Ruby Installation and Use Instructions

Contents

Overview of Your Options

The last two homeworks will use the Ruby language. We recommend editing your files using Emacs and running programs with `irb`, which is Ruby's REPL. If you prefer a different editor that has support for Ruby, that is completely fine. This document describes basic installation and usage steps sufficient for doing your homework. Unlike when we used SML, we recommend running the REPL from a terminal (command-shell) window, *not* from within Emacs. This is described below.

For installation purposes related to Homework 6, there are three key facts:

1. You need to have either some version of Ruby 1.9.3 or some version of Ruby 2.0. We have done more extensive testing with Ruby 1.9.3., but we are prepared to support Ruby 2.0 as well. It is unlikely any differences between the versions will be relevant to the homework. If you have an older version of Ruby 1.9, such as Ruby 1.9.1, that might work. However, we are not able to support Ruby 1.8 and earlier; we know they are unlikely to work.
2. You need to have a version of the Tk graphical toolkit and Ruby needs to be able to find it. This fact drives many of the more detailed installation instructions. However, this is only for Homework 6, not Homework 7 and there is an alternative: If you have Java installed, you can install and use jruby instead. These instructions do not go through the details, but we have provided a jruby-compatible version of the Homework 6 provided code and will try to help as best we can on the discussion forums.
3. If you have Emacs version 23 or higher, you should not need to configure it in any special way for using Ruby: opening a file with extension `.rb` should automatically use Ruby mode.

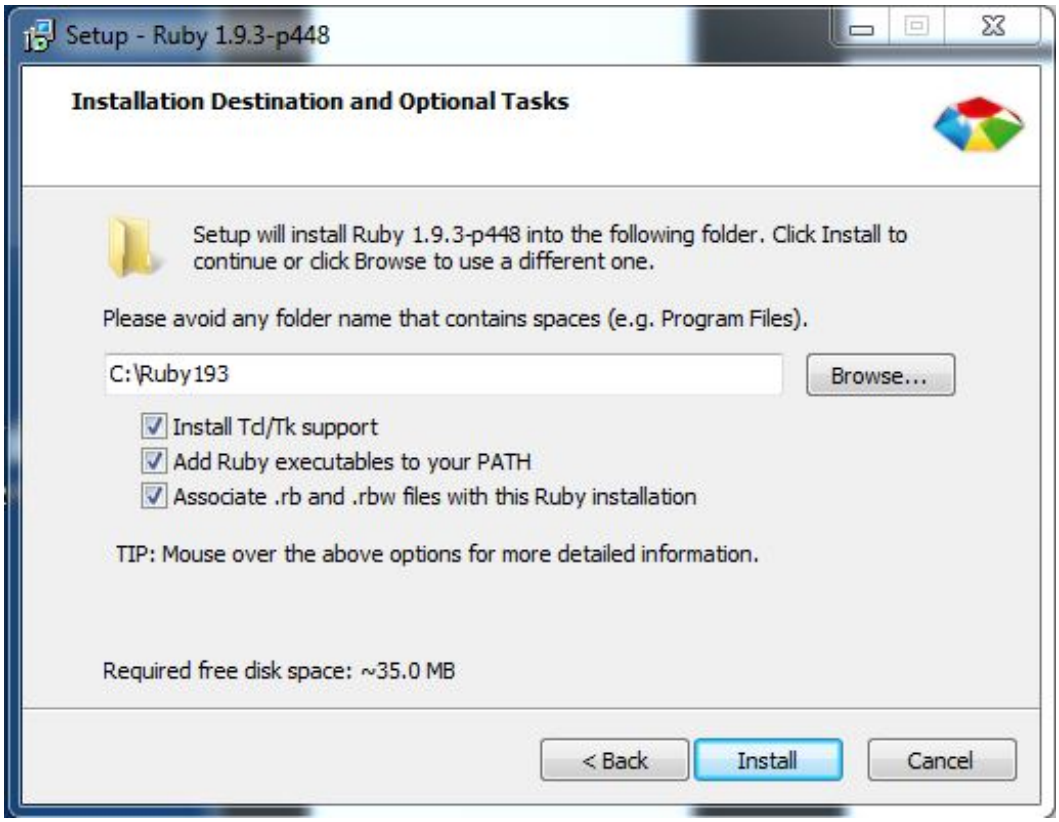
Because this document has separate installation instructions for different operation systems, you can skip sections not relevant to you. But also notice the last two sections of this document, which have usage information relevant to all operating systems.

Windows

- If you did not install Emacs earlier in the course but want to use Emacs now, install Emacs using the instructions for the SML portion of the course. Note you do not need SML mode -- all you need

to do is install Emacs.

- Go to http://rubyinstaller.org/, click on the giant red Download button, and click on the latest Ruby 1.9.3 RubyInstaller. (At the time of writing, this is Ruby 1.9.3-p545 and it is the third link from the top.)
- Run the installer as follows:
  - Accept the license.
  - **On the next screen, click all three boxes: Install Tcl/Tk support, Add Ruby executables to your PATH, and Associate `.rb` and `.rbw` files with this Ruby installation. Like this:**



  - Then click Install.
- Create a Ruby file (e.g., by downloading some provided code or by creating a blank file with extension `.rb`) wherever is convenient (e.g., where you have been storing your other homework assignments).
- Open Emacs or whatever text editor you are using.
- Open your Ruby file in Emacs, by dragging it from an Explorer window onto Emacs, or by *C-x C-f* and then entering the full path to the file, or by using the File menu. You can now edit and save the file as usual.
- In Windows, open a command shell by running the `cmd.exe` program (a Windows shell). Use the `cd` command to switch to the directory where your Ruby file is (using the Windows shell you separate directories with backward slashes `\`). *After* you are in the right directory, type `irb` to start the Ruby REPL.

Linux

Most Linux distributions should make it easy to install Ruby 1.9.3 and connect it to the Tk graphics

library. We assume here that Tk is already installed because it comes standard with most distributions. If Tk is not installed, use your preferred package manager to install it.

- When you install Ruby, explicitly get 1.9.3. Beware installing the default `ruby` package, which may not be Ruby 1.9.3. Although, as mentioned at the beginning of this document, Ruby 2.0 is also supported -- the rest of these detailed steps just assume Ruby 1.9.3.
- You can get Ruby 1.9.3 with your favorite package manager GUI, or on the command line. For example, on Debian (Ubuntu, Mint) systems, type: `sudo apt-get install ruby1.9.3`
- If `irb` is a separate package, install that too, again looking for an explicit use of 1.9.3.
- After installation, running `ruby --version` should indicate ruby 1.9.3.
- Create a Ruby file (e.g., by downloading some provided code or by creating a blank file with extension `.rb`) wherever is convenient (e.g., where you have been storing your other homework assignments).
- Open Emacs or whatever text editor you are using.
- Open your Ruby file in Emacs by *C-x C-f* and then entering the full path to the file, or by using the File menu. You can now edit and save the file as usual.
- Open a command shell however is convenient.
- Use the `cd` command to switch to the directory where your Ruby file is (separating directories with forward slashes `/`). *After* you are in the right directory, type `irb` to start the Ruby REPL.

Mac

For various reasons, Mac is by far the most difficult operating system in which to install Ruby 1.9.3 such that it can use the Tk graphical toolkit. So if you have another operating system available, you may choose to use it. Nonetheless, we have developed an installation script that we tested successfully on OS X versions 10.6, 10.7, 10.8, and 10.9, so we hope it works for you if you want to use Mac. Older Mac versions unfortunately have Ruby 1.8.7 installed, which will not work, but if you have the newer Ruby 2.0 *and* it can find the Tk graphics library, then that will work.

Follows these steps in order:

1. You need to have Xcode, which includes software-development tools for Mac. It is free. If you have OS X 10.6.6 or later, you can download the latest version of Xcode from the Mac App Store; just go to the Mac App Store and search for it. If you already have an older version of Xcode, that is probably fine, but if you later run into problems with our script, you might try deleting the directory our script creates, upgrading Xcode, and trying again. If you have a version of OS X older than 10.6.6 or you want to install an older version of Xcode, you can get Xcode from https://developer.apple.com/downloads/ (account-registration required).
2. You need Xcode to include command-line tools. If you just downloaded Xcode or otherwise have a version of Xcode >= 4.3, this is not the default. Go to the Downloads tab within the Xcode Preferences menu and click "Install" next to the Command Line Tools entry.
3. You need to make sure your computer allows applications to be downloaded and run so that the Ruby-installation and Ruby-running applications used in the following steps will work. You have likely already changed this security setting to install other programs, but if not, in OS 10.7 and later, you can open "System Preferences" from the Dock or the Apple menu, choose "Security & Privacy", go to the "General" tab and make sure "Allow application downloaded from:" is set to

"Anywhere". (You may need to click the lock icon in the lower left to authenticate as an administrator and unlock this option.) For more information, see http://support.apple.com/kb/HT5290.

4. Your computer needs to have Tk 8.5 (not another version) installed in a standard enough place our script can find it. If you have OS X 10.6, 10.7, or 10.8, you probably already satisfy this requirement and you can skip this step. If you have OS X 10.9, you probably do not satisfy this requirement and you can download 8.5 (not the more recent 8.6) from http://www.activestate.com/activetcl/downloads. In all cases, you can skip this step for now and proceed to our script below. If the script fails to find Tk 8.5, it will say so in an error message and you can install it and then re-run the script, which should automatically continue from where it stopped rather than starting over.
5. You need to download the script that can do the rest of the installation. Save

   this script

   in a folder somewhere convenient, such as where you have been keeping your homework for this course.
   Name the file `install-ruby1.9.3-with-tk.sh`.
6. Open a command shell (available as Applications/Utilities/Terminal.app) and use the `cd` command to change into the directory where you saved `install-ruby1.9.3-with-tk.sh`.
7. In the next step we are going to run `install-ruby1.9.3-with-tk.sh`, which will download and install all the software you need. This will take some time and requires that you trust the script. A lot of output will get printed while everything installs. The script will make the following changes to your computer:
   - It will add a directory `ruby1.9.3-with-tk` to your home directory (the directory you get to with `cd ~`) and put many things in it.
   - At the end of the installation process, it will ask you if you want some program names (like `ruby` and `irb`) to refer to the software just installed. If you say yes, which we recommend, the script will edit the `.bash_profile` file in your home directory appropriately.

   You need to run the script only once. If it has a problem, when you re-run it, it will skip past the (possibly time-consuming) steps where it already succeeded. If you want to re-run everything, first delete the directory `ruby1.9.3-with-tk` (dragging it to the Trash and emptying the Trash should work) and then re-run the script.
8. At the command-line run:

   ```
   bash install-ruby-1.9.3-with-tk.sh
   ```

   and follow the instructions. As mentioned above, this will take some time (more than a minute, hopefully less than an hour) as it downloads, compiles, and installs everything.

Now that we are done installing, we can write Ruby code and use Ruby:

- Create a Ruby file (e.g., by downloading some provided code or by creating a blank file with extension `.rb`) wherever is convenient (e.g., where you have been storing your other homework assignments).
- Open Emacs or whatever text editor you are using.
- Open your Ruby file in Emacs by *C-x C-f* and then entering the full path to the file, or by using the

File menu. You can now edit and save the file as usual.
- Open a command shell however is convenient.
- Use the `cd` command to switch to the directory where your Ruby file is (separating directories with forward slashes `/`). *After* you are in the right directory, type `irb` to start the Ruby REPL (assuming you chose to have `irb` refer to the just-installed software).

General Information on Using the REPL (or not)

- As described previously, run `irb` from the command line to start the REPL.
- In the REPL, to run the code in file `foo.rb`, do `load "foo.rb"` assuming the file is in the same directory where you started `irb`.
- As usual, it is least error-prone to restart the REPL after editing and resaving any files you are using. Reloading a file without restarting may or may not work depending on what has changed.
- Moreover, in Homework 6, it will probably *not* work to quit and restart the game within the REPL. Instead, exit and restart the REPL after you "quit" the game. (Using "new game" within the game without pressing "quit" should work without restarting the REPL.) But when running the game, there really is no need to use the REPL. The REPL is better when testing individual pieces of code. See below for how to run a Ruby program without using the REPL.
- To quit the REPL, type `quit` or `exit` or *Control-d* (i.e., `d` while holding down the *Control* key). As usual in Ruby, there is more than one way to do things.
- You can cycle through previous input lines by using the up and down arrows.
- Unlike in SML, do *not* end a line of REPL input with a semicolon. This will not work because the semicolon indicates to the REPL exactly that you want to enter another expression before evaluating anything.

You can also run a Ruby program that is in file `foo.rb` by running `ruby "foo.rb"` from the shell command-line (the place where you ran `irb`, *not* from within `irb`). For this to be useful, your Ruby file should have some top-level expression like a call to a method outside of any class or method definition. Otherwise, "nothing will happen" because just defining methods has no observable effect until you use one of them.

On Windows and perhaps other operating systems, you can also just double-click on your `foo.rb` file to have the same effect as running `ruby foo.rb`. This approach may bring up another blank window, which you can ignore.

Main Links for Ruby Documentation, etc.

- The main Ruby home page is http://www.ruby-lang.org/.
- For documentation, you may find http://ruby-doc.org/ at least as useful. See in particular the standard-library documentation at http://ruby-doc.org/core-1.9.3/.
- There are many excellent books on Ruby available, which you may find useful but not necessary for the course. We particularly recommend Programming Ruby 1.9: The Pragmatic Programmers' Guide by Dave Thomas et al; see http://pragprog.com/titles/ruby4.

Acknowledgments: These instructions were primarily written by Dan Grossman, but he received

significant help through the pain of others, particularly for the Mac instructions. Ben Wood wrote the Mac script that takes care of most installation steps. Earlier instructions for Mac were spearheaded by Max Forbes. Thanks also go to Brandon Holt, Alexander Kanavin, Hal Perkins, and Rachel Sobel.

Created Thu 31 Oct 2013 11:40 AM PDT

Last Modified Tue 11 Nov 2014 3:14 PM PST