

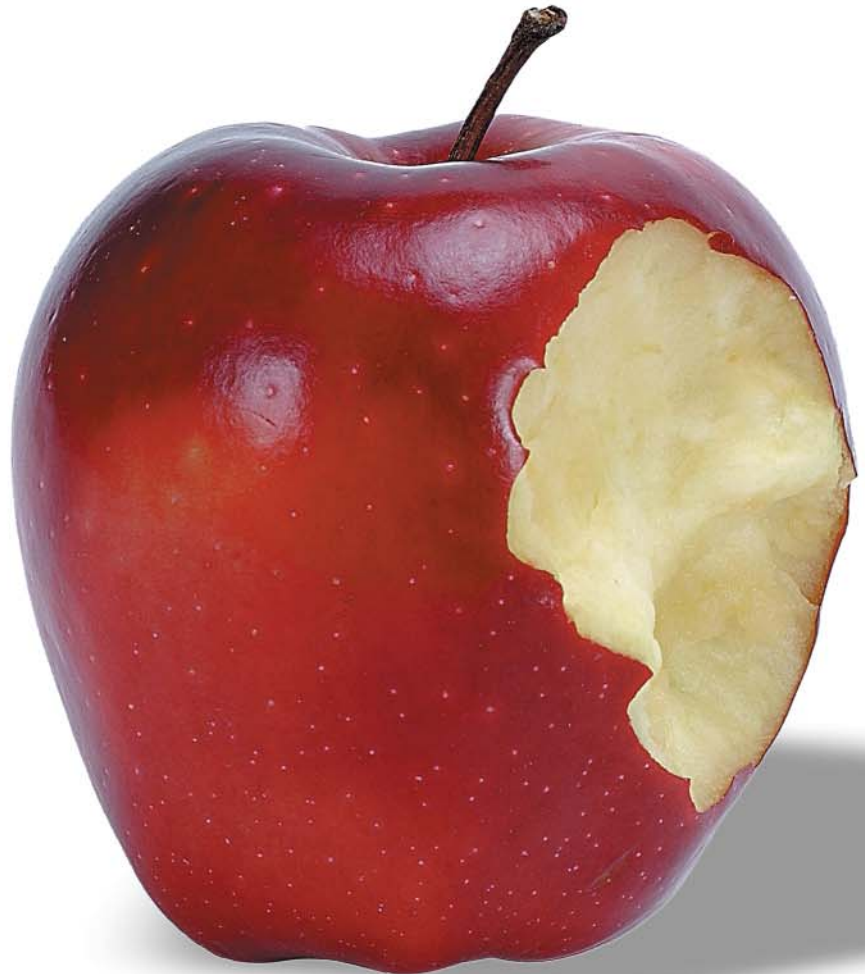
Learn how to:

- *Change Desktop Themes*
- *Give Your Dock a Makeover*
- *Go Application Dumpster Diving*
- *Explore Application and System Resources*
- *Change System and Alert Sounds*
- *Customize Application Icons*
- *Hack Property List Files*
- *Discover Undocumented Preferences*
- *Alter an Application's Interface*
- *Change Keyboard Shortcuts*
- *Script the Unscriptable*

and more...

MODDING MAC OS X

*Extreme Makeovers
for Your Mac*



O'REILLY®

Erica Sadun

Application Dumpster Diving

3

Bill of Materials

Mac OS X Software

- Help Viewer (*/System/Library/CoreServices*)
- Mail (*/Applications*)
- QuickTime Player (*/Applications*)
- iTunes (*/Applications*)
- Preview (*/Applications*)
- iMovie (*/Applications*)
- GarageBand (*/Applications*)
- System Preferences (*/Applications*)
- Terminal (*/Applications/Utilities*)
- Calculator (*/Applications*)
- TextEdit (*/Applications*)
- Xcode (<http://developer.apple.com>)
- Soundtrack (<http://www.apple.com/soundtrack>)

Third-Party Software

- Carbon Copy Cloner (<http://www.bombich.com/software/cccl.html>)
- Dantz Retrospect (<http://www.dantz.com/>)
- SnapzPro (<http://www.ambrosiasw.com>)
- Audio Hijack (<http://www.rogueamoeba.com>)

If you're not afraid to get messy, and you're willing to dive into the complex jumble that is an application's *Resources* folder, you'll find yourself well rewarded. Most Mac OS X applications are brimming with cool stuff, and nearly all of that is packed into the *Resources* folder. It's there that you'll find sounds, pictures, movies, and other fun reusable or, better yet, customizable items.

Whether you're mining for goodies or changing the application to make it "more you," a strong heart and a desire to explore are your best tools. You'll learn what it takes to start playing with resources. You'll customize sounds and pictures and strings and localize those custom files. You'll use the command line to search by extension and learn the best way to view the results. After working through this chapter, you'll know how to find and reuse the very best an application has to hide.

Preparing to Dive

Erica: I wonder whether this stuff that explains the Chinese Take-Out Equivalents should go into the Preface? It almost seems like that's a better place for it; what do you think?

Know who you are. Before you start messing with Mac OS X applications, you need to know how far you're willing (and able) to go. Are you the kind

of person who gets plenty of rest, wears a warm scarf, and doesn't go swimming for an hour after eating? Or are you the kind of person who hops on a mountain bike, doesn't wear a helmet and welcomes bugs in your teeth?

In all likelihood, your Macintosh is probably working properly right now. The applications you've installed are functioning and everything is in its correct (or at least its default) place. Once you start playing with applications, this may all change. No matter your comfort level, you've got to have a plan in mind for when you truly and utterly mess up.

The following approaches range from cautious explorer to reckless abandon. Despite that span, each has a contingency; a plan for getting out of any fix you might find yourself in.

Just one last point to keep in mind: no matter what customizations you make to your applications, they may very well get overwritten the next time you upgrade that application. So, in addition to backing up your originals, you should probably back up your changes as well.

Full Control (Chinese Take Out Equivalent: Mild Vegetables in a Tasty Sodium-Controlled Light Wine Sauce)

Despite the fact that applications are bundled and modular, they do generally create, use, and modify other files on your computer. For the highest level of responsible exploration, make a full backup of your hard drive before attempting to experiment with and/or modify your Mac OS X applications. Use a program such as Carbon Copy Cloner (<http://www.bombich.com/software/cccl.html>) or Dantz Retrospect (<http://www.dantz.com/>) to completely back up your Mac's hard drive. Or, if you have a .Mac membership and external FireWire drive (or a bunch of blank CD-Rs), use the Backup application. Should difficulties arise, you can perform a full restore of your computer to the state before you began to experiment.

High Control (Chinese Take Out Equivalent: Shredded Chicken with Garlic Sauce)

With the exception of such lame ducks as iMovie 4, applications are a lot more robust than you might think. If you feel that a bit of mild customization won't really hurt any ongoing projects or affect critical data, just copy your application. Select it in the Finder and choose File → Duplicate (⌘-D). The Finder copies the bundle at every level from the highest subfolder to the lowest. Then go ahead and experiment.

Pedants will note that duplicating an application can change the copy's ownership from `root:admin` to `username:admin`. Keep this in mind while working through the examples in this book and, if desired, change the ownership with `chown` (i.e., `sudo chown root:admin iPhoto.app`). You'll be prompted for an administrator password.

Figure 3-1 demonstrates how Mac OS X allows you to copy and use multiple versions of otherwise identical applications. With this feature, you can experiment on a copy of your application without affecting the original. Here, a copy of iPhoto displays an icon quite different from the original. If you find you've made a mistake or you want to start again from scratch, you can either trash an altered copy or restore the altered application with the unmodified copy.

Medium Control (Chinese Take Out Equivalent: Kung Pao Beef)

Unfortunately, some applications are big. iDVD alone occupies over a gigabyte of disk space. If you're short on space, or if you think that copying entire applications is overkill, you can limit your scope. Make copies of just those files you're going to alter. Say, for example, that you want to customize a program's About box; just copy an original file (e.g., *AboutImage.tif*) and save that copy with a slightly altered name (e.g., *AboutImage.orig.tif*). Then proceed with your changes. You'll need to work on a file-by-file basis to restore the application. It may help to make a log (by hand or in TextEdit) to keep track of your changes. As a rule, don't move original files out of their folders unless you're confident that you can later put them back where they belong.

Master Geek (Chinese Take Out Equivalent: Ma Po Tofu, extra spicy)

If you're a geek who has achieved that perfect state of caffeine transcendence, then consider customizing without a net. Jump in and feel the burn. People in this final category always have the most fun and generally need the most antacid. I can promise you that Apple does not have a safety squad who will rush out (a la the Fab 5) and slap your fingers for messing with the applications you've installed on your own computer. Do you want to defy Apple's Human Interface Guidelines? Go ahead. Do you want to personalize, customize, and do the wacky? Do it. Mess up? Well, you can always reinstall the application, or the system itself. Or pony up to buy one of those fancy G5's that you've been wanting.

The Xcode Tools

Apple's Xcode Tools include the environments, documentation, applications, and utilities programmers need to design, program, and package Mac



Figure 3-1. Two coexisting copies of iPhoto using different icons.

OS X applications. They also contain the same environments, documentation, applications, and utilities that end-users can use to customize already-built applications. The standard Xcode Tools collection is bundled with Mac OS X on a separate disc, which you must install. You can also download the latest distribution directly from the Apple Developer Connection (ADC) web site (<http://developer.apple.com>).

You must sign up for a free membership with ADC before you can download the developer distribution. The distribution can take hours—or days—to download and you may be better off ordering the \$20 CD.

When installed, the developer distribution appears in the top-level Mac OS X hierarchy in the *Developer* folder. Key subfolders include:

Applications

The primary developer applications. Highlights include Interface Builder (for interface customization) and Xcode (for project development).

Applications/Utilities

The special-purpose tools that augment the main developer applications. Here's where you'll find Icon Composer (icon creation tool), icns Browser (icon inspector), and Property List Editor (tool to create and edit property lists, or plists).

Documentation

All the information you could possibly ever need to know about development for Mac OS X. Important items include *UserExperience* (the Human Interface Guidelines) and *DeveloperTools* (how to use all the items in the developer distribution).

Tools

Helpful command-line utilities for managing and updating Mac OS X applications. Standouts include *SetFile*, which lets you set application metadata including type and creator; *GetFileInfo*, which retrieves application metadata; and *CpMac*, which copies files and includes their resource forks.

Take some time to explore the *Developer* folder and discover the goodies found within. Many items will prove useful when you want to customize application bundles. Others are useful only to programmers. There are even a few items that may not help you work with applications but are just fun.

For example, the */Developer/Applications/Utilities/Built Examples* folder offers a few fun apps:

- Sketch is a simple but useful MacDraw-style program.
- BlastApp brings the classic NeXT helicopter game back to life.

- World Text lets you create and edit Unicode strings.
- Appearance Sample shows off Interface Builder's suite of interactive controls.
- SimpleText does exactly what you remember, except this version has been built for Mac OS X.

Help and Xcode Tools

It takes an extra step if you want to use the Help Viewer to search for documentation from the Developer distribution. The reason for this is because Mac Help and Developer Help are indexed separately. Use these steps to search through Developer Help.

1. In the Finder, choose Help → Mac Help (⌘-?). Mac OS X Help launches and opens a new window.
2. Locate the magnifying glass in the Mac Help search field. It's at the top-right of the help viewer window, as shown in [Figure 3-2](#). Choose Search Developer Help (or Search All Help) from the pull-down menu.

As a rule, use the index pull-down to select a help topic whenever you want to change the scope of your search. When the developer distribution has been installed correctly, and help is launched from the Finder, the choices will include (at a minimum) Search Mac Help, Search All Help, and Search Developer Help. (The first option listed generally reflects the program that launched help.) Choose either of the latter two to access topics from the developer documentation.



Figure 3-2. The Help window.

TOOLS YOU NEED: HELP & XCODE TOOLS

For working through the examples in this section, you'll need the following application:

- Help Viewer (*/System/Library/CoreServices*)

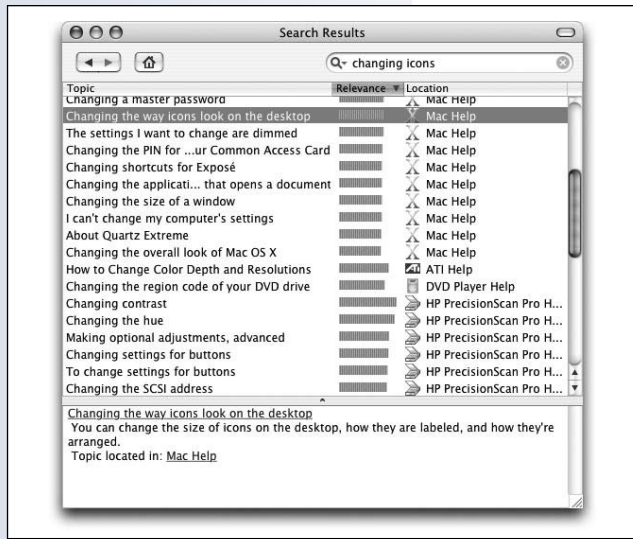


Figure 3-3. The Help Viewer lists all the topics it can find that relate to the phrase you enter in the search field.

TOOLS YOU NEED: FINDING GOODIES

For working through the examples in this section, you'll need the following applications:

- Mail (Applications)

And, optionally:

- QuickTime Player (Applications)
- iTunes (Applications)
- Preview (w)

3. Type a phrase into the top-right search field and press Return.
4. After searching, the Help Viewer returns a list of all the topics it can find that relate to the search phrase you entered. Select any of the listed topics. A topic summary appears in the text area at the bottom of the window, as shown in Figure 3-3. Double-click the hyperlinks in this text area to read more. Links lead to comprehensive overviews of the selected topic.

Finding Goodies

Any number of worthy and curious files can be found within an application bundle's innards. A healthy nosiness will help you discover some of them. Snooping through resource folders is a skill that can be honed and developed. Start with this basic approach.

1. Navigate to the *Applications* folder. Pick any application that intrigues you; in this case, use the Mail application.
2. Control-click (right-click) the application you have selected and choose Show Package Contents from the pop-up, as shown in Figure 3-4.
3. To quickly view what's inside the *Contents* folder, hit the C key and then use the keyboard shortcut ⌘-3 to switch the Finder to Column View.
4. Select the *Resources* folder, located within the *Contents* folder.



Figure 3-4. Show Package Contents exposes application contents.

- When surfing through an application bundle, it's best to organize your folders by file type. This groups together similar kinds of items, and allows you to see all the audio files or all the icon files at once. Choose View → Show View Options (⌘-J). Select Keep Arranged By → Kind, as shown in Figure 3-5.
- As you look through the files in the *Resources* folder, decide which ones you might want to use for other purposes and which ones you might want to alter when customizing the application. You may want to develop a list of the relevant files, their location and a brief description of their contents. Make sure to browse through the localized *lproj* files as well as the main *Resources* folder.

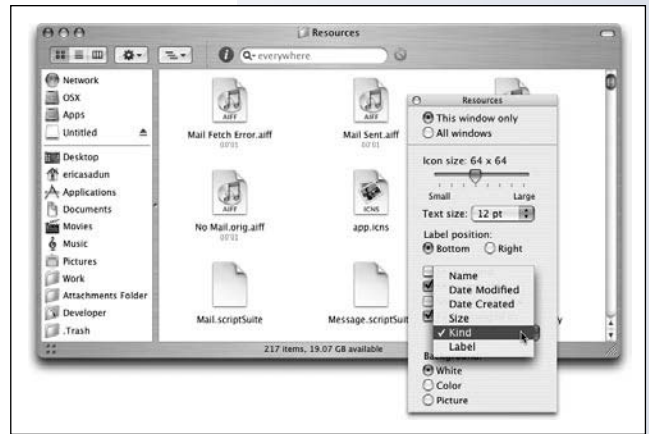


Figure 3-5. Arrange your resources by kind for easy browsing.

Image and sound files are among the easiest to assess. Use Preview to view images, and the QuickTime Player or the Finder itself to listen to the sounds. If you've set up iTunes to automatically import sounds (as I have), using QuickTime rather than iTunes allows you to preview audio without adding a hundred little sounds to your iTunes library. (See Chapter 4, *Changing Icons* to read more about setting application associations.)

Changing Sounds

Because of its simplicity, the Mail application provides a terrific guinea pig for learning how to swap AIFF files. Follow these steps to change the “No Mail” sound to an AIFF file that you select.

- Quit Mail if it is running. In fact, you should always quit any application before you attempt to modify its resources.
- Open the Mail application package and navigate to *Contents/Resources*. “No Mail.aiff” appears in the *Resources* folder. Select the “No Mail.aiff” file in the Finder, and then click on the play icon to listen to the original, metallic sound.
- Next, create a backup copy of the “No Mail.aiff” file. There's no convention on how you should rename it; just give it a name so that the original file can no longer be found by the application. In this case, I've renamed the file as “No Mail.orig.aiff”.
- Move a new, short (1 or 2 seconds, max) AIFF file into the *Resources* folder and rename it as “No Mail.aiff”. Mail is hardwired to look for a file named “No Mail.aiff” and to play that sound if no mail is found on the server.
- Launch Mail by either clicking on its icon in the Dock, or by double-clicking on its icon in the *Applications* folder.

TOOLS YOU NEED: CHANGING SOUNDS

For working through the examples in this section, you'll need the following applications:

- Mail (*/Applications*)

And, optionally

- iMovie (*/Applications*)
- SnapzPro (<http://www.ambrosiasw.com>)
- Audio Hijack (<http://www.rogueamoeba.com>)
- GarageBand (*/Applications*)
- Soundtrack (<http://www.apple.com/soundtrack>)



Figure 3-6. Mails General Settings preferences pane.

6. Choose Mail → Preferences (⌘-,) from the menu bar, and then click General to open the General settings pane. Ensure that Play Sounds For Other Mail Actions is checked (as shown in Figure 3-6), and then close the Preferences window.
7. Check your mail. If mail has arrived, wait for it to finish transferring and immediately check your mail again. Keep checking until no new mail is retrieved. When there's no new mail, Mail will play the new sound file.

Symbolic links and aliases provide a way to use sound files from one application in another. Use the Terminal to rename an original sound (e.g., *No Mail.aiff.bak*). Then use the link command to create a new version of the file (e.g., `ln -s /Applications/foo.app/Contents/Resources/English.lproj/blort.aiff No Mail.aiff`)

Internationalizing sounds

Move beyond simple sound customization by internationalizing your AIFFs. It's surprisingly easy to accomplish. Just move language-specific versions from the main *Resources* folder into localized *lproj* folders. Here's how:

1. Create several AIFF files that state there's no new mail. If you don't actually speak more than one language, use funny accents instead. Make sure the different versions are distinct and can be easily identified when played back.

Keep in mind that there's no official Apple support for Ubbi Dubbi or Latin (not even the Pig and Classic varieties).

2. Make sure that the Mail application is not running; if it is, quit Mail (Mail → Quit Mail, or ⌘-Q).
3. Open the *Resources* folder in Mail's application bundle.
4. Copy the file named *No Mail.aiff* to *No Mail.orig.aiff*. This creates a backup before you add the primary language version.

TOOLS YOU NEED: INTERNATIONALIZING SOUNDS

For working through the examples in this section, you'll need the following applications:

- Mail (*/Applications*)
- System Preferences (*/Applications*)

Recording Sounds

Although you can use existing sound clips to customize your applications, sometimes it's more fun to record your own. GriffinTechnology produces the iMic (<http://www.griffintechnology.com/products/imic/>), a USB interface that allows you to plug in a microphone or connect any other audio source.

When recording with the iMic and a microphone, make sure to push the small black switch toward the speaker icon. This enables the internal amplifier, bringing your microphone input into audible levels. To record from a CD player or other pre-amplified source, set the switch toward the microphone, to use line-level mode.

Once you've connected your input device to the Macintosh, use the Sound panel in System Preferences to select the sound input device and test the levels. Make any adjustments so the levels stay within the middle of the input range.

There are a number of Classic and OS X applications that allow you to record sounds. In addition to all the normal Sound Recorder-like applications, you might consider a few oddball approaches. For example:

- iMovie allows you to record audio. It saves that recorded audio as AIFF files in the project Media folder.
- Snapz Pro (<http://www.ambrosiasw.com>) is a screen capture utility but it also lets you record from the microphone and save the results to a QuickTime movie.
- Audio Hijack (<http://www.rogueamoeba.com>) lets you hijack your microphone with its free Sound Source application.
- GarageBand and Soundtrack (<http://www.apple.com>) both let you create custom AIFFs by editing loops and outputting the result to a file.

Sound Advice

Although it's occasionally fun to have your computer speak to you, fun has its limits. After you finish experimenting, you may want to restore the original sounds that come with the Mail application.

Despite the overwhelming success of AOL's "You've Got Mail," there are good reasons that Apple engineers use subtle audio cues rather than boldly spoken phrases. Many people use their Macintoshes in home or office environments where quiet is a commodity.

I remember back to the old days of the NeXT computer, when the company briefly (but memorably) experimented with voice cues for their applications. One afternoon, the pre-recorded

voice of my NeXT announced that the "Printah has run out of Paypah" for the hundred-oddth time. My husband, who had been resting in the adjoining room, ran in, shouting that someday, he was going to kill that \$@*&ing British woman.

As far as applications go, understated sounds almost always produce the best results. They provide just enough feedback, so you know that an operation has occurred, but not so much that they draw attention to themselves. Apple continually spends time, effort, and money engineering the sounds that ship with their applications. Use the audio found throughout their apps as a model for your own sound engineering needs.

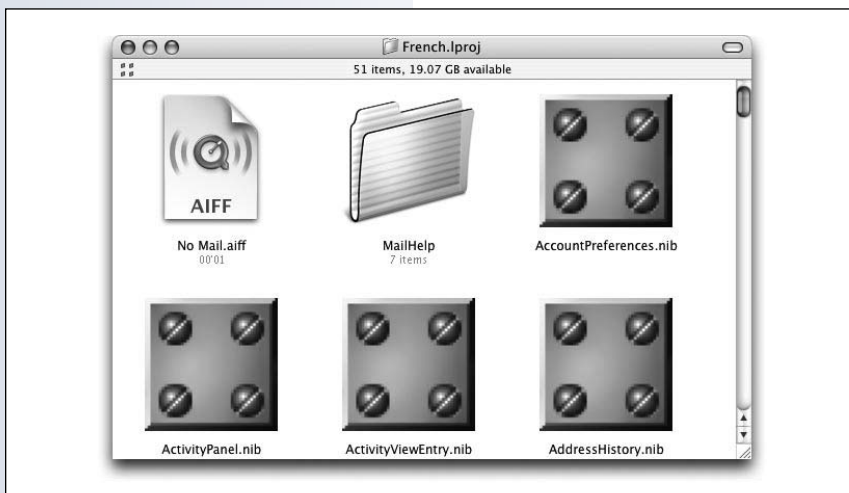


Figure 3-7. Mails French.lproj folder.

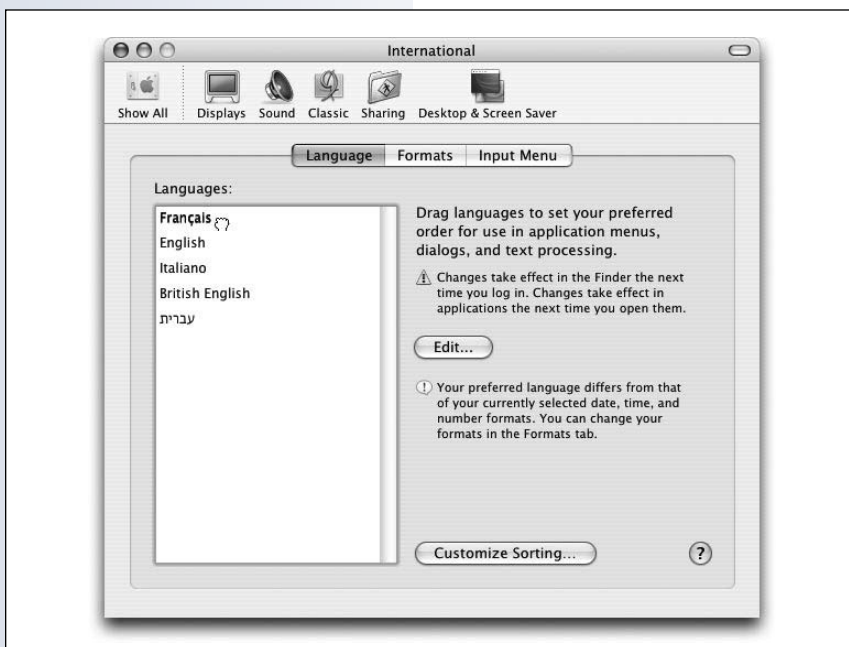


Figure 3-8. Choosing a primary language in the International pane.

5. Replace the file named *No Mail.aiff* with the version you recorded in your primary language.
6. Move the new version of *No Mail.aiff* into the *lproj* folder for your primary language, such as *English.lproj*. This leaves the main *Resources* folder without a *No Mail.aiff* sound file.
7. As described in the last section, launch Mail and ensure that the proper sound plays back when you check for new email. When you're satisfied that everything still works, despite the file relocation, quit Mail (⌘-Q) and return to the *Resources* folder.
8. Select one of your alternate language recordings. Rename the file to *No Mail.aiff* and place it in the appropriate *lproj* folder, such as *French.lproj*. This creates two versions of *No Mail.aiff* each in an *lproj* folder, leaving no copies in the main *Resources* folder. **reference Figure 3-7**
9. Launch System Preferences, either by clicking on its Dock icon or from the Apple menu, and then click on the International panel.
10. Use the instructions from Chapter 2, *Choosing Localizations*, to add and then select the language used by your alternate audio file. Make sure to drag this new preferred language to the top of the list. **reference Figure 3-8**
11. After changing your language settings, restart Mail. As before, check for new mail. If you've changed the language to French,

click on the Relever button; other icon titles vary by language. Assuming that no mail is found, your new localized version of the *No Mail.aiff* sound clip plays back.

12. Once you're convinced that the sound localizations work as promised, quit Mail (⌘-Q). Return to System Preferences, open the International preferences panel and restore your primary language.

Searching Through Alternate Locations

As you just saw, one of the easiest customization tasks you can make is to replace a sound file with one that has your own twist. For many applications, it proves extremely simple to adapt or replace alert sounds. Mail is just one of these many easy-to-change apps, since Mail stores its alert sounds as Audio Interchange File Format (AIFF) files in the top *Resources* folder. Replace the sound file and you've replaced the sound. It's as simple as that.

Unfortunately, not all applications are so standard. Unlike Mail, many of iChat's sounds are stored outside the application bundle. They can sometimes be found in an external framework. Frameworks, which are discussed further in Chapter 2, provide a modular way to program and deliver Mac OS X applications.

To locate iChat's audio files, navigate to the */System/Library/PrivateFrameworks/InstantMessage.framework* folder. Select the latest version (alphabetically) and open the *Resources* folder; for example, *Versions/A/Resources* (also known as *Versions/Current/Resources*). This directory contains iChat's "missing" AIFF files.

Despite complexity of location, both iChat and Mail use the same strategy for loading sounds. When you replace an AIFF file, you replace the sound used by that application. This works even when the sounds involved do not appear within the application bundle.

For some applications, however, replacing sounds is quite difficult. iTunes (at least up to version 4.5) was built using a very Classic approach. Its resources occur in an RSRC file rather than in individual AIFFs. You'll need to use ResEdit, or some other resource editor, to listen to or alter its one sound, a chime sequence called *Encoding Complete*. CodeWarrior offers particularly robust resource editing tools. Unfortunately, Mac OS 9 (or earlier) development skills are generally necessary to update these kinds of sounds.

Search strategies

As iChat demonstrates, useful and customizable material can be stored in more places than just the application bundle; you just have to be willing

**TOOLS YOU NEED:
SEARCHING FOLDERS**

For working through the examples in this section, you'll need the following application:

- Terminal (*/Applications/Utilities*)

to poke around on your system a bit. One particular Unix command also helps, making it very easy—and quick—to search.

Focus on these three principle directories:

- */Applications*
- */System/Library/Frameworks*
- */System/Library/PrivateFrameworks*

These directories tend to contain all kinds of interesting and amusing files.

Searching application folders

To search through the *Applications* folder for AIFF audio files, you need to search both the main *Resources* folder and at least one localization folder. Here are the commands to search the main folder:

```
$ cd /Applications
$ ls *.app/Contents/Resources/*.aiff
```

When you issue the second command (*ls*), the Terminal returns a list of matching file names. These files appear in various applications' *Resources* folders and are listed with their paths.

Next, search through *English.lproj* folders. This helps you discover any internationalized files that may not appear in the main folder. *English.lproj* appears in most application distributions, for most countries. (Japan is rumored to have a problem in this area.) Where necessary, substitute your primary language.

```
$ ls *.app/Contents/Resources/English.lproj/*.aiff
```

To search for other kinds of files, other than AIFF sounds, use the above methods with different extensions. The commands you just saw make it particularly easy to track down application resource files (*.rsrc*), images (usually *.tif* or *.tiff*, but also try searching for *.gif*, *.jpg*, etc.), movies (*.mov*), and so forth.

Searching with the locate command

The *locate* command can find files that match a particular pattern. This makes it easy to search for all aiff files (**.aiff*), or movies (**.mov*) on your computer. To use this utility, you'll need to update the locate database. This database is created and maintained as part of the weekly maintenance routines (*/etc/weekly*).

To use *locate*, issue a *locate *.aiff* command at the Terminal's command line. You may want to pipe the results through *more* to page through them or through *open -f* to open the results in TextEdit. For example:

```
$ locate *.aiff | open -f
$
```

You can update the locate database by running the updater directly (`sudo /usr/libexec/locate.updatedb`) or by running the weekly maintenance (`sudo /etc/weekly`).

Searching with inodes

Help subfolders also contain intriguing multimedia material, including movies, which are often found in the *lproj* folders because they're almost always localized. Unfortunately, there's no consistent naming pattern, other than the presence of the word *help*. Some applications use spaces (e.g., *DVD Player Help* and *QuickTime Broadcaster Help*), while others do not (e.g., *MailHelp* and *PreviewHelp*).

Although help folders use these arbitrary names, they are all highly structured, in a very non-arbitrary manner. For example, the *pgs* and *pgs2* folders contain source HTML. The *gfx* (short for “graphics”) and *shrdgfx* (short for “shared graphics”) folders hold the multimedia, but don't be surprised if you also find JPEGs, GIFs, TIFFs, PNGs, and the occasional (but rewarding) QuickTime movie file in one of these directories. To search for help movies issue the following command at the Terminal's command line:

```
$ ls -i *.app/Contents/Resources/English.lproj/*Help*/*gfx | grep mov
```

What returns is a list of help movies, each preceded by its file serial number, also called the file *inode*. The `-i` option instructs `ls` to include the file's inode number in the listing. This list will vary depending on the types and versions of the applications installed in the */Applications* folder. Most of the following files appear in archaic versions of iMovie and iPhoto, which I keep around for reference reasons.

According to Apple's developer documentation, inodes are “the focus of all file activity in the file system.” Inodes, which describe each file in the Mac OS X file system, actually consist of two parts: the serial number that's used to speed up file lookup and its file's metadata. In general, inodes are of most interest to programmers, who use them to directly access file data. You can read more about inodes by issuing a *man 5 inodes* command at the command line.

```
440286 iMA.mov
1308976 hlp132.mov
1308991 hlp22.mov
1309286 hlp23.mov
1309287 hlp24.mov
1309288 hlp31.mov
1309289 hlp33.mov
1309291 hlp80.mov
```

To find a file by inode, use the `find` command. This command may take a minute or two to run, depending on the extent of the folder and subfolders it must search.

```
$ find /Applications -inum 440286
```

find returns a list of all matching files, offering a complete path to the file; for example:

```
/Applications/iMovie 2.app/Contents/Resources/English.lproj/iMovie
Help/gfx/iMA.mov
```

Searching frameworks

Frameworks require a slightly different approach for searching than applications. Keep in mind that frameworks (both private and public) are subject to version changes. So you must always search through “current” items. That being said, frameworks use a folder structure that mirrors application bundles, including localized *lproj* subfolders. To search through the private (mostly application specific) frameworks for AIFF files, do the following:

```
$ cd /System/Library/PrivateFrameworks
$ ls *.framework/Versions/Current/Resources/*.aiff
```

Use a similar approach to search through the localizations:

```
$ ls *.framework/Versions/Current/Resources/English.lproj/*.aiff
```

Keep in mind that the results of these searches will depend on both the applications and the versions you’ve installed. For more results, you may want to search for **.tiff* rather than **.aiff*.

To search the System frameworks, simply substitute Frameworks for PrivateFrameworks.

Searching for common file types

Some resources are commonly used across different applications. Pictures are just one example of a frequently occurring file type. When a search returns dozens or even hundreds of hits, you can reissue the search and direct the output of the search command to save the results in a file on your hard disk.

The following example searches through the *Applications* folder, looking for all TIFF image files that use either a *.tif* or *.tiff* extension. Each item returned is saved into the *alltiffs.txt* file, which is created on the Desktop. Notice that the first *ls* command creates the file (using *>*) and the second appends to that file (using *>>*).

```
$ cd /Applications
$ ls *.app/Contents/Resources/*.tif* > ~/Desktop/alltiffs.txt
$ ls *.app/Contents/Resources/English.lproj/*.tif* >> ~/Desktop/alltiffs.txt
```

Use the word count command, *wc*, to count the hits. With the *-l* option, *wc* returns the number of lines listed in a file. I found over 2000 TIFFs in my *Applications* folder.

```
$ wc -l ~/Desktop/alltiffs.txt
```

Now open the *alltiffs.txt* file in TextEdit for convenient viewing:

```
$ open -e ~/Desktop/alltiffs.txt
```

The `-e` option instructs the `open` command to open the specified file in TextEdit.

Finding and Changing Images

As you're about to find out, it's just as easy to customize application images, as it is to alter their sounds. Follow these directions to add a personal touch to iPhoto's About Box.

1. Make sure that iPhoto isn't running before you start experimenting with its contents. If it is, select iPhoto → Quit iPhoto from the menu bar, or use the standard `⌘-Q` keyboard shortcut to quit iPhoto.
2. Open a Finder window and go to the *Applications* folder (Shift-`⌘-A`), select iPhoto, Control-click (right-click) on the application icon and select Show Package Contents from the context menu.
3. Navigate to the *Contents/Resources* directory. To speed up the process, hit the C key, then `⌘-3` to switch the Finder window to Column View mode.
4. Look inside the *Resources* folder for a file named *AboutImage.tif*.
5. Create a duplicate copy of *AboutImage.tif* by selecting the file in the Finder and then selecting File → Duplicate (`⌘-D`) from the menu bar, which creates a copy of the file in the same directory and names it *AboutImage copy.tif*. *
6. Now open the original image (*AboutImage.tif*) in your favorite image editor. I personally like the price point and features of Adobe Photoshop Elements. Lemke Software's GraphicConverter (<http://www.lemkesoft.com>) is another nice choice.

Since most applications will not open files from within an application bundle, either drag the original TIFF image file to an application icon in the Dock, or make an alias (File → Make Alias, `⌘-L`) and then drag the alias to the Desktop.

7. Edit the image, and let your imagination run wild. Make sure to leave room for the text that will be added to the large white space at the bottom right of the image. ** ReferenceFigure 3-9 *
8. Save your edits and make sure you output the file using the TIFF format.

TOOLS YOU NEED: FINDING IMAGES

For working through the examples in this section, you'll need the following applications:

- iPhoto (*/Applications*)
- An image editor, such as Adobe Photoshop Elements (<http://www.adobe.com>)



Figure 3-9. Editing iPhoto 4s About box.



Figure 3-10. Many applications, such as iPhoto, store window art in external files that you can edit.

9. Open the image in Preview to make sure that it's been saved correctly and displays as you expect. If needed, return to your image editor and make any changes. If it does not look right in Preview, it won't look right in iPhoto's About Box.
10. Launch iPhoto by either clicking on its icon in the Dock, or double-clicking its icon in the Finder.
11. Choose iPhoto → About iPhoto. Assuming you've edited the correct file in the correct language folder, you'll be treated to the new about box. [Figure 3-10](#) offers just one example of what you might do with that file.

If you like, you can create versions of the about box for different *lproj* folders. Make sure to move *AboutImage.tif* out of the *Resources* folder. Use the International settings pane trick (described in the AIFF example found earlier in this chapter) to view localized versions.

Peeking at (and Playing with) String Files

Most Mac OS X applications contain internationalized string files, typically named *Localization.strings*. This file appears in each language-specific *lproj* folder, and allows developers to internationalize their applications by providing language-specific alternatives for string replacement.

String files generally use UTF-16 (Unicode) encoding, which makes them difficult to view or edit in Terminal. (Some use UTF-8, but that's less common.) UTF-16 uses extra bits for each alphabetic character, allowing a much wider range of characters to be used. Unicode can handle character sets ranging from French to Norwegian, from Japanese to Finnish, from Korean to Dutch; visit <http://www.unicode.org> to learn more. The Unicode site lets you view and download character code charts and search through the character database.

Sadly, Mac OS X's installed versions of Emacs and Perl do not support UTF-16, which makes automated substitution a bit of a pain.

Figure 3-11 shows what a Terminal window might look like as you attempt to list the contents of a *Localization.strings* file. Fortunately, you can easily view and edit string files in TextEdit, which fully supports UTF-16 character encoding.

In the example shown here, *more* treats each 8-bit byte as a separate character, leading to the proliferation of the `^@` items between each readable character.

Looking at internationalized strings in TextEdit

TextEdit reads, displays, edits, and saves string translation files. Since these files are embedded in bundles, it's generally easier to access them from the command line. In the Terminal, navigate to a directory that contains one or more string files and use the `open -e` command; the file opens in a new window in TextEdit.

```
$ cd /Applications/iChat.app/Contents/Resources/English.lproj
$ open -e Localizable.strings
```

Each string file is composed of a series of comment lines and replacement lines. Comments are bounded by `/*` and `*/`, and work just as they would in C source code; items between the two symbols are ignored. Replacements consist of two strings (characters appearing between matched quotation marks) separated by an equal sign, and ending with a semicolon. The original developer language (typically English) appears first; the translation on the right appears after the equal sign.

Example 3-1 shows a small portion of iChat's Italian localization. Notice the use of the accented `è`, an item that does not normally appear in American English character sets. (You'll learn how to add your own accented Unicode characters to strings files later in this chapter.)

Example 3-1. A snippet of iChat's Italian localization file.

```
/* Alert title: enable service sheet arg1:service name */
"%@ is disabled" = "%@ è disattivato";

/* Text/Audio/Video choose window text message: person is offline */
"%@ is offline." = "%@ non è in linea.";

/* Format for a person or card name plus email (arg1:name, arg2:e-mail
address) */
"%@, with e-mail address %@", = "%@, con indirizzo e-mail %@";
```

Items that start with a percent symbol (`%`), such as `%@`, are formatting placeholders. At run time, the program replaces them with text to produce complete strings, such as “Joe is offline,” or “Joe non è in linea.” Other formatting items include `%d` (integers) and `%f` (floating point numbers). As with comments, these placeholders should be familiar to those acquainted with the C programming language.

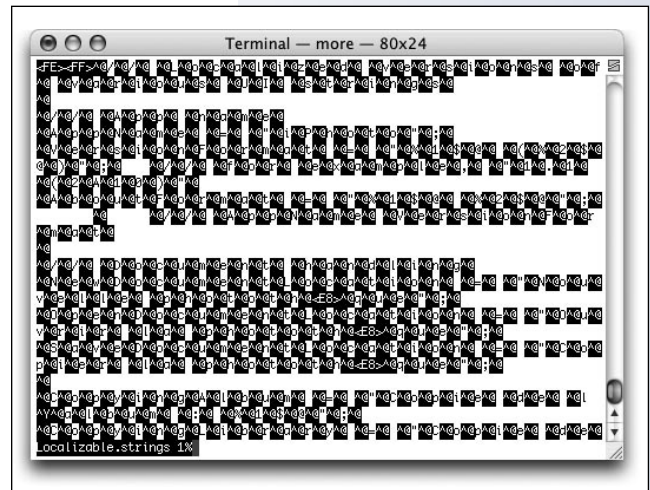


Figure 3-11. Localization.strings files use Unicode UTF-16 encoding, making them hard to view or edit using Terminal.

TOOLS YOU NEED: LOOKING AT STRINGS

For working through the examples in this section, you'll need the following applications:

- Terminal (*/Applications/Utilities*)
- TextEdit (*/Applications*)
- Calculator (*/Applications*)
- Xcode (<http://developer.apple.com>)

Using TextEdit may prove a bit frustrating for those readers who are comfortable using *Emacs*, *vi*, or even *pico* or *ed*, because it does not offer regular expression substitution.

If you're not used to working with C and want to learn a little more, one book that's a must-have is *The C Programming Language*, by Brian W. Kernighan and Dennis M. Ritchie (Prentice Hall). In programming circles, this book is a classic. There is also a companion workbook, sold separately.

You can edit the files by changing the right-hand set of strings. The right-hand strings define the localized version that automatically translates from the original. Make sure to retain any % placeholders. For example, you might add an apologetic note to a translation:

```
/* Text/Audio/Video choose window text message: person is offline */
"%@ is offline." = "%@ non è in linea. Mi dispiace molto.";
```

WARNING

Never alter the left-hand strings or the translations will not work.

Example: editing Calculator's strings

The Calculator application is one of Mac OS X's simplest built-in applications and has, perhaps, the fewest number of strings to customize. In the following steps, you'll create a new localization for Calculator and edit the internationalized strings. Rather than add meaningful localizations, you'll simply tag the strings so you can instantly identify how each string method is applied.

1. Make sure that the Calculator application isn't running; if it is, quit the application with **⌘-Q**.
 2. Open a Finder window and go to the *Applications* folder (Shift-**⌘-A**). Select Calculator and choose File → Duplicate (**⌘-D**); the Finder creates a new copy, which you can use as a backup.
 3. Select Calculator. Control-click (right-click) and choose Show Package Contents. Navigate to the *Contents/Resources* folder.
 4. Select the *English.lproj* folder and choose File → Duplicate (**⌘-D**) to make a duplicate copy of the folder. Rename the new copy *en_CA.lproj*; this name refers to Canadian English.
-

Alternate English dialects are written as *en* followed by an underscore (**_**) and a two-letter uppercase variant. Examples include British English (*en_GB*), Canadian (*en_CA*), and Australian (*en_AU*). This naming convention extends to all languages and variations. The first two-letter core language uses the ISO 639-1 standard. The two-letter uppercase variants are defined by ISO-3166. You can find a complete list of both standards at the International Organization for Standardization (<http://www.iso.ch>).

- Find the *Localizable.strings* file, located in the new *en_CA.lproj* folder. Select the *Localizable.strings* file and drag it onto TextEdit's icon in the Dock to open the file.
- In TextEdit, open the Find/Replace panel (Edit → Find → Find, **⌘-F**). Find each occurrence of = " (an equal sign, followed by Space, followed by a double-quotation mark). Replace this with = "XX (an equal sign, followed by Space, followed by a double-quotation mark, followed by XX). Click on the Replace All button.

The Find/Replace function adds the letters **XX** to the start of each translated string. This provides easily identified visual markers for you to spot while running the program. ****Reference Figure 3-12****

- Save your work by selecting File → Save (**⌘-S**).
- Next, change your Mac's language settings to Canadian English. Launch System Preferences (**⏏** → System Preferences), and click on the International preference panel. If Canadian English is not already listed, click Edit. Locate Canadian English and check it; click OK. Canadian English now appears in the list of available Languages. Drag Canadian English to the top of the list; this sets it as your new preferred language. ****Reference Figure 3-13****
- Now launch Calculator; make sure that you run the version you just edited and not the copy.

Any instances of unexpected extra "XX"s generally appear at the ends of some menu items, mainly due to the ellipsis (the three dots).

After you launch Calculator, look through the menus and examine the presented windows. The changes you made to the *Localizable.strings* file will appear on some, but not all. This is because many of the menu items are not affected by Calculator's *Localizable.strings*; they're actually defined in the application's main NIB file. In the following steps, you'll continue the localization

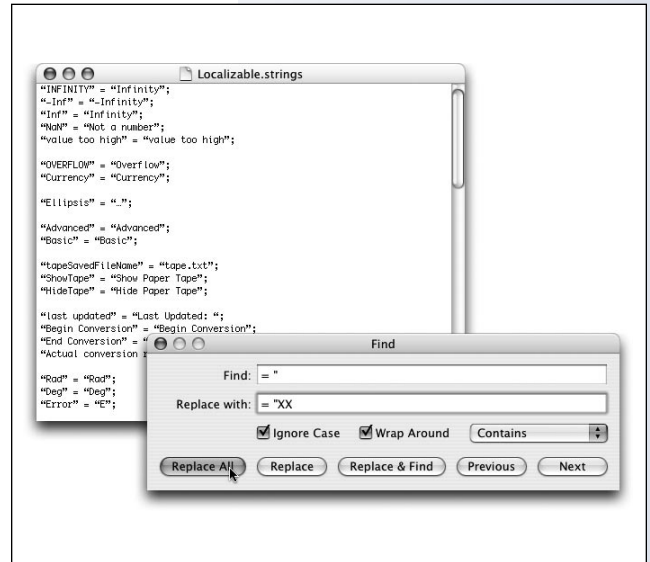


Figure 3-12. Adding markers to the replacement string with the Find/Replace panel

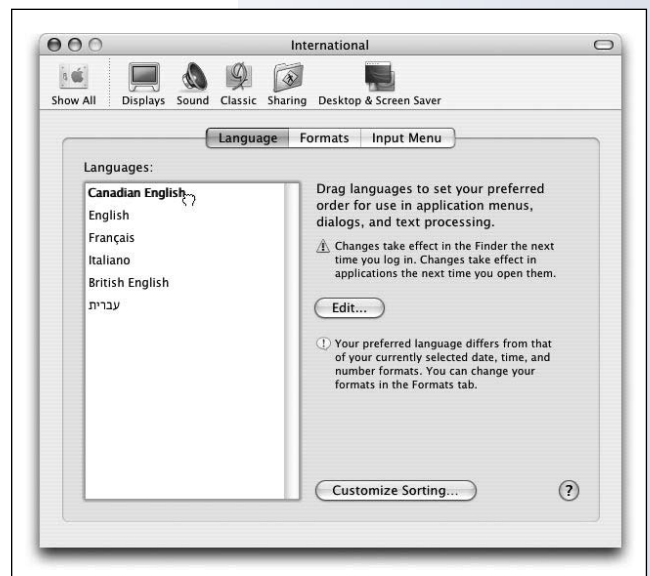


Figure 3-13. Setting the preferred language with the International preferences pane.

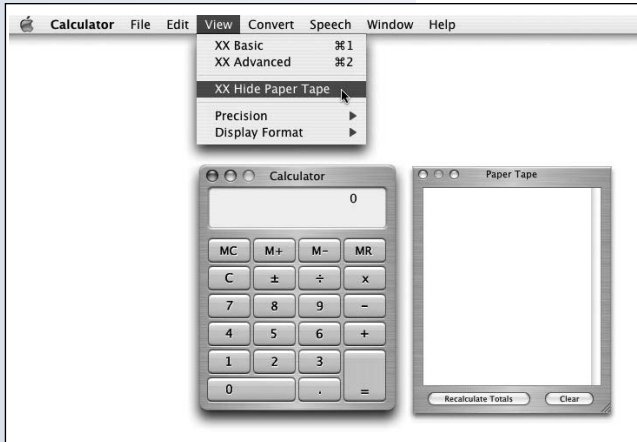


Figure 3-14. The Calculator changes its menu items when you update the localization file.

process and update the NIB strings, to complement the *Localizable.strings*. **Reference Figure3-14**

1. Quit Calculator with **⌘-Q**.
2. Launch the Terminal application (*/Applications/Utilities*); the following steps work best when run from the command line.
3. Return to the Canadian English *lproj* folder. You'll find *en_CA.lproj* in the Resources folder of the altered Calculator application.
4. Locate the *Calculator.nib* file, which is in the *en_CA.lproj* folder. It defines the windows, menus, and strings used by the Calculator application. Quite a few strings are bundled in the *Calculator.nib* file (which is actually a folder). You'll need to replace this file to update your strings by using the following command:

```
$ mv Calculator.nib Calculator.orig.nib
```

5. The *nibtool* utility (installed with the Xcode Tools), allows you to retrieve all the strings from a NIB file. Use the *-L* flag and redirect your output to a new strings file. Be careful, and watch your upper- and lowercase characters.

```
$ nibtool -L Calculator.orig.nib > nib.strings
```

6. Open the new strings file in TextEdit:

```
$ open -e nib.strings
```

7. In TextEdit, open the Find/Replace panel (Edit → Find → Find, or **⌘-F**). Find each occurrence of = " (an equal sign, followed by a Space, followed by a double-quotation mark). Replace this with = "ZZ (an equal sign, followed by a Space, followed by a double-quotation mark, followed by ZZ). Click the Replace All button. This adds the letters ZZ to the start of each translated string, providing visual contrast between the strings changed in the localization file (XX's) and the strings changed in the NIB file (ZZ's).

8. Save your work by selecting File → Save (**⌘-S**) from the menu bar.

9. Use *nibtool* to create a new *Calculator.nib*, which combines the edited strings with rest of the original NIB file:

```
$ nibtool -w Calculator.nib -d nib.strings Calculator.orig.nib
```

After the conversion finishes, check to make sure that a new *Calculator.nib* file was created.

10. Now launch the Calculator, making sure that you run the altered copy. As [Figure 3-15](#) shows, each string tag (XX or ZZ) instantly identifies which method was used to localize each string. In this example, XX tags indicate strings updated in the *Localizable.strings* file. ZZ tags correspond to strings appearing in the NIB file.

Obviously, these steps won't create a real localization; only careful edits and thoughtful translations can do that. However, by following this example, you've seen how easily you can produce and edit the strings needed for a fully internationalized application.

Before you finish, make sure to return your machine to its original settings:

Restore your original language settings

Open System Preferences and select the International preference panel. Click Language, and then drag your preferred language to the top of the list. Close System Preferences.

Quit Calculator

Exit your modified version of Calculator.

Clean up

Delete the modified version of *Calculator.app*, and then change the name of the copied original from *Calculator copy* to just *Calculator* (you don't have to worry about adding the *.app* extension).

Using Unicode strings in TextEdit

Although TextEdit allows you to add Unicode characters to your string files and other documents, it's not immediately obvious how to accomplish this task. In the following steps, you'll discover several tools you can use.

1. Launch System Preferences, either by clicking on its icon in the Dock or from **Apple** → System Preferences.
2. Click on the International preference panel.
3. Click the Input Menu tab. The Input Menu consists of a list of items, each preceded with a checkbox. [Figure 3-16](#) shows the default presentation for a US-based Macintosh.

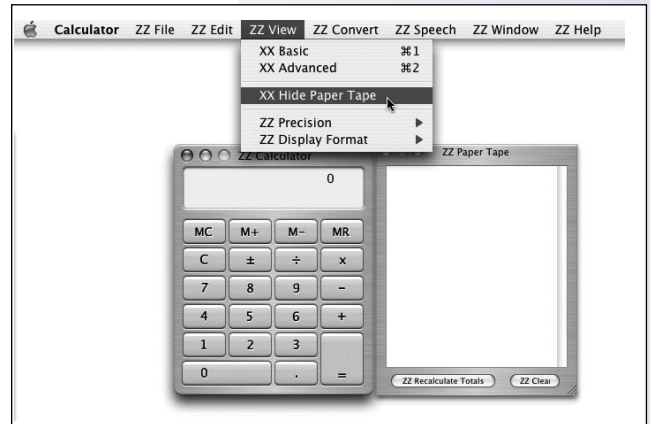


Figure 3-15. The updated Calculator application with its tagged strings.



Figure 3-16. The Input Menu pane of International settings allows you to select keyboard layouts and palettes that help you build rich, multi-language Unicode files.

If you click the checkbox next to the option “Show input menu in menu bar,” the flag of the default keyboard layout appears in the menu bar. For example, if your keyboard layout is for the U.S., you’ll see the Stars and Stripes appear as the icon for the Input Menu in the menu bar.

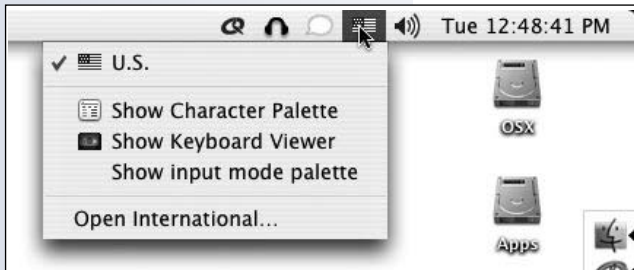


Figure 3-17. A flag icon in the menu bar indicates the current setting for internationalized keyboard input, here using US English. This flagged menu allows you to select a language and, if you’ve enabled the options, to access the Character Palette and Keyboard Viewer.

The Latin-1 Supplement list offers a particularly useful set of accented characters.

TOOLS YOU NEED: USING UNICODE STRINGS

For working through the examples in this section, you’ll need the following applications:

- System Preferences (*/Applications*)
- TextEdit (*/Applications*)

4. Check Character Palette and Keyboard Viewer. These two palettes provide direct access to extended character sets.
5. Select the checkbox next to “Show input menu in menu bar.” This option adds the Character Palette and the Keyboard Viewer to a menu in the menu bar for easy access, as shown in [Figure 3-17](#).

6. Launch TextEdit and open a new document (File → New), suitable for experimenting in.
7. In the menu bar, click on the Input Menu and select the Show Character Palette option.
8. Select Unicode from the View pop-up. Click By Category. Select any of the items in the scrolling list on the left.
9. In TextEdit, begin to type normally. When you need to add a Unicode character, locate and double-click on the character in the Character palette. New characters appear in the front-most application in the focused window at the insertion point, as shown in [Figure 3-18](#). Use the View pop-up to select from a variety of core language families. You can use the Character Palette to add ornamental punctuation, accented characters, currency symbols, and more.

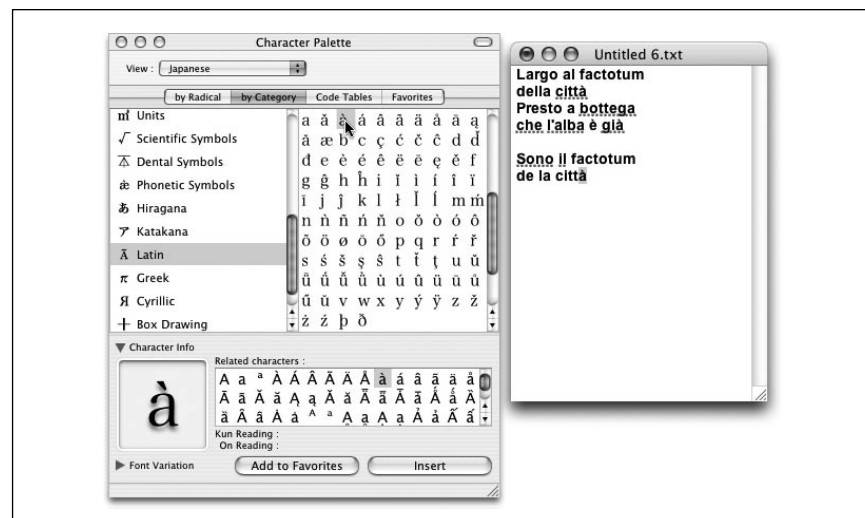


Figure 3-18. Use the Character palette to add Unicode to your text. Useful character groups in this palette include punctuation, Greek letters, digits, scientific symbols, and more.

When you encounter symbols that you use over and over again, such as an accented è or the note symbol (ÀÚ), consider adding them to your favorites for quick access. Just click Add to Favorites and the characters will appear in the Favorites tab.

10. Once you're satisfied with your understanding of the layout and operation of the Character Palette, click on the Input Menu and select Hide Character Palette (or click the small red close button at the top-left of the palette).
11. To add extra languages to the Input menu, click on the Input Menu and select Open International. Add an extra language or three, so you have several to select from. French, German, and Italian all offer items that do not appear in the standard English character set.
12. Open the Keyboard Viewer (Input Menu → Show Keyboard Viewer). This tool helps people who are more comfortable using a keyboard layout to enter text than using the Character Palette's table layout. (Superficially, the viewer is similar to KeyCaps.) It allows you to preview fonts and to type strings.
13. Return to TextEdit and bring your document window back into focus. Use the floating keyboard to enter text. Each click on a keyboard button produces a text character.
14. Click on the Input Menu and select one of the new languages. The flag for the Input menu changes to reflect the currently selected language. Return to TextEdit and continue typing with the Keyboard Viewer; the keyboard layout updates to reflect the new language, as shown in Figure 3-19.

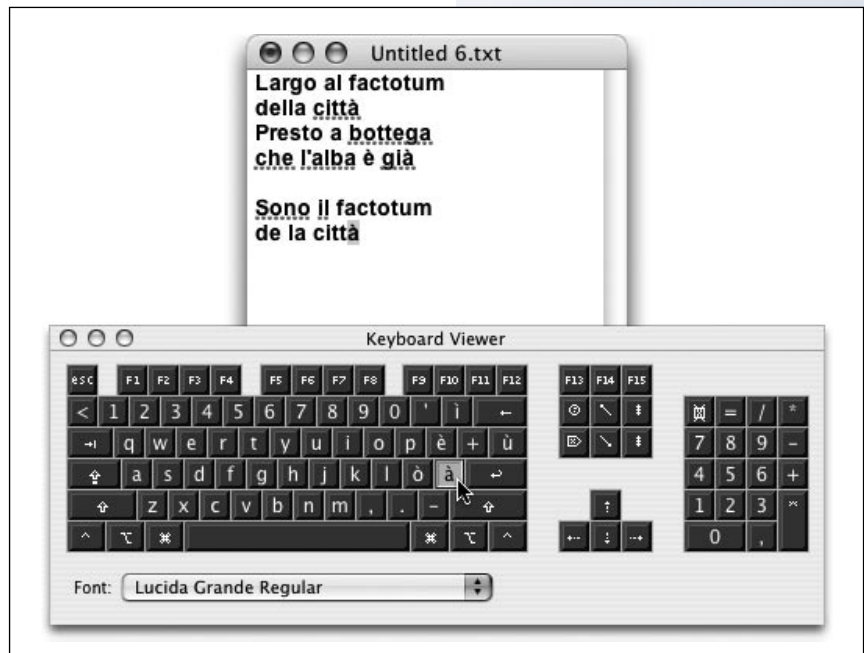


Figure 3-19. The Keyboard Viewer localizes to match the settings used by the Flag menu and the International settings pane. Changing languages allows you to access extended character sets using a friendly keyboard layout. Click any key to type it.

To learn more about Apple's Internationalization tool suite, visit <http://developer.apple.com/intl/localization/tools.html>.

Final Thoughts

This chapter introduced application diving—how to delve into application bundles, search around, and even change the resources found there. You saw how to find, customize, and use many different file types. You learned about internationalizing programs and creating Unicode strings. Before you move on to icons and extensions, here are a few key points to ponder about the material covered in this chapter.

Most items appear in the same places (or nearly the same places) in application after application.

The more familiar you are with the standard folder and subfolder hierarchy, the easier it becomes to weed out repetitive items and focus your interest on unique and intriguing assets.

Think international.

Understanding internationalization is as important a part of application diving as searching for cool pictures. Not everything wacky or wonderful appears just in English.lproj directories.

Take charge of your localization.

If you live in Canada, England, or Spain and have bought software intended for France, the United States, or South America, create a localization that fits the way you work. Don't settle for "recognize" when you prefer "recognise." Colour your world and customize your apps.

Restraint is overrated, particularly when you're just starting to explore.

If you're using the methods in this chapter to learn more about OS X applications, then don't hold yourself back by the bounds of good taste. Work on a copy and knock yourself out—at least, creatively that is. You can always return to the original.

Don't underestimate the usefulness of strong Unix command-line skills.

The command line opens up an entire world of power and possibility. If you're a bit out of practice or are new to the command-line world, you might want to pick up a copy of Learning Unix for Mac OS X Panther (O'Reilly). Even a small amount of knowledge goes a very long way in the Unix world.

Make your applications more you.

If a smile in the About box cheers up your day, why not go ahead and add one? Your software belongs to you.