

Using Ruby and irb in CSE 341 Winter 2014

Overview

This is a long version of the directions for installing Ruby.

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. This document describes basic installation and usage steps sufficient for doing your homework. We recommend running the REPL from a terminal (shell) window, *not* from within emacs. This is described below.

For installation purposes related to the homework, there are some key facts:

- The official Ruby version for the course is 2.0.0.
- If you have emacs version 24 (the current version), you should not need to configure it in any special way for using Ruby: opening a file with extension `.rb` should use Ruby mode.

This document is long, but only because we are giving information for various operating systems and choices for using or not using resources from the department (lab machines, remote machines). *Just find the section that is most convenient for you.*

Also notice the last section of this document, “General information on using the REPL (or not)” which has information relevant to all operating systems.

See the course website for the main links for the Ruby language, library documentation, etc.

Windows

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 `Ctrl-x Ctrl-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.

Using Your Own Windows Machine

- If you did not install emacs already, do so.
- Go to <http://www.rubyinstaller.org/>, click on the giant red ‘Download’ button, and click on the latest Ruby 2.0.0 RubyInstaller (at the time of writing, this is Ruby 2.0.0-p353).
- Run the installer.
 - Accept the License.
 - **On the next screen, click Add Ruby executables to your PATH, and Associate .rb and .rbw files with this Ruby installation.**
 - 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 Ctrl-x Ctrl-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.

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*.

Linux

Using Linux in the Department Undergraduate Labs

- The unit test framework doesn’t work out of the box with the linux version of Ruby 2.0.0. To make it work, type this at the command prompt:

```
gem install minitest
```

You only need to do this once.

- 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 find it under the large list of applications).
- Open your Ruby file in emacs by Ctrl-x Ctrl-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.

Using Your Own Linux machine

Most Linux distributions should make it easy to install Ruby 2.0.0.

- When you install Ruby, explicitly get 2.0.0. Beware installing the default “ruby” package, which may or may not be Ruby 2.0.0 — it may by default attempt to install Ruby 1.8.7.
- You can get Ruby 2.0.0 with your favorite package manager GUI, or on the command line. For example, on Debian (Ubuntu, Mint) systems, type: `sudo apt-get install ruby2.0.0`
- If `irb` is a separate package, install that too, again looking for an explicit use of 2.0.0.
- After installation, running `ruby --version` should indicate ruby 2.0.0.
- You can now follow the instructions above for *Using Linux in the Department Undergraduate Labs*.

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 id, 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.

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 2.0.0 installed, so you may now simply follow the instructions above for *Using Linux in the Department Undergraduate Labs*.

Mac

Using Your Own Mac Machine

The Dilemma

Current versions of OSX come default with Ruby installed, but version 1.8.7. This will probably be fine, but in case you want 2.0.0 here's how to do it. In any case you don't want to overwrite your default Ruby installation in case something else in your Mac is using it.

RVM to the Rescue

Fortunately, there's 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.

- 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 2.0.0 Ruby build. We tested on 'ruby-2.0.0-p353', but anything of the form 'ruby-2.0.0-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're probably using `bash`.
- More `bash` stuff: on one of my Macs (but not the other), I had a problem with JewelryBox installing `.bashrc` and `.bash_profile` files that caused my shell to stop reading the standard `.profile` file. I fixed this by copying the contents of both of these into `.profile` and deleting them. (Again, if this doesn't make sense, you probably haven't modified your shell file and it won't matter.)

- Type `ruby -v` to check that you're using 2.0.0.
- 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 2.0.0 to be the default Ruby, so if you have an old terminal hanging around it might stubbornly keep using Ruby 1.8.7.

All Systems Green: Let's 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, so you can follow the directions above in *Using Linux in the Department Undergraduate Labs*.
- 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/>.

General Information on Using the REPL (or not)

- 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 `^D` (that's *Control-D*). 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.

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 that is serving as your "main." 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.