
HowToMiddleLayer

Release 1.0

CAS

Jul 05, 2022

CONTENTS

1	Preamble	3
2	Reference	5
2.1	contributors	5
2.2	hot-develop	5
2.3	reduced support	5
3	Device Development	7
3.1	Process Definiton	7
3.2	<i>Important notes</i>	8
4	Macro Development	9
4.1	Process Definiton	9
5	Practical Aspects	11
6	Indices and tables	13

The below mentioned policies, guidelines and best practices cover the case where Karabo device development is contributed to from outside the DATA department, e.g. from instrument staff, and the contributions are regular and include a debugging and testing cycle driven by the contributor.

Contents:

PREAMBLE

For the remainder of this document we assume that all work is done in good faith, and that everybody involved is aware that they are operating in a sensitive environment, for which ensuring overall integrity is critical to operation of the facility. In practice, this means: if you are not sure about what your are doing, stop and ask, rather the proceed and try!

REFERENCE

2.1 contributors

People contributing to device development through code implementation in the Karabo SCADA framework, who are not members of the DATA department. A contributor needs to have appropriate software engineering skills in either C++ or Python that they are expected to generally be able to develop a devices, given sufficient access to Karabo specific concepts and documentation.

2.2 hot-develop

The process of developing on a control host in the live production system

2.3 reduced support

DATA, importantly including the DOC, will give support on restarting a device or a device server, but will only act on best effort on the specifics of the device, or the hardware it connects to. This importantly includes it's configuration parameters. If supported devices rely on a devices with reduced support, the support responsibility ends at the input of data from a reduced support device, or a the output of data to a reduced support device. Generally, support is to be given by the contributor of the device.

DEVICE DEVELOPMENT

3.1 Process Definiton

- The contributor(s) requests with CTRL to participate in this contribution scheme
- **A member of CTRL (most often ICI) discusses the request with the contributor in detail. Importantly, the following points should be clarified in a round where expertise specific to the project exists**
 - scope of contribution, e.g. what devices, what hardware, special requirements (connectivity)
 - level of support from CTRL anticipated during development: pair-programming, recurrent review, final review
 - caveats, rules of conduct, failure scenarios that could impact unrelated services
 - support contacts, under which the DOC can reach the contributor are defined
 - if there are considerable reservations by CTRL if the concept is feasible, these need to be addressed before proceeding with further steps.
 - TODO: add special cases which are especially critical, and need special consideration when being touched, e.g. pulse patterns, MPS, high data rate.
- **The CTRL contact identifies a suitable development host for the project and sets up infrastructure**
 - minimally, the device server name needs to indicate that this is a non fully DATA supported system
 - these hosts are normally excluded from CTRL driven deployment, but it is expected that contributors enable mandatory updates, e.g. by assuring their code is versioned.
 - Question: do we want a fixed set?
- **Access to the development host is provided to the contributor(s). General access policies apply, i.e. if framework modification are intended, no support is given to the service, if only devices are developed the framework installation is supported by DATA**
 - the CTRL contact trains the contributor in the gitlab system, code review and deployment tools, as well as how to hot-develop on the host
- **Contributors develop their project**
 - while under development, any devices running on the device server identified in (3) are not supported by DATA, advice and help is on a best effort basis
 - developers can ask for recurrent review, or pair development as agreed on in (2)
 - TODO: how to we assure that all operators are informed that the device in the reduced-support state.
- When the project stabilises, the contributor can choose to transfer ownership to CTRL

- Add a list of requirements? E.g. setup.py needs to work with deployment. PEP8. The devices has been successfully operated in the intended scenario.
- A timeline for the transition is agreed upon by CTRL and the author, after CTRL has gotten an initial overview of the device state
- a mandatory review by CTRL is made, and the contributor is expected to participate in the review process in a constructive fashion, it is the contributor who is responsible for following up on reviewer requests. Only after a successful review can the device be supported by CTRL. Ideally, another member of the contributor group joins this review.
- the device is tagged
- the tagged devices is migrated to a host / device server that does not have any support restrictions, and importantly properly deployed - no “dirty” tag!
- deployment, bug fixes and new tags are now created by CTRL. The contributor is encouraged to contribute, but they cannot hot-develop anymore
- at any time, should the contributor wish to hot-develop again, ownership is transferred back to them, support is restricted, and the device is moved to a device server / host which indicates reduced support. This restarts the cycle at (1) with the base for development being the last support version. This might also be necessary for a short time.

3.2 *Important notes*

- a contributed device running on a non-reduced support server is not supported at all.
- a device is either fully supported or in reduced support, not in both states
- if CTRL needs to take ownership of a device, because a device is e.g. needed by another instrument, it is expected that the original author facilitates the process
- if CTRL needs to take ownership of a device, because the author
- Devices which have a detrimental impact on services unrelated to itself, may be shutdown by CTRLs by the means they deem necessary, to preserve overall system integrity
- TODO: responsibility for restart
- TODO: framework deployments.

MACRO DEVELOPMENT

4.1 Process Definiton

- The contributor(s) requests with CTRL to participate in this contribution scheme
- **A member of CTRL (most often ICI) discusses the request with the contributor in detail. Importantly, the following points should be clarified in a round where expertise specific to the project exists**
 - scope of contribution, e.g. what devices, what hardware, special requirements (connectivity)
 - level of support from CTRL anticipated during development: pair-programming, recurrent review, final review
 - caveats, rules of conduct, failure scenarios that could impact unrelated services
 - support contacts, under which the DOC can reach the contributor are defined
 - if there are considerable reservations by CTRL if the concept is feasible, these need to be addressed before proceeding with further steps.

PRACTICAL ASPECTS

- **Importing code in gitlab**

- preserving authorship
- rearranging repos

- **trainings for people who would like to contribute**

- is this required?
- should include explanations for certain restrictions, and why to avoid them. Why to avoid...

Instruments contribution to Karabo: * Playground: * Have authentication to some extent. Adhoc development * Give some training, Should be on the email list to get up to date information related to karabo framework related changes. * How to scope the presentation : Review and train people and maintain use.

INDICES AND TABLES

- `genindex`
- `modindex`
- `search`