



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.



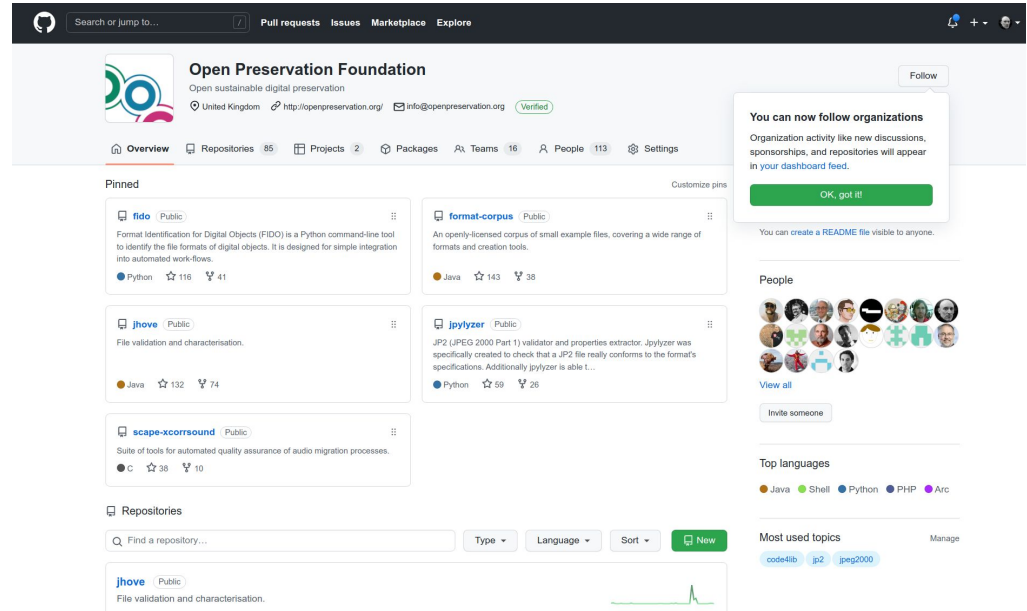
The OPF GitHub Organisation

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.

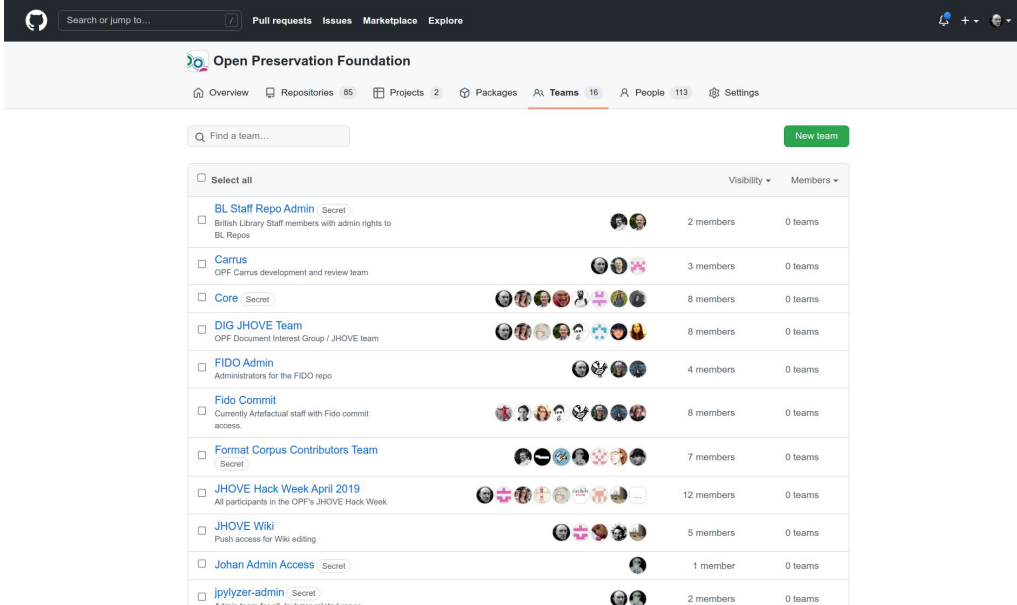


The screenshot shows the GitHub organization page for the Open Preservation Foundation. The header includes the organization's name, logo, and a 'Verified' badge. Below the header, there are navigation tabs for Overview, Repositories (85), Projects (2), Packages, Teams (16), People (113), and Settings. The main content area is titled 'Pinned' and displays a grid of repository cards. The first card is 'fido', a Python tool for digital object format identification. The second is 'format-corpus', an open-licensed corpus of example files. The third is 'jhove', a file validation tool. The fourth is 'jpylyzer', a JPEX 2000 Part 1 validator. The fifth is 'scapec-xcoursound', a suite of tools for audio migration quality assurance. Below the pinned repositories is a 'Repositories' section with a search bar and filters for Type, Language, and Sort, along with a 'New' button. The first repository listed is 'jhove'. On the right side, there is a 'You can now follow organizations' notification, a 'People' section with a grid of member avatars, and a 'Top languages' section showing Java, Shell, Python, PHP, and Arc. At the bottom right, there is a 'Most used topics' section with tags for code4lib, jp2, and jpeg2000.



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 screenshot shows the GitHub organisation page for the Open Preservation Foundation (OPF). The page is titled "Open Preservation Foundation" and has a navigation bar with links for Overview, Repositories (65), Projects (2), Packages, Teams (16), People (113), and Settings. Below the navigation bar, there is a search bar for finding a team and a "New team" button. The main content area displays a list of teams, each with a checkbox, a name, a description, a "Secret" label, a row of member avatars, the number of members, and the number of teams. The teams listed are:

Team Name	Description	Members	Teams
<input type="checkbox"/> BL Staff Repo Admin <small>Secret</small>	British Library Staff members with admin rights to BL Repos	2 members	0 teams
<input type="checkbox"/> Carrus	OPF Carrus development and review team	3 members	0 teams
<input type="checkbox"/> Core <small>Secret</small>		8 members	0 teams
<input type="checkbox"/> DIG JHOVE Team	OPF Document Interest Group / JHOVE team	8 members	0 teams
<input type="checkbox"/> FIDO Admin	Administrators for the FIDO repo	4 members	0 teams
<input type="checkbox"/> Fido Commit	Currently Artefactual staff with Fido commit access.	8 members	0 teams
<input type="checkbox"/> Format Corpus Contributors Team <small>Secret</small>		7 members	0 teams
<input type="checkbox"/> JHOVE Hack Week April 2019	All participants in the OPF's JHOVE Hack Week.	12 members	0 teams
<input type="checkbox"/> JHOVE Wiki	Push access for Wiki editing	5 members	0 teams
<input type="checkbox"/> Johan Admin Access <small>Secret</small>		1 member	0 teams
<input type="checkbox"/> jpylyzer-admin <small>Secret</small>	Admin team for all jpylyzer related repos.	2 members	0 teams

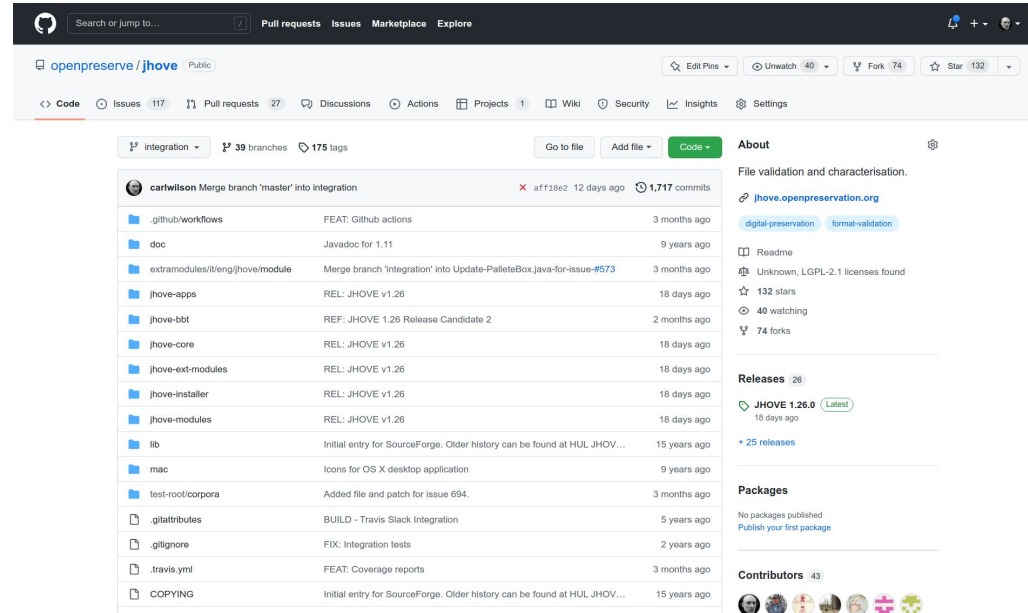


The JHOVE repository

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.



Search or jump to... Pull requests Issues Marketplace Explore

openpreserve/jhove Public

Code Issues 117 Pull requests 27 Discussions Actions Projects 1 Wiki Security Insights Settings

integration 39 branches 175 tags

Go to file Add file Code

File validation and characterisation.

jhove.openpreservation.org

digital-preservation format-validation

Readme

Unknown, LGPL-2.1 licenses found

132 stars

40 watching

74 forks

Releases 28

JHOVE 1.26.0 (Latest) 18 days ago

+ 25 releases

Packages

No packages published

Publish your first package

Contributors 43

File/Folder	Description	Age
.github/workflows	FEAT: Github actions	3 months ago
doc	Javadoc for 1.11	9 years ago
extramodules/iteng/jhove/module	Merge branch 'integration' into Update-PaletteBox.java-for-issue-#573	3 months ago
jhove-apps	REL: JHOVE v1.26	18 days ago
jhove-bbt	REF: JHOVE 1.26 Release Candidate 2	2 months ago
jhove-core	REL: JHOVE v1.26	18 days ago
jhove-ext-modules	REL: JHOVE v1.26	18 days ago
jhove-installer	REL: JHOVE v1.26	18 days ago
jhove-modules	REL: JHOVE v1.26	18 days ago
lib	Initial entry for SourceForge. Older history can be found at HUL JHOV...	15 years ago
mac	Icons for OS X desktop application	9 years ago
test-root/corpora	Added file and patch for issue 694.	3 months ago
.gitattributes	BUILD - Travis Slack Integration	5 years ago
.gitignore	FIX: integration tests	2 years ago
.travis.yml	FEAT: Coverage reports	3 months ago
COPYING	Initial entry for SourceForge. Older history can be found at HUL JHOV...	15 years ago



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

```
1 COMMON SPECIFICATION FOR INFORMATION PACKAGES
2 =====
3 This is the web site for the E-ARK Common Specification for Information
4 Packages. The site is still a work in progress as we restructure the
5 specification. The site current site contents are as follows.
6
7 E-ARK CSIP
8 -----
9 An HTML version of the E-ARK Common Specification for Information Packages the
10 \[table of Contents\]\(./specification/\) is a good place to start.
11 It's possible to refer to the main sections of the specification by URL,
12 e.g. "PART II: Implementation of the CS IP" is located at
13 https://carlwilson.github.io/E-ARK-CSIP/specification/implementation/.
14 Lower level headings have page anchors, e.g. 5.3 Use of METS has the URL
15 https://carlwilson.github.io/E-ARK-CSIP/specification/implementation/metadata/#53-use-of-mets.
16 Individual requirements also have page anchors and URLs, e.g.
17 https://carlwilson.github.io/E-ARK-CSIP/specification/implementation/metadata/#CSIP80.
18
19 Note that this site version is published from a
20 \[Markdown\]\(https://guides.github.com/features/mastering-markdown/\) source \[on
21 GitHub\]\(https://github.com/DILCISBoard/E-ARK-CSIP/\).
22
23 Archive
24 -----
25 Previous versions of the specification are available from \[the archive\]\(./archive/\)
26
```

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



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 .

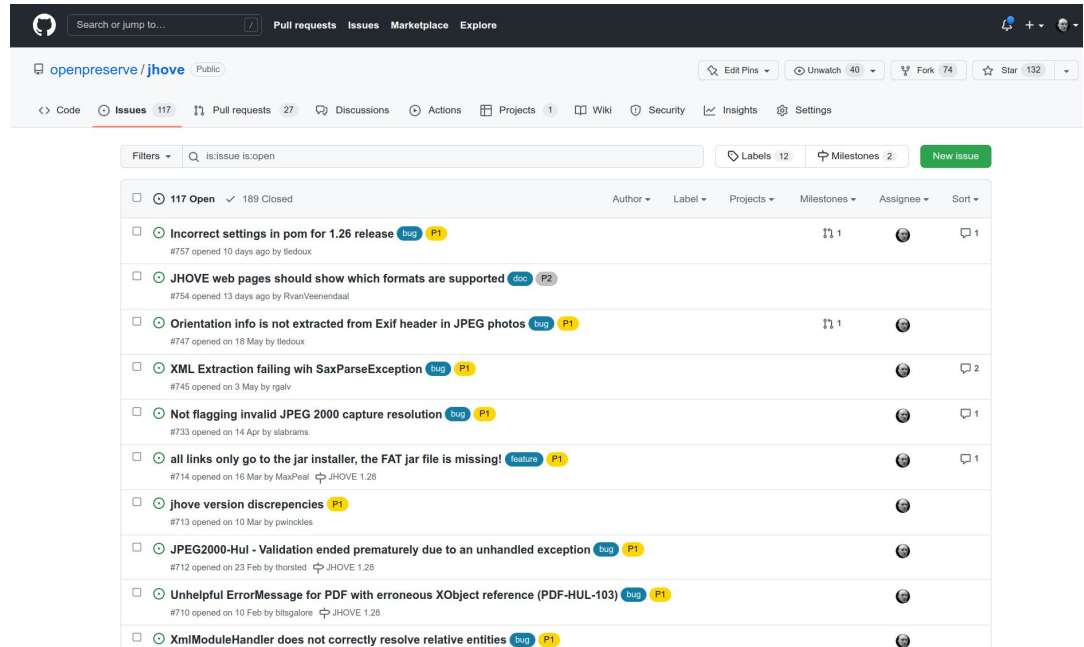
ID	Name	Element/Attribute	Description and usage	Cardinality
CSIP1	METS root element	mets	The root level element that is required in all METS documents	1..1
CSIP2	Content ID	mets@OBJID	Mandatory in this specification. It is recommended that it be the same as the name or ID of the package (the name of 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 GitHub's 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



The screenshot displays the GitHub Issues interface for the repository 'openpreserve / jhove'. The top navigation bar includes 'Pull requests', 'Issues', 'Marketplace', and 'Explore'. The repository name 'openpreserve / jhove' is shown as 'Public'. The 'Issues' tab is active, showing 117 issues. A search filter 'is:issue is:open' is applied. The list of issues includes:

- 117 Open** (189 Closed)
- Incorrect settings in pom for 1.26 release** (bug, P1) - #757 opened 10 days ago by sedoux
- JHOVE web pages should show which formats are supported** (doc, P2) - #754 opened 13 days ago by RvanVeenendaal
- Orientation info is not extracted from Exif header in JPEG photos** (bug, P1) - #747 opened on 18 May by fiedoux
- XML Extraction failing with SaxParseException** (bug, P1) - #745 opened on 3 May by rgalv
- Not flagging invalid JPEG 2000 capture resolution** (bug, P1) - #733 opened on 14 Apr by slabrans
- all links only go to the jar installer, the FAT jar file is missing!** (feature, P1) - #714 opened on 16 Mar by MaxPeal
- Jhove version discrepancies** (P1) - #713 opened on 10 Mar by gwinkles
- JPEG2000-Hul - Validation ended prematurely due to an unhandled exception** (bug, P1) - #712 opened on 23 Feb by thorsted
- Unhelpful ErrorMessage for PDF with erroneous XObject reference (PDF-HUL-103)** (bug, P1) - #710 opened on 10 Feb by bitgalore
- XmlModuleHandler does not correctly resolve relative entities** (bug, P1) - #709 opened on 10 Feb by bitgalore



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

The screenshot shows a GitHub pull request interface for issue #701, titled "FIX: XML Entity file handling". The pull request is in a "Draft" state and is being reviewed by "pwinckles". The pull request description states: "carlwilson wants to merge 2 commits into openpreserve:integration from carlwilson:fix/xmlent".

The pull request details include:

- Conversation: 7
- Commits: 2
- Checks: 2
- Files changed: 13
- +85 -156

The pull request history shows:

- carlwilson commented on 6 Jan • edited: "Fixes #700 output is now a more informative: Status: Not well-formed ErrorMessage: File not found: /home/ctw/proj/opf/jhove/article.dtd (No such file or directory) ID: XML-HUL-19 MIMETYPE: text/xml"
- carlwilson added 2 commits 6 months ago:
 - FIX: Open JOK 11 build
 - FIX: XML entity processing
- carlwilson added "bug" and "P1" labels on 6 Jan
- carlwilson self-assigned this on 6 Jan
- carlwilson marked this pull request as draft 6 months ago

The pull request is assigned to "pwinckles" and has the following metadata:

- 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: Unsubscribe
- 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/

The screenshot shows a GitHub Pull Request for the repository 'Requirements to principles #128'. At the top, a yellow banner indicates that 'kuldaraas' requested a review. Below this, the pull request title and a green 'Open' button are visible. The interface shows the pull request details, including the number of conversations (1), commits (3), checks (0), and files changed (3). A comment from 'kuldaraas' is shown, stating 'No description provided.' Below the comment, a commit history is displayed, showing three commits by 'kuldaraas' titled 'Update index.md', each with a 'Verified' status and a commit hash. A review by 'karinbrenberg' is shown, with a green checkmark and the text 'Reviewed the change to principle and agrees with the changes.' The right sidebar contains metadata for the pull request, including reviewers (karinbrenberg and carlwilson), assignees (none), labels (none), projects (none), milestones (none), and notifications (unsubscribe). At the bottom, a green box contains the status 'Changes approved' and 'This branch has no conflicts with the base branch', along with a 'Merge pull request' button.



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

tedoux wants to merge 2 commits into openpreserve:integration from tedoux:posettings

Conversation 1 Commits 2 Checks 7 Files changed 2

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

Filter changed files

jhove-modules/peg-hul pom.xml

```
@@ -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
```

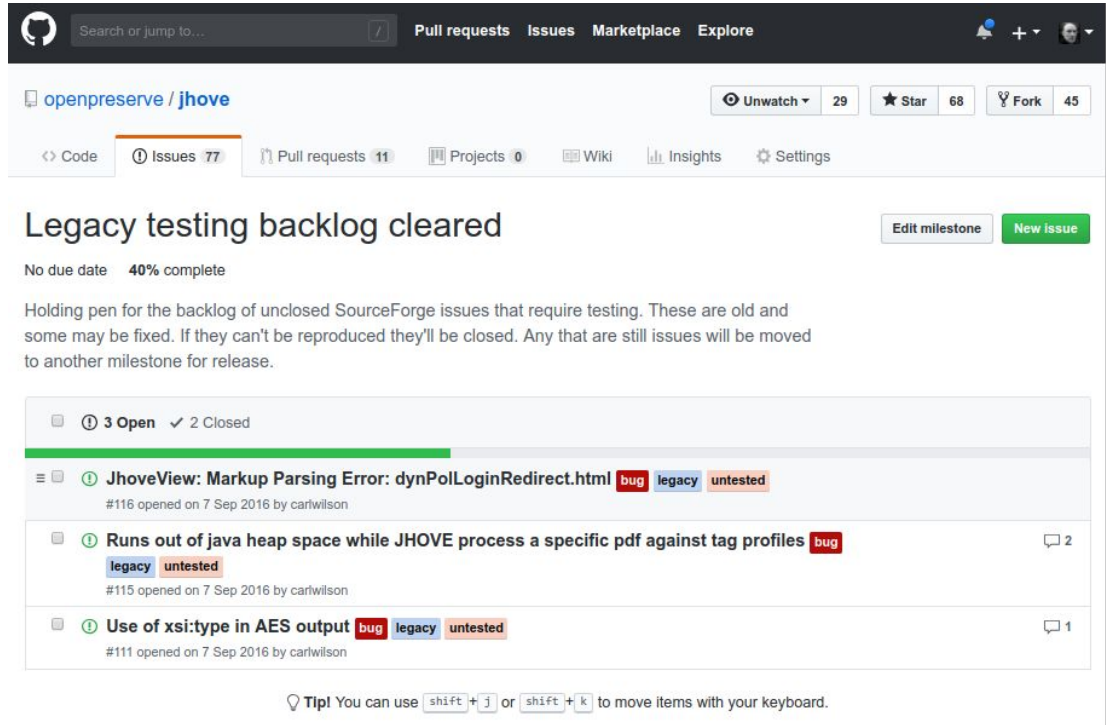
```
@@ -86,6 +86,7 @@
86 86 <jacoco.version>0.8.7</jacoco.version>
87 87 <java.source.version>1.8</java.source.version>
88 88 <java.target.version>1.8</java.target.version>
89 + <mvn.target.release>0</mvn.target.release>
89 89 <jhove.timestamp>${maven.build.timestamp}</jhove.timestamp>
90 90 <maven.build.timestamp.format>yyyy-MM-dd HH:mm:ss</maven.build.timestamp.format>
91 92 </properties>
92 92
```

```
@@ -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...



The screenshot shows a GitHub repository page for 'openpreserve / jhove'. The main heading is 'Legacy testing backlog cleared' with 'Edit milestone' and 'New issue' buttons. Below the heading, it states 'No due date' and '40% complete'. A descriptive paragraph explains that this milestone is for unclosed SourceForge issues requiring testing, some of which may be fixed or moved to other milestones. A progress bar shows 3 open and 2 closed issues. Three issues are listed:

- JhoveView: Markup Parsing Error: dynPolLoginRedirect.html** (bug, legacy, untested) #116 opened on 7 Sep 2016 by carlwilson
- Runs out of java heap space while JHOVE process a specific pdf against tag profiles** (bug, legacy, untested) #115 opened on 7 Sep 2016 by carlwilson (2 comments)
- Use of xsi:type in AES output** (bug, legacy, untested) #111 opened on 7 Sep 2016 by carlwilson (1 comment)

A tip at the bottom suggests using keyboard shortcuts: `shift+j` or `shift+k` to move items.



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.

