



# Open source implementation of PDF/UA validation



Extending veraPDF to syntax checks of Tagged PDF and PDF/UA-1

# History of veraPDF

- Initially a part of the EU-funded PREFORMA project (<http://www.preforma-project.eu/>)
- Industry supported PDF/A validation
- Implementation advised by the Validation TWG of PDF Association
- Initial EU-funded period Oct 2014 - Dec 2017
- Continued support by Dual Lab and Open Preservation Foundation
- Maintenance releases at least once a year
- Accessible via <https://verapdf.org/> or <https://github.com/veraPDF>

# Validation scope for PDF/UA-1

- Normative documents:
  - ISO specification: 14289-1:2014
  - Matterhorn protocol 1.02
- Human vs Machine verifiable (as specified by Matterhorn)
- Current validation scope:
  - Only explicit requirements of ISO 14289-1
  - Only Machine verifiable
  - Disclaimer: no full ISO 32000-1 validation, no semantic correctness checks

# Grammar based validation

- Formalized document model of linked objects and their properties
  - Does not necessarily match the PDF object model
- Parser to retrieve values of the objects
- Grammar to specify the requirements as Boolean tests on the object properties
- Object oriented
- Different parsers per flavor (=the standard) for optimization
  - Some objects required for PDF/UA validation can be safely ignored in case of PDF/A

# Validation profile: example

```
<rule object="SETable">
  <id specification="ISO_14289_1" clause="7.5" testNumber="1"/>
  <description>If the table's structure is not determinable via
    Headers and IDs, then structure elements of type TH shall
    have a Scope attribute
  </description>
  <test>useHeadersAndIdOrScope == true</test>
</rule>
```

- Total 90 rules covering all Machine checks of PDF/UA-1

# Ambiguities and riddles

- Annotations: shall always be included into the structure tree? Even PrinterMarks?
- Language of Outlines and Metadata? What if the document is multilingual with no default Language?
- Does ActualText replace all children of the Structure element, so that no further checks of children are required?
- Shall all tables be regular?
- How to provide alternate description for interactive forms with multiple widgets?

# New corpus as a ground truth

- 285 new test documents
- Both pass and fail tests
- Atomic
- Self-documented
  - In almost all cases the documentation is in the outlines
  - Language tests place the documentation into the page contents
- Available at: <https://github.com/veraPDF/veraPDF-corpus/>



# Do we have a consensus?

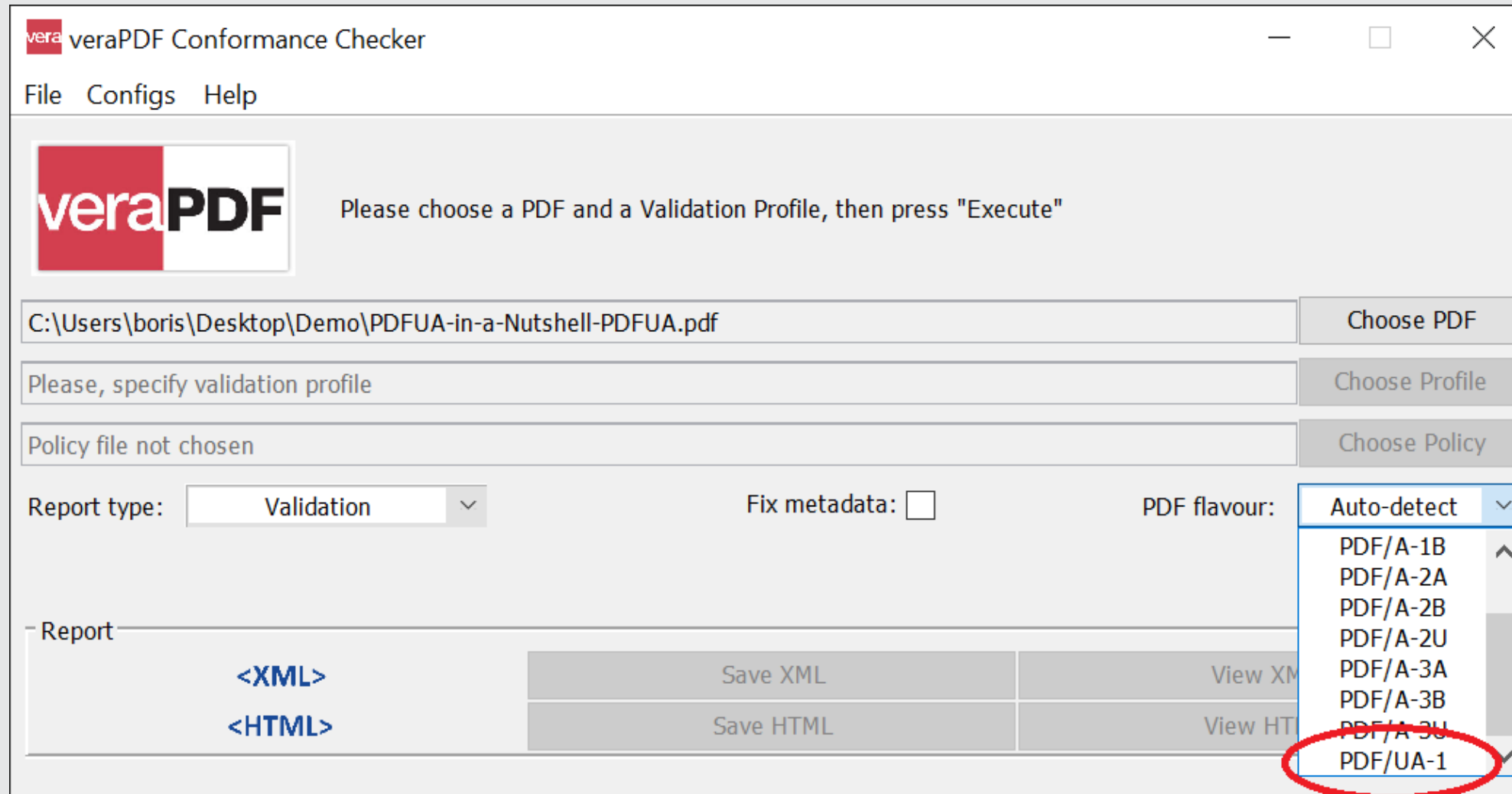
- Review of test files in progress
- Stats for PAC3, callas pdfToolbox and veraPDF
  - 85 files (30%) where PAC3 and pdfToolbox disagree
  - 33 files (11%) where PAC3 and veraPDF disagree, mainly around
    - font requirements
    - forbidden Unicode values
  - 65 files (22%) where pdfToolbox and veraPDF disagree, mainly around
    - the requirements for logical structure in ISO 32000-1
    - the cases when Lang is not present in the Catalog



# True open source

- Source code is at <https://github.com/veraPDF>
- Permissive licenses: GPLv3+, MPLv2+, CC (for the test corpus)
- Pure Java implementation
- Desktop, CLI, API, Web demo
- Issues reported and discussed at GitHub
- Tricky cases brought to the Validation TWG
- Modern CI practices and integration tests

# Desktop version



The screenshot shows the veraPDF Conformance Checker desktop application. The window title is "veraPDF Conformance Checker". The menu bar includes "File", "Configs", and "Help". The main area features the veraPDF logo and the instruction: "Please choose a PDF and a Validation Profile, then press 'Execute'".

The application has three input fields for file selection, each with a "Choose" button:

- PDF file: C:\Users\boris\Desktop\Demo\PDFUA-in-a-Nutshell-PDFUA.pdf (Choose PDF)
- Validation profile: Please, specify validation profile (Choose Profile)
- Policy file: Policy file not chosen (Choose Policy)

Configuration options include:

- Report type: Validation (dropdown)
- Fix metadata:
- PDF flavour: Auto-detect (dropdown menu is open, showing options: PDF/A-1B, PDF/A-2A, PDF/A-2B, PDF/A-2U, PDF/A-3A, PDF/A-3B, PDF/A-3U, PDF/UA-1. The PDF/UA-1 option is circled in red.)

At the bottom, there is a "Report" section with buttons for "<XML>", "<HTML>", "Save XML", "View XML", "Save HTML", and "View HTML".

# Visual report preview



← BACK TO SUMMARY

## PDF/UA in a Nutshell – Accessible documents with PDF

### Errors overview

- > A link does not have an alternate d... 86 i
- > TrueType (OpenType) font subset in... 1 i
- > Artifact is included into real content 1 i
- ✓ Table contains more than one head... 1 i

Page 10: 1 of 1

- > In a table not organized with Heade... 8 i

suitable program, or indirectly generated by adapting an existing PDF document. The indirect approach tends to require a great deal of work, as all tags and numerous other settings will need to be provided manually.

This work can also become void as soon as a new version of the PDF document replaces the old one, if any changes to the document's content need to be made. The

better option; post-creation PDF edits should be avoided or at least kept to a minimum. Either way, it is essential that the document creation program can perform the functions required and that the document creator can make use of them. Although only a few programs currently support PDF/UA in its entirety, a wide range of options still exist in the form of the programs listed alphabetically below:

Table 1: PDF/UA creation tools

| Software                 | Developer               | Application   | PDF/UA functions supported  |
|--------------------------|-------------------------|---|---|
| Adobe Acrobat XI Pro     | Adobe Systems           | PDF document creation, editing and viewing  | <ul style="list-style-type: none"> <li>• Create and edit content tags</li> <li>• Mark page content as artefacts</li> <li>• Add alternative text</li> <li>• Specify the language(s) used for a document and specific content within it</li> <li>• Feature to add basic accessibility to any PDF</li> <li>• Fast, extensive accessibility checking</li> </ul>   |
| Adobe Distiller Server 8 | Adobe Systems           | Converts PostScript files to PDF  | <ul style="list-style-type: none"> <li>• Create tagged PDFs</li> <li>• Requires specific pdfmark codes within the PostScript files</li> </ul>   |
| axesPDF for Word         | xyMedia                 | Add-in for Word 2007 and Word 2010. Helps the user to use Word functions and compensates for weaknesses and errors resulting from exporting a tagged PDF from Word. | <ul style="list-style-type: none"> <li>• Specify the language of a document</li> <li>• Set up a series of tabs based on the document structure</li> <li>• Specify the document title and include it in the document window</li> <li>• Use all standard PDF tags</li> <li>• Advanced table tagging</li> <li>• Tag all content or mark it as an artefact</li> <li>• Integrated validation function</li> </ul> |
| CommonLook               | NetCentric Technologies | Plug-in for Word and PowerPoint. It uses  | <ul style="list-style-type: none"> <li>• Guided review of document based on checkpoints</li> </ul>  |

- 10+ people submitting code modifications
- 20+ people submitting test cases and issues
- Often no information on these people except their GitHub names
- Libraries and archives via Open Preservation Foundation
- Latest download stats:
  - 9000 downloads of the latest veraPDF release 1.16 since May 12, 2020
  - 20000 visits to the dev version of veraPDF at <https://veraPDF.org> since May 2020
  - *Unknown count of people* accessing Java code directly via GitHub or Maven repo

# Future directions

- (Now) profiles to validate parent-child relationship in PDF 2.0
- (Future) work in progress for the combined PDF 2.0 and PDF 1.7
- (Future) WCAG 2.1 gap analysis and additional validation profiles
- (Future) Human checks interface + prototype implementation
- Key challenge: making validation results
  - Understandable to a non-PDF professional
  - Actionable for PDF creators

# About Dual Lab

- Product development services with heavy focus on PDF Technologies
- Applications to workflow automation, 3D engineering, NLP and other science-intensive technologies
- Operating since 2008 with offices in Belgium and Belarus
- Recent trendy directions:
  - From Desktop to Web and Mobile
  - Cloudification + containerization
  - AI and NLP for PDF

# Questions



**Boris Doubrov <[boris.doubrov@duallab.com](mailto:boris.doubrov@duallab.com)>**