User and Dial Limits

- Options for limit protection:

  1- **Dial Limits associated with User Limits**

  - user_lim = sign * dial_lim + offset
  - new_dial_lim = old_dial_lim * steps_unit_old / steps_unit_new

  2- **Independent User and Dial Limits**

  - Offset and sign acting upon user limits
  - steps_per_unit acting upon dial and user limits

  NOTE: at the moment Dial Limits are not verified

Motors and PseudoMotors Limits

- **Motors not limited by pseudomotors limits**
  - Moving **individual motors** should not allow bringing **pseudos** outside their allowed range
- **PseudoMotors not limited by motors limits**
  - Moving **pseudos** should not allow bringing **individual motors** outside their allowed range

---

- The limit protection, as it is now, is not part of the Sardana core. Sardana reuses the Tango attribute range protection.
- The SardanaAttributeConfiguration class (range, alarm and warning) exists and each Sardana Attribute has an instance of it, but no logic uses it.

relmaclib & Co:

- **addmaclib**: loads new macro modules
- **rellib**: reloads a library that does not contain macros
- **relmaclib**: reloads a macro module
- **relmac**: reloads a single macro

- New behavior: **NOT load macro libraries if code is wrong**
- **After rellib usage → use relmaclib** to reload macros using the library
- Libraries with no macros:
  - Indicated in: OS PYTHONPATH or
  - Indicated in: PythonPath property of MS
- Macro libraries:
  - Indicate in: MacroPath

Error Traceability for reload macros

- **Objective**: Improve error traceability in reload macros

- **Current state**:
  - MS does not load erroneous modules at startup (e.g. modules with unexpected indent)
  - **addmaclib**: error traceback; library already loaded; macro overridden; successful load;
  - **relmaclib**: error traceback; signal successful reload;
  - **relmac**: error traceback; signal successful reload;
  - **rellib**: signal wrong usage to reload a macro library; signal successful reload of library;

- **Proposal:**
  - **relmaclib:** should not remove the macro library if there is an error in the module. It should warn the user about the error and leave the system in the state before calling the relmaclib (with the original library and macros)
  - **relmaclib:** should notify about macro overriding (of macros already existing in other modules)

MS_env of nested Macros

- When a Macro is executed inside another macro...
  - **MS environment variables can be defined at general scope or at Macro Scope. E.g:**

    senv ScanFile general.txt

    senv macroA.ScanFile myMacro.txt

  - **MS env variable of parent macro should set MS env of child. eg:**
    - ScanFile/ScanDir of 'parent' macro → ScanFile of 'child' macro
    - As today: if macroB executed inside macroA:
      macroA → myMacro.txt

      macroB → general.txt

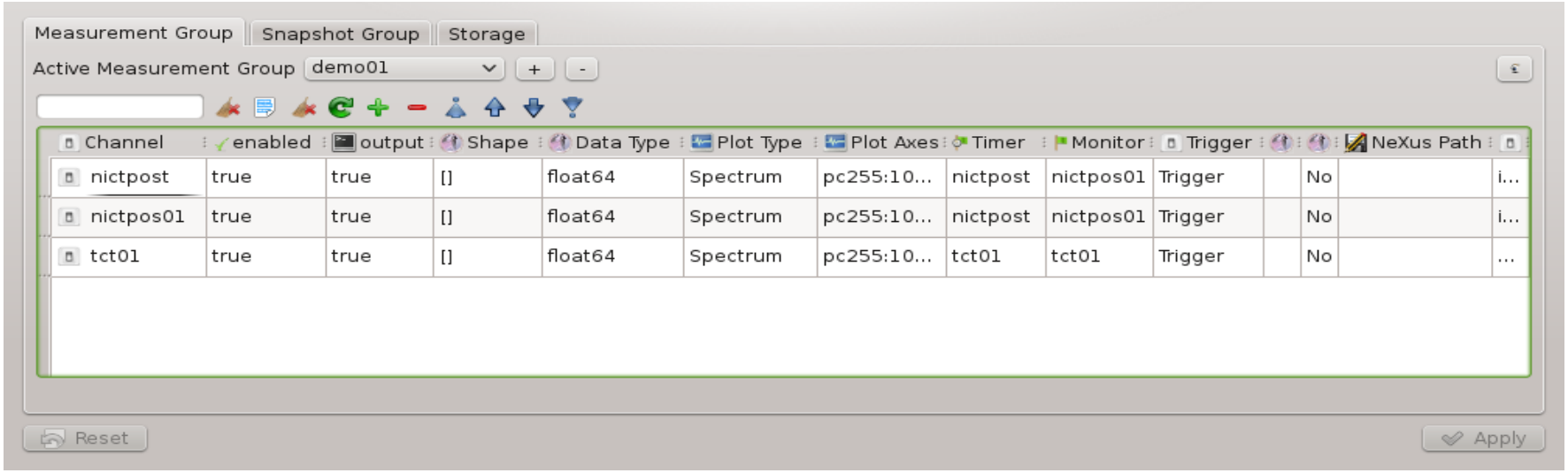
- Options:
  - **Set the correct variables in the Macro code (to be done each time...)**
  - **Create a hook associated with createMacro method**
  - **...**

Wish: Generic NeXus recorder

- **ALBA wish:**
  - Generalize the DESY NeXus Recorder
  - The Recorder could be used in all institutes of the Sardana/Taurus collaboration
  - Recorder could take:
    - One NeXus application definition in xml
    - Mapping dictionary
      - Key: data/metadata NeXus name to be used by the recorder
      - Value: Taurus model name (eg. taurus attribute of institution name)
  - Each institute could create its own Taurus Schemes to access the data in form of Taurus models to be stored in the NeXus file by the recorder (eg. a Taurus Scheme to access data stored in SQL databases)

Improve Experiment Configuration GUI

- Now:



- Possible Future:
  - **Plot/Trend: Line Color, Width, Symbol, etc.**
  - **Use expconf to define NeXus data organization:**
    - **Instrument definition used by NeXus recorder shall be improved (#388)**
    - **NeXus Path on expconf should override the path set by instrument (#236)**
    - **This feature would be deprecated by a general NeXus recorder**

Improve Experiment Configuration GUI

- Demands of some ALBA scientists:
  - Multiple clients should be refreshed on-the-fly when the experiment configuration changes
  - Non-interactive experiment configuration (without GUI: e.g. from spock or from macros)

- Others ?
  - **W**hich are your institution needs?