Contributing to the Fermitools

A guide for collaborators and the public

Contents

Fermitools is the new public Sciencetools

Sciencetools

- 2 sets of CVS repos: FSSC & SLAC
- 2 build mgmt. tools
  (SCons, hmake)
- Manual download & compilation
- Manual dependency mgmt.
- Manual version mgmt.
- Manual testing, ver. & val.
- 1 release per year (optimistically)

Fermitools

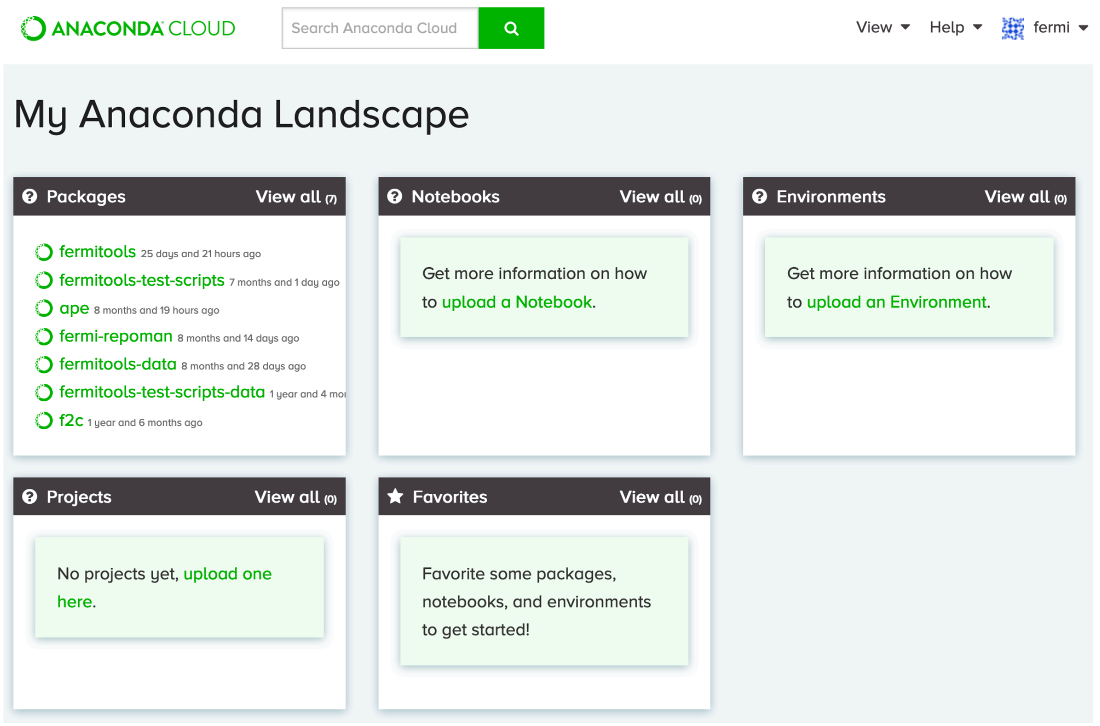
- 1 shared Github organization
- 2 build mgmt. tools
  (SCons, ~conda-build~)
- Conda Package Manager
- Conda dependency mgmt.
- Conda version mgmt.
- Azure Pipeline CI testing
- 1 release per 2 months

New Words

- Conda
- Github
  - Special bonus Slide!!!
    Submitting Bugs and Issues!!!
- Azure Pipelines
- Repoman

Conda

- https://anaconda.org/fermi/dashboard
- Cross-Platform & Cross-Language Package Manager
- Universal Linux Binaries
- Easier dependency management
- Easier package distribution
- Easier version management
- Easier conflict separation (Environments)
  - 2 conflicting versions of the tools can be installed in different environments.
- Harder Development



My Anaconda Landscape

Packages — View all (7)

- fermitools 25 days and 21 hours ago
- fermitools-test-scripts 7 months and 1 day ago
- ape 8 months and 19 hours ago
- fermi-repoman 8 months and 14 days ago
- fermitools-data 8 months and 28 days ago
- fermitools-test-scripts-data 1 year and 4 mo
- f2c 1 year and 6 months ago

Notebooks — View all (0)

Get more information on how to upload a Notebook.

Environments — View all (0)

Get more information on how to upload an Environment.

Projects — View all (0)

No projects yet, upload one here.

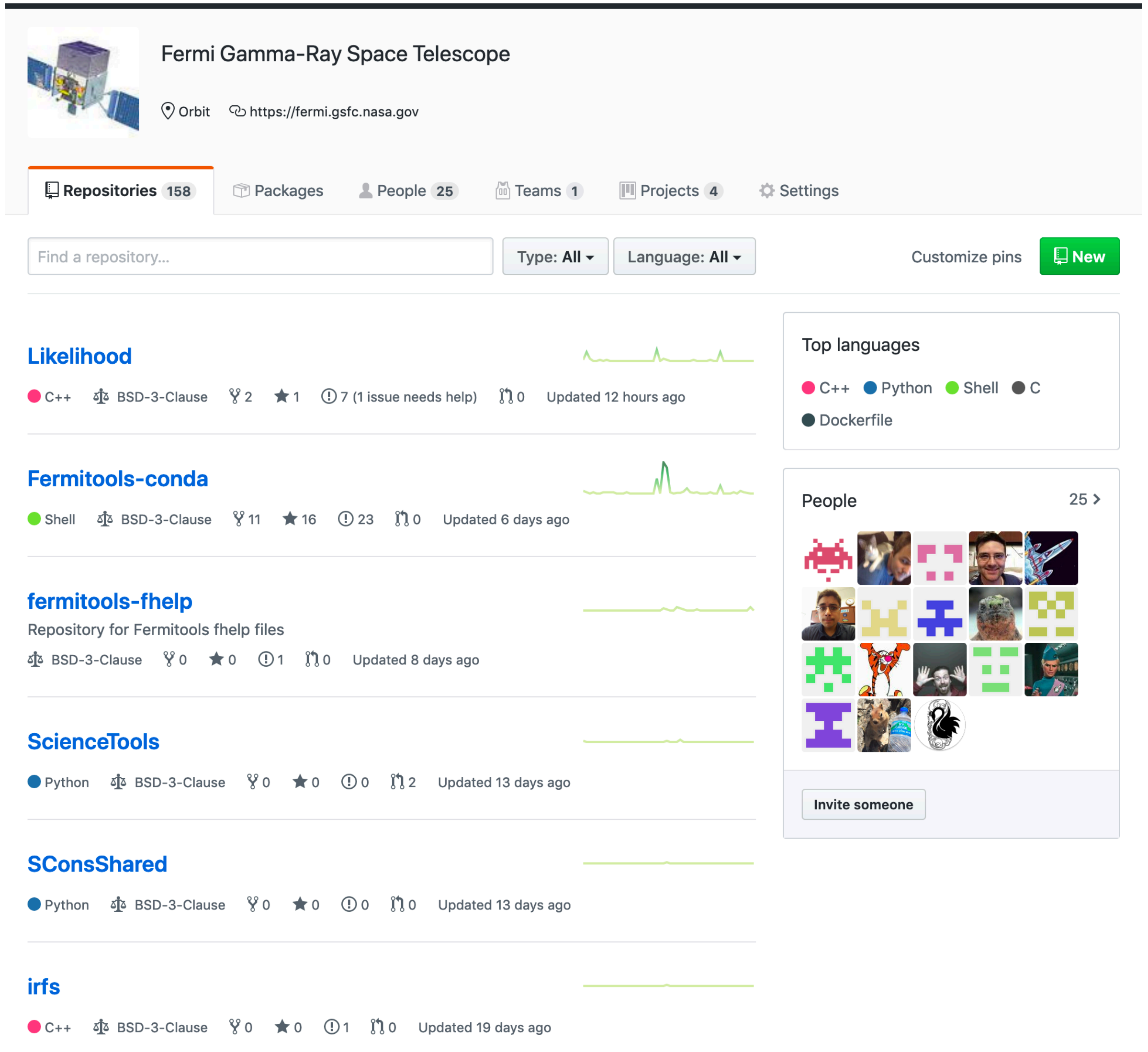
Favorites — View all (0)

Favorite some packages, notebooks, and environments to get started!

Github

- https://github.com/fermi-lat

- Unified Fermi LAT collaboration collection of package repositories
- 158 repositories (And Growing!)
- Git Version Control System
  - Branches!
  - Tags!
- Wiki documentation
- Bug + Issue tracking



Fermi Gamma-Ray Space Telescope

Orbit https://fermi.gsfc.nasa.gov

Repositories 158 | Packages | People 25 | Teams 1 | Projects 4 | Settings

Find a repository... | Type: All | Language: All | Customize pins | New

Likelihood
● C++ BSD-3-Clause 2 ★ 1 7 (1 issue needs help) 0 Updated 12 hours ago

Fermitools-conda
● Shell BSD-3-Clause 11 ★ 16 23 0 Updated 6 days ago

fermitools-fhelp
Repository for Fermitools fhelp files
BSD-3-Clause 0 ★ 0 1 0 Updated 8 days ago

ScienceTools
● Python BSD-3-Clause 0 ★ 0 0 2 Updated 13 days ago

SConsShared
● Python BSD-3-Clause 0 ★ 0 0 0 Updated 13 days ago

irfs
● C++ BSD-3-Clause 0 ★ 0 1 0 Updated 19 days ago

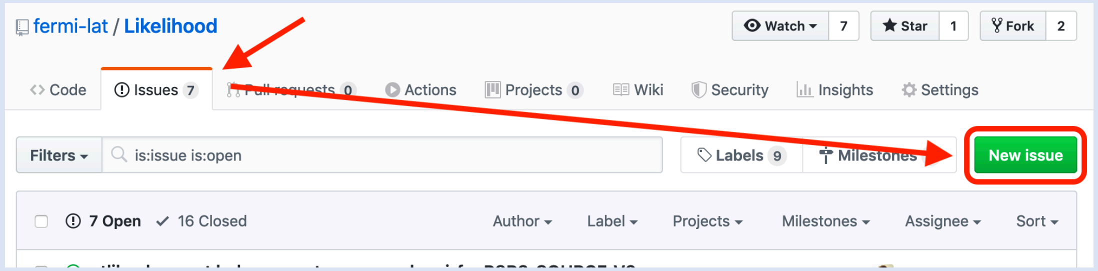
Top languages
● C++ ● Python ● Shell ● C ● Dockerfile

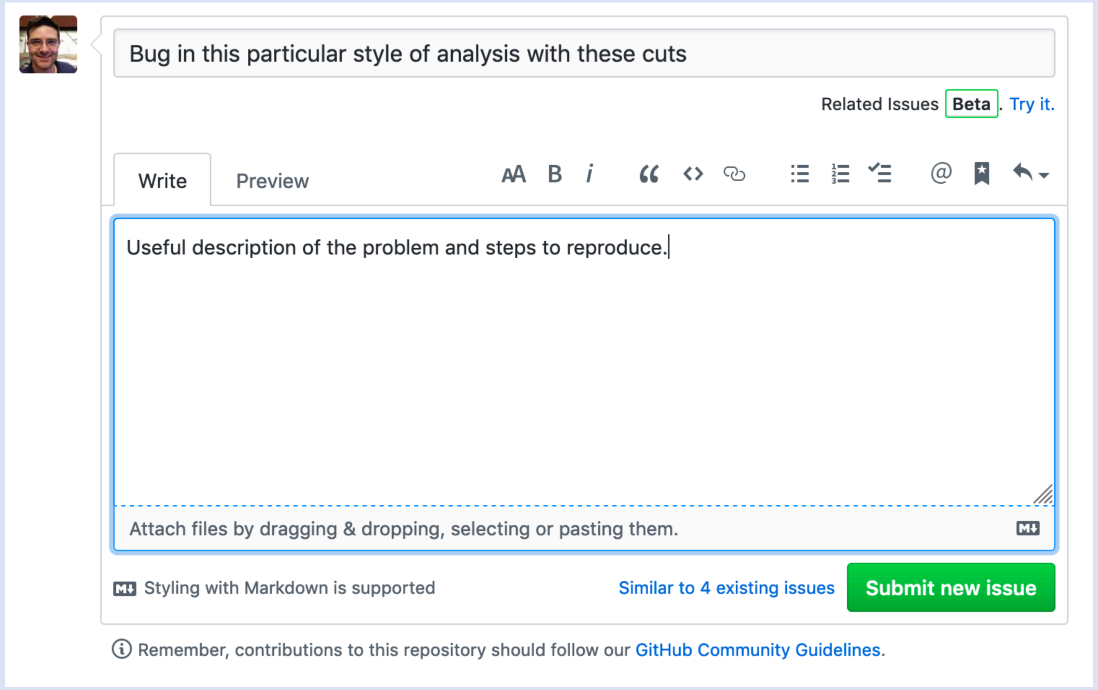
People 25

Invite someone

Submitting Issues on Github

- If bugs are found please submit them on the Github issue tracker.
- It’s open to the public!
- Find the affected package repo.
- Click the “Issues” tab.
- Click “New Issue”.
- Write a useful description, preferably with steps to reproduce.
- Submit and engage in follow up discussion until resolved.




Bug in this particular style of analysis with these cuts

Related Issues Beta. Try it.
Write
Preview

Useful description of the problem and steps to reproduce.

Attach files by dragging & dropping, selecting or pasting them.

Styling with Markdown is supported
Similar to 4 existing issues
Submit new issue

Remember, contributions to this repository should follow our GitHub Community Guidelines.

Azure Pipelines (Continuous Integration)

- https://dev.azure.com/FermiSpaceTelescope/Fermitools/
- Cloud-based build and test platform
- MacOS and Linux support
- Automatic Builds of new tools
- Automatic Testing
- Automatic Versioning
- Automatic upload to Conda cloud storage (with tagging)
- Triggered automatically by:
  - GitHub commits
  - Pull Requests
  - Merge Requests to the master branch

Repoman

- Tool to interact with multiple Github repositories all at once
- Thank you Brian Van Klaveren
- Specify target repositories, branches, tags, commits, etc.
- Get “master” from 1 project and “dev” from another all with 1 command.
- conda install -c fermi fermi-repoman

Development workflow

- Prep environment
- Obtain source code
- Developers! Developers! Developers!
- Build with SCons
- Build with conda-build

https://github.com/fermi-lat/Fermitools-conda/wiki/Contributing-to-the-Fermitools

Fermitools Development Workflow

https://github.com/fermi-lat/Fermitools-conda/wiki/Contributing-to-the-Fermitools

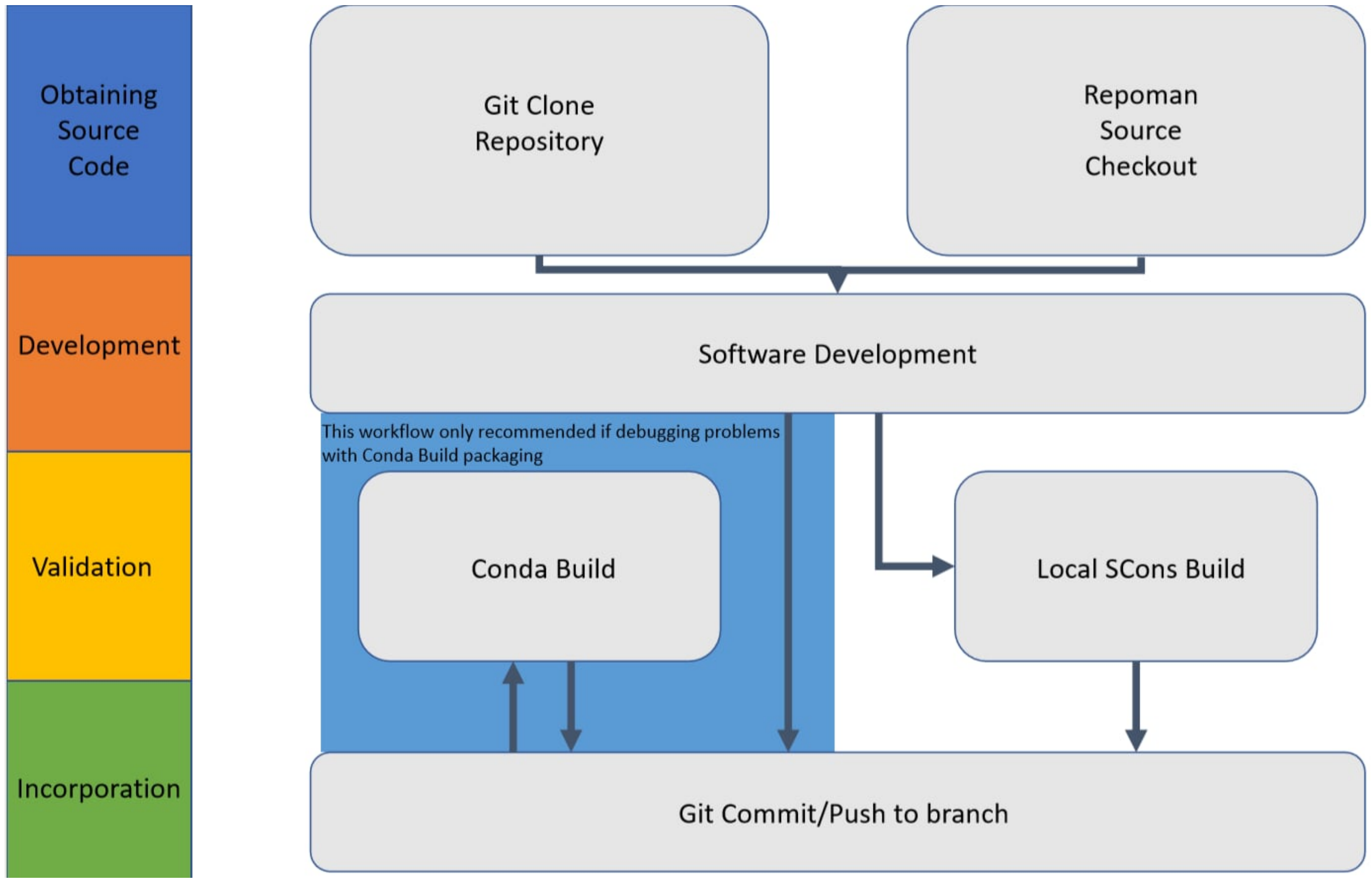


Image courtesy of Joe Asercion

Prep: Setup and Needed tools

- OS: Linux or Mac (WSL “supported” on Windows)
- Shell: Bash or Zsh (sorry (t)csh. It’s for the best)
- Install miniconda
  - https://docs.conda.io/en/latest/miniconda.html
  - Prefer the script installers, *.sh.   no root or sudo required
  - Python2 or Python3. It Doesn’t matter.
- Make sure you can activate / deactivate a base environment.
  - conda activate
  - conda deactivate

Step 0: Obtain external dependencies (and prep a workspace)

- For local development using SCons you need the 3rd party external dependencies available in an environment you can call.
- We will create a conda environment named “dev” and install the runtime Fermitools dependencies into it.
- conda create --name dev --only-deps -c conda-forge/label/cf201901 -c fermi fermitools --yes
- All runtime dependencies saved in ${CONDA_ENV_PREFIX}/envs/dev/

Step 0: Obtain dependencies (and prep a workspace)

• conda create	• Create new environment
--name dev	• Named ”dev”
--only-deps	• Install runtime dependencies
-c conda-forge/label/cf201901	• Use channel conda-forge with the cf201901 label (For older compilers)
-c fermi	• Use channel fermi
fermitools	• The target
-y	• Optionally say yes to all prompts

Step 0: Obtain dependencies (and prep a workspace)

- Activate the “dev” environment with the new dependencies.
  - conda activate dev
- These are the runtime dependencies, which differ from the build dependencies and don’t include libraries like Scons, SWIG, repoman…
- Install any other build dependencies you require:
  - conda install –c conda-forge –c fermi scons=3.03 swig fermi-repoman
- All official build and run dependencies are listed in https://github.com/fermi-lat/Fermitools-conda/blob/master/meta.yaml

Fermitools Development Workflow

https://github.com/fermi-lat/Fermitools-conda/wiki/Contributing-to-the-Fermitools

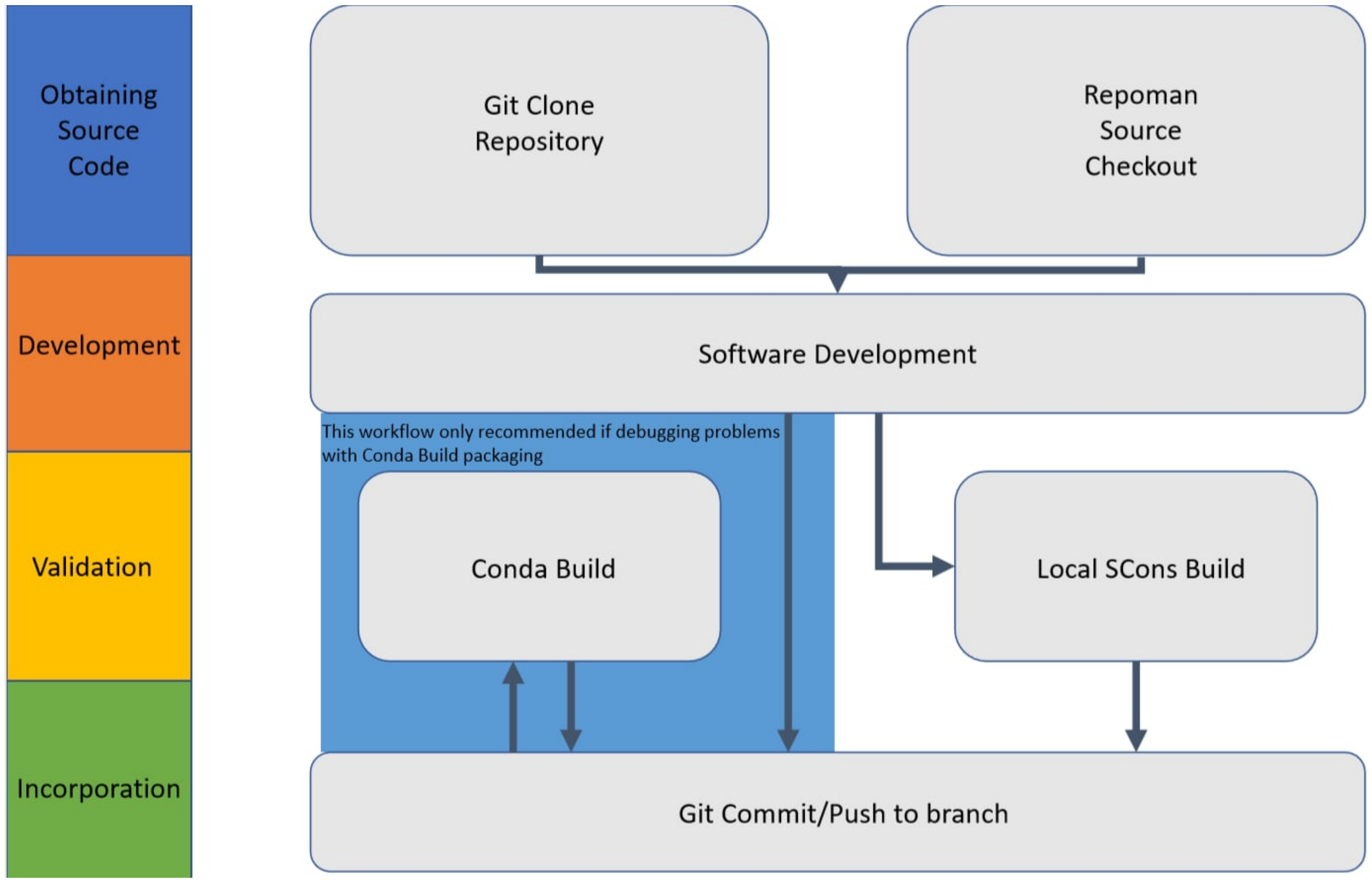


Image courtesy of Joe Asercion

Step 1: Obtain Source Code

- Make a development folder and move into it.
- Developers with github accounts + ssh keys set up:
  - repoman checkout --develop ScienceTools conda
- Developers without github accounts + ssh keys set up:
  - repoman --remote-base https://github.com/fermi-lat checkout --develop ScienceTools conda
- Repoman downloads all the fermitools packages listed in Sciencetools/packageList.txt
- Branches tags and commits can be listed at the end of the repoman call to get different variants of source code.
  - Closest to tail has highest priority.
- External contributors should fork the repositories they intend to work on and repoint their remote-base at their own username.

Step 1: Obtain Source Code

```
repoman --remote-base https://github.com/fermi-lat

checkout
--develop


ScienceTools

conda  tagB  branch1  tagA
```

- Invoke Repoman and target this organization for remotes
- Checkout remote repos
- Default to the master branch. Otherwise use tags in packageList.txt
- Location of packageList.txt
- Space-separated list of branches, tags etc. to prefer over master.

Fermitools Development Workflow

https://github.com/fermi-lat/Fermitools-conda/wiki/Contributing-to-the-Fermitools

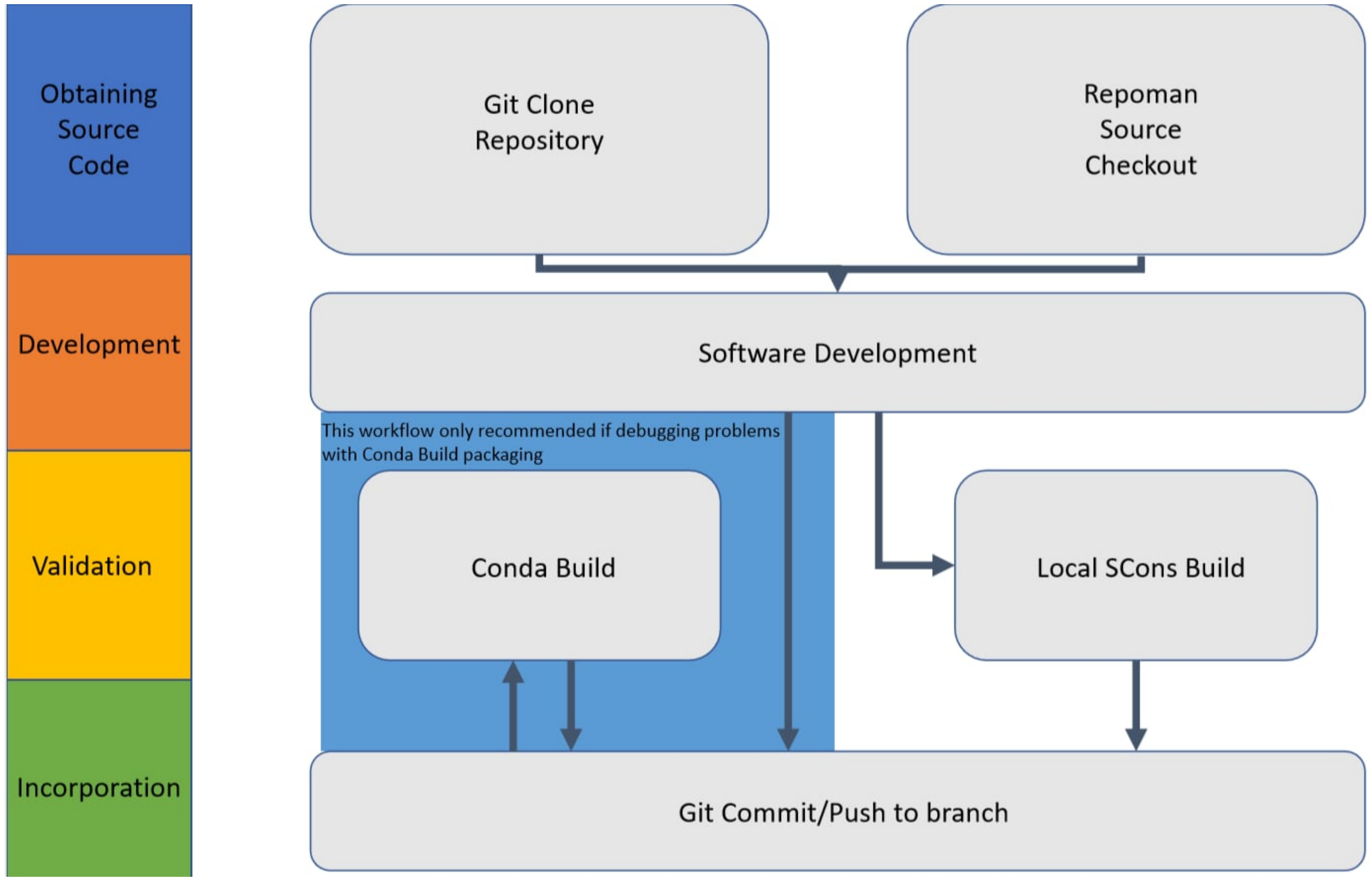


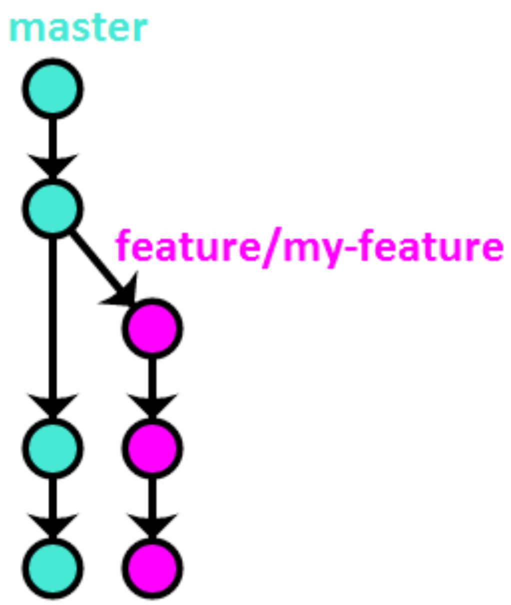
Image courtesy of Joe Asercion

Step 2: Develop the software

- Left as an exercise to the reader.

Git branching model

- Do Not develop on master
  - Development must occur on branches
- Create new git branch and go to it
  - git checkout -b mybranch
- Share branch with the fermi-lat github organization
  - git push -u origin mybranch
  - git push
- Prefer to branch from the head of master unless you have a good reason not to
- www.endoflineblog.com/oneflow-a-git-branching-model-and-workflow



Fermitools Development Workflow

https://github.com/fermi-lat/Fermitools-conda/wiki/Contributing-to-the-Fermitools

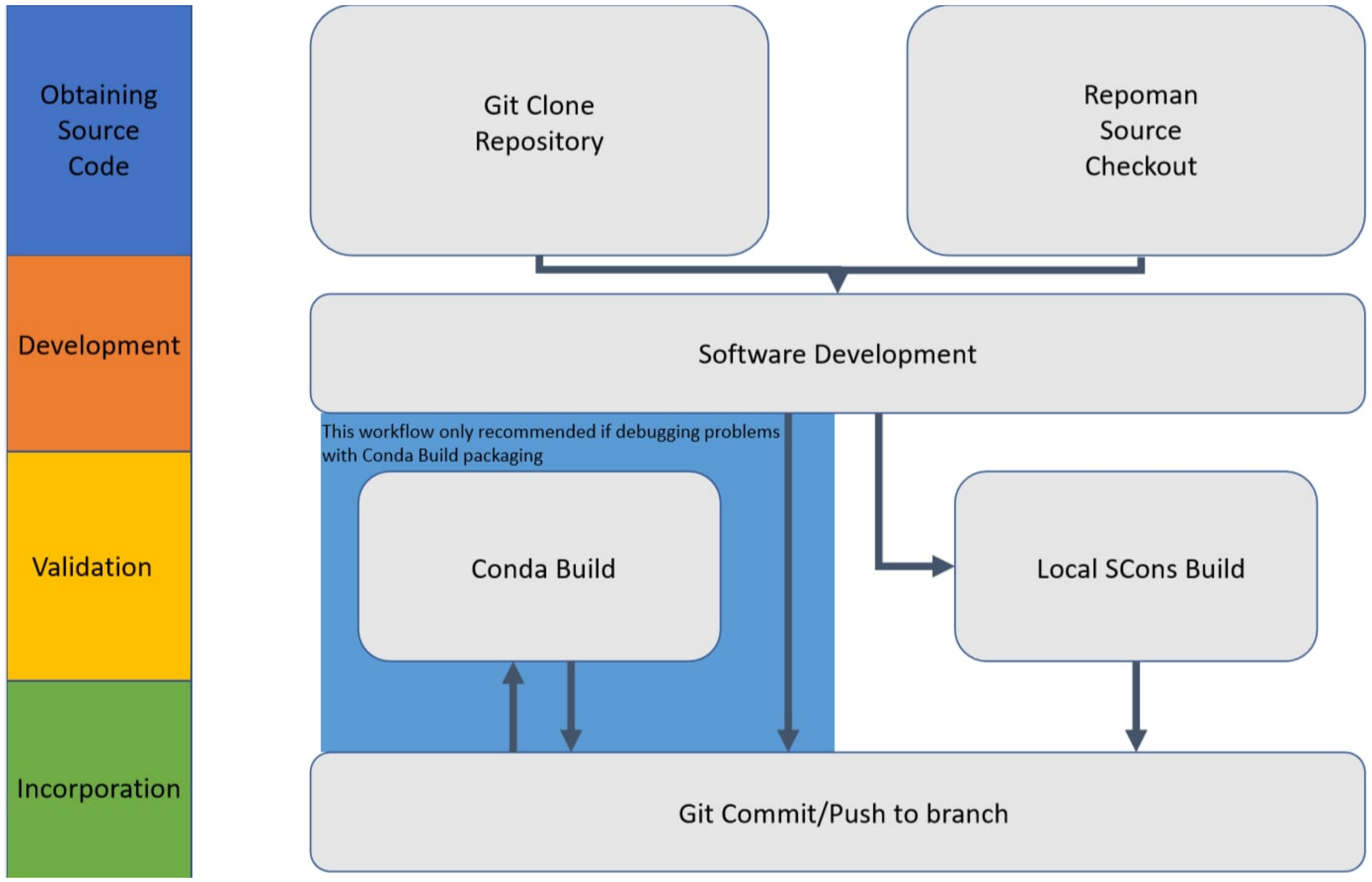


Image courtesy of Joe Asercion

Step 3: Build the Software (SCONS)

- Build locally using SCons
- Very similar to the old methodology.
- replace $GLAST_EXT with $CONDA_ENV_PREFIX
- scons -C ScienceTools --site-dir=../SConsShared/site_scons --conda=$CONDA_ENV_PREFIX --use-path all
- For more information consult: https://www.slac.stanford.edu/exp/glast/wb/prod/pages/a_IA_II_instrAnalysis/IA_II_SCons/scons_forDevelopers.html

Step 3: Build the Software (SCONS)

- Now test locally as you would normally and ensure the problem is solved.
- Take a well deserved break.
- Commit the software on a branch.
    - Optionally tag it.
- If changes span 1 or few repositories: git push
- If changes span many repositories: repoman release

Step 3: Build the Software (conda-build)

- Conda-build is more complex, time-consuming and annoying than the normal development process.
- All that annoyance stems from the fact that conda-build is just trying to be super helpful.
- Conda-build is your friend. :D
- With conda-build you do not need to manually install dependencies.
- With conda-build you do not need to manually clone repositories.
- With conda-build you do not need to manage your environments.
- Being helpful is s.l.o.w.

Step 3: Build the Software (conda-build)

- Docs and details for conda-build: https://docs.conda.io/projects/conda-build/en/latest/
- Mostly it's annoying because it's under active development. The standard is something of a moving target.
  - Lots of community buy-in is a good thing.
- conda install -c conda-forge conda-build
- All the conda-specific package and build info is in https://github.com/fermi-lat/Fermitools-conda

Important Conda files in Fermitools-conda

- meta.yaml -- Metadata, dependencies, version numbers.

- build.sh -- Script which calls repoman, sets environment variables, runs SCons build, runs simple build tests.

- activate.sh -- Sets up fermi environment upon conda activate fermi.
- Deactivate.sh -- Unsets fermi environment upon conda activate fermi.

- azure-pipelines.yml -- configuration information for CI pipeline.

Step 3: Build the Software (conda-build)

- https://docs.conda.io/projects/conda-build/en/latest/
- conda install -c conda-forge conda-build
- git clone https://github.com/fermi-lat/Fermitools-conda.git
- conda build --python=2.7 -c conda-forge/label/cf201901 -c fermi Fermitools-conda

Fermitools Development Workflow

https://github.com/fermi-lat/Fermitools-conda/wiki/Contributing-to-the-Fermitools

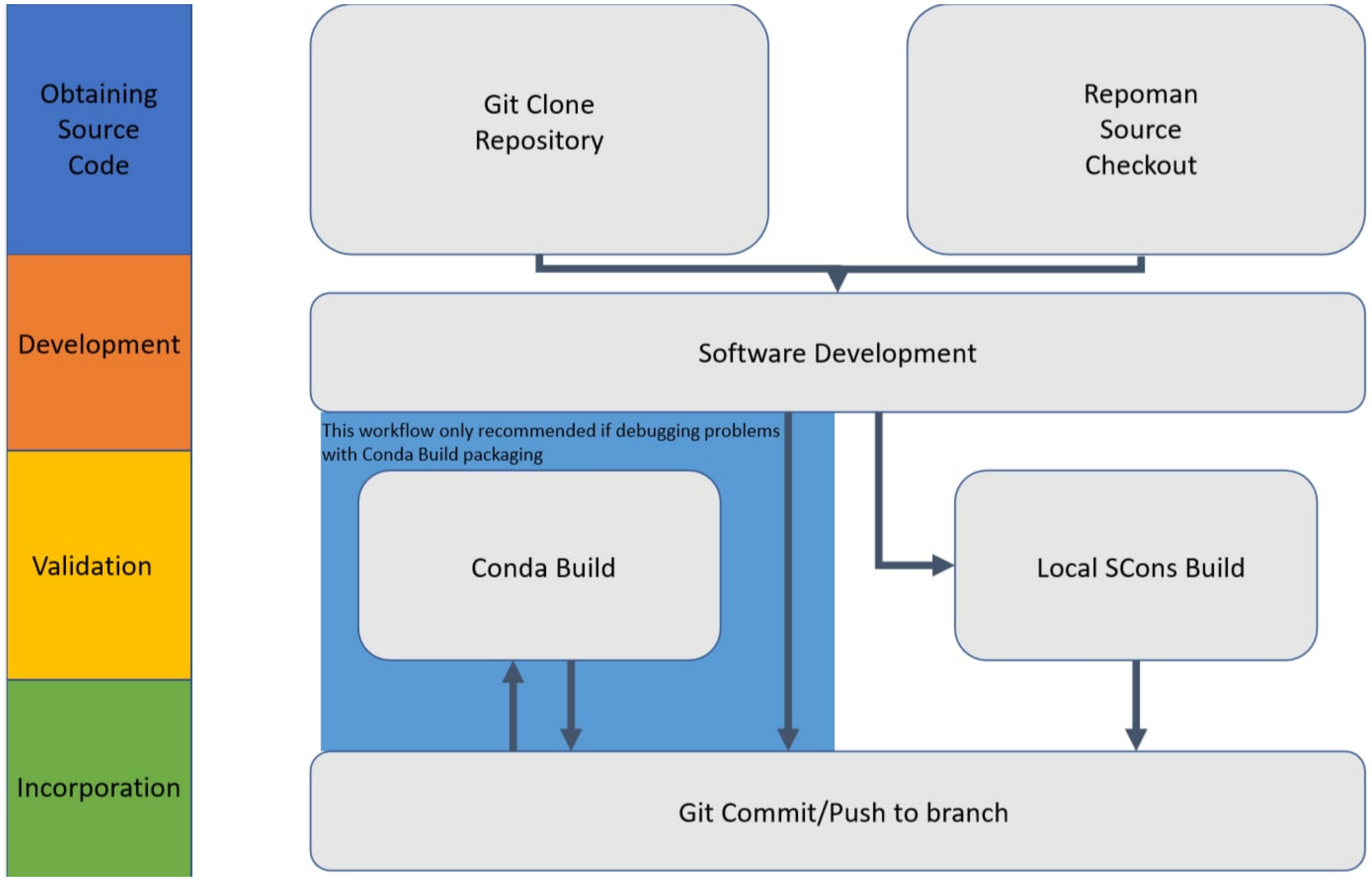
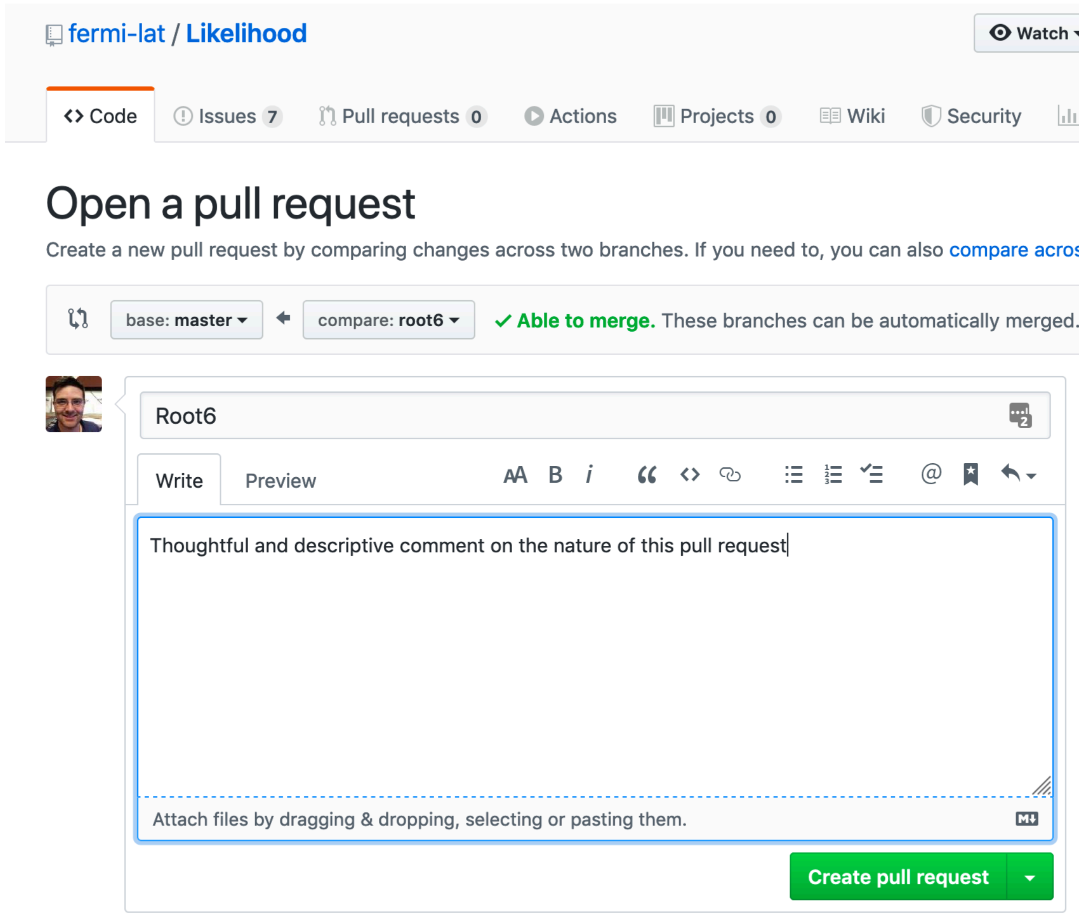


Image courtesy of Joe Asercion

Step 4: Incorporation

- Open a GitHub pull request to have your change incorporated into the next release.
- Enjoy the easy to use github comment system and our helpful CI integration.
- Respond courteously to fellow developers when new problems arise.
- Await the next release.



Open a pull request

Create a new pull request by comparing changes across two branches. If you need to, you can also compare acros

✓ Able to merge. These branches can be automatically merged.

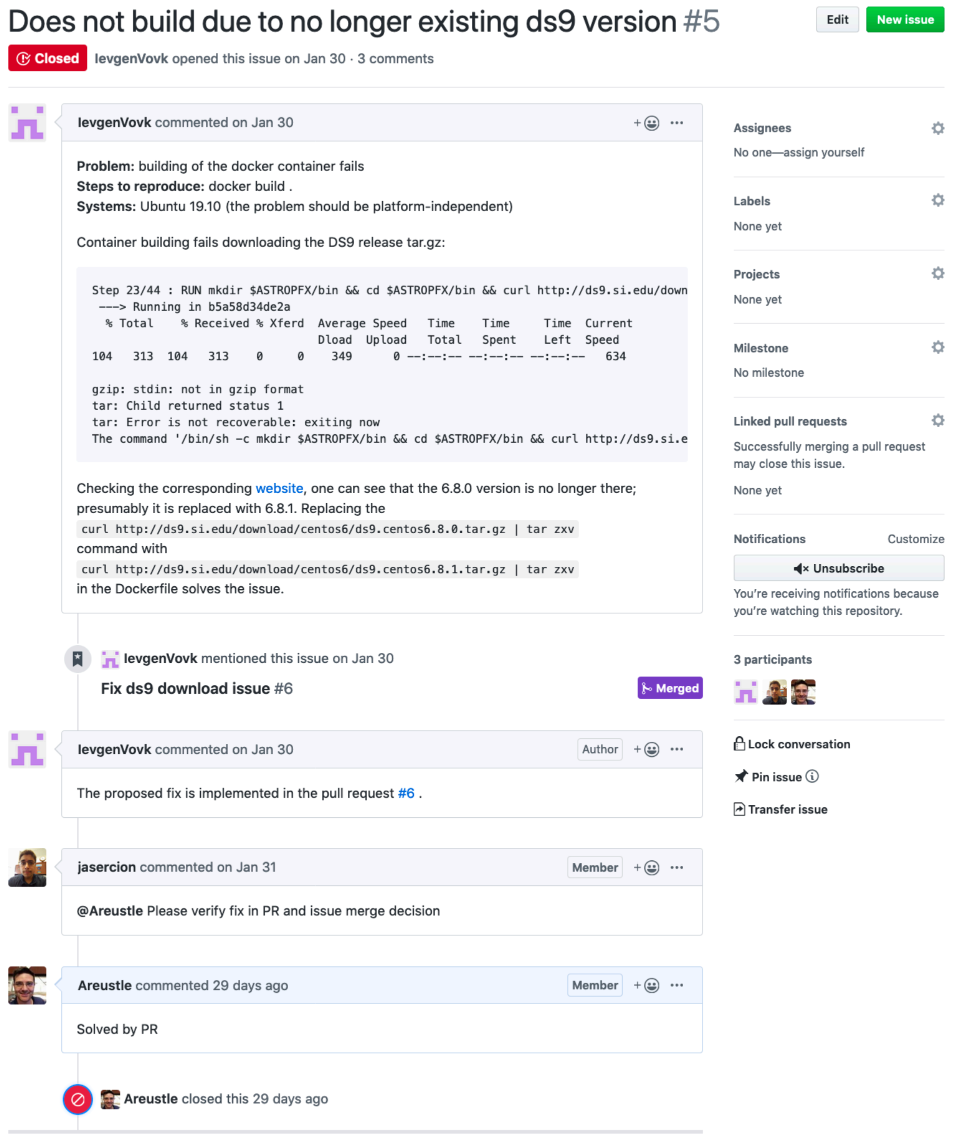
Root6

Thoughtful and descriptive comment on the nature of this pull request

Attach files by dragging & dropping, selecting or pasting them.

Step 4: Incorporation

- Open a GitHub pull request to have your change incorporated into the next release.
- Enjoy the easy to use github comment system and our helpful CI integration.
- Respond courteously to fellow developers when new problems arise.
- Await the next release.



Does not build due to no longer existing ds9 version #5

Edit · New issue

Closed · IevgenVovk opened this issue on Jan 30 · 3 comments

IevgenVovk commented on Jan 30

Problem: building of the docker container fails
Steps to reproduce: docker build .
Systems: Ubuntu 19.10 (the problem should be platform-independent)

Container building fails downloading the DS9 release tar.gz:

```
Step 23/44 : RUN mkdir $ASTROPFX/bin && cd $ASTROPFX/bin && curl http://ds9.si.edu/down
 ---> Running in b5a58d34de2a
  % Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
                                 Dload  Upload   Total   Spent    Left  Speed
104   313  104   313    0     0    349      0 --:--:-- --:--:-- --:--:--   634

gzip: stdin: not in gzip format
tar: Child returned status 1
tar: Error is not recoverable: exiting now
The command '/bin/sh -c mkdir $ASTROPFX/bin && cd $ASTROPFX/bin && curl http://ds9.si.e
```

Checking the corresponding website, one can see that the 6.8.0 version is no longer there; presumably it is replaced with 6.8.1. Replacing the
`curl http://ds9.si.edu/download/centos6/ds9.centos6.8.0.tar.gz | tar zxv`
command with
`curl http://ds9.si.edu/download/centos6/ds9.centos6.8.1.tar.gz | tar zxv`
in the Dockerfile solves the issue.

IevgenVovk mentioned this issue on Jan 30

Fix ds9 download issue #6 · Merged

IevgenVovk commented on Jan 30 · Author

The proposed fix is implemented in the pull request #6 .

jasercion commented on Jan 31 · Member

@Areustle Please verify fix in PR and issue merge decision

Areustle commented 29 days ago · Member

Solved by PR

Areustle closed this 29 days ago

Labels, Versions and Tags! Oh My!

- Versions
- Tags
- Labels

Fermitools Versioning Scheme (Conda)

- Conda cloud binaries given 3-digit version numbers following Major.Minor.Patch pattern.
- Major Update: large functionality changes, new tools, api changes…
  - Example: Python 3 update
- Minor Update: new models and data, individual tool updates, new features
  - Example: edisp features in Likelihood.
- Patch: Bug fixes, issue mitigations, dependency updates, etc.

Fermitools Tagging Scheme (Github)

- Source code tags in fermi-lat GH repositories can be managed easily using repoman release
- Should follow Descriptor-xx.xx.xx style
  - Likelihood-0.0.0
  - Fermitools-0.0.0
- We don’t use github tags very frequently other than to tag new releases.

Fermitools labeling scheme (Conda)

- Labels allow another dimension of version segmentation in conda cloud binary downloads.
- conda install -c conda-forge/label/cf201901 -c fermi/label/dev fermitools
- main: the default label, applied to all releases [-c fermi]
- dev: a throwaway label with broken binaries [-c fermi/label/dev]
- alpha: binaries ready for verification tests [-c fermi/label/alpha]
- beta: binaries ready for validation (HitL) tests [-c fermi/label/beta]