

# **JUPYTER / ROOT / GALLERY INTEGRATION**

**Wesley Ketchum (FNAL)**

# EXAMPLE REPOSITORY

- [https://github.com/wesketchum/Sample\\_Notebooks](https://github.com/wesketchum/Sample_Notebooks)
- Contains basically everything I'll mention here today
  - *Startup readme:*  
[https://github.com/wesketchum/Sample\\_Notebooks/blob/master/README.md](https://github.com/wesketchum/Sample_Notebooks/blob/master/README.md)
  - *Four notebook examples:*
    - Two simple ROOT examples
    - Two using gallery/larsoft utils

# JUPYTER NOTEBOOKS

- <http://jupyter.org/>
- [http://jupyter-notebook-beginner-guide.readthedocs.io/en/latest/what\\_is\\_jupyter.html](http://jupyter-notebook-beginner-guide.readthedocs.io/en/latest/what_is_jupyter.html)
- Basically: an executable document
  - *Combination of documentation, code, and results*
  - *Generally interactive/allows for quick re-execution of code snippets*
- Default execution kernel: python

# HOW TO INSTALL JUPYTER?

- Generally:
  - *pip install jupyter*
- But ... what if you don't have write access to the python installation area?
  - *Typically true for gpvms, or if you are otherwise using cvmfs*
- It's easy to setup a user python area to install personal products into
  - ***You should make sure you have the version of python you want setup first: likely the one you get with larsoft***

# PYTHON USER AREA

## **#setup**

```
export PYTHONUSERBASE=/$EXPERIMENT/app/users/$USER/python_libs
mkdir -p $PYTHONUSERBASE
export PYTHONPATH=$PYTHONUSERBASE/bin:$PYTHONPATH
export PATH=$PYTHONUSERBASE/bin:$PATH
```

## **#install**

```
pip install --user jupyter metakernel zmq
```

# ROOT NOTEBOOKS/KERNELS

- ROOT v6.05 and greater ships with support for Jupyter
  - *“magic” keywords to make C++ snippets inside a python kernel*
  - *a full C++ kernel*
- ROOT tutorials all exist in notebook form online
  - [https://root.cern/doc/master/group\\_Tutorials.html](https://root.cern/doc/master/group_Tutorials.html)
- This also makes it possible and easy\* to execute gallery code directly inside a Jupyter notebook
  - *\*well, with a few tricks I'll show you*

# QUICK GALLERY SIDEBAR

- <http://art.fnal.gov/gallery/>
- What gallery is
  - *Lightweight access to event data in art/ROOT files, outside the art framework*
- What gallery is not
  - *An alternative full-featured framework*
- Options for compiled C++, ROOT macros, and PyROOT
  - *Demos/tutorials available at website*

# STARTING A ROOT NOTEBOOK

- After following previously mentioned setup, do:

```
root --notebook
```

- Notebooks want to run in browsers
  - *Notebook starts a local web server on localhost:8888 or another available port*
  - *Above command will attempt to launch a browser to connect to that port*
  - *Can alternatively (preferably if running on gpvms, e.g.) use an ssh tunnel and port forwarding to open in local browser*
    - I believe this is secure...
- See:
  - [https://github.com/wesketchum/Sample\\_Notebooks#starting-the-root-notebook](https://github.com/wesketchum/Sample_Notebooks#starting-the-root-notebook)



# ROOT NOTEBOOK EXAMPLES

- Basics python-based kernel with C++ magics
  - [http://nbviewer.jupyter.org/github/wesketchum/Sample\\_Notebooks/blob/efc8dc14d0807d183cc42c390027d3c9fb8bd/a46/ROOT\\_Example.ipynb](http://nbviewer.jupyter.org/github/wesketchum/Sample_Notebooks/blob/efc8dc14d0807d183cc42c390027d3c9fb8bd/a46/ROOT_Example.ipynb)
- Basic C++-based kernel
  - [http://nbviewer.jupyter.org/github/wesketchum/Sample\\_Notebooks/blob/efc8dc14d0807d183cc42c390027d3c9fb8bd/a46/ROOT\\_CPP\\_Kernel\\_Example.ipynb](http://nbviewer.jupyter.org/github/wesketchum/Sample_Notebooks/blob/efc8dc14d0807d183cc42c390027d3c9fb8bd/a46/ROOT_CPP_Kernel_Example.ipynb)

# USING GALLERY IN A NOTEBOOK

- **Why?**

- Why not?
- At its simplest: a way to structure a gallery macro interactively, more deliberately, and with a nice user interface
- But, yeah, it's not mind-shatteringly magical
  - *You may prefer to use macros/C++*

# EXAMPLE: READING MCTRUTH

- [http://nbviewer.ipython.org/github/wesketchum/Sample Notebooks/blob/efc8dc14d0807d183cc42c390027d3c9fb8bda46/Analyze\\_MCTruth.ipynb](http://nbviewer.ipython.org/github/wesketchum/Sample%20Notebooks/blob/efc8dc14d0807d183cc42c390027d3c9fb8bda46/Analyze_MCTruth.ipynb)
- Suggestions:
  - *Put common/typical headers in a separate header file and compile it*
  - *Put your variable declarations in one cell, and (re)initialize them in another*
  - *Use the “Kernal > Restart & Run All” utility as needed*
- Nothing special: everything is as you do in gallery
  - *e.g. with compiling the right headers, art::Assns “just work”*

# EXAMPLE: SPACECHARGE

- No services in gallery, but you can still create objects typically held/delivered by a service
  - *Space charge in MicroBooNE example:*  
[http://nbviewer.jupyter.org/github/wesketchum/Sample\\_Notebooks/blob/efc8dc14d0807d183cc42c390027d3c9fb8bda46/SpaceCharge\\_Example.ipynb](http://nbviewer.jupyter.org/github/wesketchum/Sample_Notebooks/blob/efc8dc14d0807d183cc42c390027d3c9fb8bda46/SpaceCharge_Example.ipynb)
- Suggestions
  - *Make fhicl configs from strings*
  - *Link against existing shared libraries as needed*
- Easy to extend this to other things: e.g. I wrote a “swizzler” to convert ICARUS TPC artdaq::Fragments to raw::RawDigits in a notebook

# GENERAL SUGGESTIONS

- Play around and see what works best for you
- Speed is still best with a compiled C++ program, but still...
  - *You can use notebooks as proving ground for that*
  - *You can compile functions inside the notebook*
  - *You can link against externally built libraries*
- Python notebooks work best if you want to utilize other python utilities
  - *Install what's needed in pip!*
- Lots of documentation for Jupyter online: use it!

# To LARSOFT/SCIISOFT PROJECT

- This appears to be a potentially valuable workflow/set of utilities
  - *It'd be good to make sure we retain capability to do this: have a python that can be used with jupyter/metakernel/zmq*
- This could be a great way to document tutorials/workflows/instruction sets
  - *There are bash kernels too, for example: could imagine a fully integrated setup/run/analyze demonstration in a notebook*