

# Take Control *of* iKey 2

by Adam C. Engst

## Table of Contents (Version 1.2)

iKey Overview .....	1.1
Using This Manual .....	2.1
Introduction to iKey .....	3.1
Installation .....	4.1
Usage Tutorial .....	5.1
Preference Reference .....	6.1
Menu Reference .....	7.1
Hotkey Reference .....	8.1
Window Reference .....	9.1
Command Reference .....	10.1
Launcher Reference .....	11.1
Context Reference .....	12.1

### Check for Updates

Click Here to Look for  
Updates to This Ebook

**TidBITS** Electronic  
Publishing

## iKey Overview

Have you ever noticed yourself, while working at your Mac, repeating the same action over and over again? Perhaps you're typing your mailing address, or mounting the same file server volume every morning, or wishing you could switch to your Web browser at the touch of a key. Or maybe you want to back up your day's work to your iDisk every night at 11 PM. Or what if you want to remap the keyboard shortcuts in an application that believes that Command-Delete should delete a word when you would prefer Option-Delete.

For all these tasks you need iKey. With iKey, you can automate almost any task that you find yourself repeating, whether it's once a week or many times each day. And you can initiate those tasks with the press of a key, by choosing an item from a menu, clicking a button on a palette, when your Mac's clock ticks over to a particular time, or whenever your Mac wakes up. Personally, I couldn't live on my Mac without an automation utility like iKey, and after you set up iKey to take over some of your repetitive work, I'm sure you'll feel the same as well.

### System Requirements

iKey 2 requires Mac OS X 10.2 or higher.

### How to Purchase

iKey 2 costs \$30. Upgrades from previous versions cost \$10. You can purchase iKey 2 and buy upgrades from Script Software's secure online store at <http://www.scriptsoftware.com/store/>. (Note that you must login to the store to purchase upgrades.) Site licenses are available; contact Script Software for details via [http://www.scriptsoftware.com/store/contact\\_us.php](http://www.scriptsoftware.com/store/contact_us.php).

### Tech Support

If you have questions about iKey that aren't answered in this manual or if you need technical support, please visit <http://www.scriptsoftware.com/support/> for links to iKey's FAQ; the support, suggestions, and bug reporting forums; and to find out what to do if you've lost your registration email message and need to reinstall.

## USING THIS MANUAL

Welcome to *Take Control of iKey 2*, version 1.2, written by Adam C. Engst, edited by Tonya Engst, and published by TidBITS Electronic Publishing. This manual helps you learn and use Script Software's iKey 2 automation utility.

To get in touch or learn more about TidBITS Electronic Publishing, you can:

- Contact us by sending email to [tc-comments@tidbits.com](mailto:tc-comments@tidbits.com).
- See [About This Ebook](#) to learn about the author and publisher.
- Read the fine print on the [copyright page](#).
- See more Take Control ebook titles at <http://www.takecontrolbooks.com/catalog.html>.

We offer free updates to this manual. Click the Check for Updates button on the [cover](#) to access a Web page that informs you of any available or upcoming updates. On that page, you can also sign up to be notified about updates via email.

### Onscreen Reading Tips

We carefully designed the Take Control ebooks to be read onscreen, and although most of what you need to know is obvious, note the following for the best possible onscreen reading experience:

- Blue text indicates links. You can click any item in the Table of Contents to jump to that section. Cross-references are also links, as are URLs and email addresses.
- Work with the Bookmarks tab or drawer showing so that you can always jump to any main topic by clicking its bookmark.
- If you use Adobe Acrobat Professional, please make sure your Web Capture preference is set to view Web URLs in a Web browser. To do so, choose Acrobat > Preferences, switch to the Web Capture pane, and choose In Web Browser from the Open Web Links pop-up menu.
- Find more tips in the [Take Control FAQ](#) on the Web.

## Printing Tips

Although our layout is aimed at making online reading an enjoyable experience, we've made sure that printing remains a reasonable option. Please review these tips before you print:

- Use the Check for Updates button on the [cover](#) to make sure you have the latest version of the ebook and to verify that we don't plan to release a new version shortly. If you want to commit this ebook to paper, it makes sense to print the latest possible version.
- Don't throw out your PDF after you print! You must click the Check for Updates button on the cover to get future updates. The link *must* be accessed from the cover of your PDF.
- For a tighter layout that uses fewer pages, check your printer options for a 2-up feature that prints two pages on one piece of paper. For instance, your Print dialog may have an unlabeled pop-up menu that offers a Layout option; choose Layout, and then choose 2 from the Pages per Sheet pop-up menu. You may also wish to choose Single Hairline from the Border menu.
- When printing on a color inkjet printer, to avoid using a lot of color ink (primarily on the yellow boxes we use for tips and figures), look for an option to print entirely in black-and-white.
- In the unlikely event that Adobe Acrobat or Adobe Reader cannot successfully print this PDF, try Preview; several readers have solved printing problems by using Preview.

## Basics

In reading this ebook, you may get stuck if you don't understand Take Control syntax for things like working with menus or finding items in the Finder. Please note the following:

- **Path syntax:** This ebook occasionally uses a *path* to show the location of a file or folder in your file system. Path text is formatted in bold type. For example, Panther stores most utilities, such as Terminal, in the Utilities folder. The path to Terminal is:  
**/Applications/Utilities/Terminal.**

The slash at the start of the path tells you to start from the root level of the disk. You will also encounter paths that begin with ~ (tilde), which is a shortcut for any user's home directory. For example, if a person with the user name **joe** wants to install fonts that only he can access, he would install them in his **~/Library/Fonts** folder, which is just another way of writing **/Users/joe/Library/Fonts.**

- **Menus:** This manual sometimes instructs you to choose, or describes someone as choosing, a command from a menu in the menu bar. In such cases, the command is specified starting with the menu title in the menu bar and proceeding down the hierarchy. A “greater-than” sign separates menu titles and menu item names. Thus, to describe the command you choose to save and activate your changes in iKey, I would speak of File > Save and Activate.

## Terminology

In reading this manual, you may encounter a few unfamiliar terms, but it's extremely important that you understand them from the start.

<b>command</b>	An individual action that you can automate with iKey, such as typing some text, launching an application, or mounting a network server. You can combine multiple commands within a single shortcut.
<b>context</b>	The application(s) in which a shortcut, menu, or palette is available. Shortcuts which use the Universal context are available at all times; other contexts limit shortcuts to a specific application, with the additional variables of whether the applica-

tion is in the front or back, and whether or not it's running.

**hotkey**

One or more keys used in a keyboard launcher to activate a shortcut. Keys available for use as hotkeys include any normal alphanumeric key coupled with one or more modifier keys; a function key (the F keys at the top of the keyboard) either by itself or in conjunction with one or more modifier keys; or any key from the numeric keypad (including any arrow key or other navigation key), either by itself or with one or more modifier keys.

**launcher**

The way a shortcut, menu, or palette is activated. Although the most common launchers are attached to specific key combinations, other launchers can trigger based on date and time; based on a particular application launching, activating, deactivating, or quitting; or system events like startup, waking from sleep, going to sleep, restarting, or shutting down.

**library**

The iKey window from which you drag already-defined contexts, launchers, and commands to add them to a shortcut, menu, or palette.

**menu**

Within iKey, a user-defined collection of individual commands or shortcuts. You can create any number of menus and access them by clicking them in the menu bar or pressing a key combination to make them appear at the pointer location.

**palette**

A window containing icons that, when clicked, initiate individual commands or shortcuts. When the pointer is outside a palette, it can become transparent or shrink to its title bar.

**shortcut**

A collection of one or more commands that can be activated by a launcher or included in a menu or on a palette. As much as commands are the building blocks from which shortcuts are made, shortcuts are in essence the core object around which all of iKey is based.

## What's New in Version 1.2

Version 1.2 of this manual covers the changes made in iKey 2.2, including the following:

- [Added additional description](#) of what happens (or more accurately, what doesn't happen) when iKey launches.
- Noted in a [tip](#) that Apple's Backup program can now automatically back up iKey settings.
- [Added information](#) about Control-clicking or right-clicking a shortcut's or menu's icon well for new commands that help you choose an icon.
- Mentioned in a [tip](#) that iKey 2.2 now preserves the contents of the clipboard when you use certain Clipboard commands.
- Described two new commands: [Save Clipboard](#) and [Restore Clipboard](#).
- Added a [note](#) with advice about duplicating shortcuts that use Type Text commands.
- Added the [Drag By](#) mouse action to Table 10.2.
- Described how to [create and use double-key hotkeys](#) with keyboard event launchers.
- Added a [tip](#) about how iKey now warns you if there's a conflict between an iKey hotkey and a system hotkey.
- Described iKey 2.2's new [USB device event launcher](#).

## What's New in Version 1.1

Version 1.1 of this manual tracks the significant changes made in iKey 2.1. Most notable among these changes in the manual are:

- A new [note](#) explains how iKey 2.1 uses existing an existing context, launcher, or command if you attempt to create a new one with the same parameters as an existing one.
- [Installation instructions](#) now acknowledge that you could be installing over a previous version of iKey 2.

- The [Usage Tutorial](#) and [Window Reference](#) have been updated to account for the Contexts pane, the Library window, and the new methods of assigning already-defined contexts, launchers, and commands to shortcuts, menus, and palettes.
- The [Find a Particular Shortcut, Menu, or Palette](#) section now takes into account the Contexts pane and the Used By pop-up menus in the Library window.
- Six new sections in the Usage Tutorial summarize all the actions you can perform on [shortcuts](#), [menus](#), [palettes](#), [contexts](#), [commands](#), and [launchers](#).
- A pair of [tips](#) explain three commands—Show Invalid Parts, Show Unused Parts, and Activate/Deactivate Hotkeys—that no longer exist in the File menu because they’re unnecessary in iKey 2.1.
- The following commands have been added to the Command Reference or modified significantly: [Run Last Shortcut/Command](#), [Run Next to Last Shortcut/Command](#), [Test Edited Shortcut/Command](#), [Select Menu Item](#), [Sleep](#), and [Wait for window events](#).



## INTRODUCTION TO iKEY

We humans enjoy creative work, whether it's writing text, drawing and painting, composing music, analyzing numbers, or discovering relationships in data. But, and there's no shame in this, most of us are lousy at repetitive work. We get bored, we make mistakes, and we forget both individual steps and entire tasks.

That's where machines—computers in this context—come in. A machine can perform exactly the same task for hundreds, thousands, or millions of times without so much as missing a beat. It can be easy to lose sight of this fact when working with programs that enhance our creativity or enable us to perform tasks that might not otherwise be possible, but the reality is that almost everything we do on our Macs has some aspect of repetition to it.

Perhaps the repetition comes in typing out your email address, in opening the same set of Web sites to read with your morning coffee, or pausing your music in iTunes when the phone rings. Or perhaps you find yourself opening the same database and performing the same search every morning when you come into work; what's important here is that you're wasting your time and energy on a task that your Mac can do faster and more accurately.

Of course, your Mac can't take over your grunt work on its own. You need a program to help you take all these shortcuts in your everyday tasks. You need iKey.

iKey is an automation utility, a program that creates shortcuts to accomplishing repetitive tasks. In essence, an iKey *shortcut* is a little program in its own right, but you don't need to know the first thing about programming to create an iKey shortcut. All you have to do is put together three necessary parts of a shortcut: one or more *commands* that give the shortcut its functionality, a *context* in which it runs, and a *launcher* that defines how the shortcut is activated.

Shortcuts are the core of iKey, and much of the time you will find yourself invoking individual shortcuts to perform your repetitive tasks, but there are two other interfaces for invoking shortcut actions: *menus* and *palettes*. A menu, as you'd expect, contains multiple shortcuts in the traditional Macintosh menu form, though you can either pin your menu in the menu bar or have it appear when you

press a combination of keys (called a *hotkey*). Choosing an item in the menu activates its associated shortcut. Similarly, a palette displays multiple commands as buttons within a window that you can either have showing all the time or invoke only when you want it to appear; clicking a button in a palette activates the associated shortcut.

Menus and palettes are built from the same parts as shortcuts, though they're slightly different, since menus pinned to the menu bar don't require launchers (since they're always available) and their functionality comes from both commands and other shortcuts. Let's look at each of these parts in turn.

## Commands

Arguably, everything you do on your Mac can be reduced to mouse movements and keystrokes, since those are the two primary ways you communicate with running programs. So in fact, iKey could do nothing more than simulate the mouse and the keyboard and still be able to automate everything you can do.

But just because mouse movements and keystrokes are the most elemental ways to perform actions in Mac OS X doesn't mean that attempting to automate a task using only simulated mouse movements and keystrokes would be an easy task. Just think of how hard it would be choose a particular menu item using mouse movements: you'd have to show iKey how to click the mouse button over the menu's screen location, drag down to the appropriate menu item, and release the mouse button. And if the position of that menu item were to change (think about the names of an arbitrary set of open windows in a Window menu), your shortcut would break.

So while iKey of course lets you simulate the keyboard and mouse, it also offers a plethora of higher level commands: launching and quitting applications, choosing menu items, copying and pasting text, running an AppleScript script, visiting URLs, creating email messages, and much more.

These commands are the basic building blocks of a shortcut's functionality. Want to create a shortcut that changes your location in the Network preference pane? A single command is all that's necessary. On the other end of the spectrum is a shortcut I created to help me locate bouncing email addresses in a mailing list; it copies the selected text in an email message in Eudora, switches to a BBEdit

document containing the mailing list, opens the Find dialog, pastes the copied text into the Search For field, clicks the Find button to locate the text, deletes it by pressing the Delete key, and returns to Eudora so I can look for the next bouncing email address. That shortcut relies on a number of different commands that copy text, switch between applications, and enter keystrokes.

Many of iKey's commands are simple—they do just what you'd expect and don't allow much customization—whereas others allow you to tweak and fiddle to your heart's content. Refer to the appropriate section in [Command Reference](#) for details of each command.

I wasn't kidding when I said iKey has a plethora of commands—it sports over 100 unique commands. Needless to say, that's too many to present together, so iKey gathers them into categories, as you can see in **Table 3.1**.

**Table 3.1: Command Categories**

Category	Function of Commands in this Category
Application	Launch, quit, hide, and switch among applications
Clipboard	Cut, copy, paste, and work with text on the clipboard
Document	Open documents
Folder	Show the contents of folders
iKey	Control iKey itself
Internet	Create email and visit URLs
Keyboard	Press keys and type text
Menu	Choose items from menus
Mouse	Simulate mouse movements, clicks, and scroll wheel actions
Network	Change network settings and mount servers
Printer	Select printers and print documents
Script	Run AppleScript scripts
Sound	Play sounds
System	Perform basic system tasks like sleeping, ejecting discs, invoking the screen saver, and more
Unix	Run collections of Unix commands
Utility	Wait for certain things to be true before continuing
Window	Open, close, move, resize, scroll, and switch among windows

## Contexts

One of the aspects of growing up is learning how to behave appropriately in context. A toddler won't think twice about throwing a temper tantrum in a restaurant because his parents won't pick up the fork he just threw on the floor for the third time, whereas an adult (at least a well-mannered one) would never consider making a loud scene in a

public place like a restaurant. Adults understand context—even if a loud scene is warranted, a restaurant isn’t the place for it.

The same is true for iKey. Every shortcut, menu, and palette needs a context in which it exists. By default, every shortcut, menu, and palette you create starts with the Universal context, which means that it’s always available, no matter what application you might be using.

But imagine that you hate typing sets of Unix commands in Terminal, so you’ve created a number of shortcuts that type those commands for you. You could invoke them individually with hotkeys, or you could create a menu or palette that holds them all and enables you to activate them by choosing a menu item or clicking a palette button. Since these commands work only Terminal, there’s no reason to make them available when you’re in any program other than Terminal (well, unless you want to make sure that the shortcuts switch to Terminal before executing, which is certainly possible).

Another good reason to assign a particular context to a shortcut is if it’s correct only in a particular application. For instance, perhaps you believe (rightly, in my opinion) that Command-Delete should always delete the word to the left of the insertion point. However, it’s likely that some application that you use regularly doesn’t include such functionality. You can simulate it with a shortcut that includes a command that holds down Shift and Option and presses the left arrow key, combined with another command that presses Delete. However, you don’t want such a workaround to use the Universal context since then your shortcut (which is slower, clumsier, and more prone to failure than a properly implemented Command-Delete) will override the built-in Command-Delete functionality in those applications that have it. So instead of using the Universal context, set the context to be just the application that benefits from your shortcut.

**NOTE** If you assign multiple contexts to a shortcut, menu, or palette, it will be active whenever any of those contexts is satisfied. In other words, if you make a menu that includes context for Safari being in front and OmniWeb being in front, that menu will be active whenever either Safari or OmniWeb is the front application.

In short, contexts are both essential and extremely useful, and even though the default Universal context is often completely appropriate,

don't hesitate to apply more specific contexts. All told, iKey offers five context types (**Table 3.2**).

<b>Table 3.2: Context Types</b>		
<b>Context</b>	<b>Restriction</b>	<b>Notes</b>
Universal	None; shortcuts that use the Universal context are always available.	The default context; Universal is appropriate in many situations.
Application in Front	Shortcuts that use this context are available only when the specified application is the front application.	After Universal, the second most commonly used context
Application in Background	Shortcuts that use this context are available only when the specified application is in the background.	Useful only in very specific situations.
Application Running	Shortcuts that use this context are available only when the specified application is active, without caring if it's in the front or background.	Useful only in specific situations where Application in Front isn't appropriate.
Application Not Running	Shortcuts that use this context are available only when the specified application has yet to be launched.	Useful for avoiding conflicts that could occur between different versions of applications that might fight for control of files.

## Launchers

The final part of a shortcut is the launcher, which you can think of as the trigger that invokes the shortcut, menu, or palette. However, launchers are unnecessary in three situations:

- A menu that appears in the menu bar doesn't need a launcher because it's always available by definition.

- A palette that's set to show when activated by its context doesn't need a launcher because it appears appropriately whenever its context is correct.
- A shortcut that you want to invoke only from within a menu or palette doesn't need a launcher, because the action of choosing it from the menu or clicking its button in the palette takes the place of the launcher.

When you do want to include a launcher for a shortcut, the most obvious type of launcher is a hotkey, a combination of keys that you press whenever you want to activate the shortcut. In the example I gave earlier of a shortcut that deletes the word to the left of the insertion point, the launcher would be Command-Delete; the action of pressing Command-Delete triggers the shortcut that selects the word and then deletes it.

Menus and palettes can also be activated with a hotkey; pressing it displays the menu or palette, displaying the menu (at the position of the pointer) or the palette (in its assigned location).

iKey shortcuts can also be triggered by date/time events, application events, and system events. Once created, these shortcuts are triggered without user interaction. **Table 3.3** describes these other types of launchers in more detail, but what's interesting about these launchers is that they enable you to create shortcuts that require even less work on your part. For instance, I have a shortcut that opens a number of Web pages in tabs every day. But I want to do this (with different pages) in both OmniWeb and Safari, and it's not something I want to have to remember to do; I'd like to have it happen every day at 9:00 AM, when I start my day. A simple date/time event launcher triggers this shortcut, saving me both the effort of remembering and of invoking the shortcut. Once again, iKey does for me the boring repetitive things that I don't want to do on my own.

**Table 3.3: Launcher Types**

Launcher	What Triggers It	Notes
Keyboard Event	Pressing a combination of keys (including the function keys), often including one or more modifier keys	The most common type of launcher; used to create a hotkey
Date/Time Event	The Mac's system clock ticking over to the specified time	Works only if the Mac is turned on and awake at the specified time, though iKey can alert you if a timed launcher failed to trigger because of the Mac being off or asleep
Application Event	An application launching, quitting, activating, or deactivating	Often good when combined with application-specific contexts
System Event	A system event, including startup, shutdown, sleep, or wake from sleep	Good for housekeeping shortcuts that do things like making backups before shutdown
Context Event	Activation of the context for a shortcut, menu, or palette	Helpful when you want to trigger a shortcut based on application state

**NOTE** In iKey 2.1, if you add a new context, launcher, or command that exactly duplicates an already-defined one, iKey Editor uses the existing one instead of creating a new one. Similarly, if, while modifying a context, launcher, or command, you make it the same as an existing one, iKey Editor gives you the option of overwriting the existing one with your new one or changing the one you're editing to make it different from an existing one.



## INSTALLATION

Installing iKey is a simple process, whether you're installing fresh or updating from a previous version. A standard installation includes:

- The iKey application package.
- This manual.
- The release notes for the current version.
- A folder containing some sample menu bar icons.
- A Purchase folder that contains two files that are links to the Script Software store for purchasing iKey (the .url file launches Internet Explorer, and the .webloc file launches your default Web browser), along with an RTF file that you can print and mail in if you prefer not to order over the Internet or wish to pay by check.

### Installing and Launching iKey 2

To install iKey 2, first quit any previous version of iKey that is running (by choosing Quit iKey from the iKey menu), and then simply drag the iKey folder to your Applications folder (it's fine to replace a previous iKey 2 folder; iKey stores your shortcuts elsewhere). Then double-click the iKey icon to launch iKey.

Because iKey works behind the scenes, you won't see the normal zoom animation when it launches, nor will any splash screen or window appear. The only indication that iKey is running is the iKey menu icon, which is a little white key, with a blue, i-shaped, lightning bolt emblazoned on it. Along with the Quit iKey item, that menu also contains an iKey Editor item; choose it to launch iKey Editor, which provides the visual interface for working in iKey.

On the first launch, iKey adds itself to the Startup Items list of your Accounts preference pane; in the future, iKey will launch automatically whenever you log in. iKey also creates an iKey folder in **~/Library/Preferences** to store all your shortcuts, menus, palettes, and other preferences.

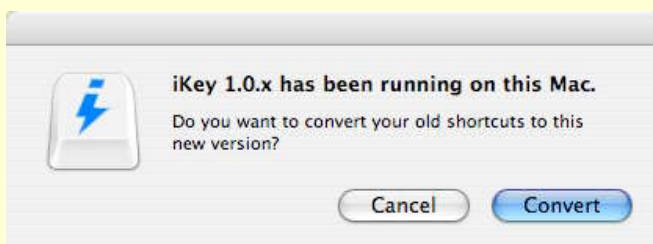
**TIP** Although I strongly encourage you to back up at least everything in your home folder, or even your entire Mac, if you are using a selective backup strategy, make sure to back up `~/Library/Preferences/iKey/` to protect the effort you've put into your shortcuts. iKey 2.2 adds support to enable Apple's Backup application, available to all .Mac members, to back up your iKey shortcuts, menus, palettes, and settings.

**WARNING!** Since iKey is watching for keyboard events to activate shortcuts, it's possible that you may experience unusual or incorrect behavior if you're running other automation utilities like QuicKeys, Keyboard Maestro, or KeyQuencer, and that's true even if the utility in question is running in Classic.

## Updating from iKey 1.0.x

If you have previously used iKey 1.0.x, follow the installation instructions for a new installation, but note that the first time you launch iKey 2, it will ask if you want to convert your iKey 1.0.x shortcuts (**Figure 4.1**). It's highly likely that you do want to convert your existing shortcuts, because otherwise you'll have to recreate them manually. Note that converting your existing iKey 1.0.x shortcuts has the effect of replacing the default iKey example shortcuts. See below if you want to keep them as well.

**FIGURE 4.1**



Convert your iKey 1.0.x shortcuts when you're prompted.

## Converting old shortcuts and retaining new shortcuts

If you want to convert your old iKey 1.0.x shortcuts *and* combine them with the new examples included with iKey 2, follow these steps:

1. Install and run iKey 2 without converting your old shortcuts.
2. Launch iKey Editor

3. Select those shortcuts you want to keep (note that you must repeat this step and the next with any menus or palettes you want to keep).
4. Choose File > Export to export them to a file on the Desktop.
5. Quit iKey Engine and iKey Editor.
6. Delete the folder located at `~/Library/Preferences/iKey`.
7. Relaunch iKey Engine by double-clicking the iKey icon in the Finder.
8. On this second launch, iKey will again ask if you want to convert your old shortcuts; agree to the conversion this time.

**TIP** For iKey 2 to ask if you want to convert your old shortcuts, these items must be present in `~/Library/Preferences`: **iKey.plist**, **iKey Editor.plist**, and **iKey Folder**. If, for some reason they aren't there, you can reinstall them from backup or from an old copy of iKey 1.0.x to enable the conversion to happen properly.

9. Drop the exported file on iKey Editor's Dock icon (or use File > Import) to import iKey 2's example shortcuts back in.

### Removing iKey 1.0.x

After you've installed iKey 2, you can delete the iKey 1.0.x folder from your Applications folder. However, to leave your Mac sparkling clean, you may also wish to delete the following items from `~/Library/Preferences`:

- iKey.plist
- iKey Editor.plist
- iKey Installer.plist
- iKey Folder (note the exact name and be careful not to delete the folder called "iKey" that iKey 2 uses)

## USAGE TUTORIAL

Let's get started with iKey by learning how to perform the basic tasks of creating new shortcuts, menus, and palettes, managing the different types of objects, and more.

### Launch iKey Editor

All the tasks of creating and editing shortcuts take place in a separate application called iKey Editor. Follow these steps to launch iKey Editor:

1. If the iKey lightning bolt icon isn't in your menu bar, double-click the iKey application icon.
2. From the iKey lightning bolt menu, choose iKey Editor.

iKey Editor opens, displaying all your current shortcuts.

**NOTE** The iKey lightning bolt menu, though installed by default, isn't anything special at all. It's just a normal iKey-created menu that you can alter or remove as you like.

**TIP** If you double-click the iKey icon in the Finder while iKey Engine is already running, you're presented with a dialog that gives you Launch iKey Editor and Quit iKey Engine buttons; these buttons do exactly what they say they'll do.

**TIP** I strongly encourage you to add a keyboard event launcher to the iKey Editor command so you can open iKey Editor easily with a hotkey.

**TIP** You won't find iKey Editor in the iKey folder because it's hidden within the iKey application package itself, along with the iKey2Converter application that converts iKey 1.0.x shortcuts.

## Create a Shortcut

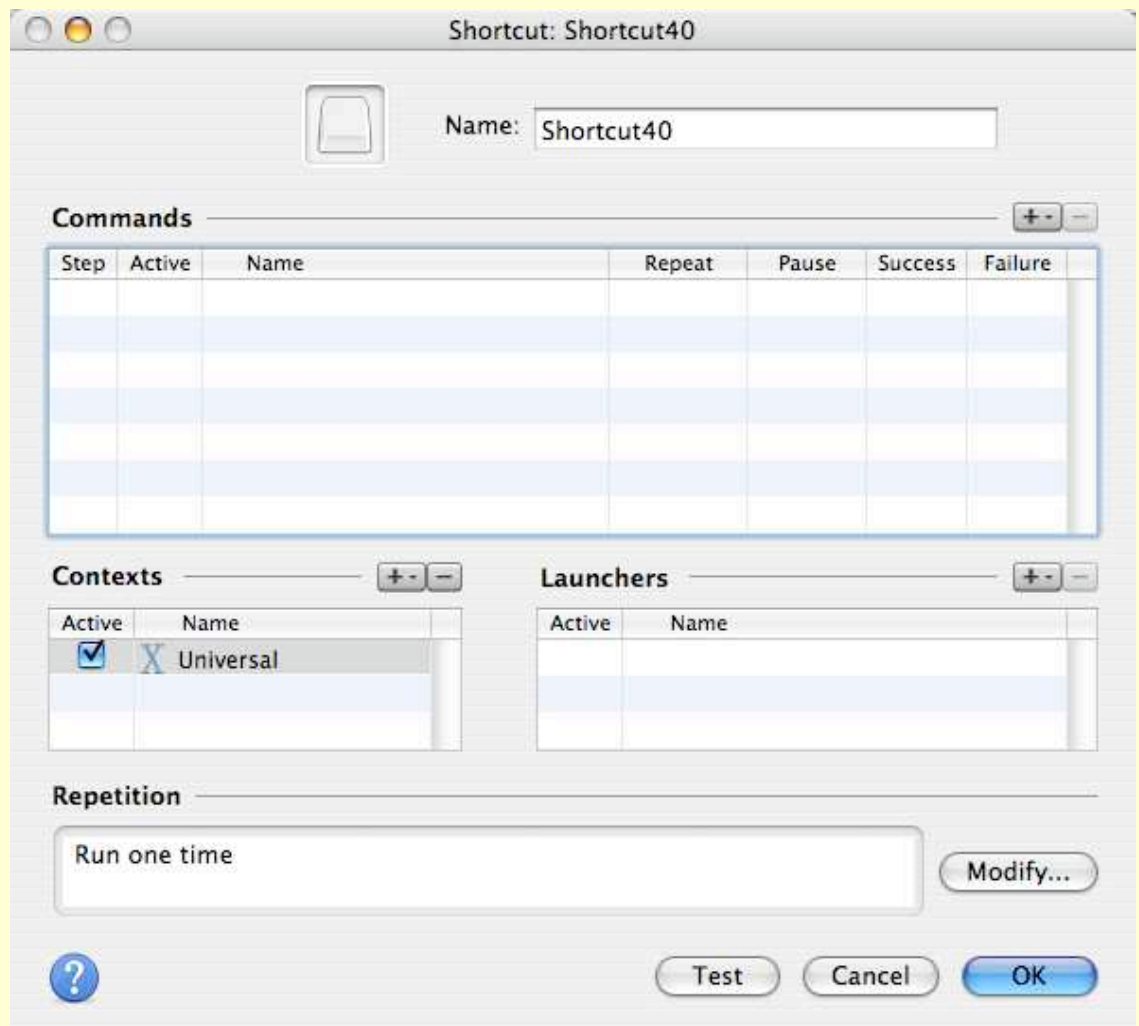
Follow these steps to create a shortcut in iKey Editor. Obviously, the details will vary depending on what you want your shortcut to accomplish, but the basic steps are the same. I assume in these steps that you haven't created any commands, launchers, or contexts; after you've created some commands, launchers, and contexts in the process of making shortcuts, you may find that they can be reused by dragging them into the shortcut creation dialog from the Library window.

**NOTE** For this example, I show you how to create a shortcut that types your email signoff (mine is **cheers... -Adam**) and plays a beep sound. The shortcut will work only in your email program.

1. **Open the shortcut creation dialog:** Click the Create button (+) in iKey Editor's toolbar when the Shortcuts tab is selected.

iKey Editor opens an empty shortcut creation dialog (**Figure 5.1**).

**FIGURE 5.1**

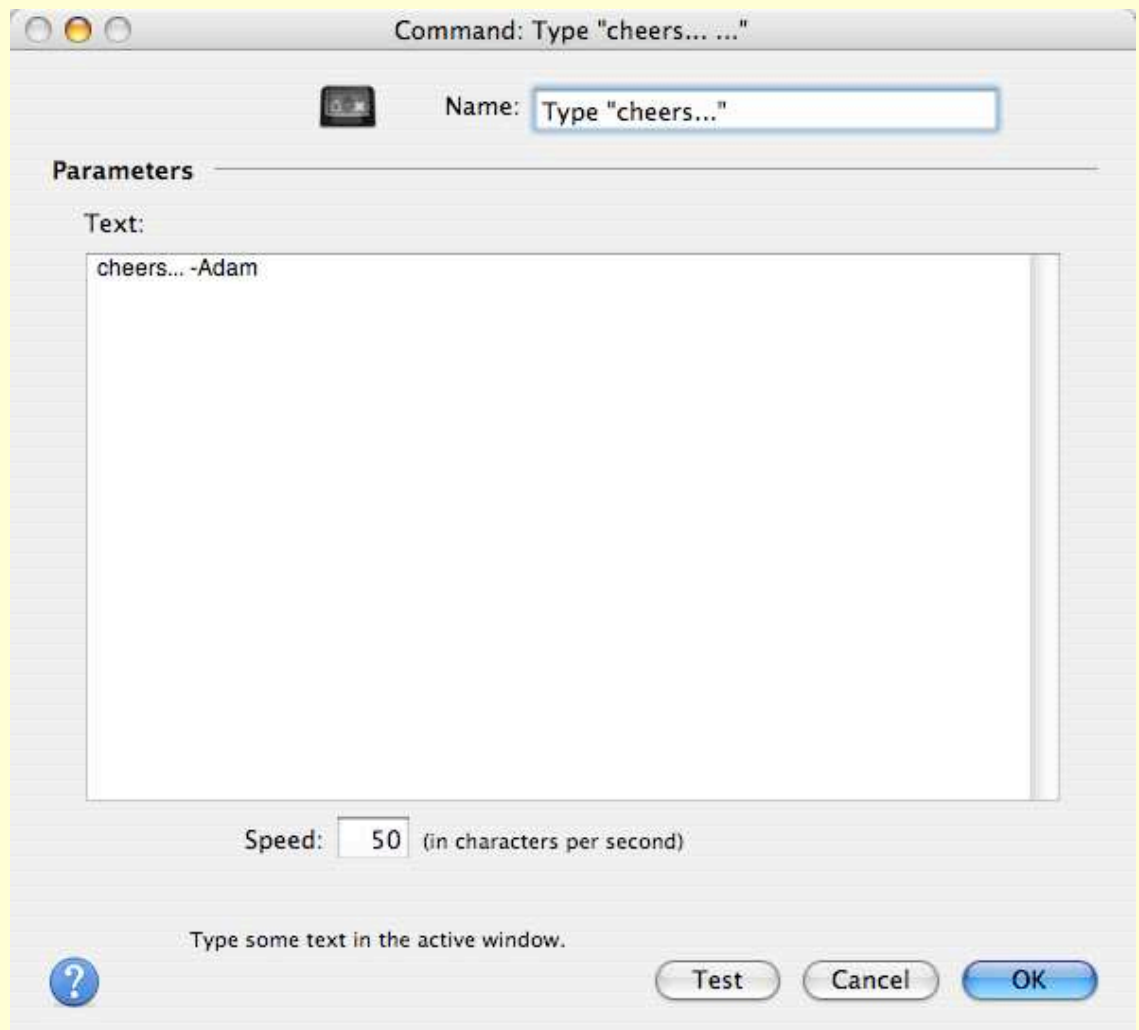


An empty shortcut creation dialog.

2. **Add your first command:** Using the + pop-up menu in the Commands area, navigate through the hierarchical menus to choose the command you wish to have executed first by your shortcut.

For this example, I've chosen + > Keyboard > Type Text, and iKey has opened the Type Text command dialog (**Figure 5.2**). Enter some text here; I'm using my standard email signoff. Click OK when you're done.

**FIGURE 5.2**

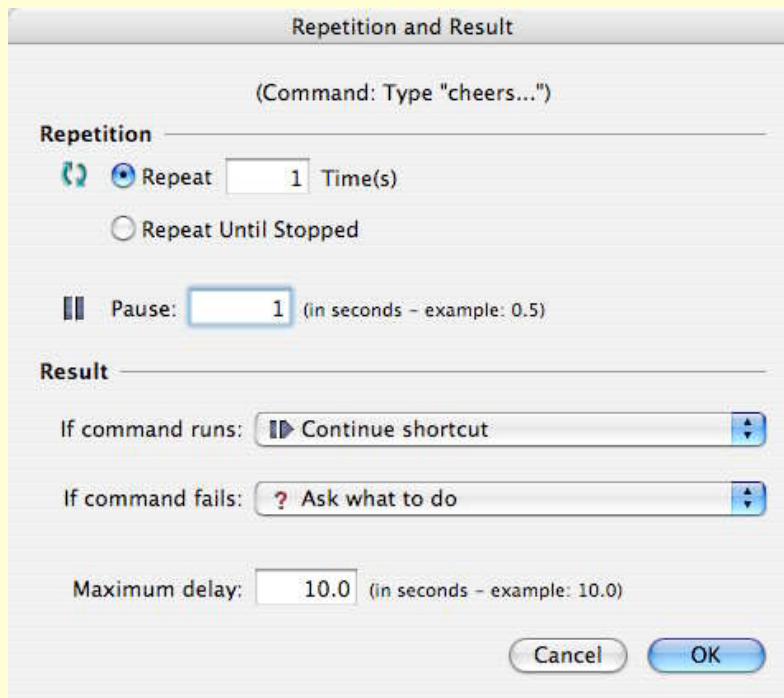


Configure the command you've chosen.

**TIP** To get started even faster, instead of clicking the + button and then choosing your first command in the shortcut creation dialog, choose the first command you want to add from the appropriate Create > Shortcut submenu. In this case, it would be Create > Shortcut > Keyboard > Type Text.

3. **Tell iKey what to do after it issues the first command:**  
Back in the shortcut creation dialog, double-click the Pause cell in the row for the command you just created to display the Repetition and Result dialog (**Figure 5.3**). We're going to add another command, but we want to make sure this one finishes before the next one runs. In the Pause field, enter **1** to make the shortcut pause for 1 second after the command executes, and then click OK.

**FIGURE 5.3**



Configure any repetitions, pauses, or success or failure responses, necessary for the command.

(You could also use the Repetition and Result dialog to add repetitions, increase the maximum delay for commands that take a long time to complete, and change how iKey responds to the success or failure of the command.)



4. **Add more commands:** Repeat Steps 2 and 3 to add additional commands for the shortcut to execute in order.

For this example, choose + > Sound > Play Sound File and in the Play Sound File command dialog, from the Sound pop-up menu choose Select. Then, select **/System/Library/Sounds/Sosumi.aiff (Figure 5.4)** and click Select.

**FIGURE 5.4**



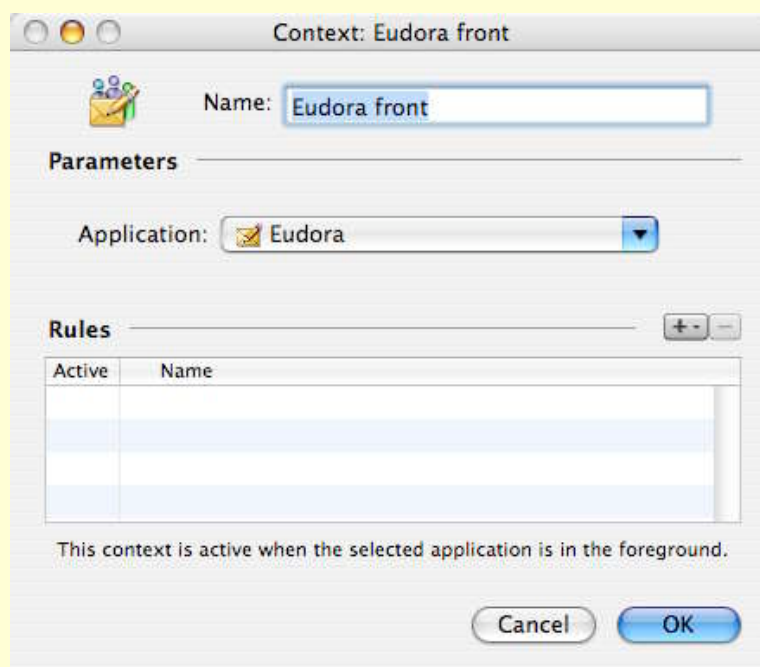
Add another command.

5. **Set the context for the shortcut:** We could leave the Contexts area alone, since there's probably no harm in having this shortcut work in any application (at this point the default Universal context is already set), but since it's only meant to be used in email, let's create an application-specific context:

- a. Using the + pop-up menu in the Contexts area, navigate through the hierarchical menus to choose the New item for the context you wish to apply to your shortcut.

In this case, I chose + > Application in Front to open the Application in Front context dialog, where I chose Eudora (my email program) from the Application menu (**Figure 5.5**).

**FIGURE 5.5**



Create a context for your shortcut.

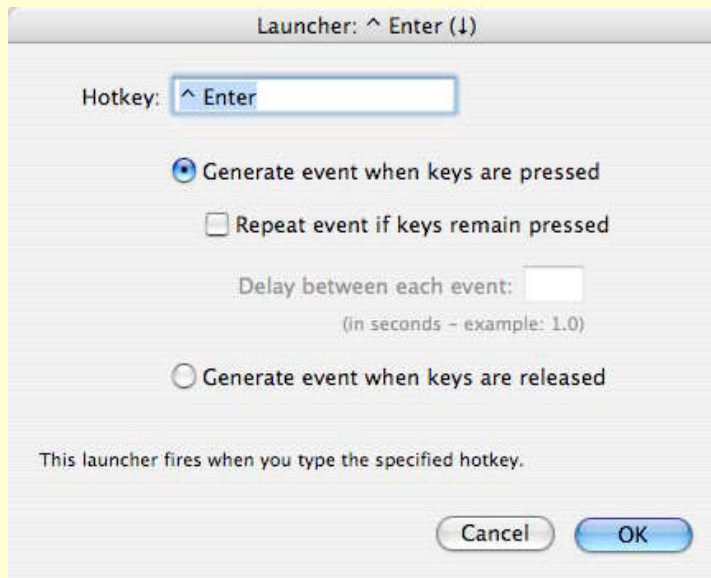
**NOTE** You can use the Rules to set conditions for the context: Deactivate One Hotkey and Deactivate All Hotkeys. These rules basically say, "When this context is active, deactivate these iKey hotkeys so they don't conflict with hotkeys built into that application."

- b. Click OK when you're done.
- c. Select the Universal context and either click the — button or press Command-Delete to remove it.

6. **Set up a launcher:** Using the + pop-up menu in the Launchers area, navigate through the hierarchical menus to choose the New item for the launcher you wish to use to invoke your shortcut.

For this example, I chose + > Keyboard Event, and iKey Editor opened the Keyboard Event launcher dialog (**Figure 5.6**). Type the hotkey you want to use; I used Control-Enter. Click OK when you're done.

**FIGURE 5.6**

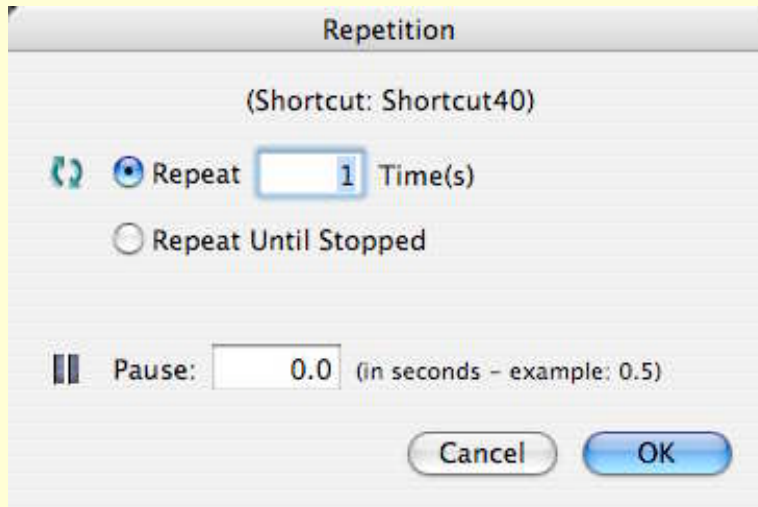


Create a launcher with which you'll invoke your shortcut.

**TIP** The options in the Keyboard Event launcher dialog are generally unnecessary. Some people prefer to trigger shortcuts when they release the hotkey, rather than when they press it; hence that radio button. And, in certain cases, you may want your shortcut to repeat automatically if you keep the hotkey pressed; if so, select that checkbox and enter a delay between executions.

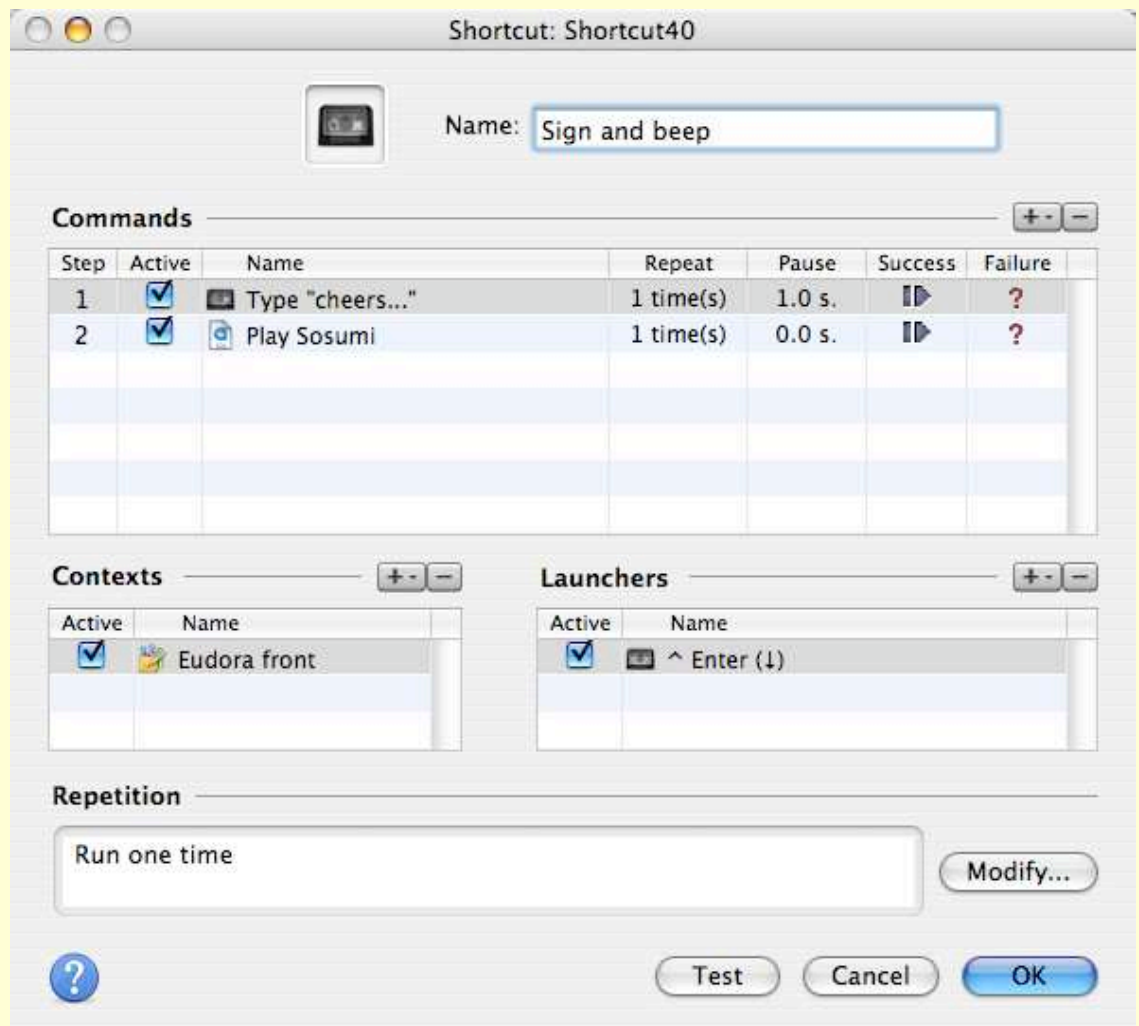
7. **Set repetitions for your shortcut:** For this example, we'll leave the Repetitions alone; however, if you wanted the shortcut to run some particular number of times, you could click the Modify button and enter a number of repetitions and a delay between them (**Figure 5.7**).

**FIGURE 5.7**



Enter a number of repetitions, if desired.

8. **Name your shortcut:** Enter a name in the Name field, and then click OK to save it (**Figure 5.8**). Obviously, you can name it at any time, but it's usually easiest to come up with a name right after you've defined it.

**FIGURE 5.8**

The finished shortcut.

9. **Test the shortcut:** With some shortcuts, you could click the Test button to see how well they work, but with this one, clicking the Test button won't do anything useful, since typing text requires a window in which to type it.
10. **Save and try your shortcut:** Choose File > Save and Activate (Command-S) to make your new shortcut ready to use, and then switch to your email program and invoke it with Control-Enter (or whatever hotkey you used). If it doesn't work properly, some of the settings may not match those shown in **Figure 5.8**. To fix that, return to iKey Editor, double-click the shortcut to edit it, and make the necessary changes before saving and activating again.

## Create a Menu

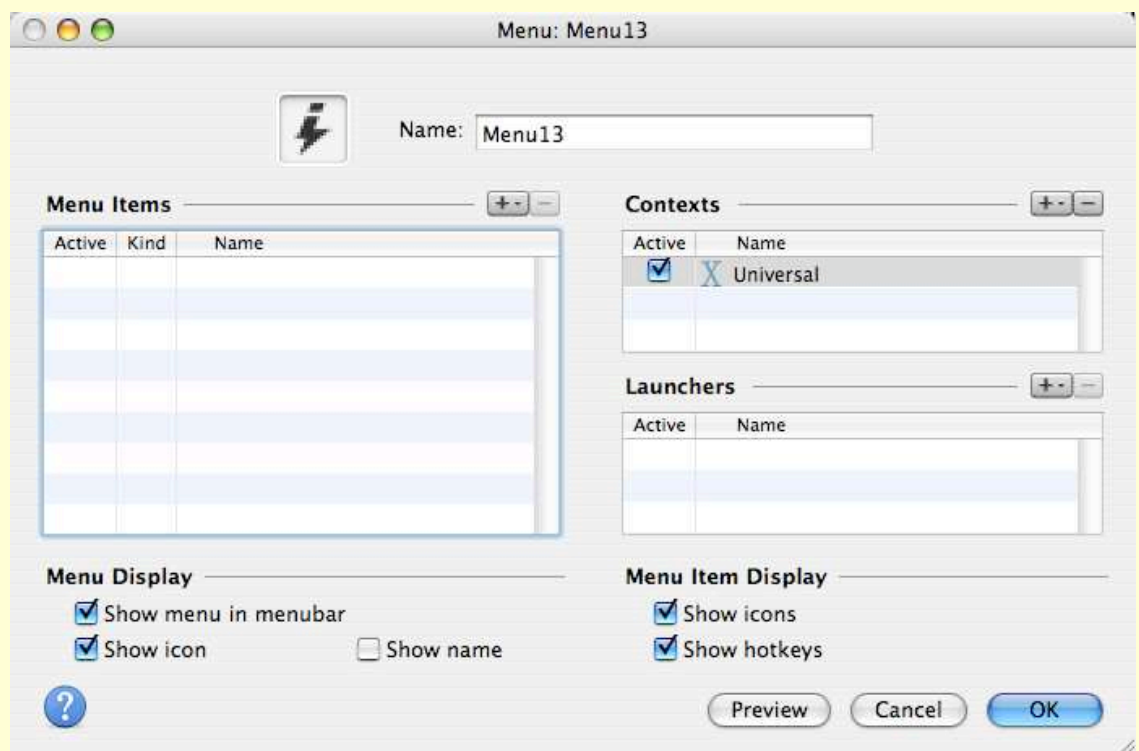
Follow these steps to create a menu in iKey Editor. Obviously, the details will vary depending on what you put in your menu, but the basic steps are the same. For the purposes of this example, I assume in these steps that you've created only the commands, launchers, and contexts from the [Create a Shortcut](#) steps.

**NOTE** For this example, I show you how to create a menu in the menu bar that contains an item that displays the locations in your Network preference pane (you may have only Automatic; the item is more useful if you have additional locations) and provides an item that invokes the shortcut you created previously.

1. **Open the menu creation dialog:** With the Menu tab selected, click the Create button (+) in iKey Editor's toolbar, or choose Create > Menu.

iKey Editor opens an empty menu creation dialog (**Figure 5.9**).

**FIGURE 5.9**



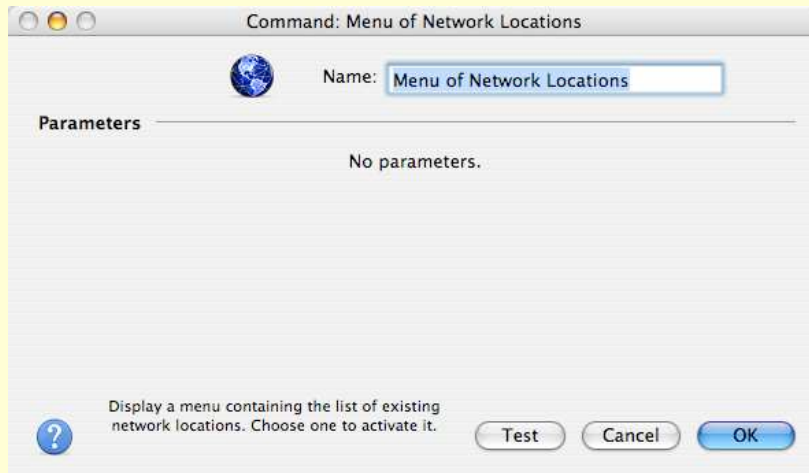
An empty menu creation dialog.

2. **Add your first menu item:** Using the + pop-up menu in the Menu Items area, navigate through the hierarchical menus to

choose the command or shortcut you wish to appear first in your menu.

For this example, I've chosen + > Commands > Network > Menu of Network Locations, and iKey Editor has opened the Network Locations List command dialog (**Figure 5.10**). There aren't any options; the command just displays a menu of network locations to choose from and switches your Network preference pane's settings to the chosen one. Click OK when you're done.

**FIGURE 5.10**



Configure the command you've chosen, if necessary.

**TIP** If you put only a single command or shortcut in a menu that displays in the menu bar, iKey turns it into a button. In other words, you don't have to click the menu, wait for it to drop, and choose the item; you can merely click the menu's icon or name (whatever's showing) to execute the command or shortcut.

3. **Add more menu items:** Repeat Step 2 to add additional items to the menu. For this example, arrange the menu creation dialog so you can see the main window, click the Shortcuts tab, and drag the shortcut you just created in the [Create a Shortcut](#) steps previously into the Menu Items list. Similarly, if you wanted to add another menu (which would then appear as a submenu), you'd drag it in from the list of menus in the main window.

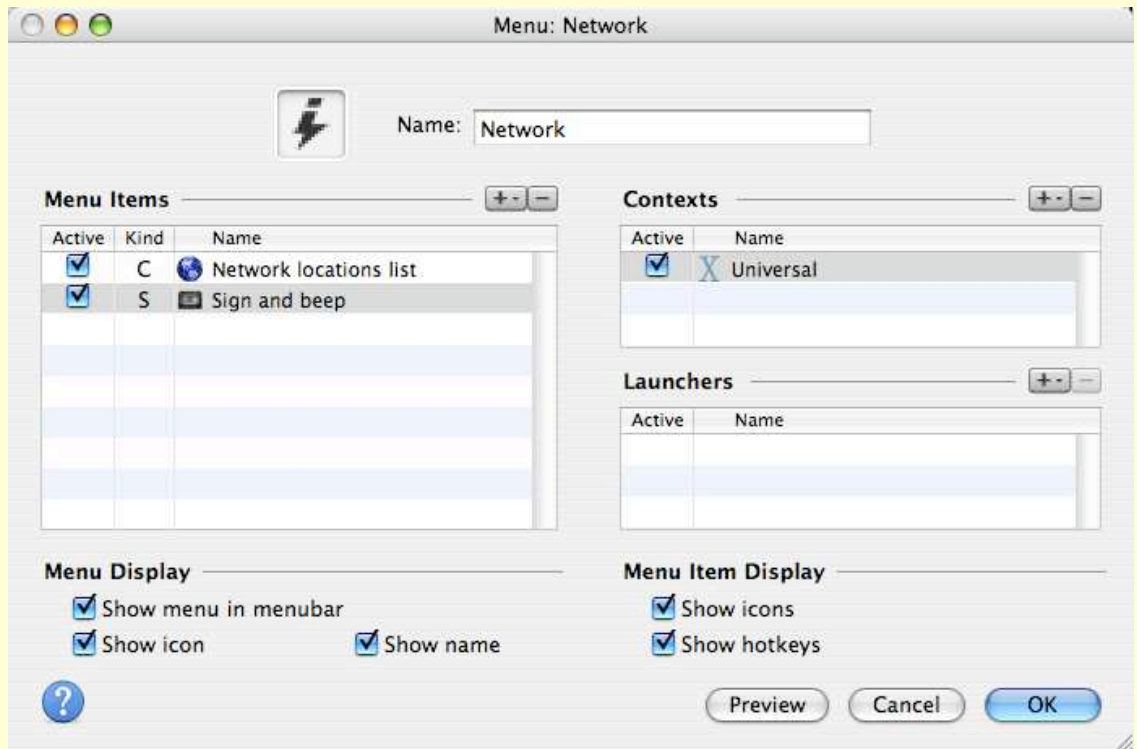
**TIP** If your menu has both related and unrelated items, consider adding a separator (choose + > Separator) to set them apart and make the menu easier to use.

**NOTE** If you add a shortcut to a menu, but the shortcut is deactivated (either manually or because its context makes it inactive), the item will appear dimmed in the menu and you won't be able to choose it.

4. **Set a context for your menu:** Although we could set a context so this menu appears in only a specific application, let's stick with the Universal context so it's always available.
5. **Set a launcher for the menu:** In this case, there's no need for a launcher since it appears in the menu bar automatically.
6. **Name and configure the menu:** Enter a name in the Name field, and use the checkboxes in the Menu Display and Menu Item Display areas to set how your menu displays. In **Figure 5.11** I've set the menu to display in the menu bar, with both its icon and its name, and I have all items show their icons and hotkeys, if any. You can click Preview to see how its contents will look, though other system-wide contextual menu items show up in the Preview menu that won't appear in the real thing.

**TIP** You can drag a picture into the icon well next to the name to change the icon, or you can Control-click a shortcut or command in the Menu Items list and choose Set Image to Menu. Since iKey's default menu is just a normal iKey menu, this means you can change its icon to whatever you like.



**FIGURE 5.11**

Configure the display options for your menu.

7. **Save your menu:** Click OK to save your menu.
8. **Save and try your menu:** Choose File > Save and Activate (Command-S) to make your new menu ready to use, and then try using your new menu, which should appear in the menu bar. If it doesn't, return to iKey Editor, double-click the menu to edit it, and make the necessary changes before saving and activating again.

## Create a Palette

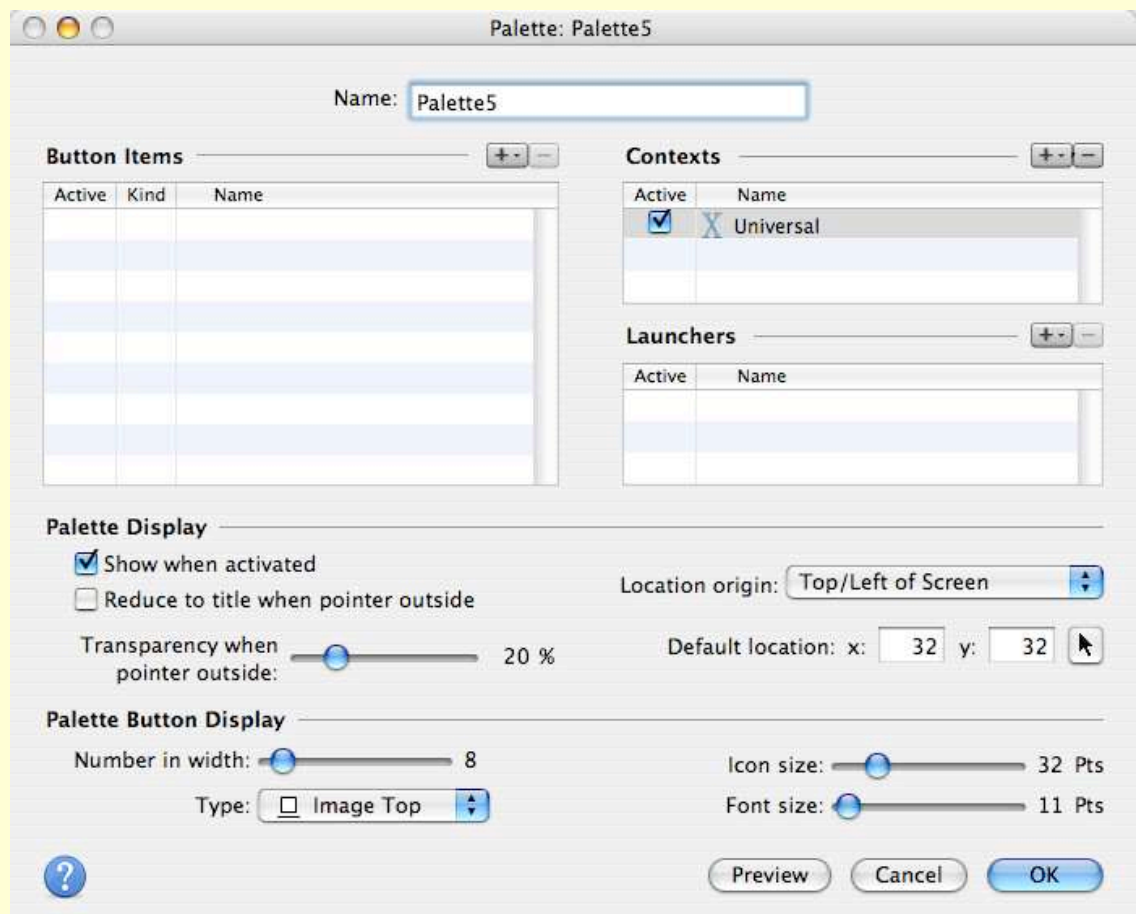
Follow these steps to create a palette in iKey Editor. Once again, the details will vary depending on what you put in your palette, but the basic steps are the same. For the purposes of this example, I assume in these steps that you've created only the commands, launchers, and contexts from the [Create a Shortcut](#) and [Create a Menu](#) steps.

**NOTE** For this example, I show you how to create a palette that contains buttons that, when clicked, type various Unix commands into a Terminal window. The palette appears only when Terminal is the front application.

1. **Open the palette creation dialog:** With the Palettes tab selected, click the Create button (+) in iKey Editor's toolbar, or choose Create > Palette.

iKey Editor opens an empty palette creation dialog (**Figure 5.12**).

**FIGURE 5.12**

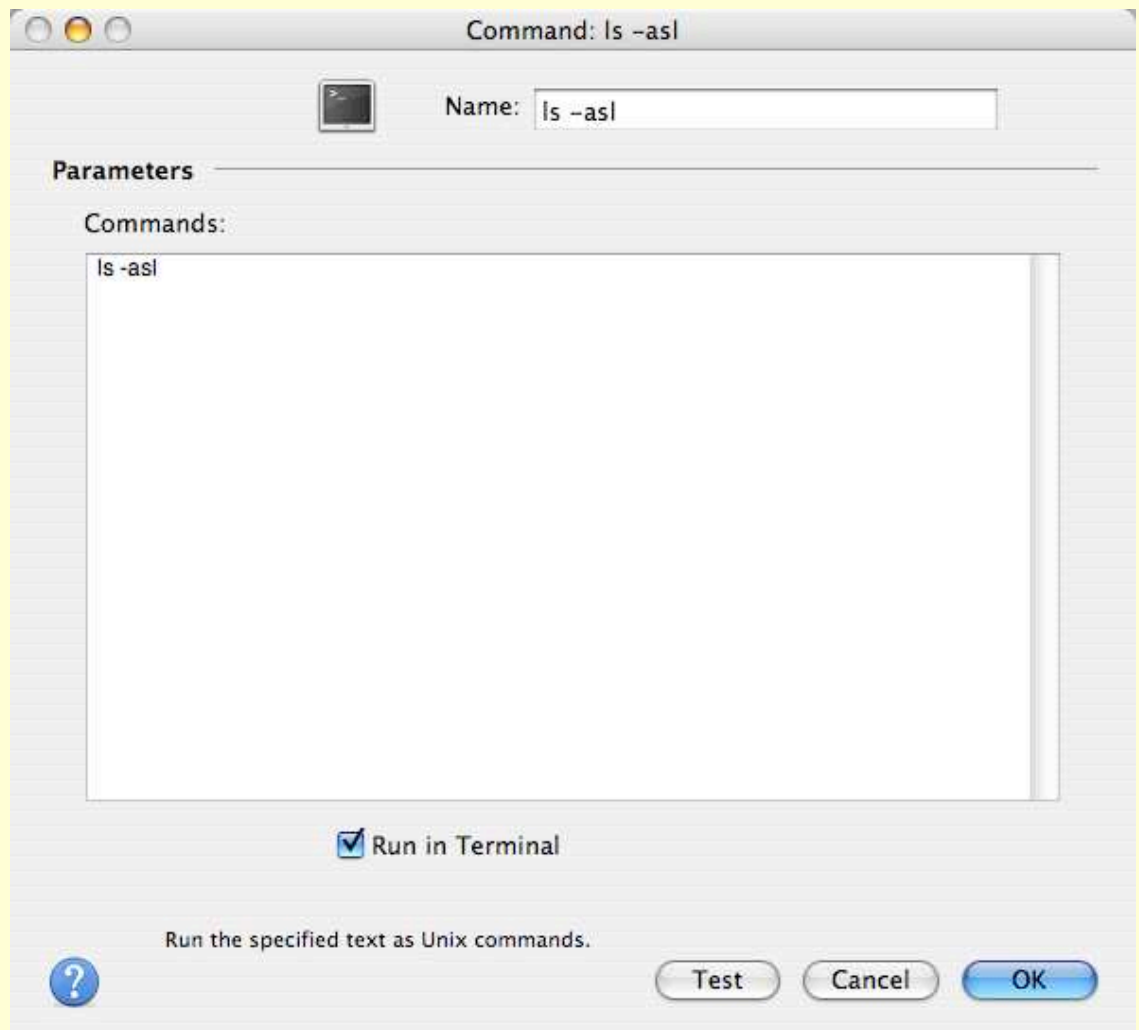


An empty palette creation dialog.

2. **Add the first item:** Using the + pop-up menu in the Button Items area, navigate through the hierarchical menus to choose the command or shortcut that you wish to place at the upper left of your palette.

For this example, I've chosen + > Commands > Unix > Run Commands from Text, and iKey Editor has opened the Unix Command dialog (**Figure 5.13**). Enter a Unix command that you don't like typing, like `ls -as1` to list all files. Also check Run in Terminal so that output will be displayed in Terminal's window. Click OK when you're done.

**FIGURE 5.13**



Configure the command you've chosen.

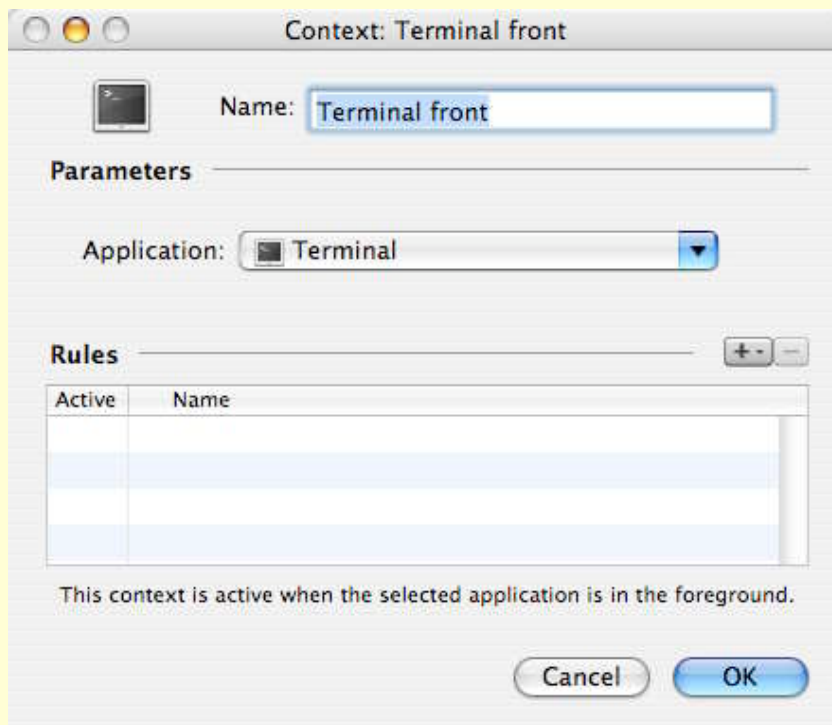
3. **Add more items:** Repeat Step 2 to populate your palette with additional buttons. As you'll see at the end, the other Unix commands I've used are **top -u** and **ps -aux**.

**NOTE** If you want to add an already-defined command to your palette, drag it from the Commands pane of the Library window to the Button Items area in the palette creation dialog. Similarly, to add an existing shortcut, drag it from the Shortcuts pane of the main window into the palette creation dialog.

**NOTE** If you add a shortcut to a palette, but the shortcut is deactivated (either manually or because its context requires that it be inactive), the button will appear dimmed in the palette and you won't be able to click it.

4. **Set the context:** Since this palette is pointless unless you're working at the command line already, let's create an application-specific context to make the menu appear only when you're in Terminal:
  - a. In the Contexts list area, choose + > Application in Front to open the Application in Front context dialog, where I've chosen Terminal from the Application menu (**Figure 5.14**).

**FIGURE 5.14**

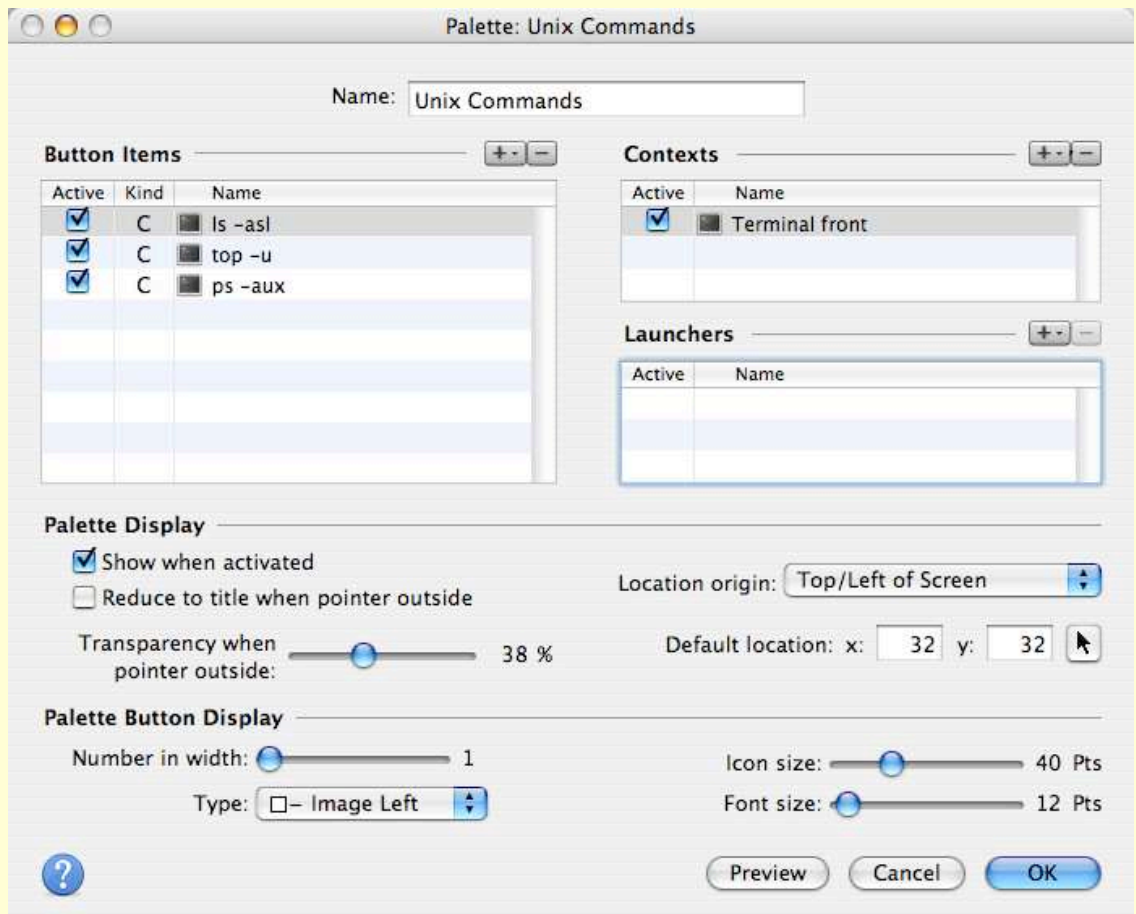


Add a context for your palette.

- b. Click OK.
  - c. Select the Universal context and either click the — button or press Command-Delete to remove it.
5. **Set a launcher:** There's no need for a launcher for this palette since it appears automatically whenever you switch to Terminal, thanks to the context we assigned.

6. **Name and configure your palette:** Enter a name in the Name field, and use the checkboxes in the Palette Display and Palette Button Display areas to set how your palette displays (**Figure 5.15**).

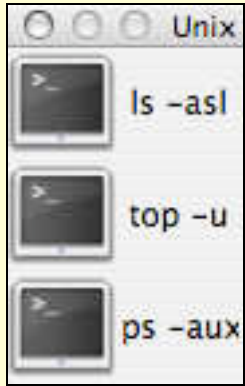
**FIGURE 5.15**



Configure the display options for your palette using the controls in the lower portion of the dialog.

7. **Preview your palette:** Click Preview to see how the palette will look; iKey warns you that the buttons aren't functional in preview mode (**Figure 5.16**). Close the palette when you're done previewing it, and make any desired changes. When you're done, click OK.

**FIGURE 5.16**



Preview your palette to make sure you like how the display settings work out.

**8. Save and try the palette:** Choose File > Save and Activate (Command-S) to make your new palette ready to use, and then try switching to Terminal to see if your palette appears as you want. If it doesn't, return to iKey Editor, double-click the palette to edit it, and make the necessary changes before saving and activating again. To invoke any of the commands or shortcuts in the palette, just click them, as you would any normal palette in another application.

## Find a Particular Shortcut, Menu, or Palette

Once you have created a number of shortcuts, menus, and palettes, you may find it difficult to locate a specific one for editing. iKey Editor provides two ways to find particular items: searching by name and viewing by context.

**TIP** You can, of course, click a column header to sort the list of shortcuts, menus, or palettes by that particular column, which can make finding an item easier.

## Use the Search field

The easiest way to find a shortcut, menu, or palette, assuming you have some idea of what it's called, is to click the appropriate tab (Shortcut, Menu, or Palette) and type a few characters from the name in the Search field (on the toolbar; pressing Command-F moves the insertion point to the Search field). In **Figure 5.17**, I've typed "Launch" to show only those shortcuts I use to launch applications. iKey Editor immediately limits the items showing in the list to those whose names contain the characters you typed.

**NOTE** To view the full list again, click the little X button in the Search field or delete the characters you typed.

**FIGURE 5.17**



Enter a few characters in the Search field to limit the list of shortcuts, menus, or palettes.

### Limit the view by context

If you have no idea what a shortcut, menu, or palette might be called, but you know what context it uses, you can use the Contexts pane to narrow the list as well; just click a context in the Contexts pane to accomplish this task (**Figure 5.18**).

**FIGURE 5.18**



Select a context, and the main window list narrows to show just those that include the selected part. Here, I've selected the Eudora front context, which is used by four shortcuts.



**TIP** As with searches hanging around after you are done with them, if the list of shortcuts, menus, or palettes has been limited by the selection of a context, select All Contexts from the top of the Contexts pane to show everything.

**TIP** Limiting the list using the Search field and by selecting a context can work simultaneously; this can be powerful, but pay attention so you don't get confused. For instance, in **Figure 5.18**, if I'd left the word "Launch" in the Search field and selected the context that has Eudora in the front, nothing would appear. Clearing "Launch" from the Search field would reveal the shortcuts that use the Eudora context.

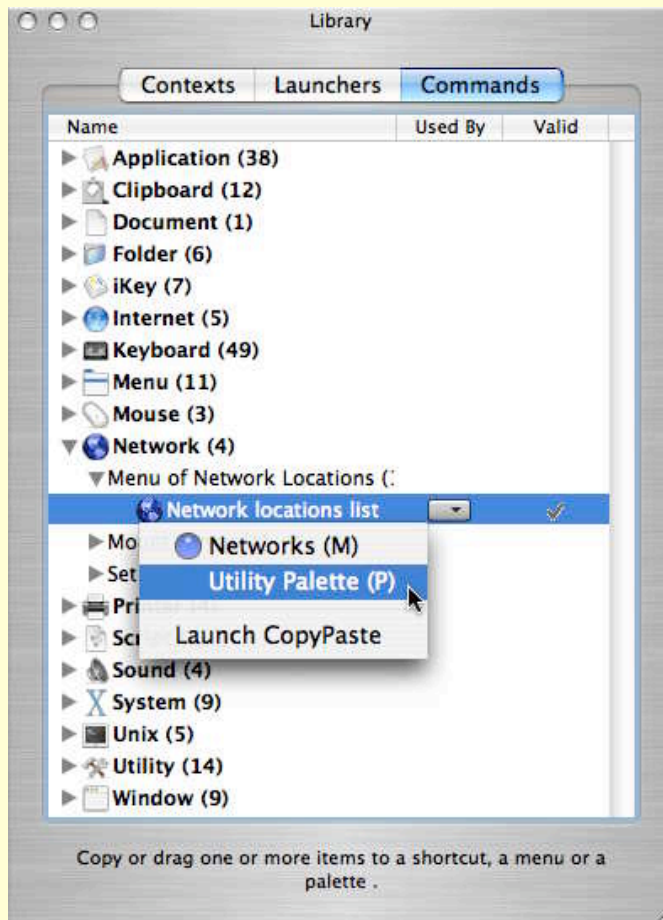
### Select with Used By menu

The final way you can find shortcuts, menus, and palettes is if you remember which commands or launchers you used in creating them. The Used By column in each pane in the Library window provides a pop-up menu that lists each shortcut, menu, and palette that uses the selected command or launcher. Choose a listed shortcut, menu, or palette from that pop-up menu (the name has an S, M, or P following it to help you differentiate between similar names), and iKey Editor opens it.

For instance, I've used a Network Locations List command in a menu and a palette, and both are listed in the Used By pop-up menu in **Figure 5.19**. Choosing one opens it directly.



**FIGURE 5.19**



Choose an item from the Used By pop-up menu in the Library window to open the associated shortcut, menu, or palette.

## Work with Shortcuts in iKey Editor

iKey Editor's interface is quite flexible, which is good, but it also can lead to confusion if you don't realize all the possibilities. Here's a quick list of all the things you can do with shortcuts within iKey Editor.

### To create shortcuts:

- With the Shortcuts pane showing in the main window, click the Create (+) button.
- Choose a command from the hierarchical Create > Shortcut menu to create a shortcut with that command.
- Drag one or more commands from the Commands pane of the Library window into the main window (with the Shortcuts pane showing) such that you get a thick black line in between two existing shortcuts. This action creates a shortcut with the dragged commands.

- Select one or more commands in the Commands pane of the Library window, Control-click them, and choose New Shortcut with Selection from the contextual menu that appears.
- With the Shortcuts pane showing in the main window, Control-click anywhere in the window and choose Create Shortcut.
- Select a shortcut, and choose Edit > Duplicate (Command-D) to make an exact copy that you can then modify.

**NOTE** Shortcuts are automatically created with the context that's currently selected in the Contexts pane.

### **To edit shortcuts:**

- Double-click a shortcut in the main window, or in the Menu Items or Button Items areas of a menu or palette creation dialog.
- Select one or more shortcuts in the main window and click the Edit button (pencil).
- Control-click a shortcut or group of selected shortcuts and choose Edit from the contextual menu.

### **To delete shortcuts:**

- Select one or more shortcuts in the main window and click the Delete (—) button.
- Select one or more shortcuts in the main window and choose Edit > Clear, or more commonly, press Command-Delete.
- Control-click a shortcut or group of selected shortcuts and choose Delete from the contextual menu.

### **To add shortcuts to a menu or palette:**

- Drag one or more shortcuts from the main window to the Menu Items or Button Items area in the menu or palette creation dialog.
- Select one or more shortcuts, choose Edit > Copy (Command-C), switch to the menu or palette creation dialog, click the spot in the Menu Items or Button Items list where you want the shortcut(s) to be placed, and choose Edit > Paste (Command-V).

### **To export shortcuts:**

- Select one or more shortcuts in the main window, and choose File > Export.
- Select one or more shortcuts, Control-click them, and choose Export from the contextual menu that appears.

## **Work with Menus in iKey Editor**

Here's a quick list of all the things you can do with menus within iKey Editor.

### **To create menus:**

- With the Menu pane showing in the main window, click the Create (+) button.
- Choose Create > Menu.
- Drag one or more commands from the Commands pane of the Library window into the main window (with the Menus pane showing) such that you get a thick black line in between two existing menus. This action creates a menu with the dragged commands.
- Select one or more commands in the Commands pane of the Library window, Control-click them, and choose New Menu with Selection from the contextual menu that appears.
- With the Menu pane showing in the main window, Control-click anywhere in the window and choose Create Menu.
- Select a menu, and choose Edit > Duplicate (Command-D) to make an exact copy that you can then modify.

**NOTE** Menus are automatically created with the context that's currently selected in the Contexts pane.

### **To edit menus:**

- Double-click a menu in the main window, or in the Menu Items area of a menu creation dialog.
- Select one or more menus in the main window and click the Edit button (pencil).

- Control-click a menu or group of selected menus and choose Edit from the contextual menu.

#### **To delete menus:**

- Select one or more menus in the main window and click the Delete (—) button.
- Select one or more menus in the main window and choose Edit > Clear, or more commonly, press Command-Delete.
- Control-click a menu or group of selected menus and choose Delete from the contextual menu.

**WARNING!** The above three actions also delete any commands and launchers that are associated with the menu.

- Control-click a menu or group of selected menus and choose Delete Only Structure from the contextual menu to delete the menu(s) without deleting the associated commands or launchers.

#### **To add menus to another menu or to a palette:**

- Drag one or more menus from the main window to the Menu Items or Button Items area in the menu or palette creation dialog.
- Select one or more menus, choose Edit > Copy (Command-C), switch to the menu or palette creation dialog, click the spot in the Menu Items or Button Items list where you want the menu(s) to be placed, and choose Edit > Paste (Command-V).

#### **To export menus:**

- Select one or more menus in the main window, and choose File > Export.
- Select one or more menus, Control-click them, and choose Export from the contextual menu that appears.

## **Work with Palettes in iKey Editor**

Here's a quick list of all the things you can do with palettes within iKey Editor.

#### **To create palettes:**

- With the Palettes pane showing in the main window, click the Create (+) button.

- Choose Create > Palette.
- Drag one or more commands from the Commands pane of the Library window into the main window (with the Palettes pane showing) such that you get a thick black line in between two existing palettes. This action creates a palette with the dragged commands.
- Select one or more commands in the Commands pane of the Library window, Control-click them, and choose New Palette with Selection from the contextual menu that appears.
- With the Palettes pane showing in the main window, Control-click anywhere in the window and choose Create Palette.
- Select a palette, and choose Edit > Duplicate (Command-D) to make an exact copy that you can then modify.

**NOTE** Palettes are automatically created with the context that's currently selected in the Contexts pane.

#### **To edit palettes:**

- Double-click a palette in the main window.
- Select one or more palettes in the main window and click the Edit button (pencil).
- Control-click a shortcut or group of selected palettes and choose Edit from the contextual menu.

#### **To delete palettes:**

- Select one or more palettes in the main window and click the Delete (—) button.
- Select one or more palettes in the main window and choose Edit > Clear, or more commonly, press Command-Delete.
- Control-click a palette or group of selected palettes and choose Delete from the contextual menu.

**WARNING!** The above three actions also delete any commands and launchers that are associated with the palette.

- Control-click a palette or group of selected palettes and choose Delete Only Structure from the contextual menu to delete the palette(s) without deleting the associated commands or launchers.

#### **To export palettes:**

- Select one or more palettes in the main window, and choose File > Export.
- Select one or more palettes, Control-click them, and choose Export from the contextual menu that appears.

### **Work with Contexts in iKey Editor**

The actions you can perform with contexts are a bit different, not just from shortcuts, menus, and palettes, but also from launchers and commands. Here's a quick list of what's possible.

#### **To create contexts:**

- Click the small + button below the Contexts pane in the main window, and choose the desired context type from the pop-up menu.
- In the Contexts area of a shortcut, menu, or palette creation dialog, click the little + button, and choose the desired context type from the pop-up menu.

#### **To edit contexts:**

- Double-click a context in the Contexts pane or in the Contexts pane of the Library window.
- Control-click a context or group of selected contexts in the Contexts pane or in the Contexts pane of the Library window, and choose Edit from the contextual menu.

#### **To associate contexts with shortcuts, menus, and palettes:**

- Drag a context from the Contexts pane of the Library window or the Contexts pane either to a shortcut, menu, or palette in the main window, or to the Contexts area in a shortcut, menu, or palette creation dialog.
- Drag a shortcut, menu, or palette to a particular context in the Contexts pane. Note that this action replaces whatever context was previously associated with the shortcut, menu, or palette that you dragged.

- Select one or more contexts in the Contexts pane of the Library window, choose Edit > Copy (Command-C) (or Control-click them and choose Copy from the contextual menu), switch to the creation dialog, select the Contexts list, and choose Edit > Paste (Command-V).
- Select a context in the Contexts pane and then create a shortcut, menu, or palette.

**To remove contexts from a shortcut, menu, or palette:**

- Select one or more contexts in the Contexts area of a creation dialog, and click the — button in that area.
- Select one or more contexts in the Contexts area of a creation dialog, and choose Edit > Clear, or more commonly, press Command-Delete.
- Control-click one or more contexts in the Contexts area of a creation dialog, and choose Remove from the contextual menu.

**TIP** Removing a context from a particular shortcut, menu, or palette in this way does not delete it entirely, so it remains in other shortcuts, menus, and palettes.

**To delete contexts:**

- Select one or more contexts, either in the Contexts pane or in the Contexts pane of the Library window, and choose Edit > Clear, or more commonly, press Command-Delete.
- Control-click one or more contexts, either in the Contexts pane or in the Contexts pane of the Library window, and choose Delete from the contextual menu.

**NOTE** iKey Editor warns you if a context to be deleted is used by existing shortcuts, menus, or palettes. You can see, in the Used By column of the Contexts pane of the Library window, whether or not a context is in use.

## Work with Commands in iKey Editor

Here's a quick list of what you can do with commands in iKey Editor.

### To create new commands:

- In the Commands, Menu Items, or Button Items area of a shortcut, menu, or palette creation dialog, click the little + button and choose the desired command type from the pop-up menu.
- From the hierarchical Create > Shortcut menu, choose the desired command type to create a shortcut that uses a new command of the chosen type.
- Select a command in the Commands pane of the Library window and either choose Edit > Duplicate or Control-click it and choose Duplicate from the contextual menu. This action creates a copy of the selected command for you to modify.

### To edit existing commands:

- Double-click a command in the Commands pane of the Library window.
- Control-click a command or group of selected commands in the Commands pane of the Library window, and choose Edit from the contextual menu.

### To associate already-defined commands with shortcuts, menus, and palettes:

- Drag an already-defined command from the Commands pane of the Library window either to a shortcut, menu, or palette in the main window, or to the Commands, Menu Items, or Button Items area in a shortcut, menu, or palette creation dialog.
- Drag one or more commands from the Commands pane of the Library window into the main window such that you get a thick black line in between two existing shortcuts, menus, or palettes. This action creates a shortcut, menu, or palette with the dragged commands.
- Select one or more already-defined commands in the Commands pane of the Library window; choose Edit > Copy (Command-C) (or Control-click them) and choose Copy from the contextual menu;



switch to the creation dialog; select the Commands, Menu Items, or Button Items list; and choose Edit > Paste (Command-V).

**To remove commands from a shortcut, menu, or palette:**

- Select one or more commands in the Commands, Menu Items, or Button Items area of a creation dialog, and click the — button in that area.
- Select one or more commands in the Commands, Menu Items, or Button Items area of a creation dialog, and choose Edit > Clear, or more commonly, press Command-Delete.
- Control-click one or more commands in the Commands, Menu Items, or Button Items area of a creation dialog, and choose Remove from the contextual menu.

**TIP** Removing a command from a particular shortcut, menu, or palette in this way does not delete it entirely, so it remains in other shortcuts, menus, and palettes, as well as in the Commands pane of the Library window.

**To delete commands:**

- Select one or more commands in the Commands pane of the Library window, and choose Edit > Clear, or more commonly, press Command-Delete.
- Control-click one or more commands in the Commands pane of the Library window, and choose Delete from the contextual menu.

**NOTE** iKey Editor warns you if a command to be deleted is used by existing shortcuts, menus, or palettes.

## Work with Launchers in iKey Editor

Here's a quick list of what you can do with launchers in iKey Editor.

### To create new launchers:

- In the Launchers area of a shortcut, menu, or palette creation dialog, click the little + button and choose the desired launcher type from the pop-up menu.

### To edit existing launchers:

- Double-click a launcher in the Launchers pane of the Library window.
- Control-click a launcher or group of selected launchers in the Launchers pane of the Library window, and choose Edit from the contextual menu.

### To associate already-defined launchers with shortcuts, menus, and palettes:

- Drag an already-defined launcher from the Launchers pane of the Library window either to a shortcut, menu, or palette in the main window, or to the Launchers area in a shortcut, menu, or palette creation dialog.
- Select one or more already-defined launchers in the Launchers pane of the Library window, choose Edit > Copy (Command-C) (or Control-click them and choose Copy from the contextual menu), switch to the creation dialog, select the Launchers list, and choose Edit > Paste (Command-V).

### To remove launchers from a shortcut, menu, or palette:

- Select one or more launchers in the Launchers area of a creation dialog, and click the — button in that area.
- Select one or more launchers in the Launchers area of a creation dialog, and choose Edit > Clear, or more commonly, press Command-Delete.
- Control-click one or more launchers in the Launchers area of a creation dialog, and choose Remove from the contextual menu.

**TIP** Removing a launcher from a particular shortcut, menu, or palette in this way does not delete it entirely, so it remains in other shortcuts, menus, and palettes, as well as in the Launchers pane of the Library window.

**To delete launchers:**

- Select one or more launchers in the Launchers pane of the Library window, and choose Edit > Clear, or more commonly, press Command-Delete.
- Control-click one or more launchers in the Launchers pane of the Library window, and choose Delete from the contextual menu.

**NOTE** iKey Editor warns you if a launcher to be deleted is used by existing shortcuts, menus, or palettes.

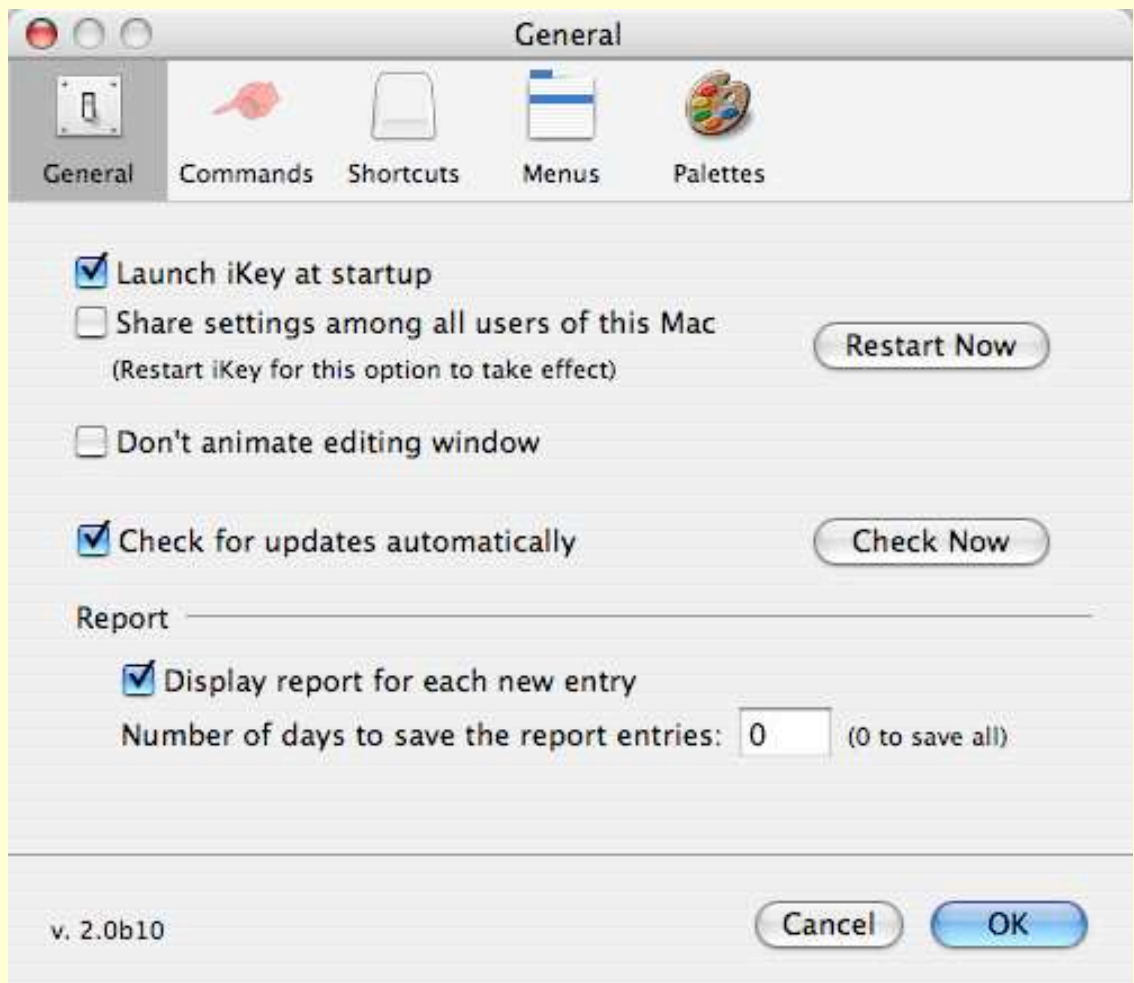
## PREFERENCE REFERENCE

To access iKey 2's preferences, open iKey Editor and choose Preferences (Command-,) from the iKey Editor application menu.

### General

Use the General preference pane to control overall application behavior (**Figure 6.1**).

**FIGURE 6.1**



General preference pane.

It contains a number of options:

- **Launch iKey at Startup** tells iKey to add itself to the Startup Items list in Mac OS X's Accounts preference pane. In almost all situations, you should leave this checkbox selected so iKey Engine is always running.

- **Share Settings Among All Users of This Mac** makes iKey's preferences and shortcuts apply to all users, rather than just the current user. You must restart iKey Engine for this setting to take effect; click Restart Now to do so.
- **Don't Animate Editing Window**, if checked, causes iKey Editor to forgo the window opening and closing animation when working with various windows. Selecting this option may make iKey Editor feel faster.
- **Check for Updates Automatically** causes iKey to connect to Script Software's server to see if updates are available and enables you to download an update if one has been released. In general, I recommend leaving this checkbox selected unless you have an intermittently connected, dial-up Internet account.
- **Display Report for Each New Item** opens the iKey Report window whenever a command you've configured to add to the report executes. You can also set how long to save the report entries.

## Commands

Use the Commands preference pane to set the defaults for commands you create (**Figure 6.2**).

**FIGURE 6.2**



Commands preference pane.

Two options are available:

- **Suggest Name when Commands Are Created** makes iKey Editor generate a likely name for each command based on how you configure the command. There's no reason to turn this setting off, unless you often override iKey's suggestions.
- **Suggest Name when Command Parameters Are Modified** causes iKey Editor to change a command's name when you change its parameters. For instance, if you create a command to open a Web address, and then later change the URL, this setting causes iKey Editor to update the command's name to the new URL.

## Shortcuts

Use the Shortcuts preference pane to set the defaults for shortcuts you create (**Figure 6.3**).

**FIGURE 6.3**



Shortcuts preference pane.

Two options are available:

- **Shortcut Default Repetition Parameters** controls the default parameters for the number of times a new shortcut should run and the delay between executions if it should run multiple times. Unless you anticipate creating a lot of shortcuts that you want executing numerous times in a row, I recommend leaving these settings alone. To adjust the number of executions and the delay between them, click the Modify button.
- **Command Default Repetition and Result Parameters** similarly controls the default parameters for the number of times a new command should run, the delay between executions if it

should run multiple times, how commands should act if the action they embody succeeds or fails, and how long to wait to determine if success or failure has occurred. Again, leave these settings alone unless you have a specific use in mind. To adjust these settings, click the Modify button.

- **Create Copies of Commands When Duplicating a Shortcut**, when selected, causes iKey to duplicate commands used in a shortcut you're duplicating. Normally, this wouldn't make sense, since there's no reason to have multiple, identical commands defined. However, if you regularly want to duplicate a shortcut and modify its commands slightly (which is likely with Type Text commands), you'll find this option a useful time-saver.

## Menus

Use the Menus preference pane to set the display defaults for menus you create (**Figure 6.4**).

**FIGURE 6.4**



Menus preference pane.

Two sets of options are available:



- **Menu Display:** The first set of checkboxes control whether newly created menus will display in the menu bar, and if so, if they'll display their icon, their name, or both.
- **Menu Item Display:** The second set of checkboxes set whether items in the menu will display the icon associated with the type of command and if they'll show their associated keyboard shortcut.

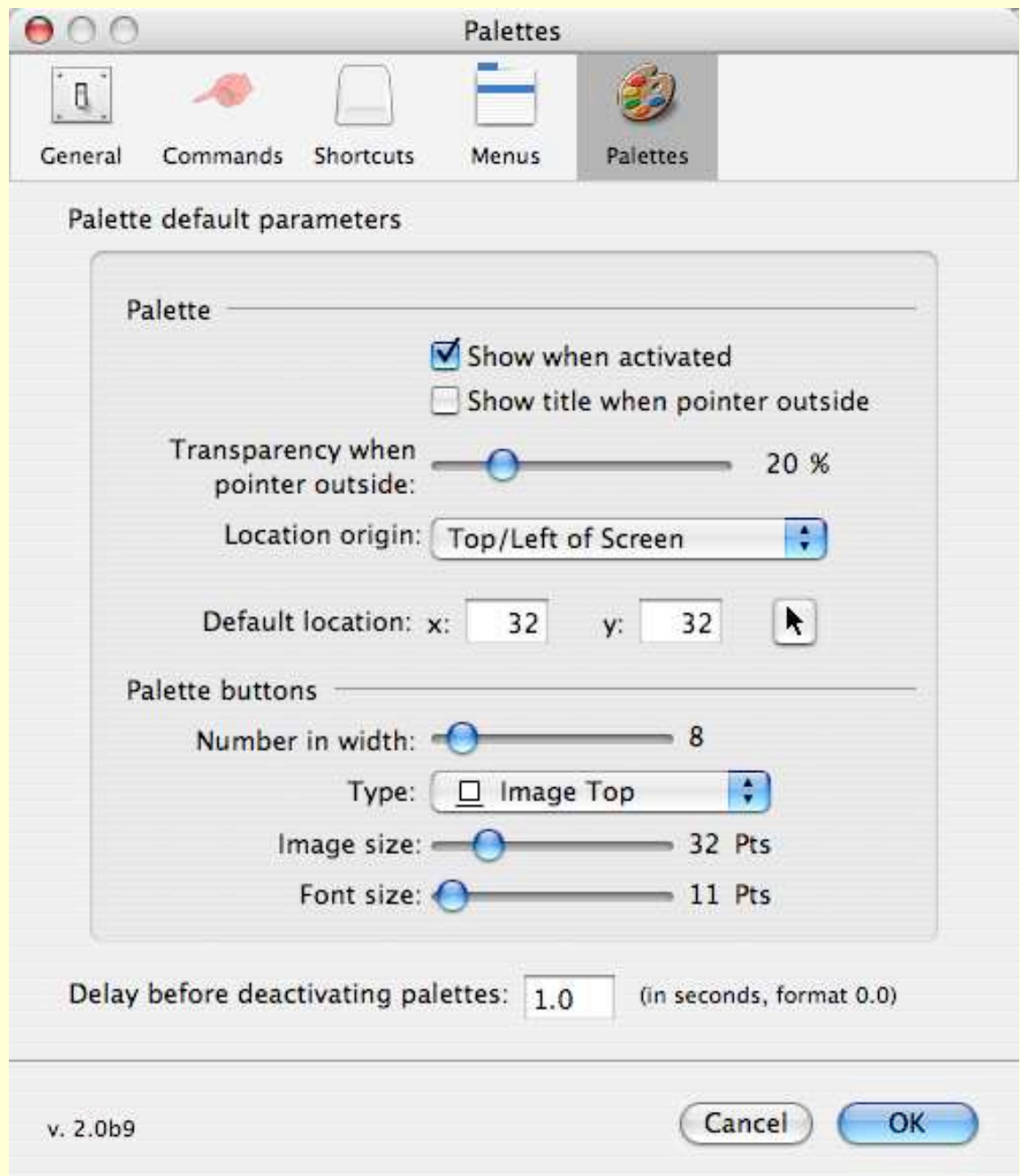
## Palettes

Use the Palettes preference pane to set the default display options for palettes you create (**Figure 6.5**). Of course, these are just the default options that you want to apply in general; you can always override them for any individual palette.

Several options are available; here's how I suggest you work with these preferences:

- **Delay Before Deactivating Palettes:** You can enter a number of seconds in this field (at the bottom) to speed up or slow down the length of time before a palette becomes transparent after the pointer moves outside the palette. Note that you must enter the number of seconds with a decimal place, even if you want an even number of seconds.
- **Other options:** A variety of other options are available in this preference pane, and you should definitely adjust them to your liking. However, you'll probably have trouble knowing just how you want to change these settings until you've created and modified a few palettes, so I recommend sticking with the defaults until then. After that point, if you decide you prefer your palettes to be more transparent, or to use larger buttons or text, come back to this preference pane and make the appropriate changes.

**FIGURE 6.5**



Palettes preference pane.

## MENU REFERENCE

For the most part, work in iKey takes place within the iKey Editor window, using the pop-up menus and other controls there. However, other functions are available in iKey Editor's menus. For each of these menus, I explain the action you can use each item to perform.

### iKey Editor

The iKey Editor application menu offers the standard commands found in any application menu (**Figure 7.1**).

**FIGURE 7.1**



The iKey application menu.

- **About iKey Editor:** Displays the iKey version number, a list of credits, and links to the iKey tech support forum, along with a contact email address.

**NOTE** Be sure to check the version number of the copy of iKey you're using before asking for support. Also, it's usually a good idea to make sure your problem isn't solved by downloading and using a later version.

- **Preferences:** Opens iKey's preferences window. See [Preference Reference](#).
- **Services:** This hierarchical menu provides access to services provided by Mac OS X and other applications. For the most part, they aren't particularly relevant when working in iKey Editor.
- **Hide iKey Editor:** Hides the iKey Editor windows. You can also Option-click on another application to switch to that application and hide iKey Editor at the same time.

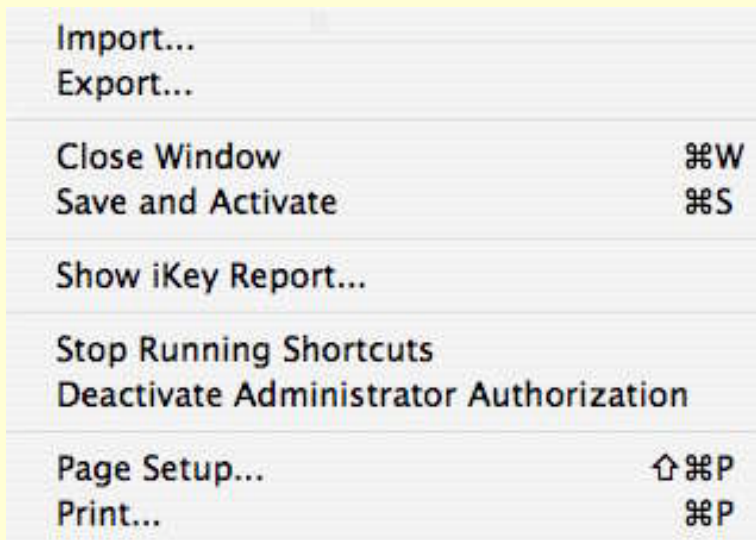
- **Hide Others:** Hides the windows of all programs other than iKey Editor.
- **Show All:** Reveals all running programs, should any be hidden.
- **Quit iKey Editor:** Quits the iKey Editor application, but does not quit iKey Engine, so you can use your shortcuts, menus, and palettes. Choose this menu item when you're done creating and editing shortcuts.

**TIP** Although there aren't many reasons for quitting iKey Engine (at which point all shortcuts, menus, and palettes stop working), you can do so by double-clicking the iKey icon in the Finder and clicking the Quit button when prompted. If you want to quit iKey Engine often, make a [Quit iKey Engine](#) command and store it in a menu.

## File

Use the items in the File menu for working with iKey Editor in general (**Figure 7.2**):

**FIGURE 7.2**



File menu.

- **Import:** Imports property list files exported from iKey.
- **Export:** Use the Export menu item to export a property list file containing the selected shortcuts, menus, or palettes. Exported files are a handy way to share your shortcuts, menus, and palettes with other iKey users.

**TIP** The exported file is just a text file, so you could open it in a text editor and make changes, but be careful, since you could easily make a change that would confuse or even crash iKey upon import.

- **Close Window:** Closes the currently active window; note that iKey Editor has only two windows: the main window and the Library window, everything else is a dialog and doesn't respond to Close Window.
- **Save and Activate:** Although your changes are automatically saved when you quit iKey Editor, this command saves any shortcuts you've created or edited, if you want to save them without quitting. It's important to save not just to avoid losing your work, but also because the Save and Activate command communicates your changes to iKey Engine so they're available right away in any application.
- **Show iKey Report:** Displays the iKey Report dialog, which lists messages generated by the Result Parameters of specific commands.
- **Activate/Deactivate Hotkeys:** This item enables you to deactivate hotkeys while you're creating and editing shortcuts in iKey Editor. It does not affect general usage of hotkeys when you're in any other application.

**TIP** iKey 2.0 had a pair of commands—Show Invalid Parts and Show Unused Parts—that have been removed from the File menu in iKey 2.1.

Now, to see invalid or unused parts, simply look in the Library at the Used By column and, for commands, the Valid column. Any parts that lack a pop-up menu in the Used By column are unused, and any commands that lack a checkmark in the Valid column are invalid. To remove such unnecessary parts, select them (remember that you can Command-click to select multiple parts at once), and press Command-Delete (or Control-click the selected parts and choose Delete from the contextual menu that appears).

**TIP** Another command has disappeared from the File menu in iKey 2.1: Activate/Deactivate Hotkeys. It was intended to enable you to turn off hotkeys while creating and editing keyboard event launchers in iKey Editor. iKey 2.1 eliminates the need for it by automatically deactivating hotkeys when you're defining a keyboard event launcher.

- **Activate/Deactivate Administrator Authorization:** This item, which switches between Activate and Deactivate appropriately, enables you to run iKey with administrator authorization or to remove that authorization. Unless you have a good reason not to, I suggest that you activate iKey administrator authorization.

**TIP** At the moment, iKey uses this authorization only for the [Set Network Location](#) command and the wake up option for the [Sleep](#) command.

- **Page Setup:** Opens the standard Page Setup dialog for you to adjust print settings. You may wish to switch the page orientation to landscape before printing.
- **Print:** Opens the standard Print dialog so you can print a list of your shortcuts, menus, or palettes—essentially whatever appears in the main list in the iKey Editor window.

**TIP** Since a printout from iKey Editor includes the hotkeys used as launchers, printing could be a useful way to create a reference for yourself of all the hotkeys you've attached to different shortcuts.

## Edit

The Edit menu contains a number of familiar items along with a few that are specific to iKey Editor (**Figure 7.3**):

**FIGURE 7.3**



Edit menu.

- **Undo:** Undoes the last action. iKey Editor supports multiple levels of undo, so you can work back in time if necessary. The Undo item changes to reflect whatever action you just performed.
- **Redo:** Redoes whatever action you last reversed with Undo. Like Undo, Redo changes to reflect the action it will perform when chosen.
- **Cut:** Copies the selection to the clipboard and then deletes it. You can use Cut on any text that you type, as well as on commands, launchers, and contexts in the drawer. Cut also works in each of the separate panels of the shortcut, menu, and palette creation dialogs.
- **Copy:** Copies the selection to the clipboard without deleting it. This, of course, works with text, as well as with all types of objects. Also, you can copy in places (such as the main window and the drawer) that don't accept paste actions.



- **Paste:** Pastes the contents of the clipboard. Paste is available only where it makes sense, so you can paste text whenever you have the cursor in a text entry field, and you can paste objects only in the appropriate panel of the shortcut, menu, and palette creation dialog.
- **Clear:** Deletes the selected object. I particularly like using the Command-Delete hotkey for this menu item.
- **Duplicate:** Makes an identical copy of the selected object. Duplicate works in one place where Paste doesn't (the main window), making it a useful way to create a slightly different shortcut, menu, or palette without recreating the entire thing from scratch.
- **Edit Part:** Choosing Edit Part from the Edit menu performs exactly the same action as double-clicking the part in the main window or Library window; now that you know this fact, I can't see any reason why you would want to use this menu item.
- **Test/Preview:** This menu item, which changes depending on whether you have a shortcut or command (Test), or menu or palette (Preview) selected, executes the shortcut or command, or displays the menu or palette. You can think of Test/Preview as another type of launcher (see [Launcher Reference](#)).
- **Find:** Choosing Find from the Edit menu merely moves the cursor to the Search field in the main window; you can type a few characters there to restrict the list in the main window to matching items.
- **Special Characters:** Should you wish to include a character that you can't access from the keyboard (an unusual Unicode character, for instance), choosing Edit > Special Characters displays the Mac OS X Character Palette, from which you can drag the desired character. Inserting special characters will work only with a [Put Text in the Clipboard](#) command, not with a [Type Text](#) command.

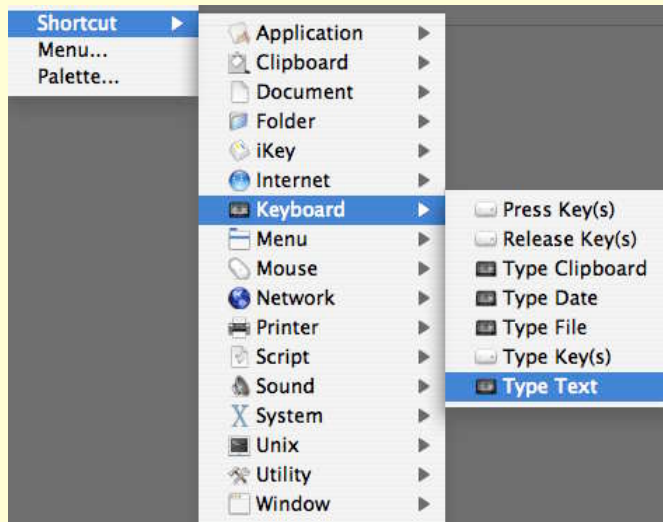
## Create

The Create menu contains items that duplicate the function of the + button in the main window for creating new shortcuts, menus, and palettes (**Figure 7.4**). You don't need to use the Create menu items unless you find a normal menu easier to use than the + buttons. The only caveat is that the hierarchical Shortcut menu is probably the fastest and easiest way to create a single-command shortcut. I won't



describe each of these items; refer to the [Usage Tutorial](#) and appropriate reference sections for details.

**FIGURE 7.4**

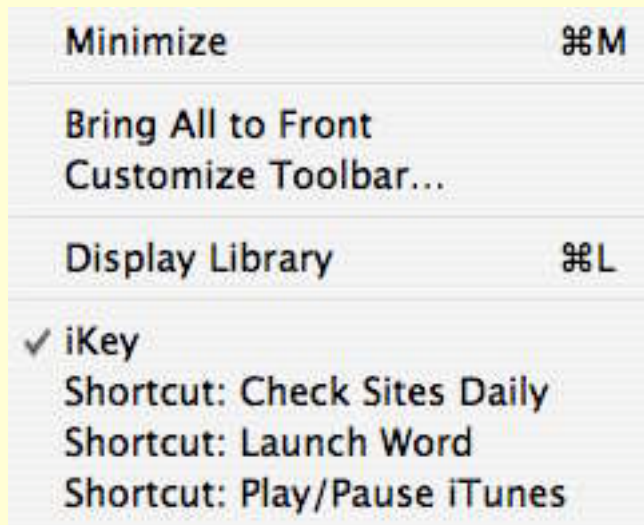


Create menu, showing the Shortcut > Keyboard submenu.

## Window

The Window menu items are essentially standard (**Figure 7.5**):

**FIGURE 7.5**



Window menu.

- **Minimize:** Shrinks the front window to an icon in the Dock. The Minimize menu item works the same as clicking the yellow minimize button in the upper left corner of the window.
- **Bring All to Front:** If you have multiple iKey Editor windows open, but some of them are behind the windows of other applications, Bring All to Front moves them in front of the other applica-

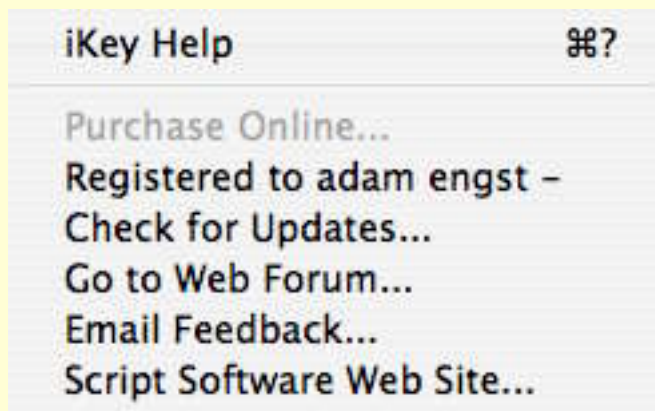
tion windows. Clicking the iKey Editor icon in the Dock has the same effect.

- **Customize Toolbar:** Choosing Customize Toolbar displays a sheet that lets you reposition toolbar controls, remove unwanted controls, and choose a display style from Icons, Text, and Icons & Text (plus the additional choice of a small size for the text).
- **Display Library:** Choose Display Library to open the Library window; you can also click the Library button in the main window.
- **Specific Windows:** The last item, or items if multiple windows are open, in the Window menu lets you bring a particular window to the front. The name of the menu item reflects the name of the window.

## Help

Turn to the Help menu for assistance with iKey (**Figure 7.6**):

**FIGURE 7.6**



Help menu.

- **iKey Help:** Opens this manual.

**TIP** Click the blue ? buttons scattered through iKey Editor's interface to be sent to the part of this manual that's appropriate to the dialog you're in.

- **Purchase Online:** Opens the Script Software orders page so you can purchase iKey.
- **Registered to:** This isn't exactly a menu item; choosing it does nothing. It simply shows the name of the person who owns this copy of iKey.

- **Check for Updates:** Choosing Check for Updates causes iKey to connect to the Script Software Web site and determine if it's up to date. If not, you're given the opportunity to download a new version. I recommend checking for updates if you have troubles to see if a newer version may fix your problem.
- **Go to Web Forum:** Opens the Script Software discussion forum Web page in your default Web browser. You can ask questions about iKey here; it's also a good place to see if others have already answered your question.
- **Email Feedback:** Creates a new email message in your default email program. The message is pre-addressed to **ikey-info@scriptsoftware.com** and includes the versions of iKey and Mac OS X that you're running. Please leave the version information intact when typing your message, since it adds context to the rest of your feedback.
- **Script Software Web Site:** Loads the Script Software home page in your default Web browser.

## HOTKEY REFERENCE

iKey Editor has a few default hotkeys for its menu items. These are not hotkeys that you make; they are built-in hotkeys for iKey Editor functions. Refer to **Table 8.1** for a full list.

**Table 8.1: iKey Editor Keyboard Shortcuts**

Keyboard Shortcut	Menu Item	Function
Command-,	iKey Editor > Preferences	Opens the Preferences window
Command-?	Help > iKey Help	Displays help information
Command-C	Edit > Copy	Copies the selected object
Command-D	Edit > Duplicate	Duplicates the selected item
Command-Del	Edit > Clear	Deletes the selected object
Command-Shift-E	Edit > Edit Part	Opens a dialog to edit the selected item
Command-F	Edit > Find	Moves the insertion point to the Search field in the toolbar
Command-H	iKey Editor > Hide iKey Editor	Hides iKey Editor
Command-L	Window > Display Library	Opens the Library window
Command-M	Window > Minimize	Shrinks the iKey Editor window to an icon in your Dock
Command-P	File > Print	Displays the Print dialog
Command-Shift-P	File > Page Setup	Displays the Page Setup dialog for changing paper size
Command-Q	iKey Editor > Quit iKey Editor	Quits iKey Editor

**Table 8.1: iKey Editor Keyboard Shortcuts (continued)**

<b>Keyboard Shortcut</b>	<b>Menu Item</b>	<b>Function</b>
Command-S	File > Save and Activate	Saves your changes and makes them available
Command-V	Edit > Paste	Pastes the contents of the clipboard
Command-W	File > Close Window	Closes the frontmost window
Command-X	Edit > Cut	Cuts the selection
Command-Z	Edit > Undo	Undoes the last action
Command-Shift-Z	Edit > Redo	Redoes the last undo

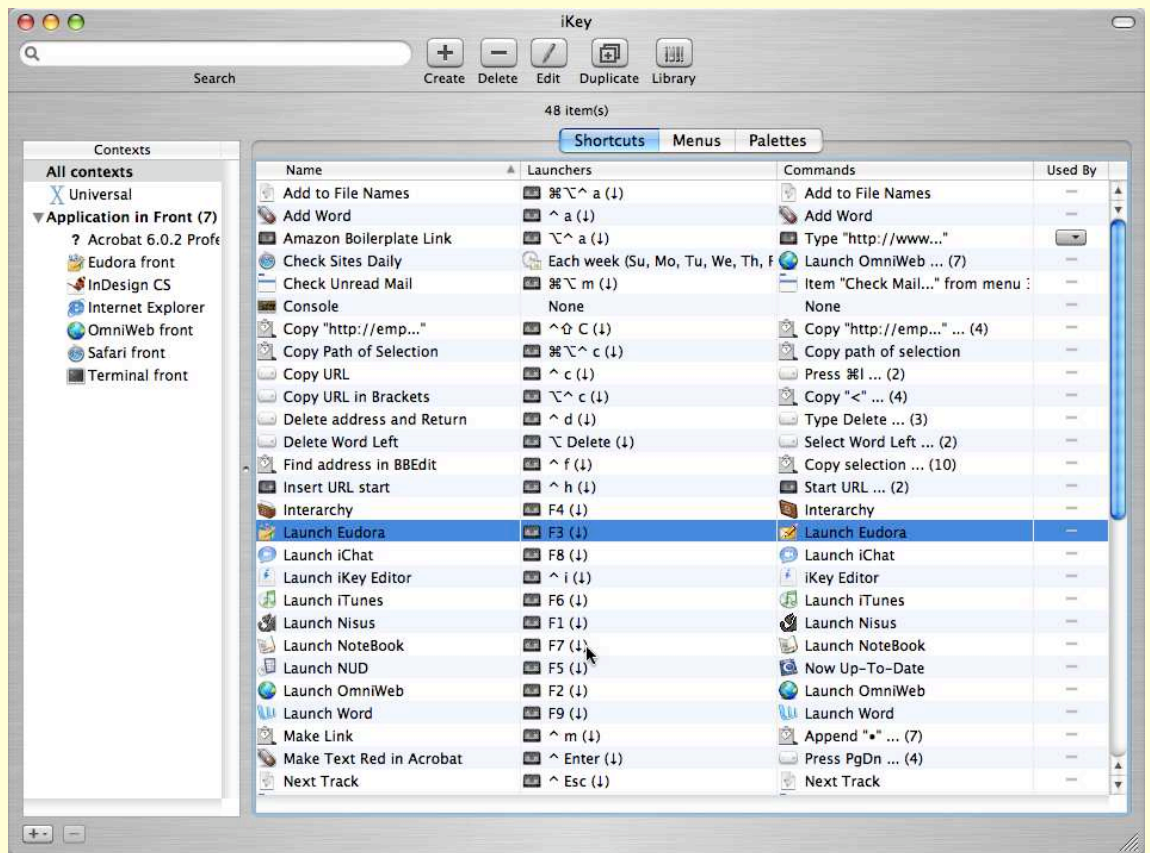
## WINDOW REFERENCE

Most of your work in iKey Editor takes place in the main window and in a number of object creation dialogs. In this section, I describe the function of the different parts of the windows and dialogs; refer to the [Usage Tutorial](#) for instructions on how to create and edit shortcuts, menus, and palettes.

### Main Window

iKey Editor's main window provides controls for creating and editing all of iKey's objects, and it lists those you've created (**Figure 9.1**).

**FIGURE 9.1**



iKey Editor's main window

In particular, you'll probably use the following controls regularly:

- **Contexts pane:** This pane, which displays on the left side of the main window, just like the Source pane in iTunes or iPhoto, lists all the contexts within which you have shortcuts, menus or palettes. Clicking a context works just like clicking a playlist in iTunes or an album in iPhoto—it shows the subset of shortcuts that

are active within only that context. Similarly, the All Contexts item is like the Library item in iTunes or iPhoto; it shows every shortcut whether or not it's specific to a context.

**TIP** There are some hidden ways to edit and delete contexts within the Contexts pane: Command-Delete deletes a selected context, and double-clicking a context edits it. You can also Control-click (or right-click, with a multi-button mouse) to access a contextual menu that provides Edit, Export, and Delete commands.

- **Toolbar controls:** By default, iKey Editor displays a number of controls in the toolbar, although you can customize it by choosing Window > Customize Toolbar (see [Window](#)). First comes the Search field, where you can enter a few letters to restrict the list in the main window to matching items. After that come the Create (+), Delete (—), Edit (pencil), Duplicate (+ in a square), and Library (bookshelf icon) buttons. Use them to create, delete, edit, and duplicate shortcuts, menus, and palettes, and to display the Library window.
- **Shortcut/Menu/Palette list:** The bulk of the main window is taken up with a list of shortcuts, menus, or palettes (click the tabs at the top of the list to switch among the three). For each, columns summarize the settings, so, for instance, for a shortcut, you can see its active context, its launcher, which commands it uses, and which menu or palette it's attached to. (Limitations on the display space mean that you see only the first command, launcher, and context, should a shortcut have more than one.) Double-click an item's row in the list to edit it.

**TIP** The easiest way to populate a menu or palette is to select the desired shortcuts, Control-click or right-click them, and choose Create Menu/Palette with Selected Shortcuts from the contextual menu. The contextual menu also duplicates the commands you find in the normal menus, adding just one new command, Delete Only Structure, that deletes a shortcut without deleting the commands or launchers associated with the shortcut.

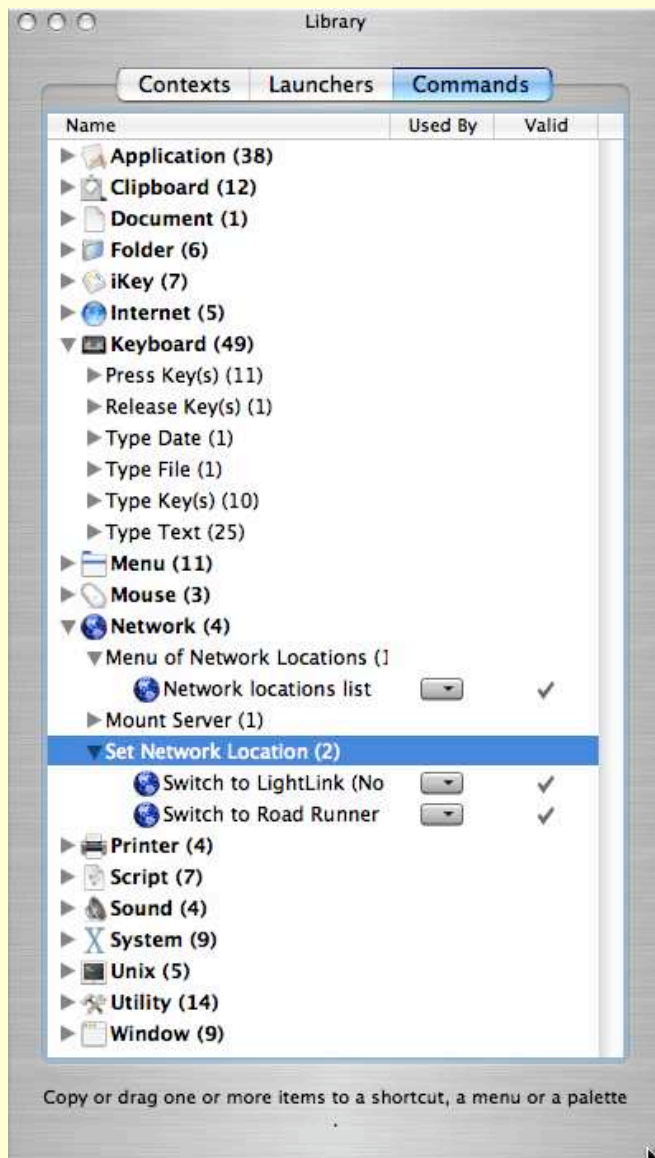


## Library Window

New in iKey 2.1 is the Library window, which offers three panes that list all the contexts, launchers, and commands that you have created (**Figure 9.2**). Click the triangles to reveal or hide all the contexts, launchers, and commands in each category; Option-click a triangle to show or hide all the sub-triangles as well. The Library window floats above all other windows in iKey Editor so it's always accessible.

You drag items from the Library window, to shortcut, menu, or palette creation dialogs, in order to add the items to the shortcut, menu, or palette you're creating.

**FIGURE 9.2**



The Library window, showing commands.

As an example, imagine that you've created a command to launch Safari, and you use it in a shortcut with a hotkey launcher for



switching to Safari from the keyboard. But then you decide that you want to have iKey open a few select Web sites in Safari every day at 9:00 AM, so you create a new shortcut with an appropriate time launcher. Rather than create a new Launch Application command for Safari, though, you can (and should) just drag your existing Launch Safari command from the Library window's Commands pane to the commands area in the shortcut creation dialog.

If you find dragging awkward, you can also select a context, launcher, or command, choose Edit > Copy (Command-C), switch to the shortcut creation dialog, click in the appropriate area, and choose Edit > Paste (Command-V).

**TIP** If you Control-click a particular context or launcher in the Library window, a contextual menu appears with Copy, Edit, and Delete commands. Control-clicking a command provides those three, along with Duplicate, Text, and Export, and, more interestingly, New Shortcut with Selection, New Menu with Selection, and New Palette with Selection. These last three commands are useful shortcuts for creating objects quickly with selected commands.

## Shortcut Creation Dialog

In the shortcut creation dialog, you name and define all the aspects of your shortcut (**Figure 9.3**). (For a tutorial that walks you through using this dialog, see [Create a Shortcut](#).)

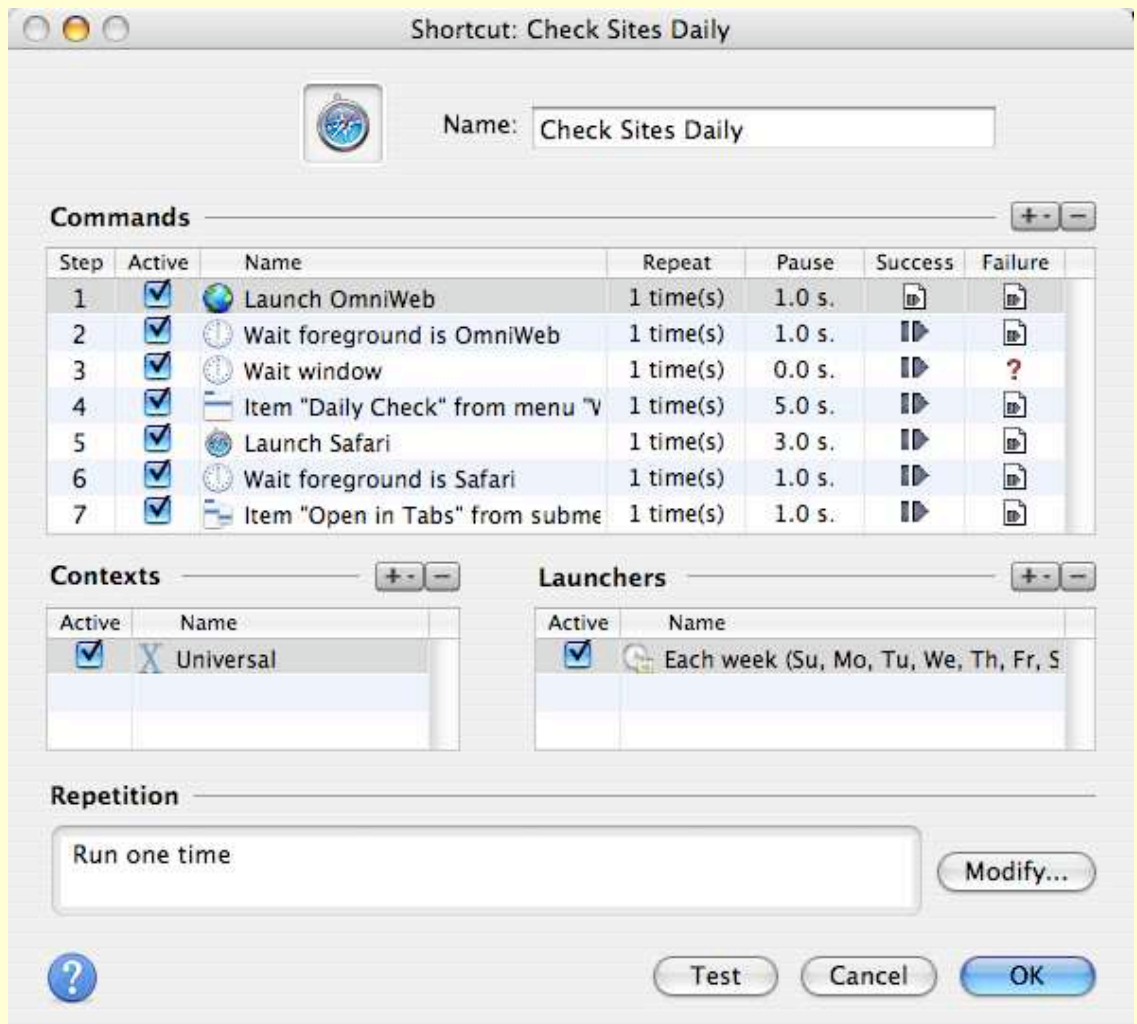
Pay attention to the following areas of the dialog:

- **Name:** iKey Editor suggests names for your shortcuts, but it isn't always successful at coming up with names that makes sense. You can keep the suggestion or enter your own name. The name appears in the Name column of the shortcut list, as well as in any menus or palettes that contain the shortcut, so keep the name short and descriptive.
- **Icon well:** The icon well is located left of the Name field. Drag a graphic into the icon well to use that icon for the shortcut when it appears in menus or, more important, in palettes. Starting in iKey 2.2, you can also Control-click or right-click the icon well to display a contextual menu with two useful commands: Select Image File and Select Image from Icon of File. With the first, select

an appropriate graphic file in the Open dialog that appears; with the second, select any file to use its icon as the shortcut's icon.

**WARNING!** Don't drag photos or other large graphics files into the icon well, since doing so will significantly slow down iKey Editor's launch time. In most cases, only small, custom-designed graphics will work well anyway.

**FIGURE 9.3**

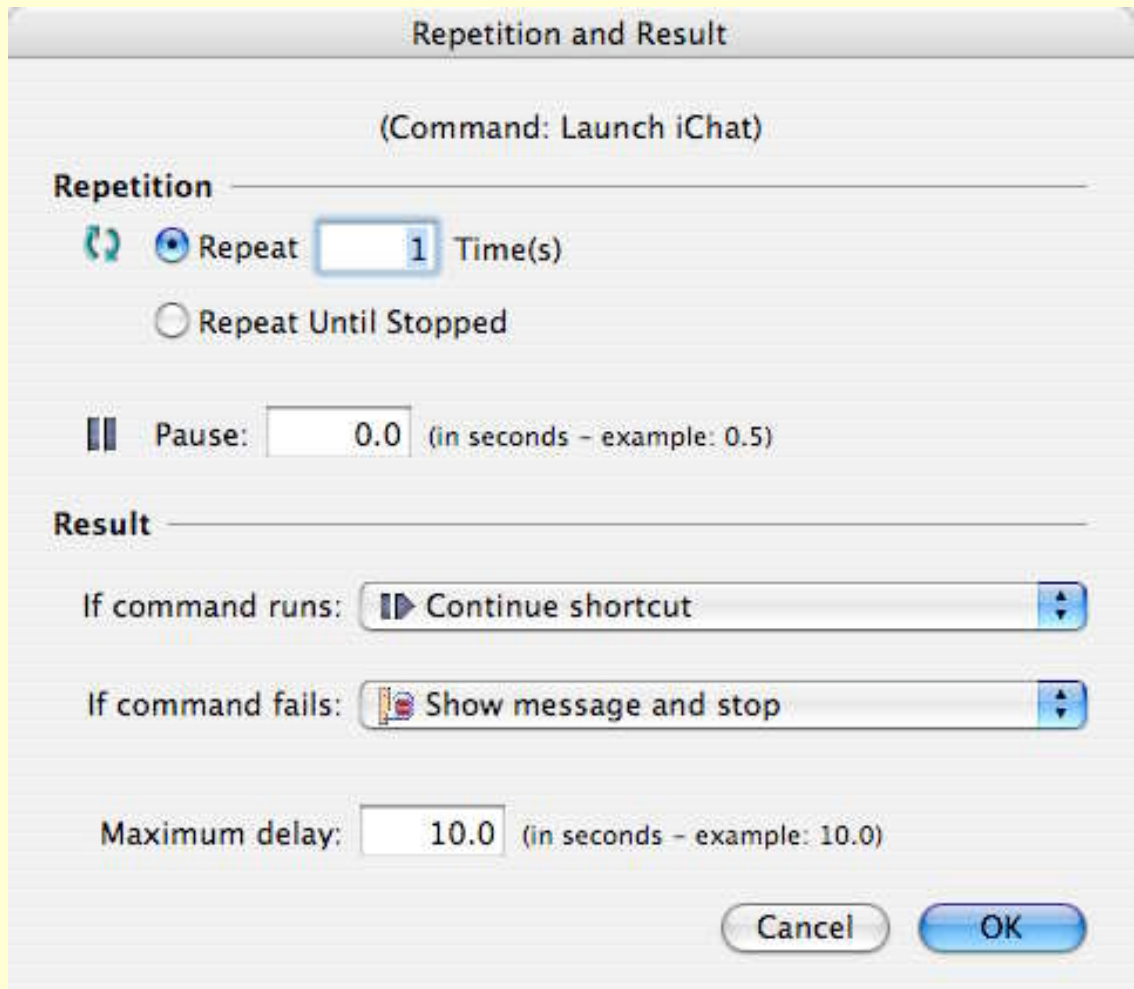


Shortcut creation dialog.

- **Commands list:** Here's where you define the guts of your shortcut. The + pop-up menu and — button enable you to add new commands and remove existing commands; remember that you can drag, or copy-and-paste, already-defined commands in from the Commands pane of the Library window. Drag commands around in the list to change their order of execution. You can also

deactivate a command without deleting it (useful for testing!) by deselecting its checkbox. Each command has its own repetition and result settings. Double-click a command's name to edit it, and double-click its Repeat, Pause, Success, or Failure column to open the Repetition and Result dialog (**Figure 9.4**).

**FIGURE 9.4**



Repetition and Result dialog.

The top half of the dialog lets you specify how many times the command should repeat and how long it should pause between executions.

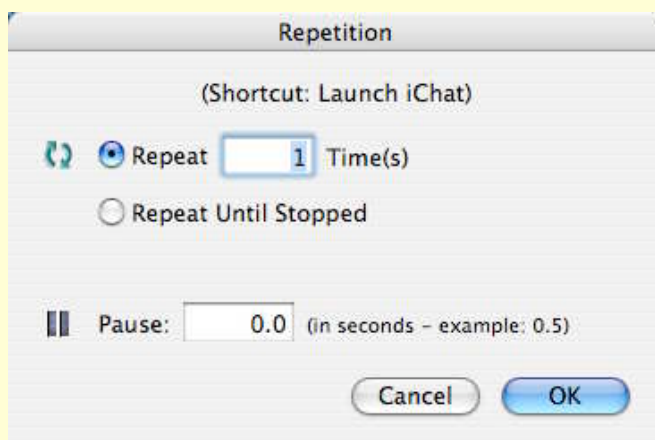
The bottom half is different. In it you can define what happens if a command runs successfully, what happens if it fails for some reason, and how long it should be given to run. Most of the time, the default settings are fine, since they ensure that successful commands continue the shortcut after running, and failed commands ask you what to do. But you can also have the command

continue after showing a message or writing an entry to the iKey report, or you can have it stop cold, stop after showing a message, or stop after writing a line to the iKey report.

**TIP** You may want to extend the default delay of 10 seconds when working with multiple-step shortcuts or when using commands that run AppleScript scripts that may take a while to complete.

- **Context list:** The context list shows the contexts in which the shortcut is active. The Universal context is always included by default, although you can remove it by selecting it and clicking the — button at the upper right of the list, or by selecting it and pressing Command-Delete. In that case you'd want to add a new context using the pop-up + menu or add an existing one by dragging it in from the Contexts pane of the Library window.
- **Launcher list:** The launcher list shows the launchers that can activate the shortcut. Again, use the + pop-up menu to add a new launcher or drag an already-defined launcher in from the Launchers pane of the Library window. To remove a launcher, click the — button (or select it and press Command-Delete).
- **Repetition:** Click the Modify button to open the Repetition dialog (**Figure 9.5**) and control how many times the shortcut itself runs, and with what pause between runs. By default, each shortcut runs once.

**FIGURE 9.5**



Repetition dialog.

In the Repetitions dialog, set how many times you want the shortcut to repeat, and how long iKey should pause between repetitions.

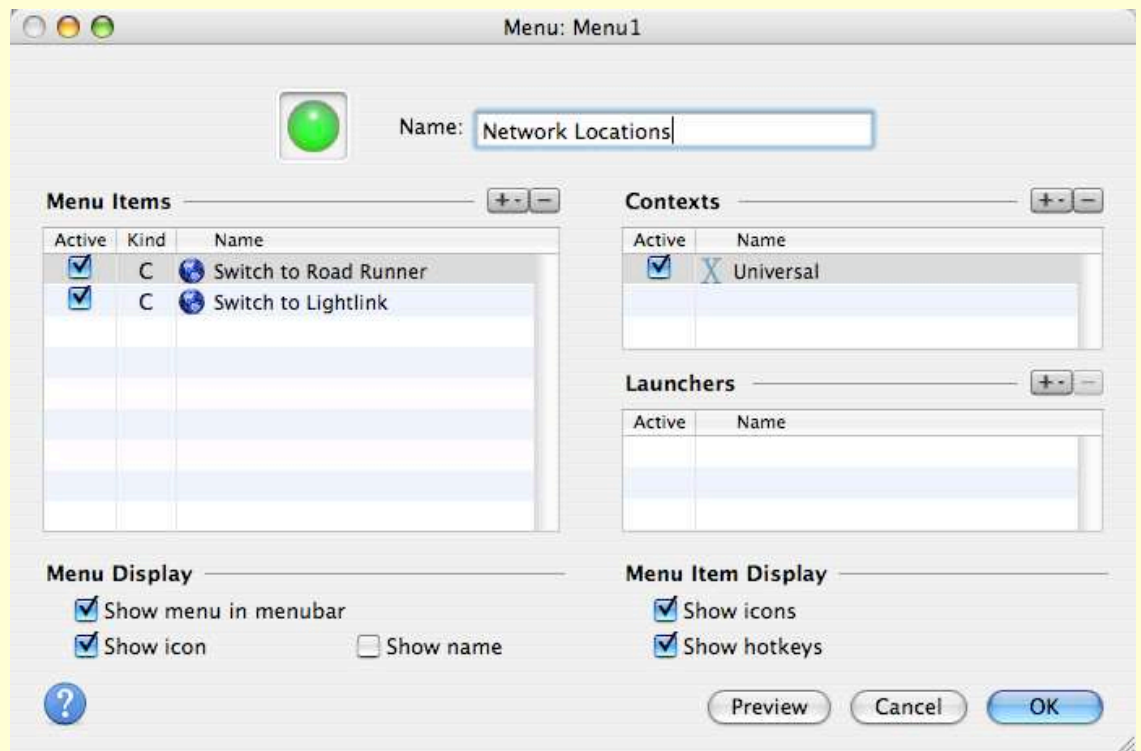
**TIP** How do you stop a shortcut that runs until stopped? Invoke the shortcut again. So if the launcher you use to activate the shortcut is Control-Enter, press Control-Enter again to stop a shortcut that's intent on running forever. If that doesn't work, see the warning in [Type File](#) for another possibility.

- **Control buttons:** The four buttons at the bottom of the shortcut creation dialog are mostly self-explanatory. The blue ? button opens this manual. Clicking the Test button causes iKey to run the shortcut as it currently exists; note that you don't have to save before clicking Test, as you do if you were to use the defined launcher for the shortcut instead. Click Cancel to close the dialog without making any changes, and click OK to save your changes and close the dialog.

**TIP** Honestly, the Test button isn't as helpful as it might seem, since a lot of shortcuts depend on pre-existing conditions that aren't true when you're still working in iKey Editor. For such shortcuts, either use the [Test Edited Shortcut/Command](#) command or just define them, save, and then test in a real-world situation.

## Menu Creation Dialog

The menu creation dialog provides controls for making a menu, whether it's a pop-up menu that appears when you press a hotkey or one that lives in the menu bar (**Figure 9.6**). (For a tutorial about creating menus, read [Create a Menu](#).)

**FIGURE 9.6**

Menu creation dialog.

The options in the menu creation dialog control how the menu and its items look, along with what's in the menu and how it's invoked. Here are some details about the options:

- **Name:** If you intend to have your menu show in the menu bar, I recommend giving it a short name so it doesn't take up too much valuable space.
- **Icon well:** The icon well is located left of the Name field. Drag a graphic into the icon well to use that icon for the menu. If you want to use the icon associated with a particular command or shortcut for the menu, Control-click the item in the Menu Items list and choose Set Image to Menu from the contextual menu that appears. Starting in iKey 2.2, you can also Control-click the icon well to display a contextual menu with two useful commands: Select Image File and Select Image from Icon of File. With the first, select an appropriate graphic file in the Open dialog that appears; with the second, select any file to use its icon as the shortcut's icon.



**TIP** iKey comes with a folder of icons you can use for menus; they're simple, but simple is necessary with such small icons.

**WARNING!** Don't drag photos or other large graphic files into the icon well, since doing so will significantly slow down iKey Editor's launch time. In most cases, only small, custom-designed graphics will work well anyway.

- **Menu Item list:** You define the contents of your menu—commands, shortcuts, other menus, or separators—in the Menu Item list. To add a new command or a separator, use the + menu. To add an already-defined command, drag it in from the Commands pane of the Library window. To add a shortcut or menu, drag it in from iKey Editor's main window (this may require rearranging windows or falling back on copy-and-paste instead). To remove an item from the Menu Item list, select it and click the — button. You can also deactivate a menu item without deleting it by deselecting its checkbox, but note that doing this removes the item from the menu rather than dimming it.

**TIP** Putting a menu in the Menu Item list creates a hierarchical menu.

- **Context list:** The context list shows contexts in which the menu is active. The Universal context is always included by default, although you can delete it by selecting it and clicking the — button to the right, or by selecting it and pressing Command-Delete. In that case you'd want to add another context using the pop-up + menu or dragging an already-defined context in from the Contexts pane of the Library window.
- **Launcher list:** The launcher list shows the launchers that can activate the menu. Again, use the + pop-up menu to add a new launcher or drag an already-defined launcher in from the Launchers pane of the Library window. To remove a launcher, click the — button (or select it and press Command-Delete).

**TIP** If your menu appears in the menu bar, there's no need for a launcher. The date/time, application, and system event launchers do work for menus, causing them to appear wherever the pointer is, but they don't make a lot of sense to use that way, so I recommend sticking with keyboard event launchers for menus. That way you can press a hotkey and have a menu appear at the pointer location.

- **Menu Display:** Use the three checkboxes in this area to determine if the menu should display in the main system menu bar, and if so, if it should display its icon and/or name.
- **Menu Item Display:** These two checkboxes control the look of the individual items within the menu, letting you choose if you want to see icons and hotkeys (which map to the keyboard launchers for shortcuts) in the menu.
- **Control buttons:** The four buttons at the bottom of the shortcut creation dialog are mostly self-explanatory. The blue ? button opens iKey's help. Clicking the Preview button causes iKey to show your menu as it currently exists; note that you don't have to save before clicking Preview, as you do if you wanted to see the menu in the menu bar. Click Cancel to close the dialog without making any changes, and click OK to save your changes and close the dialog.

## Palette Creation Dialog

The palette creation dialog provides all the controls you need to create a palette of buttons (**Figure 9.7**). (For a tutorial about creating a palette, see [Create a Palette](#).)

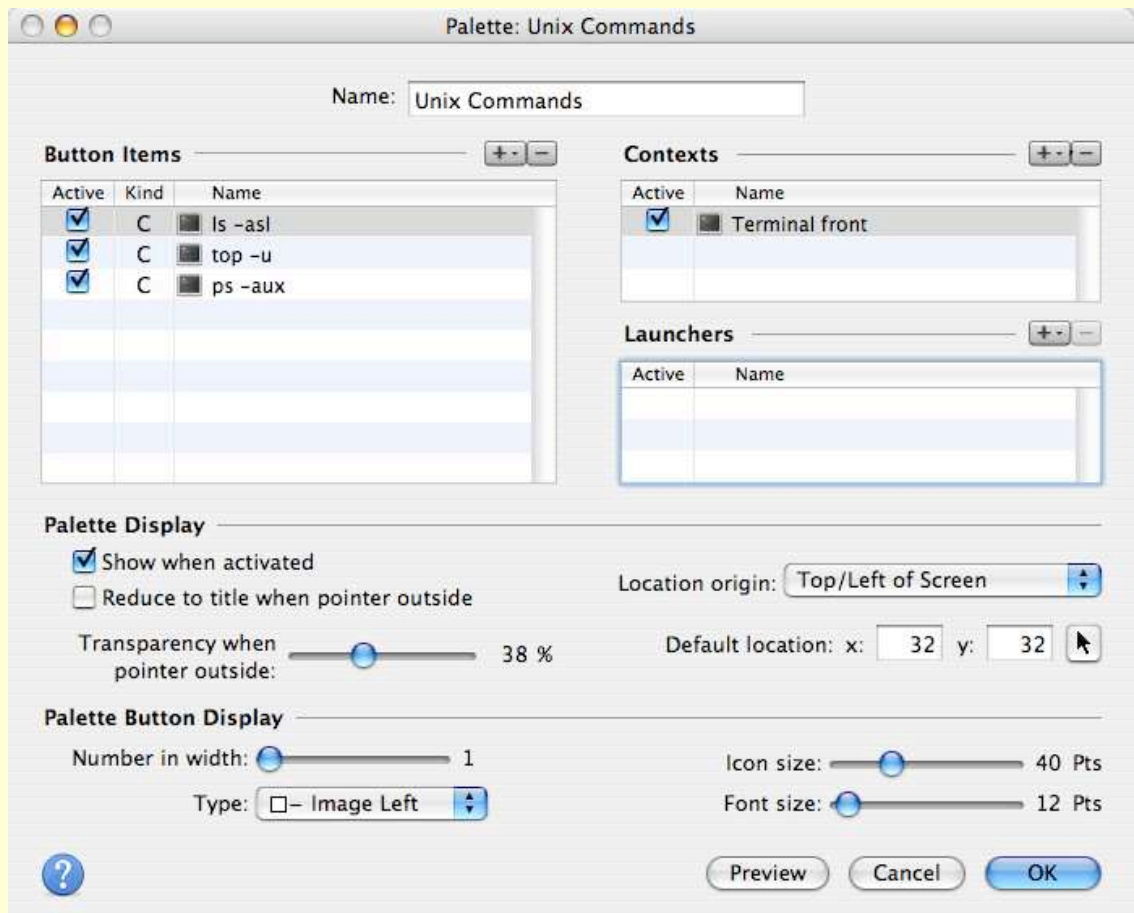
The top of the palette creation dialog contains the list of button items and the familiar lists of contexts and launchers. The middle section controls the appearance and behavior of the palette, and the bottom section controls the appearance of the buttons within it. Here are more details:

- **Name:** The palette's name appears in its title bar, so make the name short enough to fit nicely.
- **Button Item list:** You define the contents of your palette—commands and shortcuts—in the Button Item list. To add a new command, use the + menu. To add an already-defined command, drag it in from the Commands pane of the Library



window. To add a shortcut or menu, drag it in from iKey Editor's main window (this may require rearranging windows or falling back on copy-and-paste instead). To remove an item from the Button Item list, select it and click the — button. You can also deactivate a button without removing it by deselecting its checkbox.

**FIGURE 9.7**



Palette creation dialog.

**TIP** If you add a command, like the [Menu of Network Locations](#), that normally displays a menu, clicking it (don't click-and-hold) in the palette displays the expected menu.

- **Context list:** The context list shows contexts in which the palette is active. The Universal context is always included by default, though you can delete it by selecting it and clicking the — button to the right, or by selecting it and pressing Command-Delete. In that case you'd want to add another context using the pop-up + menu

or dragging an already-defined context in from the Contexts pane of the Library window.

**TIP** By assigning an application-specific context to the palette, you can ensure that it appears only when a specific application is active. In essence, then, you can use iKey to add your own palettes to any application!

- **Launcher list:** The launcher list shows the launchers that can activate the palette. This is important if you've deselected the Show When Activated checkbox. Again, use the + pop-up menu to add a new launcher or drag an already-defined launcher in from the Launchers pane of the Library window. To remove a launcher, click the — button (or select it and press Command-Delete).
- **Palette Display:** Use the controls in this area to set where the palette appears when invoked, how transparent it should be when the pointer is outside the palette, and if the title should show when the pointer is outside the palette. Pay attention to the Show When Activated checkbox—if it isn't selected, your palette will appear only when you invoke it with a launcher. Check Show Title When Pointer Outside if you want the palette to shrink down to just its title bar until you move the pointer over it.
- **Palette Button Display:** The sliders in this section enable you to set how wide, in number of buttons, your palette should be; how large the icons should be, and how large the icon names should be. Use the Type pop-up menu to choose the icon style (with or without name, and where the name should appear).
- **Control buttons:** The four buttons at the bottom of the shortcut creation dialog are mostly self-explanatory. The blue ? button opens iKey's help. You can click the Preview button to see your palette as it currently exists, although the buttons in the palette don't work when it's in preview mode. Note too that you don't have to save before clicking Preview, as you do if you wanted to be able to test the buttons. Click Cancel to close the dialog without making any changes, and click OK to save your changes and close the dialog.

## COMMAND REFERENCE

The full power of iKey is revealed in the breadth and depth of the many individual commands you can create and string together to create shortcuts, menus, and palettes.

Most of iKey's commands should seem familiar because they're in fact the same actions you use to control your Mac every day. Almost anything you can do with your Mac manually—pressing keys, moving the mouse, opening and closing windows, launching applications, and much more—you can perform automatically with iKey.

**NOTE** Where possible, I tell you if there's a direct equivalent between an iKey command and a menu item or other control in Mac OS X. For instance, iKey offers a [Hide Background Applications](#) command that functions identically to the Hide Others menu item that appears in the application menu for every program.

You can think of iKey's commands as templates, since you can create multiple instances of each command, each customized differently. This way of thinking makes sense in relation to a command like Type Text, since it's easy to see why you would want to create different Type Text commands with different bits of boilerplate text to type. However, there are also commands like Menu of Network Locations that can't be customized, so there's no need to create more than one of them, and once you've created one, you can re-use it in as many different shortcuts, menus, and palettes as you like.

### Creating Commands

You create commands in one of five ways:

- While creating a shortcut, menu, or palette, click the + button and choose New from the appropriate command category submenu.
- Choose Create > Command, and then choose New from the appropriate command category submenu.
- Drop an object on the iKey Editor icon in the Dock; iKey Editor responds with a dialog listing commands that are appropriate to create with the type of object you dropped, including Open Document, Display Folder Hierarchy, Run Script From File, and

so on. Select one and click Add to create a new shortcut using the dropped object.

- Copy an object (a file from the Finder, or some text, for instance), switch to iKey Editor, make sure the Commands tab in the Parts drawer is selected, and paste. iKey Editor responds with a dialog just as in the previous method.

In each case, iKey Editor then displays a Command window with controls specific to each type of command. Let's look at the different categories of commands now.

## Application Commands

iKey offers a slew of different Application commands for launching and quitting applications, switching among applications, hiding and showing applications, and more. For clarity, I've collected them into several groups, starting with the Launch and Quit commands.

### Launch Application

Use the Launch Application command to launch an application that's not currently running. Using the Launch Application command is like double-clicking an application's icon in the Finder, or clicking its Dock icon, so if an application is running when you invoke its Launch command, it will come to the front. To hide all other applications when the specified application launches, select the Hide Others checkbox.

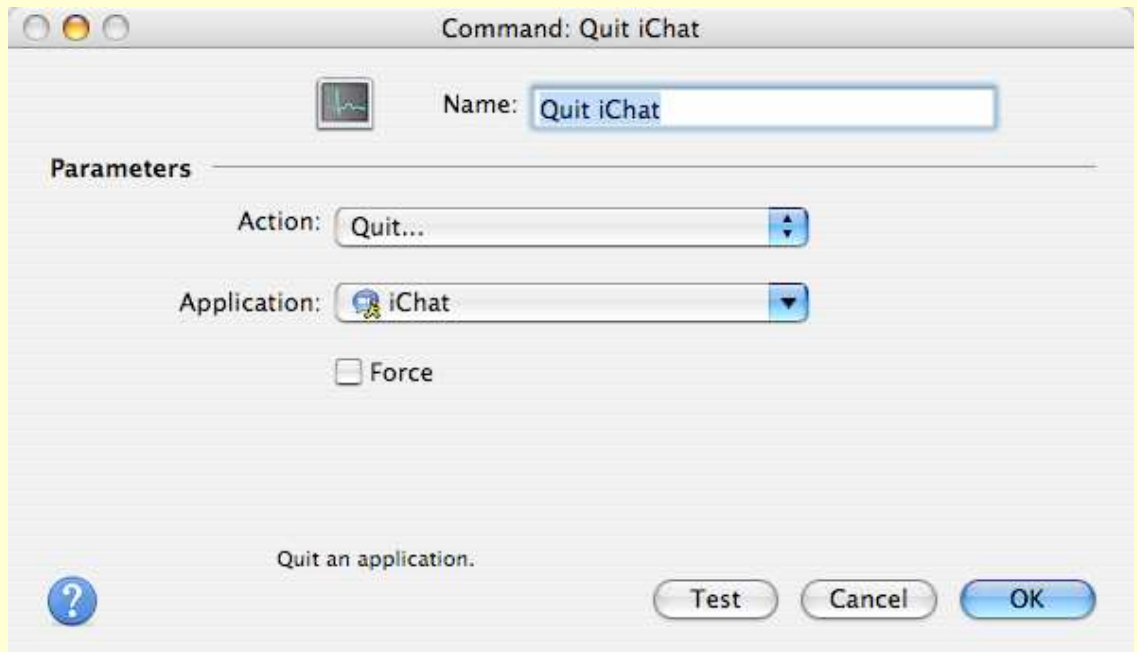
**TIP** Whenever you specify an application or document when creating a command like Launch Application, you can move that application or document to a different folder without fear of breaking your command.

### Quit Front Application

Use the Quit Front Application command to quit the front application (**Figure 10.1**). This command works exactly like choosing Quit (Command-Q) from a program's application menu.

**WARNING** For all the Quit and Relaunch commands, you can check the Force checkbox to force quit the application. This option is useful if you're working with an application that freezes in known ways, but this option is dangerous to use normally, since it might prevent an application from saving changes to its preferences when it quits.

**FIGURE 10.1**



Quit command.

### **Quit Background Applications**

Use the Quit Background Applications command to quit all applications other than the front application. Keep in mind that if some of those applications have unsaved documents open, those applications will ask you how to proceed.

### **Quit Application**

Use the Quit Application command to quit the application you choose from the Application menu.

### **Relaunch Front Application**

Use the Relaunch Front Application command to quit and relaunch the front application.

### **Relaunch Background Applications**

Use the Relaunch Background Applications command to quit and relaunch applications other than the front one. Honestly, the chances of you wanting to do this is pretty low, but hey, the capability is here if you need it.

### **Relaunch Application**

Use the Relaunch Application command to quit and relaunch the application you choose from the Applications menu.

**NOTE** With its switching commands, iKey gives you tight control over which application is in front at any given time. For all the switching commands, select the Hide Others checkbox to hide the windows of other applications.

### **Switch to Next Application**

Use the Switch to Next Application command to switch to the next application. When determining which application is “next,” iKey uses the order in which applications were launched.

### **Switch to Previous Application**

Use the Switch to Previous Application command to switch to the previous application. When determining which application is “previous,” iKey uses the order in which applications were launched.

**TIP** Switch to Next and Switch to Previous can be hard to predict, since you seldom know exactly the order in which applications were launched. I prefer using Switch to... and specifying precisely which application I want to activate.

### **Switch to Application**

Use the Switch to Application command to switch to the application you choose from the Application menu.

### **Save Front Application**

Use the Save Front Application command to record the name of the front application for use with the Switch to Saved Application command.

### **Switch to Saved Application**

Use the Switch to Saved Application command to switch to the application you previously recorded with the Save Front Application command.

**TIP** Save Front Application and Switch to Saved Application are useful primarily in multiple-step shortcuts where you’re moving among a changing set of applications and want to return to the initial front application when you’re done.

## Switch to Previous Front

Use the Switch to Previous Front command to switch to the application that was in front just before you brought the current one to the front. Invoking Switch to Previous Front repeatedly switches between two applications quickly. It's exactly the same as pressing Command-Tab quickly to invoke Mac OS X's built-in application switching hotkey.

## Menu of Recent Applications

Use the Menu of Recent Applications command to display a pop-up menu of recent applications, sorted by launch order or name. Select the Hide Others checkbox to cause your Mac to hide other applications after you choose an application from the pop-up menu. The contents of the menu are the same as those in the Applications section of Apple > Recent Items.

**TIP** Set the number of recent applications that iKey displays in the Recent Applications menu in Mac OS X's General preference pane. iKey Editor provides a button in the command's configuration dialog that you can click to open the General preference pane.

## Menu of Running Applications

Use the Menu of Running Applications command to display a pop-up menu of currently active applications, sorted by launch order or name. Select the Hide Others checkbox to make your Mac hide other applications after you choose an application from the pop-up menu.

**NOTE** Use the Application Hide and Show commands discussed next to hide and show applications. These are easy to create and almost entirely self-explanatory, since all you must do to create one is choose your desired action from the Action pop-up menu; in two cases, you can also choose a specific application from the Application pop-up menu.

## Show Front Application

One subtle change between Mac OS 9 and Mac OS X is that clicking an application window in Mac OS 9 brings all of that application's windows to the front, whereas in Mac OS X it brings only the clicked window to the front. (Clicking an application's Dock icon in Mac OS X brings all its windows to the front.)



Use the Show Front Application command to bring back that Mac OS 9 behavior in select situations—when an application is in the front but some of its windows are not, I invoke a shortcut that runs the Show Front command to bring all the other windows forward.

### **Show Background Applications**

Use the Show Background Applications command to reveal the hidden windows of background applications.

### **Show All Applications**

The Show All Applications command does roughly the same thing as Show Background Applications, displaying the hidden windows of all applications. It's exactly the same as the Show All menu item present in every program's application menu.

### **Show Application**

To show a specific application, choose Show Application and then choose your desired application from the Application pop-up menu.

### **Hide Front Application**

Use this command to hide the front application. It's equivalent to the Hide menu item present in every program's application menu.

### **Hide Background Applications**

To clear your screen of windows from applications other than the front one, use the Hide Background Applications command. It's exactly the same as choosing Hide Others from the application menu in any program.

### **Hide Application**

To hide a specific application, choose Hide Application and then choose your desired application from the Application pop-up menu.

## Clipboard Commands

The clipboard has long been a mainstay of Macintosh usage. When you copy text or graphics, they're placed on the clipboard, and when you paste, the contents of the clipboard are inserted in the current document. iKey's Clipboard commands significantly extend the basics of clipboard use by letting you manipulate text on the clipboard.

**TIP** A number of the Clipboard commands operate on the selected text, so you might wonder what makes them Clipboard commands. In short, these command actually copy the text to the clipboard, modify it there, and then paste it back. However, even though this sort of behavior would normally replace whatever was in the clipboard, iKey 2.2 puts the original clipboard contents back after running a shortcut that includes these Clipboard commands: Capitalize Words in Selection, Lower Case Selection, Replace in Selection, and Upper Case Selection.

Because different Clipboard commands take different parameters, the windows in which you create them look different.

### Capitalize Words in Selection

Use the Capitalize Words in Selection command to capitalize the first letter in each word in the selection, so “End of the Internet. News at 11!” would become “End Of The Internet. News At 11!”. Note that this command doesn't attempt to do proper title case, in which certain types of words are not capitalized.

**WARNING!** Be careful when using Capitalize Words in Selection, Lower Case Selection, and Upper Case Selection in the Finder. If you select an icon for a file or folder instead of its name, these commands will make a copy of the selected item rather than change the case of its name.

### Copy Selection

Use the Copy Selection command to put the selection on the clipboard (replacing what was there before) for pasting elsewhere; it's exactly the same as the Copy menu item (Command-C) in every application. Select the Append to Clipboard checkbox to append the

selection to the clipboard without replacing what was there before; this isn't something normal Copy commands can do.

### **Cut Selection**

Use the Cut Selection command to remove the selection from your document and put it on the clipboard (replacing what was there before) for pasting elsewhere. It's the same as the Cut menu item (Command-X) in every application. Select the Append to Clipboard checkbox to append the selection to the clipboard without replacing what was there before; this isn't something normal Cut commands can do

### **Lower Case Selection**

Use the Lower Case Selection command to change all the letters in a text selection to lower case, so "TidBITS" would become "tidbits".

### **Paste Clipboard**

Use the Paste Clipboard command to paste the contents of the clipboard at the insertion point in the current document or application. iKey's Paste Clipboard is identical to the Paste menu item (Command-V) in every application. Paste Clipboard does not change the contents of the clipboard.

### **Put Path of Selection in the Clipboard**

Mac OS X's Unix underpinnings make knowing the full path to a file or folder useful. Use the Put Path of Selection in the Clipboard command to copy the full Unix path of the selected item in the Finder to the clipboard. Select the Append to Clipboard checkbox to append the path to the clipboard without replacing what was there before.

**TIP** Appending a path to the contents of the clipboard could be a useful way to build Unix commands within iKey. Also note that the Put Path of Selection in the Clipboard command makes sense only in the Finder; it won't work in other applications.

### **Put Text in the Clipboard**

Your clipboard control in iKey is not limited to text you can select. With the Put Text in the Clipboard command you can put any text you want directly onto the clipboard, either replacing what was there or appending to it (select the Append to Clipboard checkbox).

Selecting the Paste Clipboard checkbox continues the action, pasting the text at the insertion point. This is an excellent technique for inserting large quantities of text into a document.

**TIP** Also see the [Keyboard](#) commands, which simulate the action of typing. Sometimes, such as in dialogs, Keyboard commands work where Clipboard commands don't, but in situations where you want to enter a fair amount of text, a Clipboard command will likely operate more quickly.

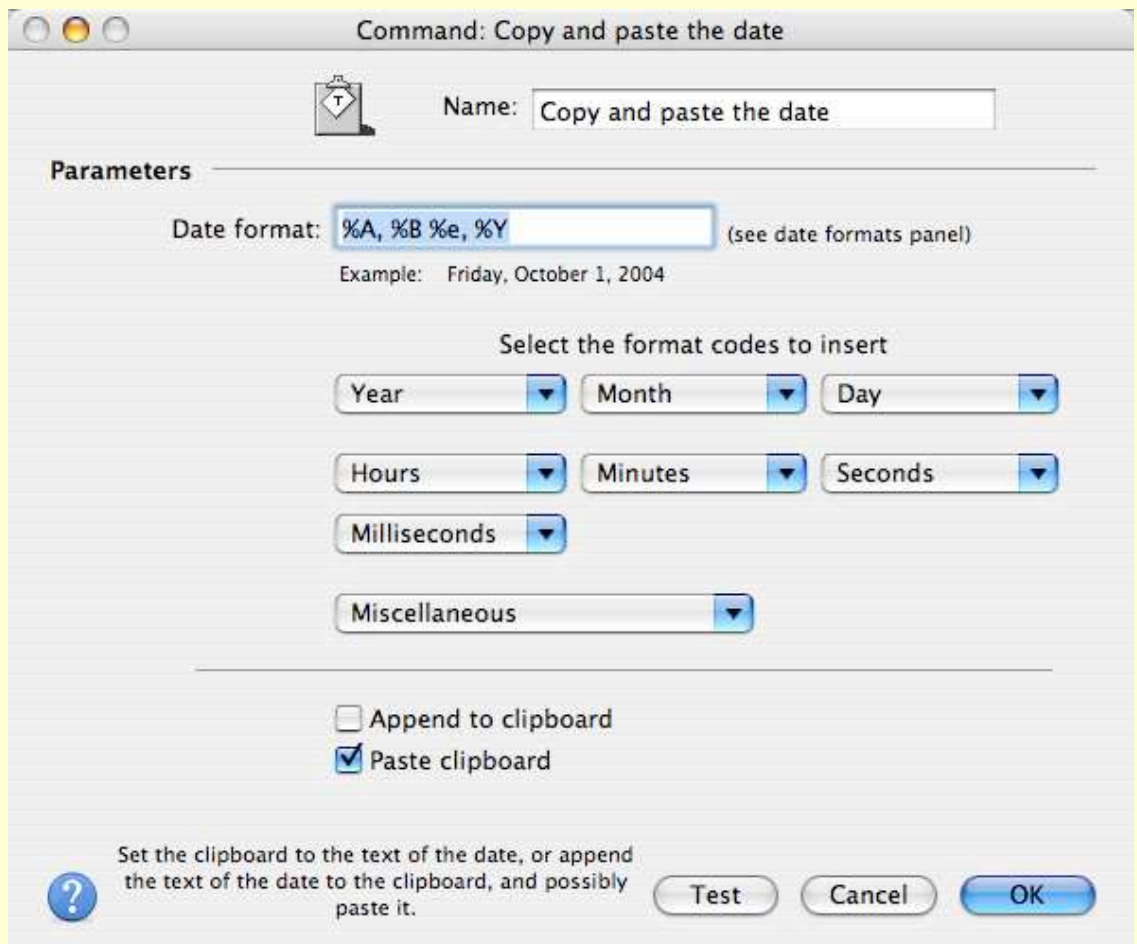
### Put Text of the Date in the Clipboard

With the Put Text of the Date in the Clipboard command, you can add the current time and date to the clipboard, either replacing what was there or appending to it (select the Append to Clipboard checkbox). And, as with the Put Text in the Clipboard command, select the Paste Clipboard checkbox to paste the date at the insertion point.

The tricky part of the Put Text of the Date in the Clipboard command is that iKey gives you extremely fine-grained control over the format of your date text. So, if you want your date to read Monday, July 12, 2004, that's just as easy as having it read 12-Jul-04. In fact, that's why, when you create a Put Text of the Date in the Clipboard command, iKey opens a Date Formats window showing you all the possible codes for different aspects of the date (see **Table 10.1** in the [Type Date](#) command for a complete list). The Date Formats window closes automatically when you close the window for the Put Text of Date in the Clipboard command.

Looking up codes in the Date Formats window can be daunting, so iKey Editor provides an interface for building these date formats (**Figure 10.2**). For each code you want to insert in the Date Format field, choose the corresponding item from a pop-up menu.

What about niceties such as commas and spaces? Just enter those manually in the Date Format field until the example underneath looks the way you want.

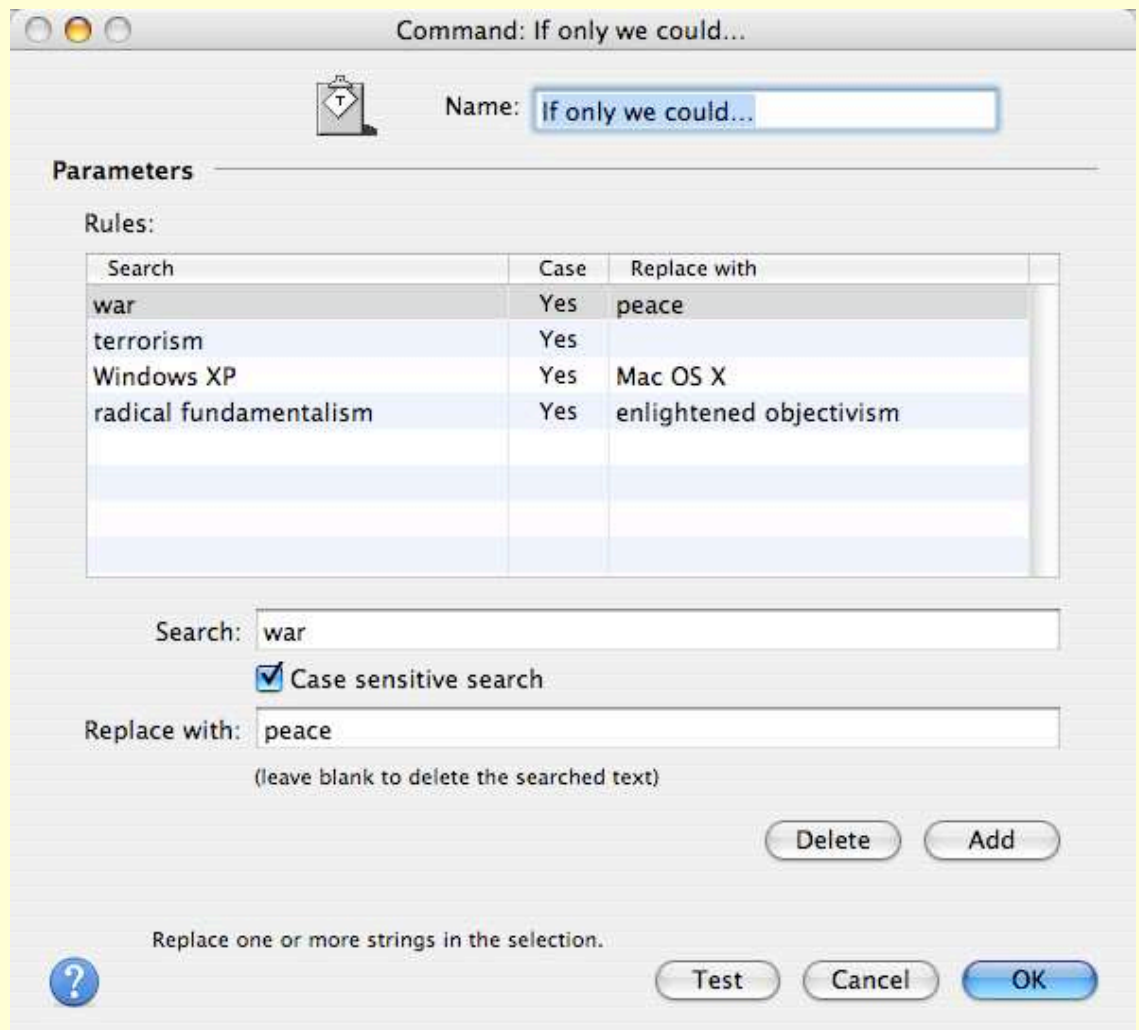
**FIGURE 10.2**

Either enter the date format you want to use or build it by choosing items from the different pop-up menus.

## Replace in Selection

The Replace in Selection command is perhaps the most powerful of the Clipboard commands. With it, you can replace runs of one or more characters with other characters—it's basically a Find/Replace command that works anywhere you can select text.

So, for example, imagine you're creating a shortcut to convert normal English to use the kind of abbreviations common in text messaging. You might want to change "for" to "4" and "to" to "2" throughout the selection. You can create multiple replacements within a single command, as shown in **Figure 10.3**, and you can also delete the text instead of replacing it by leaving the Replace With field blank.

**FIGURE 10.3**

Identify the text strings to search for, and what, if anything, you want to replace them with.

To create a command like the one shown in **Figure 10.3**, follow these steps in the Replace in Selection window:

1. Give your command a name that identifies what it will replace.
2. Click the Add button to add a Search/Replace With set.
3. Enter the text to search for in the Search field.
4. To make the search case-sensitive (so “TidBITS” is not the same as “tidbits”), select the Case Sensitive Search checkbox.
5. Enter the replacement text in the Replace With field. Leave the field blank to delete the Search text.

6. Repeat Steps 2–5 for additional combinations of text to search for and replace. If you make a mistake, select the pair in the list and click the Delete button to remove it.

**TIP** To search and replace by ASCII code, enter the characters to look for and replace in the form /#### where ### is the decimal number corresponding to the character's ASCII code. This feature is handy when you want to search for or replace with non-printing characters.

## **Restore Clipboard**

Use the Restore Clipboard command at the end of a shortcut to restore the contents of the clipboard as saved by a Save Clipboard command.

## **Save Clipboard**

The Save Clipboard command helps you preserve the contents of the clipboard while using some of iKey's Clipboard commands. A number of the clipboard commands behave in such a way as to replace the previous contents of the clipboard with new data. For instance, if you use the Put Text in the Clipboard command in a shortcut, iKey overwrites whatever is in the clipboard when you invoke that command.

iKey 2.2 automatically restores the contents of the clipboard after using Capitalize Words in Selection, Lower Case Selection, Replace in Selection, and Upper Case Selection, but if you wish to preserve the contents of the clipboard after using any other Clipboard commands, put the Save Clipboard command at the start of your shortcut, use the rest of your Clipboard commands, and then end the shortcut with the Restore Clipboard command.

## **Upper Case Selection**

Use the Upper Case Selection command to change all the letters in the selection to upper case, so "Upper case used in email is considered shouting." would become "UPPER CASE USED IN EMAIL IS CONSIDERED SHOUTING." Shush.



## Document Commands

Where the Application commands let you work with applications, the Document commands let you work with documents. However, since documents are subordinate to applications, you can do many fewer things with them: open a document, open the selected documents, and view a menu of recent documents.

**TIP** What about other document-related activities like saving and closing? Handle those with commands that work within the application, such as by choosing a Save menu item or pressing Command-W (which closes windows in most applications).

### Open Document

As you would expect, the Open Document command opens the specified document, much as you'd do by double-clicking its icon in the Finder. However, iKey provides a useful extra feature: you can choose a particular application that you want to open the document. Why would you want to do this? Imagine that you want to edit your Web home page (probably called `index.html`) regularly, but Mac OS X always opens `.html` files with your default Web browser. With an Open Document command in a shortcut, it's easy to have iKey open your `index.html` file in BBEdit, TextEdit, or any other text editor.

### Open Selected Documents

The problem with the Open Document command is that you must know exactly what document you want to open, so it works only for documents you open repeatedly. If you want to open an arbitrary set of selected documents, use the Open Selected Documents command instead. Realistically, Open Selected Documents is most useful in a multiple-command shortcut, since if all you wanted to do was open a bunch of selected documents in the Finder, you could just select them and press Command-O. It's also good if you want to open a bunch of documents in an application other than the one that created them.



## Menu of Recent Documents

Use the Menu of Recent Documents command to display a pop-up menu of recently opened documents, sorted by launch order or name. The contents of the menu are the same as those in the Documents section of Apple > Recent Items.

**TIP** Set the number of recent documents that iKey displays in the Recent Documents menu in Mac OS X's General preference pane. iKey provides a button you can click to open it.

## Folder Commands

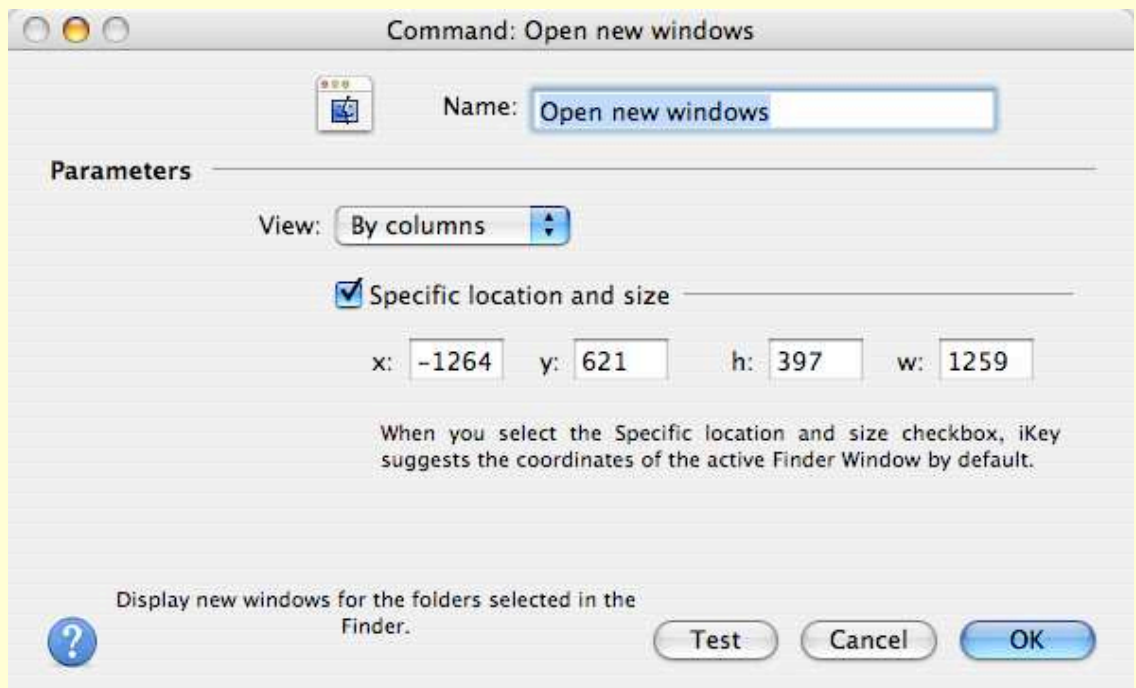
iKey provides a number of commands that let you work with folders, either opening them or displaying pop-up menus with their contents.

**TIP** Obviously, iKey can automate many other tasks that you normally perform in the Finder, but you'll find those commands in the [System](#) category.

### Create New Windows With Selected Folders

This command does exactly what its name suggests: it creates new windows for the folders that you (or another shortcut) have selected in the Finder. As you can see in **Figure 10.4**, you can choose the desired view (icons, list, columns, or default) and you can set a specific location and size for the new windows. Obviously, this command basically just duplicates the result of double-clicking a set of selected folder icons or selecting multiple folders and pressing Command-N.

**FIGURE 10.4**



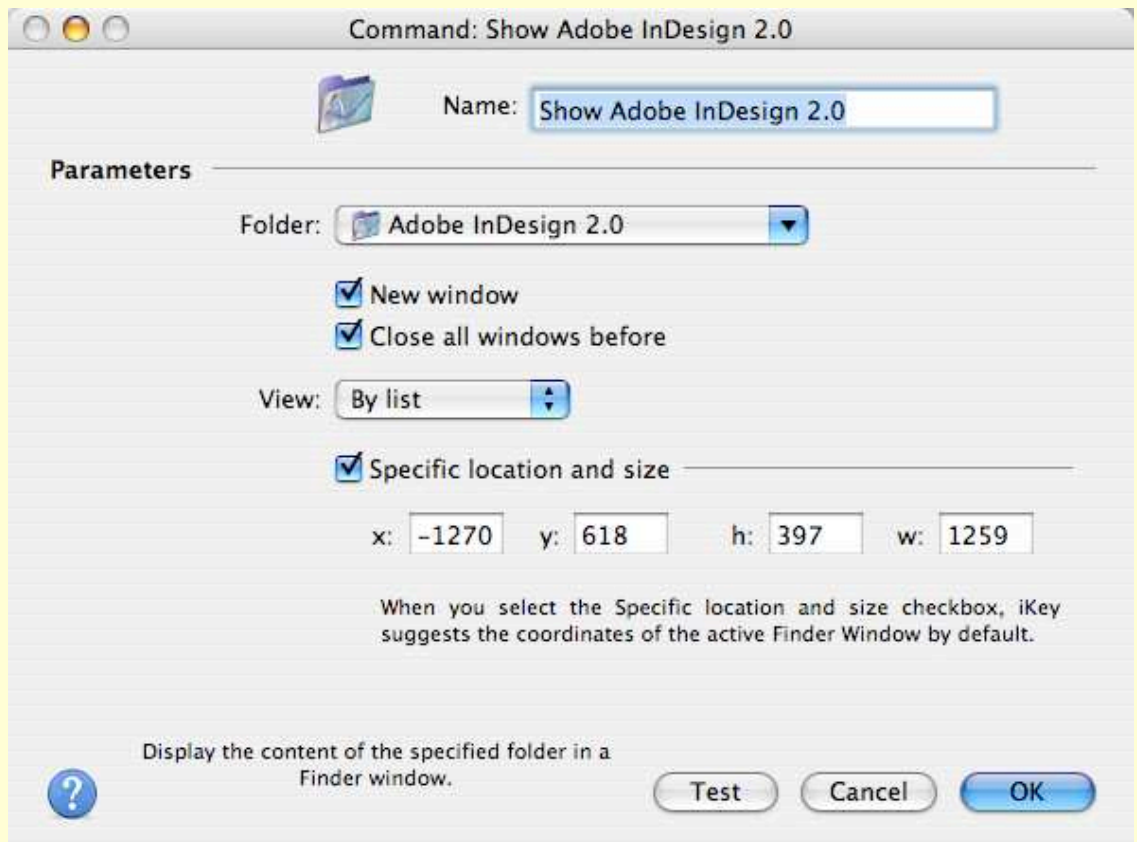
Choose the desired view for your new Finder windows, and set the size and location if you want.

### Display Folder in Finder

Whereas the previous command requires that specific folders be selected first, this command displays the contents of a folder that you

specify in advance. Options (see **Figure 10.5**) let you select whether a new window should be created, and if so, if other windows should be closed to clean up the screen. As with the previous Finder command, you can specify the view and location and size if desired.

**FIGURE 10.5**



Choose options for the folder whose contents you want to display, including whether it should appear in a new window; whether other windows should close; and the view, location, and size.

## Display Folder Hierarchy

Back in the early days of the Macintosh, utilities that displayed a hierarchical view of your entire hard disk from the Apple menu were quite popular. With iKey's Display Folder Hierarchy command, you can do exactly the same thing; display the contents of a specific folder in a hierarchical menu that you either ensconce in an iKey menu or pop-up at the mouse location. The Display Folder Hierarchy command provides a number of options for how the menu is drawn and what items appear in it (**Figure 10.6**). Be careful here, since if the folder you choose and its sub-folders contain lots of items, entering a high number for menu depth and checking Display Icons could result in slower-than-ideal performance.

**FIGURE 10.6**



Configure the Display Folder Hierarchy command so it displays just the items you want, in the way you want.

### **Display Hierarchy of Selected Folder**

Whereas the Display Folder Hierarchy command requires that you pick a folder to start from, the Display Hierarchy of Selected Folder uses the currently selected folder in the Finder.

## **iKey Commands**

You can automate four different actions within iKey using the iKey commands. None of these commands take any parameters.

### **Launch iKey Editor**

Use the Launch iKey Editor command to open the iKey Editor window to create, modify, and delete shortcuts, menus, and palettes. I strongly recommend that you create a shortcut using this command, or at least put it in an always accessible menu or palette, since it's where you do all your real work in iKey. I have a shortcut set to Control-I that I use all the time.

### **Quit iKey Engine**

Although there's seldom a good reason to quit iKey Engine (since once you do, all your shortcuts, menus, and palettes stop working, and if you're like me, that feels like being dead in the water), it's good to be able to do so when troubleshooting. The Quit iKey Engine command lets you do just that; I recommend stashing it away in a menu where you don't have to see it most of the time.

### **Run Last Shortcut/Command**

Use this command on its own as a simple "Do it again" action, since all it does is run the last shortcut you triggered again.

### **Run Next to Last Shortcut/Command**

This command works exactly like the previous command, but executes the next-to-last shortcut instead.

### **Show iKey Report**

Use the iKey Report command to display the iKey Report dialog, which lists entries created by the result parameters of commands. It's probably easier to access it from File > Show iKey Report.

### **Stop Shortcuts**

Some, but not all, shortcuts can be stopped in mid-execution using the Stop Shortcuts command. If you've been using the Mac for a long time, you might consider creating a shortcut based on the Stop Shortcuts command that uses a keyboard launcher set to Command-Period.

## **Test Edited Shortcut/Command**

The Test button in shortcut and command editing dialogs often doesn't work because it relies in some way on a particular application being frontmost, which is never true while you're editing the shortcut itself. New in iKey 2.1 is a command that enables you to test a shortcut or command while you're editing it; I recommend you create a shortcut with this command right away so you can use it whenever you want to test a shortcut or command while editing it.

## Internet Commands

iKey's Internet capabilities are relatively rudimentary, but still highly useful. You can use them to create an email message and to visit Web pages.

### Create Mail

iKey's Create Mail command enables you to compose an email message in your default email program (or another specified email program, if you prefer). Note that the Create Mail command just composes the message; it doesn't also send it (you would have to create another shortcut to make your email program do that).

iKey Editor provides you with three fields: Mail Address, Subject, and Message (**Figure 10.7**). All three are optional, and whatever you leave blank, you'll need to fill in later in your email program. In the Mail Address field, enter the address of the person to whom you wish the mail to be sent (separate multiple email addresses with commas). You can use just the email address, such as **ace@tidbits.com**, or any legitimate format for an email address, such as "**Adam C. Engst**" <**ace@tidbits.com**>, though it's conceivable that some email clients may not accept that text. If you want, fill in the Subject and Message fields. Lastly, choose the email program you wish to use to send the email: your default email application or another one.

**WARNING!** Be careful to enter the email address correctly or your messages may not reach their intended recipients.

**FIGURE 10.7**

Command: Mail to info@scriptsoftware.com

Name: Mail to info@scriptsoftware.com

**Parameters**

Mail address: info@scriptsoftware.com  
(Optional)

Subject: Another cool thing I've discovered about iKey  
(Optional)

Message: Hey Julian and Philippe,  
(Optional)  
cheers... -Adam

Application: Default mail application

Create a message in a mail application.

? Test Cancel OK

In the Create Mail window, enter as much or as little information as is appropriate for the email message you want to create.

I can think of at least four useful ways of using the Create Mail command:

- Make a completely blank Create Mail command in a shortcut to switch to your email program and start a new, blank email message with a single keystroke.
- Fill in just the email address specified to create a blank email message to a specific person.
- Leave the email address field blank, but fill in the Subject and Message fields to create a boilerplate message that you would address in your email program.



- Enter information in all the fields and add the Create Mail command to a shortcut that also makes your email program send the message; this could be great for providing feedback about a task running on a remote Mac.

## Open URL

The Open URL command lets you automate visiting specific pages in your Web browser; it's just like clicking a bookmark in your Web browser. You can have iKey send the URL to your default Web browser, or, if you prefer, you can specify a particular Web browser.

The somewhat subtle thing about this command is that it works for any URL type, not just **http** URLs, and it reads your default helper application from the system list for each URL type. For instance, if you enter an **ftp** URL, and your default FTP helper application is Fetch, iKey launches Fetch and sends it to the URL you specify.

Being able to specify a particular application is handy. For instance, imagine that you're designing a Web site and you need to test it in four different Web browsers. A single iKey shortcut using multiple Open URL commands would let you view a changed page in four programs at the same time.

**TIP** You must enter the address to visit in full URL form. In other words, entering **www.tidbits.com** won't work, whereas entering **http://www.tidbits.com/** will.

## Keyboard Commands

Some of the most common commands you're likely to find yourself using are iKey's Keyboard commands, since, like a player piano, they let iKey type on the keyboard just as you would.

iKey provides a number of keyboard commands, and although some of them may seem unnecessary, they exist because they work in situations where more obvious commands may not. For instance, there's a Keyboard command to type the contents of the clipboard. But if you have text in the clipboard, why wouldn't you just paste it? You would... if you could. But if you're working in a wacky application that doesn't let you paste into fields in a dialog, for instance, you may need to resort to a Keyboard command that types the contents of the clipboard instead.

**NOTE** All the Keyboard commands that type text at the insertion point offer an option to adjust the speed at which iKey types. The default is 50 characters per second. Why might you want to increase or decrease that number?

- Try increasing it if you're using iKey to enter a larger amount of text and you want to speed up the process.
- Decrease the speed if characters are being dropped, which is most likely on slower Macs.

The Press/Release Key(s) commands are inherently linked, so I discuss them first, followed by the rest.

### Press Key(s)

The Press Key(s) command simulates the process of pressing (but not releasing!) a key on the keyboard. It's most commonly used in conjunction with a Simulate Mouse command to make iKey perform an Option-drag, for instance.

### Release Key(s)

If you use the Press Key(s) command to simulate pressing a key, you must use the Release Key(s) command to simulate releasing it when you're done.

**WARNING!** It's extremely important that you pair the Press Key(s) and Release Key(s) commands within a shortcut, because leaving out the Release Key(s) command could freeze your Mac.

## Type Key(s)

Use the Type Key(s) command to simulate pressing one or more keys simultaneously. Just press the desired key in the Type Key(s) dialog and it will record your action. You can specify which modifier keys should be pressed at the same time (**Figure 10.8**).

**FIGURE 10.8**



Press the key you want the Type Key(s) command to enter, and select any desired modifier keys.

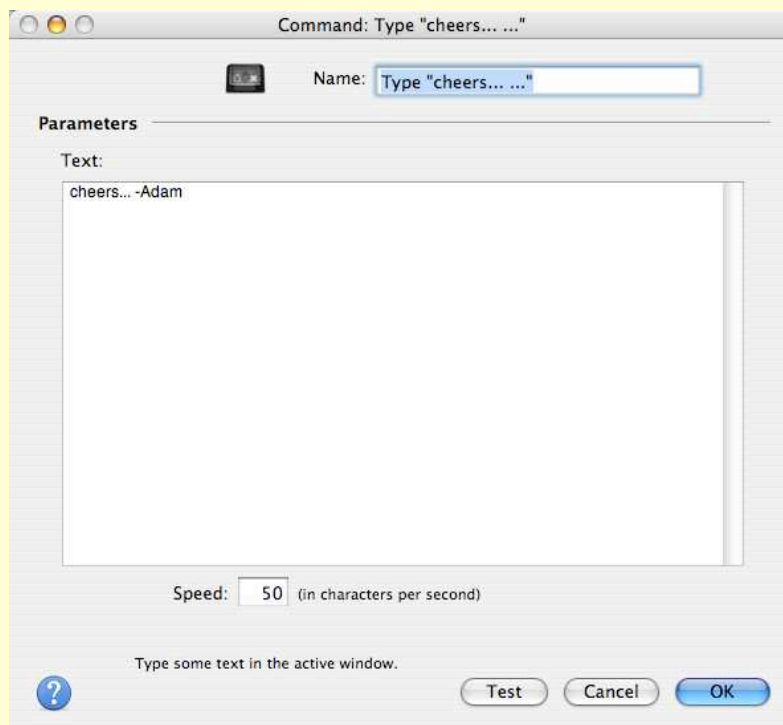
For instance, if your favorite word processor uses Command-Delete to delete a word at a time, and you prefer Option-Delete for that hotkey, it's easy to create a shortcut in iKey that uses a Type Key(s) command to press Command-Delete and uses a Keyboard Event launcher that you trigger by pressing Option-Delete.

**TIP** You would likely want to use an [Application in Front](#) context for such a remapping shortcut, since it wouldn't be necessary in most applications and it could be problematic in programs where Command-Delete had an entirely different function.

## Type Text

I'm extremely fond of the Type Text command, because it enables iKey to type any relatively small bit of text anywhere you can normally type (for larger chunks of text, use a [Paste Clipboard](#) command, because it's instantaneous). Typing text, along with application launching and switching, is a canonical use of an automation utility like iKey. For example, I have shortcuts that use Type Text commands to enter my “cheers... -Adam” signature when I finish an email message (**Figure 10.9**), to type my snail mail address, and to start URLs for me (the `<http://www.>` bit).

**FIGURE 10.9**



Enter the text you want iKey to type.

**NOTE** The Type Text command just simulates the keyboard, which means that any text it enters will use the style of the text next to the insertion point when you invoke the command. If you want to change styles, add other commands to do so.

**NOTE** If you want to create a number of shortcuts with Type Text commands that differ in relatively small ways, you might want consider duplicating one of them (with Edit > Duplicate; or Command-D) and changing each duplicate. However, before you do so, open iKey Editor's Preferences window and in the Shortcuts pane, check Create Copies of Commands when Duplicating a Shortcut. This setting ensures that iKey Editor creates a new Type Text command for each duplicated shortcut; otherwise changing the Type Text command for a duplicated shortcut would change it for every shortcut. I prefer to leave this setting turned off, except when duplicating shortcuts whose commands will need customization.

## Type Date

The Type Date command causes iKey to type the time and date in one of a large number of different formats. The tricky part of the Type Date command is that iKey gives you extremely fine-grained control over the format of your date text. So, if you want your date to read Monday, July 12, 2004, that's just as easy as having it read 12-Jul-04. In fact, that's why, when you create a Type Date command, iKey opens a Date Formats window showing you all the possible codes for different aspects of the date (see **Table 10.1**, which is sorted in order of increasingly large units of time—seconds, hours, minutes, days, months, years—followed by some utility codes). The Date Formats window closes automatically when you close the window for the Type Date command.

**Table 10.1: Date Format Codes**

Date Code	What It Means
%F	Milliseconds as a decimal number (000-999)
%1F	Same as %F but does not print the leading 0 for 1 through 99
%S	Seconds as a decimal number (00-59)
%1S	Same as %S but does not print the leading 0 for seconds 1 through 9
%M	Minute as a decimal number (00-59)
%1M	Same as %M but does not print the leading 0 for minutes 1 through 9

**Table 10.1: Date Format Codes (continued)**

Date Code	What It Means
%H	Hour based on a 24-hour clock as a decimal number (00-23)
%iH	Same as %H but does not print the leading 0 for hours 1 through 9
%I	Hour based on a 12-hour clock as a decimal number (01-12)
%iI	Same as %I but does not print the leading 0 for hours 1 through 9
%a	Abbreviated weekday name
&A	Full weekday name
%w	Weekday as a decimal number (0-6), where Sunday is 0
%d	Day of the month as a decimal number (01-31)
%e	Same as %d but does not print the leading 0 for days 1 through 9
%j	Day of the year as a decimal number (001-366)
%ij	Same as %j but does not print the leading 0 for days 1 through 99
%m	Month as a decimal number (01-12)
%1m	Same as %m but does not print the leading 0 for months 1 through 9
%b	Abbreviated month name
%B	Full month name
%Y	Year with century (such as 1990)
%y	Year without century (00-99)
%1y	Same as %y but does not print the leading 0 for years 1 through 9
%c	Shorthand for %X %x, the local format for date and time
%p	AM/PM designation for the local format for date and time
%x	Date using the local format for date and time
%X	Time using the local format for date and time
%Z	Time zone name (such as Pacific Daylight Time)

**Table 10.1: Date Format Codes (continued)**

Date Code	What It Means
%z	Time zone offset in hours and minutes from GMT (HHMM)
%%	A '%' character

**Table 10.1** is daunting, and looking up all those codes would be even more so, so iKey Editor provides an interface for building these date formats by choosing items from pop-up menus (**Figure 10.10**). For each code you want to insert in the Date Format field, choose the corresponding menu item.

**FIGURE 10.10**



Either enter the date format you want to use or build it by choosing items from the different pop-up menus.

What about niceties such as commas and spaces? Just enter those manually in the Date Format field until the example underneath looks the way you want.

**TIP** To make the example reflect your changes immediately, click it!

## Type File

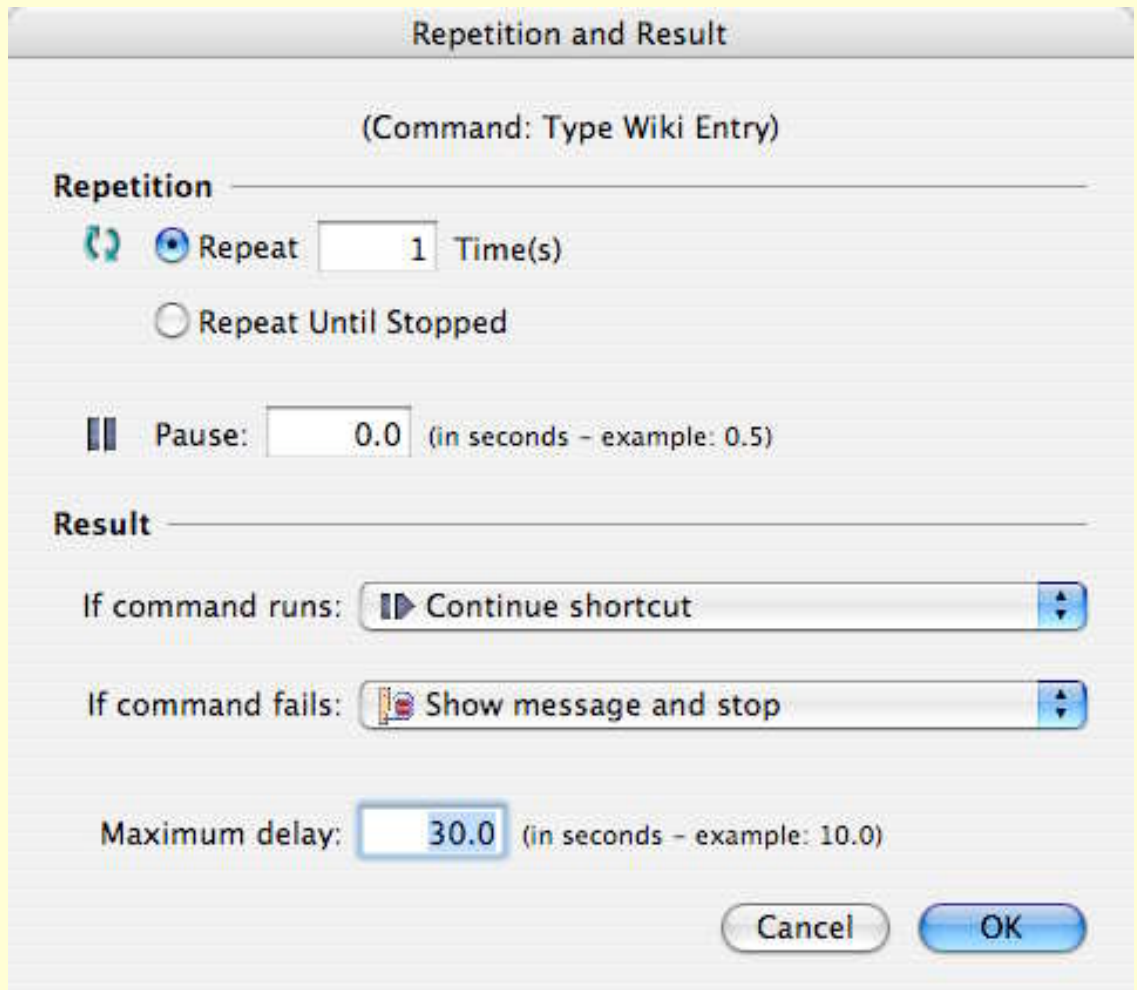
If, for some reason, you have a text file (and it must be a text file) that you want iKey to type into another document, you can use the Type File command. Just select the file you want iKey to type, along with the desired typing speed (here's a case where you might want to experiment with bumping up the speed, since a long file could take a long time for iKey to retype). It would probably be better to use iKey's [Paste Clipboard](#) command to paste the text.

**TIP** Normally, invoking a shortcut a second time while it's running cancels it, as does invoking a shortcut that uses the [Stop Shortcuts](#) command. But what if that doesn't work for some reason? You can always force quit iKey Engine by launching Activity Monitor, selecting the iKey process, and clicking Quit Process. But if even that's not possible because the shortcut prevents you from using the Mac, you can try using another Mac to connect to your main Mac via **ssh IPnumber**, use the **ps -aux** command to find the process ID (pid) of iKey, and then use the **kill pid** command to force quit iKey.

There is one important aspect of the Type File command to consider. When you're adding a Type File command to a shortcut, remember that all commands have a maximum delay, which you can (and should) increase for large files. To change the delay, double-click the Type File's Failure column in the shortcut window, and enter a higher number in the Maximum Delay field (**Figure 10.11**). Don't bother trying to calculate what it should be based on the typing speed—my experience is that different applications accept characters from iKey at different rates. Instead, just invoke the shortcut and let it run for 10 seconds (at which point it will fail), estimate how much of the file was typed in that time, and multiply out accordingly.



**FIGURE  
10.11**



Make sure to increase the maximum delay value for Type File commands that won't be able to complete typing in the default time of 10 seconds.

## Type Clipboard

As I noted previously, the only reason to use the Type Clipboard command is if you need to insert text from the clipboard into a dialog that doesn't accept pasting. Otherwise, always use a [Paste Clipboard](#) command.

## Menu Commands

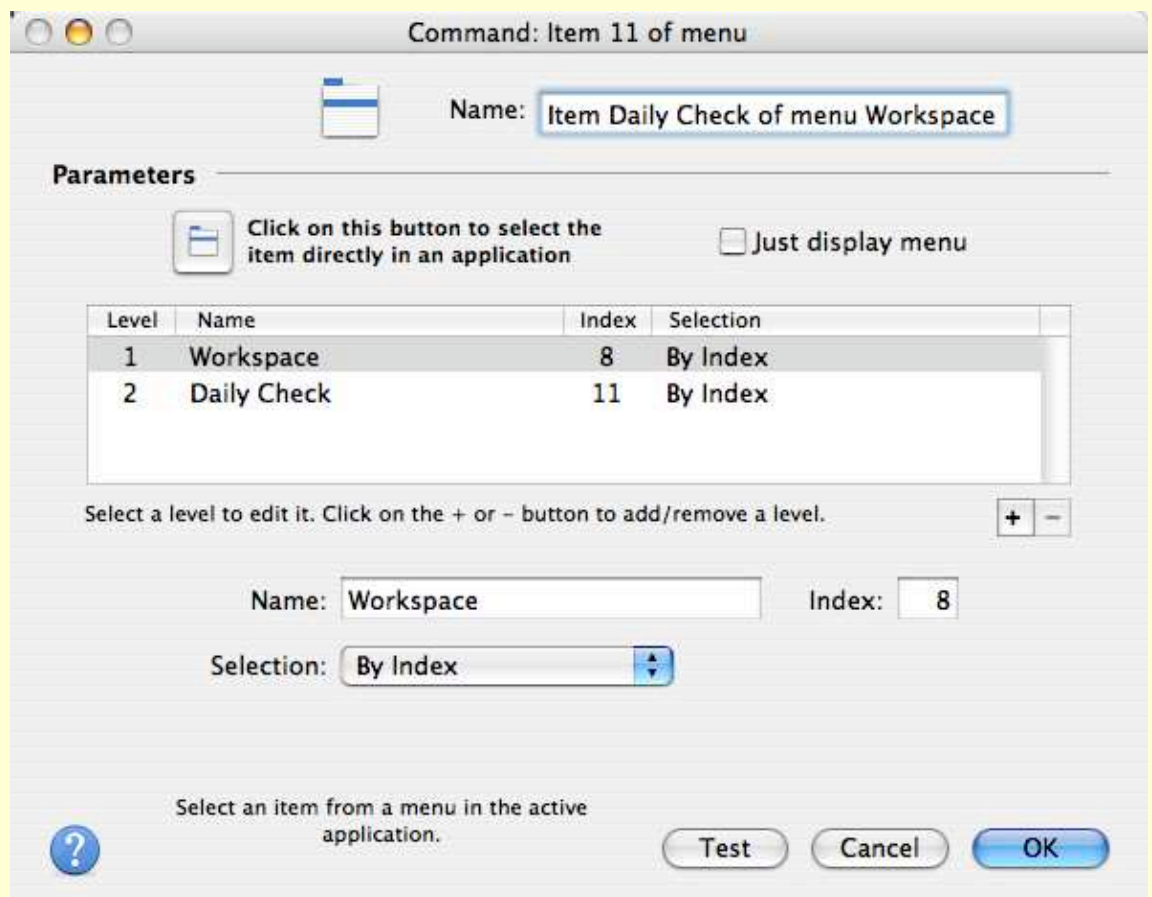
To control many applications, you must set iKey to choose items from menus, just as you would do with the mouse. iKey can choose items from top-level menus and from multiple levels of submenus.

**NOTE** iKey works only with native Mac OS X applications that support accessibility. That means that it may not work with all native Mac OS X applications, and it won't work with Classic applications. You may be able to work around this limitation with AppleScript.

### Select Menu Item

To make iKey choose a menu item, you can specify the menu and item manually or use iKey's menu item picker button (**Figure 10.12**).

**FIGURE 10.12**



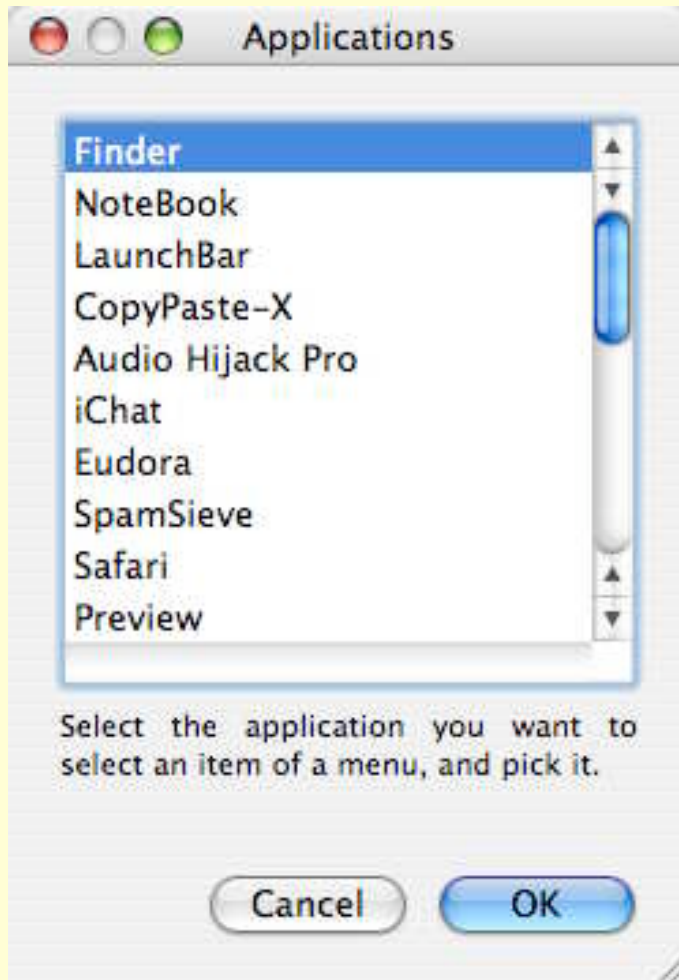
Specify the details about the menu item you want iKey to choose.

I recommend that you start with the menu item picker and modify its results if necessary. Follow these steps:

1. Click the menu item picker button.

iKey Editor displays a small dialog that provides a list of running applications (**Figure 10.13**).

**FIGURE 10.13**



Select the application in which you want to choose a menu item, and then click OK.

2. Select an application and click OK.

iKey Editor switches to the selected application.

3. Choose the desired menu item.

iKey Editor records your selection, filling in the appropriate values in the Name and Index fields, and setting the Selection pop-up menus to By Index.

4. Give the command a recognizable name and click OK.

**TIP** The index numbers reflect the position of the menu and its item, counting from the left for menus and from the top for menu items. So, the Apple menu is in menu index position 1, the application menu is in position 2, File is usually in position 3, and so on. Then, in the Apple menu, for instance, About this Mac is in index position 1, Software Update is in position 2, and so on. However, there's a catch. In the Apple menu (and possibly in other menus as well) there are hidden items that affect the index numbers, so you can't necessarily count the number of items in the menu to come up with the index number. Use the menu item picker instead.

You can configure the Select Menu Item command manually instead, either by name or index position. It's usually easier to let the menu item picker discover the index positions than to figure them out manually, so I usually rely on names when creating Select Menu Item commands manually. You don't even have to know the exact name of the menu or the menu item (which is good, since some change slightly based on context). The trick here is that the Selection pop-up menus tell iKey how to look for the menu and the menu item, and you can choose from six methods:

- **By Index:** Tells iKey to look only at the contents of the Index field.
- **By Reverse Index:** Tells iKey to count in the other direction (from the right for menus, starting with the Help menu usually, and from the bottom for menu items).
- **By Full Name:** Tells iKey to search for the menu or menu item by the exact name you enter in the Name field.
- **By Content of Name:** Tells iKey to search for the menu or menu item based on the partial text of the name.
- **By Name Beginning:** Tells iKey to search for the menu or menu item based on text that appears at the start of the name.
- **By Name End:** Tells iKey to search for the menu or menu item based on text that appears at the end of the name.

**NOTE** The Display Only Menu checkbox causes the command to ignore the Item section entirely and just display the specified menu, after which you can manually choose an item from it.

In iKey 2.1, the Select Submenu Item command disappeared, because Select Menu Item can now look into submenus on its own. Whereas **Figure 10.12** showed a second-level menu item, **Figure 10.14** shows a fourth-level menu item being chosen.

**FIGURE 10.14**

Command: Choose TidBITS Bookmark

Name: Choose TidBITS Bookmark

Parameters

☐ Click on this button to select the item directly in an application ☒ Just display menu

Level	Name	Index	Selection
1	Bookmarks	7	By Index
2	Old OmniWeb Defaults	14	By Index
3	Mac News	2	By Index
4	TidBITS	12	By Index

Select a level to edit it. Click on the + or - button to add/remove a level. + -

Name: Bookmarks Index: 7

Selection: By Index

Select an item from a menu in the active application.

Test Cancel OK

Specify details about the submenu item you want iKey to choose.

## Mouse Commands

Just as iKey can simulate a person typing on a keyboard, so too it can simulate a person with mouse in hand, moving, clicking, dragging, and scrolling. Unlike other commands, a single Simulate Mouse command can itself be made up of multiple movements, clicks, and drags, which makes it easier to combine the necessary actions.

**WARNING!** Simulation of mouse movements is tricky to get right. I strongly recommend that you avoid using the Simulate Mouse command if at all possible; rely on other iKey commands whenever you can.

### Simulate Mouse

When you create a Simulate Mouse command, you're presented with a dialog that you use to build a set of actions, choosing each from the Add pop-up menu (and removing unwanted ones by selecting them and clicking the Delete button or pressing Command-Delete). For the actions that actually move the mouse, the real work goes on in the Movement Parameters section of the dialog, where you specify exactly where the pointer should be, how far it should move from its current location, and so on (**Figure 10.15**).

**FIGURE  
10.15**

Command: Just mousing around

Name:

Parameters

Area of movement:

Movements	Pause
Move to 52,460 (T/L)	0.2
Move by 0,0	0.2
Drag to 0,0 (T/L)	0.2
Move by 0,0	0.2
Move to 1280,1024 (B/R)	0.2
Move to 0,0 (T/L)	0.2

Add

Delete

Movement parameters

Location: x:  y:

Origin:

Pause after:  (in seconds - 0.0 format)

Simulate mouse movements. ☐

Build your Simulate Mouse command by adding actions to the list and configuring each in turn.

For a complete list of the actions you can perform, see **Table 10.2**.

**Table 10.2: Mouse Actions**

Action	Function	Parameters
Click	Clicks the primary (left) mouse button at the current pointer location. Useful for basic clicking.	Length of time to pause before next action.
Right-Click	Clicks the right mouse button at the current pointer location.	Length of time to pause before next action.



**Table 10.2: Mouse Actions (continued)**

Action	Function	Parameters
Double-Click	Clicks the primary (left) mouse button twice at the current pointer location. Useful for opening files in the Finder.	Length of time to pause before next action.
Press Left Button	Press the left (primary) button down at the current pointer location (also known as mouseDown). Useful in situations that require the mouse button to stay down during a movement (like a menu). Pair with a Release Left Button action.	Length of time to pause before next action.
Release Left Button	Lets the left (primary) button up (also known as mouseUp). Releases the button pressed by a Press Left Button action.	Length of time to pause before next action.
Press Right Button	Press the right button down at the current pointer location (also known as mouseDown). Useful in situations that require the button to stay down during a movement (like a menu). Pair with a Release Right Button action.	Length of time to pause before next action.
Release Right Button	Lets the right button up (also known as mouseUp). Releases the button pressed by a Press Right Button action.	Length of time to pause before next action.
Move To	Move the pointer to a specific location as described by X and Y coordinates given a particular origin.	Location as expressed in an X/Y coordinate system.  Origin of where 0/0 should be in the X/Y coordinate system.  Length of time to pause before next action.



**Table 10.2: Mouse Actions (continued)**

Action	Function	Parameters
Move By	Move the pointer by adding or subtracting from the current X/Y location. Useful for moving the pointer by a set amount rather than to a specific location.	Distance to move in both X and Y directions from the current location.  Length of time to pause before next action.
Save Location	Saves the location for later use by a Move to Saved Location action.	Length of time to pause before next action.
Move to Saved Location	Moves the pointer to the previously saved location.	Length of time to pause before next action.
Drag To	Clicks and drags (moves with the mouse button down) to a specific location as described by X and Y coordinates given a particular origin. Useful for moving objects around in the Finder or graphics applications.	Location as expressed in an X/Y coordinate system.  Origin of where 0/0 should be in the X/Y coordinate system.  Length of time to pause before next action.
Drag By	Clicks and drags (moves with the mouse button down) to a location by adding or subtracting from the current X/Y location. Useful for moving the pointer by a set amount rather than to a specific location.	Distance to move in both X and Y directions from the current location.  Length of time to pause before next action.
Turn Wheel Up	Scrolls the current window up by a specified number of lines.	Number of lines to scroll up.  Length of time to pause before next action.
Turn Wheel Down	Scrolls the current window down by a specified number of lines.	Number of lines to scroll down.  Length of time to pause before next action.

Most of these mouse actions are self-explanatory—if you’ve been using a Macintosh for more than a few days, you undoubtedly understand clicking and double-clicking and so on. However, you probably don’t think of moving the pointer in terms of an X/Y coordinate system, which is necessary to understand how iKey programs mouse movements.

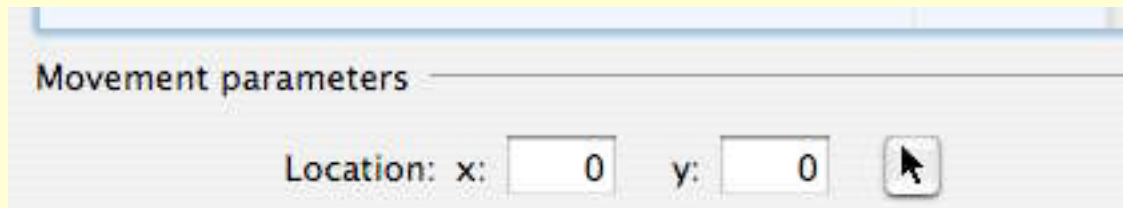
Think of your screen as a big piece of graph paper. Each pixel is a square on the graph paper, and if you were to write from 0 to 1024 across the top and numbers from 0 to 1280 going down the left side (assuming your monitor runs at 1280 by 1024 like my Apple 17-inch LCD does), you’d have created an X/Y coordinate system where you can identify every pixel by giving its X (the number along the top) and Y (down the side) coordinates. So every mouse location can be specified by a set of X/Y coordinates, and you can also move the pointer by an offset—adding or subtracting from the X and Y coordinates.

In this example, the origin, or 0/0, is at the top left of the piece of graph paper. But you could just as easily number up the side or from right to left instead of left to right—that’s just using a different origin. the Move To and Drag To mouse actions let you choose where to set your origin from a pop-up menu. It makes no difference what you choose, so I recommend either sticking with Top/Left because it’s how most people read (left to right, top to bottom) and will thus be the most intuitive.

**TIP** What if you have two monitors? It depends on which monitor is your primary monitor. My right monitor is my primary, so all X values on my left monitor are negative numbers. If my left monitor was my primary, the X values would all be positive, and they would go to 2560 instead of 1280.

You might be wondering how you would know what coordinates correspond to an object you can see on the screen. You could guess, but unless you’re way better at estimating pixel size than I am, it’s not worth the effort. Instead, notice the arrow pointer button that appears in the Move To and Drag To mouse actions (**Figure 10.16**).

**FIGURE  
10.16**



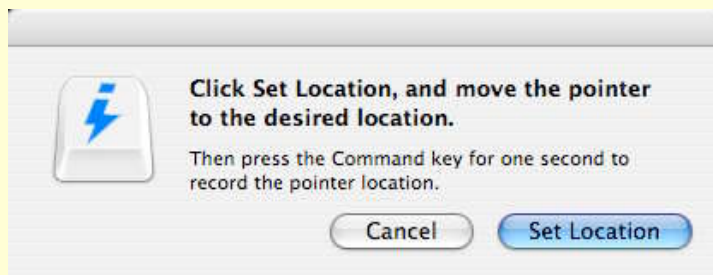
The Move To and Drag To mouse actions provide an arrow pointer button that helps you identify pointer locations in the X/Y coordinate system.

To determine the mouse location, follow these steps:

1. In a Move To or Drag To action, click the arrow pointer button.

iKey Editor displays a dialog with instructions (**Figure 10.17**).

**FIGURE  
10.17**



When you're ready to record a pointer location, click Set Location, move the pointer to the desired location, and press the Command key for 1 second.

2. Click Set Location.
3. Move the pointer to the desired location. You can click if necessary (if you're moving to a menu location, for instance).
4. Press the Command key for 1 second to record the location.

Sometimes you need to restrict the area of movement to a specific window. That's what the Area of Movement pop-up menu enables you to do. Choosing Screen doesn't limit the area at all, whereas Main Window limits it to the currently active window and Focused Window limits it to the window that has the focus. You might wonder how these last two could be different, but the reason is that a sheet (the dialogs that drop down from the top of a window) can be the focused window even when it's not the main window.

## Network Connection

iKey offers three commands related to networking: two that affect the Locations setting in the Network preference pane and one that mounts servers.

### Menu of Network Locations

Use this command to display a menu of the available network locations; choosing one switches the settings in your Network preference pane. This command mimics the functionality of the Locations menu in the Network preference pane itself, although you can't create or edit locations from iKey's Network Locations menu.

In most cases, the Set Network Location command, next, is probably more efficient. I'd use the Menu of Network Locations command only if I had more than four or five locations that I regularly switched among.

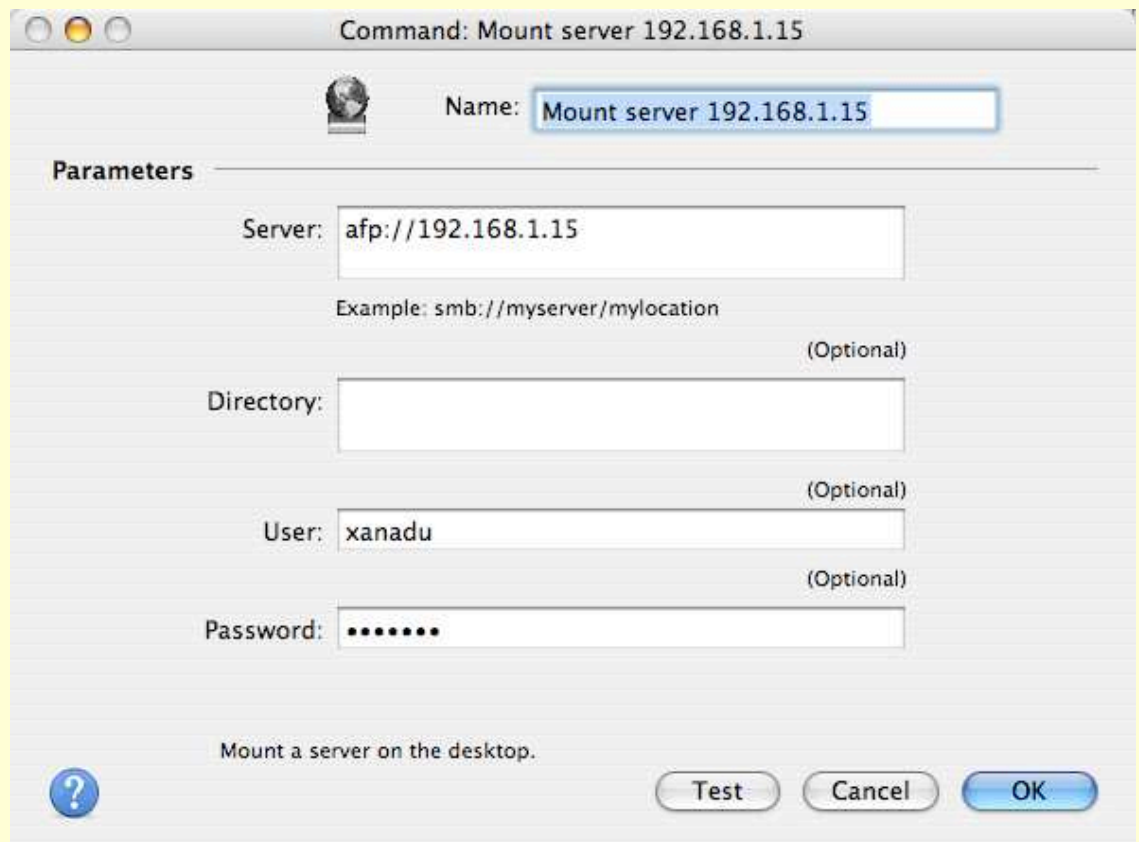
### Set Network Location

If you have only a few network locations, instead of using the Menu of Network Locations command and choosing one manually, it might make sense to use the Set Network Location command to set the location automatically. You can think of this command as iKey automatically choosing a location from the Network Location menu.

### Mount Server

iKey's third Network command helps you automatically mount network servers. In it, you enter the server's URL, the desired starting directory, the user name, and the password (**Figure 10.18**). The last three fields are all optional, assuming of course that your server doesn't require a user name and password as most do.

**FIGURE  
10.18**



Enter the details necessary to connect to your server.

Although I can't test all these server types (not having servers that use all these protocols), I believe that the Mount Server command should work with servers using the protocols shown in **Table 10.3**.

**Table 10.3: Protocols Likely Supported by Mount Server**

Protocol	Schema for URL	Examples
AppleShare	afp://	afp://officeserver.local afp://afp.example.com
FTP	ftp://	ftp://ftp.example.com/
WebDAV	http://	http://rwgl.foo.com/pages/
Samba	smb://	smb://192.168.1.4/SharedDocs smb://UNPUNDPC/ShareDocs
NFS	nfs://	nfs://63.211.215.7/usr/www

## Printer Commands

Just as iKey enables you to control your network settings, it enables you to control your default printer settings. Other commands help you automate printing batches of documents.

**NOTE** With the Menu of Printers and Set Default Printer commands you'll see your Printer Setup Utility launch and quit in the process of changing the active printer.

### Menu of Printers

Use this command to display a menu of available printers; choosing one makes it the default printer. This command mimics the function of the Printers menu in the Print dialog, although you can't edit the printer list from iKey's Printers menu.

### Set Default Printer

If you have only a few printers, instead of using the Menu of Printers command and choosing one manually, it might make sense instead to use the Set Default Printer command to set the default printer automatically. You can think of this command as iKey automatically choosing a particular printer from the Printers menu.

### Print Documents in Folder

It's easy to imagine an automated process (automated by iKey or some other tool, perhaps a Unix utility) that creates a number of files you would want to print in a batch. That's where the Print Documents in Folder command comes in. It takes only two parameters: the printer you wish to use (leave blank to use your default printer) and the folder containing the documents to print.

The Print Documents in Folder command relies on the application that created each document to print (and because of that, the documents do not need to be all of the same type). As a result, when you invoke this command, you'll see a lot of application launching and screen flashing as documents are opened and printed.

**WARNING!** Not all applications work properly with iKey's AppleScript-based approach to printing with the Print Documents in Folder command. Some applications may not work at all, and others may require you to click the Print button in the Print dialog. If this problem is a snag for you, look into working around it with a shortcut that opens and prints the documents using [Menu](#) or [Keyboard](#) commands.

Also keep in mind that the documents in question must be associated with an application that can print them; it's conceivable that you would end up with documents that either don't open in a particular application automatically, or that open in one that can't be told to print by iKey.

**TIP** Remember that all commands have a maximum delay, which you probably will need to increase when printing documents in a folder, since iKey has to open each document and ask the application to print it. To change the delay, double-click the Type File command's Failure column in the shortcut window, and enter a much higher number in the Maximum Delay field.

### Print Selected Documents

The Print Documents in Folder does print everything in a specified folder, but sometimes you may want to print a few select documents. That's where the Print Selected Documents command comes in. It simply opens the documents selected in the Finder in their creating applications and asks those applications to print. You can choose which printer should handle the job, but that's it.

**WARNING!** Not all applications work properly with iKey's AppleScript-based approach to printing with the Print Selected Documents command. Some applications may not work at all, and others may require you to click the Print button in the Print dialog. If this problem is a snag for you, look into working around it with a shortcut that opens and prints the documents using [Menu](#) or [Keyboard](#) commands.

You may want to increase the maximum delay on the Print Selected Documents command, particularly if you plan to print more than a couple documents at a time or if it takes a long time to launch the applications that will be doing the printing.



## Script Commands

iKey is wonderful for making shortcuts for commonly performed actions, but it's not a programming language like AppleScript, which can be used to automate significantly more complex tasks, complete with user interaction. However iKey can run AppleScript scripts with two commands: Run Script from File and Run Script from Text.

The difference between the two is simple. With Run Script from File, you point iKey at a file, and the command runs the AppleScript script in that file. With Run Script from Text, you type the script into the command window.

**TIP** You can also control iKey from within AppleScript, which may be helpful for actions in a script that are easiest in iKey (such as simulating the mouse movement). The AppleScript commands you can use are as follows (their function should be obvious, though note that **display menu** shows the menu in question at the current pointer location):

```
run command "commandName"
run shortcut "shortcutName"
display menu "menuName"
show palette "paletteName"
hide palette "paletteName"
```

## Run Script from File

Numerous AppleScript scripts come with the Mac OS, and oodles more are available for download. By selecting one of these scripts to use with the Run Script from File command, you can have iKey execute the script when you invoke it via a shortcut, menu, or palette, or you can build it into a multiple-step shortcut. Click the Test button to try the script, and click the Edit Script button to open the script in Script Editor.

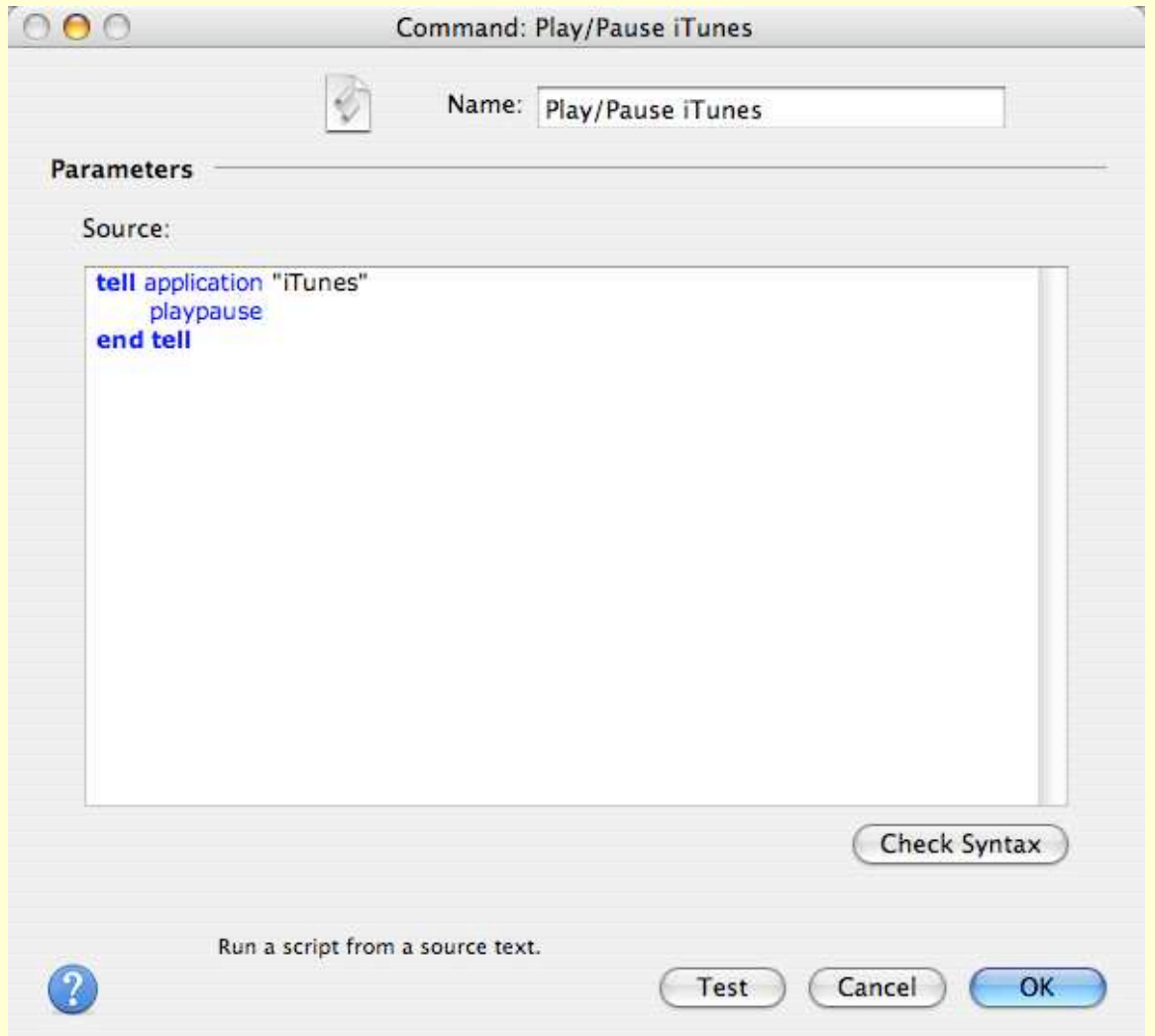
It's best to use Run Script from File only with scripts that you already know work properly, especially if they are long and better edited using the tools provided in Script Editor.



## Run Script from Text

The other option for running an AppleScript script from within iKey is Run Script from Text, which lets you type (or more likely, paste) a short script into a text field, complete with syntax coloring (**Figure 10.19**). Click the Test button to try the script, and click the Check Syntax button to verify that it's syntactically correct.

**FIGURE 10.19**



Enter your script in the text field and use the Check Syntax and Test buttons to verify that it's correct and works as you expect.

iKey Editor doesn't provide as good an environment for editing scripts as Script Editor, so I recommend restricting yourself to relatively short scripts when using Run Script from Text.

**TIP** In particular, I like having simple scripts that tell iTunes to play or pause, or to move on to the next track. They're tiny, but highly effective.

## Sound Commands

iKey offers several commands for playing sounds and adjusting the volume.

### Play Sound File

The Play Sound File command plays the sound file you select—nothing more. You can click Test button to listen to the sound.

For the most part, you will probably want to use Play Sound File to play system beeps in AIFF or WAV format, although it will happily play longer sounds, such as MP3 or AAC music files.

**TIP** To find the alert sounds provided by Apple with Mac OS X, look in **/System/Library/Sounds**. You can download additional beep sounds for free on the Internet from sites like Partners In Rhyme. <http://www.partnersinrhyme.com/soundfx/PDsoundfx/beep.shtml>

### Volume Up, Volume Down

Use the Volume Up and Volume Down commands to raise and lower the volume, just as you can do by pressing the volume keys on current Mac keyboards. As with other commands that replicate easily performed system actions, the main utility of these commands is within multiple-step shortcuts or for keyboards that lack the volume buttons.

**TIP** Since these commands change the volume by only a single step, you will either want to invoke these commands manually multiple times, or set them to repeat multiple times in the shortcut.

### Volume Mute

Use the Volume Mute command to toggle muting on and off, while remembering the current volume level setting when the sound comes back on. I could easily imagine using this command in a multiple-step shortcut that worked with another program to dial the phone; you wouldn't want the other person to hear your music.

## System Commands

iKey provides numerous commands for controlling your system, and although you might wonder why you'd want to use iKey's Open/Close Tray command instead of pressing the Eject key on the keyboard, remember that you can put multiple commands in a single shortcut and that not everyone has a keyboard with an Eject key.

**NOTE** Once again, I've grouped related commands rather than showing them purely alphabetically.

### Open/Close Tray

Use this command to open or close your optical drive. It's exactly the same as pressing the Eject key on your keyboard, so it's mostly useful in multiple-step shortcuts or in working with a keyboard that lacks an Eject key or whose Eject key isn't functioning properly.

### Launch System Preferences

Use this command to launch System Preferences, exactly as happens if you choose System Preferences from the Apple menu or click its icon in the Dock.

### Menu of Preference Panes

It's often clumsy to open System Preferences and click a particular preference pane since there are so many. Many people don't realize that the View menu in System Preferences actually lists all the preference panes; choosing one from that list is often easier than clicking its icon.

iKey's Menu of Preference Panes command lets you provide access to that list of preference panels anywhere you like (at the pointer, in a menu, or from a palette button), making it easy to choose a particular one at any time without first having to open System Preferences.

### Open Preference Pane

If you're like most people, you probably use only a few preference panes most of the time (Network, Displays, and Accounts, perhaps). Why waste time clicking icons or going through a menu (as in the previous command) when you could open a particular one directly? That's what the Open Preference Pane command does for you; just choose the one you want from the pop-up menu.

## Log Out

Use this command to log the current user out and show the login window. You might think about using it in a multiple-step shortcut that backs up certain important files and then logs you out for security reasons.

## Restart

As you might expect, this command restarts the Mac, just like choosing Restart from the Apple menu or pressing Command-Power and clicking the Restart button. It's primarily useful in multiple-step shortcuts (as the last step, of course). Note that the Mac starts restarting immediately; iKey doesn't ask for confirmation (although individual applications with unsaved work will ask you to save first, and clicking Cancel aborts the restart process).

## Shut Down

This command is as obvious as Restart, though it shuts the Mac down, just like choosing Shut Down from the Apple menu or pressing Command-Power and clicking the Shut Down button. It's primarily useful in multiple-step shortcuts (as the last step, of course). Note that the Mac starts shutting down immediately without presenting a confirmation dialog (individual applications with unsaved work will ask you to save first, and clicking Cancel aborts the shutdown process).

## Sleep

This command puts the Mac to sleep immediately, just like choosing Sleep from the Apple menu or pressing Control-Power and clicking the Sleep button. It's primarily useful in multiple-step shortcuts (as the last step, of course).

iKey 2.1 extends the function of the Sleep command by providing additional options that enable you to set your Mac to wake up automatically after a particular delay or at a specified date and time. This would let you, for instance, put your Mac to sleep with a hotkey when you're done for the day, and know that it would wake up automatically at 2:30 AM for a nightly backup. You could even schedule multiple sleep/wake cycles, so it could then go back to sleep (via a timed shortcut) 2 hours later, and wake up again at 7:50 AM before you get to work at 8:00 AM.

**NOTE** Administrator authorization is required for the automatic wakeup.

### **Run Screen Saver**

Most of the time, you want a screen saver to come on automatically after your Mac has been idle for a particular amount of time, and you set that in the Desktop & Screen Saver preference pane. But what if you want to invoke the screen saver manually? You can set a hot corner in the Desktop & Screen Saver preference pane, and you can create a Simulate Mouse shortcut in iKey if you wish to invoke the screen saver within a shortcut.

Why am I telling you this if iKey has, as it does, a Run Screen Saver command? Because that command doesn't work in Mac OS X 10.3 due to a bug in Mac OS X. If you're using a previous version of Mac OS X, it may work for you, and it's likely that Apple will fix the bug eventually. In the meantime, stick with the hot corner.

### **Show/Hide Dock**

If you don't want to see the Dock at all times, you can hide it. There's a menu item for doing this in Apple > Dock, along with a hotkey (Command-Option-D), and with the Show/Hide Dock command in iKey, you can build this action into other shortcuts.

**TIP** I can easily imagine a shortcut that launches an application that needs the full screen, and which thus would benefit from using the Show/Hide Dock command to hide the Dock first.

### **Start Classic, Stop Classic**

These two commands do exactly what their names imply: they start and stop the Classic environment, just as you would normally do with the command in the Classic preference pane or in the Classic menu. Since Classic can take a while to start, building it into a macro that runs at a particular time might be a time-saver.

## Unix Commands

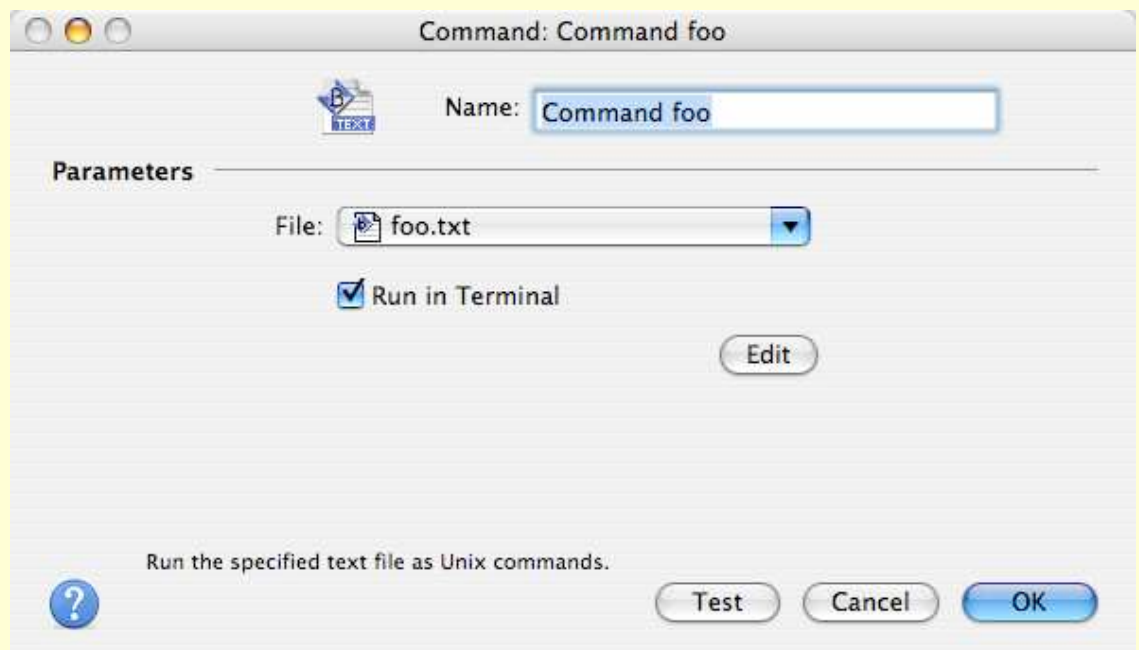
iKey provides a pair of commands for automating the command line—that's right, you can use iKey to control Mac OS X's Unix underpinnings. As with AppleScript scripts, you can run Unix commands either from a file or by typing them into iKey Editor.

### Run Commands from File

Not being a major Unix user, I can't see much use for this command, since I can't imagine wanting to run so many Unix commands that it was somehow more efficient to do so from a file than from within iKey using the Run Commands from Text command. One reason might be if you can generate the file in some other way, and then use iKey to execute it.

When creating a Run Commands from File command, you must select the text file that contains the command (create this first), and decide if you want the commands to run in Terminal, where you can see what they do, or not (**Figure 10.20**). An Edit button enables you to edit the file; to run it from within the command window, click Test.

**FIGURE 10.20**



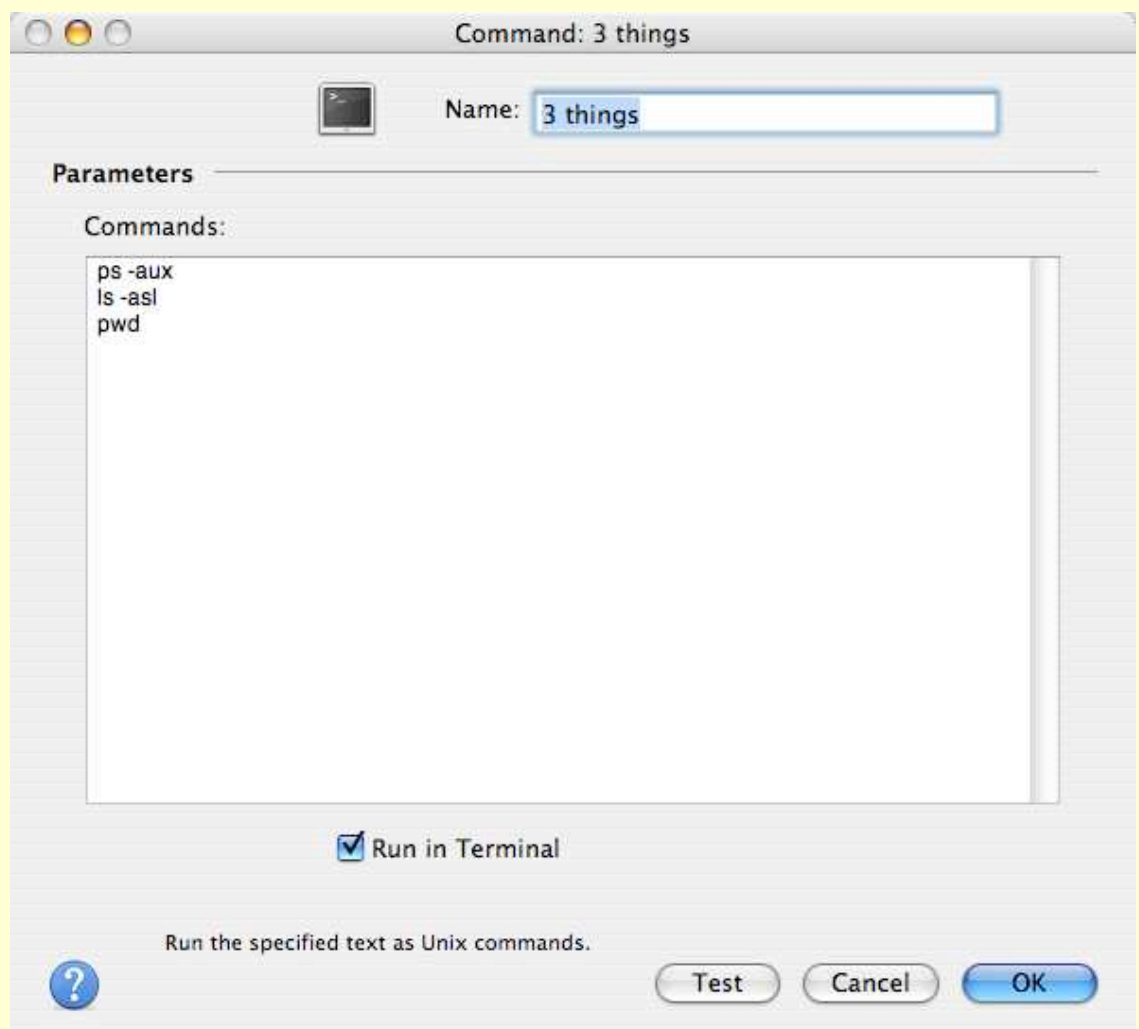
Select your file of Unix commands and decide if you want to see feedback from their execution in Terminal.

**NOTE** Don't confuse the types of files this command executes with Unix shell scripts, which are written in a specific language and which are executed by the Unix shell, not by iKey. These files are simply collections of Unix commands that iKey executes, one after another.

## Run Commands from Text

More people will probably combine a number of Unix commands directly within iKey, using the Run Commands from Text command. All you do is enter your Unix commands and decide if you want to see the results in Terminal (**Figure 10.21**). In most cases, you probably do want to see the results of the commands in Terminal, since it's the main way you'll know if everything worked as you expect.

**FIGURE 10.21**



Enter a set of Unix commands to be executed at the command line. Make sure to check Run in Terminal if you want to see the results.



## Utility Commands

When you're creating multiple-step shortcuts, you must keep in mind the fact that some commands take longer than others, and, more to the point, some commands may not execute instantly. Imagine that you have a shortcut that launches an application, opens a new window, and starts typing text. If you create the shortcut with just the three basic commands necessary, it likely won't work because the application takes some time to launch, and in that time, the other two commands will try to execute. But they'll fail, because the command to create a new window can't work until the application has finished launching, and the command to type text can't work until the window is up.

iKey's Utility commands help you work around these problems, by providing commands that let you make shortcuts that can wait until certain conditions are true. Note that I've collected similar commands into groups below.

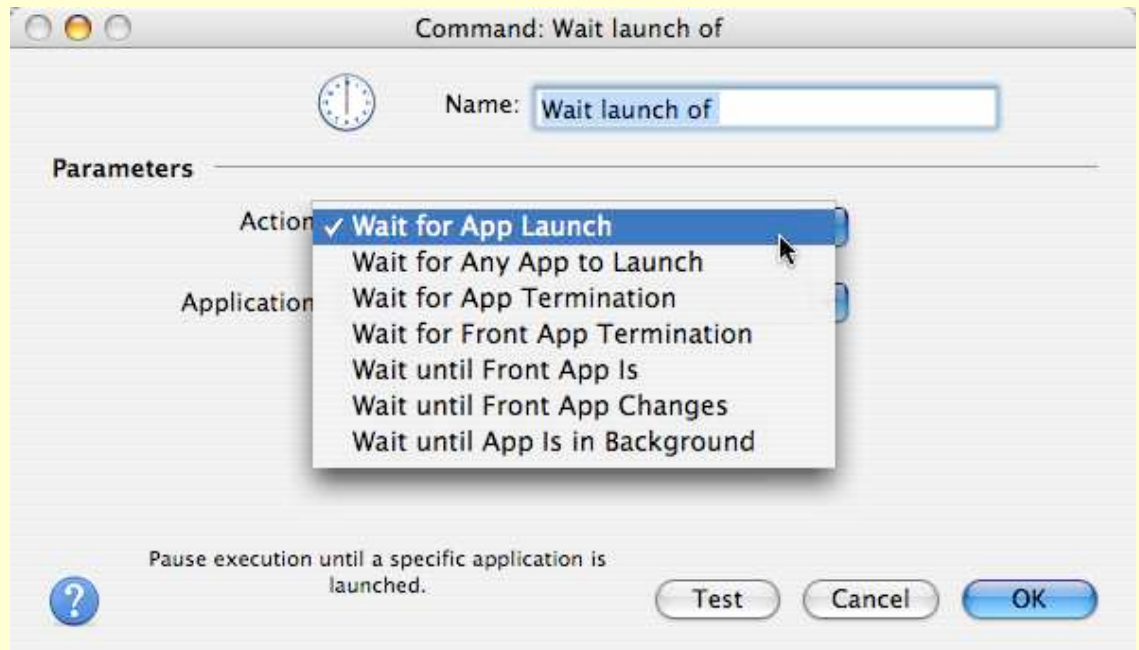
### Wait Time

The easiest way to pause a shortcut, though by far the least accurate, is the Wait Time command, which simply pauses execution for a specified number of seconds. The problem with the Wait Time command is that actions can take different amounts of time depending on what else is happening on the Mac. If you must use Wait Time in a shortcut, I recommend erring on the longer waits until you have a sense of how long the action is likely to take in different conditions.

### Wait for application events

More accurate are the wait commands that watch for some event related to a particular application. In each case, choose the desired action from the Action pop-up menu and, if appropriate, select a particular application (**Figure 10.22**).

**FIGURE  
10.22**



Choose what to wait for from the Action pop-up menu, and if appropriate, select a particular application using the Application pop-up menu.

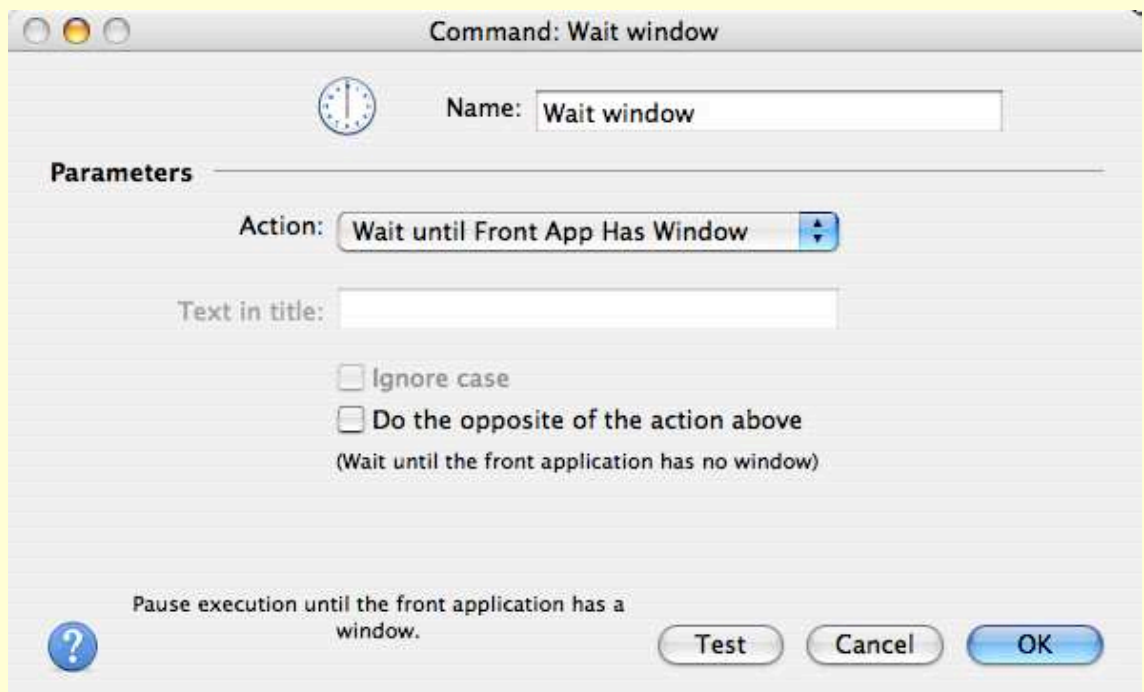
- **Wait for App Launch:** Pause until the specified application has finished launching.
- **Wait for Any App to Launch:** Pause until any application has launched.
- **Wait for App Termination:** Pause until the specified application has quit.
- **Wait for Front App Termination:** Pause until the current front application has quit.
- **Wait until Front App Is:** Pause until the specified application is the front application.
- **Wait until Front App Changes:** Pause until the current front application, whatever it may be, is no longer front.
- **Wait until App Is in Background:** Pause until the specified application is no longer the front application.

### **Wait for window events**

The wait commands that wait for application events are highly useful, but for more finely grained control, turn to the wait commands that

wait for window events. For each of these commands, choose an action from the Action pop-up menu and, where appropriate, enter the text you want to match in the window title. If you're not sure of the case, check the Ignore Case checkbox (**Figure 10.23**).

**FIGURE 10.23**



Choose what to wait for from the Action pop-up menu, and, if appropriate, enter the text to match in the window title in the Text in Title field.

- **Wait until Front App Has Window:** Pause until the front application is displaying any window.
- **Wait until Front Window Changes:** Pause until the front window's title changes in some way.
- **Wait for Window with Title:** Pause until the front window's title exactly matches the specified text.
- **Wait for Window Title Contains:** Pause until the front window's title contains the specified text.
- **Wait for Window Title Starts With:** Pause until the front window's title starts with the specified text.
- **Wait for Window Title Ends With:** Pause until the front window's title ends with the specified text.

**NOTE** For each of the commands above, except for Wait until Front Window Changes, a new checkbox labeled “Do the opposite of the action above” enables you to reverse the function of the command. In other words, if you want to pause execution of a shortcut until a window title *doesn't* end with a particular bit of text, select that checkbox.

## Display Dialog

If you're not sure exactly what event your shortcut should wait for, or for how long you should wait, you can use the Display Dialog command to make iKey show a standard dialog that asks the user whether or not to continue execution of the shortcut.

The Display Dialog command is quite configurable. You can set one or two messages, and include one or two buttons (leave the Button 2 field blank to have just one button in your dialog). You can also set the action for each button: whether the shortcut should keep running or stop (**Figure 10.24**). Message 1 always appears in bold type; and Button 1 is always a default button.

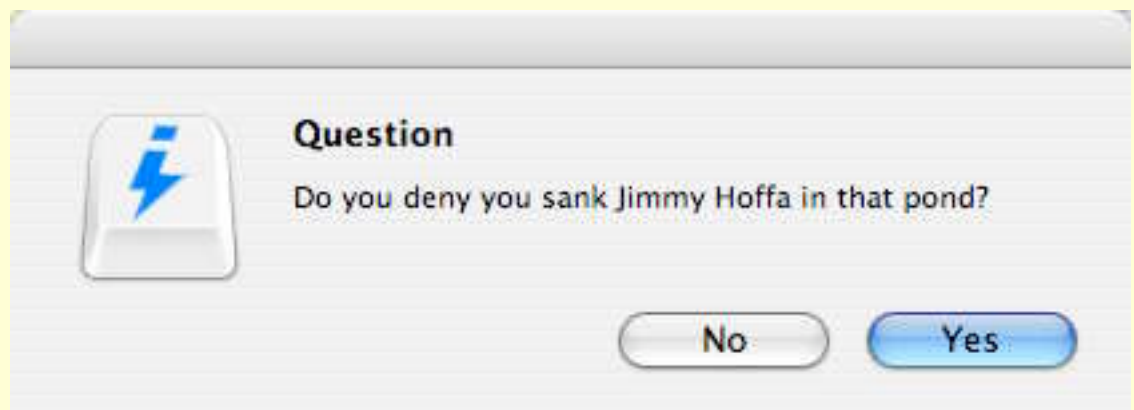
**FIGURE  
10.24**



Configure your dialog with up to two lines of text and two buttons.

You can see what your dialog looks like by clicking the test button; the dialog configured in **Figure 10.24** appears in **Figure 10.25**.

**FIGURE  
10.25**



The dialog produced by the settings shown in the previous figure.

## Window Commands

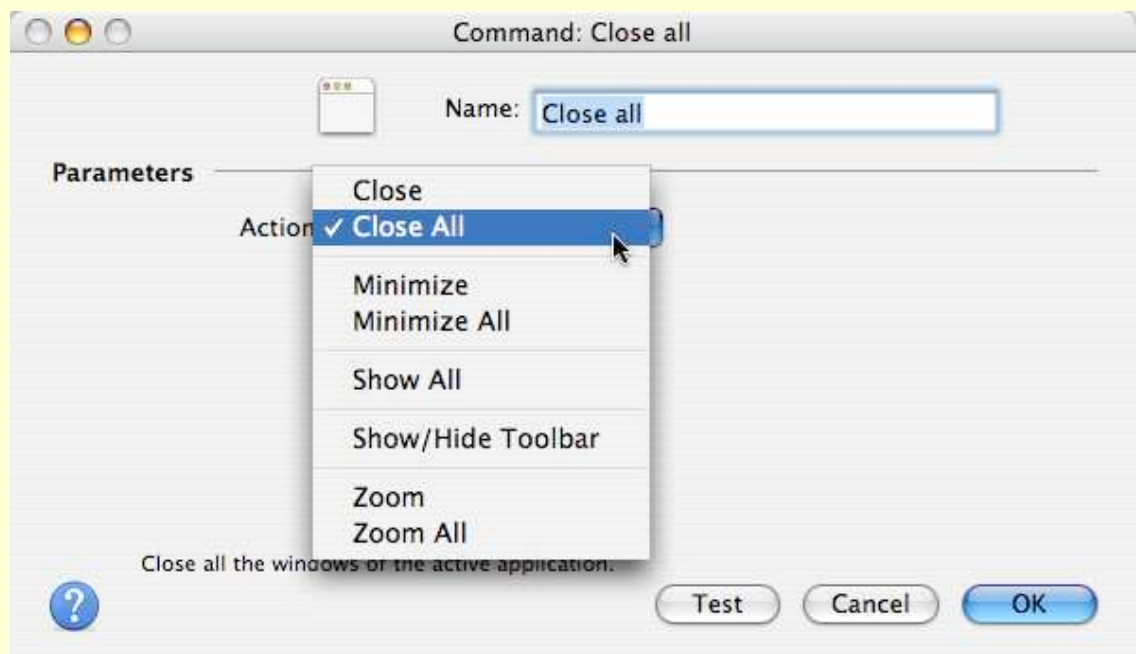
iKey provides a slew of commands you can use to control windows, much as you'd do with the mouse normally. I've grouped them into related categories in this section.

**NOTE** iKey's window commands work only with native Mac OS X applications that support accessibility. That means that you must select the Enable Access for Assistive Devices checkbox in the Universal Access preference pane, these commands may not work with all native Mac OS X applications, and they won't work with Classic applications.

### Window control commands

The window control commands duplicate the function of the window interface elements in the title bar of the window (**Figure 10.26**).

**FIGURE 10.26**



Choose the desired action from the Action pop-up menu.

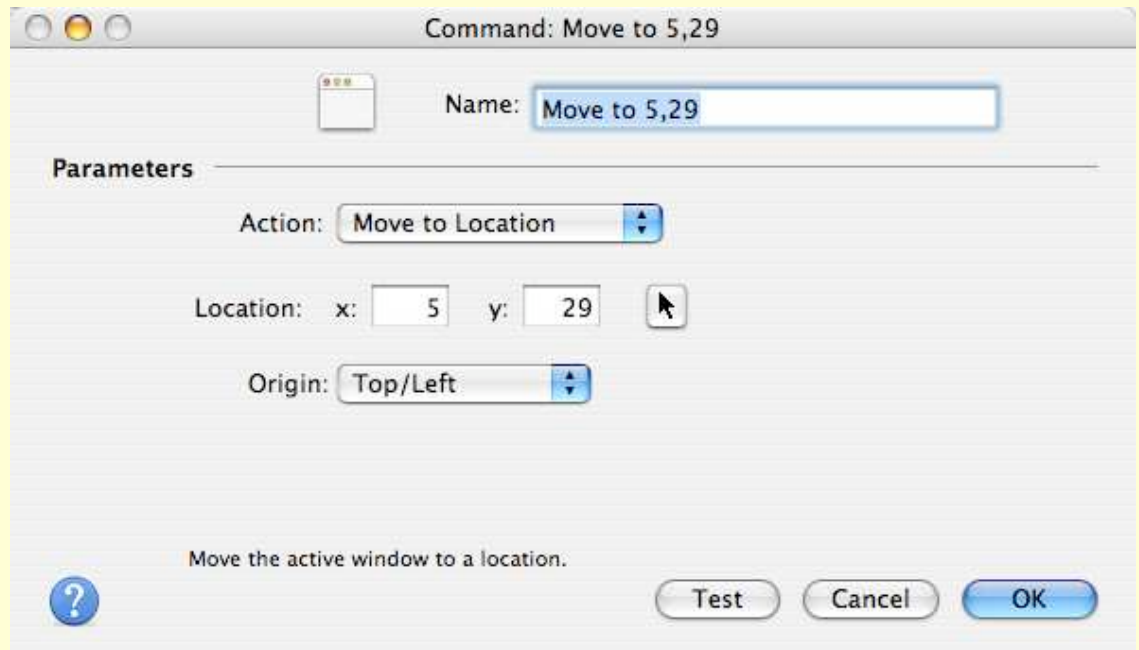
- **Close Window:** Closes the current window.
- **Close All Windows:** Closes all open windows in the current application. This command is equivalent to Option-clicking a window's close button.

- **Zoom Window:** Zooms the current window to the largest appropriate size.
- **Zoom All Windows:** Zooms all windows in the current application to the largest appropriate size.
- **Minimize Window:** Minimizes the current window to an icon in the Dock.
- **Minimize All Windows:** Minimizes all windows in the current application to icons in the Dock. This is equivalent to Option-clicking a window's minimize button.
- **Show All Windows:** Expands any minimized windows.
- **Show/Hide Toolbar:** Toggles whether or not the toolbar is showing in the current window. This command is equivalent to clicking the lozenge button at the top of windows with toolbars.

### **Window moving commands**

Use these commands to move windows on the screen, just as you'd drag them with the pointer. For the latter two commands, you must specify the location, either in absolute values or in a relative distance from the current location (**Figure 10.27**).

**FIGURE  
10.27**



Choose the desired action from the Action pop-up menu and the desired point of origin from the Origin pop-up menu, and enter the location or distance in the Location fields.

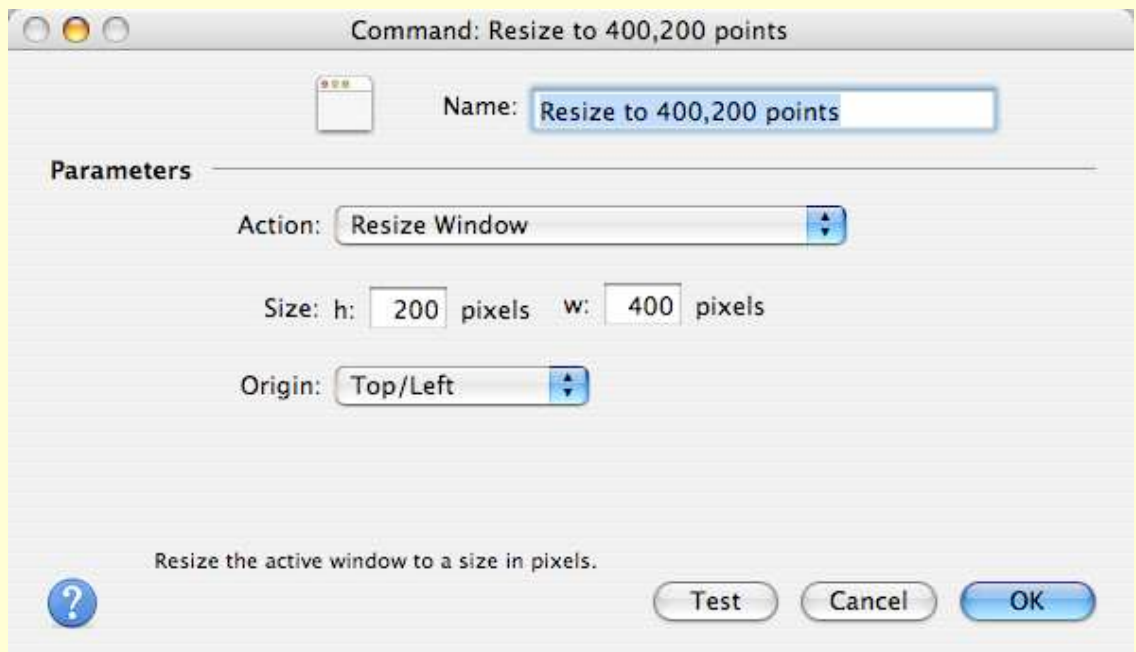
- **Move Window to Center:** Moves the current window to the center of the screen in the horizontal or vertical planes, or both. Check the Horizontally and Vertically buttons as desired.
- **Move Window to Location:** Moves the current window to the location on screen as specified in X/Y coordinates. Set the point of origin as desired using the Origin pop-up menu.
- **Move Window by Distance:** Moves the current window a specified distance from its current location.



## Window resizing commands

Use these commands to change the size of the current window, much as you would do by dragging the size box in the lower right corner of a window (**Figure 10.28**):

**FIGURE 10.28**



Choose the desired action from the Action pop-up menu; then enter the appropriate variables for how you want to resize the window.

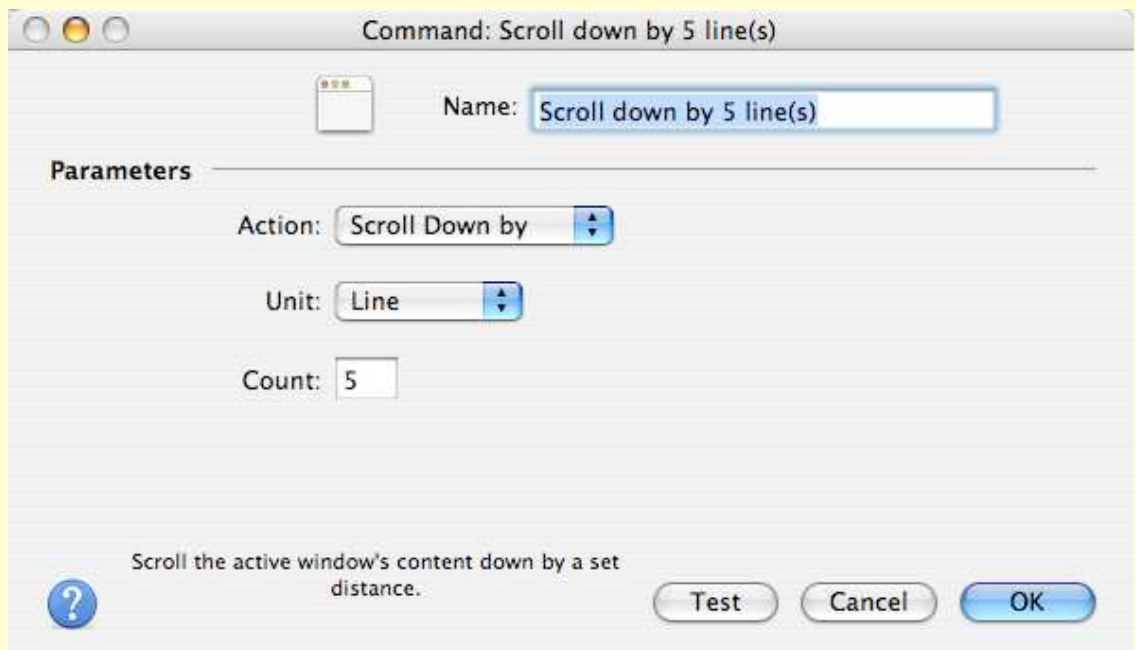
- **Resize Window:** Resizes the current window to a specified size in pixels, pegging one corner to its current position according to your choice from the Origin pop-up menu.
- **Resize Window Relative to Original:** Resizes the window by specified amounts, starting from the current size. Use this command to make a window 100 pixels taller, for instance.
- **Resize Window Relative to Screen:** Resizes the window by percentages relative to the size of the screen. This command enables you to resize windows to, say, 50 percent of the size of the screen in height, and 30 percent of the size of the screen in width.
- **Resize Window by Percentages:** Resizes the window by specified percentages, starting from the current size. Use this command to make a window 10 percent shorter and 30 percent wider, for instance.

## Window scrolling commands

Use the window scrolling commands to scroll around in a document, much as you would by clicking in the scroll bar or by using the arrow keys. For each one, choose how you want to scroll from the Action pop-up menu, the units to scroll by (lines, pages, or a percentage of the document), and the number of lines, pages, or the percentage to scroll by (**Figure 10.29**).

**WARNING!** Not all applications work with iKey's window scrolling commands. You'll have to test with specific applications to know, unfortunately.

**FIGURE 10.29**



Choose how you want to scroll and the units to scroll by, and enter the amount to scroll.

- **Scroll Window Up/Down by Distance:** These two commands scroll the window up and down by the specified amount.
- **Scroll Window Up/Down to Level:** These two commands scroll the window up and down to the specified location.
- **Scroll Window Right/Left by Distance:** These two commands scroll the window right and left by the specified amount.
- **Scroll Window Right/Left to Level:** These two commands scroll the window right and left to the specified location.

## Window switching commands

Use the window switching commands to move among open windows.

**TIP** The order of windows isn't necessarily predictable, so make sure these commands do what you want. It may make more sense to use a [Menu](#) command to choose a particular window from a Window menu.

- **Switch to Next Window:** Switches to the next open window in the current application.
- **Switch to Previous Window:** Switches to the previous open window in the current application.

## Auto-Complete Dialog

The Auto-Complete Dialog command is particularly powerful, because it lets you auto-fill an entire dialog or window without having to link together several tabbing, typing, and clicking commands.

**TIP** Useful though Auto-Complete Dialog is, don't assume it will work in every application. It works only with applications that use entirely standard interface elements and techniques, so it may fail when used in some programs. In those cases, just create a shortcut that mimics all the actions you would take to set up the dialog the way you want it.

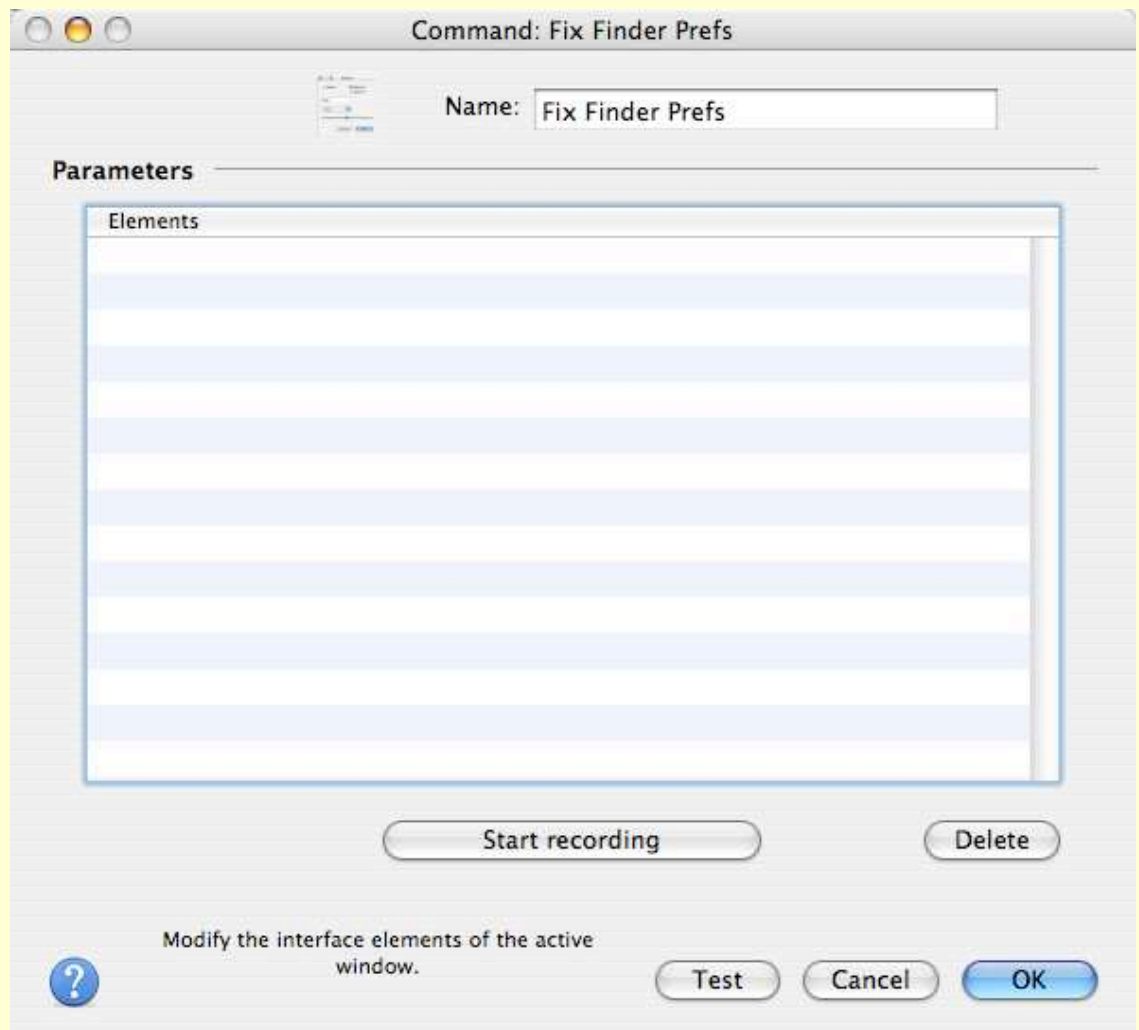
Follow these steps to make an Auto-Complete Dialog command that sets a few Finder preferences automatically:

1. Switch to the desired application (the Finder in this case), open the dialog you want to auto-fill (the Finder Preferences dialog), and set in exactly the way you want iKey to automate later.

**NOTE** The Auto-Complete Dialog command can select and deselect checkboxes and radio buttons, enter text in text fields, move sliders, and more. It cannot choose from pop-up menus, and you should not use it to click buttons that change the window state.

2. Switch back to iKey Editor, create a new Auto-Complete Dialog command, and give it a name (**Figure 10.30**).

**FIGURE  
10.30**



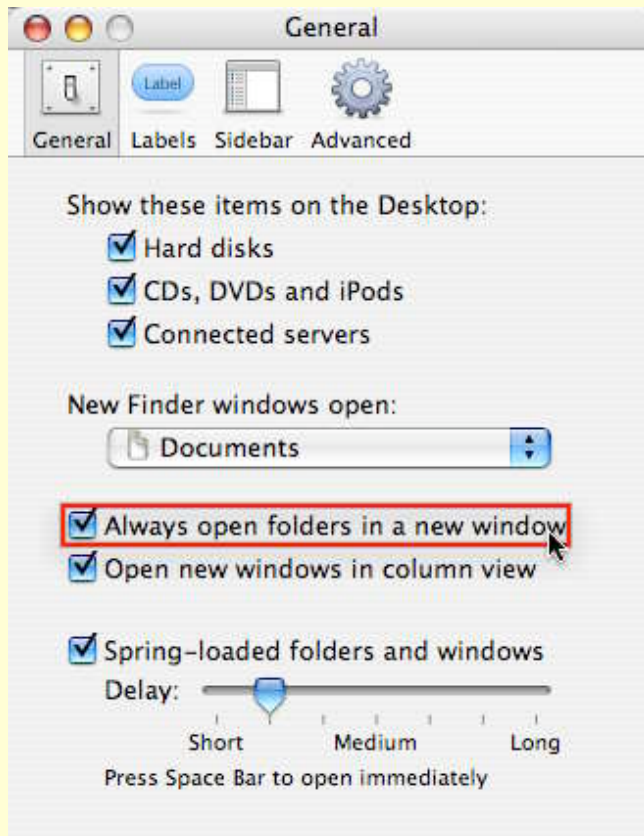
Create a new Auto-Complete Dialog command.

3. Click the Start Recording button.

iKey Editor presents you with an instructional dialog; read it carefully! Click OK when you understand the procedure.

4. Click in the dialog you want to auto-complete, and notice that a red box outlines controls you mouse over (**Figure 10.31**).

**FIGURE 10.31**



The red box marks the control whose state iKey can record.

5. To record the state of a control, press Command-Option when that control has the red box around it. To record the state of all the controls in the dialog, press Command-Shift-Option.
6. When you're done, switch back to iKey Editor and click Stop Recording.
7. If you have added any unnecessary interface elements, remove them by selecting them in the list and clicking the Delete button.

**TIP** Do not click the Test button within this command. Since the dialog in question isn't in front, the command will always fail.

8. Close the Auto-Complete Dialog window, add your newly created command to a shortcut (if you weren't creating it within one to start with), make sure the shortcut has a launcher, and then choose File > Save and Activate to save your changes. Then open the dialog in question and activate your shortcut to see if it works.

## LAUNCHER REFERENCE

The traditional method of launching a shortcut is by pressing a hotkey, but iKey extends the number of ways you can launch a shortcut with several other types of events. Plus, launchers can apply equally as well to menus and palettes; the launcher displays the menu or palette so you can choose a menu item or click a palette button.

**TIP** Invoking a menu's hotkey displays the menu at the current pointer position; it's a great way to present yourself with a selection of commands and shortcuts to choose from.

iKey provides four basic types of launchers: keyboard events (hotkeys), date/time events that fire when the specified time rolls by, application events that are tied to specific applications, and system events that kick off based on a system-wide occurrence.

**TIP** Remember that any given shortcut, menu, or palette can have multiple launchers, so you can launch a shortcut with a hotkey and also have it fire automatically at a specific time. Or you could have a shortcut fire at multiple times throughout the day by assigning it multiple date/time event launchers. Also remember that the same launcher can trigger multiple shortcuts.

## Creating Launchers

You can create launchers in two ways:

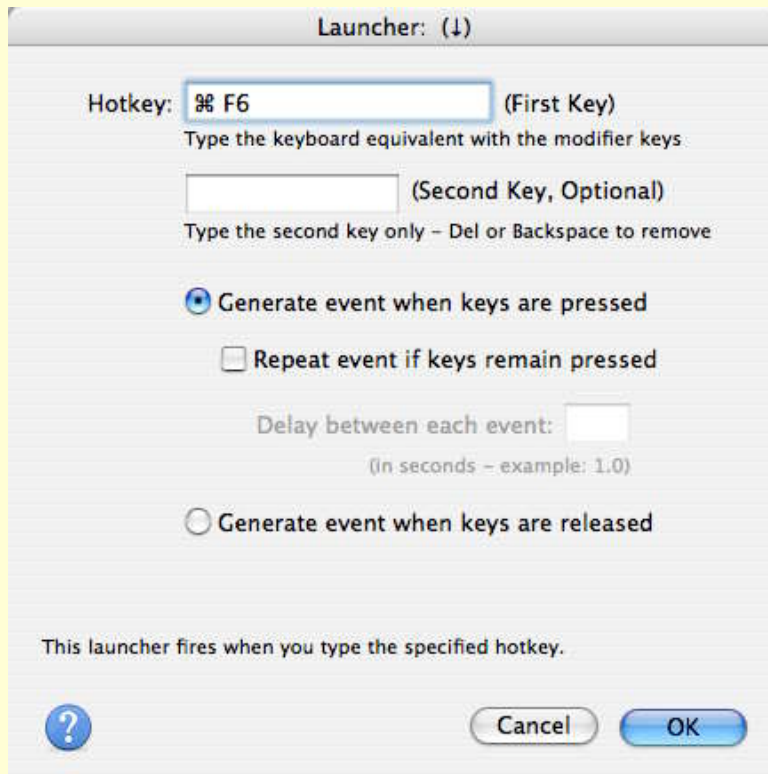
- While creating a shortcut, menu, or palette, click the + button in the Launcher list area and choose New from the appropriate launcher category submenu.
- Choose the desired launcher category from the Create > Launcher menu.

In each case, iKey Editor displays a Launcher window with controls specific to each type of launcher.

## Keyboard Event

You launch the vast majority of shortcuts through keyboard events; in other words, by pressing a combinations of keys. In this respect, iKey works like any other application in which you could press Command-Q to quit or Command-N to create a new file. As you can see in **Figure 11.1**, though, iKey's keyboard events are a bit more flexible than normal. Plus, in iKey 2.2, you can have keyboard shortcuts that use not just a single key with modifiers, but two keys. More on that after we look at creating a basic single-key hotkey.

**FIGURE 11.1**



Single-key keyboard event launcher.

Type the hotkey for the launcher in the Hotkey field, and adjust the other settings to control when and how often the launcher fires.

- **Generate Event When Keys Are Pressed** causes the launcher to fire a single time as soon as the Mac registers that you have pressed the keys. It's the default, and is appropriate for the vast majority of situations.
- **Repeat Event if Keys Remain Pressed** extends that functionality by causing the launcher to fire repeatedly as long as you keep pressing the keys. Enter how long iKey should wait between events in the Delay between Each Event field.



**TIP** You might use the repeat option if you want a shortcut to cycle through options, such as rotating through all the active applications.

- **Generate Event When Keys Are Released** changes the launcher's behavior such that it fires only after you stop pressing the keys. Some people prefer this behavior to that of having the event generated as the keys are pressed.

You might be wondering exactly what combinations of keys are valid. They are:

- Any normal alphanumeric key coupled with one or more modifier keys.
- A function key (the F keys at the top of the keyboard) either by itself or in conjunction with one or more modifier keys. Note that some function keys may not be available for use in launchers because Mac OS X may use them for other functions like changing brightness, opening and closing the tray, or working with Exposé (although you can change the Exposé hotkeys in the Exposé preference pane).
- Any key from the numeric keypad (including the arrow keys and other navigation keys), either by itself or with one or more modifier keys.

**TIP** iKey 2.2 displays a warning message if one of iKey's hotkeys conflicts with a system hotkey, because the system hotkey takes precedence over iKey. You could run into this if, for instance, you used F12 to launch an iKey shortcut, given that Mac OS X 10.4 uses F12 to invoke Dashboard by default. To resolve the conflict, either change iKey's hotkey (perhaps add a modifier) or change the system hotkey in the Keyboard Shortcuts pane of the Keyboard & Mouse preference pane.

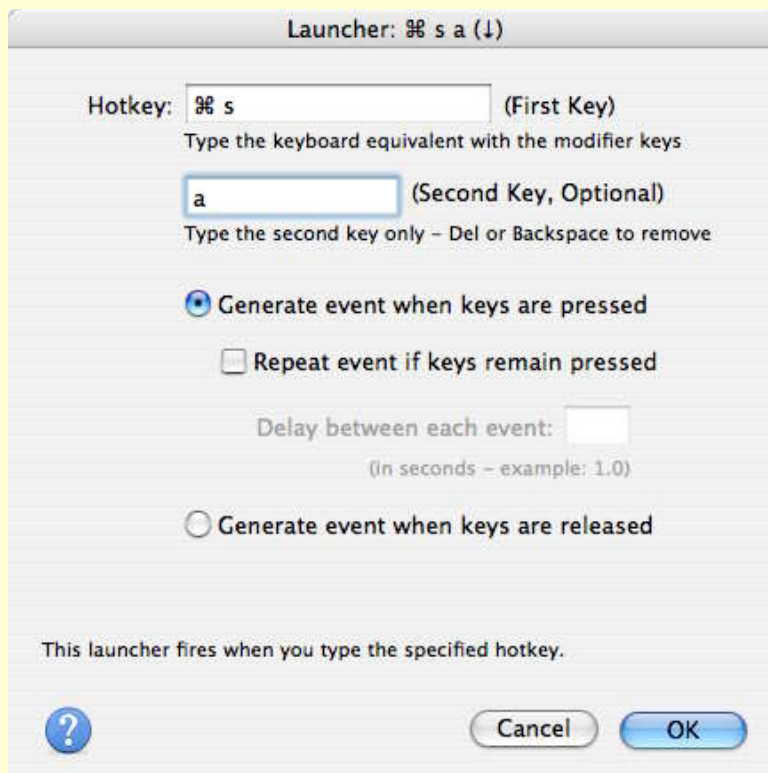
**NOTE** If you're wondering, the modifier keys are Command, Option, Shift, and Control. It is possible to create a launcher that uses a normal alphanumeric key combined with just Shift (Shift-e), but be careful since such a launcher prevents you from typing capitals (E).



That's how you define a single-key hotkey, which is fine in many cases. But if you've ever found yourself trying to remember if you assigned Option-S or Control-S to Save As—or was it Control-Shift-S?—you'll absolutely adore iKey 2.2's new double-key hotkeys. Here's how they work.

In the Hotkey field, type the first key you want to use to invoke your shortcut, along with the necessary modifier key. Then, in the next field down, type the second key without a modifier (**Figure 11.2**). Set any other options you want and click OK to save your launcher.

**FIGURE 11.2**



Double-key keyboard event launcher to be used to invoke a menu command that chooses Save As from the File menu.

Once you've finished creating and saving the shortcut you want to invoke, it's time to test. In this case, because my shortcut chooses the Save As command only in a particular application, I switched to that application, held down Command-S, and, then quickly, without releasing the Command key, I let up on S and pressed A. How quickly? Within 1 second; if I had been slower than that, my keystrokes would have been interpreted as Command-S, followed by Command-A (which would probably have meant Save, then Select All; definitely not the Save As menu item I wanted).

The brilliance of double-key hotkeys is that a great many menu items, Web pages, and other actions are best described by two words—Save

As, Page Setup, Empty Trash, Reset Safari, and so on—and it's much easier to remember hotkeys that use the first letter of both words, modified only by the Command key—Command-S-A, Command-P-S, Command-E-T, Command-R-S, and so on. With double-key hotkeys, you can avoid using awkward combinations of modifier keys to make your hotkeys unique and memorable.

## Date/Time Event

Another common way of launching a shortcut (though likely not a menu or palette) is when the Mac's clock arrives at a particular time. For instance, I use a date/time event launcher with a shortcut that checks a specific set of Web sites every weekday morning at 9:00 AM (see **Figure 11.3**).

**FIGURE 11.3**

Launcher: Each week (Su, Mo, Tu, We, Th, Fr, Sa) at 09:00:00

Time:  (Example: 14:10:00)

☐ Generate event each day

☒ Generate event each week on these days:

Sunday	Monday	Tuesday	Wednesday
Thursday	Friday	Saturday	

☐ Generate event each month on these days:

1	2	3	4	5	6	7	8	9	10	11
12	13	14	15	16	17	18	19	20	21	22
23	24	25	26	27	28	29	30	31		

If the date/time has passed:

This launcher fires at the specified date and time.

Date/time event launcher.

You must set date/time event launchers to fire at a specific time of day (make sure to enter it as the interface suggests), but you have three options for how often the launcher repeats:

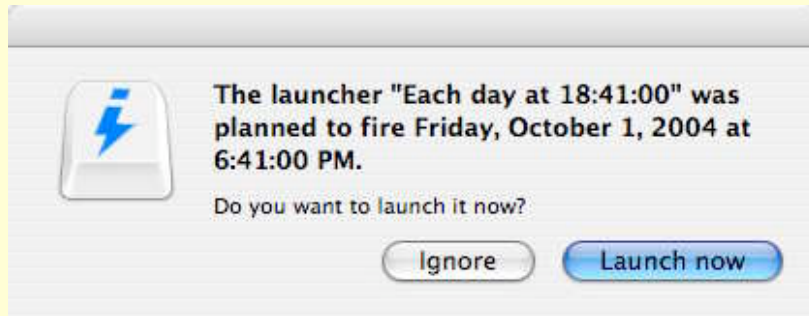
- **Generate Event Each Day** causes the launcher to fire every day at the specified time. If you want to launch a shortcut multiple times per day, create additional date/time event launchers.
- **Generate Event Each Week on These Days** causes the launcher to fire on only the specified days of the week. It's useful for launchers that, for instance, you want to fire only on weekdays; only on weekends; or only on Mondays, Wednesdays, and Fridays.
- **Generate Event Each Month on These Days** causes the launcher to fire on only the specified days of the month. It's useful for launchers that you want to fire on the first of every month.

**NOTE** There's no way to have a date/time event launcher that doesn't repeat. If creating a one-time date/time event launcher is important for some reason, you might be able to create a shortcut that removed the launcher from iKey Editor after the first time it fired.

Your Mac must be turned on and awake and with iKey Engine running for date/time launchers to fire at the specified time; iKey cannot start your Mac or wake it to fire a launcher. However, if iKey notices that it was unable to fire a date/time event launcher, it can take one of three actions, which you choose from the If the Date/Time Has Passed pop-up menu:

- **Ask What to Do** displays a dialog asking if you would like to fire the launcher or ignore it (**Figure 11.4**).
- **Do Nothing** makes iKey ignore the fact that the launcher's time has passed.
- **Fire Launcher** causes the launcher to fire as soon as iKey Engine is active.

**FIGURE 11.4**



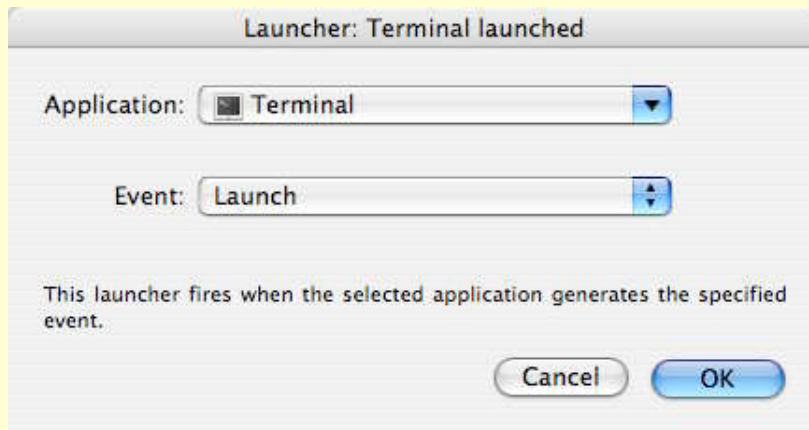
iKey asks if it should run or ignore a date/time event launcher whose time has passed.

Give these three options some thought when creating date/time event launchers. For instance, consider my shortcut for displaying Web sites I like to check each weekday. If I have an early appointment that keeps me from waking my Mac up until after 9:30 AM, I probably still want to have that launcher fire when I wake the Mac up, so I'd set that one to Fire Launcher. However, if the launcher was instead attached to a shortcut that performed a lengthy set of tasks that involved switching back and forth between multiple actions, I'd recommend having iKey either ask what to do or do nothing, since you may not want to watch iKey catch up on everything it missed while your Mac was asleep.

## **Application Event**

iKey can trigger shortcuts (and, nominally, menus and palettes, but it's difficult to think of useful examples) when a specific application launches, quits, activates, or deactivates (see **Figure 11.5**). For instance, imagine that you're forced to use an ugly custom application that doesn't remember window positions or other preferences. You could create an iKey shortcut that would set those preferences appropriately each time the custom application launches.

**FIGURE 11.5**



Application  
event  
launcher.

Choose an application from the Application pop-up menu; note how iKey Editor shows currently launched applications at the top, followed by recent applications. If the application you want isn't listed, choose Other and locate it.

Then choose an application event from the Event pop-up menu. You have four choices:

- **Launch:** Choose Launch to cause your launcher to trigger when the specified application launches. You will probably use launch events more than any other application events.
- **Termination:** Choose Termination if you want your launcher to trigger when the specified application quits.

**TIP** Use termination event launchers carefully because applications quit automatically when the user logs out or the Mac restarts or shuts down. You may not want a shortcut kicking in as you try to restart.

- **Activation:** Use activation events to cause your launcher to trigger when the specified application comes to the front (which usually happens when the user clicks its Dock icon or one of its windows).
- **Deactivation:** In contrast, use deactivation events to cause your launcher to trigger when some other application replaces the specified application in the front. Note that when an application quits, that action counts as a deactivation.

**TIP** Be careful with activation and deactivation event launchers because you'll likely find that you switch between applications more than you expect, and the constant triggering of shortcuts may be annoying.

## System Event

Along with application events, iKey launchers can trigger when specific system events happen, particularly startup, shutdown, going to sleep, and waking from sleep (see **Figure 11.6**).

**FIGURE 11.6**



System event launcher.

All you can choose with a system event launcher is the event itself; you have four possibilities:

- **Startup:** Choose Startup to cause your launcher to trigger a shortcut when your Mac starts up. If you find yourself performing a set of actions every time you restart your Mac (like moving a window for an application that refuses to remember your preferred window location), think about how a shortcut activated by a startup event launcher can save you time. For startup launchers to work, iKey must be set to launch at startup.

- **Shutdown:** Choose Shutdown to cause your launcher to trigger when your Mac restarts or shuts down. Shutdown events are particularly good for triggering shortcuts that make backups of important data, for instance.
- **Go to Sleep:** Choose Go to Sleep if you want your launcher to trigger a shortcut before your Mac goes to sleep.

**TIP** Be sure to test your shortcut to make sure it operates as you expect in all situations where your Mac can go to sleep (choosing Sleep from the Apple menu, pressing the power button, closing your laptop's lid, running out of battery power, and so on).

- **Wake from Sleep:** Choose Wake from Sleep to cause your launcher to trigger a shortcut when you wake your Mac from sleep. Given how little many people actually power down their Macs, waking from sleep is probably much more common than booting the Mac from a powered-down state.

## Context Event

Imagine a shortcut that beeps every time you switch to the Finder. Annoying, I know, but bear with me. You could create an application event launcher that looked for activation of the Finder as its trigger. But what if you wanted your shortcut to run only the first time you switched to the Finder? For that you'd need a context event launcher set to "First Activation," combined with a context that makes the shortcut valid only when the Finder is in front. First Activation means the first time the context occurs after iKey starts up; the other option is Every Activation, which triggers whenever the context occurs.

Honestly, I don't expect many people to use context event launchers right now, but they may become more interesting and useful as iKey evolves and adds additional types of contexts that aren't related to applications. For instance, if iKey had a context related to your network location, and you switched your network location, a context event launcher could automatically mount certain servers that were available only with a particular network location. For now, though,



think of context event launchers as a way of triggering a shortcut on each activation of an application-specific context, or on only the first activation of an application-specific context (**Figure 11.7**).

**FIGURE 11.7**



Context event launcher.

## USB Device Event

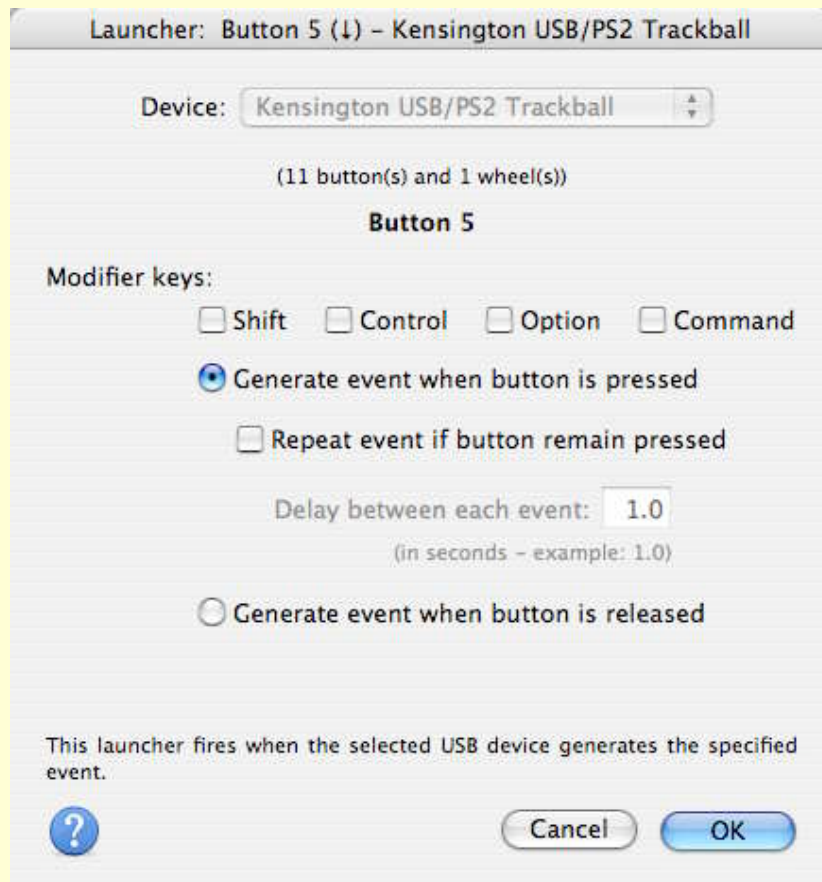
A major new feature in iKey 2.2 is support for USB device events, which enable you to trigger an iKey shortcut when you click a button or move a scroll wheel on a USB pointing device like a multiple-button mouse or trackball, or one of those keyboards with numerous special buttons.

Setting up a USB device event is extremely simple; just create a new USB device event launcher, choose your device from the Device pop-up menu in the Launcher dialog, click the button or move the scroll wheel that you want to use to invoke an iKey shortcut, and decide if you want to require that any modifier keys be pressed at the same time (**Figure 11.8**).

**TIP** For a USB device event to work (or even to define one), you will likely have to disable any special drivers for the device in question, such as Kensington's MouseWorks software or Alessandro Levi Montalcini's USB Overdrive (which is a general purpose USB driver). In some cases, iKey may be able to replace those drivers entirely (if you were using them only for assigning button functions); in other cases, you may need to stick with the driver for features like custom cursor acceleration, and in those cases, you cannot use iKey's USB device events.



**FIGURE 11.8**



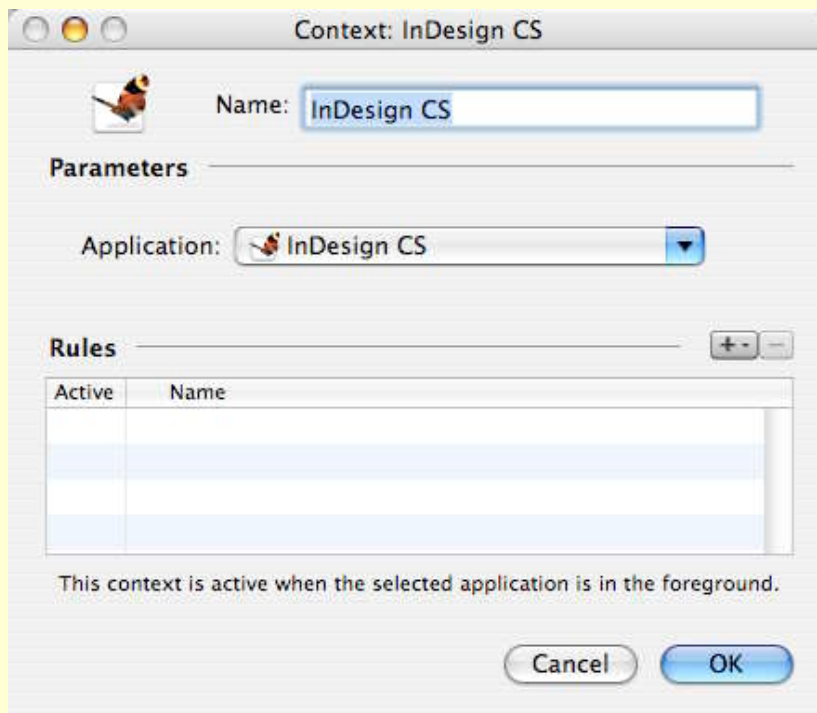
USB device  
event  
launcher.

## CONTEXT REFERENCE

Every shortcut, menu, and palette must have a context in which it is active; in essence, a context is an application state. iKey offers five contexts: Universal, Application In Front, Application in Background, Application Running, and Application Not Running. If you have other ideas for contexts (such as when specific network configurations are active), send feedback to Script Software.

For each of the application contexts, you must specify the application to which the context applies (**Figure 12.1**), and for all five of the contexts you can add rules that deactivate either all iKey hotkeys or specific hotkeys. These rules might be useful in select situations where you need to make sure that an iKey shortcut's hotkey does not override the application's standard hotkey definition.

**FIGURE 12.1**



For the application contexts, you must specify the desired application.

## Creating Contexts

You can create contexts in two ways:

- While creating a shortcut, menu, or palette, click the + button in the Context list area and choose New from the appropriate context category submenu.

- Choose the desired context category from the Create > Context menu.

In each case, iKey Editor displays a Context window with controls specific to each type of context.

## Universal

If you want a shortcut, menu, or palette to be available at all times, which is likely true for most, use the Universal context. Unless you're absolutely certain that you want a shortcut, menu, or palette to be available only within one or more specific applications, I recommend starting with the Universal context. For example, I have shortcuts that launch or switch to specific applications when I press particular function keys. F1 switches to Nisus Writer, F2 switches to my default Web browser, F3 switches to Eudora, and so on. Those shortcuts must use the Universal context because I need them to work no matter what I'm doing.

The main problem you can run into with the Universal context is that a shortcut's hotkey will override a hotkey used by an application. For instance, if you created a shortcut that used Command-Q for its hotkey, you wouldn't be able to use Command-Q for Quit in any application. Oops.

The Universal context has no parameters, and it's highly unlikely that you'd want to deactivate any hotkeys when it's active (since they'd be deactivated almost all the time).

## Application in Front

Even though you'll probably use the Universal context most of the time, the Application in Front context is also incredibly useful, for two reasons:

- The fact that iKey's shortcut hotkeys override application hotkeys means that you can use iKey to remap keyboard shortcut behavior in any application. For instance, imagine that your favorite layout program uses Option-Delete to change kerning and Command-Delete to delete a word at a time. But you prefer Option-Delete as the hotkey for deleting a word at a time, so you can use iKey to create a shortcut that virtually presses Command-Delete whenever you type Option-Delete. (Of course, you lose the kerning hotkey

entirely in this example, but you could remap that as well if you liked.)

- You may want to create menus and palettes containing specific commands and shortcuts that make sense only in a particular application. As an example, consider a palette with buttons corresponding to particular Unix commands. There's little point in having that palette showing except when you're in Terminal, so using an Application in Front context that specifies Terminal is just what the doctor ordered.

## **Application in Background**

The Application in Background context allows shortcuts, menus, and palettes to be active only if the specified application is not the front one. Honestly, I don't see most people needing to use this context very often.

## **Application Running**

Slightly more useful is the Application Running context, which checks to make sure the specified application is active. That could be essential in shortcuts that will fail if a particular target application isn't running.

## **Application Not Running**

Similarly, the Application Not Running context ensures that shortcuts, menus, and palettes that use it are available only if the specified application is not running. I can imagine a use for the Application Not Running context when working with multiple incompatible versions of an application.

## ABOUT THIS EBOOK

In contrast to traditional print books, Take Control ebooks offer clickable links, full-text searching, and free minor updates. We hope you find them both useful and enjoyable to read.

### About the Author

Adam C. Engst is the publisher of *TidBITS*, one of the oldest and most respected Internet-based newsletters, distributed weekly to many thousands of readers. He has written numerous technical books, including the best-selling *Internet Starter Kit* series, and many magazine articles (thanks to Contributing Editor positions at *MacUser*, *MacWEEK*, and now *Macworld*). His innovations include the creation of the first advertising program to support an Internet publication (in 1992), the first flat-rate accounts for graphical Internet access (in 1993, with Northwest Nexus for *Internet Starter Kit for Macintosh*), and the new Take Control electronic book series. In addition, he has collaborated on several Internet educational videos and has appeared on a variety of internationally broadcast television and radio programs.



Adam's indefatigable support of the Macintosh community and commitment to helping individuals has resulted in numerous awards and recognition at the highest levels. In the annual MDJ Power 25 survey of industry insiders, he ranked as the second (2000), third (2001 and 2002), fourth (2003), and fifth (2004) most influential person in the Macintosh industry, and he was named one of *MacDirectory's* top ten visionaries. And how many industry figures can boast of being turned into an action figure?

## Author's Acknowledgments

Thanks to Julian Miller of Script Software for working with us on this project, and thanks to Philippe Hupé (iKey's amazing programmer) for being so open to suggestions.

A tip of the mouse to iKey beta testers Steve Wise and Randal Million for their excellent comments during our collaborative editing phase.

Lastly, thanks to Tonya for her excellent editing work. As always.

## Shameless Plug

If you liked this ebook, you'll undoubtedly like Adam's other works:

### ***TidBITS***

Along with its award-winning Macintosh commentary and editorial, you can also learn about new Mac releases from *TidBITS*.

<http://www.tidbits.com/about/list.html>

### ***The Wireless Networking Starter Kit, Second Edition***

Written with Glenn Fleishman, this definitive book covers everything for individuals and small businesses looking to create, improve, and extend their wireless networks, including details about Wi-Fi networks, Bluetooth, cellular data, wireless security, long-range wireless networking, using Wi-Fi on the road, and much more.

<http://wireless-starter-kit.com/>

### ***iPhoto 5 for Mac OS X: Visual QuickStart Guide***

Adam's other current print book explains how to get started with iPhoto quickly and easily, using illustrated step-by-step instructions rather than lengthy explanations.

<http://iphoto.tidbits.com/>

### ***Take Control of Buying a Mac***

Adam gives step-by-step advice for everything you need to know in order to save money and buy the right Macintosh at the right time.

<http://www.takecontrolbooks.com/buying-mac.html>

## About the Publisher

TidBITS Electronic Publishing has been publishing online since 1990 when publishers Adam and Tonya Engst first created their online newsletter, *TidBITS*, about Macintosh- and Internet-related topics. *TidBITS* has been in continuous, weekly production since then. Adam and Tonya are well-known in the Macintosh world as writers, editors, and speakers. They are also parents to Tristan, who thinks ebooks about trains, clipper ships, and castles would be cool.



At the TidBITS Web site you can subscribe to *TidBITS* for free, join in TidBITS Talk discussions, or search 15 years of news, reviews, and editorial analysis (<http://www.tidbits.com/>).



## Production Credits

- **Cover:** Jeff Carlson, <http://www.necoffee.com/>
- **Take Control logo:** Jeff Tolbert, <http://jefftolbert.com/>
- **Editor in Chief:** Tonya Engst, <http://www.tidbits.com/tonya/>
- **Publisher:** Adam Engst, <http://www.tidbits.com/adam/>



## FEATURED TAKE CONTROL TITLES

Take control of computing with the Take Control series of highly practical, tightly focused electronic books! Written by leading Macintosh authors, edited by TidBITS, and delivered to your electronic doorstep within moments of “going to press,” Take Control ebooks provide exactly the technical help you need. Click any book image below or [visit our Web store](#) to add to your ebook collection!

### **Take Control of Apple Mail in Tiger**

by Joe Kissell



Learn to work smart and reduce email overload with practical advice in the only book available about Apple Mail in Tiger.

**\$10**

### **Take Control of Your iPod: Beyond the Music**

by Steve Sande



Have you ever wondered what your iPod can do beyond playing music? Find out in this engaging compendium.

**\$10**

### **Take Control of Mac OS X Backups**

by Joe Kissell



Create a rock-solid backup strategy so you can restore quickly and completely, no matter what the catastrophe.

**\$10**

### **Take Control of Customizing Tiger**

by Matt Neuburg



Find real-world advice in this roadmap to customizing Tiger, and get started with new features, such as Spotlight and Automator.

**\$5**

### **Take Control of Permissions in Mac OS X**

by Brian Tanaka



Solve quirky problems, increase privacy, and share files better by managing Mac OS X permissions.

**\$10**

### **More Titles!**

Delve into even more topics, including:

- Running your Mac: upgrading the OS, understanding accounts, and sharing files.
- Buying gear: Macs, cameras, and digital TVs.
- More topics: .Mac, AirPort and Wi-Fi security, spam, podcasting, GarageBand, and Microsoft Office.



*Take Control of iKey 2*  
February 2006. Version 1.2  
Copyright © 2006, Adam C. Engst. All rights reserved.

TidBITS Electronic Publishing  
50 Hickory Road, Ithaca, NY 14850 USA  
<http://www.takecontrolbooks.com/>

TAKE CONTROL ebooks help readers regain a measure of control in an oftentimes out-of-control universe. Take Control ebooks also streamline the publication process so that information about quickly changing technical topics can be published while it's still relevant and accurate. Please send any comments to [tc-comments@tidbits.com](mailto:tc-comments@tidbits.com).

Although most Take Control ebooks are commercial products, this manual is, of course, only of interest to iKey 2 users, so feel free to share it with friends who might be considering iKey and want to learn more about the program.

Although the author and TidBITS Electronic Publishing have made a reasonable effort to ensure the accuracy of the information herein, they assume no responsibility for errors or omissions. The information in this ebook is distributed "As Is," without warranty of any kind. Neither TidBITS Electronic Publishing nor the author shall be liable to any person or entity for any special, indirect, incidental, or consequential damages, including without limitation lost revenues or lost profits, that may result (or that are alleged to result) from the use of these materials. In other words, use this information at your own risk.

Many of the designations used to distinguish products and services are claimed as trademarks or service marks. Any trademarks, service marks, product names, or named features that appear in this title are assumed to be the property of their respective owners. All product names and services are used in an editorial fashion only, with no intention of infringement of the trademark. No such use, or the use of any trade name, is meant to convey endorsement or other affiliation with this title.

*Take Control of iKey 2* is an independent publication and has not been authorized, sponsored, or otherwise approved by Apple Computer, Inc.

Apple, Finder, Macintosh, Mac, Mac OS, Panther, and Tiger are trademarks or registered trademarks of Apple Computer, Inc.