

CSE341: Programming Languages Using Ruby and irb Winter 2013

Contents

1	Overview	1
2	Windows	1
3	Linux	3
4	Mac	4
5	General Information on Using the REPL (or not)	6
6	Main Links for Ruby Documentation, etc.	7

1 Overview

The last two homeworks will be using the Ruby language. We recommend editing your files in emacs and using irb, which is Ruby's REPL. (If you prefer a different editor that has support for Ruby, that is 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 (shell) window, *not* from within emacs. This is described below.

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

- You need to have some version of Ruby 1.9.3, *not* Ruby 1.8.
- You need to have a version of the Tk graphical toolkit.
- 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 use Ruby mode.

Because this document describes installation for various operating systems, you do not need to read it all. Just find the section that is most convenient for you. But also notice the last two sections of this document, which have information relevant to all operating systems.

2 Windows

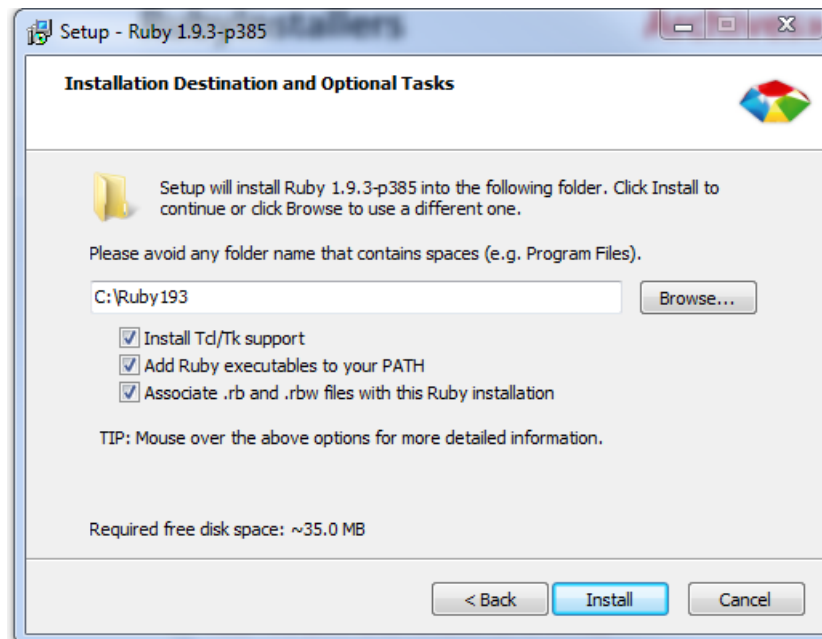
2.1 Using Windows in the Department Undergraduate Labs

- You do not need to install any software.
- Create a Ruby file (e.g., by downloading the provided code and renaming it) wherever is convenient (e.g., where you have been storing your other homework assignments). **Do not store it on the desktop or under C:\ because all changes to these directories get erased when you log out. Store your files under Z:\ (your directory on the department file system) or somewhere else (e.g., on a personal usb drive).**
- Open emacs however is convenient. You should be able to find it by, e.g., typing `emacs` in the "Search Programs and Files" from the Start Menu.

- 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. Edit and save the file as usual.
- In Windows, open a command shell, either by running `cmd` (a Windows shell) or a cygwin bash 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 (`\`) and with cygwin you use forward slashes (`/`)). *After* you are in the right directory, type `irb` to start the Ruby REPL.

2.2 Using Your Own Windows Machine

- If you did not install emacs earlier in the course, do so 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://www.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-p385).
- Run the installer.
 - 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 the provided code and renaming it) wherever is convenient (e.g., where you have been storing your other homework assignments).
- Open emacs however is convenient, e.g., from the Start Menu.
- 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 `cmd` (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.

2.3 Using the Department’s Windows Machines Remotely

- Follow the instructions for your operating system at <http://vdi.cs.washington.edu/vdi/>.
- As they mention there, note that you will be logged off (and your programs will be closed) if you are idle for 1 hour. So, please save often, and save in your `Z:\` directory!
- You can now follow the directions above for *Using Windows in the Department Undergraduate Labs*.

3 Linux

3.1 Using Linux in the Department Undergraduate Labs

- You do not need to install any software.
- Create a Ruby file (e.g., by downloading the provided code and renaming it) wherever is convenient (e.g., where you have been storing your other homework assignments).
- Open emacs however is convenient (one way is to *activities*, then *applications*, then the icon for *emacs*).
- 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 (one way is to find the terminal program under the large list of applications).
- 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.

3.2 Using Your Own Linux machine

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

- When you install Ruby, explicitly get 1.9.3. Beware installing the default “ruby” package, which may or may not be Ruby 1.9.3 — it may by default attempt to install Ruby 1.8.7.
- 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 the provided code and renaming it) 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.

3.3 Using the Department's Remote Linux Server `attu`

- On Unix or Mac
 - Open a terminal window
 - Run `ssh -X uname@attu.cs.washington.edu`, replacing `uname` with your CSE id, and log in with your CSE Linux password.
- On Windows
 - Use a program like PuTTY (<http://www.putty.org>) to gain access to SSH.
 - Connect to `uname@attu.cs.washington.edu`, replacing `uname` with your CSE NetId, and log in with your CSE Linux password.
- On `attu`, open emacs and your Ruby code.
- On `attu`, run `irb` or Ruby from the command line as described below.

3.4 Using the Department's Linux Virtual Machine on Your Machine

The department provides a Linux virtual machine that runs on your own computer no matter what operating system you have.

Installation

- The instructions for the current release, Fedora 17, are available here:
<http://www.cs.washington.edu/lab/homeVMs/f17notes.html>
- For a more general overview of what's going on and how this works, read up here:
<http://www.cs.washington.edu/lab/homeVMs/f17notes.html>

Running Ruby

- The Department's Linux virtual machine comes with Ruby 1.9.3 installed, so you may now simply follow the instructions above for *Using Linux in the Department Undergraduate Labs*.

4 Mac

Note: running Homework 6 on a Mac is imperfect: the button text in the GUI is poorly formatted and pressing the quit button causes an exception. You can ignore these glitches.

The Dilemma

Current versions of OSX come default with Ruby installed. Perfect, right? Well, they come with Ruby 1.8.7, and we need 1.9.3. What's more, you do not want to overwrite your default Ruby installation in case something else in your Mac is using it.

RVM to the Rescue

Fortunately, there is a program called *Ruby Version Manager (RVM)* that lets multiple versions of Ruby coexist peacefully on your Mac. Even better, it now has an official, beautiful GUI called *JewelryBox*.

Hurry up, Let's Do It

The following steps were tested using **OSX 10.7 and 10.8**. See below for *OSX 10.6*.

- Download the JewelryBox app at <http://jewelrybox.unfiniti.com/> (click the green 'Download' button) and install it (extract the `.dmg` file, double click to mount it, and drag the *JewelryBox* app into your *Applications* folder).
- Open the JewelryBox app.
- Click 'Install' in the main JewelryBox window, and click 'Continue'.
- Click 'Add Ruby' on the top bar.
- Pick any stable 1.9.3 Ruby build. We tested on 'ruby-1.9.3-p125', but anything of the form 'ruby-1.9.3-pXXX' should work. Click on your desired Ruby.
- **Check the 'Make Default' box at the top.**
- Click 'Install', and wait a while while it downloads and builds Ruby. Click 'Done' when it's finished.
- Click on the icon for JewelryBox at the top bar of your Mac (it looks like a red gem) and click 'Sync Environment'.
- Open a terminal window. Use `bash`. If you don't normally use `bash`, open it with `/bin/bash --login`. RVM sets up its path with `bash`, so if you want to use another shell like `fish`, open it from a `bash` window. If none of this makes sense, you are probably using `bash`.
- Type `ruby -v` to check that you're using 1.9.3.
- If Ruby is still 1.8.7, try restarting your Mac and doing 'Sync Environment' again, and closing and opening your terminal program. RVM has to set 1.9.3 to be the default Ruby, so if you have an old terminal hanging around it might stubbornly keep using Ruby 1.8.7.

Now, naturally, *skip* the instructions for *OSX 10.6* and continue with, "Now We Can Run Ruby" below.

The following steps were tested using **OSX 10.6.8**. See above for *OSX 10.7* and *OSX 10.8*.

- Download the JewelryBox app at <http://jewelrybox.unfiniti.com/>, but not click the green 'Download' button. Instead, click the link for *download JewelryBox v.1.3*. Install it: extract the `.dmg` file, double click to mount it, and drag the *JewelryBox* app into your *Applications* folder.
- Open the JewelryBox app.
- Click 'Install' in the main JewelryBox window, and click 'Continue'.
- Click 'Add Ruby' on the top bar.
- Pick any stable 1.9.3 Ruby build. We tested on 'ruby-1.9.3-p125', but anything of the form 'ruby-1.9.3-pXXX' should work. Click on your desired Ruby.

- During the installation, it may insist on choosing RVM 1.18. We had no problem with this despite the warning on the JewelryBox website to choose an older version of RVM. Feedback is appreciated on this issue.
- **Check the ‘Make Default’ box at the top.**
- Click ‘Install’, and wait a while while it downloads and builds Ruby. Click ‘Done’ when it’s finished.
- Click on the icon for JewelryBox at the top bar of your Mac (it looks like a red gem) and click ‘Sync Environment’.
- In a text editor, open the `.bash_profile` file from your home directory. In Emacs, you can open this file with `C-x C-f ~/.bash_profile` (notice the tilde, slash, dot). If it does not have this line, add it:

```
[[ -s "$HOME/.rvm/scripts/rvm" ]] && source "$HOME/.rvm/scripts/rvm"
```

Save this file, after which you can exit the text editor.

- Open a terminal window. Use `bash`. If you do not normally use `bash`, open it with `/bin/bash --login`. RVM sets up its path with `bash`, so if you want to use another shell like `fish`, open it from a `bash` window. If none of this makes sense, you are probably using `bash`.
- Type `ruby -v` to check that you are using 1.9.3.
- If Ruby is still 1.8.7, try restarting your Mac and doing ‘Sync Environment’ again, and closing and opening your terminal program. RVM has to set 1.9.3 to be the default Ruby, so if you have an old terminal hanging around it might stubbornly keep using Ruby 1.8.7.

Now We Can Run Ruby

- At this point, you should be able to use emacs to edit files and your *Terminal* program to navigate directories and run `irb` and `ruby`. In other words, your Mac *Terminal* behaves like a Linux terminal. 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.
- For a better terminal application, consider *iTerm2*, available free at <http://www.iterm2.com/>.
- For a native emacs experience on Mac, consider *Aquamacs*, available free at <http://aquamacs.org/>.

5 General Information on Using the REPL (or not)

- As described previously, run `irb` from the command line to start 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 work depending on what has changed.
- To quit, 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 this 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” since just defining methods has no 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.

6 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/> as 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 required for the material in this course. We particularly recommend *Programming Ruby 1.9: The Pragmatic Programmers' Guide* by Dave Thomas et al; see <http://pragprog.com/book/ruby3/programming-ruby-1-9>.