



Adam Barton

Introduction

- Jekyll has recently become popular for presenting software documentation
- Instead of using databases, Jekyll takes the content, renders Markdown or Textile and Liquid templates, and produces a complete, static website ready to be served by Apache HTTP Server, Nginx or another web server.

Currently used in ATLAS

- We have one public site [here](#) hosting:
 - Athena Guides
 - Tutorials
 - Procedures for software shifters
- Designed to be viewable on all screen sizes

ATLAS Software Documentation Guides ▾ Tutorials ▾ Links ▾

Shift guide for release coordinators

Last update: 03 May 2017 [\[History\]](#)

Introduction

This page contains information about the workflows and procedures a release coordinator is concerned with. It is assumed that you are familiar with the general workflow for ATLAS Offline software development as summarised in the [Workflow Quick Reference](#). Having an overview of the tasks of the software review shifters as well as knowing basics of git is also helpful.

Finding open merge requests for your release

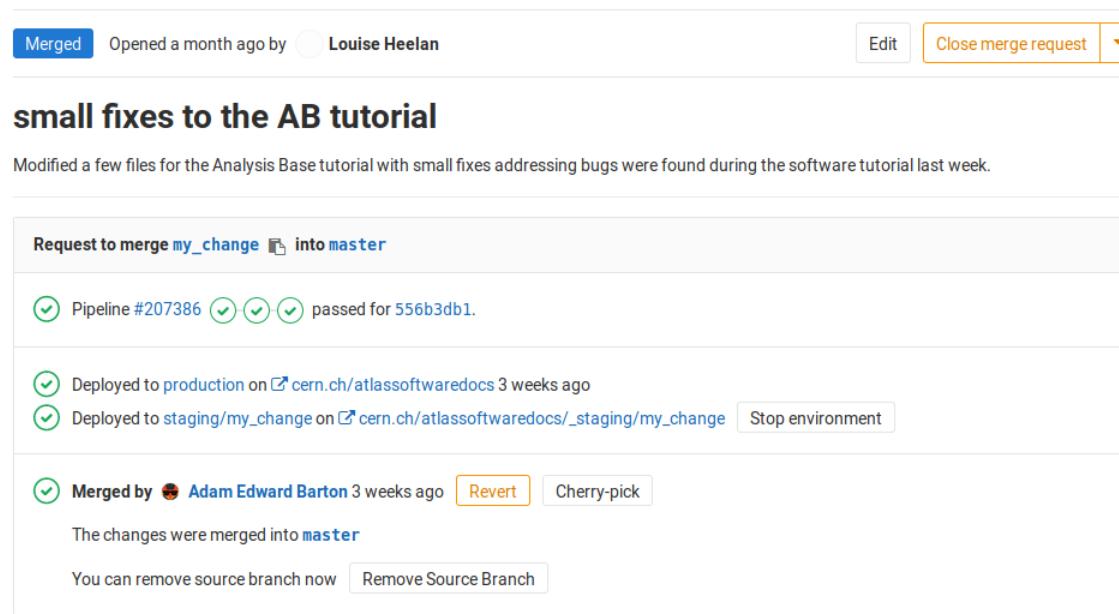
In git, different releases are represented by different branches of the [atlas/athena](#) git repository. One subtle point is that *what* code we build from the branch is controlled by the project specific scripts that live in the [Projects](#) directory of the branch (e.g., [here](#) for the [master](#) branch). You are, however, the release coordinator for *all* projects built from a branch.

Upon creation, merge requests will be tagged automatically by the CI system with a label indicating their target branch. In addition, the CI system also adds a label indicating the stage and result of the software review process. Changes approved by the software review shifters are labeled as *review-approved*. Therefore, you should be able to select all ready-to-be-accepted merge requests from the [Gitlab merge request overview page](#). Simply filter the list of merge requests by selecting *review-approved* and your release branch name from the dropdown menu for labels.

Full rebuilds need to be triggered after updates where an incremental rebuild may not catch all changes properly, such as a LCG version update. Such updates should be performed in late evenings, with advance notification emailed to [ATLAS](#).

Jekyll

- In our implementation it is not trivially edited
- Edits to the source code are validated in a gitlab merge request procedure
- Once validated they are merged and changes go live the following day



Merged Opened a month ago by Louise Heelan Edit Close merge request

small fixes to the AB tutorial

Modified a few files for the Analysis Base tutorial with small fixes addressing bugs were found during the software tutorial last week.

Request to merge `my_change` into `master`

- ✓ Pipeline #207386 passed for 556b3db1.
- ✓ Deployed to production on cern.ch/atlassoftwaredocs 3 weeks ago
- ✓ Deployed to staging/my_change on cern.ch/atlassoftwaredocs/_staging/my_change Stop environment
- ✓ Merged by Adam Edward Barton 3 weeks ago Revert Cherry-pick

The changes were merged into `master`

You can remove source branch now Remove Source Branch

How to make Jekyll contributions

- Install Jekyll
- Use git to clone the repository:
 - `git clone https://gitlab.cern.ch/atlas-sw-git/atlassoftwaredocs.git`
 - `cd atlassoftwaredocs`
- Run Jekyll:
 - `jekyll serve`

```
Configuration file: /home/adam/jekyllTest/ModalImage_PopoutMenu_update2/atlassoftwaredocs/_config.yml
Configuration file: /home/adam/jekyllTest/ModalImage_PopoutMenu_update2/atlassoftwaredocs/_config.yml
   Source: /home/adam/jekyllTest/ModalImage_PopoutMenu_update2/atlassoftwaredocs
  Destination: /home/adam/jekyllTest/ModalImage_PopoutMenu_update2/atlassoftwaredocs/_site
Incremental build: disabled. Enable with --incremental
Generating...
done in 3.505 seconds.
Auto-regeneration: enabled for '/home/adam/jekyllTest/ModalImage_PopoutMenu_update2/atlassoftwaredocs'
Configuration file: /home/adam/jekyllTest/ModalImage_PopoutMenu_update2/atlassoftwaredocs/_config.yml
  Server address: http://127.0.0.1:4000/
Server running... press ctrl-c to stop.
```

Editing code

- Open the local address in your web browser to see the rendered page
- Edit the source code, the new code is automatically compiled so you can just reload your webpage.
- Jekyll has a mark up language for more advanced features

```
<ul>
  {% for post in site.posts %}
    <li>
      <a href="{{ post.url }}">{{ post.title }}</a>
      {{ post.excerpt }}
    </li>
  {% endfor %}
</ul>
```

Markup

- `##` Header 2 (H1 is reserved for post titles)`##`
- `###` Header 3
- `####` Header 4
- A link to [Jekyll Now](http://github.com/barryclark/jekyll-now/).
- An image, located within `/images`
 - `![an image alt text]({{ site.baseurl }}/images/jekyll-logo.png "an image title")`
- `*` A bulleted list
- `-` alternative syntax 1
- `+` alternative syntax 2
- `-` an indented list item
- `-` `_italics_`
- `-` `**bold**`
- `-` ``code()``

Review and Commit

- The web page should be correct, renders correctly and be readable on all screen sizes
- Commit your code to a new git branch

ATLAS Software Documentation Guides Tutorials Links

Tutorial Home [↗](#)

Basics

- Help With Git

Detailed Tutorial

- Set Up
- Fork the Repository
- Clone Repository Locally
- Develop Code
- Make a Merge Request
- Resolving Conflicts

Code Review

- Continuous Integration
- Review a request

Reference

- Workflow Quick Reference

Misc

- Migration from SVN

ATLAS Software Git Workflow

Last update: 28 Feb 2017 [\[History\]](#)

Introduction

These web pages describe how to work with ATLAS offline code hosted in the git version control system.

If you have no experience with git at all, it's worth checking out our collection of links in the [Help with git](#) section before anything else. There you'll find a getting started guide for git itself and a leg up from SVN.

As well as using git as the source control management program, ATLAS uses the [GitLab](#) service at CERN to provide web based code management, and so we also suggest some [GitLab bootstrap links](#).

Workflow Overview

Before starting to follow each step of the workflow in detail, it's useful to have an overview of the whole process, uncluttered from all minutiae. That's what we give here, so refer back to this page if you start to lose a sense of where you are overall or why you are doing what you are doing.

In a Nutshell

1. Get your own copy of the offline code from GitLab
2. Develop, test, develop, retest, ... until you are happy
3. Commit your code and push it back to GitLab
4. Create a merge request that asks for your code to be added to the main repository
5. Continuous integration runs to test your changes, a human being checks your code - when there are no problems with your code, the merge is accepted and your code has been integrated

ATLAS Software Documentation

ATLAS Software Git Workflow

Last update: 28 Feb 2017 [\[History\]](#)

Introduction

These web pages describe how to work with ATLAS offline code hosted in the git version control system.

If you have no experience with git at all, it's worth checking out our collection of links in the [Help with git](#) section before anything else. There you'll find a getting started guide for git itself and a leg up from SVN.

As well as using git as the source control management program, ATLAS uses the [GitLab](#) service at CERN to provide web based code management, and so we also suggest some [GitLab bootstrap links](#).

Workflow Overview

Before starting to follow each step of the workflow in detail, it's useful to have an overview of the whole process, uncluttered from all minutiae. That's what we give here, so

GitLab

- Open a Merge Request and ping a Documentation team member or Software coordinator to review and accept the merge.

Add page summarising 2017 branch gymnastics in the 21 release to the software docs

Various merge operations between release 21 branches in 2017 can make it less obvious what is included in any release.

This MR aims to add a page to visually summarise the releases and merges from 2017 and 2018 to act as a reference.

Uses <http://gitgraphjs.com/> under the MIT licence

Edited 2 weeks ago by Tim Martin

Request to merge add-branch-structure-page into **master**

🚨 Pipeline #244495 🟢🟢🚨 passed with warnings for 7af2a121.

🟢 Merged by 🧑‍💻 **Adam Edward Barton** a week ago Revert Cherry-pick

The changes were merged into **master**

The source branch has been removed

👍 1 🗨️ 0 😊

Conclusion

- Jekyll is an interesting new tool for producing documentation
- We have implemented quite a few guides and these have proven quite successful
- It has the potential to be expanded in new and interesting ways