

General Hooks

Sardana Follow-up Meeting - 02.11.2017

Hooks

- Python code that will be executed at given points of the macro
- Can be a Python callable or a macro
- Hooks can be attached to the hook places that are defined by macros
- Macro must inherit from Hookable class to allow hook places
- Hook places are defined and handled by the macro in the code
- Exists two possibilities to attach hooks to the macros:
 - programmatically - using the hooks macro attribute (it requires defining a wrapper macro)
 - graphically - using the sequencer widget

Hooks - demo

- Demo will use macros provided in
`sardana.macroserver.macros.examples`
- `prdef loop`
- `prdef captain_hook`
- `captain_hook 0 10 1`

General Hooks

- General hooks would allow to define hooks by means of configuration e.g. environment variables - see [#200](#) and [#202](#)
- Derivative features: general stop function and general condition for repeating a scan point
- Set via already implemented macros
- Implemented at DESY and widely used:
 - [Documentation](#) (Scans -> Hooks -> General hooks, conditions and on_stop function)
 - Latest code private

Evolution of the General Hooks implementation

- Configuration based on an external file
 - Hooks could be python functions
 - General Hooks for scans only
 - File - another configuration mechanism with its own syntax
 - Why not to use MacroServer environment?
- Configuration based on the MacroServer environment variables
 - Macros were better received by the users - no more support to the functions
 - Warning messages if general hook execution fails
 - General Hooks still only for the scans
- Next:
 - Merge of Hooks and General Hooks into one concept? - just different configuration mechanisms
 - Decouple them from the scans and allow to use from any macros - move it to the core

How to proceed?

- Concentrate on the core implementation:
 - Agree on unifying Hooks and General Hooks into one concept
 - Develop tests e.g. reuse this [code](#)
 - Implementation:
 - Look how MacroExecutor deals with XML macros and hooks (see method `_prepareXMLMacro`)
 - Do the same but instead of hooks defined in XML it should work with the environment