Getting Started with JHOVE

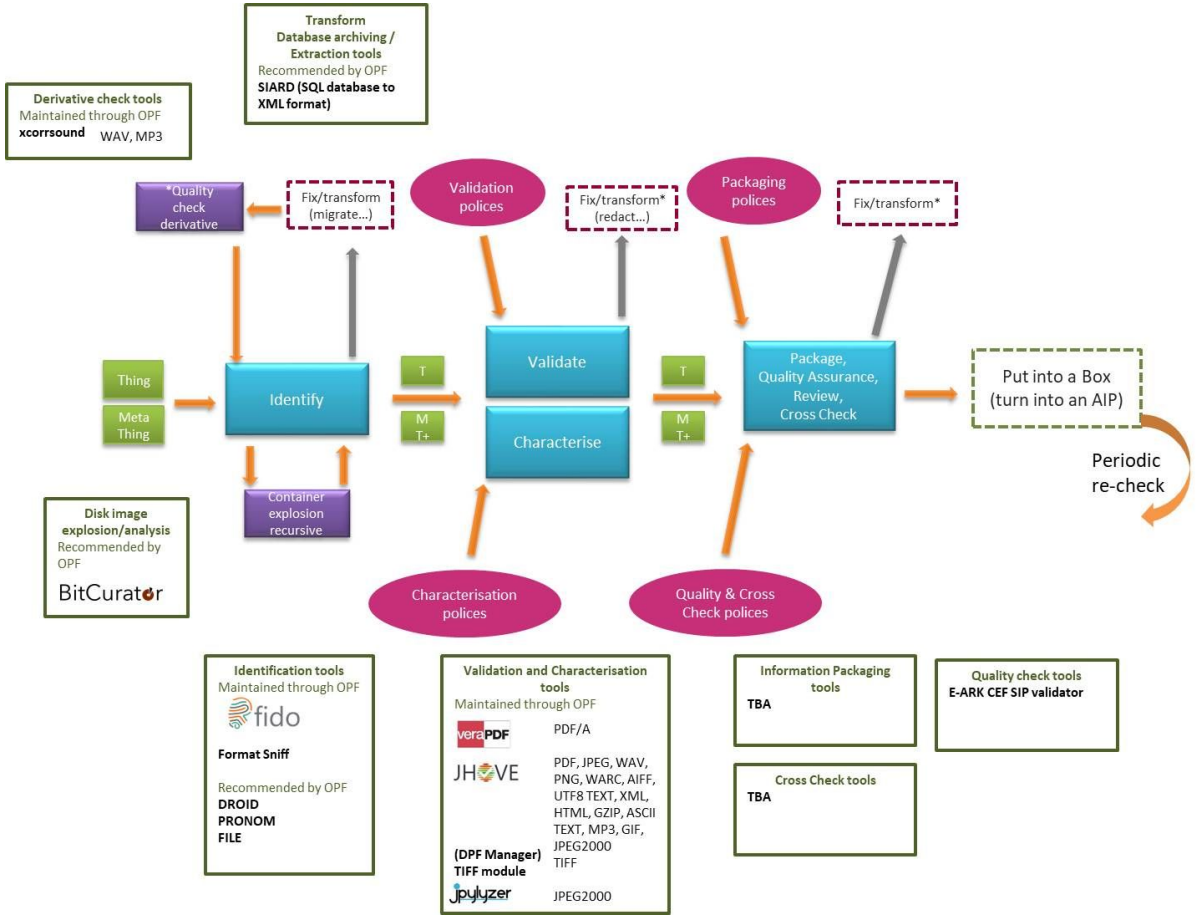
Carl Wilson, Technical Lead
Open Preservation Foundation

25 January, Koninklijke Bibliotheek

The Open Preservation Foundation

- A not for profit, global membership association providing stewardship of open-source tools for the digital preservation community.
- Founded in 2010 to sustain the results of the EU PLANETS project.
- The OPF provides stewardship for JHOVE, veraPDF, FIDO and more.
- Carl is the Technical Lead at the Open Preservation Foundation

OPF reference toolset



Over to you: Lightning talks

Please tell us:

- Your name, organisation and role
- What you are working on (and any content samples you have brought)
- Why you signed up for the workshop and what you hope to get out of it

Agenda

- A brief overview and history of JHOVE
  - Why do we validate file formats? (Policy and Practice)
  - Where JHOVE can fit into your workflow
- Installing and configuring and running JHOVE

  Break
- JHOVE results and how to interpret them
- How you can contribute to JHOVE
- A look at the Foundation's plans for JHOVE

Open Preservation Foundation

What is JHOVE and a little history

Purpose & Background

What is JHOVE?

JSTOR/Harvard Object Validation Environment

An EXTENSIBLE software framework for format:

- identification,
- validation, and
- characterisation.

Or Put Another Way...

JHOVE provides some answers for the following questions regarding a digital object:

- What format is it?
- The object is of format: F, is it?
- The object is of format: F, what are its properties?

Why do we validate formats?

Knowledge of the file formats and the properties of the files you're preserving provides the basis for preservation planning:

- What software do I need to be able to provide access to preserved content?
- Is content at risk due to some newly discovered problem?
- Hypothetically, somebody could discover a problem with PDFs that contain JPEG2000 images, or particular fonts. How would you know if your content was at risk?

Levels of conformance

Format validation conformance is determined at three levels: **well-formedness**, **validity**, and **consistency**.

- A digital object is well-formed if it meets the purely syntactic requirements for its format
- An object is valid if it is well-formed and it meets the higher-level semantic requirements for format validity
- An object is consistent if it is valid and its internally extracted representation information is consistent with externally supplied representation information

Format validation limits

- Validation tools can change over time so we recommended running validation software regularly over your digital objects
- Not every file format type has corresponding validation software available
- Validation software will detect issues with files that do not relate to the specific rules set for the file format specification

Where can JHOVE fit in your workflow?

JHOVE is available in three forms:

- Command line Tool - used by researchers and practitioners / embedded in DP workflows.
- Desktop GUI - file at a time interface, aimed at general users wanting to try JHOVE.
- JHOVE API - used by developers embedding JHOVE in their own projects.

JHOVE Origins

Versions 1.1 (April, 2008) - 1.6 (January, 2011) developed by:

- Harvard University Library &
- JSTOR; with funding from the
- Andrew W. Mellon Foundation

and made available under the GNU Lesser Public License (LGPL).

The Heroics of Gary McGath

Versions 1.7 (August, 2012) - 1.11 (September, 2013) maintained and released independently by Gary McGath.

Around May 2014 Gary put out a call for somebody to take over the maintenance of JHOVE……

JHOVE and the OPF

In March-April 2015 the Open Preservation Foundation took over the mantle of JHOVE maintainer from Gary following discussions with Gary, Harvard and JSTOR (Portico).

The OPF Inheritance

JHOVE assets OPF is now responsible for:

- code base (GitHub);
- 15 format modules;
- website http://jhove.openpreservation.org/;
- 16 downloadable release packages;
- mailing lists;
- issue tracker with 87 open issues, 75 closed; and
- 10 open pull requests, 208 closed on GitHub

Open Preservation Foundation

Installing & using JHOVE, a demonstration

Getting Started with JHOVE

Getting Started

JHOVE documentation is available on a dedicated site:

- http://jhove.openpreservation.org/

Open Preservation Foundation

Analysing the JHOVE results

What's JHOVE Telling Me?

Anyone can contribute and help

Improving JHOVE

Log an Issue

Use the GitHub Issue tracker:
https://github.com/openpreserve/jhove/issues

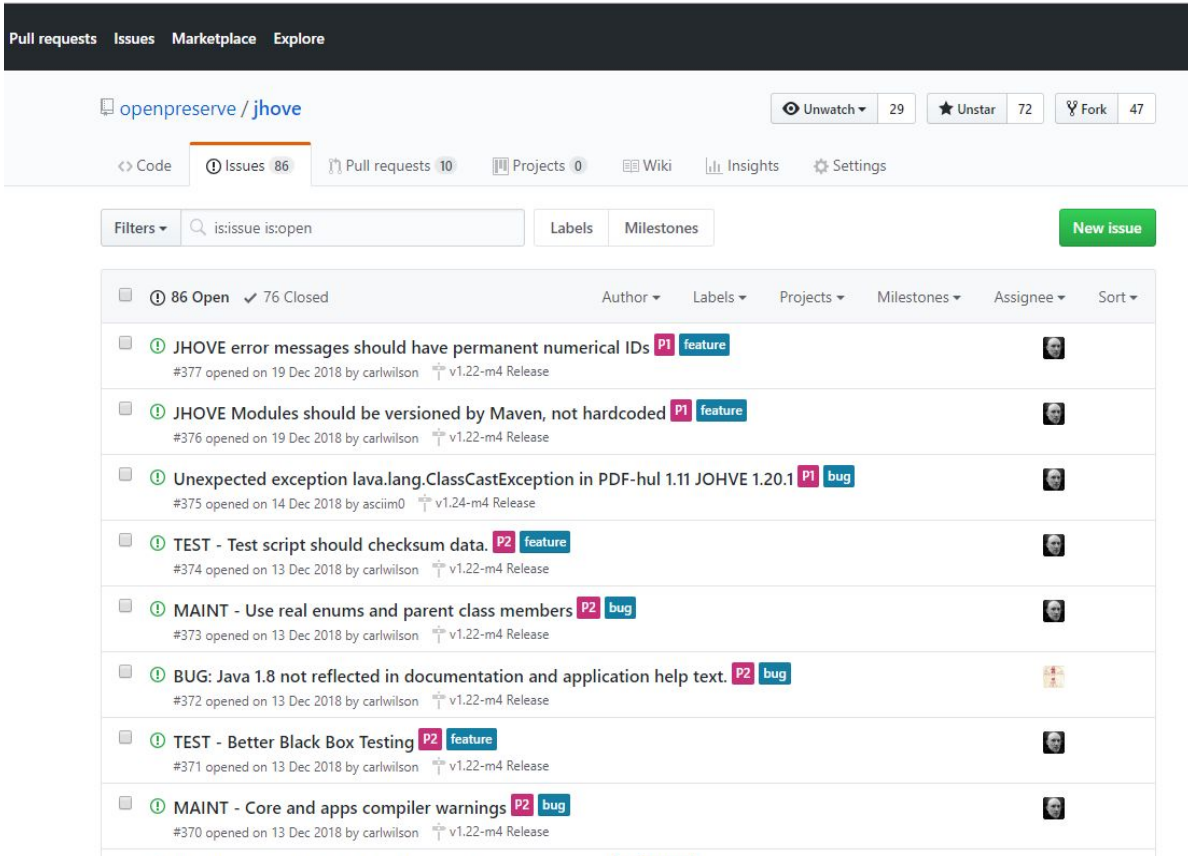
- If there's something wrong.
- If you have an idea about something that can be improved.
- If you'd like to see a new feature.

Contribute an Improvement

You can also submit a contribution as a GitHub Pull Request: https://github.com/openpreserve/jhove/pulls

- This can be a fix for the code base
- It could be improvements to the documentation or website
- It could even be a whole new module

Issue triage



Role of the OPF Product Board

- Steer the overall strategy for OPF products
- Prioritise development effort for each release

Release plan

Product	Jan	Feb	Mar	Apr	May	Jun	Jul	Aug	Sep	Oct	Nov	Dec
fido											●	
JHOVE		●										●
veraPDF						●						●
jpylyzer											●	

A look at the JHOVE roadmap and questions

The Future of JHOVE

The Future of JHOVE

- Better process for module versioning
- Assign IDs for error messages
- Instrumentation/monitoring/measurement
- JHOVE webservice REST API
- Standardise underlying reporting mechanism
- Java version increment
- Updating JHOVE test corpus / error reporting coverage
- Strategies for user updates and major changes (pre major changes in 1.24)

Resources

- http://jhove.opepreservation.org/
- http://lists.openpreservation.org/listinfo/jhove (user mailing list)
- https://github.com/openpreserve/jhove

- @openpreserve
- info@openpreservation.org
- http://openpreservation.org/

Feedback

Please help us improve our training