

Enhancing File Format Standardization

Introducing the MPEG Systems File Format Conformance Framework

Dimitri Podborski

File Format Conformance Framework

Overview

- Necessity of File Format Conformance in WG3
 - The approach taken by the File Format group
 - The contribution process for new files
- Functionality of the framework
 - Key operational features and workflow
- Conformance File Search Engine
 - Search capabilities: 4CC, type, syntax, features, etc.
- Evaluating Conformance Coverage

Necessity of File Format Conformance in WG3

- MPEG WG3 FileFormat group has around 15 different projects
 - Key Projects: ISOBMFF, MP4, NALUFF, HEIF, Timed Text, CENC
- Increasing Complexity of Specifications
 - Example: ISOBMFF 8th Edition, Spanning 322 Pages
- Many features not yet tested and new features might build up on top of those
- Key Enhancement Goals:
 - Elevate the quality of standards we develop
 - Identify and address issues early in spec development
 - Simplify developer experience for efficient file access

What does File Format group do about it?

WG3 FileFormat group mandates conformance files

- Starting from MPEG#140 (IIRC) all incoming technologies need to be supported by conformance files
- New Feature = New Conformance File
- Limitations of the Old Conformance Framework:
 - Features were summarized in excel sheets, lack of information etc.
- Necessity and Objectives of the New Conformance Framework:
 - Improve linkage of metadata with conformance files
 - Perform fundamental checks on conformance files
 - Assess and ensure comprehensive coverage of features etc.

Contributing Conformance Files

How MPEG members contribute conformance files now

- ! Make sure your files are small in size! (We only care about File Format features)
- Bring an input contribution to MPEG (as usual, but **give us necessary metadata**)
 - **SHALLs** for each file:
 - Short description & Copyright info
 - **SHOULDs** for each file:
 - User defined features (e.g.: taken from 14496_12/user_defined.json)
 - Associated files (if applicable). Files which are required for processing.
- Then the File Format group will move your files to GitHub repo
 - We will open a PR
 - We will ask you for further input if necessary

Functionality of the New Framework

Overview

- Use JSON for all Metadata
- Extract Metadata from Specifications
- Extract Metadata from Conformance Files
 - Require contributors to supply metadata when automatic extraction is not possible
- Mapping of Standard and Conformance Features
- Basic Verification Checks
 - Is there such a box in a spec? Are versions/flags ok? Registered at MP4RA? Etc.
- Web Interface for Navigation and Coverage Statistics

<https://github.com/MPEGGroup/FileFormatConformance>



AVOID CLONING LFS

GIT_LFS_SKIP_SMUDGE=1

How does it work?

Standard Features

```
data/
└── standard_features/
    └── 14496-12/
        ├── boxes.json
        ├── item_properties.json
        ├── ...
        ├── spec_info.json
        └── user_defined.json
    ...
```

```
item_properties.json

{
  "name": "Item Properties",
  "description": "Image item properties",
  "entries": [
    {
      "fourcc": "brnd",
      "itemtype": "descriptive",
      "type": "GeneralTypeBox",
      "description": "Item brand (type)",
      "containers": ["ipco"],
      "syntax": "aligned(8) class BrandProperty extends GeneralTypeBox ('brnd') \n{ }"
    }
  ]
}
```

type.json

Box entries for **type**

spec_info.json

Brief info about spec

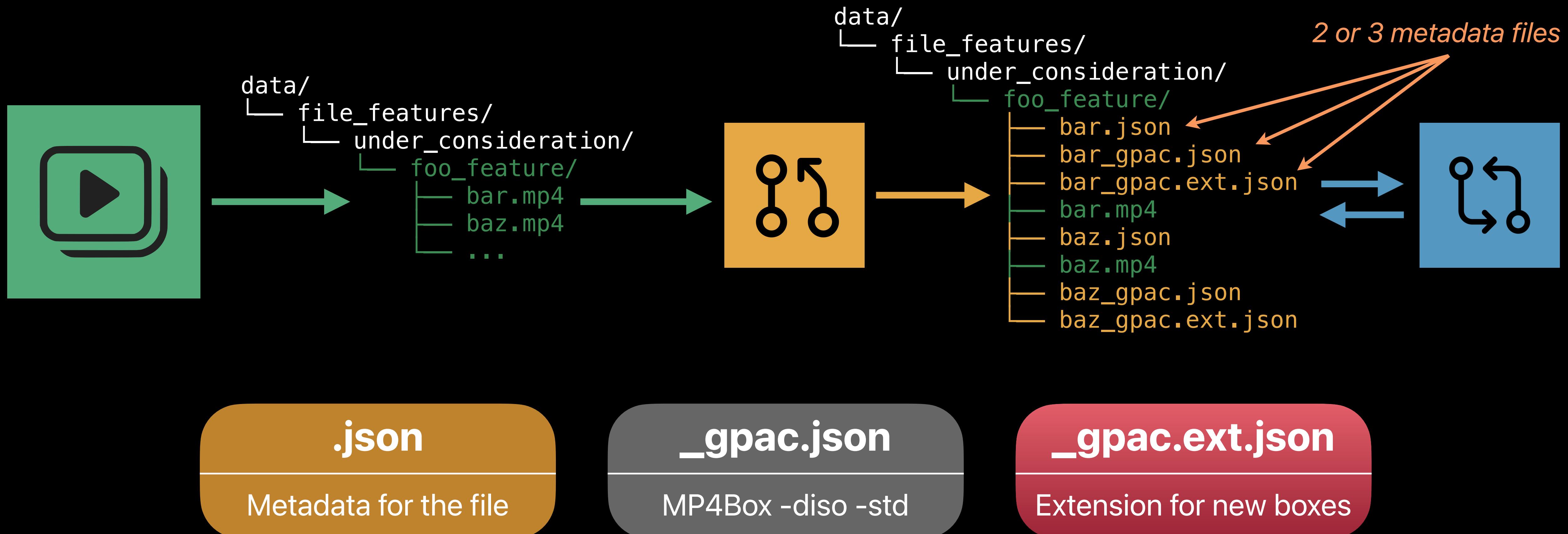
user_defined.json

User defined features

JSON schemas for these files are available on the repository

How does it work?

Conformance Files



JSON schemas for these files are available on the repository

Validations

Conformance Files

STANDARD

- Incomplete boxes
- Unknown types
- Unknown boxes
- Duplicate entries
- Conforms to the schema
- Standard is the super set of files
 - Versions
 - Flags

FILES

- Duplicate user defined feature
- [WARN] Unknown box path
- [WARN] Unknown user defined feature
- [WARN] Ignored files
- MP4Box extensions is satisfactory
 - No unknown box paths
 - Covers all unknown boxes from MP4Box
- Conforms to the schema
- No runaway files

PAGE

- E2E test for basic search
- Core search library functions
 - Normal search
 - Various of different filter configurations

MP4RA

- Check if there are 4CCs registered that we don't yet have in the suite
- Check if introduced 4CCs are already registered

File Format Conformance Framework

Search Page

File Format Conformance Framework

About Contributing Coverage Report

Start by typing a search term...

Add a filter

Note: You can use unix-style search operators. For example, type `=moof` to search exactly for `moof` box.

File Format Conformance Framework

Coverage Statistics

154 FILES **521** BOXES **52** FEATURES

5 specifications covered

- ISO/IEC 14496-12:2022 (**ISOBMFF**): 7th Edition
- ISO/IEC 14496-14:-1 (**MP4**): -1th Edition
- ISO/IEC 14496-15:2022 (**NALu**): 6th Edition
- ISO/IEC 14496-30:-1 (**Timed Text**): -1th Edition
- ISO/IEC 23008-12:2022 (**HEIF**): 2nd Edition



Try it out: <https://mpeggroup.github.io/FileFormatConformance/>

Fundamental Filters

Edit search query

trun

- Container ▾ Exact tr

+ Add a filter

Note: You can use unix-style search operators.

FOURCCS

- traf
- trak
- tran

TYPES

- TrackGroupTypeBox
- TrackReferenceTypeBox

Edit search query

trun

- Box Type ▾ Start by typing a type...

+ Add a filter

Note: You can use unix-style search operators. For example, type `=moof` to search exactly for `moof` box.

Edit search query

trun

- Registered Type ▾ Select one...

+ Add a filter

Note: You can use unix-style search operators.

- Select one...
- Boxes
- Codecs
- Entity Group Types
- Item Properties
- Item References
- Sample Groups
- Track Group Types
- Track References

Edit search query

trun

- Specification ▾ Select one...

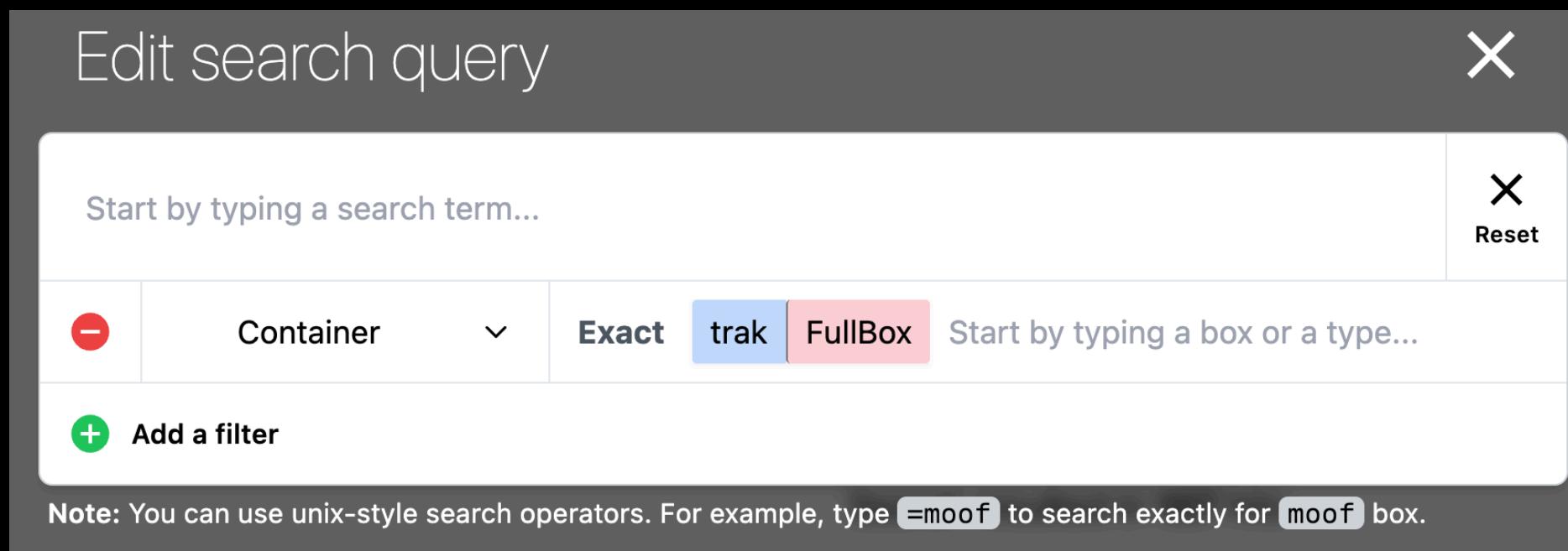
+ Add a filter

Note: You can use unix-style search operators.

- Select one...
- 14496-12
- 14496-15
- 14496-30
- 23008-12

Fundamental Filters

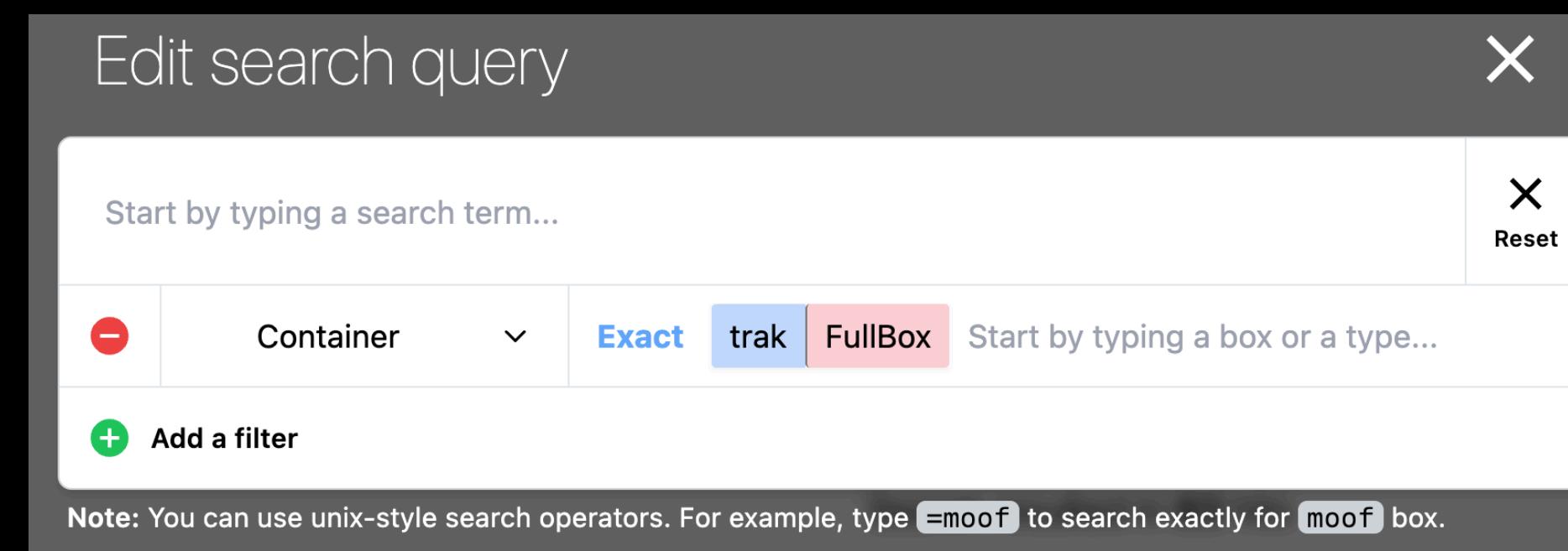
Container Filter



410 Results

```
"trak:Box": {  
    "meta:FullBox": {  
        "bxml:FullBox": {},  
        "dinf:Box": {  
            "dref:FullBox": {}  
        }  
    }  
}
```

Loosely follows the given container path



282 Results

```
"trak:Box": {  
    "meta:FullBox": {  
        "bxml:FullBox": {},  
        "dinf:Box": {  
            "dref:FullBox": {}  
        }  
    }  
}
```

Exactly follows the given container path

Matches only first descendants of the path

Further Filtering by Versions and/or Flags

FullBox • 14496-12

trun

track fragment run

This box can only be found under **trac** box.

✓

✗ Versions

Version 0

Version 1

✗ Flags

data-offset-present 0x000001
No description.

first-sample-flags-present 0x000004
No description.

sample-duration-present 0x000100
No description.

sample-size-present 0x000200
No description.

sample-flags-present 0x000400
each sample has its own flags, otherwise the default is used

sample-composition-time-offsets-present 0x000800
each sample has a composition time offset

Combined flags must match exactly (no extra flags).

> Syntax

FullBox • 14496-12

tkhd

Track header, overall information about the track

This box can only be found under **trac** box.

✓

✗ Versions

Version 0

Version 1

✗ Flags

track_enabled 0x000001
Track is enabled.

track_in_movie 0x000002
Direct part of the presentation

track_in_preview 0x000004
Deprecated

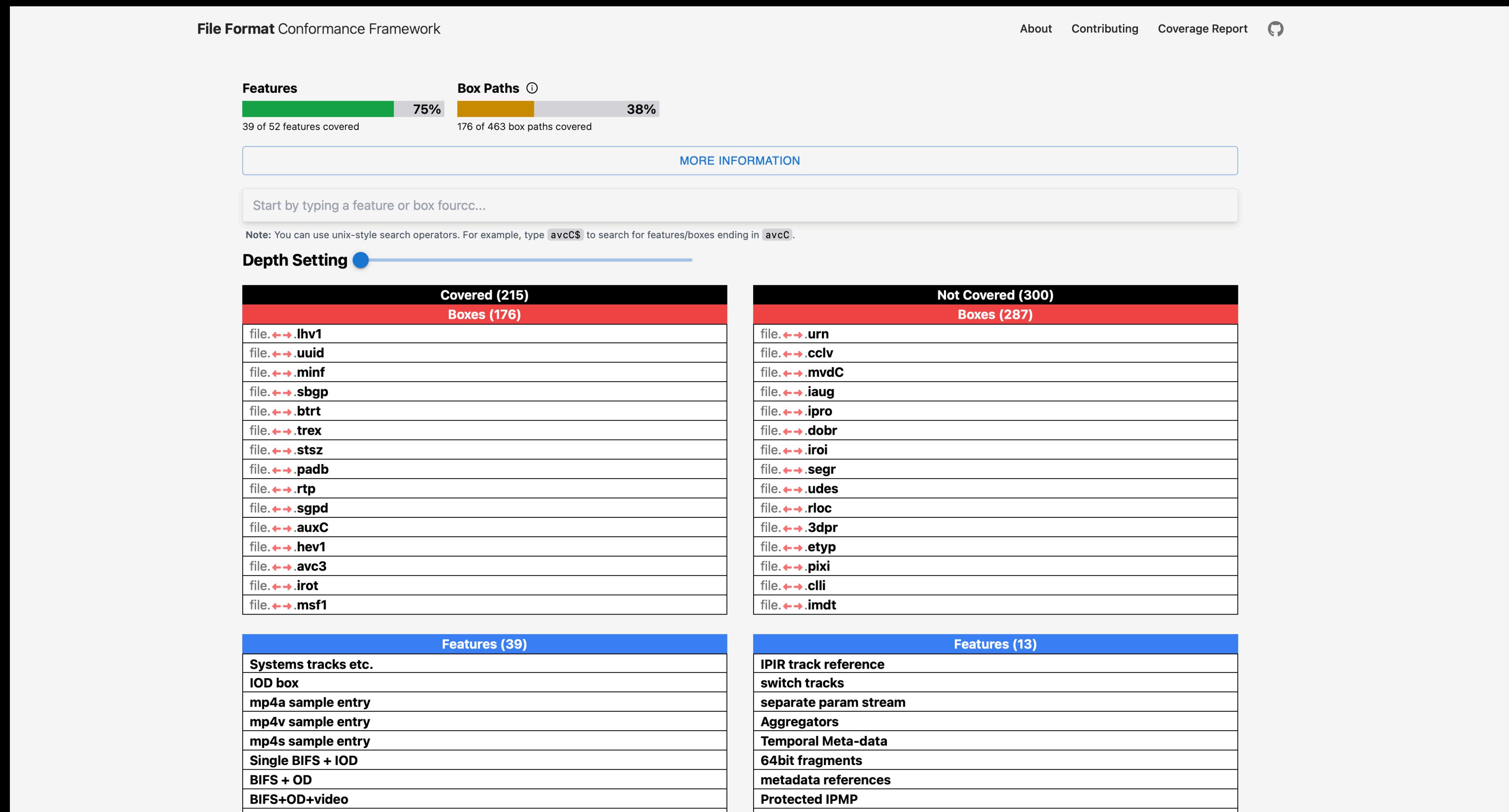
track_size_is_aspect_ratio 0x000008
No description.

Combined flags must match exactly (no extra flags).

> Syntax

Conformance Coverage Page

Not including 22 pending PRs with new conformance files from previous meetings!



Generic info

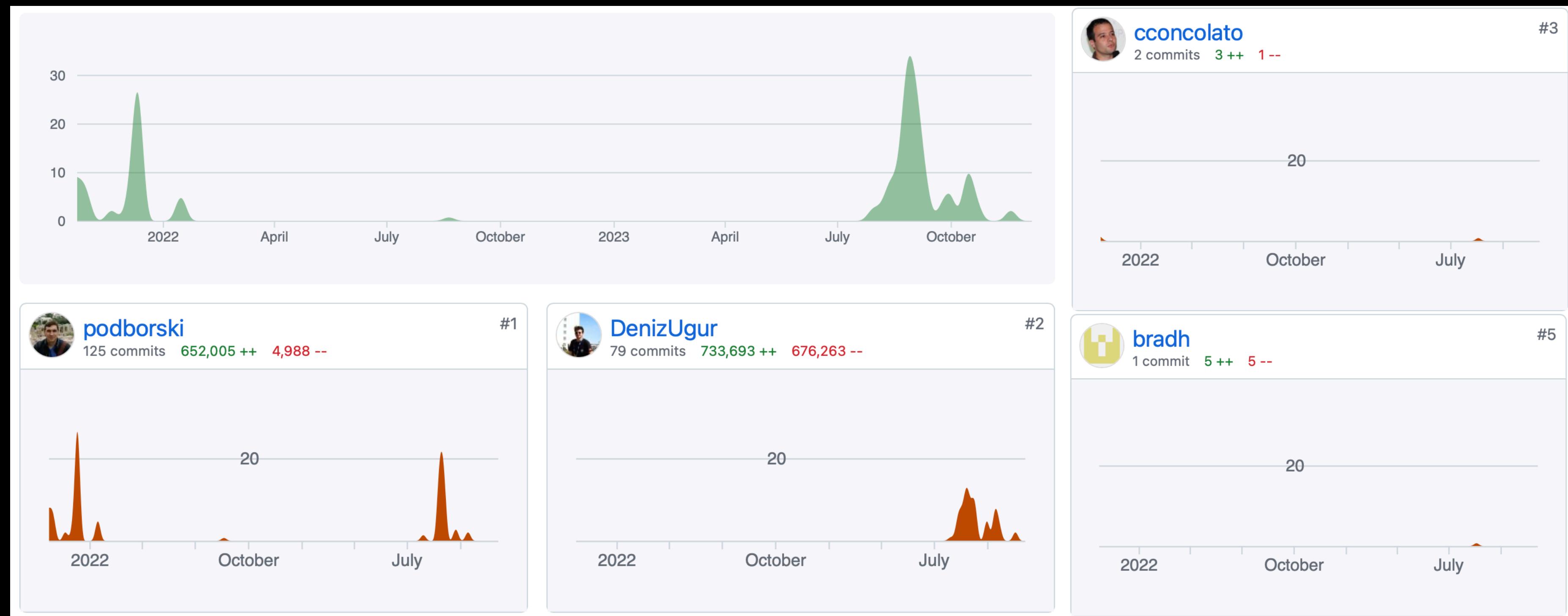
Coverage statistics

Search function

Depth setting

Contributors

Oct 24, 2021 – Dec 6, 2023



Special Thanks to Deniz Ugur for Outstanding Contributions

Thank you!
Any questions?