

Meeting Minutes -MXCuBE Developers code camp 12th- 13th of April 2020

The meeting was held virtually at whereby.com/mxcube between the 12th and 13th of October 2020.

Participants

Rasmus (GPhL)
Ivars (EMBL-HH)
Marcus, Antonia (ESRF)
Martin (Soleil) – part-time
Mikel (MAX IV) - part time

Issues

#580 Can you get a list of accessible transmission values?

At ESRF (at least) transmission is calculated from a look-up table, and interpolated between values for different energies. There are twelve blades that can be in or out. You can calculate which transmission you will actually get for a given setting, but a list of all possible values in advance is not practical. **Issue closed**

#572 Abstract Flux

Agreed to have a nominal dose rate function in Abstract Flux. **ACTION:** Rasmus Implemented by having `dose_rate_per_mm_squared_and_photon` (on advice from Gleb Bourenkov) since this is not dependent on dubious assumptions about the effective beam profile. **Done.**

#555 Units

It was agreed to keep a class level attribute giving the unit, for documentation, and to settle on a standard set of recommended units (mm, degrees, %, A, s, keV, pixels, ...) and put them in the contributing guidelines. More complex systems involving multiple units and conversion were discussed but not adopted.

#500 Lucid

It was discussed whether to add `lucid` as a requirement. If added, it should be ‘`lucid`’, generically, rather than a specific version as now imported. `lucid` is only used in two places: automatic centring (which is not necessary for the program to function, but you do get an error message in the log) and `QtManager` (which is, unfortunately) crucial to the Qt version. The issue will be reconsidered later, with Jordi. **Left Open**

#455 AbstractCollect #454 Data Classes

Will be considered together with `AbstractProcedures`, later. **Left Open**

#448 AbstractMachineInfo

In progress, with Lais. **Left Open**.

#579, #589 Naming conventions and use of properties.

It is agreed that names should preferably reflect the kind of object they name: function names should contain a verb, collections and maps should be marked as such, or plural, etc. This is a recommendation rather than a rigid rule, and will be added to the contributing guidelines.

It is agreed that for boolean attributes we should use positive rather than negative names (e.g. ‘`connected`’ rather than ‘`disconnected`’). It is recommended to have a single attribute with a single function, i.e. to use `set_connected(True)` or `set_connected(False)` (and *not* `connect/disconnect`).

In a parallel discussion, it is agreed that we should recommend using properties for attributes that simply return/change a value, do not cause hardware to change, where the change happens fast, and where additional parameters (e.g. switches, timeouts) are not likely to be needed. Again details will be put into the contributing guidelines. **Issue Closed**, but the matter must be kept in mind.

#515 re_emit values

It is agreed to introduce a new function, `force_emit_values`, that will emit signals for all values whether they have recently changed or not, for use from the UI. The existing `update_commands` will be kept with their current behaviour, and `re_emit_values` will survive as a way of doing multiple `update_commands` at once. **Issue Closed**

#555 specific states

Specific states have a well established mechanism, and there being on need or scope for harmonisation their contents can be freely chosen by whoever codes the object. There are no outstanding disagreements. **Issue Closed**

#527 snake_case v. camelCase

Ivars’ PR was accepted without further discussion. A short further round of bug fixes and additional case changes is expected. **Issue Closed**

Deprecated files

It was agreed to make a ‘`deprecated`’ directory for code that is no longer used but kept for information / possible later re-use.

‘attributeChanged’ signal

It was agreed to stick to ‘valueChanged’, ‘limitsChanged’ for the ‘value’ and ‘limits’ attributes, but to use ‘attributeChanged’ signals, giving both the name of the attribute and the new value, for other cases.

Python 2 v. 3

EMBL Hamburg expects to move to Python 3 as of January 2021 ca. ESRF is already on Python 3. It is expected that MXCuBE could move fully to Python 3 around the start of 2021, but there are too few developers at this meeting to make a final decision. Meanwhile Python 2 compatibility for site-specific code is not considered important.

Queue directory

It is agreed to keep QueueManager in the HardwareObjects directory (it is an HardwareObject) but to make a queue subdirectory for e.g. queue_entires, queue_model_objects, queue_model_enumerables, similar to what is done with sample_changer.

Version 2 and 3 configuration

It was agreed that mock object configuration ought to be shared between versions 2 and 3. Before the end of the meeting, Ivars had implemented the split, using a lookup path approach to distinguish v2-specific from v3-specific from shared files. **Done.**

Exceptions

It is agreed that we should *avoid* the construction ‘except BaseException’. Just like the bare ‘except:’, that catches *all* exceptions including greenlet exceptions and keyboard exceptions. Instead use ‘except Exception’, that does not catch the exceptions mentioned above.

If you for some reason need to catch absolutely **all** exceptions, e.g. in order to do something and then re-raise, you could consider using the Python exception hook. Failing that, use a bare ‘except:’ which will, quite rightly, be flagged by linters as potentially problematic.

Releases and planning

It was agreed to aim for a stable release with its own branch in the fairly near future, to be maintained separately from the master (development) branch.

First release

The first release should be versioned 3.0.0; EMBL-Hamburg and ESRF hope to start using it in production from around January 2021. It would include

1. Merging the current set of outstanding PRs
2. A round of general clean-up and minor refactoring, documentation and testing, till all tests run correctly.
3. Reorganising the directory and import structure, and making a Python setup.py script so that MXCuBE can be installed using standard Python installation procedures.

4. Configuring the GitHub CI so that the package(s) are rebuilt and pushed to PyPi or conda forge automatically for each tag.
5. Recommending that MXCuBE synchrotrons upgrade to the 3.0 series of releases.
6. Possibly moving to Python 3 (see above)
7. Aiming for a regular cadence of releases in this branch (e.g. every three months?).

Further work

Other major changes will be undertaken in the development branch, with a view to making a new version (3.1?) at a later date. These include:

1. Abstract Procedures and the Abstract Diffractometer
2. Data Classes
3. Development of AbstractCollect
4. A new DataPublisher.