WALKINGRANDOMLY

Because it's more fun than getting there in a straight line.

Home | About Me | Highlights | Contact | Testimonials

Type text to search here...

A portable Jupyter (IPython) install on Windows with an R kernel

April 28th, 2015 | Categories: Free software, python, R, Scientific Software, software deployment, Windows | Tags:

I recently found myself in need of a portable install of the Jupyter notebook which made use of a portable install of R as the compute kernel. When you work in institutions that have locked-down managed Windows desktops, such portable installs can be a life-saver! This is particularly true when you are working with rapidly developing projects such as Jupyter and IRKernel.

It's not perfect but it works for the fairly modest requirements I had for it. Here are the steps I took to get it working.

Download and install Portable Python

I downloaded Portable Python 2.7.6.1 from http://portablepython.com/ and installed into a directory called **Portable Python 2.7.6.1**

Update IPython and install the extra modules we need

This version of Portable Python comes with a portable IPython instance but it is too old to support alternative kernels. As such, we need to install a newer version.

Open a **cmd.exe** command prompt and navigate to **Portable Python 2.7.6.1\App\Scripts**.

Enter the command

```
easy_install ipython.exe
```

You'll now find that you can launch the ipython.exe terminal from within this directory:

```
C:\Users\walkingrandomly\Desktop\Portable Python 2.7.6.1\App\Scripts>ipython
Python 2.7.6 (default, Nov 10 2013, 19:24:18) [MSC v.1500 32 bit (Intel)]
Type "copyright", "credits" or "license" for more information.

IPython 3.1.0 -- An enhanced Interactive Python.
?         -> Introduction and overview of IPython's features.
%quickref -> Quick reference.
help      -> Python's own help system.
object?   -> Details about 'object', use 'object??' for extra details.

In [1]: exit()
```

If you try to launch the notebook, however, you'll get error messages. This is because we haven't taken care of all the dependencies. Let's do that now. Ensuring you are still in the **Portable Python 2.7.6.1\App\Scripts** folder, execute the following commands.

```
easy_install pyzmq
easy_install jinja2
easy_install tornado
easy_install jsonschema
```

You should now be able to launch the notebook using

```
ipython notebook
```

Install portable R and IRKernel

- I downloaded Portable R 3.2 from http://sourceforge.net/projects/rportable/files/ and installed into a directory called **R-Portable**
- Move this directory into the Portable Python directory. It needs to go inside **Portable Python 2.7.6.1\App** (see this discussion to learn how I discovered that this location was the correct one)
- Launch the Portable R executable which should be at **Portable Python 2.7.6.1\App\R-Portable\R-portable.exe** and install the IRKernel packages by doing

```
install.packages(c("rzmq","repr","IRkernel","IRdisplay"), repos="http://irkernel.github.io/")
```

Install additional R packages

The version of Portable R I used didn't include various necessary packages. Here's how I fixed that.

- Launch the Portable R executable which should be at **Portable Python 2.7.6.1\App\R-Portable\R-portable.exe** and install the following packages

```
install.packages('digest')
install.packages('uuid')
install.packages('base64enc')
install.packages('evaluate')
install.packages('jsonlite')
```

Install the R kernel file

Create the directory structure **Portable Python 2.7.6.1\App\share\jupyter\kernels\R_kernel**

Create a file called **kernel.json** that contains the following

```
{"argv": ["R-Portable/App/R-Portable/bin/i386/R.exe","-e","IRkernel::main()",
"--args","{connection_file}"],
 "display_name":"Portable R"
}
```

This file needs to go in the **R_kernel** directory created earlier. Note that the kernel location specified in kernel.json uses Linux style forward slashes in the path rather than the backslashes that Windows users are used to. I found that this was necessary for the kernel to work –it was ignored by the notebook otherwise.

Finishing off

Everything created so far, including R, is in the folder **Portable Python 2.7.6**

I created a folder called **PortableJupyter** and put the **Portable Python 2.7.6** folder inside it. I also created the folder **PortableJupyter\notebooks** to allow me to carry my notebooks around with the software that runs them.

There is a bug in Portable Python 2.7.6.1 relating to scripts like IPython.exe that have been installed using easy_install. In short, they stop working if you move the directory they're installed in – breaking portability somewhat! (Details here)

The workaround is to launch Ipython by running the script **Portable Python 2.7.6.1\App\Scripts\ipython-script.py**

I didn't want to bother with that so created a shortcut in my **PortableJupyter** folder called **Launch notebook.** The target of this shortcut was the following line

```
%windir%\system32\cmd.exe /c "cd notebooks && "%CD%/Portable Python 2.7.6.1/App\python.exe" "%CD%/Portab
le Python 2.7.6.1\App\Scripts\ipython-script.py" notebook"
```
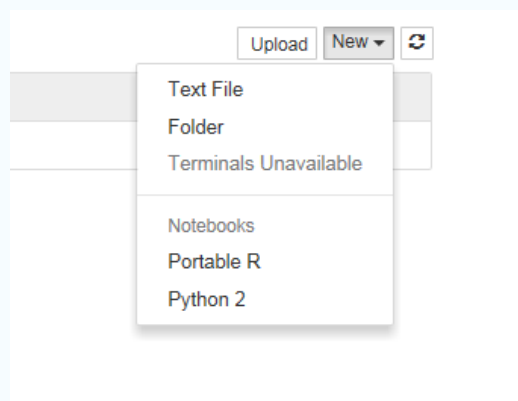
This starts the notebook using the default web browser and puts you in the notebooks directory.

The pay off

My folder looks like this:
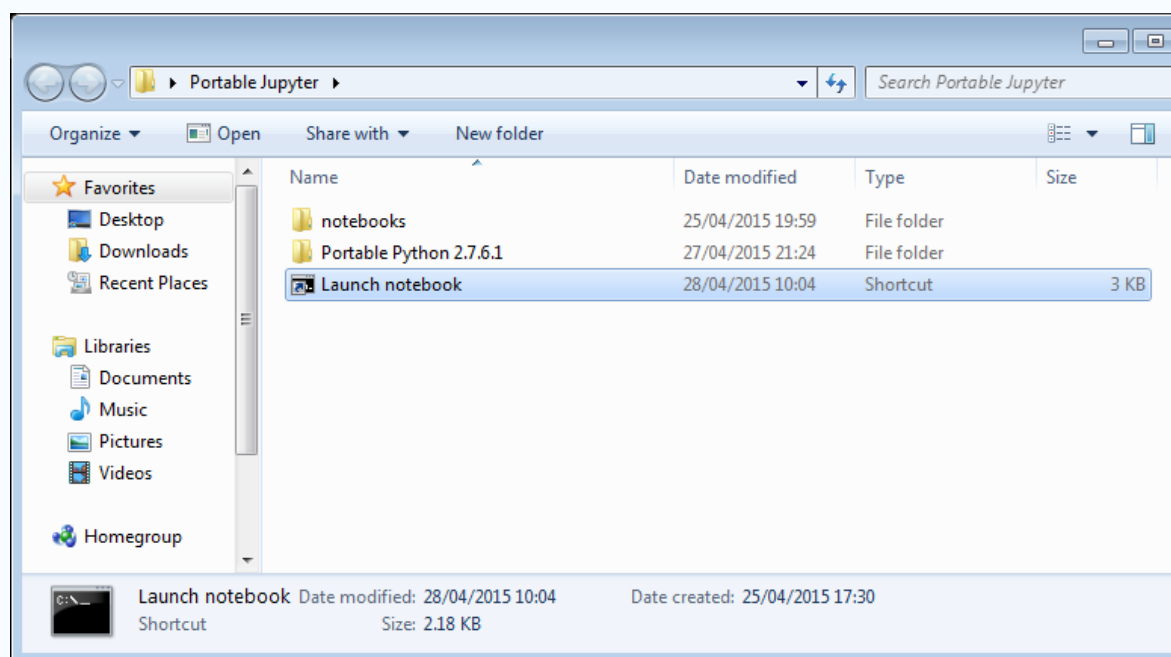
If I click on the Launch Notebook shortcut, I get a Jupyter session with 2 kernel options

I can choose the Portable R kernel and start using R in the notebook!

```
In [1]: mtcars
```

Out[1]:

	mpg	cyl	disp	hp	drat	wt	qsec	vs	am	gear	carb
Mazda RX4	21	6	160	110	3.9	2.62	16.46	0	1	4	4
Mazda RX4 Wag	21	6	160	110	3.9	2.875	17.02	0	1	4	4
Datsun 710	22.8	4	108	93	3.85	2.32	18.61	1	1	4	1
Hornet 4 Drive	21.4	6	258	110	3.08	3.215	19.44	1	0	3	1
Hornet Sportabout	18.7	8	360	175	3.15	3.44	17.02	0	0	3	2
Valiant	18.1	6	225	105	2.76	3.46	20.22	1	0	3	1
Duster 360	14.3	8	360	245	3.21	3.57	15.84	0	0	3	4
Merc 240D	24.4	4	146.7	62	3.69	3.19	20	1	0	4	2
Merc 230	22.8	4	140.8	95	3.92	3.15	22.9	1	0	4	2
Merc 280	19.2	6	167.6	123	3.92	3.44	18.3	1	0	4	4

Share this:

Leave a comment | Trackback

Will Furnass
April 28th, 2015 at 15:56

Reply | Quote | #1

Nice. Did you know that the conda package repository, binstar, now has an R channel? You should be able to most of the above with the following after installing miniconda and setting your path:

conda install -c r ipython-notebook r-irkernel

Mike Croucher
April 28th, 2015 at 16:17

Reply | Quote | #2

That's fantastic! Can I install completely portable versions of Python/R using conda? I.e. ones that don't need admin rights on Windows to install and don't leave any residue behind? That was the driving factor in going down this route.

For integrating with our managed desktop 'properly', however, I intend to use Anaconda. The fact that I can add R-support so easily now that you've given me the above information makes it even easier :)

Will Furnass
April 28th, 2015 at 16:27

Reply | Quote | #3

Yes if you're using Linux; haven't tried doing anything similar on Windows but imagine it would work.

A follow-up post on setting up jupyterhub for students at the University of Sheffield would be interesting. There have been several posts on the ipython-dev list in recent months on people using jupyter for different courses; I'm sure the ipython-dev team would be interested in hearing about anything you set up for students. FYI Jessica Hamrick's post on the Rackspace dev blog gives a particularly good overview of the current usability of the project for this purpose.

Mike Croucher
April 28th, 2015 at 16:39

Reply | Quote | #4

Windows is vital for teaching – at Sheffield at least.

I'll use the portable install created above to get us through the next two weeks which we are using as a dry run and pre-requisite set of sessions for a Bioinformatics module which will be delivered in October.

I'll work with the managed desktop team at Sheffield to get the Anaconda solution 'packaged' by then — kernels in Python 2, Python 3 and R to start with – all running on the Windows desktop.

Longer term, I'd like to do something more interesting with this technology and the blogpost you mentioned (this one, yes? https://developer.rackspace.com/blog/how-did-we-serve-more-than-20000-ipython-notebooks-for-nature/) looks like the kind of thing that would be fun to do :)

Will Furnass
April 28th, 2015 at 17:25

Reply | Quote | #5

Nope, it was https://developer.rackspace.com/blog/deploying-jupyterhub-for-education/. The jupyterhub demo for Nature was concerned with providing Notebooks to anonymous users whereas Jessica was more interested in students being able to authenticate against the notebook server, complete Notebook-based coursework then the tutor grade submissions manually or automatically using a unit testing approach (see https://github.com/jupyter/nbgrader).

Are there plans to introduce researchers to Jupyter using something like Software Carpentry workshops? This would be awesome, particularly if researchers could 'try before they buy' by being able to participate in the course using their own machines without having to install anything locally.

Mike Croucher
April 28th, 2015 at 18:41

There are plans but so far just a twinkle in my eye. I hope to have more time to devote to it soon though.

umair durrani
May 10th, 2015 at 05:46

@Will Furnass
This is awesome!

Rohit
September 4th, 2015 at 21:19

Thank Will Furnass. The tip is awesome!

Name

E-Mail (will not be published)

Website

Submit Comment

🔲 Subscribe to comments feed

☐ Notify me of follow-up comments by email.
☐ Notify me of new posts by email.

↑ TOP

Sidebar:

Blogroll

Annoying Precision

Dead Reckonings

God Plays dice

Let's play math

Math Notations

Math Puzzles

Nick Higham (Applied math)

PleaseMakeANote

Quantitative Finance Collector

Terminally Incoherent

The Endeavour

The Universe of Discourse

Meta

Log in

Entries RSS

Comments RSS

WordPress.org