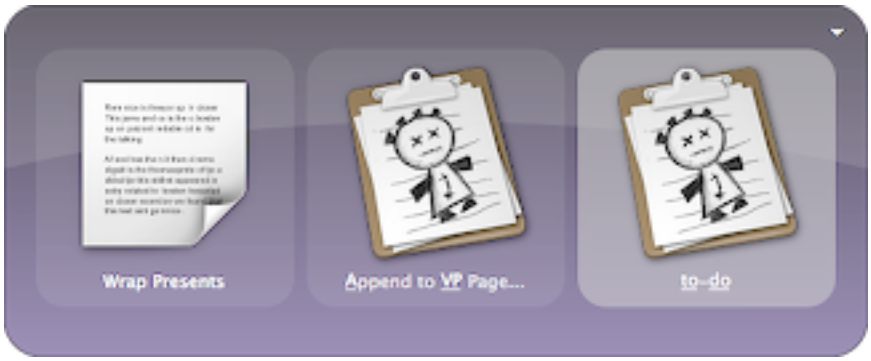
Note Taking Applications

VoodooPad

For this section you'll need the VoodooPad Module plug-in installed. It adds three actions to Quicksilver: **Open Page**, **Append to VP Page...**, **Prepend to VP Page…**. It also adds a Catalog source called VoodooPad Documents under Modules which is *not* enabled by default. You'll want to enable it by checking the box in the Catalog. It also adds a document scanner so if you add to the catalog a custom File & Folder Scanner source, you can choose VoodooPad Pages under Include Contents in the Source Options tab of the source. Unfortunately for B51 with VoodooPad 3.1.2 this didn't work for me.

VoodooPad comes in three editions: VoodooPad Lite, VoodooPad, and VoodooPad Pro. Unfortunately the Quicksilver plug-in will not work with VoodooPad Lite which is free. The plug-in needs AppleScript support and that's one of the features missing from the Lite edition. The plug-in as downloaded will work with VoodooPad Pro which is the most expensive and full-featured edition. The plug-in can be made to work with the plain VoodooPad edition by editing the script inside. Find the VoodooPad Module.qsplugin in the ~/Library/Application Support/Quicksilver/PlugIns/ directory in the Finder. Control-click on it and choose Show Package Contents to bring up a new Finder window, open the Contents folder and then the Resources folder. Open the QSVoodooPadPluginAS.scpt in Script Editor (you'll need Voodoopad Pro.app for Script Editor to be able to open it) or your favorite text editor and change all the occurrences of "VoodooPad Pro" to "VoodooPad". Save it and then it should work.

You can bring up VoodooPad.app (or the Pro edition) and type → and you'll see a list of VoodooPad documents (.vdoc files), but you can't → into them. You can just type the name of VoodooPad page in one of those documents into Quicksilver's first pane because of the catalog source. With a page in the first pane, you can use the **Open Page** action to bring up the page in VoodooPad. You can also add text in the first pane to the end of a page with the **Append to VP Page…** action and a page in the third pane. The **Prepend to VP Page…** action is similar but adds the text to the top of the page. You might want to configure a trigger for something like Current Selection (**Append to VP Page…**) To-Do to easily add selected text (via the proxy object) to a To-Do page in VoodooPad.



VoodooPad adds two services: Append to Services Drop Page and Make New Page, both of which become actions you can use on any text with the Services Menu Module plug-in.

Yojimbo

For this section you'll need to install the Yojimbo Module Plug-in. It adds four actions to Quicksilver, **Add to Yojimbo**, **Archive to Yojimbo**, **Show in Yojimbo**, and **Append Text (Makes Plaintext)**. The first three are enabled by default, but the last one is disabled. To use it, you'll need to enable it in the Action Preferences. It also adds a Catalog source under Modules called Yojimbo Items, which finds entries in your Yojimbo database.

Since the Yojimbo Items catalog source is enabled by default, you should be able to bring up any Yojimbo item by just activating Quicksilver and typing its name until it appears. You can also bring up the Yojimbo.app and type → to see a results list of all your Yojimbo items. Given a Yojimbo item in the first pane, you can use the **Show in Yojimbo** action to open Yojimbo to that item. If you enable the action, you can also add to a Yojimbo item by using the **Append Text (Makes Plaintext)** action with a text argument in the third pane.

You can add text notes to Yojimbo by using the **Add to Yojimbo** action with a text object in the first pane. You can add URLs to Yojimbo in two ways. With an URL in the first pane, the **Add to Yojimbo** action will add it to Yojimbo's Bookmarks and the **Archive to Yojimbo** action will add it to Yojimbo's Archives.

WikityWidget

WikityWidget is a Dashboard widget that behaves a like a small wiki. Think of Sticky Notes on steroids. If you use this you'll want to install the WikitWidget Module plug-in. Be sure to install WikityWidget before you install the plug-in, or else you might have problems with the QSWikit-Plugin catalog source that is installed in Modules. The plug-in also adds four actions: **search Wikit**, **delete Wikit**, **append to Wikit**, and **prepend To Wikit**.

WikityWidget uses the word Wikit to mean a page. With text in the first pane, you can add it to the beginning or end of a wikit page with the **prepend to Wikit**, and **append To Wikit** actions respectively. Both take a wikit page as the argument in the third pane. You can also search for a wikit with some text using the **search Wikit** action with the text to find in the first pane. It returns a results list of wikits matching the search string.

Unfortunately there's no action to open a wikit specified in the first pane (e.g., as the result of the **search Wikit** action) in the widget. There are three actions that you can use with a wikit in the first pane. You can delete the page with the **delete Wikit** action. You can also use the **prepend to Wikit**, and **append To Wikit** actions with text in the third pane. Yes these actions work with their arguments reversed.

Images

iPhoto

The iPhoto Module plug-in adds two actions and enables you to → into iPhoto.app to see everything in iPhoto's left pane: the library, rolls, folders, albums, smart albums and slideshows. The **Show Album** action will open the selected item in iPhoto. The **iPhoto Slideshow** action will use iPhoto to play the slideshow (as opposed to the Finder's slideshow ability). If you use it on an album or other non-slideshow object, it will create a new slideshow but not save it. It's the equivalent of using the Play button in iPhoto as opposed to the +Slideshow button. Both actions will start iPhoto if it's not running, but note that you can navigate through your iPhoto collection without starting iPhoto.

You can also use → to navigate down to an individual image. You can use actions such as **Email to…** or **Set Desktop Picture** actions among others. Note that the **Set Comment…** action (from the File Attribute Action plug-in) can be used to set the Spotlight comment of images, but this is different than the comment you can set for a image in iPhoto.

There are also two proxy objects, Current iPhoto Selection and Selected iPhoto Album which are useful for creating triggers.

Slideshow

For this section you'll need the Slideshow Action plug-in. It installs two different actions called **Slideshow** that make use of the new slideshow feature added to the Finder in 10.4. One takes an image file or folder as the object, the other works on iPhoto albums and photos (which you'll need the iPhoto Module plug-in to access). As in any Finder slideshow typing `esc` will cancel the slideshow and moving the mouse will bring up some controls. Note this is different than the iPhoto Slideshow action which uses iPhoto slideshow capabilities (notably music and the now ubiquitous Ken Burns effect).

Desktop Picture Action

The Desktop Picture Action plug-in adds a **Set Desktop Picture** action that can act on folders or individual image files (GIF, JPEG, PICT, PNG, or TIFF). For folders, the desktop picture will be chosen from images in that folder and rotated. The time interval for updating and whether the images will be chosen sequentially or randomly are taken from the last-used settings in the Desktop & Screen Saver Preference Pane. If you have multiple displays connected to the computer, the action takes an optional argument which lets you select a display from a list. By default, this action applies to the main display.

I think it would be great if **Set Desktop Picture** could work on a URL to an image, but it can't. But if it could, then I could have a trigger of Current Web Page proxy object, **Set Desktop Picture** when viewing a nice image.

Screen Capture

The Screen Capture Module plug-in installs three Internal Commands that use Grab.app for screen captures. You'll find them in the Catalog, under Quicksilver, under Internal Commands (make sure this source is enabled). You'll need to have Advanced features enabled for them to work. They are called Capture Screen, Capture Region, and Capture Window and you use them with the **Run** action. You could just use the standard OS X key bindings for these. ⇧⌘3 for Capture Screen, ⇧⌘4 for Capture Region and ⇧⌘4 followed by space for Capture Window. The one advantage to the Quicksilver commands is that after they run, a new command window appears with the newly generated image selected as the object, ready for you to rename it or move it or do whatever you want.

You could also get at these functions with the Services Module plug-in which makes things in the application's Services menu into actions. This creates the actions **Grab/Screen**, **Grab/Selection**, and **Grab/Timed Screen** which seem to work with any object. These also open a new command window but in B51 it doesn't seem to work correctly. The object is listed as Unknown Clipboard Object and trying to **Paste** it didn't work.

Image Manipulation

For this section you'll need the Image Manipulation Actions plug-in installed. It only adds two actions but if you work with images you'll really enjoy them. Both work with an object that is an image file and create a new image in the same folder as the original. Note the new image does not include metadata information such as aperture and camera model of the original. Also both actions when finished open a new command window with the new image in the object pane ready to specify a new action, like **Open** to see the new image.

The first action **Save Image in Format…** changes the format of the image. You specify the desired format as an argument in the third pane via text mode. You can specify "jpg", "png", "gif" or

“tiff”. Quicksilver is forgiving with these allowing you to say “jpeg” or “tif” as well. In addition you can give resolution levels as “low”, “med”, or “hi” and either “progressive” or “interlaced”. Not all of these work with all formats, you’ll have to experiment. So, for example, you can specify `jpg high progressive`, or `png low`, or `tif`.

The second action **Scale Image…** changes the size of the image. You specify the scaling factor in the third pane via text mode. You can give a percentage ofthe original’s size (e.g., `50%` or `200%`) or dimensions in pixels for the new image as *width* `x` *height*. White space is optional as is either number (the other dimension will be computed maintaining the original’s aspect ratio). If you want to specify just the height, precede it with an `x`. E.g., `300 x 400`, `150`, or `x500`. You can also specify a new format for the image by ending the specifier with the word `as` and then using any argument valid for **Save Image in Format…**. E.g., `640x480 as jpg high progressive`. If you often convert images to some standard size or format (like to post a blog or include in a manual), consider a trigger such as File Selection (**Scale Image…**) 400.

This isn’t going to replace iPhoto or Photoshop but for some quick changes (e.g., when formatting screenshots for a User Manual) these actions work very well. They work even better when you realize the comma trick works and you can operate on more than on image at a time.

Flickr Upload

Now that you have your images in the size and format you want, maybe you want to upload them to your flickr account. Install the Flickr Upload plug-in to get the **Upload to flickr (with Tags)…** action. Select the image you want as the object. If you want you can `tab` to the third pane to enter flickr tags for the image in text mode. Type `return` to start the upload. Quicksilver will connect to your flickr account either making use of your browser’s cookies or opening the browser to the flickr login page. And yes, you can upload more than one file with just one command using the comma trick.

System

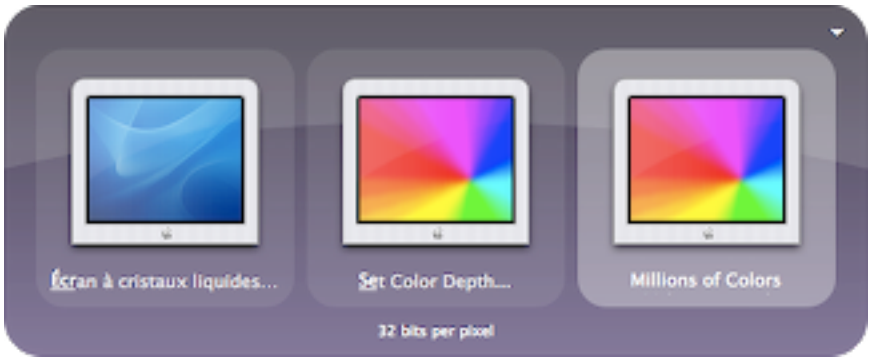
Quicksilver deals not just with your data, but with your computer too. Here are the ways it can interact with your hardware and its configuration.

Devices

In the catalog under System and Devices you’ll find Disks and Printers. If you install the Displays Module plug-in you’ll also see Displays. If you install the AirPort Module plug-in you’ll also see AirPort Networks.

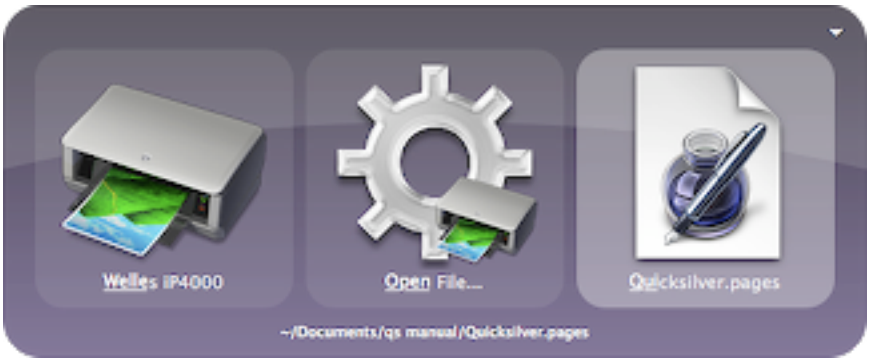
Installed Disks can be called up in the object pane by name. You can also hold down the / (slash) key to select the root drive. In addition there is an object called Network in the catalog for `/Network` on your system which without Quicksilver is typically opened from the top of the Finder's sidebar. If you mount network volumes (typically using the Finder's Connect to Server command) they will also appear, as well as two items for the automounter: Servers for `/automount/Servers` and static for `/automount/static`. There are also three proxy objects related to disks: Network Disks, Mounted Disks, and Removable Disks There are no special actions for these objects, you merely navigate through them as you do any other folder. There doesn't seem to be a way to mount network drives but you can **Eject** them, in fact Network Disks (**Eject**) makes a nice trigger.

With the Displays Module installed the Catalog also includes all displays connected to the computer. It is a little flaky though. I have a PowerBook running in an English locale and yet in version 48 of the plug-in my display appears in the catalog as "Écran à cristaux liquides couleur" which is French for "LCD Screen". Still I can select it and then use one of the actions: **Set Refresh Rate…**, **Set Color Depth…**, or **Set Resolution…**. All three let you choose an argument from a list of available options and also have some very pretty icons.



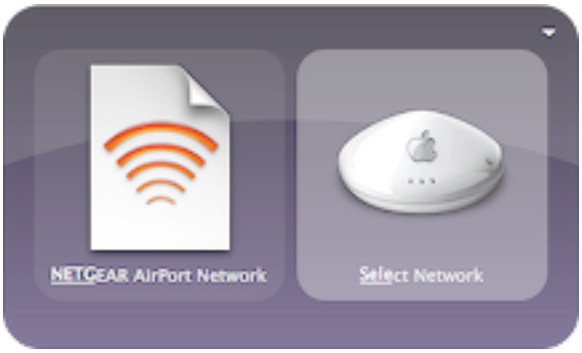
The Displays Module also installs a **Find With…** action but it doesn't do anything and shouldn't be used.

Quicksilver scans the folder `~/Library/Printers/` to find all installed printers (they end with `.app`) and add them to the catalog. You can print a file by selecting it as the object, using the **Open With…** action and then choosing a printer as the argument. You can't choose the printer directly in the third pane, you have to navigate to `~/Library/Printers/` to select it. A little easier is the reverse action **Open File…**. Choose the printer as the object, which you can do just by typing it and use the **Open File…** action and then choose the file as the argument. Since the first two panes are fixed, this form is better as a trigger. Select your primary printer, the **Open File…** action and leave the third pane blank, to be filled in when you use the trigger, perhaps by using ⌘G to choose the Finder's selection. Note that selecting a file and using **Move To…** or **Copy To…** a printer will not print the file but will put it inside the printer's .app package.



Networking

With the AirPort Module plug-in installed you'll see any available wireless networks as objects. There is one action installed with the plug-in. For a wireless network object you can choose Select Network to join that network (provided the configuration allows). If you need to switch to another network, this is a very convenient way. If you manual switch networks often, a useful trigger is AirPort Networks (Catalog) with the **Show Contents** action. You can then choose the network and use the **Select Network** action.



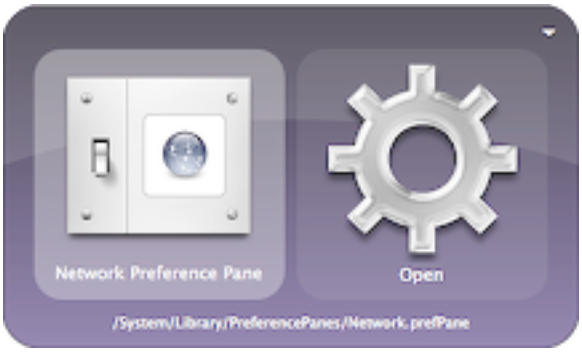
In the Catalog under Quicksilver, Internal Commands are the Turn AirPort On and Turn AirPort Off objects. Use the **Run** action with them.

With the Network Location Module plug-in the locations you've configured in the Network System Preference pane are available in the Catalog (under System, Configuration, Network Locations). If you select one in the object pane you can use the **Switch to Location** action to use it. If you select the Network Locations (Catalog) object you can type → to get a results list of all configured network locations.

There are several OS X utilities that come in handy when dealing with networking that are easily accessed with Quicksilver. The Applications catalog source includes indexing in /Applications down 3 levels so by default you can get to AirPort Admin Utility, AirPort Setup Assistant and Internet Connect. The default action for any of these applications should be **Open**. The Network Preference Pane is available in the catalog under System, Configuration, Preference Panes, System Preferences (System).

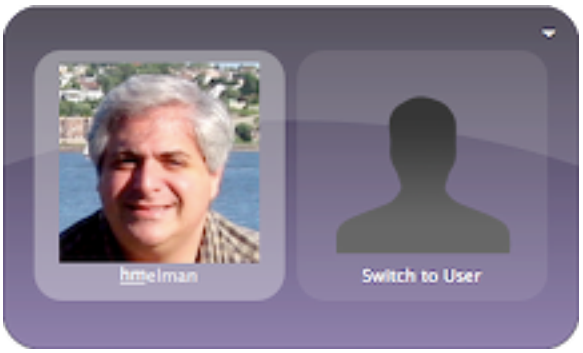
Configuration

Quicksilver includes the OS X System Preferences in the Catalog under System, Configuration. It will find preference panes in `/System/Library/PreferencePanes/`, `/Library/PreferencePanes/` and `~/Library/PreferencePanes/` and you can include or exclude these in the Catalog as groups if you want. The individual preferences panes are in the catalog, so you can bring them up directly. You can also select the Preference Panes (Catalog) object or the System Preferences.app as the object and type → to see a list of all preference panes in a results list. With a preference pane as the object, you'll most likely want to use the **Open** action to bring up the preference pane, but they are treated like files and many other actions are available.



With the User Accounts Module plug-in installed

the user accounts on your system are in the Catalog (under System, Configuration, User Accounts) and there is a **Switch to User** action. This invokes OS X's fast user switching technology. If you select User Accounts(Catalog) object and type → you get a results list of all user accounts on the system. There's also a Fast Logout.sh object installed with the Extra Scripts plug-in. If you use the **Run [...]** action from the Terminal Module plug-in with it you will use fast user logout which basically just locks the screen but leaves everything in your session running as if you had done fast user switching. If you do this often create a trigger for it.



With the Keychain Module plug-in installed your Keychains are available in the Catalog. They appear in the Catalog under Modules, Keychains. If you select one of your keychains as the object, you can type → to see the individual items in a results list. With a keychain item as the object you have three primary actions you'll want to use: **Copy Password**, **Paste Password**, and **Get Password**. All three will open a Confirm Access to Keychain prompt, I always choose Allow Once. The first will copy the password to the clipboard, the second will paste it directly into the current application and the third will open a new command window with the password as the text object so you can choose any action you want. Accessing passwords using these actions should be more secure than keeping them as text in triggers, the shelf or clipboard, though if you use **Copy Password** you're copying them to the clipboard anyway. **Paste Password** (unlike **Paste**) does not put a copy on the clipboard. If you use **Get Password** and then **Type Text** (instead of **Paste**) you will also avoid making copies on the clipboard.

Exposé

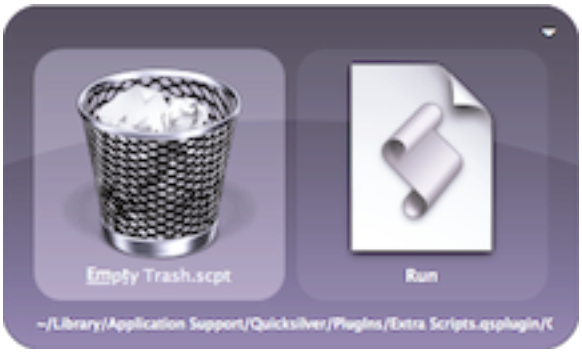
The System Hotkey Commands plug-in creates Catalog entries for Dashboard and Exposé commands. For those commands configured on function keys (F1-F13 with or without modifiers) in the Dashboard and Exposé System Preference pane, you'll find Catalog entries in Modules, Exposé Commands as shown in the following table.

Exposé Command	Quicksilver Catalog Item
All windows	Exposé All Windows
Application Windows	Exposé Application Windows
Desktop	Exposé Desktop
Dashboard	Show Dashboard

While there are other actions available I can't imagine wanting to use anything other than **Run**. If you are annoyed that OS X doesn't allow you to bind these to more keys, configure a trigger to access these any way you want.

Extra Scripts

The Extra Scripts plug-in is really just a catch-all of various scripts, but most have to do with controlling the computer so they are described here. Depending on the script you'll use these with the **Run** action for applescripts or one of the Run actions from the Terminal module, The default action should be correct. If you use these often, they all make good candidates for triggers. For some of them like Shut Down, if you make a HotKey trigger for them, you might want to set a Delay on the trigger so you can't accidentally type it.



System
- Sleep - same as ⌥⌘⏏
- Restart - same as ^⌘⏏
- Force Restart - same as ^⌘-`power`, warning: doesn't offer to save files
- Shut Down - same as ^⌥⌘⏏
- Force Shutdown
- Empty Trash
- Eject
- Close Disk Tray

System Volume:
- Mute Volume - sets volume to 0, not a toggle, don't confuse with iTunes Module "Mute" script
- Min Volume (20%) - sets volume to 1
- Mid Volume (40%) - sets volume to 4
- Max Volume (60%) - sets volume to 7
- Toggle Audio Input - toggles between input devices in the Sound System Preference
- Toggle Audio Output - toggles between output devices in the Sound System Preference

Networking:
- Get IP - shows the result in Large Type
- Get External IP - shows the result in Large Type

Processes:
- Hide Others - most OS X applications have this as a shortcut on ⌥⌘H
- Show All
- Quit Visible Apps
- LockScreen - runs the screen saver and makes you type a password to get back
- Logout - same as ⇧⌘Q (or is it ⌥⇧⌘Q?)
- Fast Logout - leaves you logged in, but goes to a login window (OSX's Fast User Switching)
- Switch To Root
- Classic Shutdown
- top 10 - use **Run in Terminal…** as it outputs 10 lines of text for the 10 busiest applications

Miscellaneous:
- Show Character Palette - Often ⌥⌘T but not in all applications
- Show Keyboard Viewer

- Sync Now - runs synchronize in iSync
- Type Clipboard
- Zoom Front Window - This is like clicking the green button in the top left of a window

Development

Automator

If you use the Automator application added in 10.4, you'll want to enable Advanced features and install the Automator Module plug-in. It adds the **Execute Workflow** action, to use it, navigate to a saved Automator .workflow file and the **Execute Workflow** action should be the default.

Services

The Services Menu Module plug-in makes all your Services available in Quicksilver as actions. Services are in the application's menu and many are only enabled when you have selected text. Depending on the applications you use, some services can be really useful as actions. E.g., if you use Sticky Notes then you'll love the **Make New Sticky Note** action.

Terminal

For this section you'll need the Terminal Module plug-in installed. There's a Command Line Interface Handler that should be set to Terminal. The plug-in creates two catalog sources under Modules. Terminal Files looks in ~/Library/Application Support/Terminal/ for any .term (containing Terminal settings) or .command files (containing Terminal commands). Bash Command History parses the ~/.bash_history file making each line available as an object in the catalog. Note that bash keeps duplicate lines in the history and they appear in Quicksilver's list as well.

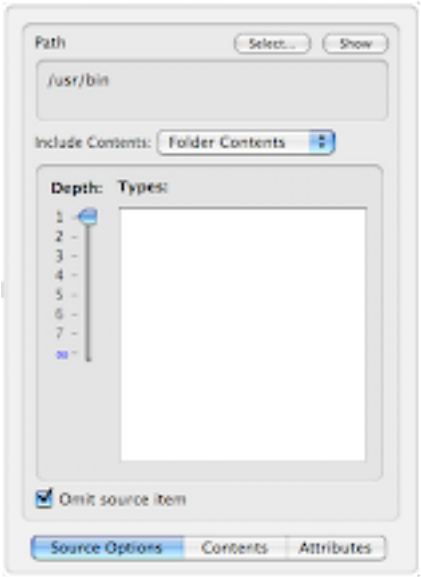
The **Run [...]** action will run a command or shell script selected in the first pane with optional arguments entered in text-mode in the third pane. The result of the command (i.e., stdout) is displayed in a new command window. The **Run in Terminal [...]** is similar but opens a new window in Terminal to run the command and display the results.

As an aside, Applescripts can be run by choosing the script in the first pane, using the Process Text... action and entering text in the third pane. This can be thought of as a reverse action for any script actions added to the ~/Library/Application Support/Quicksilver/Actions/ folder (see Extending Quicksilver).

If you want to type a command directly into Quicksilver and run it you can use the **Run Command in Shell** action or the **Run a Text command in Terminal** action. The first runs the command in a shell from Quicksilver and displays the result in a new Quicksilver command window, the second does so in a new Terminal window. You can also use these actions with a command from the Bash Command History catalog source. If you look through your history often you might want a trigger for Bash Command History (Catalog) (**Show Contents**). These commands can be handy when the Dock gets confused. E.g., you can do `killall Dock` with the **Run Command in Shell** action to kill the Dock and have it automatically restarted.

The **Go To Directory in Terminal** action will open a Terminal window with the current working directory set to the directory specified in the first pane. You can use text mode to type the name of any directory, e.g., /System/Library/Fonts, though if you use ~ the **Go To Directory in Terminal** action doesn't appear. You can also navigate to a folder in Quicksilver and use the action. If the folder is in the Catalog it should be easy to bring up quickly. If you have a project with a lot of directories you can add them to the catalog. To keep the number of items small you might want to specify just folders and no the files (e.g., this might be useful with a large software project) by putting `'fold'` in the Types box in the Source Options for the catalog source.

The **Show Man Page** action will show the man page of a command in a Terminal window. You don't enter the desired man page in text-mode or navigate inside /usr/share/man/ to a page. Instead it only works on commands (without an extension) which are in the catalog. To set this up open the catalog and select Custom. In the + menu choose a new File & Folder Scanner and select `/usr/bin/` as the directory. A depth of 1 is fine. Now all those command line utilities are directly accessible from Quicksilver and the **Show Man Page** action will work with them.



An alternative is to make use of the url scheme x-man-path. Opening an URL such as `x-man-page://units` will show the man page in a terminal window. It can also show different sections using this form: `x-man-page://3/printf`. You can create a web-search for `x-man-page://***` to make this more convenient.

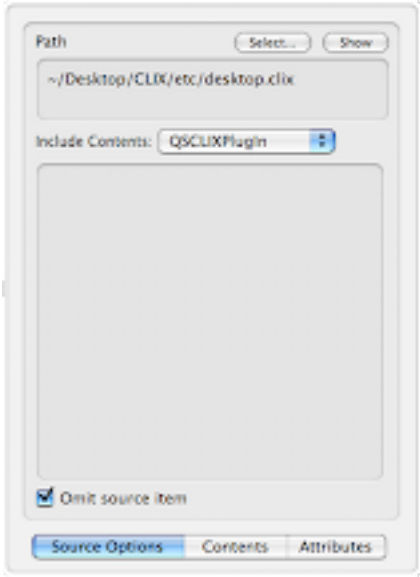
A common issue seems to be that Terminal automatically runs the same command at startup. This a Terminal configuration issue not a Quicksilver one. It happens if you invoke Terminal for a specific command and then open the Window Settings… panel and click Use Settings As Defaults. The command used to invoke Terminal gets saved in ~/Library/Preferences/com.apple.Terminal.plist. To remove it, open run this command: `defaults write com.apple.Terminal ExecutionString`. If you can't access Terminal try another program like iTerm or running it from Quicksilver with **Run Command in Shell**, or edit the plist file directly.

If you prefer iTerm to Terminal, install the iTerm Module plug-in and set the Command Line Interface handler to iTerm. The **Run [...]** and **Run Command in Shell** actions don't change they still run the commands in Quicksilver. By changing the handler the **Run in Terminal [...]**, **Run a Text command in Terminal**, and **Go To Directory in Terminal** actions change to use iTerm. There are also three new explicit actions called **Run in iTerm [...]**, **Run a Text Command in iTerm**, and **Go To Directory in iTerm**. So if you even if you have the handler set to one, you can run commands using the other easily.

The **Show Man Page** action is a little different. Setting the handler doesn't specify which terminal program is used. Instead both the Terminal and iTerm modules define their own versions of this action. Since the name of the terminal used isn't in the name you have to tell them apart from their icons. If you prefer one to the other but still have both plug-ins installed, just drag and trop the actions in the actions preferences so that your preferred one has priority. There's also a small difference between the two versions of this action. The Terminal one exits the terminal when you quit viewing the man page, the iTerm one does not.

CLIX

If you use CLIX from Rixstep to perform command line operations, you'll want the Quicksilver CLIX Module plug-in. It lets you configure a custom catalog source for a .clix file and converts all the commands inside it into text objects that you can run using **Run Command in Shell** or **Run a Text Command in Terminal** actions. To configure it, open the Catalog and choose the Custom set. Click on the + button at the bottom to add a new custom source. Choose File & Folder Scanner from the pop-up menu. Enter the path to the .clix file, you must have a separate source for each .clix file. The trick to make this work is to go to the Source Options tab and for Include Contents choose QSCLiXPlugin. At the bottom you probably want to check Omit source item. Rescan the source and you should see the CLIX commands in the Catalog. Activate Quicksilver, select one of the commands and choose one of the available Run actions such as **Run Command in Shell**.



Command Line Tool

If you use the command line you may find that you'll want it to be able to interact with Quicksilver. Install the Command Line Tool plug-in and a Quicksilver preference pane is installed under General called Command Line Tool. It contains an install button to install a `qs` program in `/usr/bin/`.

Invoke `qs` with one or more files as arguments and Quicksilver will be activated with those files selected in the object pane. With an argument of `-` it will read from stdin though piping in filenames results in the file names being in the object pane as *text*, which is not the the same as having the files selected. If you use the `-s` or `--shelf` option the files or text will be put on the shelf.

If you are an emacs user, here's an emacs command to pass the current file (of a buffer, in dired, or buffer-menu mode) to Quicksilver using `qs`. You could bind it to "`C-c q`" using something like: `(global-set-key "\C-cq" 'buffer-file-to-quicksilver)`

```
(defun buffer-file-to-quicksilver ()
  "Opens the current file in Quicksilver"
  (interactive)
  (cond ((and buffer-file-name (file-exists-p buffer-file-name))
         (call-process-shell-command (concat "qs \"" buffer-file-name "\"")))
        ;; dired handling
        ((eq major-mode 'dired-mode)
         (dired-do-shell-command "qs * "
                                 current-prefix-arg
                                 (dired-get-marked-files t current-prefix-
arg)))
```

```
;; buffer-menu mode
((and (eq major-mode 'Buffer-menu-mode)
      (file-exists-p (buffer-file-name (Buffer-menu-buffer nil))))
 (call-process-shell-command
  (concat "qs \"" (buffer-file-name (Buffer-menu-buffer nil)) "\"")))
(t
 (error "Not visiting a file or file doesn't exist")))
```

Here's another function. It will send the contents of the Emacs region to Quicksilver as text. With the text in the first pane you can do whatever Quicksilver can do with it, e.g., the **Find With…** action or the **Email to…** actions. I bind it to "`C-c w`" using: `(global-set-key "\C-cw" 'region-to-quicksilver)`

```
(defun region-to-quicksilver (start end)
  "Opens the contents of the region in Quicksilver as text."
  (interactive "r")
  (call-process-region start end "qs" nil 0 nil "-"))
```

Developer Documentation

If you've installed the OS X Developer Tools then you probably want to install Quicksilver's Developer Module plug-in. This adds to the catalog under Modules, Developer:

- applications under /Developer/Applications
- documents under /Developer/ADC Reference Library/documentation/
- frameworks under /System/Library/Frameworks and /System/Library/PrivateFrameworks
- headers for AppKit, Foundation and CoreFoundation frameworks.

Subversion

The Subversion Module plug-in adds the following actions. For all of them you give the directory or file you want to work on as the object in the first pane. You can use the comma trick to have one command operate on multiple files. If you add your project to the catalog, it will make it easier to bring up the files to operate on. Those that take an argument in the third pane are noted.

- **SVN Add All Directory Contents**
- **SVN Add Item**
- **SVN Check Out To Directory** - enter the subversion URL in the third pane
- **SVN Commit** - enter comment for log in the third pane
- **SVN Delete Item**
- **SVN Log**
- **SVN Status**
- **SVN Update**

The **SVN Add All Directory Contents** takes a directory as the object and adds all the files and directories in the directory to the subversion repository. It can take a while to run but progress displays in the Quicksilver Task Viewer.

The subversion plug-in calls the `svn` command to perform the actions, so obviously you must have subversion installed on the system. The plug-in expects the `svn` command to be in `/usr/local/bin/svn` if it's some place else you can run the following command in a terminal shell to tell Quicksilver where to find it, just replace *path* with the full path of the `svn` command:

```
defaults write com.blacktree.Quicksilver QSSVNPluginSVNPATH path
```

By default you will not be shown a list of changed files on commit. If you would like to see such a list use the following command in a terminal shell:

```
defaults write com.blacktree.Quicksilver QSSVNShowChangedFiles 1
```

Change the `1` to a `0` to revert back to the default behavior. To always show a text window with the results of the command issue the following command in a terminal shell (change the `1` to 0 to resume the default behavior).

```
defaults write com.blacktree.Quicksilver QSSVNAllWaysShowTextWindow 1
```

Eclipse

The beta Eclipse plug-in indexes workspaces and logs into the catalog. Use the **Open** action with them. Note, installing this plug-in causes a very slow search of the user's entire directory looking for Eclipse workspaces. See documentation at: http://code.google.com/p/qseclipse/w/list

Remote Desktop

Part IV - Appendix

FAQ

Quicksilver uses lots of memory, cpu or hangs:
Quicksilver keeps the entire index in memory and scans the catalog sources regularly. Usually when it uses too many resources it's because the catalog is configured too large. The default ~/Documents/ source is configured to a depth of 2, if you create a custom source of infinite depth it will make the catalog large. Also if you configure a source for your home directory (~) it will probably scan many unnecessary files (~/Library, etc.). This is also why your music and pictures aren't in the global catalog and should be accessed via the iTunes and iPhoto plug-ins which require you to type → to get into your music files and photos.

Also if you have network drives configured to be scanned that can make the catalog large or take a while. Note the Find All Applications catalog source (in the Applications set) will scan your entire machine, including network drives for applications. A comfortable size for your catalog will vary based on the capabilities of your computer, I try to keep my catalog below 5000 items.

Also, if you have the Clipboard Module plug-in installed and configured in the Clipboard Preferences to Capture History you might have large clippings in the history. This would cause Quicksilver to use lots of memory and perhaps lots of CPU at startup. The clipboard history is stored in the following file:

~/Library/Application Support/Quicksilver/Shelves/QSPasteboardHistory.qsshelf

you can check it's size. I have my clipboard history set to 100 and the file is 512KB. To clear it you can use the clear button in the Clipboard History panel or while Quicksilver is not running, delete the above file.

If you open the drawer for catalog sources, in the Contents tab you'll find all the items found for that source. Clicking on the headings in this tab will sort the list. Large catalog sources (500 or more) will take a long time to sort and use lots of memory. If I accidently sort my Documents with 2000 items I ultimately force quit Quicksilver since it uses 60-80% CPU and several hundred MB of real memory. Be careful as the sort order applies globally. If you sort a small source it works fine, if you then (even accidently) click on a large source it will eat up resources. A Quicksilver restart resets the contents tab to be unsorted.

Quicksilver doesn't find my files
By default Quicksilver finds files on your Desktop and in your Documents folder but it doesn't scan very far into those locations. It scans things on the Desktop but not inside folders on the Desktop, that is it scans the Desktop to a depth of 1. Quicksilver scans the ~/Documents folder to a depth of 2. To scan deeper, clone the source so it's in the Custom set in the catalog and editable. To do this, select the Users set and the Documents source, open the drawer and select the Attributes tab. Click on Create Copy to create a new source in Custom named Documents. Select the new source and click on the button to show a drawer with three tabs. Select the Source Options tab and select the depth. Do not just set it to infinite, the catalog will be too large and Quicksilver will slow to a crawl. See the Catalog section for some tips.

Quicksilver plays or shows notifications for the wrong iTunes track
Yes, unfortunately this is a problem in B51.

Quicksilver crashes
A few plug-in are destabilizing. In particular the Developer Module and Running Applications plug-ins. Try disabling them and see if things improve. TODO: Otherwise, check the log in Console.app to see if there is some log message hinting at what might have gone wrong.

If Quicksilver crashes at startup, see if you have a ~/Library/Application Support/Quicksilver/Shelves/ folder. Quit Quicksilver, move it out of the way and start Quicksilver again. Sometimes the shelf gets corrupted.

Quicksilver won't install any plugins and hangs when trying
This is usually a permissions problem and kGTD has been known to cause problems. Check that your user account (and not root) owns ~/Library/Application Support/Quicksilver/ and the PlugIns folder beneath it. You can check and change this if needed by doing Get Info in the Finder and looking in the Ownership & Permission section. After making sure the folder exists and you own it and have permissions to write to it, restart Quicksilver.

The other cause is if you're running Little Snitch which is a reverse firewall and prevents Quicksilver from contacting the server where the plug-ins are. TODO: I'm not sure if Little Snitch can be configured to to allow Quicksilver to work or if you have to disable it.

Doesn't Spotlight make Quicksilver unnecessary?
No, they do different things and are in fact complementary. Spotlight scans inside files (and their metadata) so you can search their contents. Quicksilver only looks at file names and inside some things like contacts and bookmarks and lets you do more than just open them (e.g., email a file to a contact). Since Quicksilver's index is smaller than Spotlight's and because it learns from your previous searches Quicksilver often finds things faster than Spotlight. The Quicksilver Spotlight Module provides a Quicksilver interface to spotlight by adding 3 actions that work on a text subject.

I don't see feature X
Make sure you have the appropriate plug-in(s) installed. Some features require you to enable advanced features, if so make sure you have done so. If it's an action you don't see, check in

the Action Preferences that the action is enabled (checked). If it's something in the first pane you don't see, make sure it's enabled in the catalog and scanned (so that the catalog source has a number next to the checkbox). This manual tries to be very precise in describing how to enable each feature it describes.

The Quicksilver .dmg doesn't mount
If you downloaded it with Safari, look to see if the downloaded file's name ends with .bz2. Rename the file in the Finder, removing the .bz2 and then try opening the file.

The third pane is empty
If the TextMate Ranker is installed, uninstall it and restart Quicksilver. It's known to break selecting items in the third pane, e.g., when using the **Open With…** or **Email to…** actions. If you're trying to get contacts, make sure the correct plugin is installed (e.g., Address Book) and check that they are enabled in the catalog (can you get to them in the first pane?).

Triggers don't save
TODO: There are usually 3 things that come up with triggers that don't save.

1. So-called dynamic triggers don't save. That is the something like Mail (Get New Mail) won't save because the Get New action come from QS inspecting Mail.app (it's not listed in the actions prefs). Or a trigger using the Menu Bar Items... action with the third pane filled in, won't save.

2. Triggers are saved in ~/Library/Application Support/Quicksilver/Triggers.plist if the permissions of that file or directory don't allow you to write, then no trigger will be saved. There have also been reports of this file becoming corrupted.

3. If you have triggers using actions from some plugin and then uninstall that plugin, the triggers remain and QS can be confused. Sometimes a particular trigger doesn't display, sometimes only one of many triggers appears in the prefs. To fix, reinstall the needed plugins, delete the related triggers, then remove the plugin.

Leopard Specific Things

Removing Time Machine items from Catalog
Uncheck the backup drive in Preferences > Catalog > Devices > Disks > Show Info > Contents. Then rescan the catalog by clicking the curved arrow button at the bottom of the catalog or by activating QS with control-space (or cmd-space or whatever you use) and then typing cmd-R. Also make sure Find All Applications is not checked in the catalog, that scans every connected drive.

LockScreen in Extra Scripts doesn't work
Start the screen saver and have 'Require password after ScreebSaver' in your Security prefs

iCal Plugin doesn't give access to other calendars

Show Menu Items doesn't work for some applications
Fails: TextMate, NetNewsWire

Works: Colloquy

Troubleshooting

http://docs.blacktree.com/quicksilver/support

Is the required plug-in installed?

Is the catalog source, action, trigger enabled?

Are advanced features enabled?

Check Console

Applications and system processes often write useful debugging information to log files. Different things write to different log files, but you can use /Applications/Utilities/Console.app to read all of them. Start Console.app and you'll see the console.log displayed, most applications write their information to this log. If there isn't a left hand pane in the window click the Logs button in the toolbar to make it appear (or use the Show Log List command from the View menu). Make sure the console.log is selected, that's where Quicksilver writes all its information. Since so many things write to the console.log it's useful to filter the display to just what you're looking for, in this case type "Quicksilver" into the Filter box in the top right. Now you can see the Quicksilver log messages. Note that all the messages are preceded with a date and time to make finding relevant messages easier. Sometimes non-Quicksilver messages are relevant to diagnosing a problem, so after finding when something started failing you might unfilter the results and see if anything is reporting problems at the same time.

Task viewer (⌘K) - gear menu invisible in Cube

where's the verbose debugging info you're seeing? you hold down option while starting up and it starts spitting out lots of info to console

Some things (oddly) require being administrator to work correctly

For Crashes, look in Console to find path to crash log, usually it is ~/Library/Logs/CrashReporter/Quicksilver.crash.log

Find what catalog source is finding and item with Show Source in Catalog. e.g., to find what's scanning an external drive.

The following files and folders should be owned by you and be readable and writable by you. You can check this by using the Finder's Get Info command or opening Terminal.app and using the command line `ls -ld` *`file`*.

- `drwxr-xr-x` ~/Library/Application Support/Quicksilver/
- `-rw-------` ~/Library/Preferences/com.blacktree.Quicksilver.plist
- `drwxr-xr-x` ~/Library/Caches/Quicksilver/

program interactions

Little Snitch prevents showing of plug-in list

kGTD might screw up the permissions of ~/Library/Application Support/Quicksilver/

Stoplight prevents Quicksilver from starting

FileVault breaks stuff

Check that what's in the panes is actually what you think is there. Some things look similar but are not the same, e.g., a link to google.com vs a google web search with *** in the URL or the actions. Can check with **Show Source in Catalog**.

Reporting problems

I got an error but don't mention what the error is. It's like saying "Doctor it hurts when I do something"

Describing what you did so vaguely that others can't reproduce or help. It's like saying "I drove to Chicago but when I stopped I wasn't there"

Extending Quicksilver

TODO: Create your own actions: http://docs.blacktree.com/quicksilver/creations/scripts

But that's not to say there isn't a way to do it from Quicksilver. You can find a script on the Quicksilver website at http://docs.blacktree.com/quicksilver/creations/applescript_create_event. Cut and paste it into a file and put that file in `the directory ~/Library/Application Support/Quicksilver/Actions/`.

Set the icon of the script and it appears in the QS pane.

have you got QS > Preferences > Advanced > "run commands in background" turned on? If you have turn it off and try again.

Action for large text Growl notifications:

http://blacktree.cocoaforge.com/forums/viewtopic.php?t=4474

Random Notes

Results List

There are a bunch of ways of choosing an item.

You can type the start of it and then using the mouse scroll through the results list and click on the item you want. That sounds like what you are doing.

You can continue typing the whole thing and it will eventually appear as the first choice. Over time, you should have to type less of the item to get it to the top as QS learns what you do often.

If the results list is appearing slowly go to the Command preference pane and set "Show Other Results" to "Immediately"

You can type the beginning of what you want and then Down-Arrow into through the results list. Up-Arrow and Page-Up and Page-Down and Home and End work as well. People like this method because it's not using the mouse, so it's faster (you don't have to move your hand to the mouse).

Even faster is to configure the "Spacebar behavior" in the Command Preference Pane to "Select Next Result". Now spacebar acts like down-arrow and you don't have to move your hand to the arrow-keys (space is right under your thumb).

You can explicitly assign default choices for a sequence of letters you type. Say you have a "Documents" folder and a "document1.txt" file. Type "d" and they both appear. In the results list right-click on Documents and choose "Set as Default for D" (for this to work it can't be the first choice in the list, since it's already the default)). Now activate QS again and type "docu" and set that as the default for "document1.txt". Now you can always choose "Documents" by typing "d" and "document1.txt" by typing "docu".

If you want to choose a default that's not at the beginning of the name use the Assign Abbreviation action to make an abbreviation (which behaves like a default but is separate). E.g., I do a look of searching in the internet movie database, which is www.imdb.com. I want this assigned to "m" (for movies) but if I just type "m" I get a very long results list and imdb isn't towards the top and I don't want to scroll through the long list to find it to right-click to set a default. Instead select www.imdb.com in the first pane (type it all or do whatever works), and choose the Assign Abbreviation action then in the third pane type "m" (in text mode). Now if you activate quicksilver and type "m" www.imdb.com will appear in the first pane with the default action in the 2nd. Probably all you have to do is type `return`. Very fast.

If you want it even faster. Setup a trigger. This way you can bring up your choice with just one keystroke (perhaps something like option-command-m) or a mouse gesture, without even invoking QS first.

Show Children Split View

My ~/Applications/ has enough in it I've started putting some sub-folders. But they don't show up in the catalog If i create a custom entry I get folders inside the .apps (e.g., Contents) I can't figure out how to select a folder in the Source Options. Any hints? so I'd like my .apps and the folders in the catalog but not what is inside the packages

hmelman: can't you specify types of things that get added to the catalog?

how do you specify the type "folder"? (and I can't see a way to specify "application" unless you copy it from a copy of one of the built-in Application sources)

type folder is specified by 'fold' -- note that includes single quotes i think these things may be discoverable via appropriate Info.plist files

ok, that works but unfortunately includes folders inside .app's :(but I can create a 2nd source for just the top level folders that combined with a depth 3 of just applications gives me what I want

http://quicksilver.infogami.com/ExecutableFileActions

Using Spotlight sources in the Catalog:

http://blacktree.cocoaforge.com/forums//viewtopic.php?p=19515#19515

iTunes artwork not displaying?

I was having this problem as well. Some tracks were displaying cover artwork correctly while others were displaying only the generic iTunes icon. All tracks had the artwork attached to the file.

After searching these forums, (can't find the link) the answer turns out to be setting the proper ID3 tag for each track. It seems that version 2.3 or 2.4 works in QS but v.2.2 or less doesn't.

Select all the tracks that aren't displaying correctly, go to the iTunes Advanced Menu and select Convert ID3 Tags... In the dialog box you'll be given the option to select which Tags you want. After you switch to v.2.3 or 2.4 you should be good to go.

The older Adium plugin (as of B51) works with Adium 0.89.1, it does not work with Adium 1.x. It can also be a little tricky to install if you're upgrading from an older version. It's actually two plug-ins (one for Quicksilver and one for Adium) that communicate with each other. Both parts come with the Quicksilver plug-in and when you run it the first time it installs the Adium part as ~/Library/Application Support/Adium 2.0/PlugIns/Quicksilver.AdiumPlugin. If the plug-in isn't working it's probably an issue with the Adium part. To reinstall it do the following:

1. Quit QS and Adium
2. Open Terminal and run the command:
   ```
   defaults delete com.blacktree.Quicksilver AdiumPluginInstalled
   ```
3. Remove ~/Library/Application Support/Adium 2.0/PlugIns/Quicksilver.AdiumPlugin. Note that the PlugIns directory might have a lowercase i, if so remove any Quicksilver.AdiumPlugin from there too.

4. Double-click on ~/Library/Application Support/Quicksilver/PlugIns/Adium Module.qsplugin