



Applications Working Group – IVOA Interop, April 2022

Tom Donaldson and Adrian Damian (chairs)

E-mail: apps@ivoa.net (subscribe at <http://mail.ivoa.net/mailman/listinfo/apps>)

Slack channel: [#applications](https://ivoa.slack.com) ([invitation link](#))

Support the development of software that utilizes VO services and data

- Standalone apps ([Aladin](#), [Topcat](#),...)
- Web apps ([ESASky](#), [Aladin lite](#), [TAPHandle](#), ...)
- Software libraries ([pyvo](#), [astroquery](#), [astropy](#))

Standards

- **MOC (Multi-Order Coverage Map)**
- VOTable – Primary IVOA format for exchanging tabular data
- SAMP – Simple Application Messaging Protocol
- HiPS – Hierarchical Progressive Survey



Standards Status

MOC 2.0 now official!

- Time coverage can now be included in Multi-Order Coverage maps
 - MOC 1.1 supported spatial coverage only
- **MOC support in Rust, Python, command line interface, etc.** (François-Xavier Pineau)
 - *Session 9, Wednesday @13:30UTC*
- Other resources
 - [MOCInfo page](#) on IVOA wiki
 - [MOC Server presentation previous interop](#) (Pierre Fernique)
 - Document on github: <https://github.com/ivoa-std/MOC>

VOTable 1.4 is current

- Multiple issues have been written up on github; some discussion
 - <https://github.com/ivoa-std/VOTable>
- Though not urgent, we should work on those issues towards a new version.



Applications session Wednesday, Apr 27 13:30

Speaker(s)	Title and Abstract
<i>Markus Demleitner</i>	Getting COOSYS ready for 2025 As the real epoch progresses away from the convenient J2000.0 that the VO was born in, being able to automatically apply proper motions to catalogues in VOTables becomes more and more important. Our existing COOSYS element is not quite sufficient for that, and while the Coords model will be, its annotation will be a VOTable-external standard. In this talk I will discuss what it would take to make COOSYS work for pure-VOTable, Coords DM-compatible epoch propagation.
<i>Francois-Xavier Pineau</i>	MOC lib Rust and its derivatives: MOCPy, MOCWasm, MOCCLI and MOCSet MOC lib Rust is a Rust implementation of the MOC 2.0 standard. Originally developed to improve MOCPy performances, it has grown as a standalone library natively used in CDS internal projects. Still at the very core of MOCPy, its functionalities are also available through MOCCLI and MOCWasm. MOCCLI is a command line utility, a simple executable file, pre-compiled for Linux, MacOS and Windows. MOCWasm is a JavaScript/WebAssembly library made to manipulate MOCs in Web Browsers. Finally, in order to fulfill VizieR needs, we have been developing MOCset. MOCSet is a command line utility for building, updating and querying a set of MOCs. Its memory footprint is possibly low.
<i>Omar Laurino</i>	Supporting Prototype Implementations in PyVO PyVO implements the IVOA standards. As part of the standard approval process, new features are proposed and need to be demonstrated before the standard may be approved. PyVO may implement features that are not yet part of an approved standard. Such features are unstable, as the standard may be subject to reviews and significant changes, until it's finally approved. The new `prototype` package provides support for such prototypes.



Hackathon Sessions – Overview

Thursday 28 April 2022, 13:30 (60min) & 15:00 (90 min) UTC

What is it?

- Informal time to collaborate on (small) programming or design projects
- Bring project ideas or join someone else's project
- Mix of expertise may help with your project

Hack project ideas being gathered on [this document](#)

- You don't need an idea. Show up with an open mind and join an interesting project.
- You don't have to work on your own idea. Feel free to join other groups.



Hackathon - Logistics

Communication

- Meet in [Gather.town](#) (not Zoom!)
- Projects can create their own collaborative pads ([yopad](#), [Google doc](#), etc.)
- [Session program page](#) will be updated with latest information

Agenda

- Welcome and short intro
- Pitch ideas (1 min each)
- Group ourselves and get to work
- Not much time, so pick smallish projects, and if you like:
 - prepare ahead of time (at least have a look around gather.town)
 - work during break(s)
 - arrange with collaborators to continue after the Interop

Summary will be presented at IVOA closing

- Present one slide summary of what you achieved, or planned, or failed at, ...