Open Preservation Foundation

An introduction to Git and GitHub

A beginners guide to the OPF GitHub Organisation and the JHOVE project

Version control, what & why?

- Change control and version control are two different sides of the same coin.
- Software projects require strong version control.
- Large distributed software projects are very complex, testing and release processes need to be reliable and repeatable.

Git

- Git is Version Control software
- Developed by Linus Torvalds for development of the Linux kernel.
- Used for software projects and copes well with distributed complexity.
- Is open source and free to use.
- Is now the most popular version control system in the world

Git is...

- Available as a command line tool for any platform.
- A command line tool that is not for the faint hearted.
- Not necessary for (nearly?) everything covered today.
- Something we'll say as little more about as possible.

GitHub

- Is a web hosting service for source code and version control based upon Git.
- Is visible to anyone with a web-browser (transparency), though you'll need an account to use any interactive features.
- Is free to use unless you want private repositories.
- Has been around for more than 10 years, in that time it has become the biggest largest host for source code in the world

GitHub in numbers

- 83 million users
- Across 200 countries
- More than 100 million Git repositories
- 49 million public Git repositories
- In Feb 2018 it was the victim of the largest distributed denial of service attack in history, 1.35 terabits per second

Git & GitHub what's the difference?

- Git is a set of software tools, generally used by "hard-core" types like developers.
- GitHub is a website and host of online Git repositories.
- GitHub also add features on top of Git.
- These features are fairly user friendly and straightforward to use and a LOT easier than using Git.

Some GitHub terms

Collaborator : someone with read and write access to a repository

Contributor : someone who has contributed changes to a project.

Issue : suggested improvements, tasks or questions related to a repository.

GitHub Pages : a static website hosting service for organisations and repositories.

Markdown : a structured text file format that supports a subset of HTML.

Organisations : shared accounts where businesses or open-source orgs can collaborate on several projects at once, these support **Teams**.

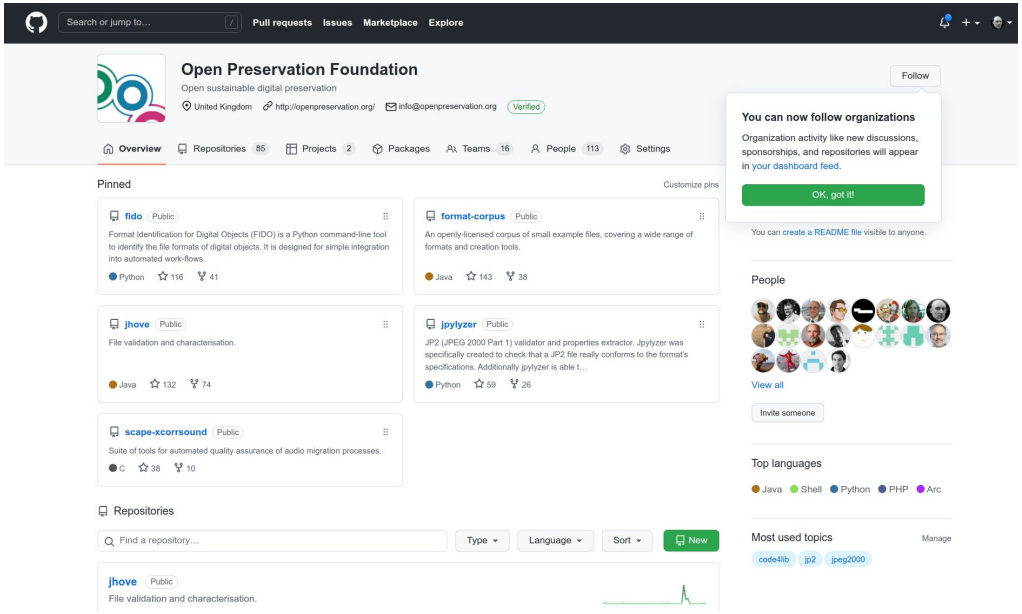
Pull request : proposed changes to a repository submitted by a contributor.

The OPF GitHub Organisation

This is the OPF's GitHub organisation page: https://github.com/openpreserve

It's the main hub for all OPF GitHub activity.

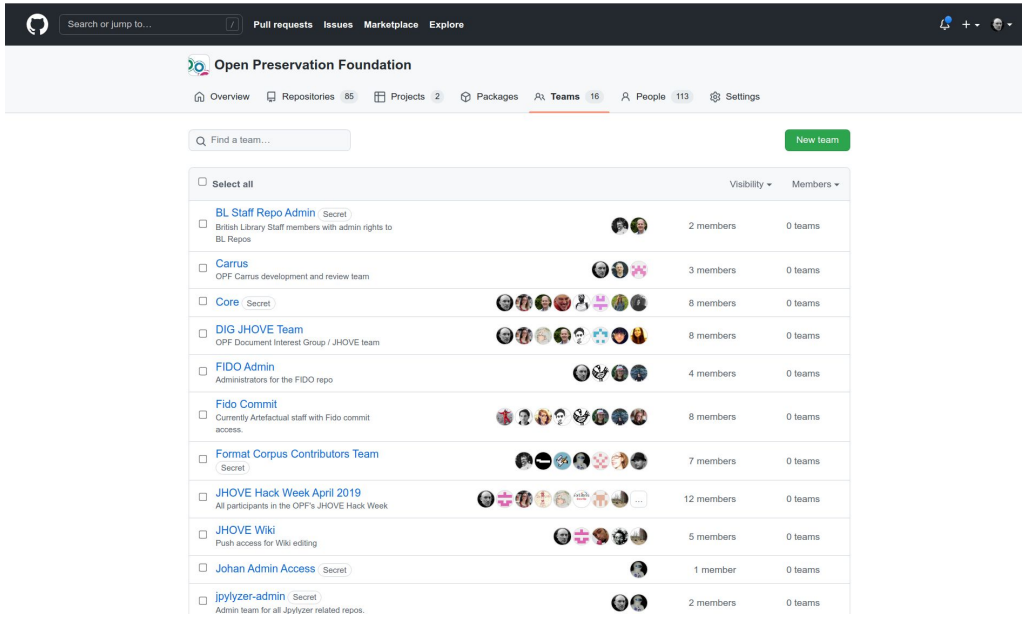
This tab lists the repositories in the order they were last modified.



The OPF GitHub Organisation

This is another view of the OPF organisation page but this time shows the current teams and members.

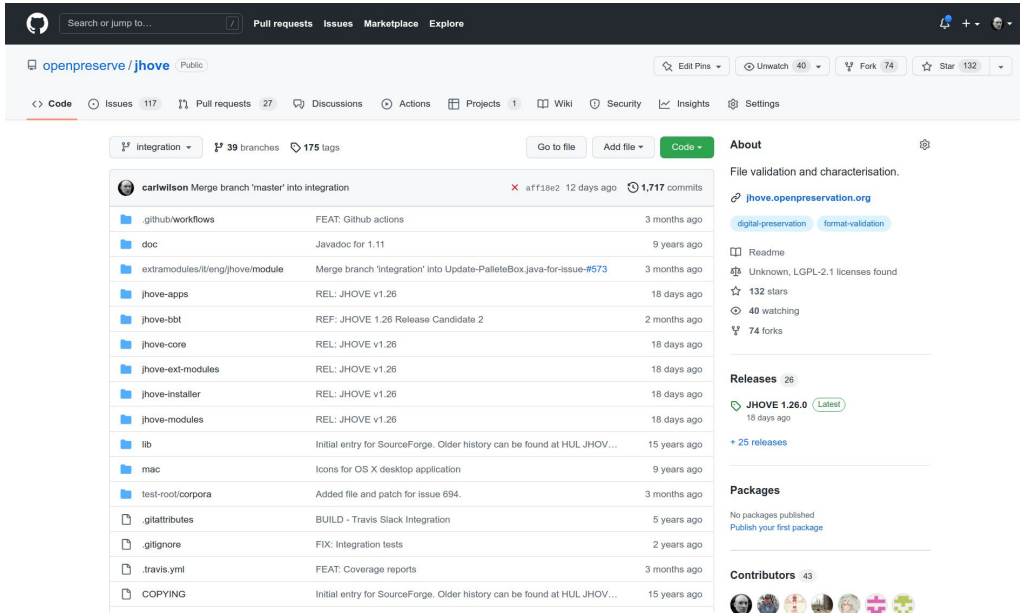
Individual teams have fine grained permissions on repositories, i.e. who can write, merge and administer teams.



The JHOVE repository

This is the GitHub repository for the JHOVE project: https://github.com/openpreserve/jhove

It's really the root directory of the project file system.



Writing on GitHub

- GitHub is NOT all about software development.
- Software projects require developer and user documentation.
- GitHub provides some straightforward tools for non-developers to write on GitHub.
- Most GitHub writing is done in Markdown……

What is Markdown?

Markdown is:

- A lightweight markup language written and formatted in plain ASCII text.
- Developed by John Gruber in 2004: https://daringfireball.net/projects/markdown/syntax
- Designed to be easily converted to HTML and other document formats by automated tools.
- The predominant format for documentation on GitHub.

Markdown and HTML

Raw Markdown

```
COMMON SPECIFICATION FOR INFORMATION PACKAGES
=============================================
This is the web site for the E-ARK Common Specification for Information
Packages. The site is still a work in progress as we restructure the
specification. The site current site contents are as follows.

E-ARK CSIP
---------
An HTML version of the E-ARK Common Specification for Information Packages the
[table of Contents](./specification/) is a good place to start.
It's possible to refer to the main sections of the specification by URL,
e.g. "PART II: Implementation of the CS IP" is located at
https://carlwilson.github.io/E-ARK-CSIP/specification/implementation/.
Lower level headings have page anchors, e.g. 5.3 Use of METS has the URL
https://carlwilson.github.io/E-ARK-CSIP/specification/implementation/metadata/#53-use-of-mets.
Individual requirements also have page anchors and URLS, e.g.
https://carlwilson.github.io/E-ARK-CSIP/specification/implementation/metadata/#CSIP80.

Note that this site version is published from a
[Markdown](https://guides.github.com/features/mastering-markdown/) source [on
GitHub](https://github.com/DILCISBoard/E-ARK-CSIP/).

Archive
-------
Previous versions of the specification are available from [the archive](./archive/)

```

Markdown as HTML

COMMON SPECIFICATION FOR INFORMATION PACKAGES

This is the web site for the E-ARK Common Specification for Information Packages. The site is still a work in progress as we restructure the specification. The site current site contents are as follows.

E-ARK CSIP

An HTML version of the E-ARK Common Specification for Information Packages the table of Contents is a good place to start. It's possible to refer to the main sections of the specification by URL, e.g. "PART II: Implementation of the CS IP" is located at https://carlwilson.github.io/E-ARK-CSIP/specification/implementation/. Lower level headings have page anchors, e.g. 5.3 Use of METS has the URL https://carlwilson.github.io/E-ARK-CSIP/specification/implementation/metadata/#53-use-of-mets. Individual requirements also have page anchors and URLS, e.g. https://carlwilson.github.io/E-ARK-CSIP/specification/implementation/metadata/#CSIP80.

Note that this site version is published from a Markdown source on GitHub.

Archive

Previous versions of the specification are available from the archive

GitHub flavoured Markdown

- Markdown is so popular on GitHub that there's a GitHub dialect, GitHub Flavoured Markdown.
- It can be used almost anywhere you can write text in the GitHub user interface.
- It can be freely used in text documents, with the extension ".md".
- There's a good online reference: https://guides.github.com/features/mastering-markdown/

GitHub pages

GitHub pages:

- Is a method of publishing web pages and sites from the contents of a Git repository.
- Uses source files that can be Markdown or HTML.
- Can be driven by the contents of a specific branch or a specific folder in the master branch.
- Can use custom URLs to host “proper” websites.
- Uses a templating engine called Jekyll to convert the source to HTML.

GitHub pages examples

veraPDF documentation

veraPDF Get started Validation Policy CLI GUI Plugins Developers GitHub

veraPDF CLI Quick Start Guide

The veraPDF command line interface is the best way of processing batches of PDF/A files. It's designed for integrating with scripted workflows, or for shell invocation from programs.

We assume you've already downloaded and installed the software, if not please read the installation guide first.

Using the terminal

We've provided a quick primer on setting up and using the terminal on our supported platforms here.

Getting help

You can get the software to output its built in CLI usage message by typing `verapdf.bat -h` or `verapdf --help`, an online version is available here.

Configuring veraPDF

veraPDF is controlled by a set of configuration files, you can read a brief overview here.

How-tos

The following examples all make use of the veraPDF test corpus. This is available on GitHub. It is also installed with the veraPDF software if you enable it at step 3. The test corpus will be installed in a sub-directory called `corpus`. The examples assume your terminal session is running in the installation directory with a suitable alias set up to avoid typing `path-to-verapdf/verapdf`. On a Mac or Linux box this can be set up by typing `export verapdf='export verapdf='path-to-verapdf/verapdf'` at the command line.

E-ARK CSIP

5.3.1. Use of the METS root element (element mets)

The purpose of the METS root element is to describe the container for the information being stored and/or transmitted, which is held within the seven sections of the METS file. The root element of a METS document has five attributes derived from the official METS specification and one attribute added for the purposes of the CS IP.

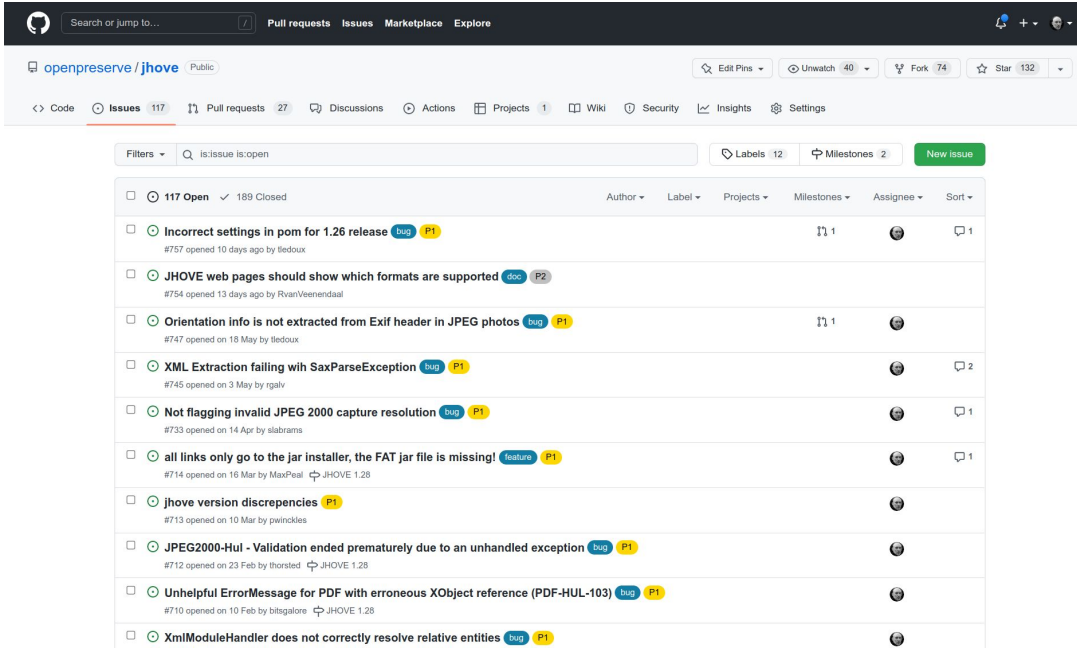
In addition to these six attributes the METS root element mets MUST define all relevant namespaces and locations of XML schemas using the `@xmlns` and `@xsi:schemaLocation` attributes. In case XML schemas have been included into the package (i.e. placed into the `schemas` folder) it is recommended to link to the schemas using the relative path of the schema file (i.e. `schemas/mets.xsd`). The specific requirements for the root element and its attributes are described in the following table .

ID	Name	Element/Attribute	Description and usage	Cardinality
CSIP1	METS root element	mets	The root level element that is required in all METS documents	1..1
CSIP2	Content ID	mets/@OBJID	Mandatory in this specification. It is recommended that it be the same as the name or ID of the package (the name of the root folder). The OBJID must meet the CS IP requirement of being unique at least across the repository	1..1
CSIP3	General content type	mets/@TYPE	Mandatory in this specification. The TYPE attribute must be used for identifying the type of the package (genre), for example ERMS, RDBMS, digitised construction plans. However, there is no fixed vocabulary and as such implementers are welcome to use values most suitable for their needs.	1..1

GitHub issues

GitHub issues:

- Are GitHubs own implementation of issue tracking, similar to Bugzilla or JIRA.
- Every repository gets an associated issue tracker though it can be turned off.
- Right is the Issue tracker for the JHOVE repository



Issues as tasks

For any issue you can:

- Assign team members or collaborators to deal with it.
- Add comments in Markdown.
- Add labels which are customisable to suit internal processes.
- Assign to Projects or Milestones

FIX: XML Entity file handling #701

Edit | Code

Draft | carlwilson wants to merge 2 commits into openpreserve:integration from carlwilson:fix/xmlent

Conversation 7 | Commits 2 | Checks 2 | Files changed 13 | +85 −156

carlwilson commented on 6 Jan • edited | Member | Tip

Fixes #700 output is now a more informative:

```
Status: Not well-formed
ErrorMessage: File not found: /home/cfw/proj/opf/jhove/article.dtd (No such file or directory)
 ID: XML-HUL-10
MIMEtype: text/xml
```

carlwilson added 2 commits 6 months ago

FIX: Open JDK 11 build | Verified | 30d75ce

FIX: XML entity processing | Verified | ac82bec

carlwilson added bug P1 labels on 6 Jan

carlwilson self-assigned this on 6 Jan

carlwilson marked this pull request as draft 6 months ago

pwinckles commented on 14 Jan | Contributor | Tip

Observation, if you're going to use an automated code formatter, it would make sense to run it on the entire project to create a baseline. This would allow for cleaner PRs and ease the burden of resolving spurious merge conflicts.

pwinckles reviewed on 2 Feb | View changes

Reviewers: pwinckles

Assignees: carlwilson

Labels: bug P1

Projects: JHOVE XML Module Improvements — Status: In Progress

Milestone: No milestone

Development: Successfully merging this pull request may close these issues. Missing XML DTD/Entity file resource causes ...

Notifications | Customize | Unsubscribe

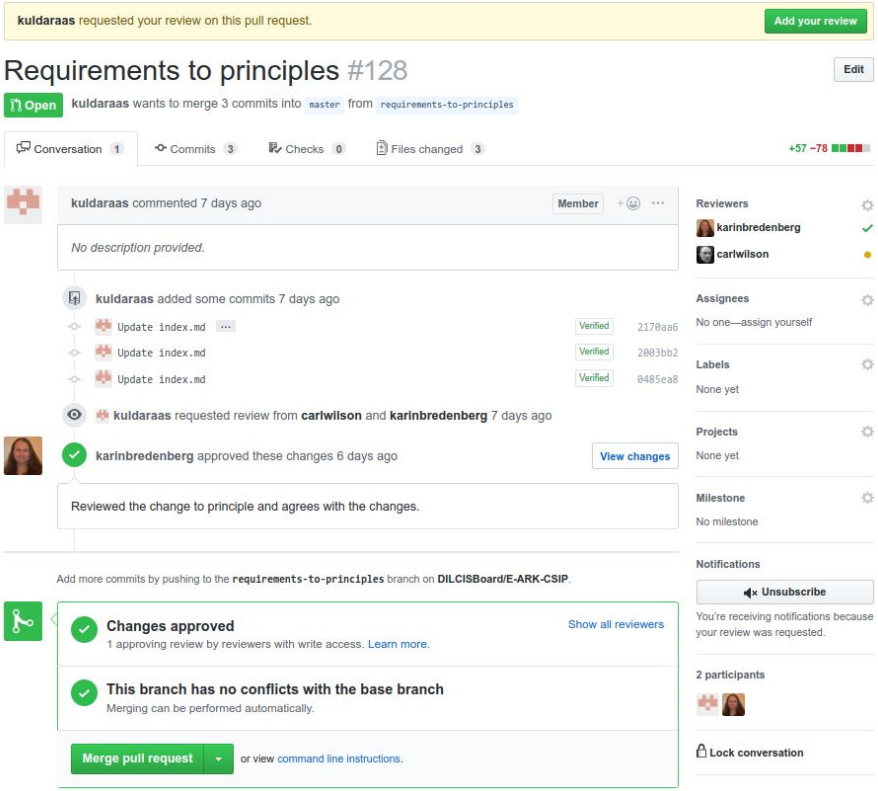
You're receiving notifications because you're watching this repository.

3 participants

Pull Requests

Pull requests:

- Are GitHub's mechanism for contributing to projects.
- Group a set of commits that are a proposed change.
- Support assignees, labels, project and milestones, the same as issues.
- Support automated QA and manual review processes/



Reviewing changes

- Pull Requests can be reviewed manually.
- Red shows the old version of text that has changed or been deleted.
- Green text is the new version that replaces the old.
- Reviewers can approve a PR or request changes

Correction pom.xml ; fixes #757 #758

Edit | Code

Open | tledoux wants to merge 2 commits into openpreserve:integration from tledoux:pomsettings

Conversation 1 | Commits 2 | Checks 7 | Files changed 2 | +18 −1

Changes from all commits | File filter | Conversations | 0 / 2 files viewed | Review changes

Filter changed files

- jhove-modules/jpeg-hul
  - pom.xml
- pom.xml

2 jhove-modules/jpeg-hul/pom.xml — Viewed

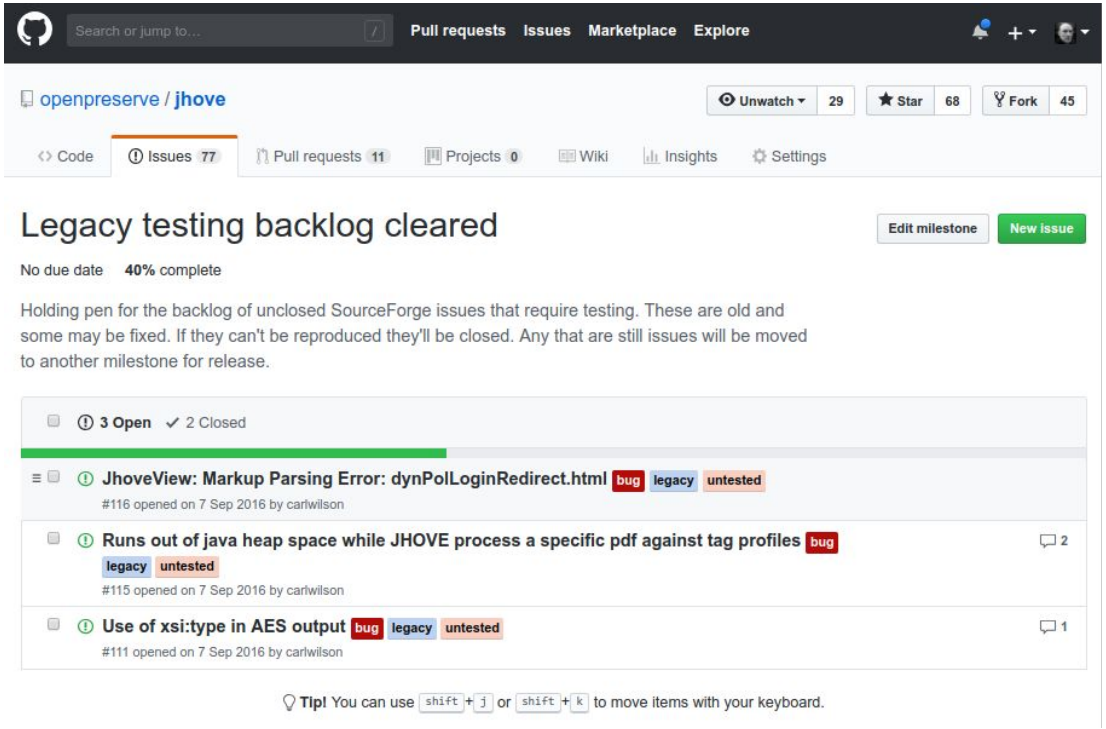
```
@@ -14,7 +14,7 @@
14  14      <dependency>
15  15        <groupId>org.openpreservation.jhove.modules</groupId>
16  16        <artifactId>tiff-hul</artifactId>
17  -         <version>1.9.1</version>
    17 +      <version>1.9.3</version>
18  18      </dependency>
19  19    </dependencies>
20  20
```

17 pom.xml — Viewed

```
@@ -86,6 +86,7 @@
86  86      <jacoco.version>0.8.7</jacoco.version>
87  87      <java.source.version>1.8</java.source.version>
88  88      <java.target.version>1.8</java.target.version>
    89 +    <mvn.target.release>8</mvn.target.release>
89  90      <jhove.timestamp>${maven.build.timestamp}</jhove.timestamp>
90  91      <maven.build.timestamp.format>yyyy-MM-dd HH:mm:ss</maven.build.timestamp.format>
91  92    </properties>
@@ -295,6 +296,22 @@
295 296     </dependency>
296 297   </dependencies>
297 298   <profiles>
    299 +   <profile>
    300 +     <id>jdk9</id>
    301 +     <activation>
```

GitHub milestones: release planning

- GitHub also provides Milestones.
- These are an arbitrary collections of Issues and Pull Requests.
- They can be assigned a deadline and used for release planning and management.
- This is a JHOVE milestone…



Git for non-software

- Using GitHub for software is common.
- Publishing test data is less widespread but not unusual.
- Managing specifications and documentation only is less common BUT not without precedent.
- If anyone has concerns or questions, now's the time.