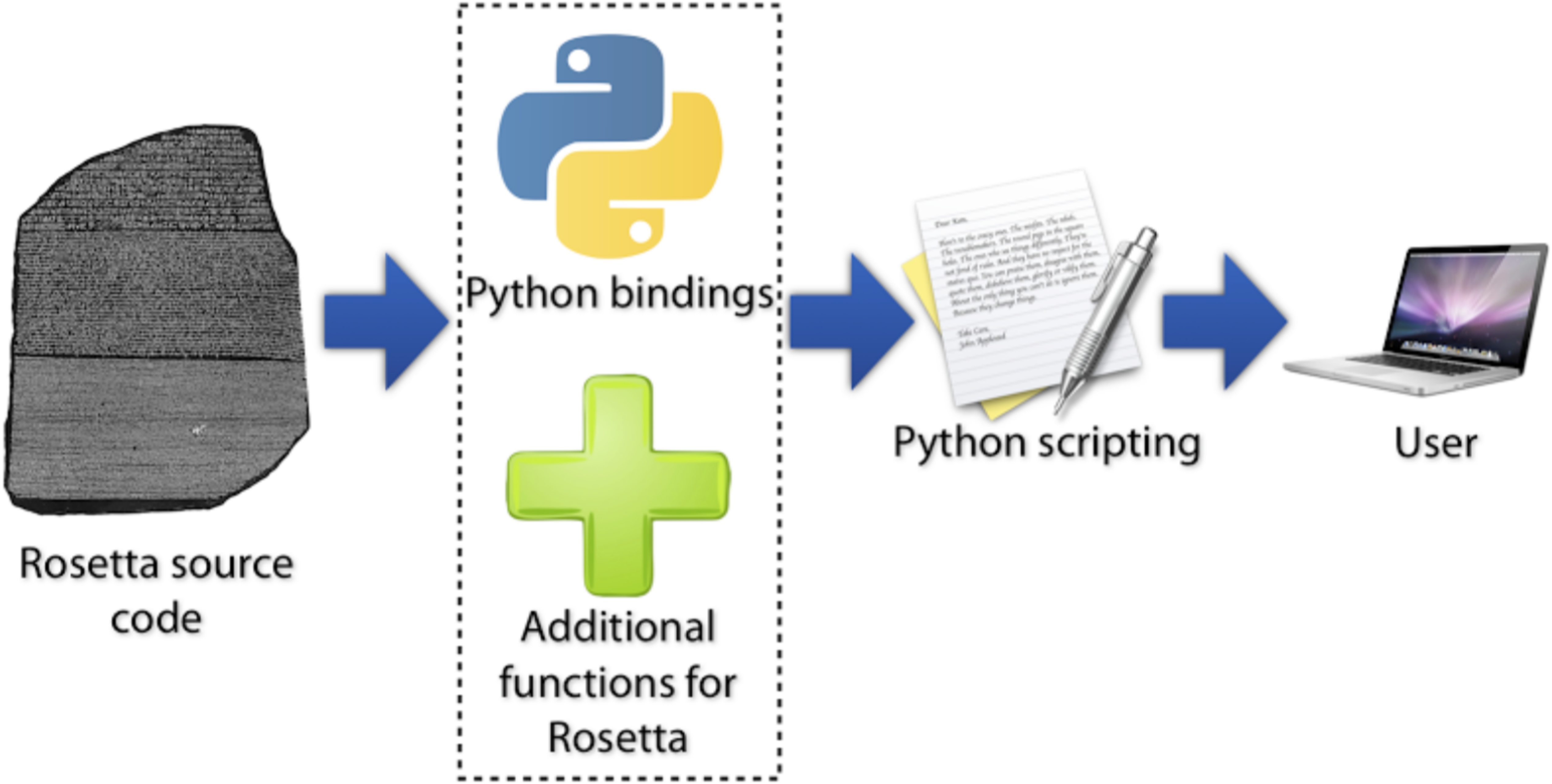
# PyRosetta-4

What’s new and why should you care!

```
% cd main/source/src/python/PyRosetta && ./build.py -j8
```

Sergey Lyskov, GrayLab@JHU



Rosetta source code

Python bindings

Additional functions for Rosetta

Python scripting

User

# PyRosetta-3

# PyRosetta-3

- GCCXML → XML
- Boost.Python
- Resulting code: C++98

# PyRosetta-3

- GCCXML → XML
- Boost.Python
- Resulting code: C++98

- Can only be built with GCC and GNU libstdc++
- Mac’s build is limited to use old version of GCC-4.1
- No support for C++11
- SLOOOOOOW

# PyRosetta-4

# PyRosetta-4

- ~~GCCXML~~

# PyRosetta-4

- ~~GCCXML~~

LLVM
COMPILER INFRASTRUCTURE

# PyRosetta-4

- ~~GCCXML~~
- Binder: tool for creating Python bindings for arbitrary C++ code.

LLVM

COMPILER INFRASTRUCTURE

# PyRosetta-4

- ~~GCCXML~~
- Binder: tool for creating Python bindings for arbitrary C++ code.
  - based on Clang libTooling

# PyRosetta-4

- ~~GCCXML~~
- Binder: tool for creating Python bindings for arbitrary C++ code.
  - based on Clang libTooling
  - implemented in C++

# PyRosetta-4

- ~~GCCXML~~
- Binder: tool for creating Python bindings for arbitrary C++ code.
  - based on Clang libTooling
  - implemented in C++
  - supports parsing of C++11, C++14, C++17,…

# PyRosetta-4

# PyRosetta-4

- Boost.Python → PyBind11

# PyRosetta-4

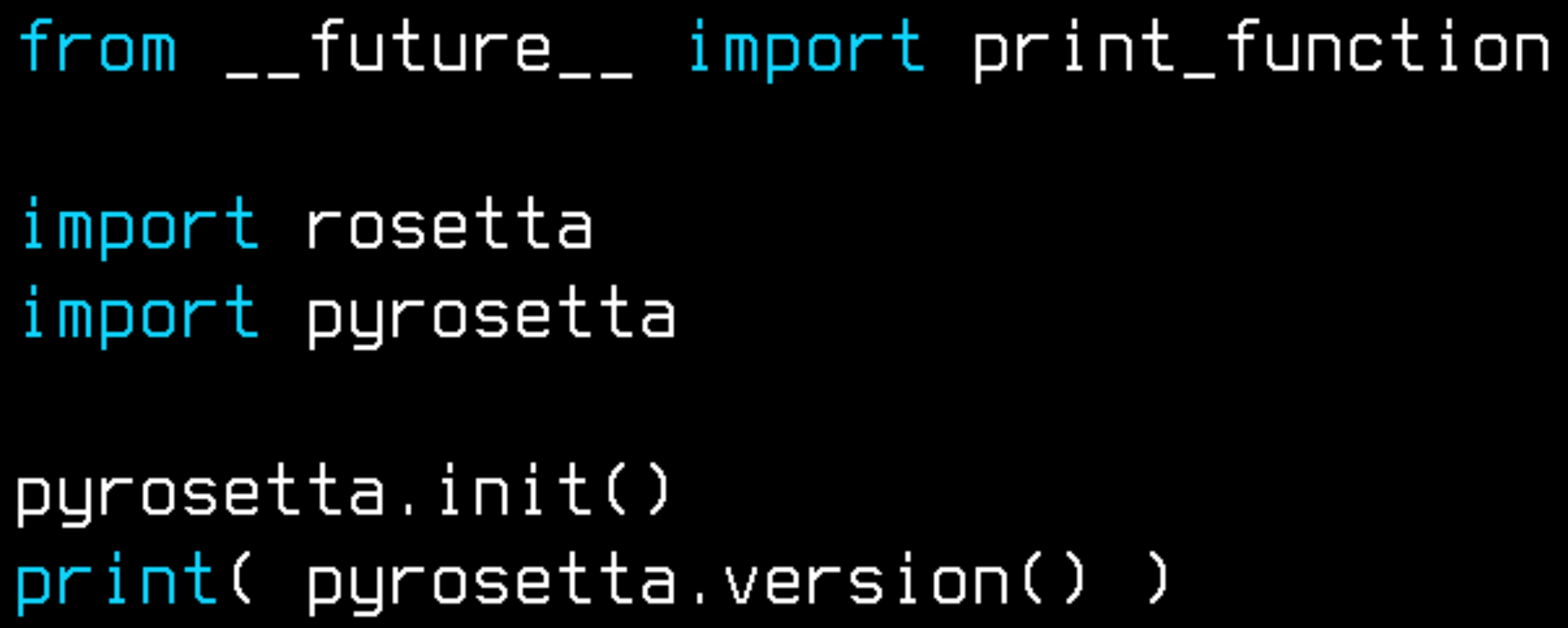
- Boost.Python → PyBind11

# PyRosetta-4

- Boost.Python → PyBind11
- Our-own-custom-build-system → CMake

# Architecture

- PyRosetta-3: ‘rosetta’ included both rosetta and PyRosetta code
- PyRosetta-4: two separate entities:
  - rosetta.so (bindings for Rosetta C++ code) and
  - pyrosetta



```
from __future__ import print_function

import rosetta
import pyrosetta

pyrosetta.init()
print( pyrosetta.version() )
```

# What’s new in PyRosetta-4?

# What’s new in PyRosetta-4?

- Automatic bindings generation for C++ templates

# What’s new in PyRosetta-4?

- Automatic bindings generation for C++ templates
- Classes with virtual function: no run-time overhead for overload classes!

# What’s new in PyRosetta-4?

- Automatic bindings generation for C++ templates
- Classes with virtual function: no run-time overhead for overload classes!
- Function default arguments now bound properly (via C++11 lambda). For example when binding void foo(int a=1, int b=2); we will generate:

```
void py_foo()             { foo(); }
void py_foo(int a)        { foo(a); }
void py_foo(int a, int b) { foo(a, b); }
```

# What’s new in PyRosetta-4?

# What’s new in PyRosetta-4?

- Binding for function accepting pointer to primitive types: int *, double *, bool *, … etc

# What’s new in PyRosetta-4?

- Binding for function accepting pointer to primitive types: int *, double *, bool *, … etc
- Support for bindings of C++11 code and C++14!

# What’s new in PyRosetta-4?

- Binding for function accepting pointer to primitive types: int *, double *, bool *, … etc
- Support for bindings of C++11 code and C++14!
- ‘python setup.py install’ → ~~`SetPyRosettaEnvironment.sh`~~

# What’s new in PyRosetta-4?

- Binding for function accepting pointer to primitive types: int *, double *, bool *, … etc
- Support for bindings of C++11 code and C++14!
- ‘python setup.py install’ → ~~`SetPyRosettaEnvironment.sh`~~
- New source location: `main/source/src/python/PyRosetta`

# Build types

# Build types

- Old PyRosetta-3: namespace/monolith builds

# Build types

- Old PyRosetta-3: namespace/monolith builds
- New PyRosetta-4: **ONLY MONOLITH BUILD**

# Build types

- Old PyRosetta-3: namespace/monolith builds
- New PyRosetta-4: **ONLY MONOLITH BUILD**
- Debug/Release/MinSizeRel/RelWithDebInfo

# What code is bound?

# What code is bound?

- All ‘bindable’ regular code (no exceptions!) And this includes:

# What code is bound?

- All ‘bindable’ regular code (no exceptions!) And this includes:
    - ObjexxFCL

# What code is bound?

- All ‘bindable’ regular code (no exceptions!) And this includes:
  - ObjexxFCL
  - Option System

# What code is bound?

- All ‘bindable’ regular code (no exceptions!) And this includes:
    - ObjexxFCL
    - Option System
- At this moment only skipped code is ether:

# What code is bound?

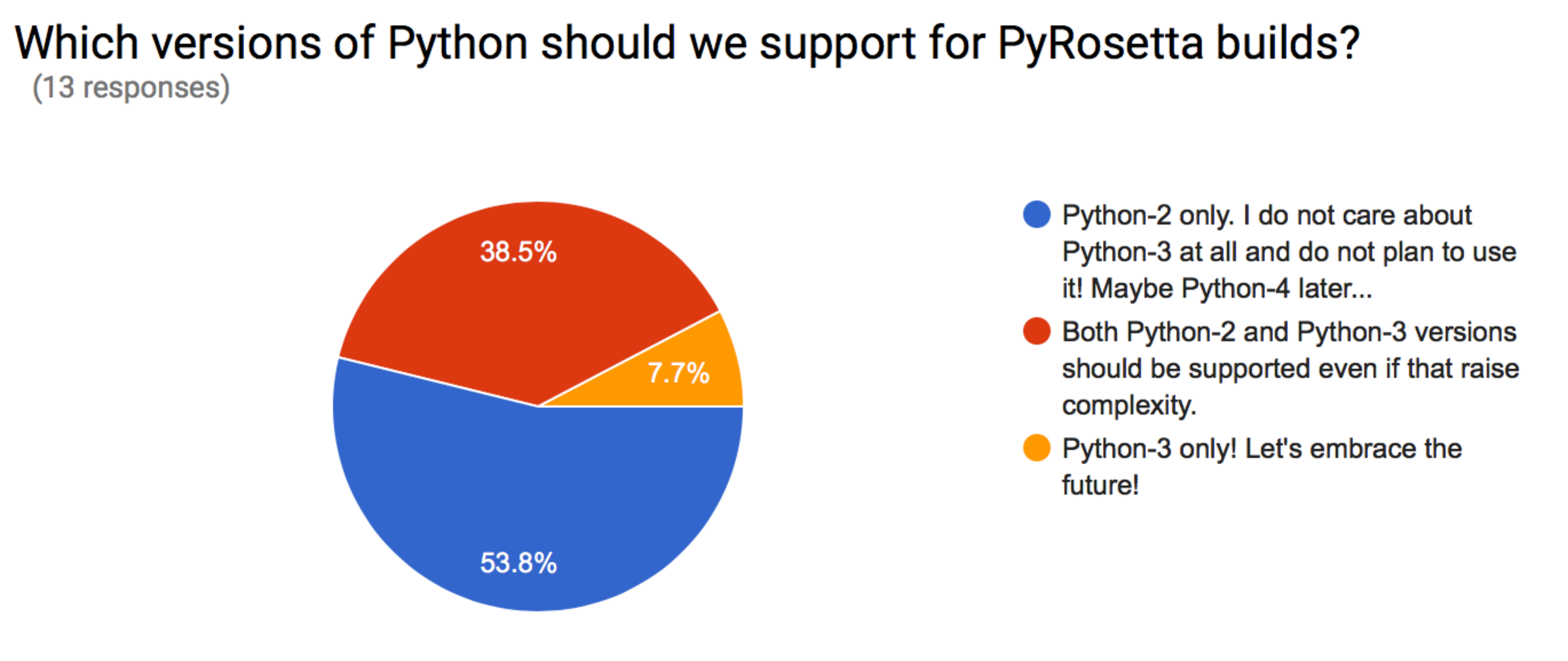
- All ‘bindable’ regular code (no exceptions!) And this includes:
    - ObjexxFCL
    - Option System
- At this moment only skipped code is ether:
    - Template code that could not be instantiated

# What code is bound?

- All ‘bindable’ regular code (no exceptions!) And this includes:
  - ObjexxFCL
  - Option System
- At this moment only skipped code is ether:
  - Template code that could not be instantiated
  - Code that could not be adequately represented in Python

# Python versions compatibility

- Thank you for participating in Python version survey!



Which versions of Python should we support for PyRosetta builds?

(13 responses)

# Python versions compatibility

- Both generation of bindings for Python-2 and Python-3 now supported

- PyRosetta python code, demos, tests are compatible with both Python-2 and Python-3

# What is not yet implemented?

- No Python ‘doc’ strings yet
- print <object>
- PyRosetta ‘apps’ is not yet ported
- PyRosetta GUI tests is not yet ported
- PyRosetta test C001_Carbohydrates_Demo01 fail when ‘installed’
- Python PyMOL Mover implementation is not ported and deprecated
  (please use C++ version instead)

# Limitations

# Limitations

- Right now needs to be built with Clang

# Limitations

- Right now needs to be built with Clang
- std::pair and std::tuple is *read-only*

# Limitations

- Right now needs to be built with Clang
- std::pair and std::tuple is *read-only*
- no support for multiple inheritance (and probably will never be due to PyBind11 limitations)

# Build Speed

# Build Speed

| | Generation phase | Build phase | Total<br>(Rosetta excluded) |
|---|---|---|---|

# Build Speed

| | Generation phase | Build phase | Total<br>(Rosetta excluded) |
|---|---|---|---|
| PyRosetta-3 | 5.9 cpu·h | 27.2 cpu·h | 33.1 cpu·h |

# Build Speed

| | Generation phase | Build phase | Total<br>(Rosetta excluded) |
|---|---|---|---|
| PyRosetta-3 | 5.9 cpu·h | 27.2 cpu·h | 33.1 cpu·h |
| PyRosetta-4 | 0.06 cpu·h<br>3.5 cpu·min | 5.6 cpu·h | 5.66 cpu·h<br>(17%!) |

# Size and Memory consumption

# Size and Memory consumption

| | | |
|---|---|---|
| PyRosetta-3<br>boost.python, release | 1,368 Mb<br>allocating ~4Gb on scoring!!! | 100% |

# Size and Memory consumption

| | | |
|---|---|---|
| PyRosetta-3<br>boost.python, release | 1,368 Mb<br>allocating ~4Gb on scoring!!! | 100% |
| PyRosetta-4<br>PyBind11, release, Linux | 346 Mb | 25% |

# Size and Memory consumption

| | | |
|---|---|---|
| PyRosetta-3<br>boost.python, release | 1,368 Mb<br>allocating ~4Gb on scoring!!! | 100% |
| PyRosetta-4<br>PyBind11, release, Linux | 346 Mb | 25% |
| PyRosetta-4<br>PyBind11, MinSizeRelease, Linux | 244 Mb | 18% |

# Overall PyRosetta-4:

# Overall PyRosetta-4:

- Binds more code (all templates!)

# Overall PyRosetta-4:

- Binds more code (all templates!)
- Binds all default function arguments

# Overall PyRosetta-4:

- Binds more code (all templates!)
- Binds all default function arguments
- Builds x5 faster

# Overall PyRosetta-4:

- Binds more code (all templates!)
- Binds all default function arguments
- Builds x5 faster
- Consumes x5 less memory

# Overall PyRosetta-4:

- Binds more code (all templates!)
- Binds all default function arguments
- Builds x5 faster
- Consumes x5 less memory
- Standard Python package

# Binder architecture

# Binder architecture

- No mention of ‘Rosetta’ or Rosetta specific classes in the code!

# Binder architecture

- No mention of ‘Rosetta’ or Rosetta specific classes in the code!
- Config file to specify bindings options, special binders, … etc: **pluggable architecture**

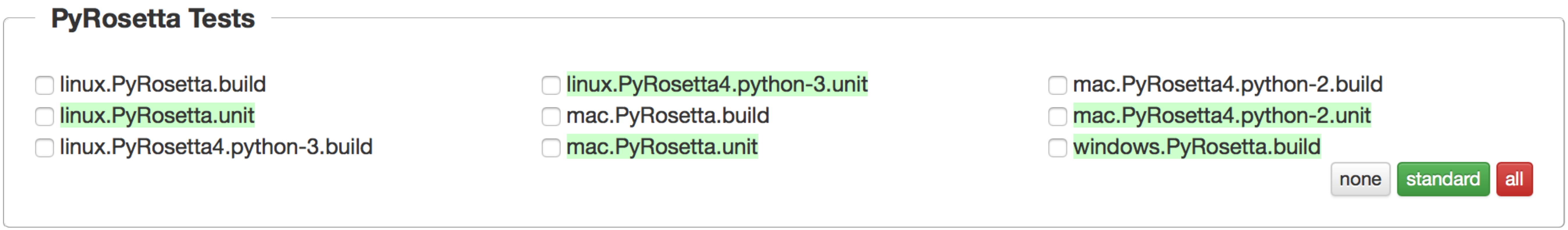
# Binder architecture

- No mention of ‘Rosetta’ or Rosetta specific classes in the code!
- Config file to specify bindings options, special binders, … etc: **pluggable architecture**
- I am going to release Binder as separate package under **FreeBSD license** this fall

# PyRosetta-4, when?

# PyRosetta-4, when?   - right now!

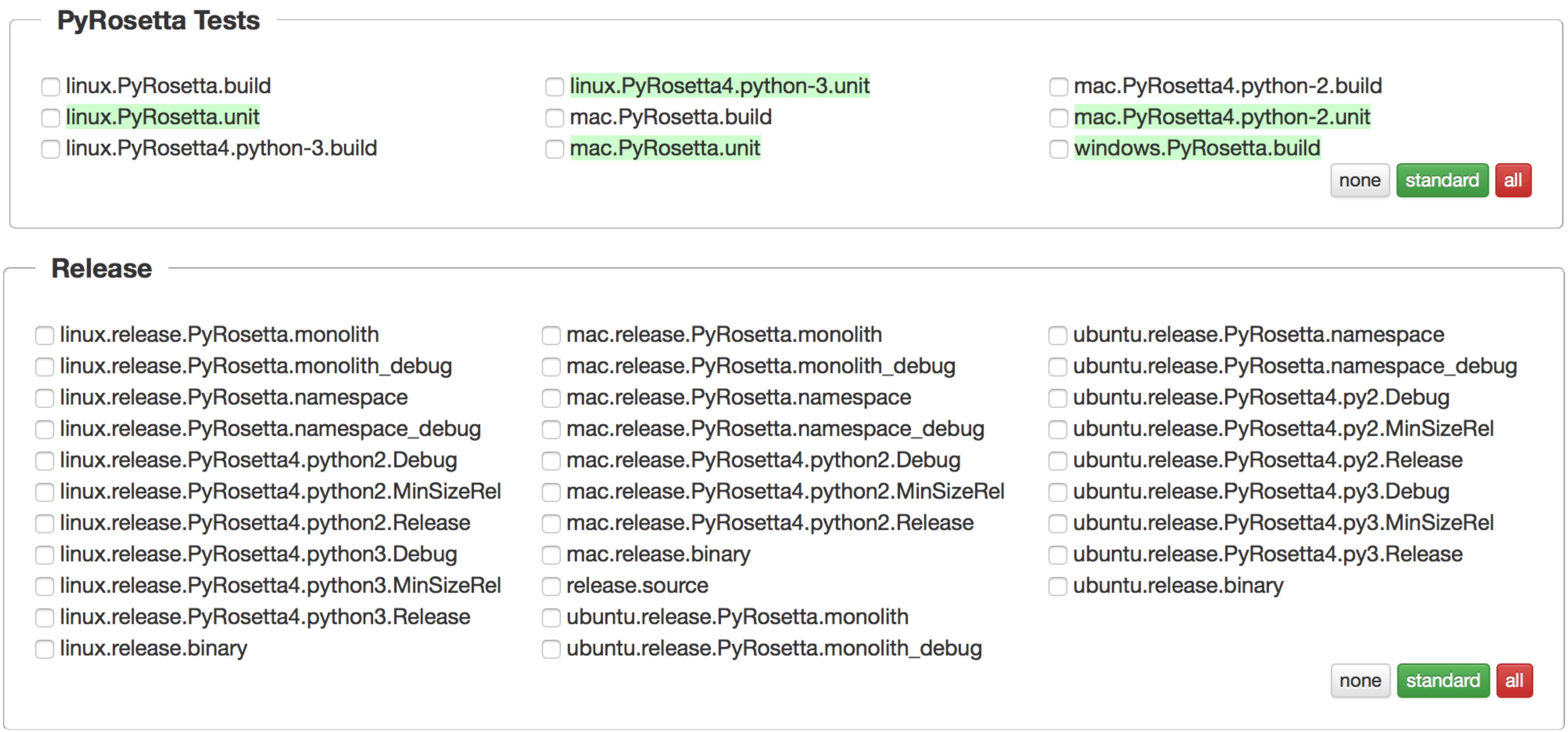
# PyRosetta-4, when? - right now!



### PyRosetta Tests

- □ linux.PyRosetta.build
- □ linux.PyRosetta.unit
- □ linux.PyRosetta4.python-3.build
- □ linux.PyRosetta4.python-3.unit
- □ mac.PyRosetta.build
- □ mac.PyRosetta.unit
- □ mac.PyRosetta4.python-2.build
- □ mac.PyRosetta4.python-2.unit
- □ windows.PyRosetta.build

# PyRosetta-4, when?   - right now!



**PyRosetta Tests**

- □ linux.PyRosetta.build
- □ linux.PyRosetta.unit
- □ linux.PyRosetta4.python-3.build
- □ linux.PyRosetta4.python-3.unit
- □ mac.PyRosetta.build
- □ mac.PyRosetta.unit
- □ mac.PyRosetta4.python-2.build
- □ mac.PyRosetta4.python-2.unit
- □ windows.PyRosetta.build

**Release**

- □ linux.release.PyRosetta.monolith
- □ linux.release.PyRosetta.monolith_debug
- □ linux.release.PyRosetta.namespace
- □ linux.release.PyRosetta.namespace_debug
- □ linux.release.PyRosetta4.python2.Debug
- □ linux.release.PyRosetta4.python2.MinSizeRel
- □ linux.release.PyRosetta4.python2.Release
- □ linux.release.PyRosetta4.python3.Debug
- □ linux.release.PyRosetta4.python3.MinSizeRel
- □ linux.release.PyRosetta4.python3.Release
- □ linux.release.binary
- □ mac.release.PyRosetta.monolith
- □ mac.release.PyRosetta.monolith_debug
- □ mac.release.PyRosetta.namespace
- □ mac.release.PyRosetta.namespace_debug
- □ mac.release.PyRosetta4.python2.Debug
- □ mac.release.PyRosetta4.python2.MinSizeRel
- □ mac.release.PyRosetta4.python2.Release
- □ mac.release.binary
- □ release.source
- □ ubuntu.release.PyRosetta.monolith
- □ ubuntu.release.PyRosetta.monolith_debug
- □ ubuntu.release.PyRosetta.namespace
- □ ubuntu.release.PyRosetta.namespace_debug
- □ ubuntu.release.PyRosetta4.py2.Debug
- □ ubuntu.release.PyRosetta4.py2.MinSizeRel
- □ ubuntu.release.PyRosetta4.py2.Release
- □ ubuntu.release.PyRosetta4.py3.Debug
- □ ubuntu.release.PyRosetta4.py3.MinSizeRel
- □ ubuntu.release.PyRosetta4.py3.Release
- □ ubuntu.release.binary

# How to build?

Install Clang, CMake and Ninja and then:

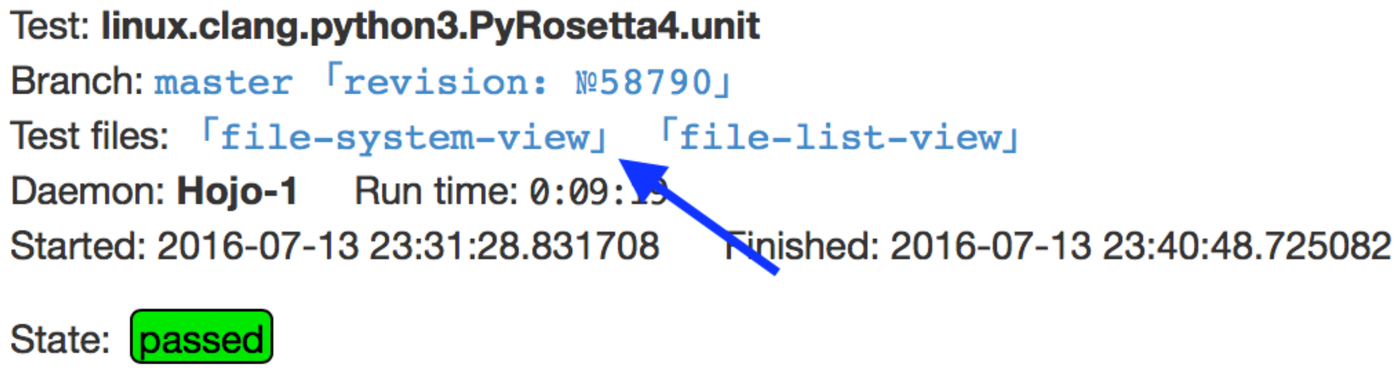
```
% cd main/source/src/python/PyRosetta

%./build.py -j8

% python3 build.py -j8
```

PyRosetta-4 generated source is available at Benchmark test page:

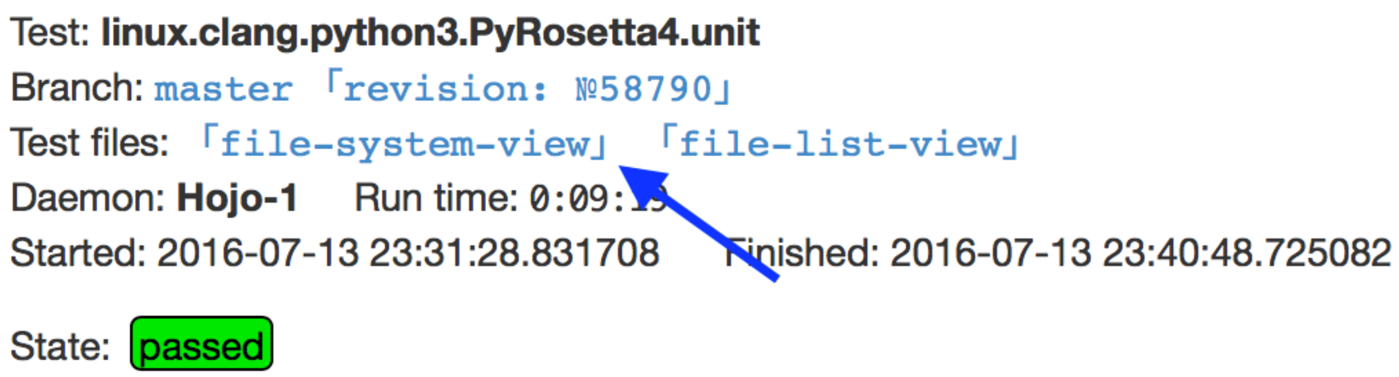
# PyRosetta-4 generated source is available at Benchmark test page:



Test: **linux.clang.python3.PyRosetta4.unit**
Branch: master 「revision: №58790」
Test files: 「file-system-view」 「file-list-view」
Daemon: **Hojo-1** Run time: 0:09:19
Started: 2016-07-13 23:31:28.831708 Finished: 2016-07-13 23:40:48.725082

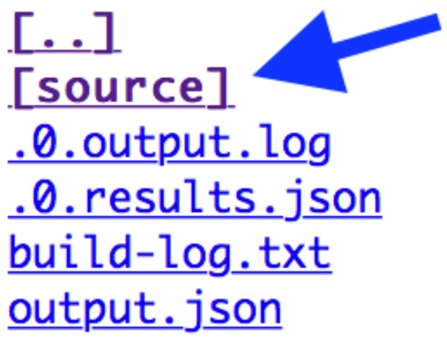
State: passed

# PyRosetta-4 generated source is available at Benchmark test page:



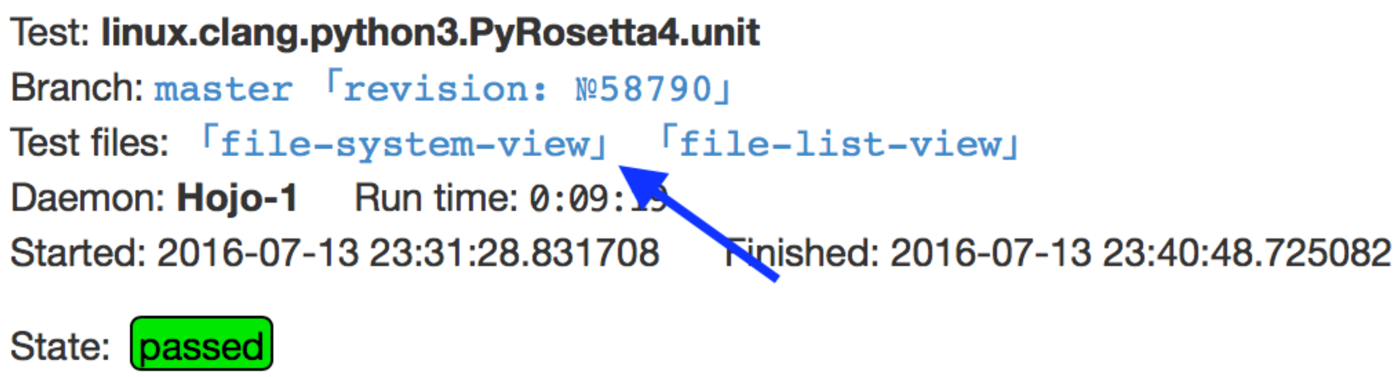
Test: **linux.clang.python3.PyRosetta4.unit**
Branch: master 「revision: №58790」
Test files: 「file-system-view」 「file-list-view」
Daemon: **Hojo-1** Run time: 0:09:19
Started: 2016-07-13 23:31:28.831708 Finished: 2016-07-13 23:40:48.725082

State: passed



```
[..]
[source]
.0.output.log
.0.results.json
build-log.txt
output.json
```

# PyRosetta-4 generated source is available at Benchmark test page:



Test: **linux.clang.python3.PyRosetta4.unit**

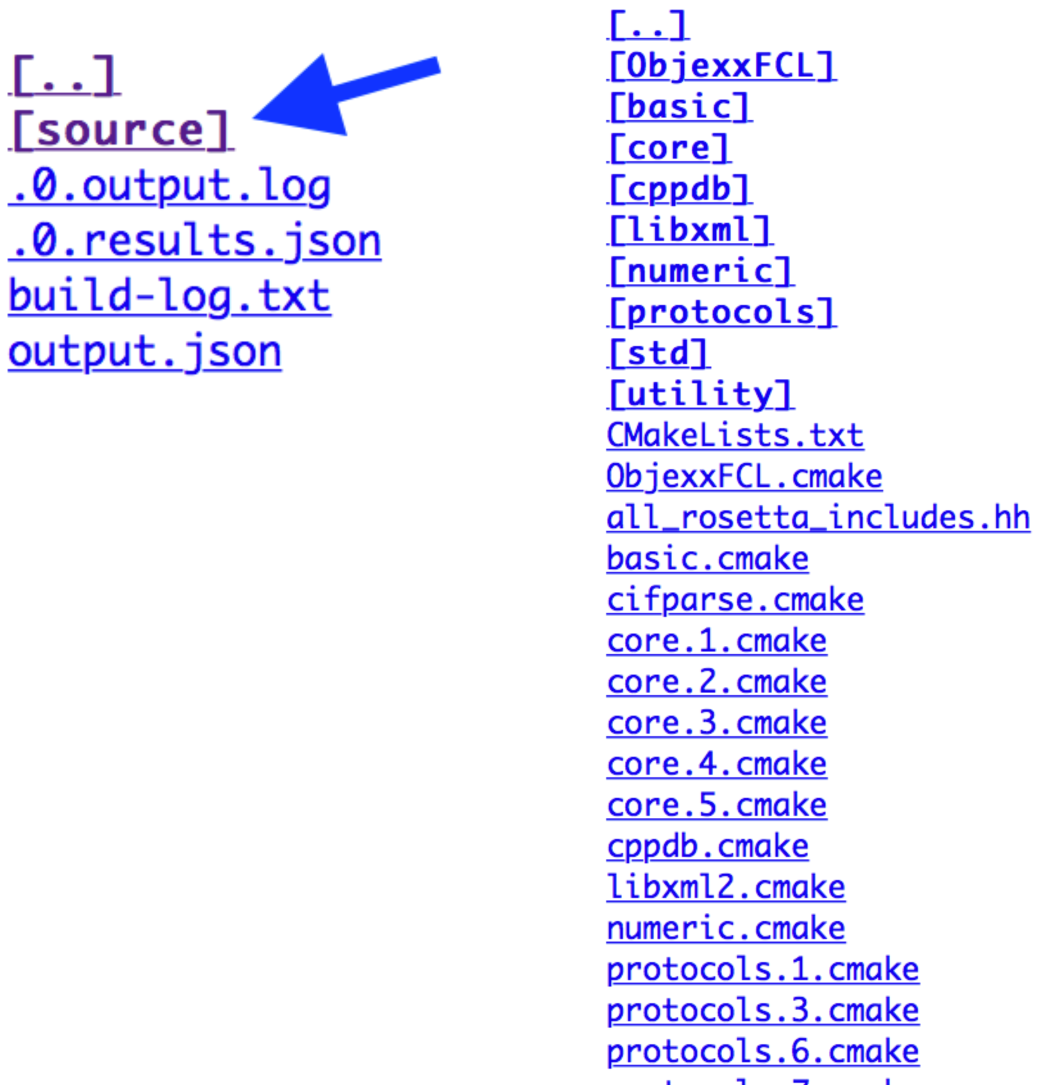
Branch: master 「revision: №58790」

Test files: 「file-system-view」 「file-list-view」

Daemon: **Hojo-1** Run time: 0:09:19

Started: 2016-07-13 23:31:28.831708 Finished: 2016-07-13 23:40:48.725082

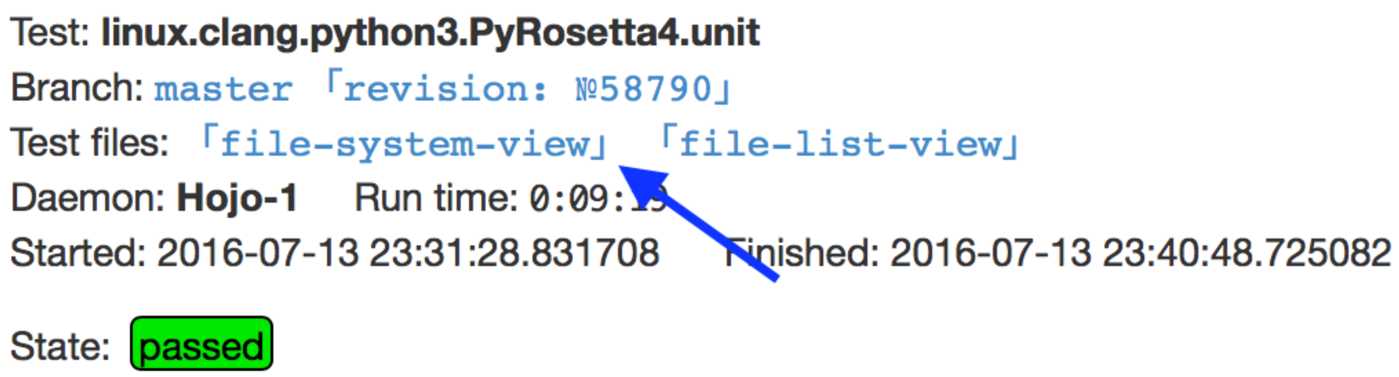
State: passed



[..]
[source]
.0.output.log
.0.results.json
build-log.txt
output.json

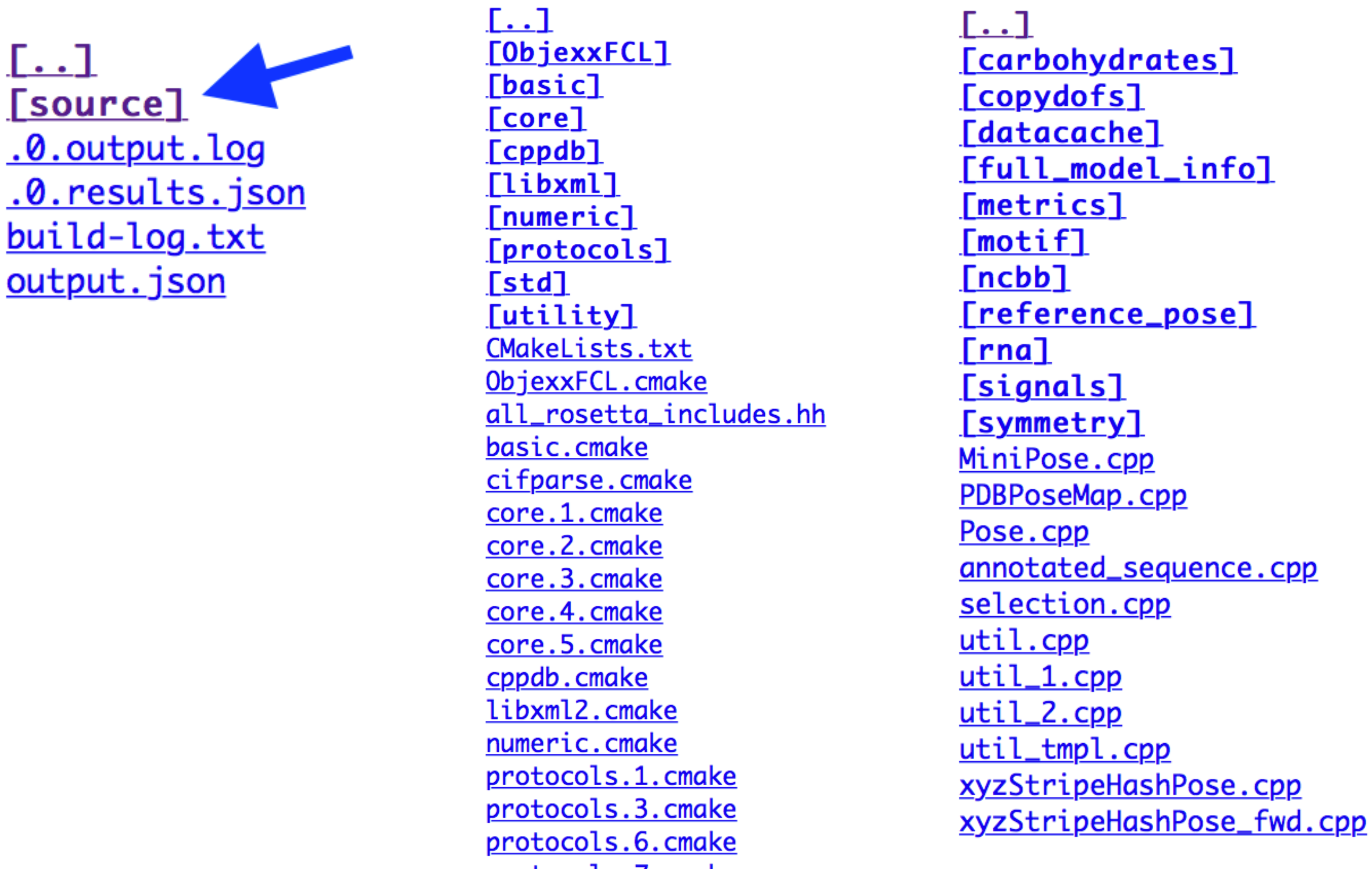
[..]
[ObjexxFCL]
[basic]
[core]
[cppdb]
[libxml]
[numeric]
[protocols]
[std]
[utility]
CMakeLists.txt
ObjexxFCL.cmake
all_rosetta_includes.hh
basic.cmake
cifparse.cmake
core.1.cmake
core.2.cmake
core.3.cmake
core.4.cmake
core.5.cmake
cppdb.cmake
libxml2.cmake
numeric.cmake
protocols.1.cmake
protocols.3.cmake
protocols.6.cmake

# PyRosetta-4 generated source is available at Benchmark test page:



Test: **linux.clang.python3.PyRosetta4.unit**
Branch: master 「revision: №58790」
Test files: 「file-system-view」 「file-list-view」
Daemon: **Hojo-1** Run time: 0:09:19
Started: 2016-07-13 23:31:28.831708 Finished: 2016-07-13 23:40:48.725082

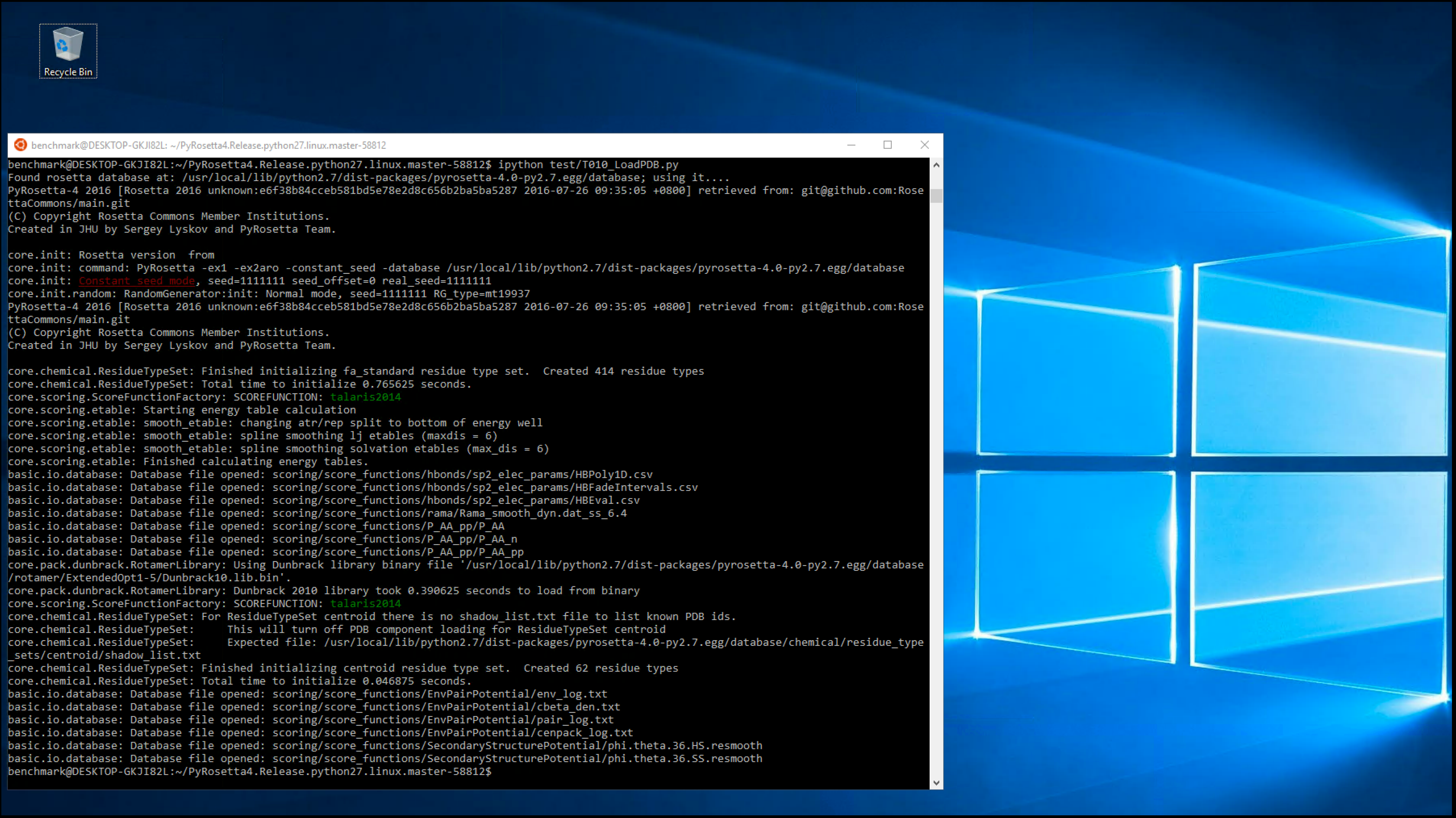
State: passed



[..]
[source]
.0.output.log
.0.results.json
build-log.txt
output.json

[..]
[ObjexxFCL]
[basic]
[core]
[cppdb]
[libxml]
[numeric]
[protocols]
[std]
[utility]
CMakeLists.txt
ObjexxFCL.cmake
all_rosetta_includes.hh
basic.cmake
cifparse.cmake
core.1.cmake
core.2.cmake
core.3.cmake
core.4.cmake
core.5.cmake
cppdb.cmake
libxml2.cmake
numeric.cmake
protocols.1.cmake
protocols.3.cmake
protocols.6.cmake

[..]
[carbohydrates]
[copydofs]
[datacache]
[full_model_info]
[metrics]
[motif]
[ncbb]
[reference_pose]
[rna]
[signals]
[symmetry]
MiniPose.cpp
PDBPoseMap.cpp
Pose.cpp
annotated_sequence.cpp
selection.cpp
util.cpp
util_1.cpp
util_2.cpp
util_tmpl.cpp
xyzStripeHashPose.cpp
xyzStripeHashPose_fwd.cpp



Recycle Bin
benchmark@DESKTOP-GKJI82L: ~/PyRosetta4.Release.python27.linux.master-58812

```
benchmark@DESKTOP-GKJI82L:~/PyRosetta4.Release.python27.linux.master-58812$ ipython test/T010_LoadPDB.py
Found rosetta database at: /usr/local/lib/python2.7/dist-packages/pyrosetta-4.0-py2.7.egg/database; using it....
PyRosetta-4 2016 [Rosetta 2016 unknown:e6f38b84cceb581bd5e78e2d8c656b2ba5ba5287 2016-07-26 09:35:05 +0800] retrieved from: git@github.com:Rose
ttaCommons/main.git
(C) Copyright Rosetta Commons Member Institutions.
Created in JHU by Sergey Lyskov and PyRosetta Team.

core.init: Rosetta version  from
core.init: command: PyRosetta -ex1 -ex2aro -constant_seed -database /usr/local/lib/python2.7/dist-packages/pyrosetta-4.0-py2.7.egg/database
core.init: Constant seed mode, seed=1111111 seed_offset=0 real_seed=1111111
core.init.random: RandomGenerator:init: Normal mode, seed=1111111 RG_type=mt19937
PyRosetta-4 2016 [Rosetta 2016 unknown:e6f38b84cceb581bd5e78e2d8c656b2ba5ba5287 2016-07-26 09:35:05 +0800] retrieved from: git@github.com:Rose
ttaCommons/main.git
(C) Copyright Rosetta Commons Member Institutions.
Created in JHU by Sergey Lyskov and PyRosetta Team.

core.chemical.ResidueTypeSet: Finished initializing fa_standard residue type set.  Created 414 residue types
core.chemical.ResidueTypeSet: Total time to initialize 0.765625 seconds.
core.scoring.ScoreFunctionFactory: SCOREFUNCTION: talaris2014
core.scoring.etable: Starting energy table calculation
core.scoring.etable: smooth_etable: changing atr/rep split to bottom of energy well
core.scoring.etable: smooth_etable: spline smoothing lj etables (maxdis = 6)
core.scoring.etable: smooth_etable: spline smoothing solvation etables (max_dis = 6)
core.scoring.etable: Finished calculating energy tables.
basic.io.database: Database file opened: scoring/score_functions/hbonds/sp2_elec_params/HBPoly1D.csv
basic.io.database: Database file opened: scoring/score_functions/hbonds/sp2_elec_params/HBFadeIntervals.csv
basic.io.database: Database file opened: scoring/score_functions/hbonds/sp2_elec_params/HBEval.csv
basic.io.database: Database file opened: scoring/score_functions/rama/Rama_smooth_dyn.dat_ss_6.4
basic.io.database: Database file opened: scoring/score_functions/P_AA_pp/P_AA
basic.io.database: Database file opened: scoring/score_functions/P_AA_pp/P_AA_n
basic.io.database: Database file opened: scoring/score_functions/P_AA_pp/P_AA_pp
core.pack.dunbrack.RotamerLibrary: Using Dunbrack library binary file '/usr/local/lib/python2.7/dist-packages/pyrosetta-4.0-py2.7.egg/database
/rotamer/ExtendedOpt1-5/Dunbrack10.lib.bin'.
core.pack.dunbrack.RotamerLibrary: Dunbrack 2010 library took 0.390625 seconds to load from binary
core.scoring.ScoreFunctionFactory: SCOREFUNCTION: talaris2014
core.chemical.ResidueTypeSet: For ResidueTypeSet centroid there is no shadow_list.txt file to list known PDB ids.
core.chemical.ResidueTypeSet:     This will turn off PDB component loading for ResidueTypeSet centroid
core.chemical.ResidueTypeSet:     Expected file: /usr/local/lib/python2.7/dist-packages/pyrosetta-4.0-py2.7.egg/database/chemical/residue_type
_sets/centroid/shadow_list.txt
core.chemical.ResidueTypeSet: Finished initializing centroid residue type set.  Created 62 residue types
core.chemical.ResidueTypeSet: Total time to initialize 0.046875 seconds.
basic.io.database: Database file opened: scoring/score_functions/EnvPairPotential/env_log.txt
basic.io.database: Database file opened: scoring/score_functions/EnvPairPotential/cbeta_den.txt
basic.io.database: Database file opened: scoring/score_functions/EnvPairPotential/pair_log.txt
basic.io.database: Database file opened: scoring/score_functions/EnvPairPotential/cenpack_log.txt
basic.io.database: Database file opened: scoring/score_functions/SecondaryStructurePotential/phi.theta.36.HS.resmooth
basic.io.database: Database file opened: scoring/score_functions/SecondaryStructurePotential/phi.theta.36.SS.resmooth
benchmark@DESKTOP-GKJI82L:~/PyRosetta4.Release.python27.linux.master-58812$
```

Thank you!

# Testing Server
# and
# various related code issues

# Testing Server Capacity

# Testing Server Capacity

- Ubuntu dedicated testing servers?

# Testing Server Capacity

- Ubuntu dedicated testing servers?
- Expansion, should buy more Testing Servers right now? Which platform: Linux, Mac, …?

# General Code issues

# General Code issues

- C++11 transition

# General Code issues

- C++11 transition
- my_class::to_string() instead of operator<<

# General Code issues

- C++11 transition
- my_class::to_string() instead of operator<<
- ban multiple inheritance (and require community for new classes thats going to use multiple inheritance)

# General Code issues

- C++11 transition
- my_class::to_string() instead of operator<<
- ban multiple inheritance (and require community for new classes thats going to use multiple inheritance)
- Windows (Py)Rosetta build

# General Code issues

- DO NOT USE ‘using namespace …’ in headers outside of class or function definition!
- How about if add a test to detect this?
- Would it be ok if we test mark file as ‘failed’ if it was modified

What features is missing from our Testing platform?

# Interesting Challenges

- How to made generated code more ‘rebuild-friendly’?
- Map all implementations of std::<thing> to ‘standard’ namespaces/names