Available as a
PDF Electronic Book
or **Print On Demand**

Creating Mac Widgets with Dashcode

89 PAGES

William H. Murray and Chris H. Pappas

Apress®
THE EXPERT'S VOICE™

www.allitebooks.com

About firstPress

Apress's **firstPress** series is your source for understanding cutting-edge technology. Short, highly focused, and written by experts, Apress's firstPress books save you time and effort. They contain the information you could get based on intensive research yourself or if you were to attend a conference every other week—if only you had the time. They cover the concepts and techniques that will keep *you* ahead of the technology curve. Apress's firstPress books are real books, in your choice of electronic or print-on-demand format, with no rough edges even when the technology itself is still rough. You can't afford to be without them.

Creating Mac Widgets with Dashcode

Dear Reader,

We have taught a lot of programming courses and written a lot of programming books in our careers, but this time we felt that Mac users would benefit the most by knowing how to develop Dashboard widgets with Dashcode rather than laboring over programming languages and endless pages of code.

Our first goal in this short book is to introduce you to the major significant components of the Dashcode product and quickly teach you how to develop some interesting and exciting Dashboard widgets. The first half of the book concentrates on Dashcode templates and shows you how to use the stock templates, with just a little modification, to create your own unique widgets.

The latter half of the book uses slightly more aggressive alterations to the templates and introduces you to some straightforward JavaScript code. With just a little effort, you'll be creating some really exciting widgets.

The second goal is to get you right into using the Dashcode product, help you have fun developing widgets, and show you how easy it is to create those essential utilities that you have wanted to create for so long. Have fun!

Chris H. Pappas and William H. Murray,
Professors of Computer Science,
State University of New York Affiliation,
Binghamton, New York

www.allitebooks.com

Contents

Chapter 1: Widgets in a Hurry.....1

A Little Widget History..... 2

Ease.....2
Focus.....3
Functionality.....4
Cost.....5
Implementation.....7
Distribution.....12

The Look and Feel of a Good Apple Widget.....18

Front Face Design Ideas.....18
Back Face Design Ideas.....19
Miscellaneous Design Ideas.....20

What Widgets Can and Cannot Do.....20

The Problems with Widgets.....20

Clutter.....20
Widget Removal.....21

More Excitement Coming.....21

www.allitebooks.com

Chapter 2: Widgets Without Worry: Using Dashcode 23
What Is Dashcode? 23
Where Is Dashcode and How Is It Installed? 23
A Quick Look at Dashcode Features 28
Front Face Changes 36
Back Face Changes 41
Important Dashcode Features and Tools 44
View 44
Library 46
Inspector 47
One Template Down, Nine To Go 52
Chapter 3: Widget Templates: Thinking Out of the Box 53
The Countdown Timer Template 54
The Maps Template 59
The RSS and the Daily Feed Templates 66
The Podcast Template 71
The Photocast Template 75
The Photocast Template and .Mac Web Gallery 78
The Quartz Composer Template 79
The Video Podcast Template 83
The Gauge Template 86
Out of the Box—Not! 89

www.allitebooks.com

Creating Mac Widgets with Dashcode

by William H. Murray and Chris H. Pappas

Welcome to the world of easy *widget programming. Dashcode is Apple's revolutionary product for automatically generating widget code. With Dashcode, you can work with a variety of widget templates and experiment with subtle changes in widget forms and properties. When you are ready to move on to more challenging projects, those same templates can serve as the base for exciting and unique widgets that you can add JavaScript code to in order to make even more functional.*

The key ingredient in Dashcode is ease of use. You'll be developing widgets in no time at all. We'll teach you how to start with small projects, and as you expand your knowledge, show you how to add interesting features to each widget.

Widget programming fills the gap in your application needs. Once, you would either have had to purchase every application you needed or become a programmer and designed your own applications using massive compilers like those provided with Xcode. With Dashcode, much of what you need for your applications is generated automatically. Then, with a little JavaScript, you can create even more robust applications.

This book will provide many of the development fundamentals you will need to be an active widget designer and creator. But, this book is only the beginning of what you can do with Dashcode! It's all in the word: Dashcode. Dash *for fast and* code *for what is being written. Dashcode, fast code.*

www.allitebooks.com

www.allitebooks.com

Chapter 1: Widgets in a Hurry

In this chapter, you will learn about what widgets are, their short history, what they can and cannot do, their various types, problems that you might encounter with them, and how they are installed on your computer.

This brief overview will put you on the right track for easily developing your own widgets with Dashcode. Of course, if all of this is familiar to you, you might want to jump right to Chapter 2, which goes into the process for developing your first widget with Apple's Dashcode.

What is a widget? This is almost a rhetorical question, since everyone familiar with OS X has probably bumped into a widget or two while investigating various Apple resources on Apple's web site (`www.apple.com`). However, there is a formal definition that should be considered.

In an Apple developer's bulletin, Apple states, "Widgets are handy mini-applications built with web technologies and some specialized resources. They exist on Dashboard, a layer that appears over the desktop with a keystroke or mouse click. Widgets allow users to assemble a collection of information and tools that are most useful to them."

In other words, widgets are just small applications that use a special layer in OS X, called Dashboard, that allows easy user access. The easiest way to open the Dashboard layer is by selecting the Dashboard icon, which appears on the OS X dock. See Figure 1-1 for the actual icon.

Figure 1-1. The Dashboard icon

The simplest widgets are built from several pieces of code that might include graphics, HTML, and JavaScript components. Of course, the rub is getting those

www.allitebooks.com

various files connected and talking to each other. That is also the reason you will prefer using Dashcode—everything will be done automatically for you.

So, just to make sure we're clear about some terminology before proceeding, Dashboard is the screen or layer where widgets are installed and modified, and appear. You open Dashboard by selecting the Dashboard icon from the OS X dock. Dashcode, on the other hand, is where we will be doing our widget design and modifications. There will be more on the ins and outs of Dashcode starting with Chapter 2.

A Little Widget History

Widgets have had a relatively short life, starting with the latter versions of OS X 10.x. However, during this time, widgets have become extremely popular. Why? The "why" can be summed up in several key concepts: ease, focus, functionality, cost, implementation, and distribution. If you have been using widgets for a while, maybe you can even think of a few additional categories.

Ease

There was a time in personal computing when most applications were written by users. Trust us when we tell you that we were there when the Apple IIs were rolling off of the production line. When early users were writing their programs, they were either using assembly language or Apple's version of interpreted BASIC. The applications were simple, short, and modestly fast. When an interpreted BASIC program is executed, the computer translates each line of program code afresh for each execution of the program. The translation process from BASIC to machine code often formed a speed bottleneck, though. It was just a short time later that Apple introduced its version of Pascal, a compiled language. In compiled programs, the translation is made once and saved. Thus, the "file" of machine code already exists, does not have to be interpreted again, and typically executes faster. However, there is a downside to compilers.

With compiled programs came larger programs and more complicated compilers and debuggers. This drove the market for more powerful compilers. As the compilers got better, the programs grew even larger. We are now at a point in

www.allitebooks.com

computing where it is very difficult to develop simple programs. The compilers are too complicated for the casual user. Use Xcode for Mac or Visual Studio for PC and you'll see what we mean. When you find a product that is simple to use, like REALbasic for the MAC or PC, you will have to lay out several hundred dollars to buy and use the product. If you are a serious programmer, these products are well worth the investment in time and money. But what if you are a guy or gal who has this idea for a neat little program that you want to write primarily for your own use? Is that simple application you had in mind really worth the cost?

On the other hand, even without the Apple Dashcode product, which is the focus of this book, widgets are still considered relatively easy to create. Why? Well, first because they use language components that are web-based and interpreted. These components include HTML files (`.html` file extensions), Cascading Style Sheets (CSS) files (`.css` file extensions), and JavaScript files (`.js` file extensions), plus several other types. That means that no language compiler is required. The lack of a compiler translates to (1) no learning curve for the compiler product, (2) no financial outlay for the compiler product (Xcode is, of course, the exception), and (3) easy project creation without the need for multiple time-consuming compiles.

Focus

Widgets tend to be focused entities. By focused, we mean that most widgets are simple, relatively short applications that focus on a specific purpose. For example, instead of a large spreadsheet application that is general in nature and capable of many different feats, a widget would more likely be a program that adds, converts, or manipulates data in a specific manner. So, while the spreadsheet application could perform lots and lots of general tasks, it just might be overkill for the user.

A simple widget might be written to convert temperatures or a number's radix from one base to another. Widgets can also be used to read an RSS feed, indicate hard disk access, check on a user's IP address, and so on. Widgets can be found, or created, to do all of those neat little tasks that we never want to open a spreadsheet or word processor in order to perform. Widgets are fun!

www.allitebooks.com

Functionality

Because widgets are focused on specific tasks, their functionality is usually easy to maintain. In this sense, functionality means how easy it is to develop and maintain a widget's capabilities (i.e., to get and keep that widget operating as planned).

Dashcode will free you from a lot of the mundane programming tasks that have hampered developers in the past. That is not to say there will never be programming code to deal with, as you will see in Chapter 5, but that the code you deal with will be under control.

Dashcode creates project components that fit together in the OOP (object-oriented programming) model, where most of the action takes place when events are fired.

For example, you might have an application that converts inches to centimeters. The user enters a value in inches—then what? Typically, a button is clicked with the mouse, which fires the event. The event is the mouse click. This is where we write the JavaScript code to do the actual conversion. That kind of coding requires a little knowledge of JavaScript (see Chapter 5), but it is manageable.

"Okay," you ask, "Where does the functionality come in?" Well, in this application, there will probably never be a problem because of the fact that 1 inch will always equal 2.54 centimeters. But what if you were designing a program to convert dollars to euros? In this type of application, the exchange rate between the two can change.

In the first application, the user never has to see the fact that 1 inch equals 2.54 centimeters—it can just be "hard-wired" into the JavaScript code. In the exchange rate program, however, you would include another value to be input by the user—the current exchange rate. Then, each time the program is used, the user would enter the dollars to be converted and the exchange rate.

An exchange rate program with *real* functionality, however, wouldn't require the user to know the exchange rate—rather, it would know where to go on the Internet to get the current exchange rate for the user. As long as that location did

www.allitebooks.com

not change, the functionality of the application would remain intact for the life of the application.

Cost

How much does a widget cost? Well, are we referring to the cost of developing the widget? Buying the widget? Let's consider both of these cases, because they have both contributed to a widget's popularity.

First, since we don't need a compiler to create a widget, there is no cost involved with a product such as that. Since Apple provides the framework (editors, the Safari browser, etc.) for developing web-based applications, there is no cost there. And since Dashcode is provided free starting with OS X and later, there is no cost there either.

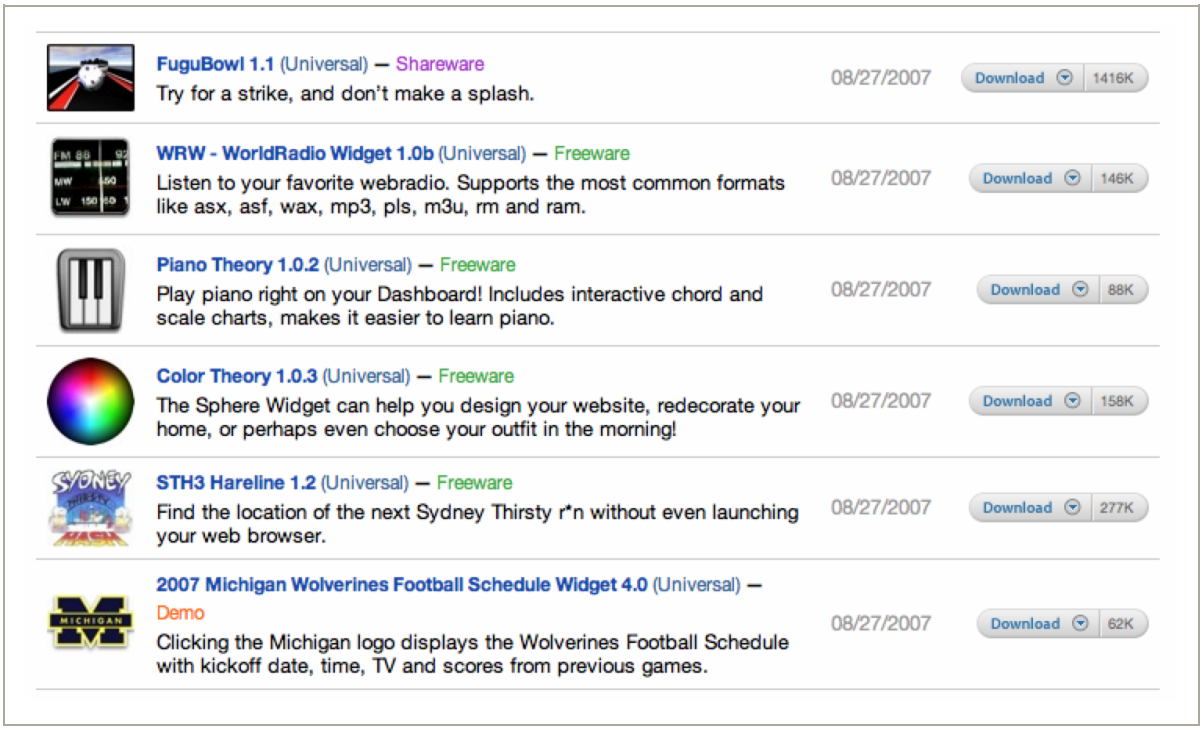
So, there are no out-of-pocket expenses required for creating your own widget applications. Free widget creation tools have certainly gone a long way toward making widgets so popular.

Now, go to the Downloads page on the Apple web site (`www.apple.com/downloads`), and look for the Top Widgets section, in which you can feast your eyes on literally hundreds of widgets. Widgets come in three types: shareware, freeware, and demo, as you can see in Figure 1-2.

Widgets with the shareware attribute are widgets whose developers are usually looking for a contribution. Typically, shareware widgets are fully functional, and the developer is asking for an optional donation. Sometimes some of the widget's functionality (there's that word again) is removed until the donation is made.

Widgets with the freeware attribute are free. That means they are free to download and free to use. There is no guarantee from the creators that the widgets will work for your particular needs, however (after all, they are free). Of the widgets that are available, we would guess that at least 80 percent of them are freeware widgets. It doesn't get any better than free.

Figure 1-2. A sample of the three types of widgets that you can download



The third category is a widget with the demo attribute. These are usually mini-programs developed commercially or by a developer who really spent a lot of time on their project. If you go down the demo road, plan to lay out some cash in order to make your download functional.

So, in summary, widgets cost nothing to develop (in terms of tools), and people have posted hundreds of free widgets on Apple's site. If everything is so "free," why would anyone want to write their own? Ah, because you have that idea for a widget that no one has created yet! If your widget turns out to be a big success, you might consider changing it into shareware or a demo.

Implementation

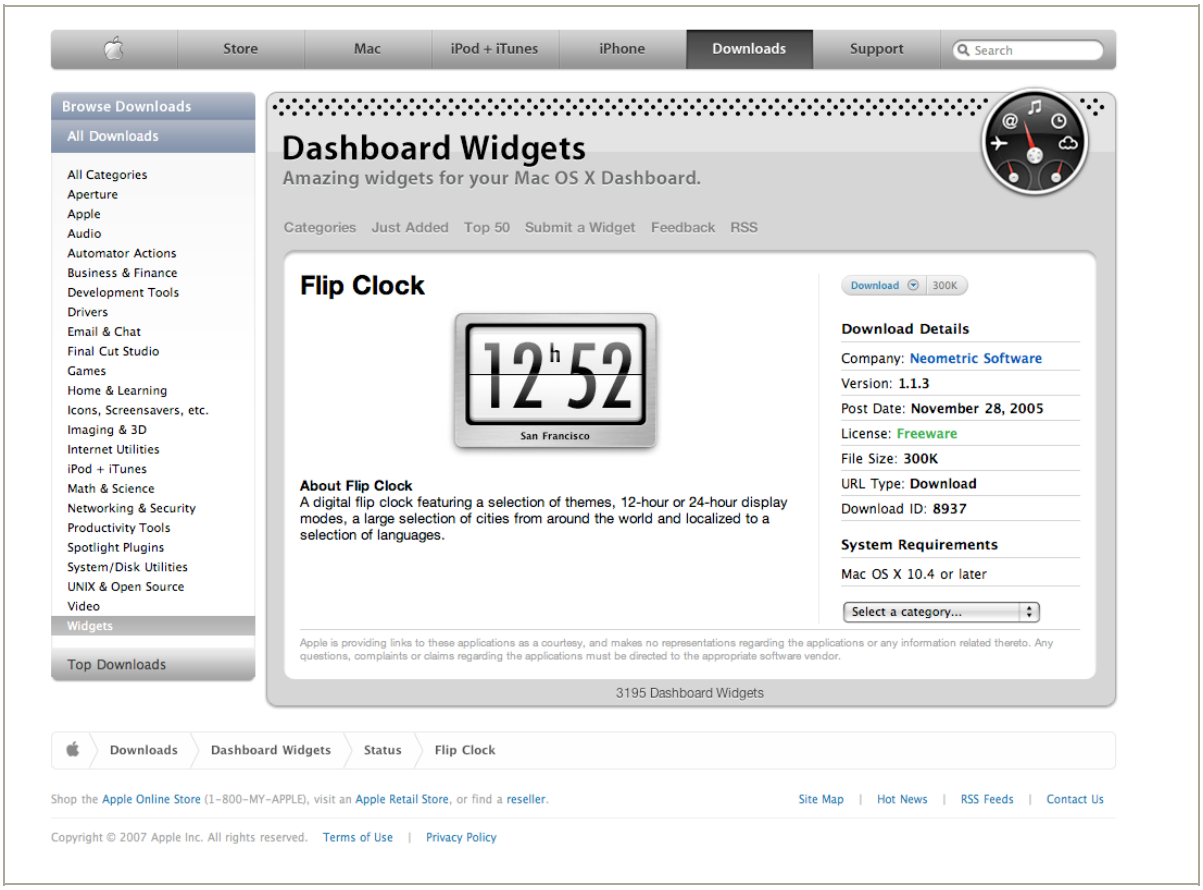
Imagine that you have just found the perfect widget for your needs on the Apple web site. How do you get it off of the web site and installed on your computer? Actually, that is a piece of cake.

Figure 1-3 shows the Downloads portion of the Apple web site, mentioned earlier.

Figure 1-3. Apple's Downloads page showing widget selections

To find widgets, select the Widgets box or select from the Top Widgets listing. For this example, we chose the Flip Clock widget from the Top Widgets list.

Figure 1-4. The download screen for the Flip Clock widget. Note that this widget is freeware.



If you click this selection, you will be taken to another screen, as shown in Figure 1-4.

If you click the Download button (near the upper right of the screen), the widget will be downloaded to your computer. Typically, you'll find downloads such as this in your Download folder. After downloading, you should see two icons in your Download folder, as shown in Figure 1-5.

Figure 1-5. These two icons should now appear in your Download folder.

Double-click the Flip Clock icon to begin the installation of the Flip Clock widget on your computer's Dashboard. When you do this, the installation process will start automatically, and you will be greeted with the screen shown in Figure 1-6.

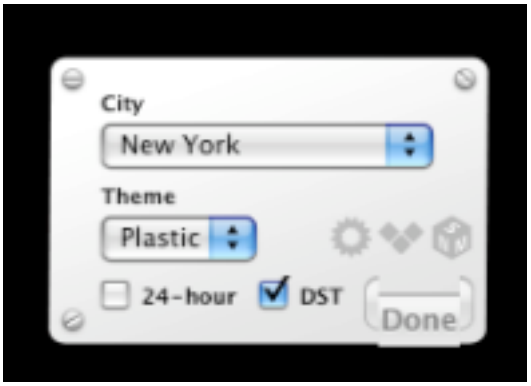
Figure 1-6. The Widget Installer dialog for the Flip Clock widget

The installation process occurs almost instantly. When Dashboard is opened, you will see the default Flip Clock widget, along with any other widgets already installed. Figure 1-7 shows this widget.

Figure 1-7. The default Flip Clock display

If you select the information button on the widget, represented by the (i) symbol, you can flip to the back of the widget and enter new information, as illustrated in Figure 1-8.

Figure 1-8. Changing the defaults of the Flip Clock widget to DST in New York City



When you have made all of the desired changes, click the Done button to flip the clock back around. Figure 1-9 shows the final widget as it will appear in Dashboard.

Figure 1-9. The Flip Clock widget is customized with our settings.

Basically all widgets are downloaded and installed in the same manner. Pretty simple, right? Now you have confirmed another reason why widgets are so popular.

Distribution

So, that widget you developed turned out really well. All of your friends have encouraged you to share it with the world. How do you do that? If it were a new song, you would go to a record producer. If it were a great book, you would seek out a publisher. With a widget, however, you can go directly to Apple!

Go to `www.apple.com`, select the Downloads menu item, and choose Widgets from the rightmost column. Your screen will look something like the one shown in Figure 1-10.

Examine the page and notice the Submit a Widget menu item near the top of the screen. You can click this menu item when you have your widget ready to submit. Clicking it will transport you to Apple's Developer Connection page, similar to that shown in Figure 1-11. In order to be able to submit software to Apple, you must be a member of the Apple Developer Connection. This is rather painless because you can get a free membership for just registering. So, if you have not registered as a developer yet, now is the time to do so. After all, you *are* a developer!

Figure 1-10. The Dashboard Widgets page from Apple's Download site

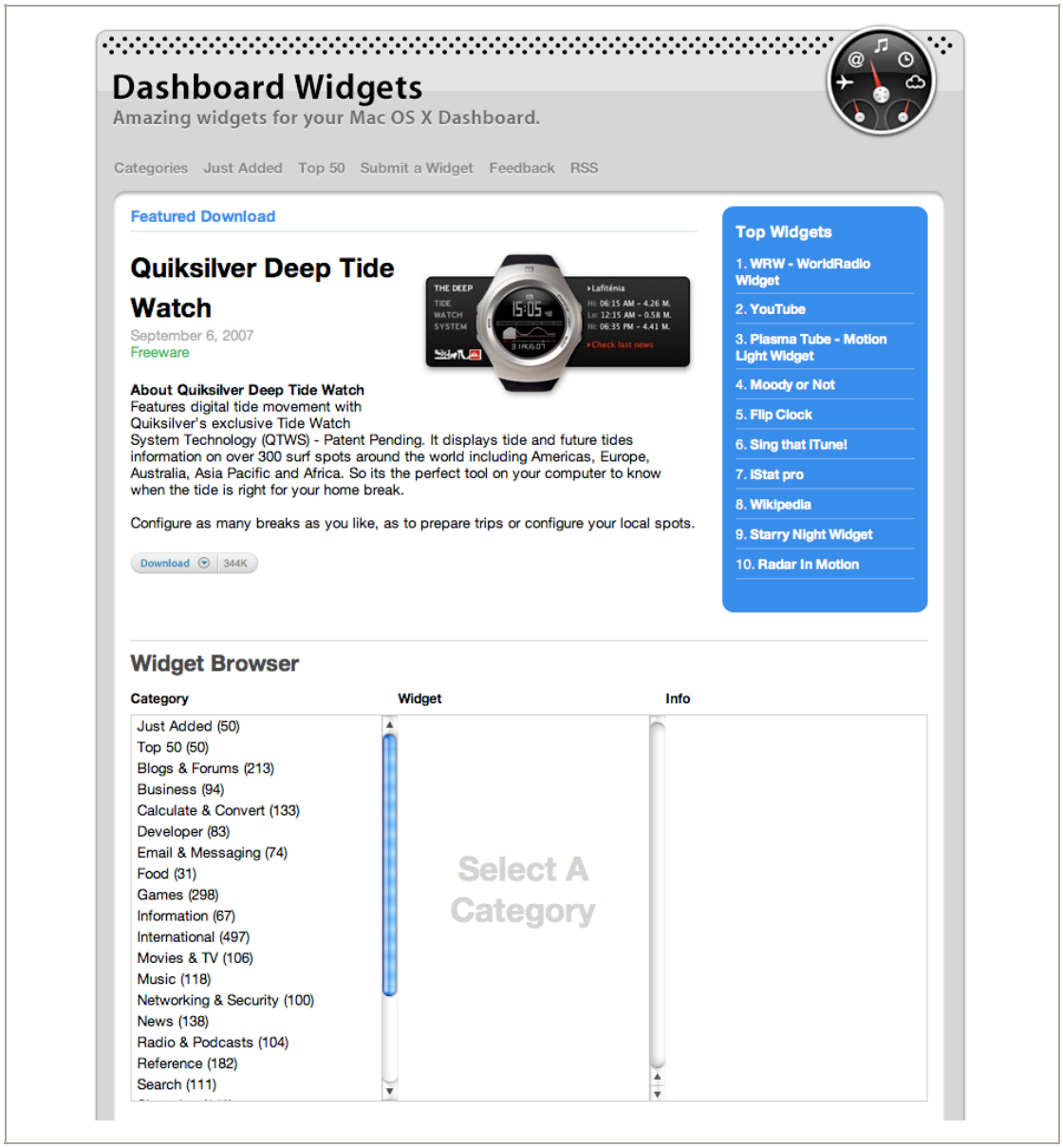


Figure 1-11. The Downloads Submission form requires membership in the Apple Developer Connection.

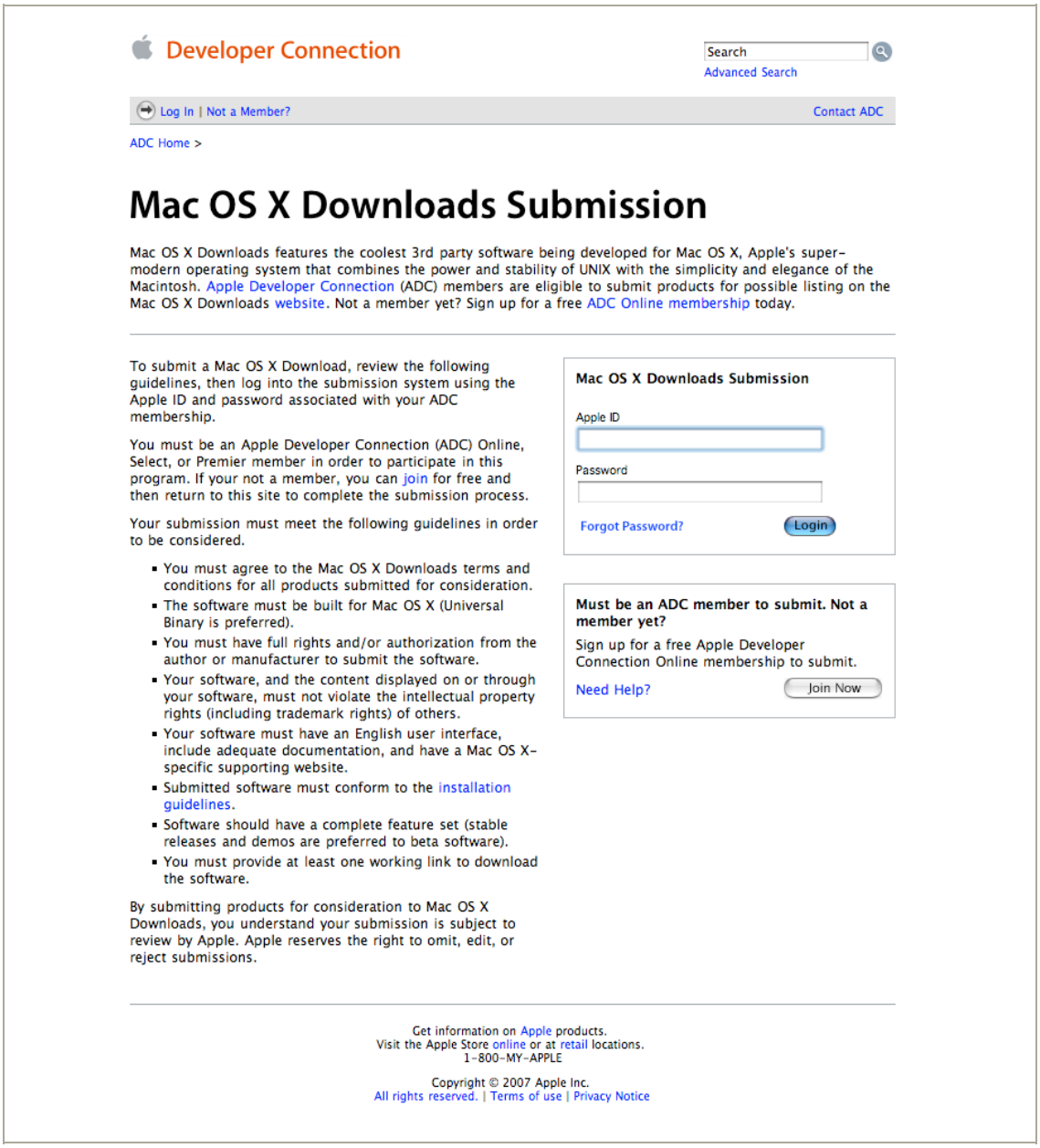
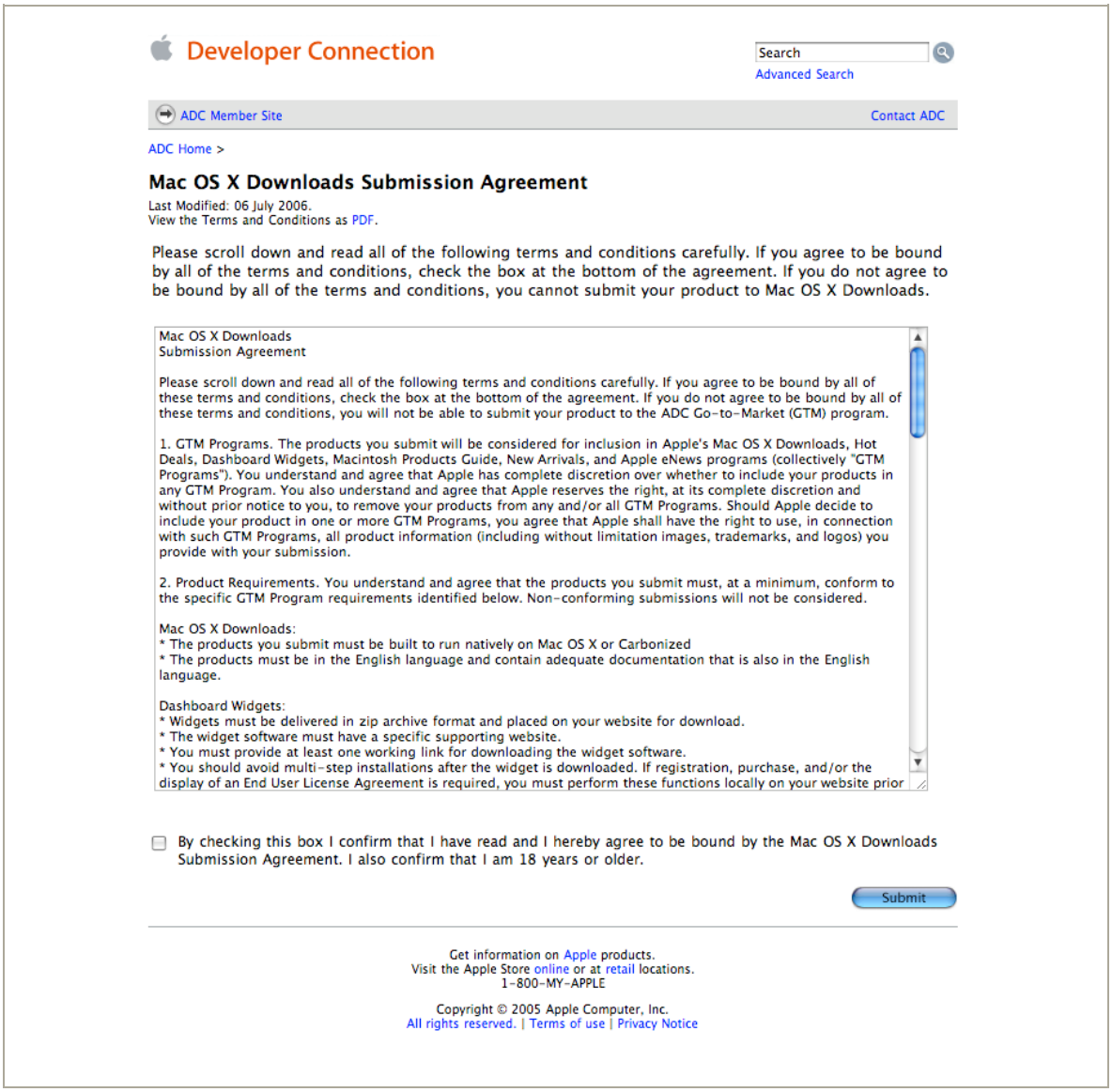
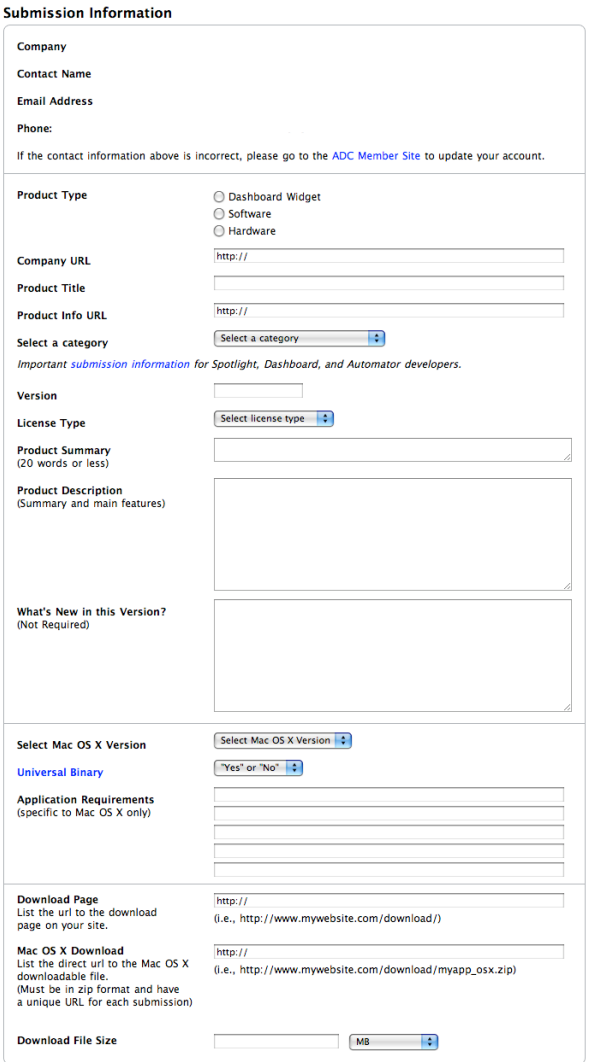


Figure 1-12. The download agreement for the widget you wish to post to Apple's site



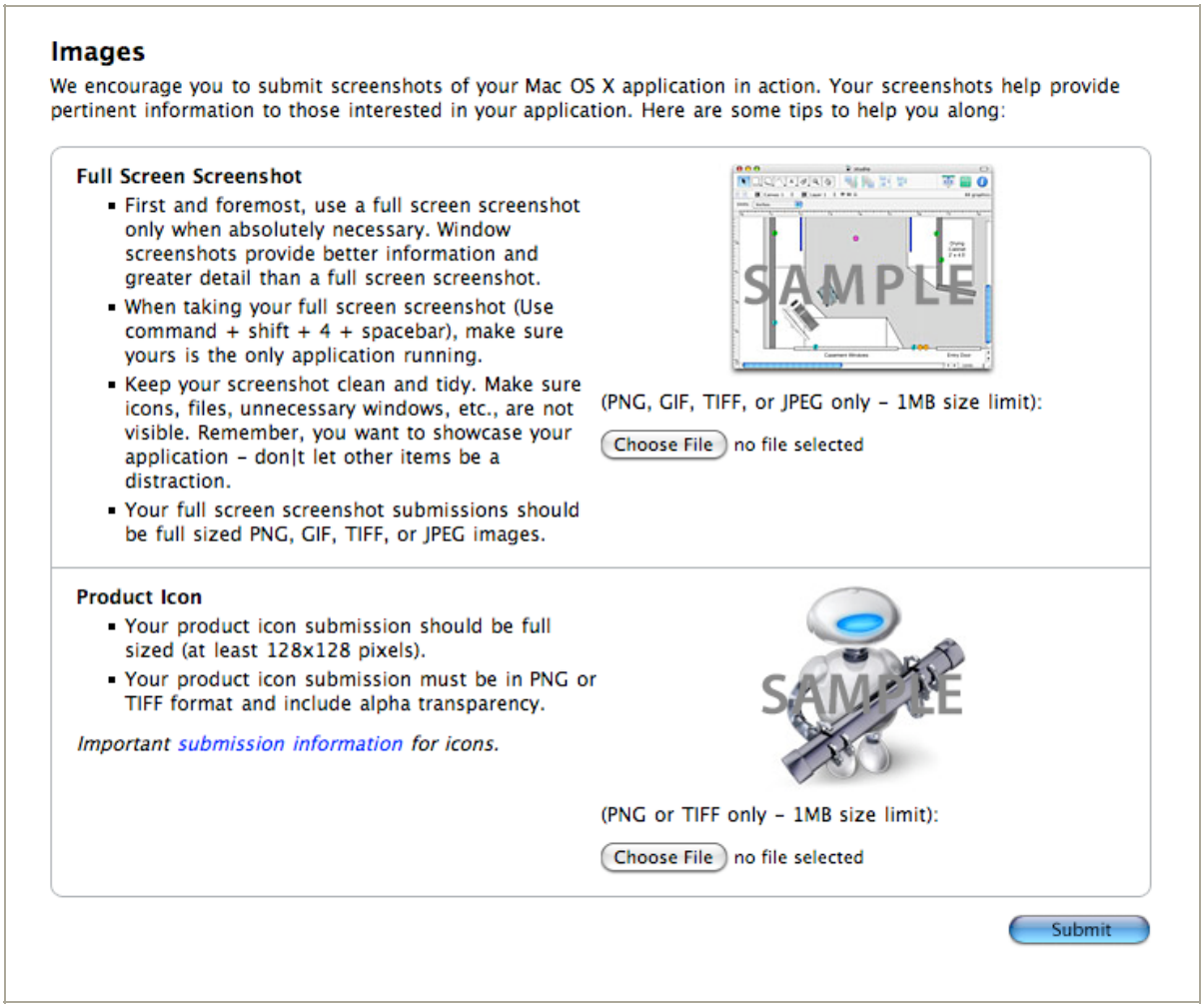
Once you register and sign into the Developer Connection, there will be a legal agreement you must agree to, similar to the one shown in Figure 1-12.

Figure 1-13. A portion of the submission agreement for your widget



Once the legal agreement is completed, you must provide Apple with specific submission information for the widget. Figure 1-13 shows a portion of the form that must be completed.

Figure 1-14. The remainder of the submission form with specific image instructions



Apple also requests that you submit a screenshot and product icon with your widget. The remainder of the submission form, including specific image instructions, is shown in Figure 1-14.

That's basically the process for uploading your widget for distribution from Apple's web site. OK, it's not as easy as downloading a widget, but it's not that

bad! And if you don't want to share a widget with anyone but your friends, the whole process can be avoided.

The Look and Feel of a Good Apple Widget

Have you ever used software that you just couldn't make heads or tails of? Software with such bad instructions and such a nonintuitive user interface that even experts would throw up their hands? Our dentist is computer savvy, and yet we frequently get calls for help about this or that application where the next step just doesn't make sense. He knows what he wants to do, but the path has just not been made clear. You do not want this to happen with your widgets.

Apple makes suggestions for producing good human interfaces for your widgets, and we're going to list a few of those recommendations here. Starting with Chapter 2, you'll see how we make use of these guidelines, as we discuss why we do things a certain way instead of another.

There are typically two faces to every widget. The main face is called the *front face* or *front side*. The other face is called the *back face* or *back side*. The user interfaces with the widget's front face and sets widget preferences on the back face.

Front Face Design Ideas

Here are several suggestions for front side widget design:

- Space is scarce, so design your widget to quickly convey its purpose. Place components on its face logically and keep text to a minimum, showing only essential information.
- Design your widget so that your dentist can use it—just kidding, but a widget should be totally intuitive and not require additional instructions for its use. For example, sometimes sample data can be entered in text boxes to illustrate how data is entered in the widget.
- Limit widget capabilities, or as Apple calls it, use "discrete functionality." What this means is that a widget has a specific, focused purpose—keep it

simple. If you want to create a spell checker and thesaurus, perhaps two separate widgets would be better than one.

- Keep widgets as small but as functional as possible. Remember that not everyone has a 30 inch Apple monitor. In a similar vein, users sometimes want more than just your widget to fit on their screen. Common sense, huh?
- It is always a good idea to avoid what Apple calls "garish" colors. So, maybe designing that widget with purple, red, and yellow might not be as good as a family of greens and blues. Likewise, fonts should be picked and sized so that they are easy to read, especially against that background of greens and blues.
- Stay with the controls provided with Dashcode and avoid Aqua controls if possible. Apple suggests using Aqua controls on a widget's back face only.

Back Face Design Ideas

The back face of a widget is often as important as the front face. Users typically set widget preferences on the back face. Here are some ideas for back face design:

- By default, Dashcode will create a back face for your widget. Plan to use it!
- The information button graphic on the front face notifies the user that there is a back face. This is the icon that appears somewhat like (i) on the front face of the widget.
- Allow users to set widget preferences on the back face. You can also use the back face for short, specific instructions, advertising, and licensing information. Don't be shy, you might be able to grow a profitable business with the right group of widgets.
- The Done button will flip the widget back to the front face. Make sure you keep the button provided by Dashcode.
- Apple suggests that subdued colors be used on the back face of a widget, leaving the brighter colors for the front face.

Miscellaneous Design Ideas

Here are some additional general suggestions for widget design:

- Do not place about information, advertising, or licensing information on the front face—use the back face of the widget instead.
- Keep everything about your widget pithy.
- Provide a suitable icon for your widget.

What Widgets Can and Cannot Do

Generally speaking, if it can be done on a web page, the same functionality can be designed into a widget. As discussed earlier, this is because widgets are based on web technology using integrated components of HTML, CSS, and JavaScript.

Additionally, widgets can tap into Apple's Dashboard APIs (application programming interfaces) in addition to standard web APIs.

The Problems with Widgets

The problems you are most likely to encounter with widget use fall into two categories: clutter and widget removal.

Clutter

Here is the plain truth: widgets are designed to appear on the Dashboard screen. There is only one Dashboard screen! How big is the monitor that you are using? If you are content with ten widgets, you will probably not notice a problem. However, if you have fallen in love with 40 or 50 widgets, real estate is going to be at a premium on your monitor when you open the Dashboard window. Clutter—it's everywhere! And trust us, if widgets are freeware, you are going to find a lot of them you want to use.

Unless you are using a special tool to help you manage your widget collection, sooner or later you are going to have to start with the house cleaning. So, how do you get rid of a widget?

Widget Removal

"OK," you say, "I'll clean things up." After all, how many times did you really use that widget to help you select a random flavor of jelly for this morning's toast? We'll bet there are a lot of those types of widgets on your Dashboard screen, because we have to go through and periodically clean out our unused widgets, too.

The X (typically located in the upper-left corner) is what you click to remove a widget from Dashboard. But there is a potential problem with this—what if the X isn't visible on your widget? The trick is to place the mouse cursor over the widget and use the Cmd++ key combination. (Note that the Cmd+Alt/Option combination also works). The X will pop up and the widget can then be removed.

More Excitement Coming

Now you know what widgets are, their advantages and disadvantages, how they are installed in Dashboard, and how they can be deleted. "But," you ask, "How do you make a widget?" Widgets can be made with Apple's free Dashcode widget generator. The best news is that it comes free on the installation disks for OS X 10.5 and later.

Let's go on to Chapter 2 where you'll learn some important details about Apple's Dashcode developer application.

Chapter 2: Widgets Without Worry: Using Dashcode

In Chapter 1, you learned that the platform for installing and using widgets is Dashboard. In this chapter, you will learn that Dashcode is the Apple product used to automatically and easily create widgets.

In addition, you will learn how to install Dashcode, how to use Dashcode, fundamentals of the Dashcode product, how to implement various Dashcode design concepts, and most importantly, how to create a couple of simple but useful widgets.

Dashcode comes with a variety of widget templates. In Chapter 3, we'll discuss each of these templates and see how they can be put to good use. Widget templates will save you time and frustration on the road to creating your "perfect" widget.

What Is Dashcode?

Dashcode is an Apple development tool that automates the design process for widget creation. Dashcode takes the drudgery out of widget design and implementation by automatically creating the necessary widget files, and then tying them together into a final seamless application. If you have had reservations about getting started with widget development, wait no longer. Dashcode is the product you have been waiting for and it's free!

Where Is Dashcode and How Is It Installed?

Dashcode is an Apple development application provided on the installation disk for OS X 10.5 and later.

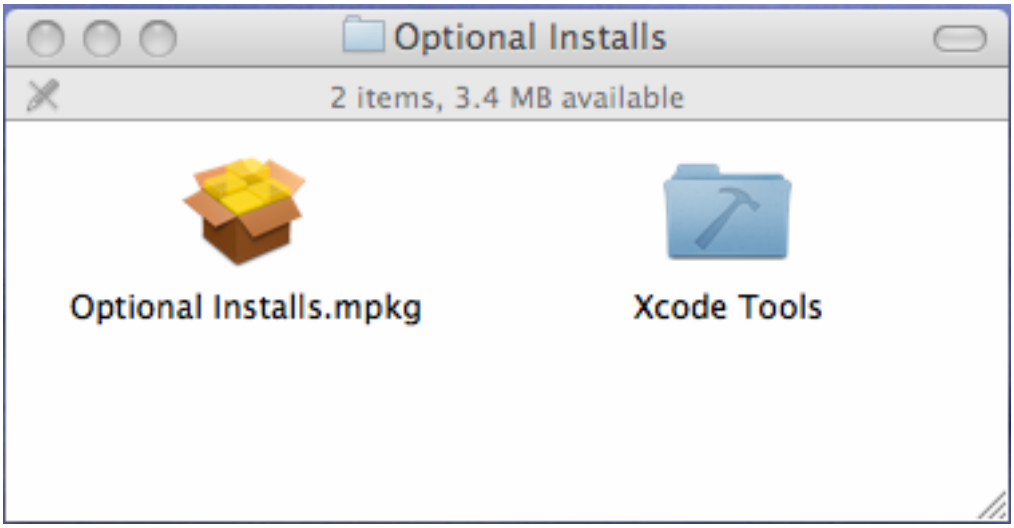
Insert your OS X 10.5 installation disk into your DVD drive and wait for a screen like Figure 2-1 to appear.

Figure 2-1. The installation startup screen for Dashcode

Click the Optional Installs icon, as shown in Figure 2-1. This will bring up a screen that will allow you to select the installation of various Xcode tools, as shown in Figure 2-2.

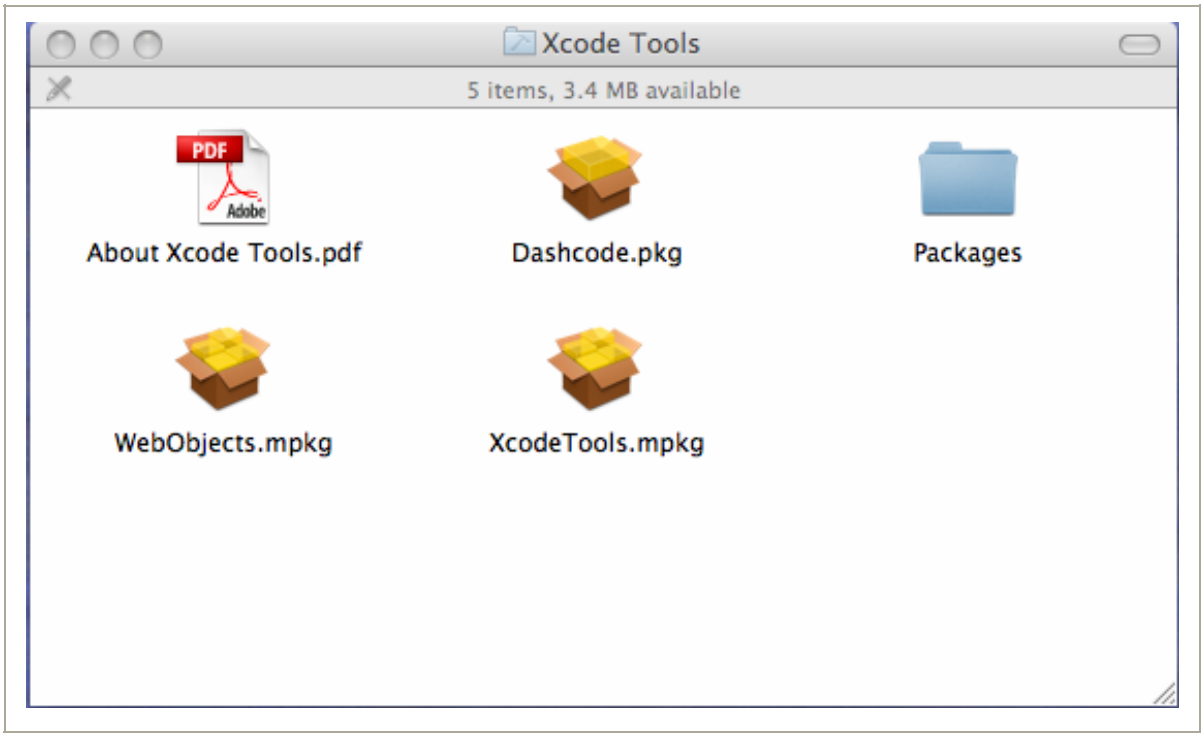
www.allitebooks.com

Figure 2-2. The Xcode Tools installation icon



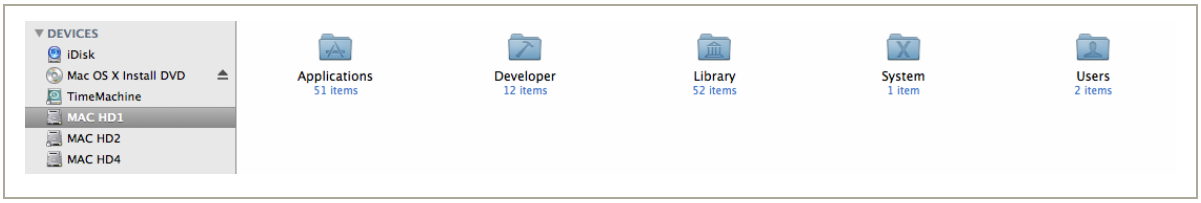
Click the Xcode Tools icon to start the process of adding Dashcode. From the host of Xcode installation options, select the Dashcode.pkg icon, as shown in Figure 2-3. When you do this, the process will automatically install the Dashcode application in your default installation drive under the icon Developer.

Figure 2-3. Once you select the Dashcode.pkg icon, you will see the Dashcode.pkg installation dialog box.



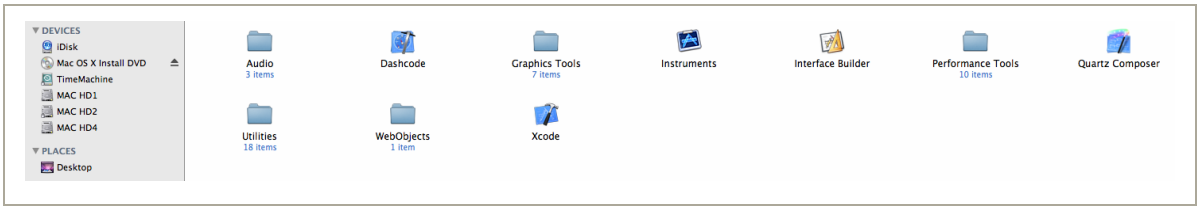
Have your computer opened to the root directory in order to complete the onscreen installation process. Our root directory is named MAC HD1. Now, select this directory to find the Dashcode application and any other Xcode tools you have decided to install. When this directory opens, you will find a Developer icon, as shown in Figure 2-4. Select this.

Figure 2-4. Use the Developer icon to find all Xcode tools.



Double-click the Applications folder to open it. The Dashcode product icon should be sitting near the top of this directory, as illustrated in Figure 2-5.

Figure 2-5. The Dashcode icon is used to open the Dashcode product.



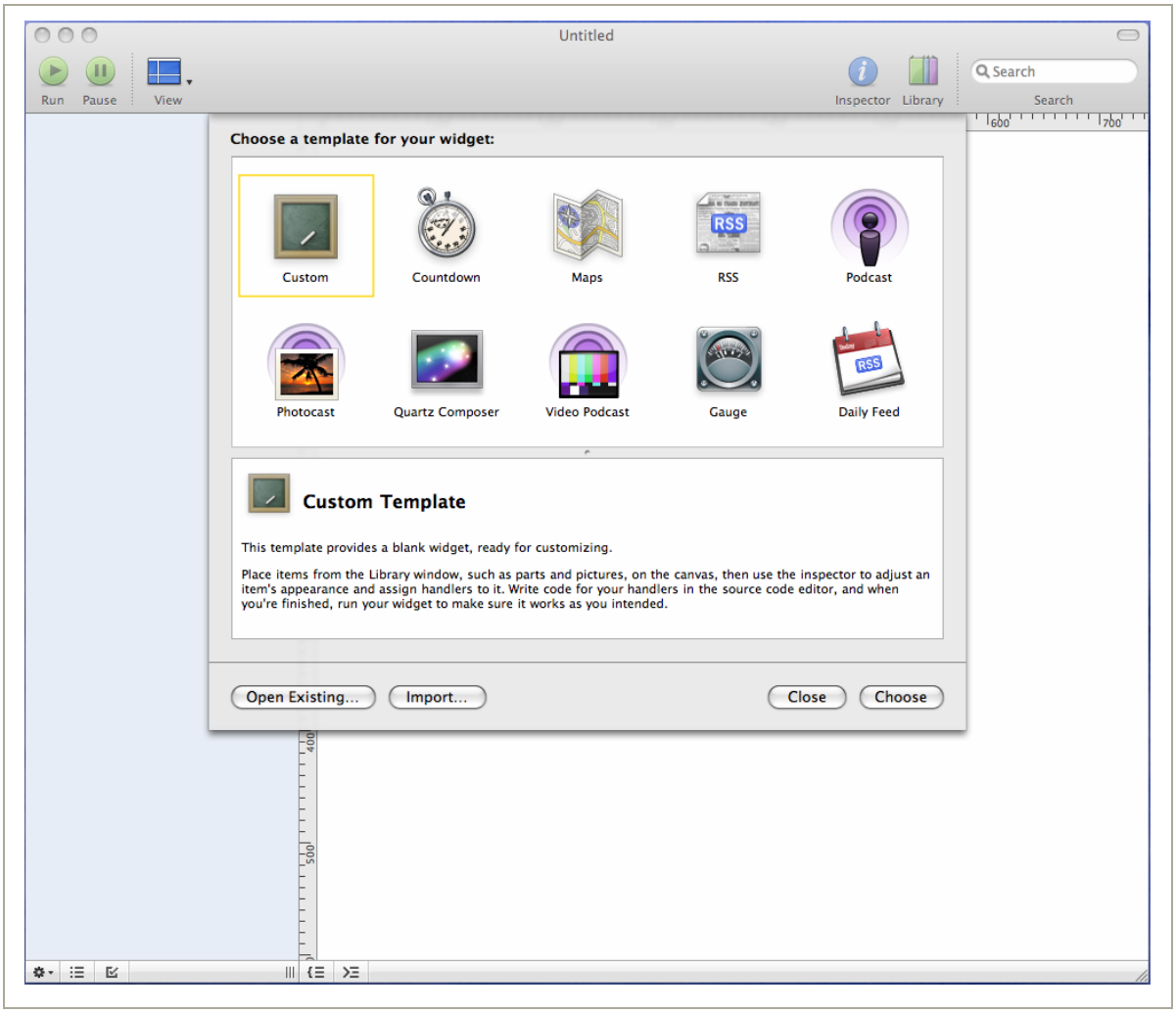
Since we are going to be using this application many times in the book, why not drag it to your computer's dock so that it's always just a click away!

Well, that is all there is to it. You have found and installed the free Dashcode application from Apple that will make all of your widget design so easy to implement.

A Quick Look at Dashcode Features

Go ahead and click the Dashcode icon to start the application. You'll be greeted with a screen that looks like Figure 2-6.

Figure 2-6. The initial Dashcode design screen



The initial default Dashcode screen allows you to select a design template from a group of ten or more candidates. From this group, the Custom Template is the default. Let's select the Custom Template for our initial widget.

Notice that the description immediately below the template selection window describes what this template will do for your widget creation process. Dashcode provides the following information:

> *This template provides a blank widget, ready for customizing.*
>
> *Place items from the Library window, such as parts and pictures, on the canvas, then use the inspector to adjust an item's appearance and assign handlers to it. Write code for your handlers in the source code editor, and when you're finished, run your widget to make sure it works as you intended.*

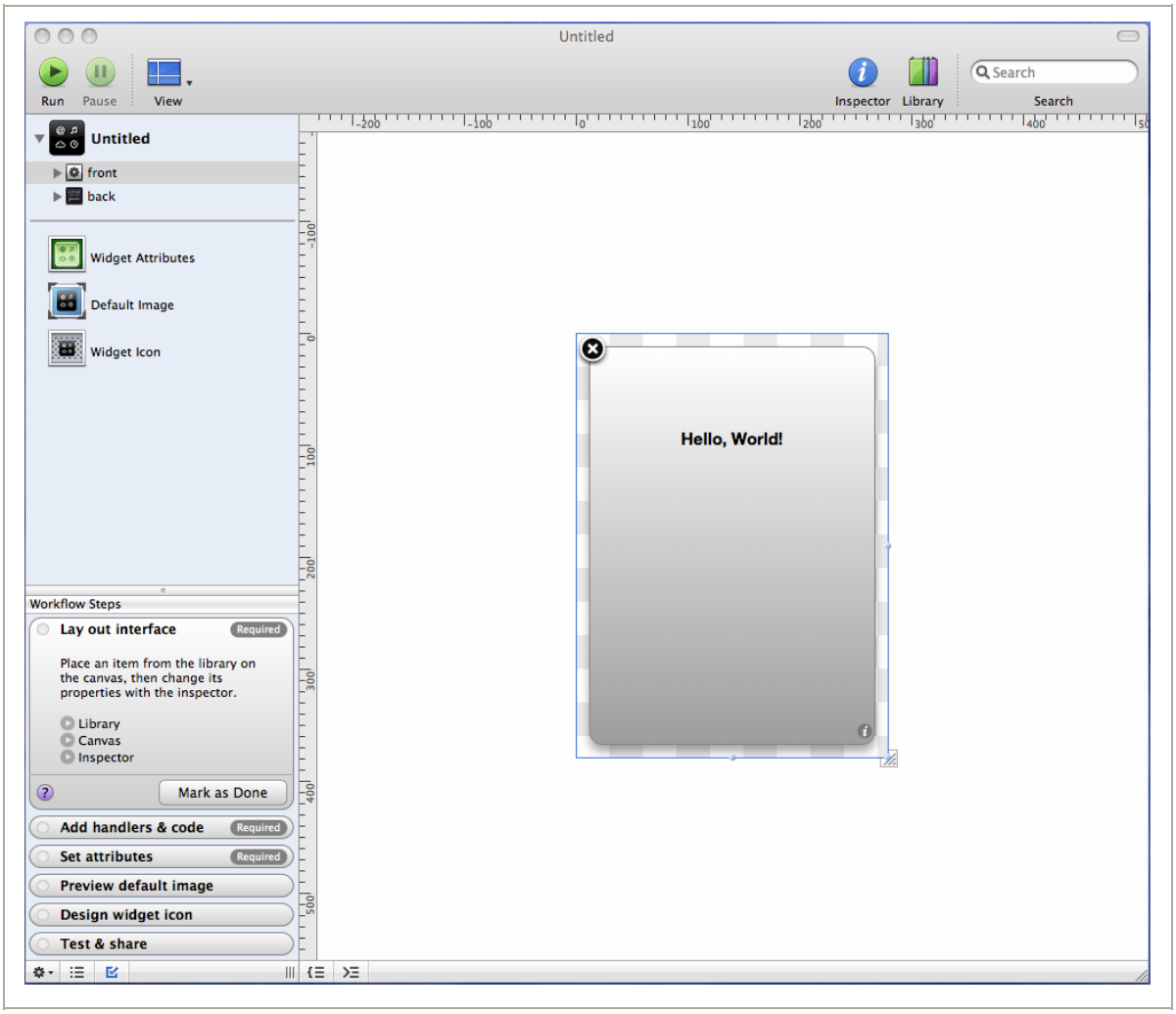
Basically, the Custom Template is a blank template that will allow you complete design freedom in creating your first widget. The other templates will provide functional code for doing specific things, such as inserting pictures, podcasts, and so on.

Select the Choose option and bring up the Custom Template in Dashcode. You'll see a design palette like that shown in Figure 2-7.

Now, without your doing anything additional, the Custom Template could be turned into a functional widget using Dashcode. To see what the default widget would look like, click the Run button on the upper left of the Dashcode design palette. Somewhere on your screen should magically appear the Hello World widget.

"But wait," you say, "How can a widget appear on my screen *instead* of in Dashboard?" The answer is simple. The code that you are looking at is basically HTML code with the graphical overlays included. At this point, it is not a true widget—but in this format, true enough to give you a peek at what the final widget will look like. Secondly, you have not gone through the formal installation process for installing the widget in Dashboard. But we're about to take care of both of these objectives!

Figure 2-7. The initial custom widget with "Hello, World!" text visible

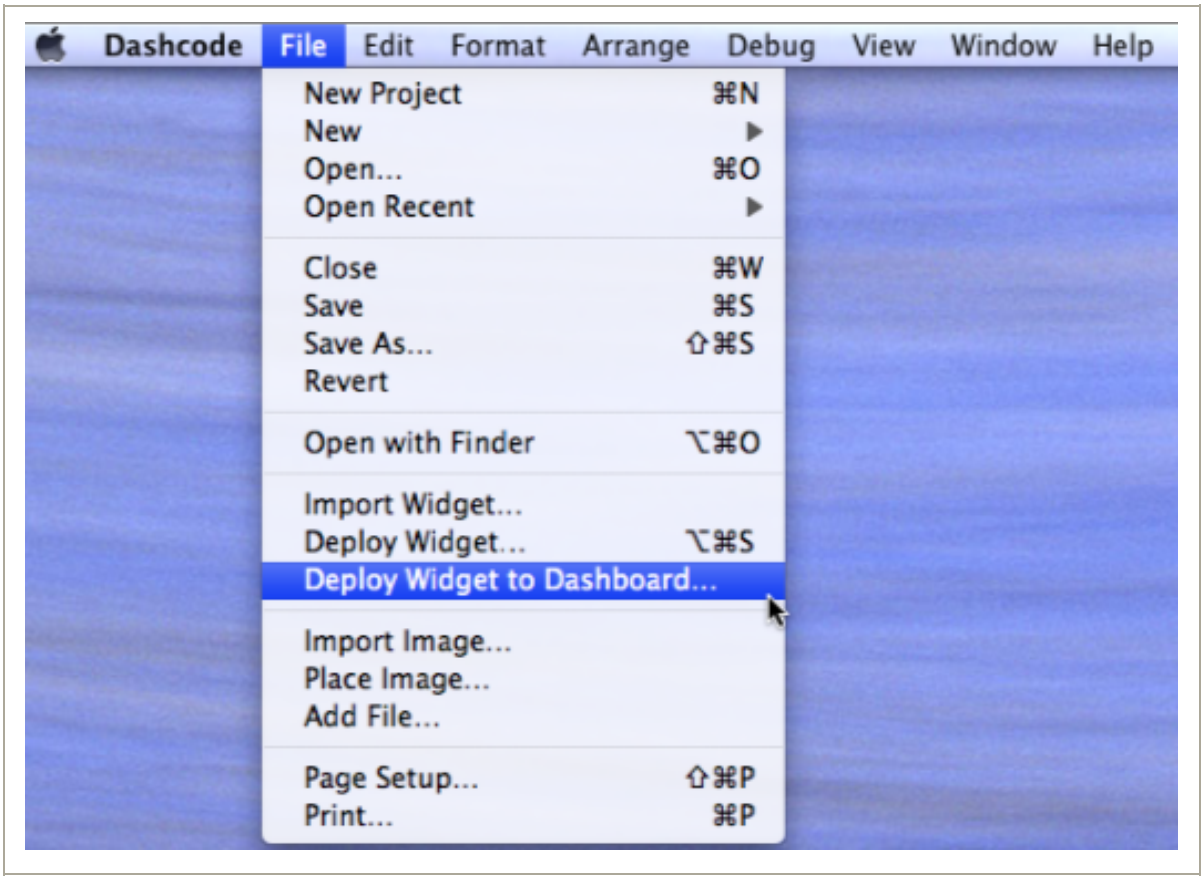


If you have spent sleepless nights longing for a Hello World widget, your dreams are about to come true. In order to deploy your Hello World widget to Dashboard, select the "Test & share" option, as shown in Figure 2-8.

Figure 2-8. Selecting "Test & share" to test the widget

From the "Test & share" option, click the Mark as Done button. Now use the Dashcode menu to select the File ➤ Deploy Widget to Dashboard menu item, as shown in Figure 2-9.

Figure 2-9. You can deploy the widget by selecting the Deploy Widget to Dashboard menu item.



You will now be greeted with an installation dialog box, as shown in Figure 2-10.

Figure 2-10. The installation dialog box installs your new widget to Dashboard.

Selecting the Install option will open Dashboard and show the Hello World widget. It is now your option to delete the widget (abort the installation process) or keep it, as shown in Figure 2-11.

Figure 2-11. Choose the Keep option to finish installation.

Of course we're going to keep the Hello World widget. Just like the first dollar bill earned by businesses is pasted on the wall, our Hello World widget will be a

Figure 2-12. The Dashboard screen with the newly installed Hello World widget

treasure to be admired for years to come. Figure 2-12 shows our Dashboard screen with the newly installed Hello World widget.

Of course, it might not be the exact widget you were looking for after all. Don't forget, you can still delete the widget by pressing the Alt key to reveal the X symbol on the widget. Removal, then, is just a mouse click away.

OK, to be honest, we really didn't modify the template code or investigate too many Dashcode features. So, let's continue working with our original Hello World Custom Template and see what we can accomplish without too much work.

Front Face Changes

By default, the Dashcode application starts with the front face of the Custom Template. Let's make a few changes here.

Changing the Text and Resizing the Text Box

Double-click the text on the front side image and type in **Welcome to Bill and Chris's World**. Well, actually, type in anything you want. That's all it takes to change the message, as shown in Figure 2-13.

Figure 2-13. Entering a new message in the text box

Now, resize the text box by positioning the mouse over the text box top, bottom, side, or corner, and dragging the boundary. Figure 2-14 shows how we completed our resizing.

Figure 2-14. The resized text box

With these changes in place, let's add a picture in the next section.

Adding an Image

A blank widget can be boring. Let's add a picture. Click the Library icon, near the top left of the Dashcode design palette. Select the Photo option on the new window. It should find your iPhoto library on your computer. You can select any image from those visible in your library. Drag the image and resize it to fit on your Custom Template, as shown in Figure 2-15.

Figure 2-15. The widget with an inserted picture

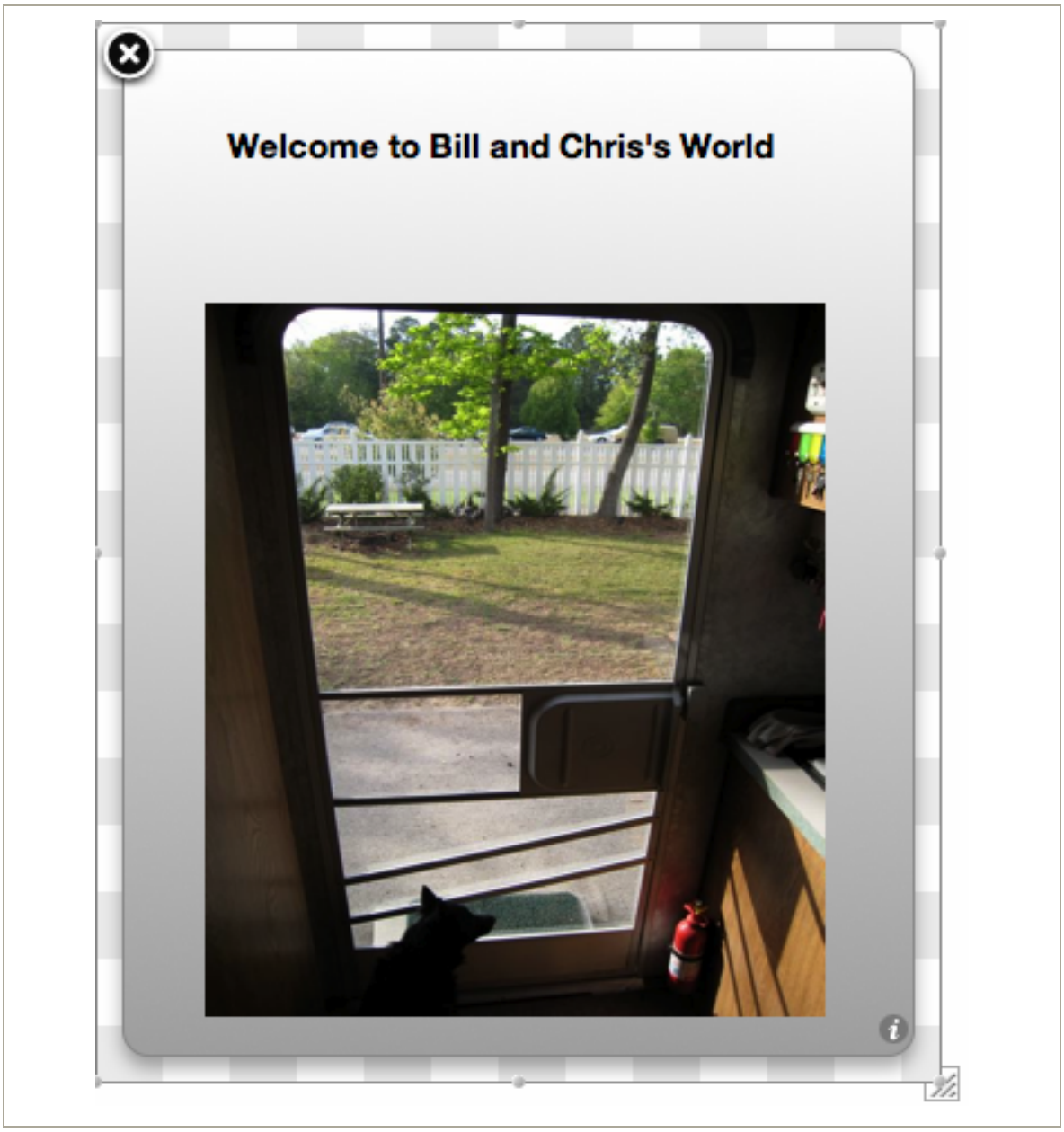
With a picture in place, let's reshape the entire widget.

Resizing the Whole Widget

Place the mouse pointer inside of the design area of the Custom Template and right-click so that the Select All menu item appears. Select this option and click the lower-right corner of the widget. Now click the size adjuster portion of the icon to resize the whole widget to the size you desire, as shown in Figure 2-16.

With the whole widget resized, we resized the original text box and picture to fit, also.

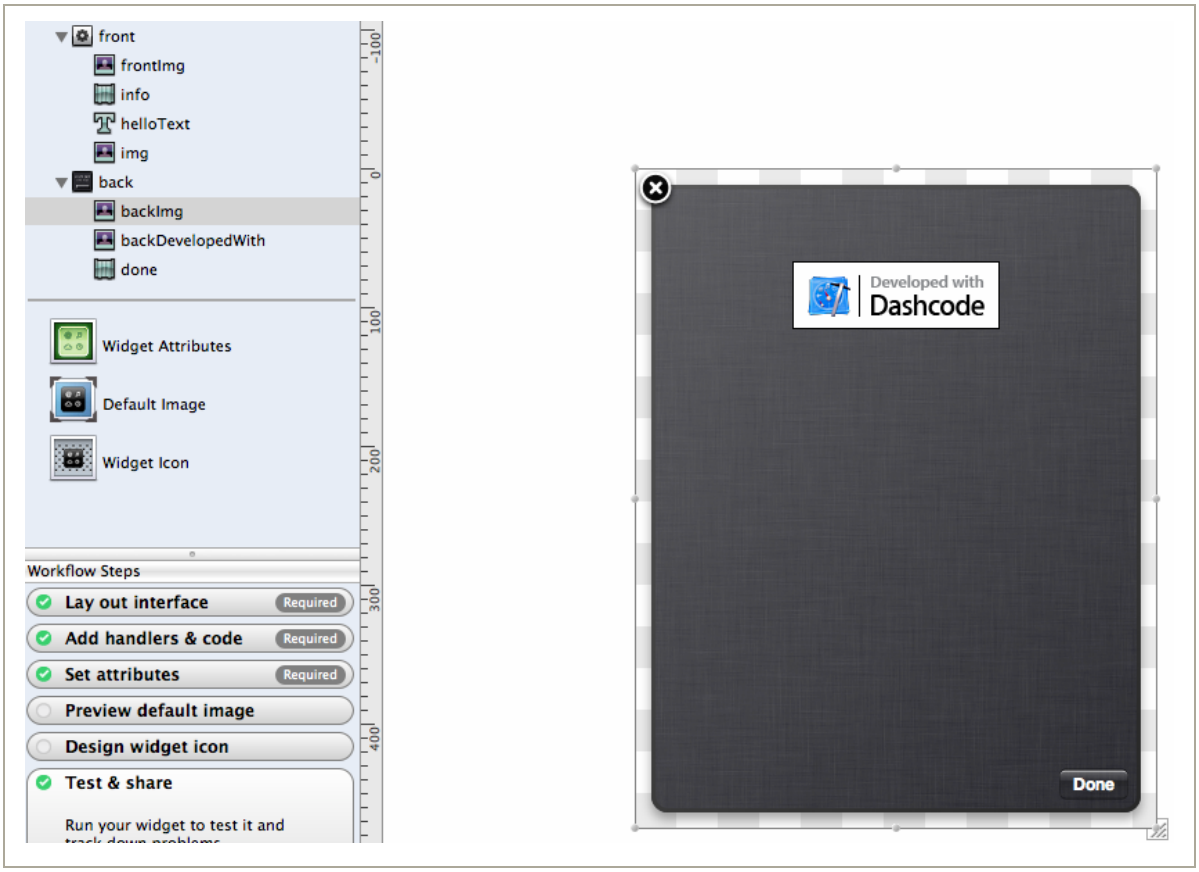
Figure 2-16. Now the whole widget can be resized.



Back Face Changes

The Dashcode application starts with the front face of the Custom Template. What's on the back face? Let's see. To view the back face, expand the back item, as shown in Figure 2-17.

Figure 2-17. The back face of the widget is made visible.



In order to modify the back face, expand the back item and select backImg.

The default image for the back face is a cool Dashcode graphic. Let's keep it! We'll move it just a bit and add some text.

Adding Some Descriptive Text

After changing the location of the Dashcode image for the back face, there will be room to add some descriptive text for our widget.

From the Library icon, select the Parts option and look for the text item. Select and drag that part to the design surface for the back face. Resize it to suit your needs.

While we're here, let's change the color of the text. To change the color attribute of the text, first select the text you wish to change, and then select the inspector at the top right of the Dashcode palette. Use the color option in order to choose the color you desire. Fonts and other attributes can be changed, too, if you desire.

Figure 2-18 shows our final revised Hello World widget created by first selecting the Custom Template option from Dashcode.

In the next section, we'll continue our widget work and take a look at some of the more important Dashcode tools.

Figure 2-18. The revised and final version of the widget

Important Dashcode Features and Tools

In the previous section, we experimented with the Custom Template and altered various component attributes on our design canvas. Let's approach this process a little more formally in this section by highlighting some of the most important design features and tools in Dashcode.

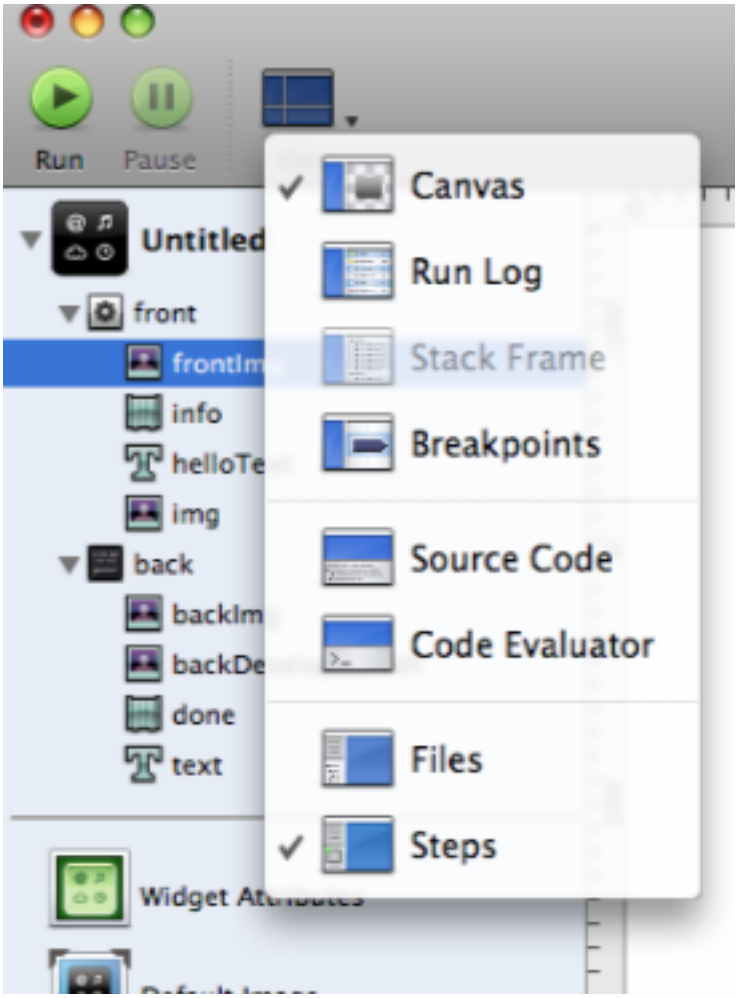
Some of the most important Dashcode features and tools are accessed using the icons at the top of the design palette. In addition to Run and Pause, there are currently three others: View, Inspector, and Library.

View

Don't click this icon! Oh, you did it anyway. Are you sorry now? The View icon, shown in Figure 2-19, allows various views of the components that make up your widget project.

www.allitebooks.com

Figure 2-19. Views that can be used in the creation of a widget



This feature reminds us of a popular commercial for an adjustable bed. A husband and wife are in the bed together. He says, "The bed arrived with a setting of 45—why change it?" She says, "I like to live dangerously. One night I try 27, another night 68, and so on."

The point? You have the front face view and the back face view—that's what Dashcode arrived with. Do you want to live dangerously? If the answer is yes, you can view the Run Log, Stack Frame, Breakpoints, Source Code, Code Evaluator, Files, or Steps windows. However, most of the time you will not need these features.

As an example, load your Hello World widget project, and select Source Code. A window will open showing the Dashcode-generated program code for this project. It is possible for you to now tweak this code to suit your needs, but you're really far away from needing to do anything like this right now.

The real design power is available from the Library and Inspector icons. Let's take a look at them.

Library

The library is the place to be! The Dashcode library allows you to add parts, code, and photos. You saw in our first example how we added a photo image from our iPhoto library to our first widget. So too can we add additional parts and code.

Parts include libraries of buttons, shapes, containers, media, and text. The Buttons part box includes a variety of buttons that you can use to activate an item (fire an event), in addition to gauges, indicators, and levels. The Code part box contains small segments of code that you can use to implement features such as copy and paste. Finally, the Photo part box allows you to add your personal images to any widget project.

So, the library is important because it allows us to add important parts to our widget projects. Well, then what does the Inspector icon do? Probably not what you think. The inspector is closely tied to the parts you add to your widget project from the library.

You'll see additional examples of using the library in the remaining chapters of this book.

Inspector

The inspector opens a world of ability when it comes to changing attributes for selected parts. Parts are added to your widget projects from the library, as you have just learned. In this section, we'll show you how the inspector can be used to change the attributes of some of those parts.

For our example, let's continue with our Hello World project started earlier in this section.

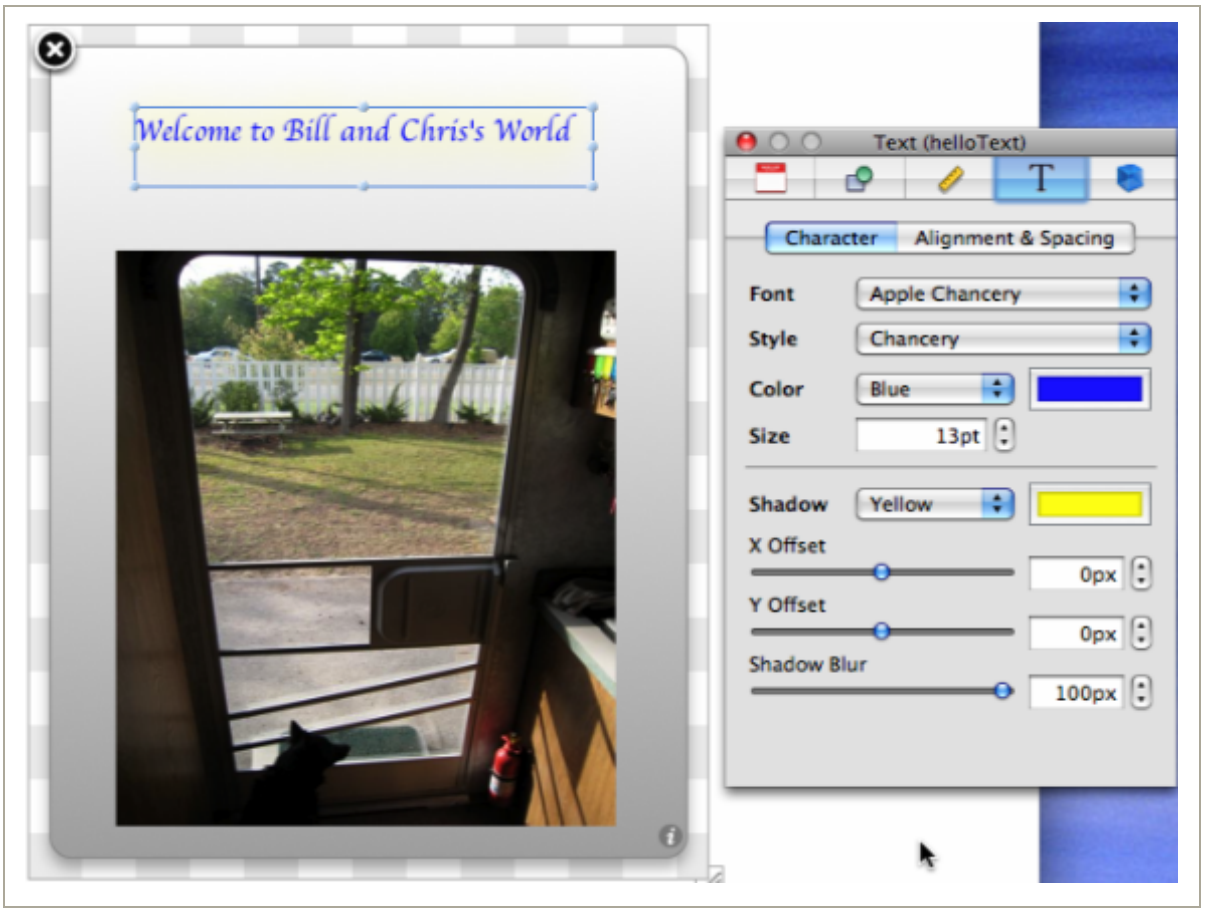
The front face of the Hello World widget project currently contains some descriptive text in a default font with a default size and color. This is a prime candidate for making changes, and we'll use the text box for our example in this section.

First, place the mouse pointer over the text box in the Dashcode design palette. Click to select the text box, and you will see a narrow blue outline around the text box. Then click the Inspector icon at the top of the Dashcode design palette, and a custom attribute window unique to the Text Box part will pop up. Let's change some text attributes.

Font Family, Size, and Color

With the T option open, notice that you can change a variety of font attributes, including family, color, and size. Click and type away. Figure 2-20 shows the changes we decided to go with in our project.

Figure 2-20. Changing the font family, size, and color attributes



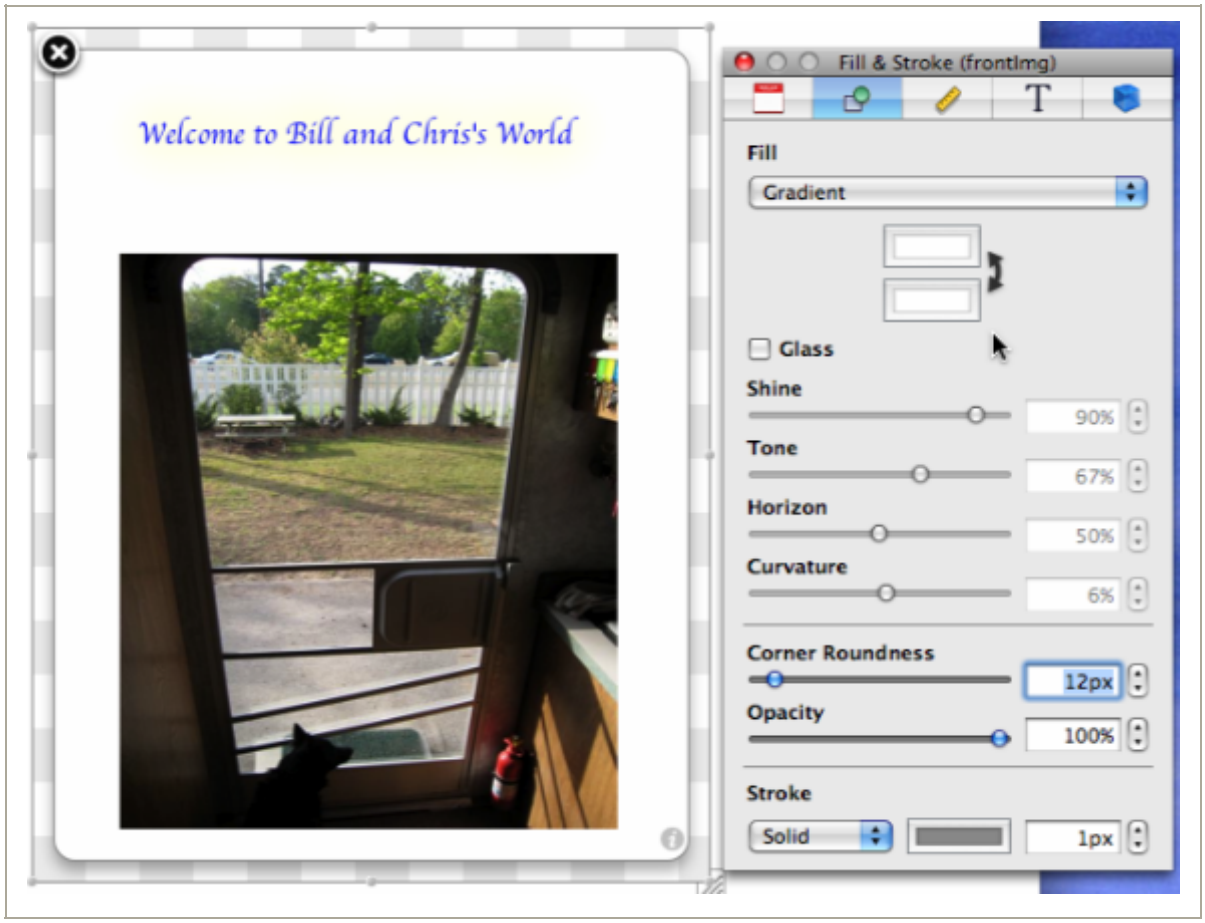
Well, that was an interesting change, but you probably also want to know how to change the background color of your widget. The technique is illustrated in the next section.

Changing the Widget's Background Attributes

Changing the attributes for the whole widget is just as easy as changing the attributes for any added part. Just click within the widget design palette, but not on a specific part. This will allow you to open the attributes for the widget as a whole. Figure 2-21 shows the start of our attempt to change some appearance attributes for the widget itself.

Figure 2-21. The default widget image before modifying attributes

Figure 2-22. Various attributes of the widget as a whole can be changed.

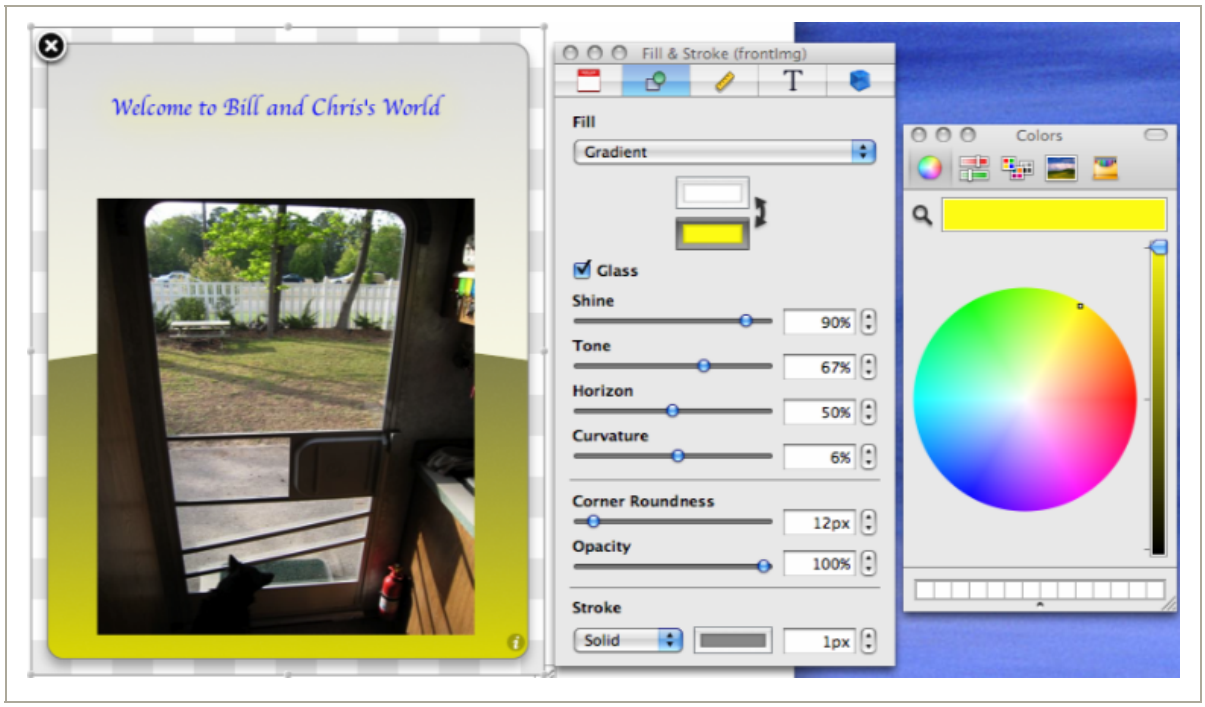


When the inspector is opened for the widget attribute window, you'll see an image similar to Figure 2-22.

Experiment with the Fill and Glass options to see if you can you produce a widget that looks like Figure 2-23.

ac77178b2018603f29f0ba68c167ee35

Figure 2-23. A widget with its main attributes changed



We've come a long way, but we have only just tinkered around with the first Dashcode template. How about the other nine templates? Stay tuned.

One Template Down, Nine To Go

In Chapter 3, we're going to spend some time investigating what the remaining nine widget templates can do for us. We'll create a couple of widgets, experiment with them, change some attributes, and just have fun. Are you ready?

Chapter 3: Widget Templates: Thinking Out of the Box

In Chapter 2, you learned how to manipulate the Custom Template in order to create your first widget. In addition, you learned how to manipulate a variety of attributes, including font families, font size, and font color. You also learned how to manipulate the size of text and pictures within a widget, as well as the size and background of the widget itself. Actually, if you think about it, you gained quite a bit of information.

In this chapter, the focus will be on the other templates provided with Dashcode. We'll discuss each of the remaining nine templates and pick a few of them for special attention.

Now, before starting, we have some homework for you. In order to take full advantage of the templates we are about to discuss, you'll need access to some external information and a couple of RSS feeds. RSS stands for Really Simple Syndication, and it is a mechanism used by a number of widget templates to update information such as weather, news, sports, and so on.

If you don't have a free Google account, now is the time to sign up. Go to `www.google.com` to do so. We're going to use the Google Maps API when we experiment with the Dashcode Maps Template.

Also, remember that you learned in Chapter 2 how to change the attributes of various widget tools, such as font family, size, color, and background color. You also learned how to resize components, including the widget itself, and set general widget attributes. All of that information will come in handy in this chapter, so if you want to review the various steps we took your through, now is the time to do it.

Understanding the various templates provided with the Dashcode development tool can make future widget development much simpler. So now it's time to start thinking about the widget templates you get right "out of the box."

Also, since this chapter and its associated widgets make use of a variety of feeds, you might want to read the "Publication Subscription Programming Guide," an

Apple publication on feeds and feed formats, available at `http://developer.apple.com/documentation/InternetWeb/Conceptual/PubSub/PubSub.pdf`.

The Countdown Timer Template

Imagine what you could do with a countdown timer? Imagine no more. From the Dashcode startup template window, select the Countdown Timer Template and begin your new project. The template is automatically loaded when you select the icon shown in Figure 3-1.

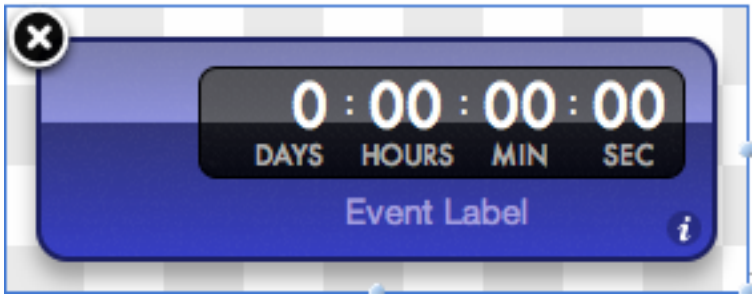
Figure 3-1. The Countdown Timer Template starts a new widget project with a countdown timer already created.

Let's look at Apple's description of the Countdown Timer Template and discover what this template can do:

> *This template creates a widget that displays a numeric countdown to a date and time.*
>
> *To customize your countdown timer widget, add a picture and set the countdown to an upcoming event. You can link your widget to a published iCal calendar, turning your widget into a timer for a series of upcoming events.*

The default countdown timer is shown in Figure 3-2.

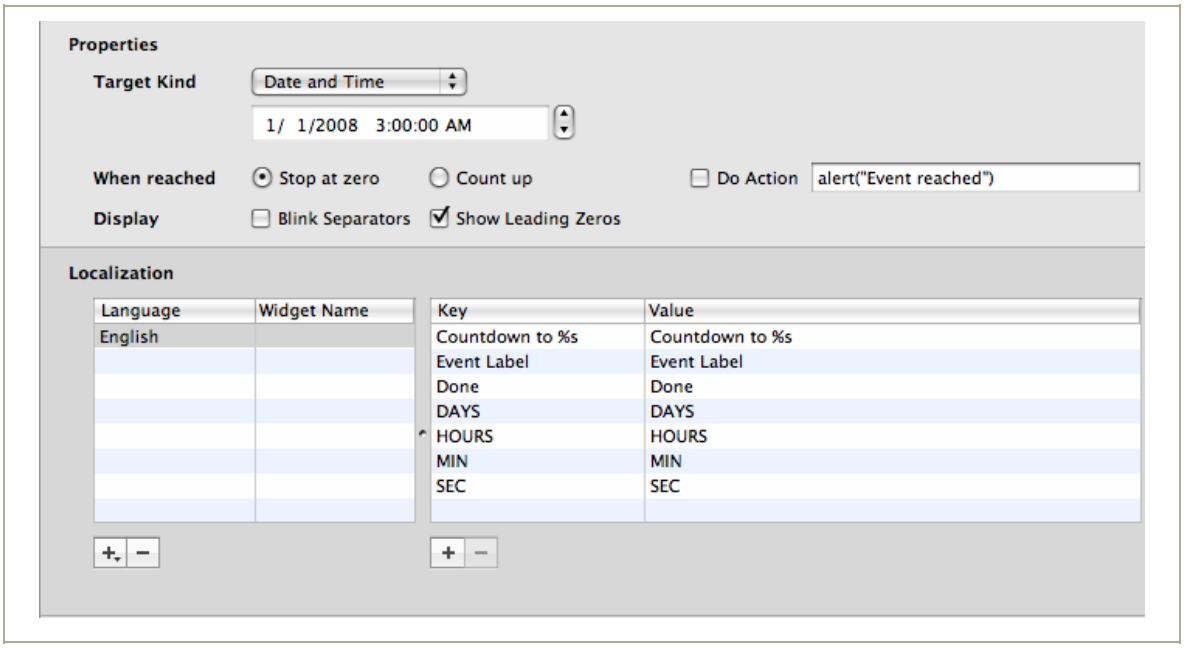
Figure 3-2. The default Countdown Timer Template in Dashcode



Let's experiment with the template by clicking the Run button in the upper-left corner of Dashcode. Wow, the countdown timer is fully functional—but wait, that's not our label for the timer, and we certainly don't know what it is counting down.

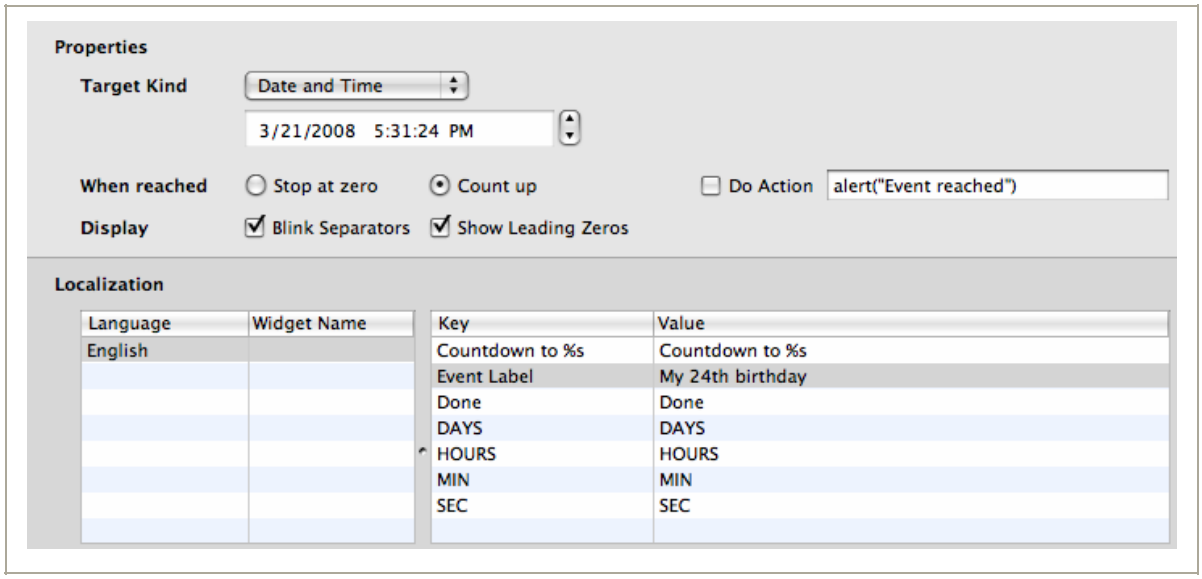
To discover where the countdown timer's default information originated, select Widget Attributes from the left edge of Dashcode. Your screen will appear something like that shown in Figure 3-3.

Figure 3-3. The countdown timer's attributes can be changed on the attribute screen.



Let's change the default information to something more interesting. How about your 24th birthday, which occurs on March 21, 2008, at 5:31:24 p.m.? Make the changes shown in Figure 3-4.

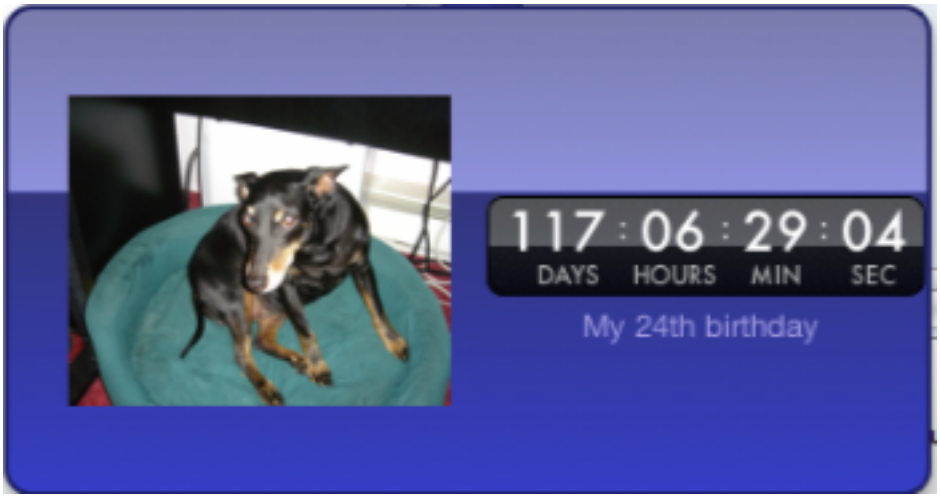
Figure 3-4. Changing the default date and time along with the event label for this timer



Note that we also changed the "When reached" attribute from "Stop at zero" to "Count up." We did this so that when we pass the date, we'll know how much time has elapsed.

Now, flip back to the front face of the timer. In order to complete our countdown timer project, we're going to resize the template and add a picture. Refer to Chapter 2 if you need specific information on how this is done. Figure 3-5 shows our final design in operation.

Figure 3-5. The final version of our countdown timer



In order to create a countdown timer, you will need to open the Countdown Timer Template in Dashcode and set the timer's attributes to the values you desire. Then the widget can be built and installed in the manner discussed in Chapter 2.

"But wait," you say, "Haven't I seen a countdown timer widget somewhere where the information can be entered directly without fooling with Dashcode?" Yes, you probably have seen these countdown timers. There are several great countdown timers that can be downloaded from Apple's site.

Manipulating input data could be built into this countdown timer, too. However, at this point in the book, we're trying to avoid all references to additional programming code. Maybe you'll want to return to this project after learning about some simple JavaScript programming in Chapters 4 and 5.

The Maps Template

The Maps Template allows you to add maps to your widgets. With map information available in your widget projects, you can give map directions to your house, find directions to a new store, and so on.

From the Dashcode startup template window, select the Maps Template and begin your new project. The template is automatically loaded when you select the icon shown in Figure 3-6.

Figure 3-6. The Maps Template is started when you select this icon.

Here is what Apple states about the Maps Template:

> *This template creates a widget that displays locations and location-specific information on a map.*
>
> *To customize your Maps widget, change the map URL to one that you publish. You can customize how your widget displays location information and the appearance of its front side.*

The default Maps Template, shown in Figure 3-7, lacks a lot of detail when compared to the Countdown Timer Template.

Figure 3-7. The default Maps Template looks kind of barren.

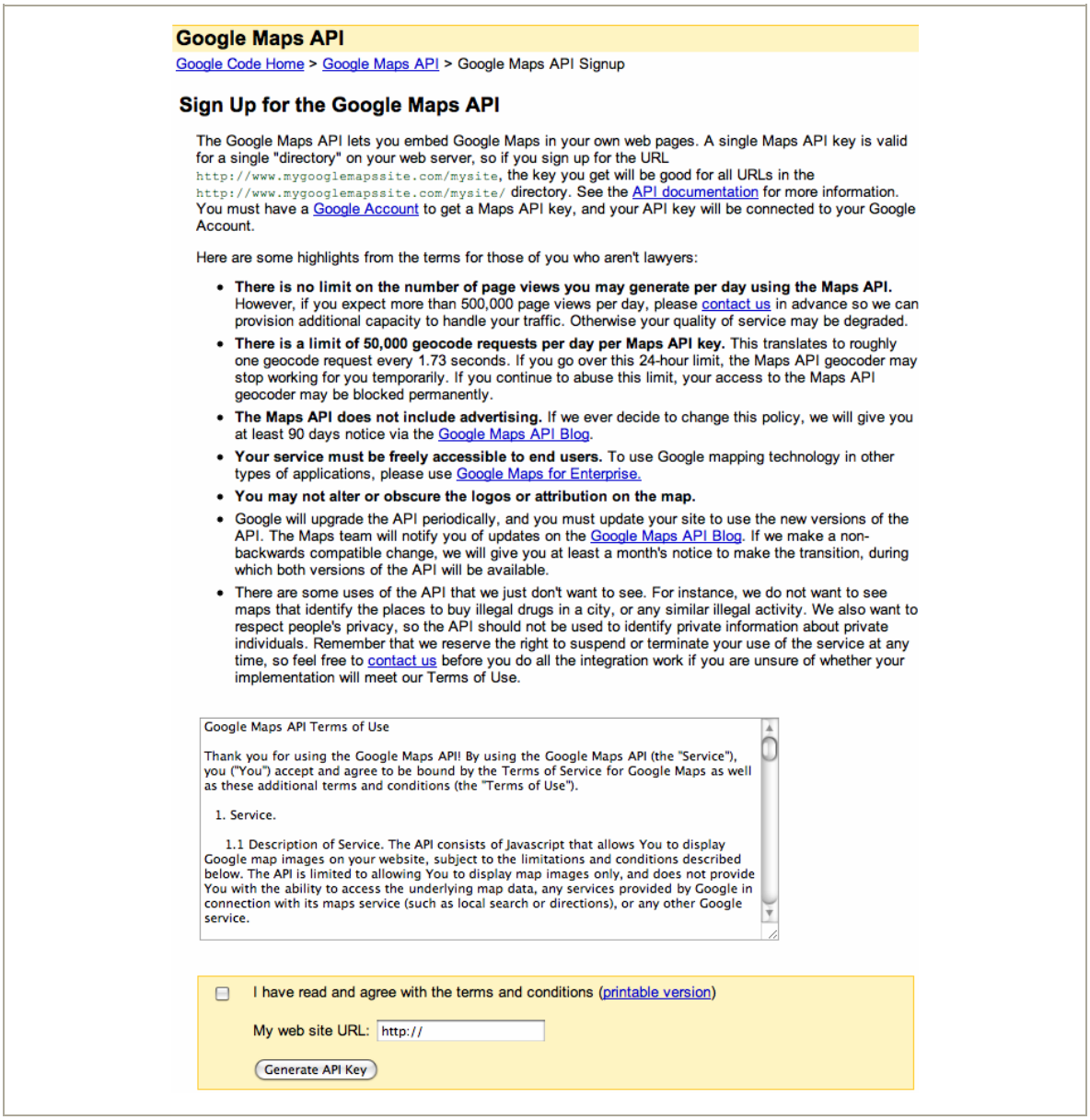
In order to make use of the Maps Template for the new widget project, you'll have to find a map source. Fortunately, there are free sources available to you. Here is the quick way to get started.

With the default Maps Template in place, click the Run button in the upper-left corner of Dashcode to start and test the template. You should now see a screen somewhat like that shown in Figure 3-8.

Figure 3-8. Running and testing the default Maps Template

Now click the Get a Key button. When you do, you will be taken to the Google Maps API page, as shown in Figure 3-9.

Figure 3-9. To access the Google Maps API, enter your account information or create a new account.



Google Maps API

Google Code Home > Google Maps API > Google Maps API Signup

Sign Up for the Google Maps API

The Google Maps API lets you embed Google Maps in your own web pages. A single Maps API key is valid for a single "directory" on your web server, so if you sign up for the URL `http://www.mygooglemapssite.com/mysite`, the key you get will be good for all URLs in the `http://www.mygooglemapssite.com/mysite/` directory. See the API documentation for more information. You must have a Google Account to get a Maps API key, and your API key will be connected to your Google Account.

Here are some highlights from the terms for those of you who aren't lawyers:

- **There is no limit on the number of page views you may generate per day using the Maps API.** However, if you expect more than 500,000 page views per day, please contact us in advance so we can provision additional capacity to handle your traffic. Otherwise your quality of service may be degraded.
- **There is a limit of 50,000 geocode requests per day per Maps API key.** This translates to roughly one geocode request every 1.73 seconds. If you go over this 24-hour limit, the Maps API geocoder may stop working for you temporarily. If you continue to abuse this limit, your access to the Maps API geocoder may be blocked permanently.
- **The Maps API does not include advertising.** If we ever decide to change this policy, we will give you at least 90 days notice via the Google Maps API Blog.
- **Your service must be freely accessible to end users.** To use Google mapping technology in other types of applications, please use Google Maps for Enterprise.
- **You may not alter or obscure the logos or attribution on the map.**
- Google will upgrade the API periodically, and you must update your site to use the new versions of the API. The Maps team will notify you of updates on the Google Maps API Blog. If we make a non-backwards compatible change, we will give you at least a month's notice to make the transition, during which both versions of the API will be available.
- There are some uses of the API that we just don't want to see. For instance, we do not want to see maps that identify the places to buy illegal drugs in a city, or any similar illegal activity. We also want to respect people's privacy, so the API should not be used to identify private information about private individuals. Remember that we reserve the right to suspend or terminate your use of the service at any time, so feel free to contact us before you do all the integration work if you are unsure of whether your implementation will meet our Terms of Use.

Google Maps API Terms of Use

Thank you for using the Google Maps API! By using the Google Maps API (the "Service"), you ("You") accept and agree to be bound by the Terms of Service for Google Maps as well as these additional terms and conditions (the "Terms of Use").

1. Service.

1.1 Description of Service. The API consists of Javascript that allows You to display Google map images on your website, subject to the limitations and conditions described below. The API is limited to allowing You to display map images only, and does not provide You with the ability to access the underlying map data, any services provided by Google in connection with its maps service (such as local search or directions), or any other Google service.

I have read and agree with the terms and conditions (printable version)

My web site URL: http://

Generate API Key

Be sure to check the check box at the bottom of this screen after entering your web site's URL. If you don't have a web site, you can open a trial account at .Mac so that you can experiment with these features.

Once you have entered the necessary information, click the Generate API Key button to retrieve your map access key information. That information can now be entered on the Widget Attributes page of the Maps Template. Alternatively, you can enter the information shown in Figure 3-10.

Figure 3-10. Generalized account information that will allow you to see a map of the United States

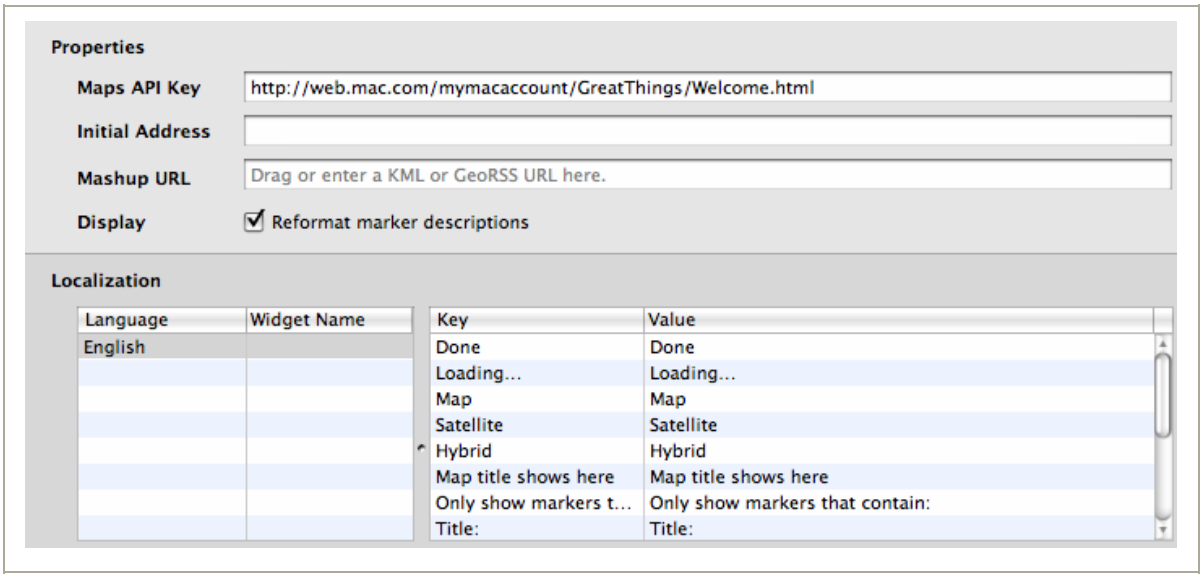
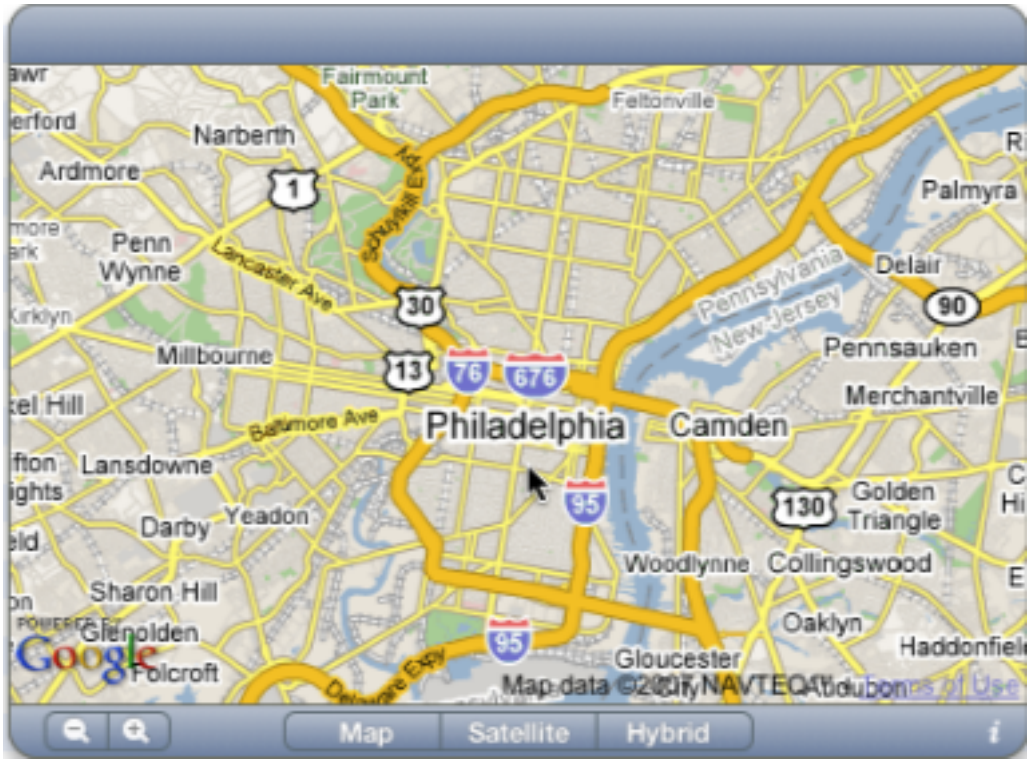


Figure 3-11 shows the Maps widget with a "default" information map. By clicking and zooming in on this map, you'll be able to see some pretty interesting images.

Figure 3-11. A view of the default information map of the United States

Of course, if you obtain the API key from Google, you can actually enter a street, city, and state where you want the map to initialize when the widget is started. Figure 3-12 shows some streets in Philadelphia, Pennsylvania.

Figure 3-12. Zooming in to view some Philadelphia streets



At this point, you may wish to resize your widget so that you can view a larger portion of your map's screen.

The RSS and the Daily Feed Templates

As mentioned at the beginning of the chapter, RSS stands for Really Simple Syndication. By using RSS feeds from the Internet, you can access a whole range of current information topics, including weather, news, sports, and pictures.

The RSS and Daily Feed templates both have built-in code templates that will allow you to quickly identify and use any type of RSS feed you desire.

The RSS Template will be discussed here, but you'll find the Daily Feed Template very similar in function. The RSS Template is automatically loaded when you select the icon shown in Figure 3-13.

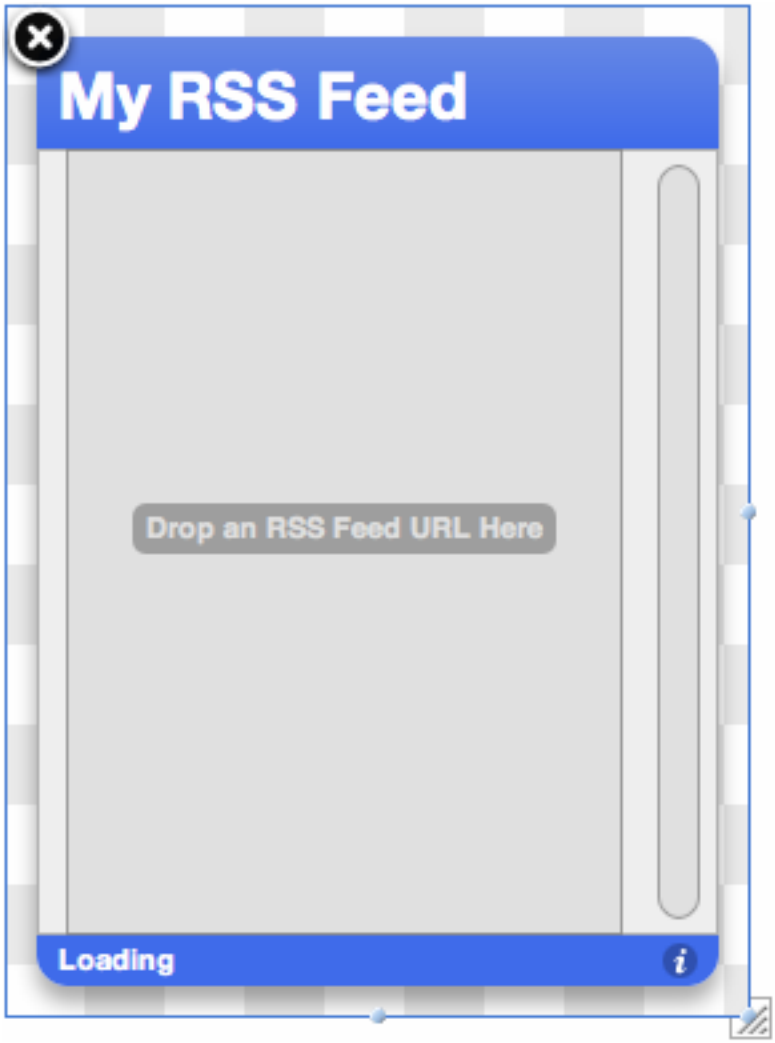
Figure 3-13. Select the RSS Template icon to load the default template into Dashcode.

What can you do with this template code? Here is what Apple has to say on the topic:

> *This template creates a widget that displays headlines and articles from an RSS source.*
>
> *To customize your RSS widget, drag the URL for your RSS feed from Safari to the canvas. You can customize how many articles are shown and how long they are displayed.*

Figure 3-14 shows the default RSS Template open in Dashcode.

Figure 3-14. The default RSS Template loaded into Dashcode

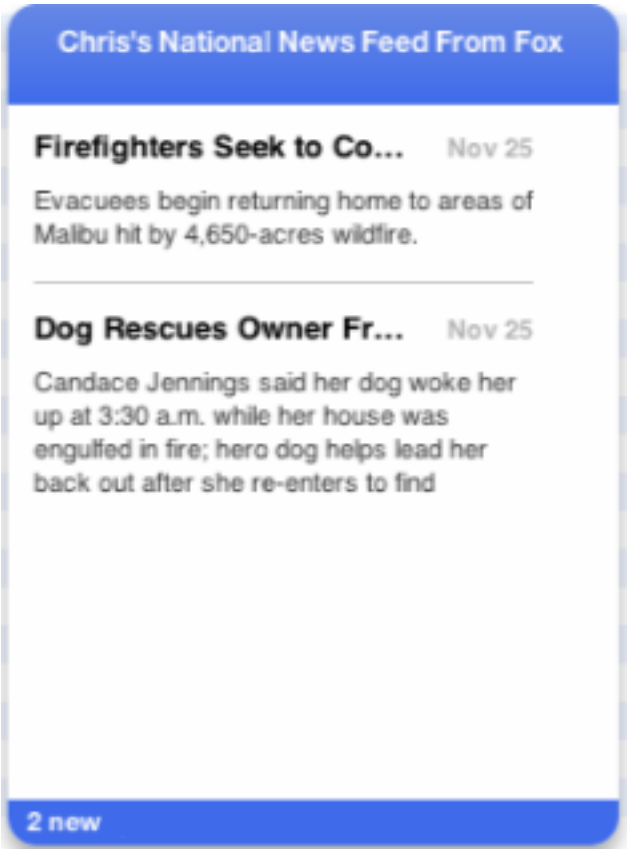


All you need to do at this point is find a source of weather, news, or sports that will provide the RSS feed. This is easier than you might expect. Go into Safari and type “free news feeds” in your search engine. Somewhere in the returned list, you’ll find a source of FOX news feeds. Here are some examples:

- *Latest headlines (all sections)*: `http://feeds.foxnews.com/foxnews/latest`
- *National:* `http://feeds.foxnews.com/foxnews/national`
- *World:* `http://feeds.foxnews.com/foxnews/world`
- *Politics:* `http://feeds.foxnews.com/foxnews/politics`
- *Business:* `http://feeds.foxnews.com/foxnews/business`
- *SciTech:* `http://feeds.foxnews.com/foxnews/scitech`
- *Health:* `http://feeds.foxnews.com/foxnews/health`
- *Entertainment:* `http://feeds.foxnews.com/foxnews/entertainment`
- *Views:* `http://feeds.foxnews.com/foxnews/views`
- *FOX Blogs*: `http://feeds.foxnews.com/foxnews/foxblogs`

All you have to do is grab the feed you want and drag it to the default RSS Template. Our example takes on the appearance of Figure 3-15 while still in Dashcode.

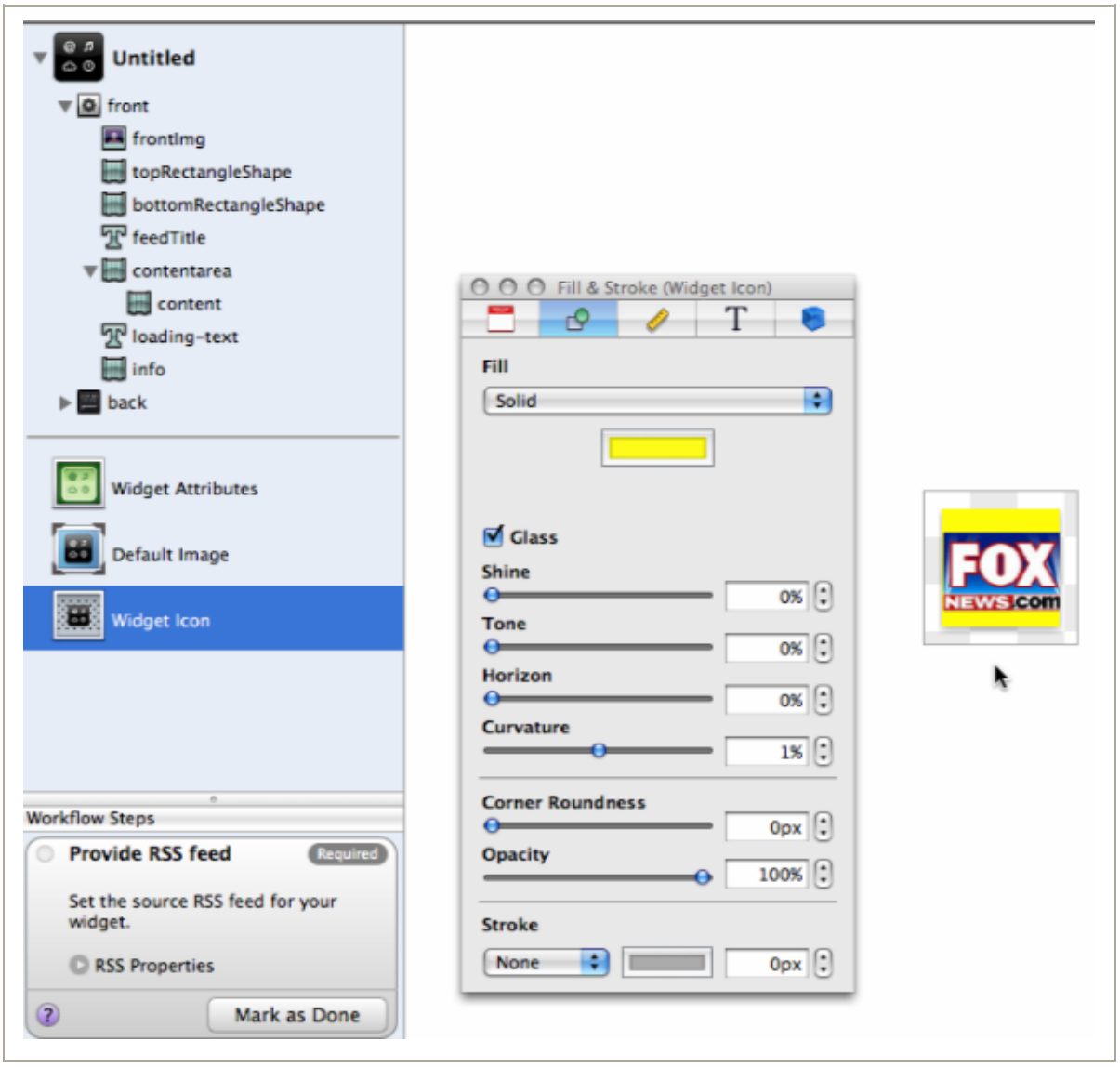
Figure 3-15. The default RSS Template loaded with a FOX RSS news feed



Before testing our new widget, let's change a couple of the widget attributes. First, let's change the title of the widget and add a snappy icon.

In order to change the default icon, select Widget Icon from the left side of Dashcode. Then, using the Inspector button at the top right of Dashcode, change the background color to yellow. Next, go onto the Internet and find some graphics of your choice. In the end, you might end up with something similar to Figure 3-16.

Figure 3-16. The default widget icon is changed to something a little snappier.



OK, you should have the idea by now. If you don't watch FOX News, perhaps a CNN feed will suit your needs. Or why not try to find some sports or health news feeds. We know you are saying, "Hey, this is fun!" Not only is this fun, but remember, you haven't had to write a single line of programming code yet!

The Podcast Template

Are you into podcasts? Who isn't these days? Well, the Podcast Template might just be the widget template for you. With the Podcast Template, you can create a personalized widget that will play any episode of your favorite podcasts. To create a podcast widget, select the Podcast Template icon and begin your new project. The template is automatically loaded when you select the icon shown in Figure 3-17.

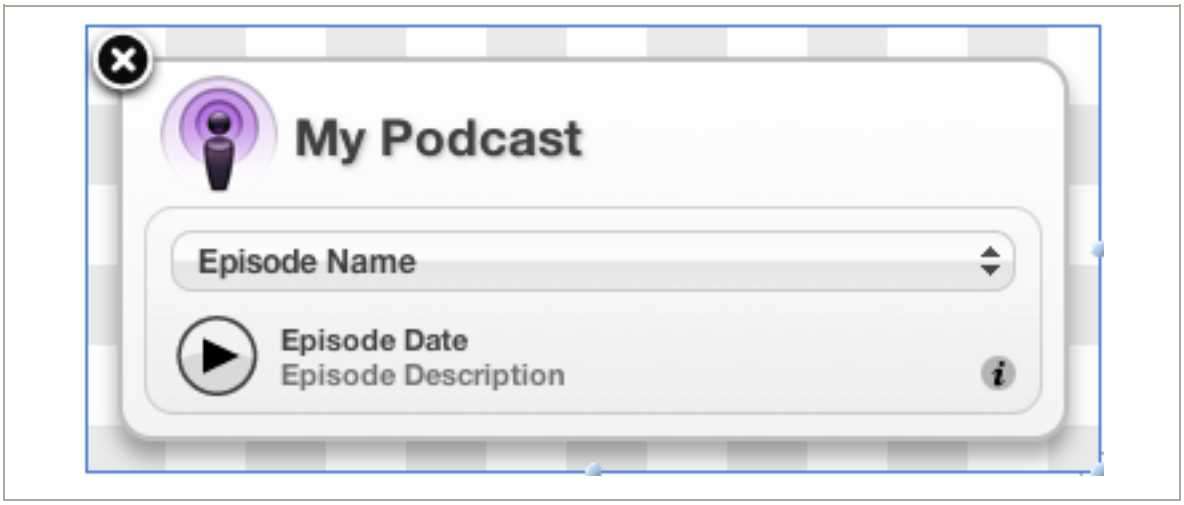
Figure 3-17. The Podcast Template will allow you to build a customized podcast-playing widget.

Apple describes the features of this unique template as follows:

> *This template creates a widget that displays and plays a podcast's episodes.*
>
> *To customize your Podcast widget, change the Podcast URL to one that you publish. You can customize how often your widget checks for updates and the appearance of its front side.*

Let's take a look at the default Podcast Template, as shown in Figure 3-18.

Figure 3-18. The default Podcast Template just begging for some customization



Before we get to the good stuff of adding a podcast source, let's customize the template just a bit. Use the Library and Inspector buttons in the upper-right corner of Dashcode to create a unique widget similar to Figure 3-19.

Figure 3-19. A few simple modifications allow you to create a customized podcast widget.

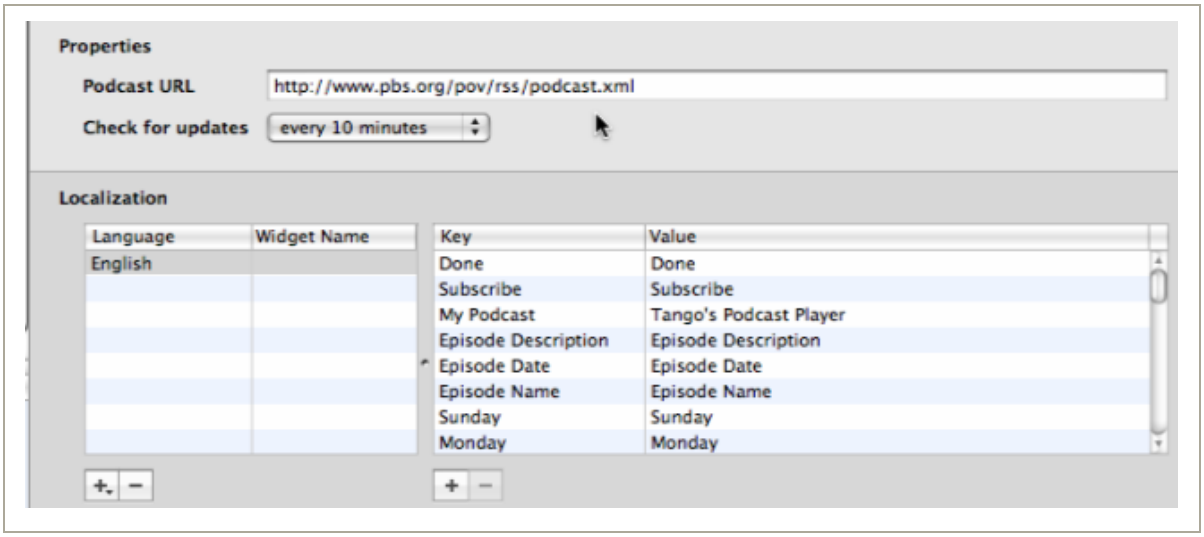
As you can see from the figure, we added a custom title and a photograph of Tango, my dog. We also modified the widget icon (using the Widget Icon button, on the left edge of Dashcode) by adding a picture of Tango to it.

Now, a podcast player is only as good as the source of the podcasts. To find something that might interest you, use the search engine in Safari and type "podcast url," or go to your favorite site and see if it provides podcast episodes.

To test our podcast player, we'll use the PBS podcast site, at `www.pbs.org/pov/rss/podcast.xml`.

You can enter the Podcast URL information on the Widget Attributes screen, as shown in Figure 3-20.

Figure 3-20. Entering the podcast URL using the Widget Attributes screen



Now you can test the widget by clicking the Run button in the upper-left corner of Dashcode.

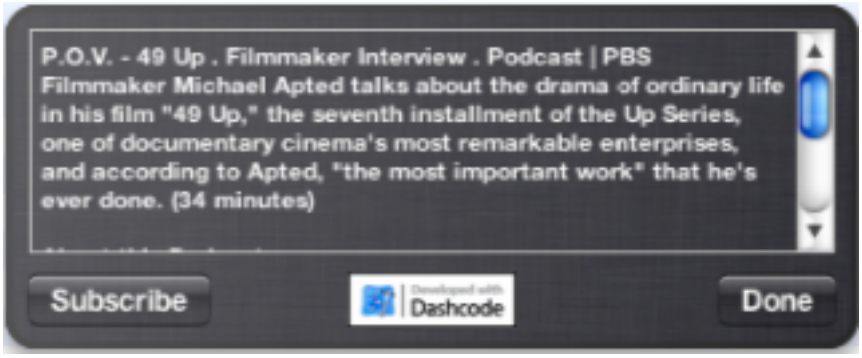
Figure 3-21 shows our customized widget with a long list of available podcasts in the Episode Name drop-down list.

Figure 3-21. The drop-down list shows a long list of available podcasts.

From the list shown in Figure 3-21, just select any podcast you desire and sit back and listen.

Now, the back face of this widget is worthy of a discussion because it contains additional features. Take a look at Figure 3-22.

Figure 3-22. The back face of the podcast widget includes additional information.



As you can see from our example, the back face provides additional episode and podcast information.

You will also notice a Subscribe button. If you click this button, the widget will automatically subscribe you to this podcast in your iTunes library.

The Photocast Template

Got photos? If you are a prolific camera buff or just love taking personal pictures, the Photocast Template will allow you view photos that you photocast through your web site or from other photocasters.

From the Dashcode startup template window, select the Photocast Template and begin creating your new widget. The default Photocast Template is automatically loaded when you select the icon shown in Figure 3-23.

Figure 3-23. The Photocast Template icon starts a new widget project that provides photocasting options.

Apple's initial description of this template reads as follows:

> *This template creates a widget that displays a slideshow of photos from an iPhoto Photocast.*
>
> *To customize your Photocast widget, change the Photocast URL to one that you publish. You can customize how often the picture changes and which transition is used between pictures.*

Before we actually investigate the features of the Photocast Template, let's talk about the source of the photos.

Earlier, we suggested that you open a free (i.e., free for 60 days) .Mac account. The .Mac account will allow you to upload a number of things, including photos that you can share with friends. You will also be able to create your own web site using Apple's iWeb tool, provided on your OS X disk.

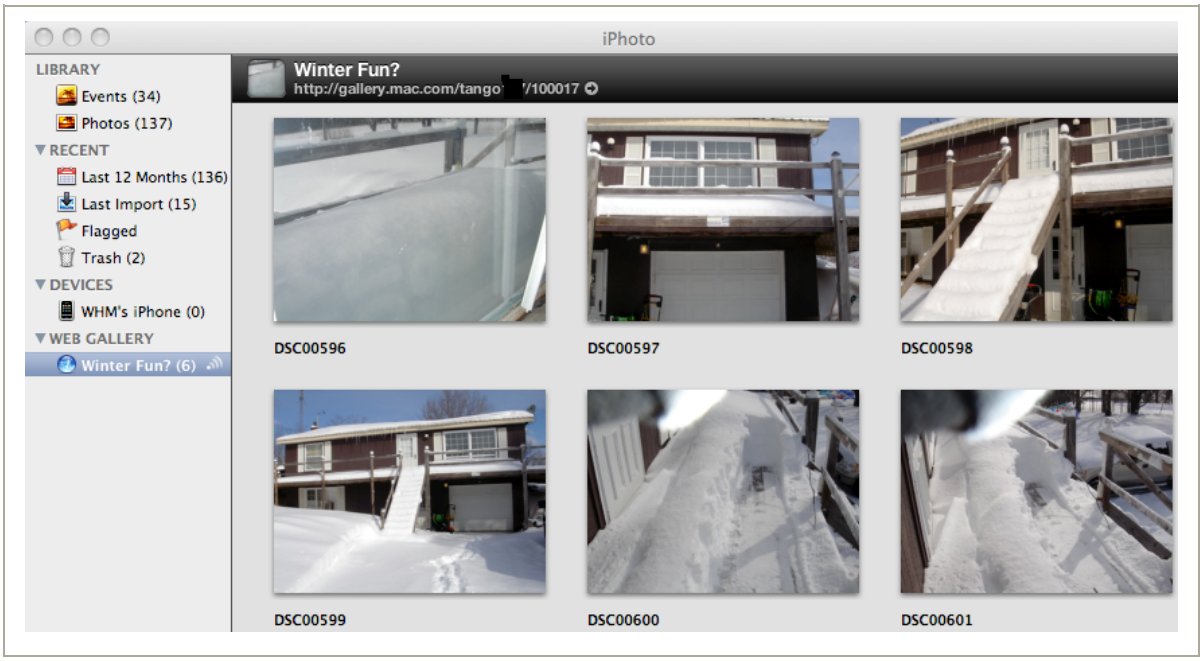
Having said this, let's go over the necessary steps to load some photos from iPhoto up to your .Mac account so they can be viewed by friends and family using the Photocast Widget you are about to create. Here are the steps:

1. Open iPhoto and select a group of photographs that you wish to share.
2. Once you've selected them, click the + button in the lower-left corner of iPhoto and name the new gallery.
3. From the Share menu item of iPhoto, select the Web Gallery menu item.

4. From the dialog box of options, select those you wish to apply to your published gallery.
5. Click the Publish button, and your shared gallery will appear under WEB GALLERY in the left column of iPhoto.

We created a web gallery named "Winter Fun?" and published it to our account. Figure 3-24 shows a portion of our iPhoto screen.

Figure 3-24. We created a Winter Fun? gallery of pictures in iPhoto and then published it to our .Mac account for sharing with others.



Examine Figure 3-24 and notice that there is a URL associated with the gallery just under the gallery's name in the upper left of the screen. The one you see in the screenshot isn't a real URL—just one we created for the screenshot.

The Photocast Template and .Mac Web Gallery

To set a .Mac web gallery as the source of a Photocast widget, do the following:

1. On the .Mac Web Gallery web page, click Subscribe.
2. Paste the subscription URL into your browser window and press Return.
3. Once the feed loads, copy and paste the new feed URL into the photocast source URL on the Widget Attributes screen.

With your new photo gallery published to your .Mac account, you are now ready to create the photocasting widget for reading and viewing the gallery. Figure 3-25 shows our new photocast widget with a photo from our gallery.

Figure 3-25. Here is a look at our Winter Fun? gallery of pictures.

You can use your own URL for friends and family or subscribe to a photocast service like Flickr.

The Quartz Composer Template

Take a deep breath. If you have not worked with or are not familiar with Quartz Composer, this section might just scare you out of the chapter. However, don't be frightened—even though a discussion of Quartz Composer is beyond the scope of this book, we can still use some previously created Quartz Composer compositions and have a little fun.

Before diving into Quartz Composer, let's see how to start our next widget project with Apple's Quartz Composer Template for Dashcode.

The template is automatically loaded when you select the icon shown in Figure 3-26.

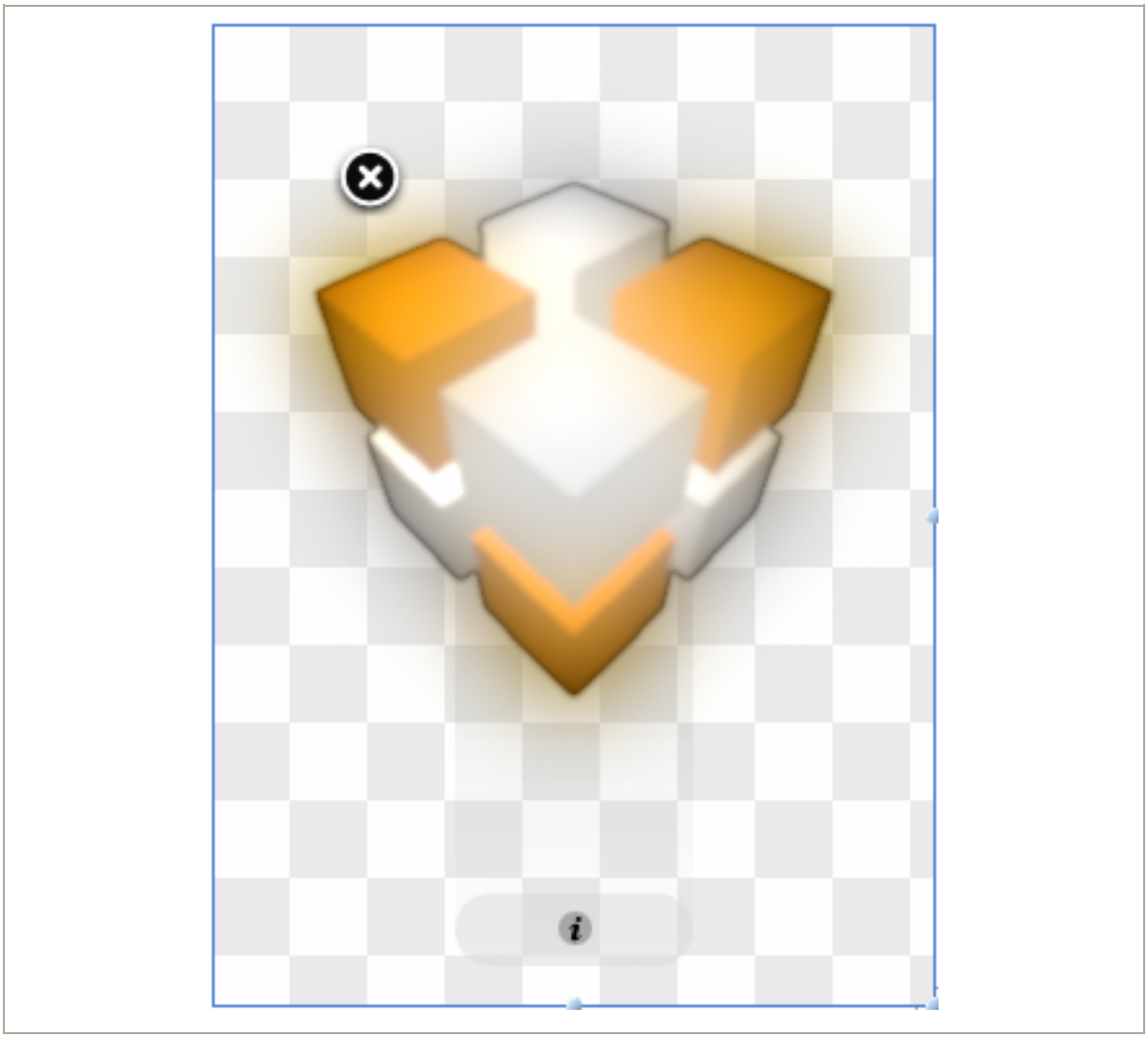
Figure 3-26. The Quartz Composer Template starts a new widget project that will allow you to experiment with Quartz Composer compositions.

Quartz Composer compositions are created by Apple's visual programming language provided with the Xcode development tools. These tools are not for the faint of heart, but their results can produce exciting 3D images that can be rotated in three dimensions on your screen.

Our default template supplies an image for our widget that will do just this. Apple describes this template's capabilities in this manner:

> *This template creates a widget that displays a Quartz Composer composition.*
>
> *To customize your Quartz Composer widget, edit the included composition or provide your own. You can use this template to process and render graphical data.*

Figure 3-27. The Quartz Composer Template with the default composition



In Figure 3-27, you'll see the default Quartz Composer composition.

Your immediate reaction might be, “That looks a little boring.” But wait, it’s more than a 2D image.

Go ahead and test the template by clicking the Run button in the upper-left corner of Dashcode.

You will see the composition on the screen with a clear background behind it. Click the composition, and while holding the left mouse button down, move the mouse a bit. Wow—the image rotates with the mouse movements.

OK, that was fun, but after a couple of minutes you are going to get bored. So, here are two things you can do to grab a few more compositions to experiment with:

- Go onto the Internet and search for Quartz Composer compositions. One source of compositions can be found at `www.quartzcompositions.com/phpBB2`.
- Install the Xcode Quartz Composer software from your OS X disk, and search your computer for files ending in `.qtz`.

For experimentation purposes, we chose the latter and found a little teapot composition named Gooch.qtz. We moved a copy of that composition to our desktop. Next, while still in Dashcode, we used the mouse to drag the teapot image over the top of the default image shown in Figure 3-27. When you do this, nothing will change in Dashcode. However, when you test the widget by clicking the Run button in Dashcode, you will now see the teapot image, as shown in Figure 3-28.

Figure 3-28. The new teapot composition added to our widget

Figure 3-29 shows another image of the teapot after rotation in our widget.

Figure 3-29. The teapot composition after rotation in our new widget

At first glance, you might feel that this is something you'll never really experiment with. However, if you snoop around long enough, you will find that a number of interesting screen savers are created with Quartz Composer compositions. This just might be something worth looking into!

The Video Podcast Template

If you worked through the Podcast Template earlier in the chapter, you'll find that the Video Podcast Template is similar. This template is specifically geared to video podcasts. From the Dashcode startup template window, select the Video Podcast Template and begin your new project. The template is automatically loaded when you select the icon shown in Figure 3-30.

Figure 3-30. The Video Podcast icon starts a new widget project with a Video Podcast Template already installed.

Here is what Apple states about the Video Podcast Template:

> *This template creates a widget that displays and plays a video podcast's episodes.*
>
> *To customize your Video Podcast widget, change the podcast URL to one that you publish. You can customize how often your widget checks for updates and the appearance of its front side.*

The default Video Podcast Template is shown in Figure 3-31.

The key to using this template is, of course, knowing sources of video podcasts. Again, use the Internet to seek out video podcasts that interest you.

One example of video podcasts can be found at the following URL: `feed://www.sesameworkshop.org/podcasts/sesamestreet/rss.xml`.

This free video podcast features characters and sketches from Sesame Street.

Figure 3-31. The default Video Podcast Template

Figure 3-32 shows the Video Podcast Template with a number of widget attributes modified and the preceding URL loaded.

Figure 3-32. The modified Video Podcast Template with video URL

If Sesame Street is not one of your favorite feeds, simply change the URL to something more to your liking.

The Gauge Template

In this section, we're going to discuss a unique and truly useful template: the Gauge Template. The Gauge Template can be loaded into Dashcode by selecting the icon shown in Figure 3-33.

Figure 3-33. The Gauge Template icon starts a new widget project with a default gauge already installed.

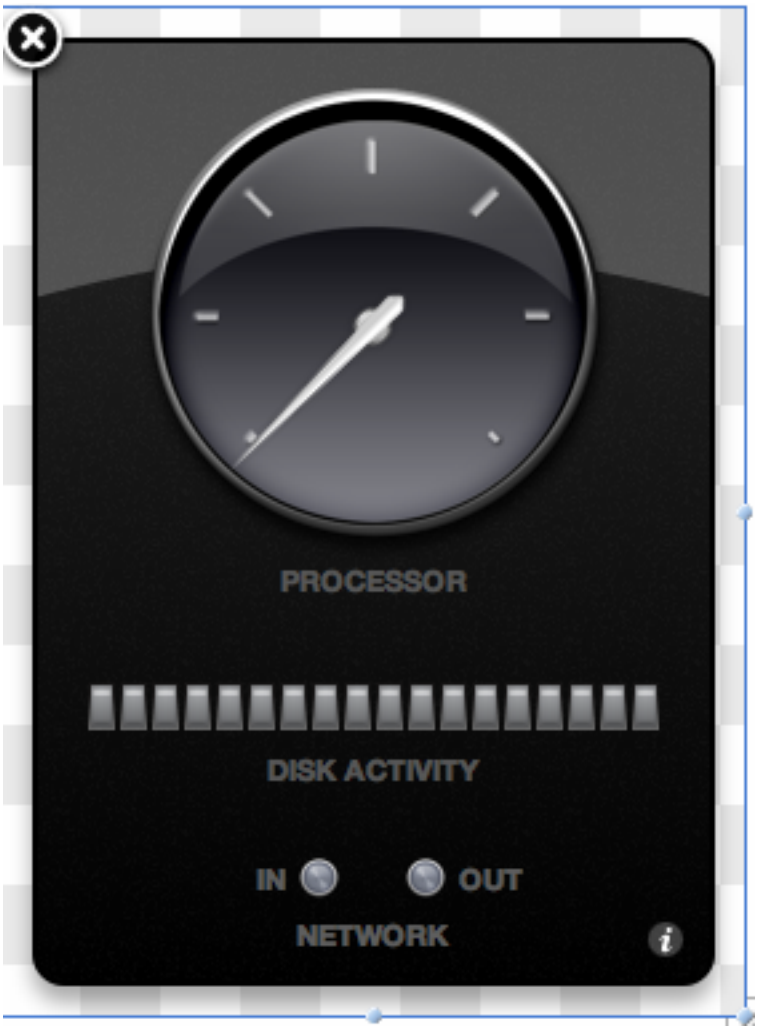
On creating this unique widget template, Apple states the following:

> *This template creates a widget that monitors activities.*
>
> *To customize your gauge widget, change the data sources for the gauges and indicators. You can use this template to monitor any activity that regularly returns numerical values.*

If you have come over to the Apple side from the PC market, perhaps there are a few things that you miss from your old PC. It used to drive us crazy trying to figure out where files were being stored. Well, we got a handle on that issue when we became more intuitive in our thinking. But the one thing we really missed was knowing when the hard drive was spinning! You know the problem—you want to save something to a particular hard drive. You send it. Did the hard drive light come on? If it did, the save operation was probably successful. But there are no hard drive lights on a Mac—well, the default Gauge Template is going to change all of that!

Figure 3-34 shows the default Gauge Template with processor, disk, and network monitors preinstalled.

Figure 3-34. The Gauge Template comes with a processor, disk, and network monitor built in.



Before doing a little customization on this template, test the widget by clicking the Run button that appears in Dashcode's upper-left corner (see Figure 3-35).

Figure 3-35. The gauge widget during a sample test

The default gauge widget is neat, but all we wanted was a widget to monitor our disk activity. Well, that is easy to fix. Just go back to the template and remove the components you no longer desire to be on your finished widget. Then resize, change the color, and change any other attributes you desire. Figure 3-36 shows our final creation.

Figure 3-36. The modified gauge widget now becomes just a disk monitor.

Out of the Box—Not!

If you have been working through the discussion of the various Dashcode templates, we're sure you are thrilled with what you can now accomplish with so little work—news feeds, picture and video viewers, and even monitoring gauges.

Perhaps you have already built lots and lots of widgets just by modifying the attributes of the templates we have discussed in this chapter. But something seems to be missing.

There are only so many things you can do with a widget designed by someone else, and that is exactly what each of the templates represents. They are someone else's idea of what you might want in a functional widget. However, to truly take any of the widget templates to a new and higher level of functionality, you're going to have to write some code. What kind of code? JavaScript.

In the final two chapters of this book, you'll learn how to incorporate elements of JavaScript into a variety of templates to create new widgets that are unique to your needs. We don't intend to make you a JavaScript programmer, but we'll work with enough code for you to accomplish your needs. Ready to get started?

Creating Mac Widgets with Dashcode

© 2008 by William H. Murray and Chris H. Pappas

All rights reserved. No part of this work may be reproduced or transmitted in any form or by any means, electronic or mechanical, including photocopying, recording, or by any information storage or retrieval system, without the prior written permission of the copyright owner and the publisher.

ISBN-13 (electronic): 978-1-4302-0968-3

ISBN-13 (paperback): 978-1-4302-0967-6

Trademarked names may appear in this book. Rather than use a trademark symbol with every occurrence of a trademarked name, we use the names only in an editorial fashion and to the benefit of the trademark owner, with no intention of infringement of the trademark.

Distributed to the book trade in the United States by Springer-Verlag New York, Inc., 233 Spring Street, 6th Floor, New York, NY 10013, and outside the United States by Springer-Verlag GmbH & Co. KG, Tiergartenstr. 17, 69112 Heidelberg, Germany.

In the United States: phone 1-800-SPRINGER, fax 201-348-4505, e-mail orders@springer-ny.com, or visit http://www.springer-ny.com. Outside the United States: fax +49 6221 345229, e-mail orders@springer.de, or visit http://www.springer.de.

For information on translations, please contact Apress directly at 2855 Telegraph Ave, Suite 600, Berkeley, CA 94705. Phone 510-549-5930, fax 510-549-5939, e-mail info@apress.com, or visit http://www.apress.com.

The information in this book is distributed on an “as is” basis, without warranty. Although every precaution has been taken in the preparation of this work, neither the author(s) nor Apress shall have any liability to any person or entity with respect to any loss or damage caused or alleged to be caused directly or indirectly by the information contained in this work.

www.allitebooks.com