

# whatrecord: A Python-Based EPICS File Format Tool

Ken Lauer <sup>(1)</sup> – klauer@slac.stanford.edu

1. SLAC National Accelerator Laboratory, USA



## WHAT IS IT?

whatrecord is a Python-based parsing tool for interacting with a variety of EPICS file formats. The project aims for compliance with epics-base by using Lark grammars that closely reflect the original Lex/Yacc grammars.

whatrecord aims to answer the question “where did *that* come from?” with its context-annotating tools and integrated shell script interpreter.

## FORMATS

Parse any of the following into intuitive Python dataclasses using lark:

- Database files (V3 or V4/V7), database definitions, template/substitution files
- Access security configuration files
- Autosave .sav files
- Gateway pvlist configuration files
- StreamDevice protocol files
- snlseq/sequencer state machine parsing

## IOC SHELL STATE INTERPRETER

Interpret IOC shell scripts (i.e., st.cmd) and track:

- What files were loaded during startup?
- What records are available?
- What errors were found?
- What file and line did record X get loaded?
- Inter- or intra-IOC record relationships

This powerful feature can help you understand an IOC.

## OTHER THINGS

- Try it out easily with docker-compose
- Makefile introspection + graph output
- GDB Python script for introspecting IOC binaries
- True-to-EPICS macro handling with epicsmacrolib (PyPI/conda-forge)
- Plugins for TwinCAT PLC projects (symbol -> source + PV), LDAP info, ...
- And some other things...

## ACKNOWLEDGMENTS

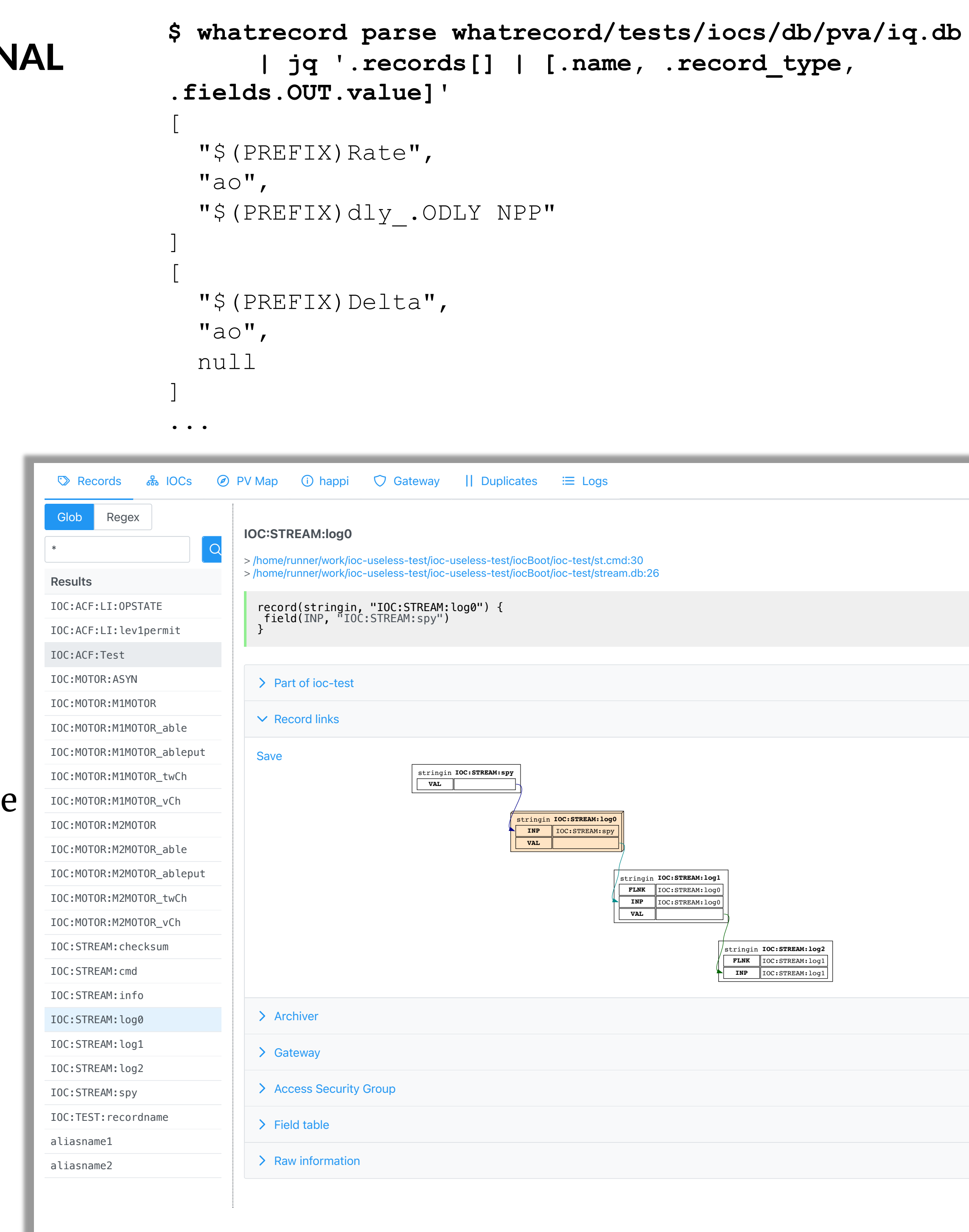
SLAC National Accelerator Laboratory is operated by Stanford University for the U.S. Department of Energy Office of Science. Work supported by U.S. D.O.E. Contract DE-AC02-76SF00515

## PARSE / INTERPRET EPICS FILES IN THE TERMINAL

```
$ whatrecord parse whatrecord/tests/iocs/db/pva/iq.db
| jq '.records[] | [.name, .info["Q:group"]]'
...
[
  "$(PREFIX)Phase:I",
  {
    "$(PREFIX)iq": {
      "phas.i": {
        "+type": "plain",
        "+channel": "VAL"
      }
    }
  }
]
...
```

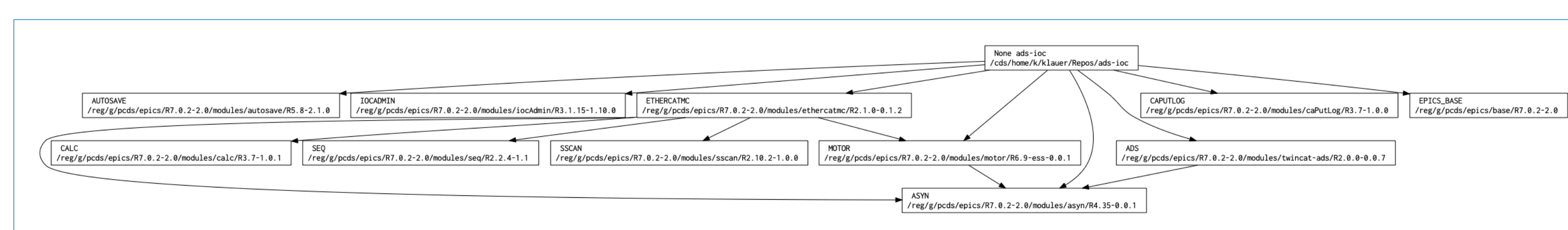
## OR IN YOUR BROWSER

- **Frontend:** Vue.js-based frontend single-page application to search for records/IOCs/etc by name and dig into the details
- **Backend:** API server to monitor IOC scripts and serve IOC/record information
- Load and monitor scripts/files for changes and provide REST API for querying
- Alternatively, static site export (see GitHub Actions Example [2])

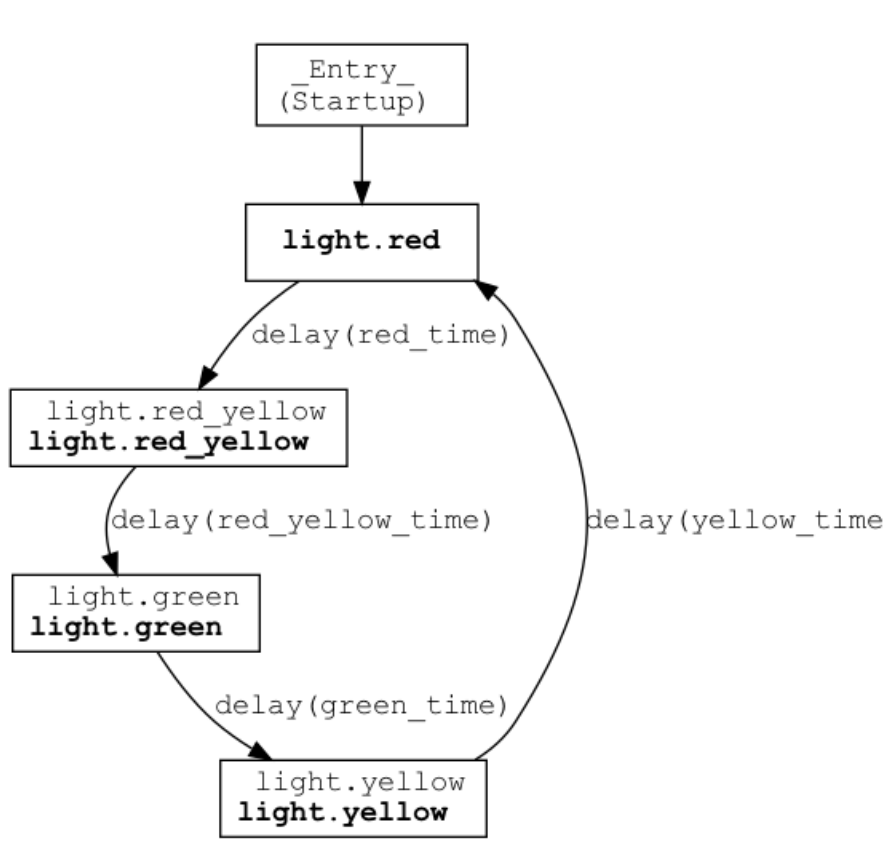


Vue.js frontend sample

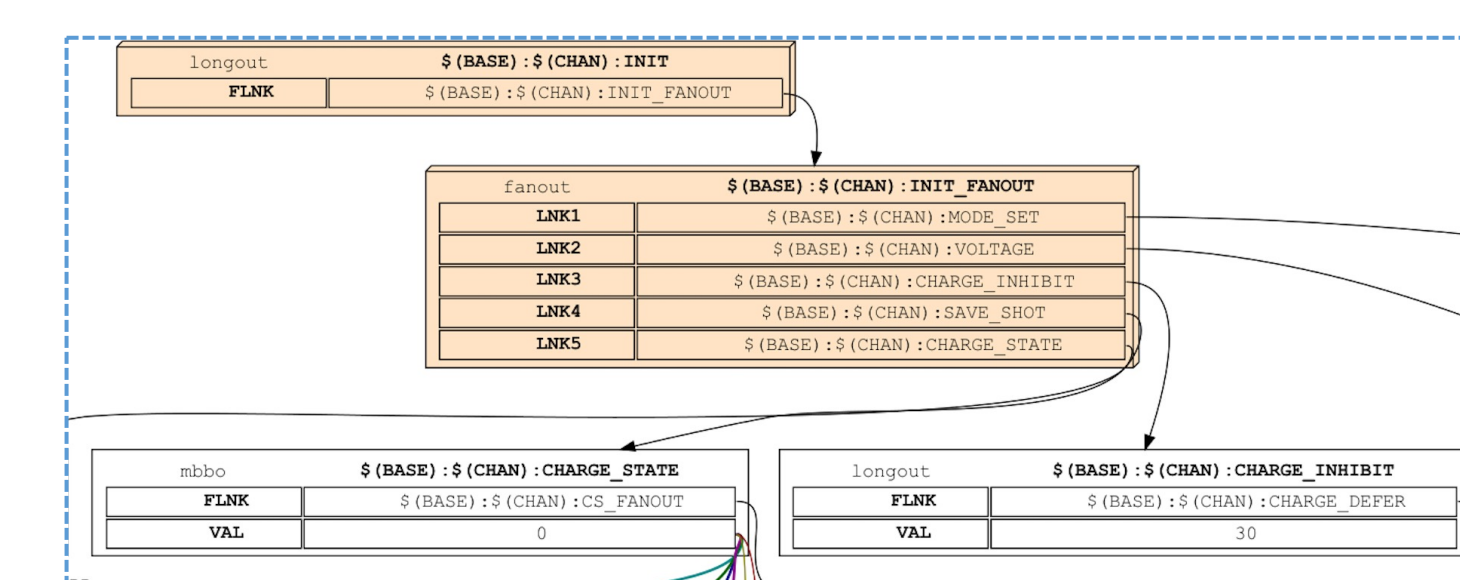
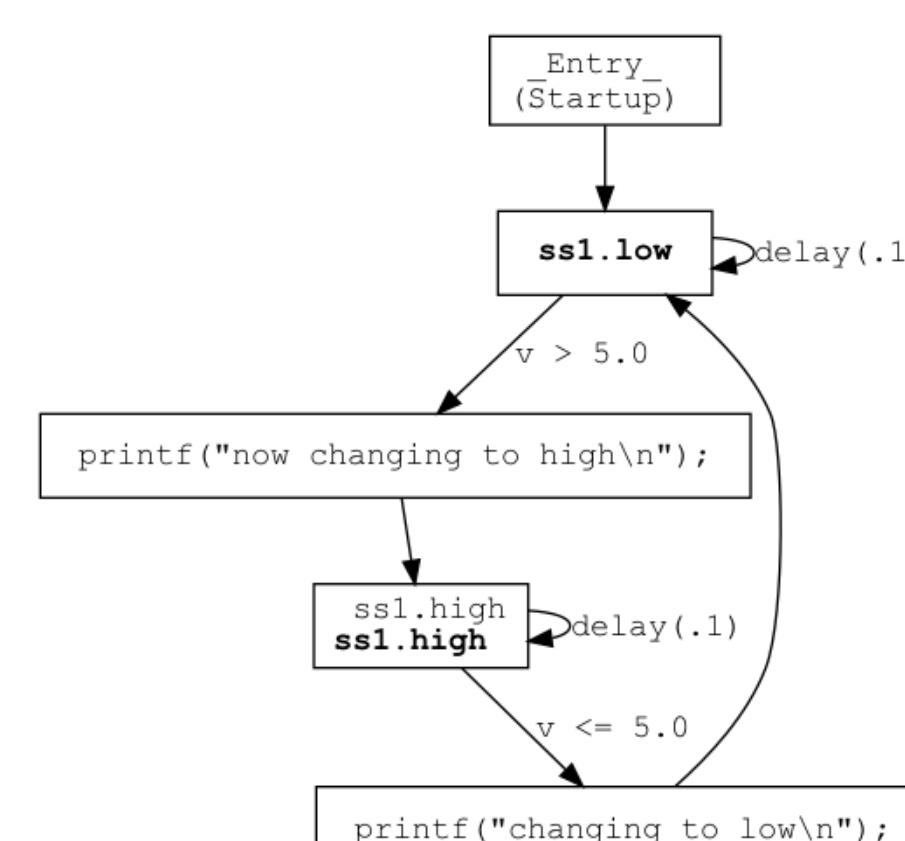
## MAKE GRAPHS... AND TRY TO UNDERSTAND YOUR IOC



Makefile introspection for dependency information



Sequencer State Notation Language support

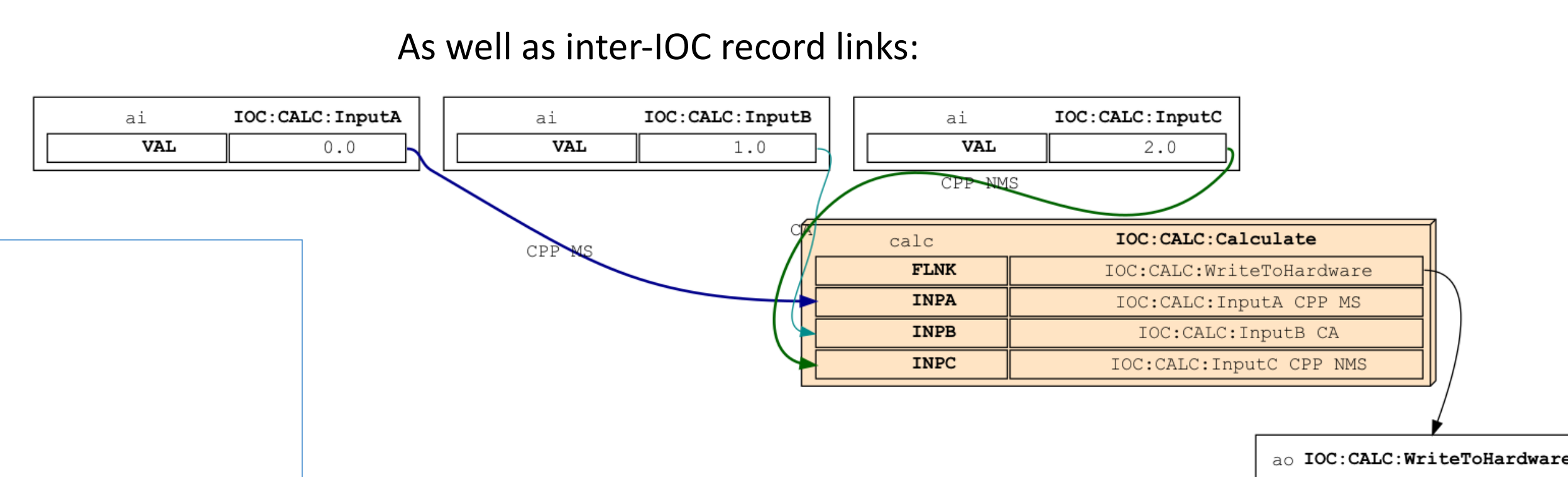


Intra-IOC record links

## EBNF GRAMMAR SNIPPET

This is what the grammars look like:

```
database: record*
record: "record" record_head record_body?
record_head: "(" string "," string ")"
record_body: "{" record_field* "}"
record_field: "field" "(" string "," string ")"
| "info" "(" string "," string ")" -> record_field_info
| "alias" "(" string ")" -> record_field_alias
```



1. Development on GitHub: <https://github.com/pcdshub/whatrecord>
2. Sample pages deployed by GitHub Actions: <https://klauer.github.io/ioc-useless-test>
3. Lark: <https://github.com/lark-parser/lark/>