Technical details

- Graphical access tot he MacroServer environment
- Improvement of the pseudo motor drift correction
- Graphical configuration tool for setting up Sardana

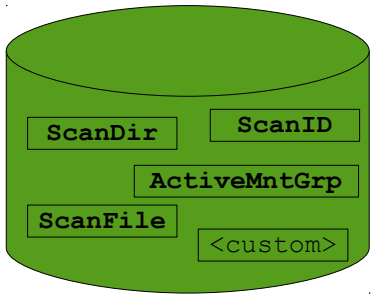
Access to the MS environment

Access to the MS environment

- **Motivation**: not user friendly access to the environment

Access to the MS environment

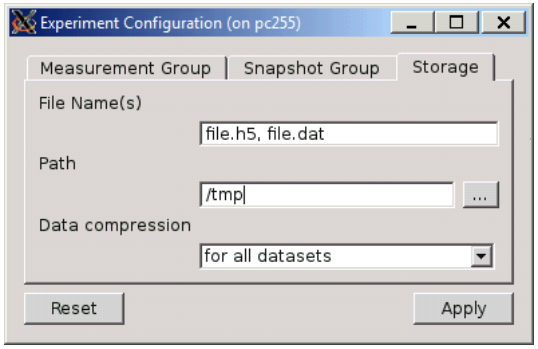
- **Motivation**: not user friendly access to the environment
- **Current situation**: env. stored in a map file, powerful API is ready→ env. manager, levels → global, door, macro, door.macro, Tango events with updates

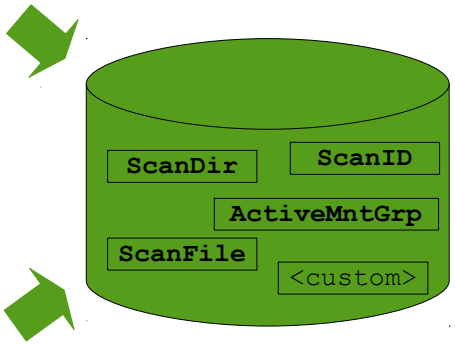


Access to the MS environment

- **Motivation**: not user friendly access to the environment
- **Current situation**: env. stored in a map file, powerful API is ready→ env. manager, levels → global, door, macro, door.macro, Tango events with updates
- **Current use:**

```
Door_1[1]: senv ActiveMntGrp mg01
ActiveMntGrp = mntgrp04
Door_1[2]: senv ScanFile '["file.h5", "file.dat"]'
ScanFile = ['file.h5', 'file.dat']
```

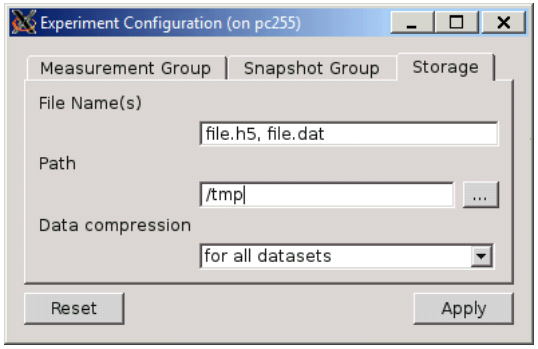


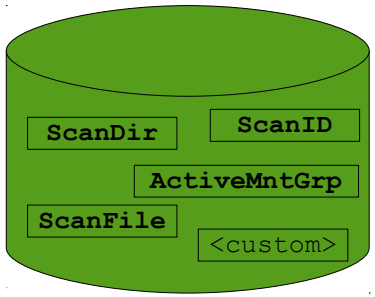


Access to the MS environment

- **Motivation**: not user friendly access to the environment
- **Current situation**: env. stored in a map file, powerful API is ready→ env. manager, levels → global, door, macro, door.macro, Tango events with updates
- **Current use:**

```
Door_1[1]: senv ActiveMntGrp mg01
ActiveMntGrp = mntgrp04
Door_1[2]: senv ScanFile '["file.h5", "file.dat"]'
ScanFile = ['file.h5', 'file.dat']
```





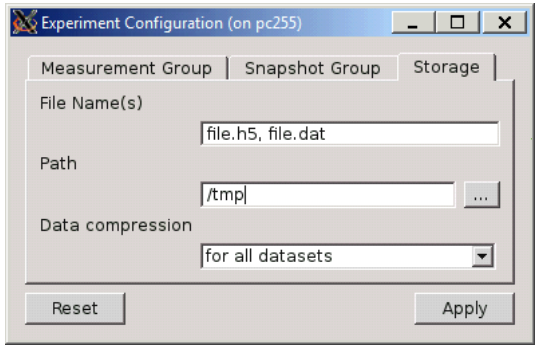

```
Door_1[2]: lsenv
                    Name          Value   Type
 -------------------- -------------- ------
                _SAR_DEMO   {'cont [...]   dict
            _ViewOptions   {'PosF [...]   dict
            ActiveMntGrp       mntgrp03    str
    DataCompressionRank              0    int
            JsonRecorder           True   bool
        PreScanSnapshot             []   list
                 ScanDir           /tmp    str
                ScanFile   ['test.dat']   list
             ScanHistory   [{'tit [...]   list
                  ScanID            649    int
```

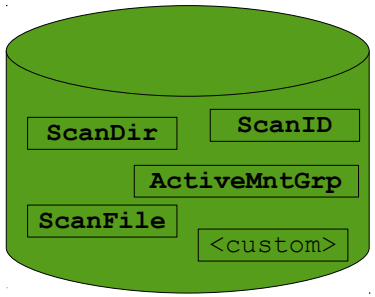
Access to the MS environment

- **Motivation**: not user friendly access to the environment
- **Current situation**: env. stored in a map file, powerful API is ready→ env. manager, levels → global, door, macro, door.macro, Tango events with updates
- **Future use:**

```
Door_1[1]: senv ActiveMntGrp mg01
ActiveMntGrp = mntgrp04
Door_1[2]: senv ScanFile '["file.h5", "file.dat"]'
ScanFile = ['file.h5', 'file.dat']
```

```
# Why not Spec-like macros
# e.g. newfile with ParamRepeat
Door_1[3]: newfile file.h5 file.dat
ScanFile = ['file.h5', 'file.dat']
```





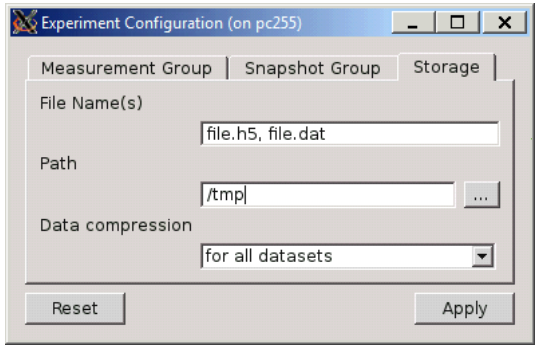

```
Door_1[2]: lsenv
                    Name          Value   Type
 -------------------- -------------- ------
                _SAR_DEMO   {'cont [...]   dict
             _ViewOptions   {'PosF [...]   dict
            ActiveMntGrp       mntgrp03    str
     DataCompressionRank              0    int
            JsonRecorder           True   bool
         PreScanSnapshot             []   list
                 ScanDir           /tmp    str
                ScanFile   ['test.dat']   list
             ScanHistory   [{'tit [...]   list
                  ScanID            649    int
```

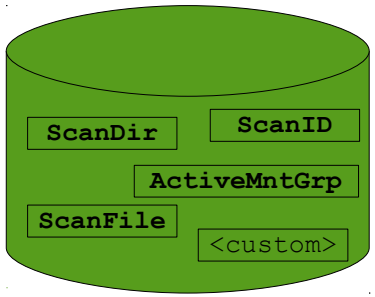
Access to the MS environment

- **Motivation**: not user friendly access to the environment
- **Current situation**: env. stored in a map file, powerful API is ready→ env. manager, levels → global, door, macro, door.macro, Tango events with updates
- **Future use:**

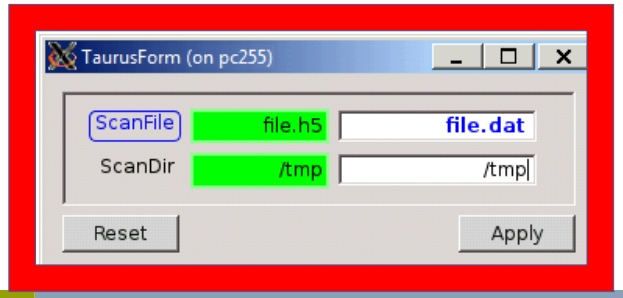
```
Door_1[1]: senv ActiveMntGrp mg01
ActiveMntGrp = mntgrp04
Door_1[2]: senv ScanFile '["file.h5", "file.dat"]'
ScanFile = ['file.h5', 'file.dat']
```

```
# Why not Spec-like macros
# e.g. newfile with ParamRepeat
Door_1[3]: newfile file.h5 file.dat
ScanFile = ['file.h5', 'file.dat']
```






```
Door_1[2]: lsenv
                    Name          Value   Type
 -------------------- -------------- ------
                _SAR_DEMO   {'cont [...]   dict
            _ViewOptions   {'PosF [...]   dict
            ActiveMntGrp       mntgrp03    str
   DataCompressionRank               0    int
           JsonRecorder            True   bool
         PreScanSnapshot              []   list
                 ScanDir           /tmp    str
                ScanFile   ['test.dat']   list
             ScanHistory   [{'tit [...]   list
                  ScanID             649    int
```



Taurus scheme: msenv

- Simple scheme to read/write environment variables:
  - **Authority → Tango DB; Value → list of all MSs.**

    Internally it should use proxy to Tango Database.
  - **Device → MS; Value → list of all variables.**

    Internally it should use proxy to MS Environment.

    Device should emit events when the env. variables are created, modified or deleted.
  - **Attribute → env. variable; Value → variable value**

    Write the specific variable e.g. ascan.ScanFile.

    Readout with fallback to the higher level e.g. ScanFile.

    Attribute should emit events when the env. variable is created, modified or deleted.
- More details in **SEP14 by Carlos Falcon**
- Other alternatives: Tango attributes for the env. variables?

Access to the MS environment

Questions? Comments?

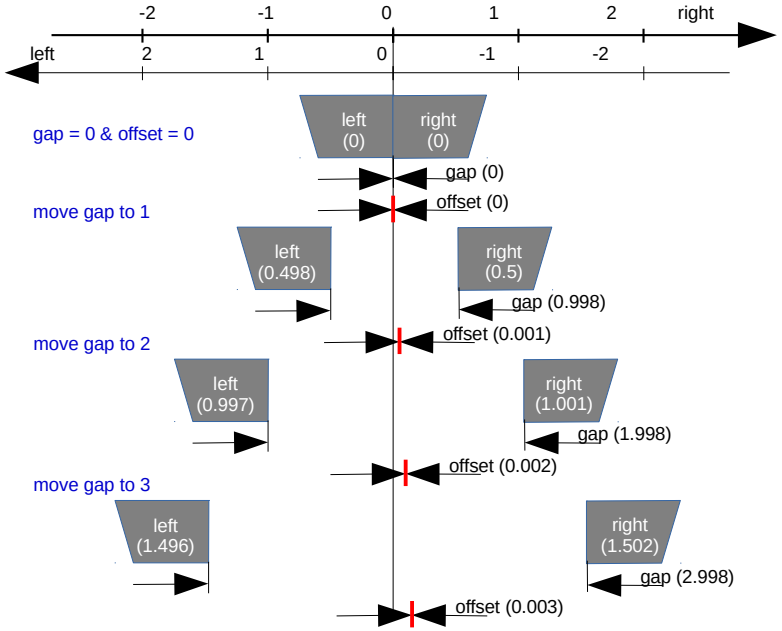
Drift of the pseudo motors

Drift of the pseudo motors

- Who is affected? Pseudo motors which have siblings and are based on physical motors with an **inaccurate or a finite precision positioning** system.

Drift of the pseudo motors

- Who is affected? Pseudo motors which have siblings and are based on physical motors with an **inaccurate or a finite precision positioning** system.
- Why does it happen? Each move of a pseudo motor requires **calculation of the physical motors positions in accordance with the current positions of its siblings**. A consecutive movements of a pseudo motor can accumulate positioning & rounding errors and cause drift of its siblings.



Drift correction

- Calculation of the new physical positions **use the write value, instead of the read value** of the siblings' positions, together with the new desired position of the pseudo motor being moved.

Drift correction

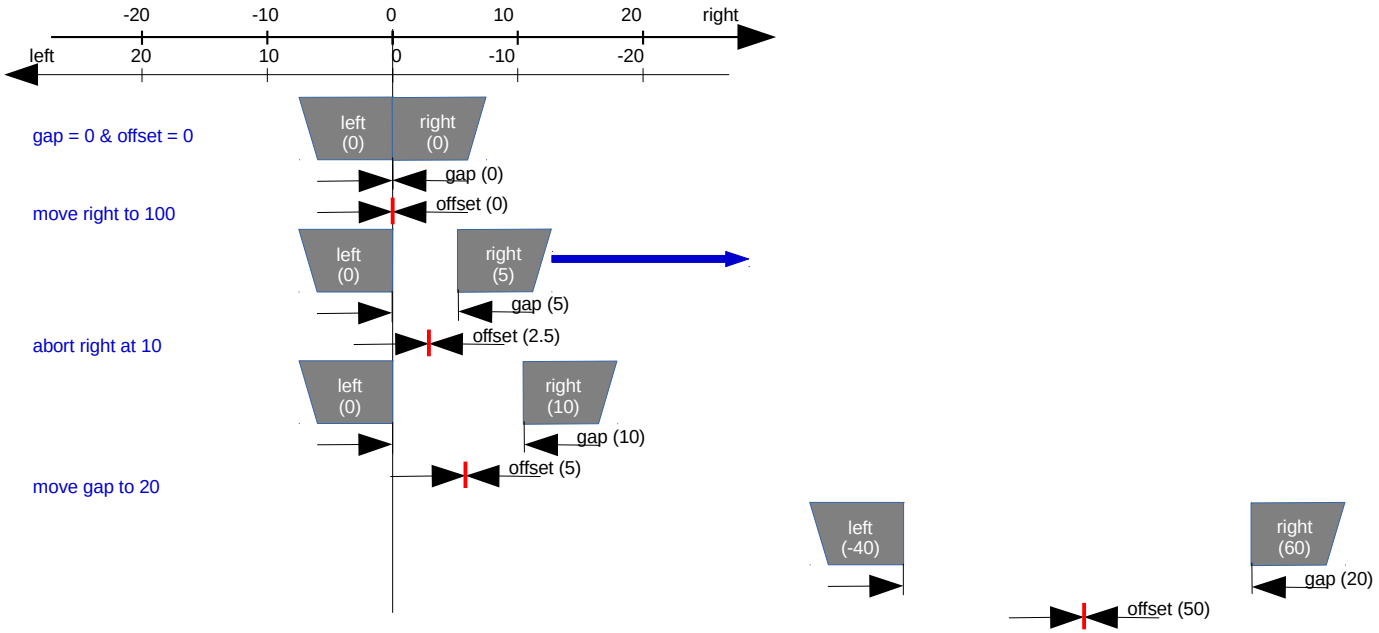
- Calculation of the new physical positions **use the write value, instead of the read value** of the siblings' positions, together with the new desired position of the pseudo motor being moved.
- The write value of the pseudo motor's position gets updated at each move of the pseudo motor or any of the underneath motors.

Drift correction

- Calculation of the new physical positions **use the write value, instead of the read value** of the siblings' positions, together with the new desired position of the pseudo motor being moved.
- The write value of the pseudo motor's position gets updated at each move of the pseudo motor or any of the underneath motors.
- Drift correction is optional but enabled by default.

Drift correction after abort

- Movements being stopped unexpectedly: abort by the user, over-travel limit or any other exceptional condition may cause considerable discrepancy in the motor's write and read positions. In the subsequent pseudomotor's move, Sardana will also correct this difference by using the write instead of the read values.



Drift correction - improvement

- Add third option to configure the drift correction:
  - drift correction enabled but applied only in case of the successful most recent movement (use write values only when last motion succeeded)
  - drift correction enabled (strict use of the write values)
  - drift correction disabled
- The *PoolMotion* action should set the **stop code** attribute for each motor object, to indicate how ended the last move e.g.
  - success
  - user abort
  - hardware abort (e.g. limit)
- More details in **feature-request #370**

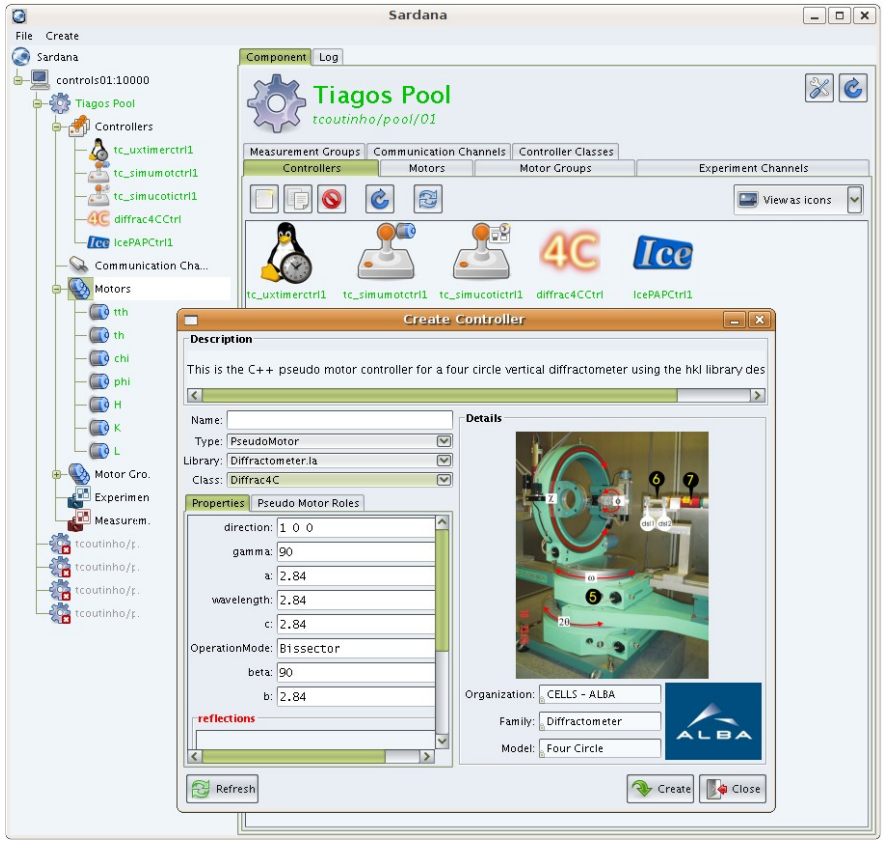
Drift correction

Questions? Comments?

Sardana configuration GUI

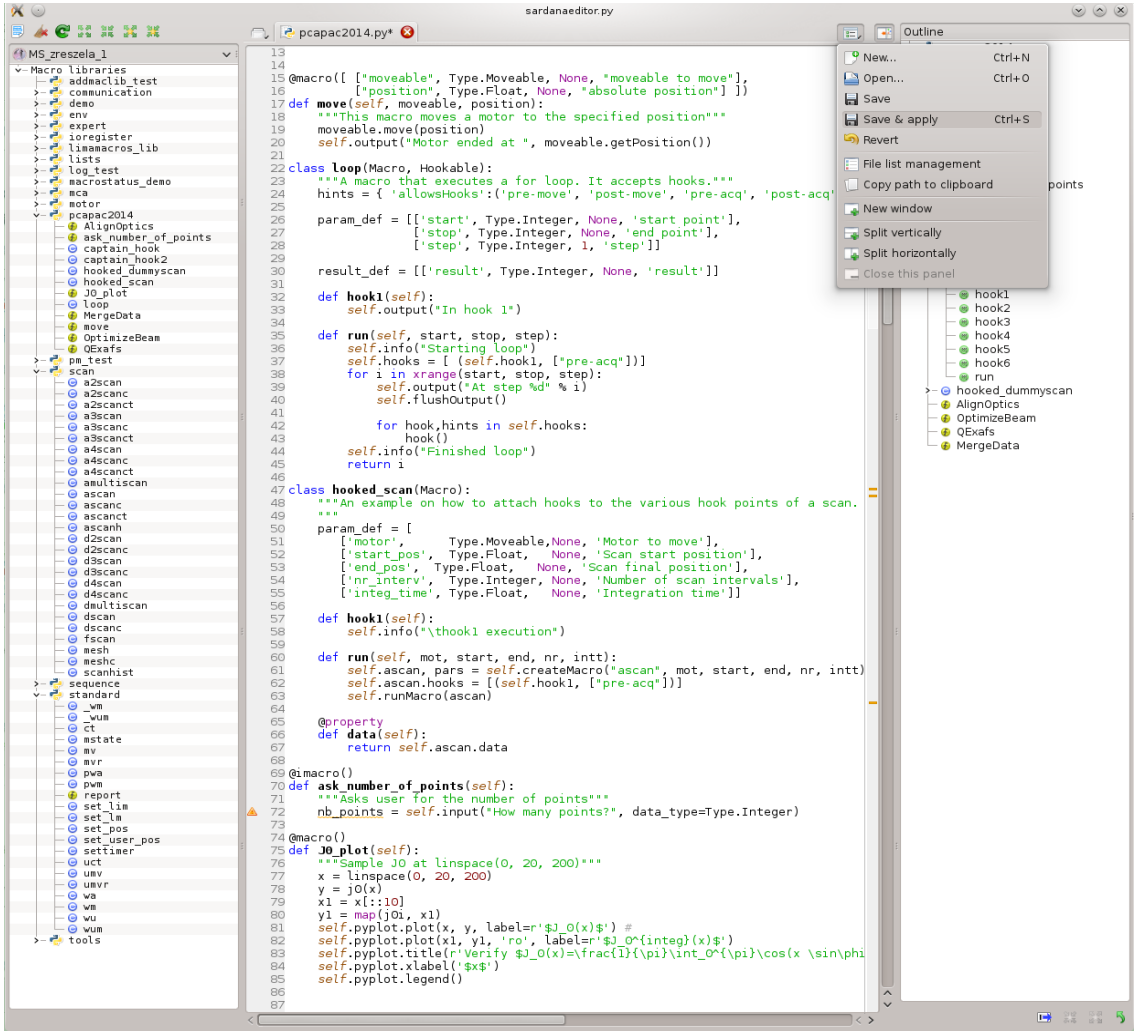
Sardana configuration GUI

- Initial Sardana IDE: written in Java, long time ago…
- OpenDocument Flat XML Spreadsheet: fods_to_sar, sar_to_fods
- Excel Spreadsheet: xls_to_sar
- Expert macros: defctrl, defelem, defm, defmeas, udefctrl, udefelem, udefmeas, sar_info



Sardana editor GUI

- Spyder based Sardana editor
- Expert macros:
  - addmaclib,
  - relmaclib
  - relmac,
  - edctrl,
  - edmac,
  - prdef
  - commit_ctrllib



Sardana configuration

- Known problems:
  - Pool accumulates motor groups – listeners of the position updates
  - Pool do not warn when deleting elements used by another elements (groups)

    e.g. a motor can be deleted even if a pseudo motor depends on it; an experimental channel can be deleted even if a measurement group uses it
  - measurement group can not be recreated from scratch (reusing the same name)
  - measurement group breaks when eliminating one of its initial experimental channels

Sardana configuration

- Known problems:
  - Pool accumulates motor groups – listeners of the position updates
  - Pool do not warn when deleting elements used by another elements (groups)

    e.g. a motor can be deleted even if a pseudo motor depends on it; an experimental channel can be deleted even if a measurement group uses it
  - measurement group can not be recreated from scratch (reusing the same name)
  - measurement group breaks when eliminating one of its initial experimental channels
- How do other institutes manage Sardana configuration?

Sardana configuration

- Known problems:
  - Pool accumulates motor groups – listeners of the position updates
  - Pool do not warn when deleting elements used by another elements (groups)

    e.g. a motor can be deleted even if a pseudo motor depends on it; an experimental channel can be deleted even if a measurement group uses it
  - measurement group can not be recreated from scratch (reusing the same name)
  - measurement group breaks when eliminating one of its initial experimental channels
- How do other institutes manage Sardana configuration?
- Do we need a Sardana configuration GUI before fixing the known configuration problems?

Sardana configuration GUI

Questions? Comments?