

# Tiger, Dashboard, and Widgets

If you have had a chance to play with Dashboard widgets, you know how useful they are. Although widgets cover applications when they're active, most of the time this won't negatively affect your work. For example, if you are checking the temperature, you really don't need to leave the thermometer open after you know how hot or cold it is. Widgets also support copy and paste, so if you look up information on the Web, you can close Dashboard and paste it into your report. Even though they can't be used simultaneously, applications and widgets work well together.

In this chapter, you learn about Dashboard in Tiger. You also find out everything you need to know about working with widgets. You find out about security issues with widgets and how Apple has modified widget installation to make them more secure.

By the end of this chapter, you will know:

- □ How to manage widgets
- □ How to install widgets
- How to reload widgets

## **OS X Tiger**

Tiger is the fourth big cat in the OS X parade. If you can't immediately remember the name before Panther and Jaguar, you shouldn't debit your Macintosh geek points. The 10.0 and 10.1—yes, there are actually five—releases had codenames, but they must not have passed the coolness test because they weren't used widely outside of Apple. As with those previous releases of OS X, Tiger adds a number of features and tweaks to an already loaded operating system. Apple's technical briefs describe over two hundred features, and the regular listing of feature tips on its website suggests as many.

As with those previous systems, Tiger has a few new features that stand out from all the others and three features have improved usability in common. Those three are Spotlight, Automator, and Dashboard. Spotlight makes it easier to find information and files you've lost on your hard drive. Spotlight creates and regularly updates an index of the files on your Macintosh, so searching in Spotlight looks at the content of the files as well as the names of the files. Automator makes it easier to create workflows that make using your Macintosh more . . . automatic. For example, you can automate the process of selecting a directory, opening each of the files in BBEdit, and changing the tabs in the files to the appropriate number of spaces. Dashboard brings the information of the Web, as well as information from your Macintosh and your network, to your fingertips. Another thing these three features have in common is that they can be extended. You can create Automator actions, Spotlight plugins, and Dashboard widgets and share them with others. But unlike the other two, widgets feel like small applications.

### **Dashboard and Widgets**

Dashboard is an overlay that zooms into place Exposé-like, floating the widgets above your applications, whenever you press a keyboard shortcut or click the Dashboard icon in the Dock. Dashboard does not affect any of your open applications, and its transparency allows you to see your applications below.

Widgets run inside of Dashboard and provide you with immediate access to information that you might otherwise have to load an additional application or two to get. Just as Dashboard's gauge icon implies, widgets give you an instant way to check the health of your system or track information you need. Widgets may be small web pages, but Dashboard is faster than waiting on your browser to load.

F12 is the key assigned to open Dashboard by default, but this assignment can be changed in the Keyboard and Mouse panel in System Preferences. To close Dashboard, press F12 again or click in the Dashboard outside of a widget.

On older PowerBooks, some of the function keys do double duty, and F12 is one of them. Pressing it ejects the CD or DVD in the PowerBook's optical drive (F1 through F6 also have predefined, hardware-related functions). You can use F12 to invoke Dashboard on these PowerBooks by holding down the function (fn) key when you press F12.

In Figure 1-1, you'll notice that you can see the Finder's desktop beneath Dashboard's widgets. Tiger ships with a suite of 14 Dashboard widgets, and you can see 5 of them in the screenshot: Calendar, World Clock, Calculator, Weather, and the Widget widget, in counter-clockwise order. Notice that the cursor is pointing to an *i* in the lower-right corner of the Weather widget. Any widget with preferences displays an *i* when you are over it. Clicking the *i* flips the widget over so you can change its preferences (Figure 1-2).

Clicking the plus sign (+) button in the lower-left portion of the screen in Figure 1-1 displays the Widget Bar at the bottom of the window. You can also display the Widget Bar by pressing Command-+ while Dashboard is open (Figure 1-3).

The Widget Bar, which has the metal grill look of a G5, runs along the bottom of Dashboard. In the Widget Bar you can see all of the widgets that are installed on your Macintosh in the root-level /Library/Widgets/ folder and your users-level /Library/Widgets/ folder. The 14 widgets that ship

with Tiger are installed in the /Library/Widgets/ folder at the root level of your hard disk and everyone who has an account on your Macintosh can see them when they log in. Any widget that you install is placed in the /Users/<your account>/Library/Widgets/ folder. Others users that have accounts on your Macintosh cannot see your widgets unless you copy them to their Widgets folder.



Figure 1-1



Figure 1-2



Figure 1-3

OS X has four filesystem domains: User, Local, Network, System. These domains control access to resources on your Macintosh. The User domain contains resources specific to the user who is logged in to the system. The System domain contains resources that are required by the system to run. Users cannot add to, remove, or change items in this domain. Each domain has its own Library folder and the contents of one Library folder can override another. If you modify the Weather widget and place a copy of it in your Widgets folder, for instance, it will be used instead of the copy in the System's Widgets folder.

Though you can copy a widget into the System's Widgets folder, it isn't advisable because any update of the operating system may overwrite the widgets you place there.

The small arrows on the left and right sides of the Widget Bar allow you to see the next page of widgets. When you move the cursor over the arrow, it shows you how many more pages of widgets you have, like the "1 of 2" on the right side of the Widget Bar. You can scroll from one page of widgets to the other by typing Command-Left Arrow or Command-Right Arrow.

You can add a widget to Dashboard by dragging it from the Widget Bar into the Dashboard area. A ripple effect, like dropping a leaf on a pond, lets you know that the widget has been added (Figure 1-4). If you want to remove a widget from Dashboard, click the Close box in the upper-left corner of the widget that you want to remove, and the widget disappears into the Close box.

You can close a widget without opening the widget by holding down the Option key whenever you move your cursor over it. This displays the Close box for the widget.



Figure 1-4

### **Managing Dashboard's Widgets**

In the Tiger 10.4.2 release, Apple introduced the Widgets widget. The Widgets widget provides a compact interface for managing widgets (Figure 1-5). You open the Widgets widget by clicking the Manage Widgets button after you have opened the Widget Bar. Widgets provides you with a scrolling list of the widgets installed in your account and the System. This widget allows you to sort, hide, remove, and find additional widgets. The Sort menu at the top of the widget lets you sort installed widgets by name or by date.



Figure 1-5

You can hide a widget without removing it from Dashboard by unchecking the checkbox next to the widget's name in the scrolling list. Doing this leaves the widget installed, but it is no longer displayed in the Widget Bar, which reduces the amount of horizontal scrolling that you have to do when you are adding widgets to Dashboard. One way to manage your widgets, for instance, is to uncheck all of the widgets that you want to keep installed, but don't use very often. The widgets that you use regularly but don't add to Dashboard are in the Widget Bar, ready for you to add quickly.

If you want to remove or uninstall a widget, you click the red minus sign next to its name in the Widgets widget. Dashboard will confirm that you want to trash the widget (Figure 1-6).

The minus also helps you distinguish between the widgets that you've installed from those included with Tiger because the latter are not followed by a minus sign in a red circle.

You will not see a red minus next to the widgets installed in the System's /Library/ Widgets/ folder. This is primarily because you don't have modification privileges to that folder as a standard user, but even if you log in as root, a red minus does not appear next to these widgets.



Figure 1-6

Clicking the More Widgets button at the bottom of the Widgets widget launches Safari, if it isn't already running, and takes you to Apple's Dashboard Widgets website (http://www.apple.com/downloads/dashboard/).

### **Installing Widgets**

So how do you get additional widgets installed? In the Tiger 10.4.2 release, Apple changed widget installation procedures to respond to concerns about security. When Tiger was first released, widgets were automatically installed after they were downloaded. In addition to explaining the problem, Stephan Meyers created a page on his website (http://stephan.com/widgets/zaptastic/) that would automatically install a benign Zaptastic widget when you visited it. If you visit that page now, you'll see how Apple has improved the installation security by preventing automatic installation and adding confirmation dialogs.

#### **Installing Local Widgets**

If you want to install a widget that you are developing, you have two options. You can open your /Library/Widgets/ folder and drag the widget directly into the folder. After you do this, you must open Dashboard and drag it from the Widget Bar onto Dashboard.

You will probably spend lots of time in your /Library/Widgets folder as you develop widgets, but another approach is to use the Finder's built-in mechanism for installing widgets. Before you run through this example, you should have Apple's Xcode development environment installed because you will use one of the example widgets supplied with it.

#### Try It Out Install a Local Widget

**1.** Open the /Developer/Examples/Dashboard/Sample Widgets/ folder.

**2.** Double-click the Hello World widget (Hello World.wdgt). The Widget Installer prompts you to make certain that you want to install the widget (Figure 1-7).



Figure 1-7

3. Click Install. Dashboard opens and prompts you to keep or delete the widget (Figure 1-8).

Hello World	
Hello, World!	
Delete Keep	

Figure 1-8

**4.** Click Delete to remove the Widget without installing it in Dashboard, or click Keep to add it to Dashboard.

#### **How It Works**

The Widget Installer is part of the Tiger 10.4.2 revision and was included to address security concerns. The Widget Installer prevents the automatic installation of widgets so a malware widget can't be installed without your knowledge. If a web page tries to install a widget, as the Zaptastica page at Stephan Meyers's website does, you will know immediately and be able to cancel the installation.

If you decide to install the widget, the Widget Installer adds it to your Widgets folder and opens Dashboard with the widget in a box. The widget is fully functional, and you can try it out before you decide if you want to keep it. If you decide that you don't want to keep the widget and click the Delete button, the widget is moved to the Trash.

Because the Hello World.wdgt has been moved to your Widgets folder, you will not have a copy in the Sample Widgets folder. You will need to keep this in mind when you are developing widgets and always keep a copy in your development area.

#### **Downloading and Installing**

Typically, users are going to download widgets from Apple's Dashboard website or from another website. When they do this, Safari interacts with the Widget Installer to install the widget in Dashboard after it has been downloaded.

When you download a widget, Safari notices that it is an application and confirms that you want to continue downloading (Figure 1-9).

This allows you to cancel the download if the file isn't what you thought it was.



Clicking the Download button continues the download and decompresses the file. You are asked if you want to install the widget after downloading it (Figure 1-10).







#### Figure 1-10

Notice that Gas.wdgt is downloaded to your downloads folder. If you use the Desktop as your downloads folder, it will show up as in Figure 1-10.

As you saw in the Try It Out section, if you click the Install button, the widget is copied to the Widgets folder in your Library folder and Dashboard opens with the widget in a box and the option to keep or delete it (Figure 1-11).



Figure 1-11

If you aren't using Safari, you will probably have to double-click the widget to open the Widget Installer.

#### **Reloading Widgets**

Because widgets are essentially web pages, they behave the same way web pages do in browsers. If you make a change to a widget while you are developing it, you will have to reload the widget. You reload a widget in the same way you reload a page in Safari—you press Command-R. Select a widget and then press Command-R, the widget will spin (Figure 1-12) to let you know that it has reloaded.



Figure 1-12

You have to reload a widget for the same reason that you have to reload a page in Safari. Safari maintains a cache to make pages that you visit regularly load faster. Dashboard also maintains a cache.

### What Widgets Are Not

Now that we know what widgets are, a word or two is in order about what widgets are not. Widgets are not applications that run in Dashboard. In other words, you should not plan to write a large, multifunction application as a widget. A word processor would not make good widget. A good widget should be small, focused in purpose.

Perhaps the best thing to do is to think of a real-world widget. That simple device typically does one thing very, very well and is better than anything else at what it does. The epitome of a widget for me is a paint scraper. You can find several varieties, but they all have one thing in common: a standard flat blade perfectly mounted to slide along glass under a layer of paint.

A Dashboard widget should functionally mimic that real world widget. Rather than create a database application to run as a widget, create your database in MySQL or Postgresql and create a widget interface to your database. The database itself runs in its own memory space as a process so you don't have the overhead of a GUI, and your widget provides at-your-fingertips access to the data so you can query or create records. At least one web browser widget has been written for Dashboard. Ignoring the recursive ness, this is a good example of what not to do. Though widgets are essentially web pages running under Dashboard, they still use memory, as we will see in Chapter 7. The larger and more complex the widget, the greater the impact it will have on memory. Widgets should be thought of as a delivery mechanism for data.

A widget can be a very elegant solution to normally intractable problem, and the best thing about widgets is that they are easy to write. So let's get started!

### Summary

Dashboard widgets zoom into place with the touch of a keystroke and provide access to information on your computer, your network, and the Internet. Because widgets are based on familiar web technologies, they are easy to create. Over a thousand widgets have already been uploaded to Apple's Dashboard website with more coming in every day — everything from Asteroid Run to Wikipedia. Because they run inside of Dashboard, you can create widgets that look more like applications than web pages because you aren't limited to a rectangular browser window.

In this chapter, you learned to:

- Manage widgets to reduce scrolling
- □ Install widgets using the Widget Installer
- □ Reload widgets whenever you make changes

In Chapter 2, you look inside one of Apple's example widgets to see how it is created from different source and image files. You also look at how to bundle the files to make an executable widget and how to install the widget to test it.

Before you move on to Chapter 2, you may want to work through the exercises below to check your understanding of the concepts in this chapter. The answers to the exercises are in Appendix A.

### Exercises

- **1.** What were the code names of OS X 10.0 and 10.1?
- 2. Why do you have to keep a copy of your widget in your development area?
- **3.** What's a Unixy way to describe the location of your Widget folder, not the System's Widget folder?