

- Scope
- New Features
- Features that could be improved
 - ACS shipping
 - Performance enhancement
- Features that may exist but not known or documented, or not exist yet
- ACS technologies you'd think needs a replacement (please indicate the reason: obsolescence, security, ...)
- Enhancing the ACS experience

Scope

- Discuss the ACS wish-list from the community.
- Input from a google form and discussions during the workshop.

New Features

- Native support of OPC UA
- Support of TAI for timestamps
- Integration with a database for more than 32 items history for an attribute
- Fully Java BACI support
- Fully Python BACI support
- Structure BACI properties (even if basic / with limitations)
- Simpler CDB:
 - The CDB XML interface feels very clumsy and not well documented.
 - Is there a way to give beginners a simpler file-based CDB for testing which is maybe based on something like YAML, TOML, JSON, INI or so? I guess the XML was chosen, so that the schemata can be validated, but is that something beginners even need?
- Add security features:
 - Authentication and authorization
 - Session binding
 - Encryption
 - ...
- Modification of omniorb to include python command-line help documentation.

Features that could be improved

ACS shipping

- Slim-down ACS dependencies - e.g. Python, Python Libraries, ...ACS & Python
- Can we get ACS to use the System Python instead of shipping their own
- Use stdlib Python logging, while ACS just hooks an "ACSHandler" into the stdlib python
- Enumerations in IDL, they do not get properly translated into Python Enums (If possible, likely not)
- Improving the installation process via graphical UI. The end would be working ACS incl. an intro etc.
 - The docker and vagrant are steps toward that
 - Not high priority, but would invite more participants.

ACS tools

- getTemplate improvement: The interface of the "getTemplate" CLI tool is very strange. No tab completion, no help.

Performance enhancement

- "acsStart" and "acsStop" optimization. They take forever, this should not need so long. (I was told this is due to a bunch of timeouts strewn all over the ACS codebase, which are not necessary.)
 - Also, why does it take so much memory?
 - 1st start of ACS needs to have at least 3-4 GB (TB)
- "getDynamicComponent" is a blocking call, that waits for the remote component to be actually loaded. This is not necessary, it could just create the "objref" and return right away independent on how long the component needs to start. Can we fix that?
- Better handling of attributes subscription, particularly when client disconnect
- The creation of mockup components within the same process could be optimized - needs investigation.
 - Be able to create a test with mock-up components without the need to define IDLs.
 - Could try to use the Python Component simulator (TBC). Can be used without ILD or test.
 - Action item: have a look.

Features that may exist but not known or documented, or not exist yet

- How can we get the contents of the "acscommandcenter" in a Python? In particular, how can we start containers, stop containers, check which containers are running, from Python?

- The correlator experts use python for this via scripts/subprocesses, and directly in Python. Action item: Correlator Team provide an example, pointer
- An "acsAvancendClient" exists to cover the items above. Action item: Jorge provide example/pointer

ACS technologies you'd think needs a replacement (please indicate the reason: obsolescence, security, ...)

- Old-style makefile
- ACS build system
 - separate ExtProds from ACS core
 - Could come from the operating system or EPEL - someone needs to maintain this
 - Note: In CTA RPMs are provided, with different packages for ACS and ExtProds
- Use of modern unit-test frameworks instead of TAT
- DDS (see yesterday discussion)
- How long can we maintain Java SWING UIs?
 - ACS team Thinking to replace to pyQT
 - Included nice tool for graphing pyQTGraph
 - Issues - another dependency - will limit the libraries dependency
 - Alternative: web interface
 - ALMA and CTA looking at UI web-based technologies
- Not urgent, but alternatives to CORBA should be studied

Enhancing the ACS experience

1. Better documentation
 - a. Link with ticketing - will come very soon (couple of weeks).
 - b. Integrating with tutorials from this workshop.
2. Means of communication - Slack, Ticketing, ...
3. More frequent community calls
4. Better and more sophisticated examples