PART 1

Foundations and Scaffolding

This section is where the foundations of your Ruby knowledge will be laid. By the end of this section you'll be able to develop a complete, though basic, Ruby program. You'll learn how to get Ruby working, what object orientation is, how to develop some basic programs, and about the data types and control structures Ruby uses and can operate on. Finally, I'll walk you through creating a small program from start to finish.

CHAPTER 1

Let's Get It Started: Installing Ruby

Ruby is a popular programming language, but not many computers understand it by default. This chapter takes you through the steps necessary to get Ruby working on your computer.

As an open source language, Ruby has been converted (or "ported," as is the technical term) to run on many different computer platforms and architectures. This means that if you develop a Ruby program on one machine, it's likely you'll be able to run it without any changes on a different machine. You can use Ruby, in one form or another, on all the following operating systems and platforms:

- Microsoft Windows 95, 98, XP, and Vista (all varieties)
- Mac OS X (all varieties)
- Linux (all varieties)
- MS-DOS
- BSDs (including FreeBSD and OpenBSD)
- BeOS
- Acorn RISC OS
- OS/2
- Amiga
- Symbian Series 60 cell phones
- Any platform for which a Java Virtual Machine exists (using JRuby, rather than the official Ruby interpreter)

■ **Caution** Some specifics of Ruby vary between platforms, but much of the code in this book (particularly in the earlier chapters) runs on all versions. When we begin to look at more complex code, such as external libraries and interfacing between Ruby and other systems, you should be prepared to make changes in your code or accept that you won't have access to every feature. However, if you're using Windows, Linux, or Mac OS X on an x86 architecture, almost everything will work as described in this book.

Before you can start playing with Ruby, you need to get your computer to understand the Ruby language by installing an implementation of Ruby on your system, which I'll cover first.

Installing Ruby

Typically, when you install Ruby onto your computer, you'll get the "Ruby interpreter," the program that understands other programs written in the Ruby language, along with a wealth of extensions and libraries to make your Ruby more fully featured. However, some installers, such as the Windows installer covered in the following section, include source code editors and more easily accessible documentation, whereas other implementations might not. Fortunately, any extras included by one distribution and not another are always available separately to install later.

To satisfy the majority of readers without referring to external documentation, I'm providing full instructions for installing and using Ruby on Windows, Mac OS X, and Linux, along with links to Ruby implementations for other platforms. In each case, I provide instructions to check that the installation is successful before sending you on to the programming fun in Chapter 2.

Windows

Ruby was initially designed for use under Unix and Unix-related operating systems such as Linux, but Windows users have access to an excellent "one-click installer," which installs Ruby, a horde of extensions, a source code editor, and various documentation, in "one click." Ruby on Windows is as reliable and useful as it is on other operating systems, and Windows makes a good environment for developing Ruby programs.

To get up and running as quickly as possible, follow these steps:

1. Open a Web browser and go to `http://www.ruby-lang.org/en/downloads/`.

2. Scroll down to "Ruby on Windows," about halfway down the page.

3. In the "Ruby on Windows" section, you'll see a few links for different versions of Ruby you can download for Windows. Ideally you want to download the file at the link that's highest in the list that's referred to as a "One-Click Installer." At the time of writing, this is version 1.8.5.

4. Click the link you found in step 3 and save it to your desktop.

5. Once download has completed, look on your desktop for the Ruby EXE file you just downloaded, and double-click it to load the installer.

6. If Windows gives you a "Security Error" box, click the "Run" button to give your approval.

7. A typical installation program appears with some instructions. On the initial screen, click "Next."

8. Work your way through the installation screens. Leave the boxes checked to install the text editors SciTE and FreeRIDE, and the Ruby package manager RubyGems (more on that in Chapter 7). Unless you have a specific reason not to, let the installation program install Ruby in its default location of `c:\ruby` and its default program group.

9. Installation takes place when you see a stream of filenames flying up your screen. Wait several minutes for the installation process to complete and enjoy the view. There are a lot of files to install!

10. Installation is complete when the installation program says "Installation Complete" and the "Next" button is clickable. Click the "Next" button, then click "Finish" to exit the installation program.

If Ruby installed correctly, congratulations! Go to the "Start" menu and then the "Programs" or "All Programs" menu. There should be a Ruby program group that contains icons for FreeRIDE, SciTE, an uninstaller, and other bits and pieces. To test that your Ruby installation works correctly for Chapter 2, you need to load the program listed as "fxri – Interactive Ruby Help & Console," so click this entry and wait for the program to load. If the program loads successfully, you'll see a screen that looks somewhat like that in Figure 1-1.

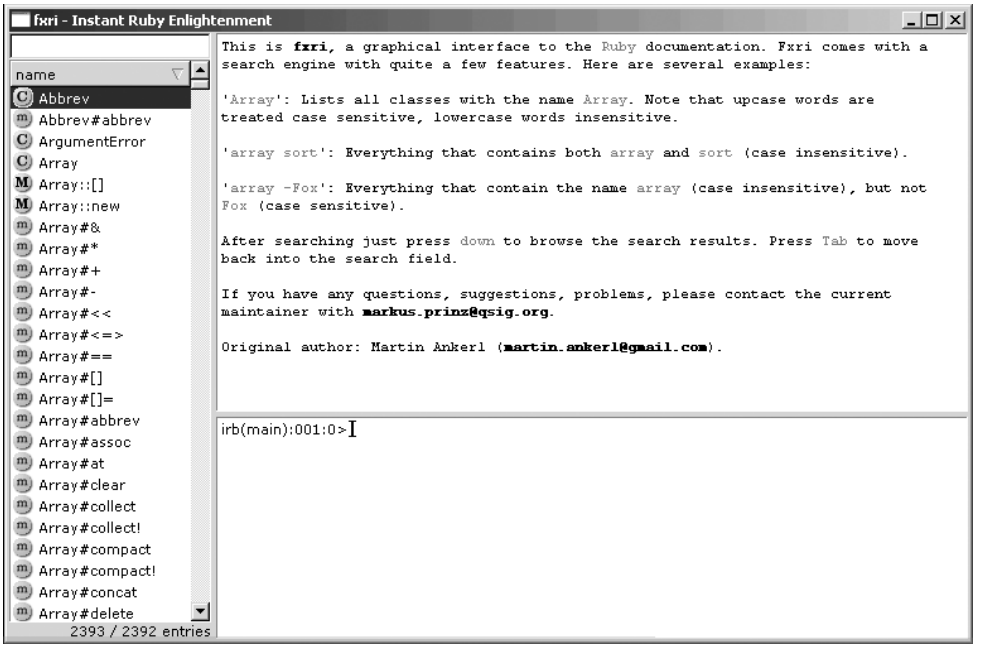

Figure 1-1. *The fxri interactive Ruby program*

If fxri started properly, then Ruby is installed correctly. Congratulations! Lastly, you need to be familiar with running Ruby and its associated utilities from the command prompt, so go to the "Start" menu, then "Run," and type **cmd** into the box and click "OK" ("Command Prompt" might also be in your "Programs" menu under "Accessories"). You should be presented with a command prompt, like that in Figure 1-2.

Figure 1-2. *The Microsoft Windows command prompt*

Throughout this book, commands that can be used at the command prompt will be given. This is because using a command prompt such as this is a standard technique in operating systems such as Linux and OS X. For example, in Chapter 7 we'll look at installing extra features (libraries) for Ruby, and the command prompt will be used for this. Therefore, it's necessary for you to know how to access it and run programs.

If you type **irb** at this prompt and press Enter, you should see something like the following:

```
irb(main):001:0>
```

If you see the preceding line, everything is set up correctly, and you can type **exit** and press Enter to be returned to the command prompt.

Now you can move on to Chapter 2 and start to play with the Ruby language itself.

Apple Mac OS X

Unlike Windows, most modern Apple machines running Mac OS X come with a version of Ruby already installed, which means you can get started straight away. Mac OS X Panther (10.3.x) comes with Ruby 1.8.2 by default, and OS X Tiger (10.4.x) comes with Ruby 1.8.4.

■ **Note** It's likely that OS X Leopard, due to be released in 2007, will come with the latest version of Ruby, so if you're running that operating system, unavailable at the time of writing, you might already be set to go!

Most of the code in this book works fine with Ruby 1.8.2 or higher, so if you're running Mac OS X Panther or Tiger, you don't need to do anything special. To find out which version of OS X you're running, click the "Apple" menu at the top left of your screen and select "About This Mac." If the version of OS X is later than 10.3, you should have Ruby installed already.

■ **Tip** If you're using OS X Tiger (10.4.x), use Apple's Software Update to upgrade to the latest version of OS X, as Apple improved Ruby distribution included in OS X from version 10.4.6 onward. Without this upgrade, you might need to reinstall Ruby manually to get some extensions, such as Ruby on Rails, to work correctly. Although this isn't a concern for the first two sections of this book, it could cause you some confusion later on.

Testing for a Preinstalled Version of Ruby

If you're using OS X Panther or OS X Tiger, you can check whether Ruby is installed by using the Terminal application. Double-click "Macintosh HD" (or whatever your hard drive is called) and go to the `Applications` folder on your drive. Once in `Applications`, go to the `Utilities` folder, where you'll find an application called Terminal. Double-click its icon to start it. If Terminal starts correctly, you'll see a screen similar to that in Figure 1-3.

Once you're in the Terminal, you're at what's called the *command prompt* or *shell*. You're talking directly with your computer, in a technical sense, when you type. The computer will execute the commands that you type immediately once you press Enter.

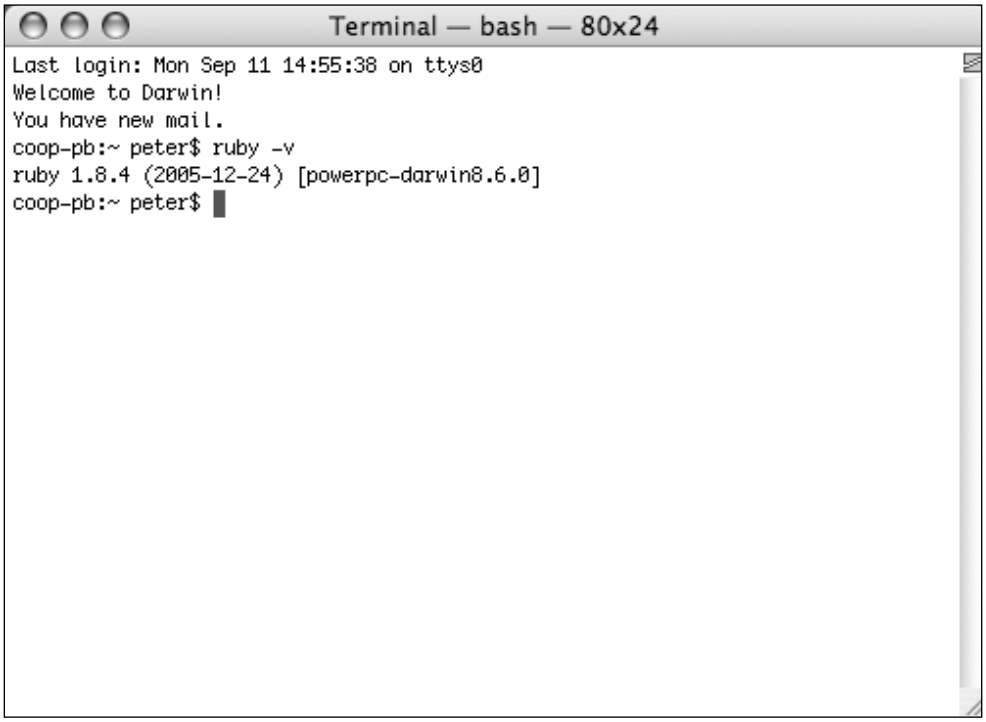

Figure 1-3. *The Mac OS X Terminal in OS X Tiger with a working Ruby installed and tested*

To see if Ruby is installed, type the following at the command prompt from within Terminal *(be sure to press Enter afterward)*:

```
ruby -v
```

If successful, you should see a result, as shown in Figure 1-3, that says what version of Ruby you're running (which should, ideally, be 1.8.2 or greater). If this works, try to run the Ruby interactive interpreter called "irb" by typing the following at the command prompt:

```
irb
```

If you get a result as shown in Figure 1-3, you're ready to go and can move to Chapter 2. If you need to install a newer version of Ruby on OS X, continue to the next section.

Installing Ruby on OS X

There are a few ways to install Ruby on OS X. You can install from a prepackaged installation, by using a package manager such as Fink or DarwinPorts, or by compiling the Ruby source directly. If you already use Fink or DarwinPorts, then refer to their respective sites for further information, but otherwise you'll find it easier to use a prebuilt installation package.

One of the most popular installation packages is called Locomotive, and is available at `http://locomotive.raaum.org/`.

As a regular DMG OS X file, you can install it like any other OS X application (on both PPC and x86 architectures). Unlike some installers, Locomotive includes Ruby on Rails and LightTPD. These tools aren't immediately useful, unless you're planning to do some Ruby on Rails development right away, but which you'll be glad of by the end of this book.

Installing Ruby from Source on Mac OS X

Installing Ruby directly from source code on OS X is similar to Linux, so continue on to the later Linux section entitled "Installing Ruby From Source Code." Please note that versus installing a package such as Locomotive, when you install Ruby by source, all you get is Ruby. You need to install components such as Rails separately later.

■ **Note** To compile the Ruby sources on OS X, you need to install the Xcode developer tools that come with OS X.

Linux

As an open source programming language, Ruby is already installed with many Linux distributions. It's not universal though, but you can check if Ruby is installed by following the instructions in the next section. If this fails, there are further instructions to help you install it.

Checking If Ruby Is Installed on Linux

Try to run the Ruby interpreter from the command prompt (or terminal window), as follows:

```
ruby -v
```

If Ruby is installed, it will give an output such as the following:

```
ruby 1.8.2 (2004-12-25) [i686-linux]
```

This means that Ruby 1.8.2 is installed on the machine. This book requires 1.8.2 as a bare minimum, so if the version is less than 1.8.2 you'll need to continue onward in this chapter and install a more recent version of Ruby. However, if Ruby appears to be installed and up to date, try to run the irb interactive Ruby interpreter, as follows:

```
irb
```

■ **Tip** On some systems, irb might have a slightly different name. For example, on Ubuntu it can sometimes be called irb1.8, and you'll need to run it as such. To find it, you can use `find / -name "irb" -maxdepth 4`.

Once you've run irb, you should get the following output:

```
irb(main):001:0>
```

If running irb results in a similar output, you can move on to Chapter 2. (You might wish to type **exit** and press Enter to get back to the command line!) Otherwise, read on to install a new version of Ruby.

Installing Ruby with a Package Manager

The installation procedure for Ruby on Linux varies between different Linux distributions. Some distributions, such as Gentoo, Debian, and Red Hat, provide "package managers" to make installation of programs easy. Others require that you install directly from source or install a package manager beforehand.

If you're comfortable with using `emerge`, `rpm`, or `apt-get`, you can install Ruby quickly with the following methods:

- *RPM*: Download Ruby RPMs and install with `rpm -Uhv ruby-*.rpm`
- *Gentoo*: Use `emerge` as follows: `emerge ruby`
- *Debian*: Use `apt-get`: `sudo apt-get install ruby`
- *Ubuntu*: Use `apt-get` as with Debian. You might also need to install irb explicitly. In the case of Ruby 1.8, this line should work:

  ```
  sudo apt-get install ruby ruby1.8 ruby1.8-dev rdoc ri irb
  ```

If one of these methods works for you, try to run Ruby and irb as shown in the preceding section, and progress to Chapter 2 if you're ready. Alternatively, you can search your distribution's package repository for Ruby, as the name for the Ruby package in your distribution might be nonstandard or changing over time. However, if all else fails, you can install Ruby directly from its source code in the next section.

Installing Ruby from Source Code

Installing Ruby from its source code is a great option if you don't mind getting your hands dirty. The process is similar on all forms of Unix (not just Linux). Here are the basic steps:

1. Make sure that your Linux distribution is able to build applications from source by searching for the "make" and "gcc" tools. From the terminal you can use `which gcc` and `which make` to see if the development tools are installed. If not, you need to install these development tools.
2. Open a Web browser and go to `http://www.ruby-lang.org/`.
3. Click the "Download Ruby" link on the right-hand side of the page. If the page's design has changed, look for a link to "downloading Ruby."
4. On the download page, click the link to the stable version of the "Ruby Source Code." At the time of writing this is "ruby-1.8.5." This downloads the `tar.gz` file containing the source code for the latest stable version of Ruby.

5. Uncompress the `tar.gz` file. If you're at a command prompt or terminal window, go to the same directory as the `ruby-1.x.x.tar.gz` file and run `tar xzvf ruby-1.x.x.tar.gz` (where `ruby-1.x.x.tar.gz` is the name of the file you just downloaded).

6. Go into the Ruby folder that was created during decompression. If you're not using a command prompt at this stage, open a terminal window and go to the directory.

7. Run `./configure` to generate the Makefile and `config.h` files.

8. Run `make` to compile Ruby from source. This might take awhile.

9. Run `make install` to install Ruby to its correct location on the system. You need to do this as a superuser (such as root), so you might need to run it as `sudo make install` and type in the root password.

10. If there are errors by this stage, read the README file that accompanies the source code files for pointers. Otherwise, try to see what version of Ruby is now installed with `ruby -v`.

If the expected version of Ruby appears at this point, you're ready to move to Chapter 2 and begin programming. If you get an error complaining that Ruby can't be found, or the wrong version of Ruby is installed, the place where Ruby was installed to might not be in your path (the place your operating system looks for files to run). To fix this, scroll up and find out exactly where Ruby was installed (usually `/usr/local/bin` or `/usr/bin`) and add the relevant directory to your path. The process to do this varies by distribution and shell type, so refer to your Linux documentation on changing your path.

Once you can check which version of Ruby is running and it's 1.8.2 or over, and you can run irb and get a Ruby interpreter prompt, your Ruby installation is complete (for now!) and you can move on to Chapter 2.

Other Platforms

If you're not using Windows, Mac OS X, or Linux, you can still use Ruby if your computer's platform and architecture are listed at the start of this chapter. If you're a user of an uncommon platform, I assume you have basic knowledge of how to install applications on your system, so I simply provide the links to the following different installation programs:

- *MS-DOS*: `http://ftp.ruby-lang.org/pub/ruby/binaries/djgpp/`.
- *FreeBSD*: Various versions of Ruby are available as standard ports.

- *OS/2*: `http://hobbes.nmsu.edu/pub/os2/dev/misc/ruby-181.zip`.
- *BeOS*: Ruby is installable in the same manner as with Linux, as described earlier.
- *Linspire or Lindows*: As a Linux distribution, you can use the same instructions as for Linux, earlier.
- *Symbian Series 60*: `http://developer.symbian.com/main/tools/opensrc/ruby/index.jsp`.
- *Java Virtual Machines (JVMs)*: `http://jruby.codehaus.org/`.
- *Other Unix versions*: Refer to "Installing Ruby from Source Code" in the preceding Linux section for instructions that are reasonably distribution generic.

In many cases, the versions of Ruby for some operating systems might be out of date or unsupported. If this is the case, and you're confident about being able to compile your own version of Ruby directly from the Ruby source code, the source code is available to download from `http://www.ruby-lang.org/en/20020102.html`.

To test that Ruby is installed sufficiently to continue with this book, you want to check which version of Ruby is installed by asking Ruby for its version, as follows:

```
ruby -v
```

You also need access to Ruby's interactive prompt irb. You access this simply by running irb (if it's in your path) as follows:

```
irb
```

If neither Ruby nor irb work without complaint, you need to seek assistance for your specific platform. Appendix C provides a list of useful resources.

Summary

In this chapter we've focused on making sure Ruby is properly installed and that you can run the irb tool that you'll be using over the next several chapters.

Although Ruby is an easy language to learn and develop with, it's easy to become overwhelmed with the administration of Ruby itself, its installation, and its upgrades. As Ruby is a language constantly in development, it's likely that points covered in this chapter will go out of date, or easier ways to install Ruby might come along for your platform.

An important part of being a Ruby developer is being able to use the resources the Ruby community makes available, and being able to find help directly as time goes by. The Ruby community can provide quick help in most cases, and a number of resources to try are available in Chapter 5 and Appendix C.