

## BornAgain - Feature #1780

### CMake: provide check for Python configuration consistency

22 Mar 2017 14:49 - pospelov

<b>Status:</b>	Resolved	<b>Start date:</b>	22 Mar 2017
<b>Priority:</b>	High	<b>Due date:</b>	
<b>Assignee:</b>	pospelov	<b>% Done:</b>	0%
<b>Category:</b>		<b>Estimated time:</b>	0.00 hour
<b>Target version:</b>	Sprint 35		
<b>Description</b>			
Triggered by <a href="https://github.com/scgmlz/BornAgain/issues/154">https://github.com/scgmlz/BornAgain/issues/154</a> .			
On Ubuntu 16.04.2, while compiling BornAgain with Python3 support, CMake finds interpreter from Python 3, and libraries from Python2. Project builds successfully, but resulting libraries are linked against Python2 and doesn't work with Python3.			
This particular Ubuntu case can be resolved by installing apt-get install python3-dev			
This changes CMake search behavior and BornAgain can be compiled for Python3.			
Within this item, provide additional check inside CMake, that version number of interpreter found by CMake coincide with library version. Make configuration failing, if it is not the case. Before 'swig-period' we had such check inside BornAgainMacros.cmake, see (ugly looking) script below:			
<pre>function(ValidatePythonInstallation)     message(STATUS "--&gt; Validating Python installation corresponding to the interpreter \${PYTHON_EXECUTABLE}")      execute_process(COMMAND "\${PYTHON_EXECUTABLE}" "-c"         "from distutils import sysconfig as s;import sys;import struct; print('.'.join(str(v) for v in sys.version_info)); print(sys.prefix); print(s.get_python_inc(plat_specific=True)); print(s.get_python_lib(plat_specific=True)); print(s.get_config_var('SO')); print(hasattr(sys, 'gettotalrefcount')+0); print(struct.calcsize('@P')); print(s.get_config_var('LDVERSION') or s.get_config_var('VERSION'));"         )      RESULT_VARIABLE _PYTHON_SUCCESS     OUTPUT_VARIABLE _PYTHON_VALUES     ERROR_VARIABLE _PYTHON_ERROR_VALUE     OUTPUT_STRIP_TRAILING_WHITESPACE )  if(NOT _PYTHON_SUCCESS MATCHES 0)     set(ALT_PYTHONLIBS_FOUND FALSE)     return() endif()  # Convert the process output into a list string(REGEX REPLACE ";" "\\;" _PYTHON_VALUES \${_PYTHON_VALUES}) string(REGEX REPLACE "\n" ";" _PYTHON_VALUES \${_PYTHON_VALUES}) list(GET _PYTHON_VALUES 0 _PYTHON_VERSION_LIST) list(GET _PYTHON_VALUES 1 ALT_PYTHON_PREFIX) list(GET _PYTHON_VALUES 2 ALT_PYTHON_INCLUDE_DIRS) list(GET _PYTHON_VALUES 3 ALT_PYTHON_SITE_PACKAGES) list(GET _PYTHON_VALUES 4 ALT_PYTHON_MODULE_EXTENSION) list(GET _PYTHON_VALUES 5 ALT_PYTHON_IS_DEBUG) list(GET _PYTHON_VALUES 6 ALT_PYTHON_SIZEOF_VOID_P) list(GET _PYTHON_VALUES 7 ALT_PYTHON_LIBRARY_SUFFIX)</pre>			

```

string(REGEX REPLACE "\\." ";" _PYTHON_VERSION_LIST ${_PYTHON_VERSION_LIST})
list(GET _PYTHON_VERSION_LIST 0 ALT_PYTHON_VERSION_MAJOR)
list(GET _PYTHON_VERSION_LIST 1 ALT_PYTHON_VERSION_MINOR)
list(GET _PYTHON_VERSION_LIST 2 ALT_PYTHON_VERSION_PATCH)

set(ALT_PYTHON_VERSION_STRING
${ALT_PYTHON_VERSION_MAJOR}.${ALT_PYTHON_VERSION_MINOR}.${ALT_PYTHON_VERSION_PATCH})

message(STATUS "----> ALT_PYTHON_PREFIX:${ALT_PYTHON_PREFIX}")
message(STATUS "----> ALT_PYTHON_INCLUDE_DIRS:${ALT_PYTHON_INCLUDE_DIRS}")
message(STATUS "----> ALT_PYTHON_SITE_PACKAGES:${ALT_PYTHON_SITE_PACKAGES}")
message(STATUS "----> ALT_PYTHON_MODULE_EXTENSION:${ALT_PYTHON_MODULE_EXTENSION}")
message(STATUS "----> ALT_PYTHON_IS_DEBUG:${ALT_PYTHON_IS_DEBUG}")
message(STATUS "----> ALT_PYTHON_SIZEOF_VOID_P:${ALT_PYTHON_SIZEOF_VOID_P}")
message(STATUS "----> ALT_PYTHON_LIBRARY_SUFFIX:${ALT_PYTHON_LIBRARY_SUFFIX}")

if(NOT PYTHON_INCLUDE_DIRS STREQUAL ALT_PYTHON_INCLUDE_DIRS)
  message(STATUS "----> Python interpreter reports include directory (see ALT_PYTHON_INCLUDE_DIRS) which differs from
what we have learned before (see PYTHON_INCLUDE_DIRS).")
  message(STATUS "----> Setting PYTHON_INCLUDE_DIRS=${ALT_PYTHON_INCLUDE_DIRS}")
  set(PYTHON_INCLUDE_DIRS ${ALT_PYTHON_INCLUDE_DIRS} PARENT_SCOPE)
endif()

if(PYTHONLIBS_FOUND)
  #if(NOT PYTHON_VERSION_STRING STREQUAL PYTHONLIBS_VERSION_STRING)
  message(STATUS "----> PYTHON_VERSION_STRING ${PYTHON_VERSION_STRING} differs from
PYTHONLIBS_VERSION_STRING ${PYTHONLIBS_VERSION_STRING}")
  if(APPLE)
    set(ALT_PYTHON_LIBRARIES "${ALT_PYTHON_PREFIX}/lib/libpython${ALT_PYTHON_LIBRARY_SUFFIX}.dylib")
    message(STATUS "----> Will use library from ${ALT_PYTHON_LIBRARIES} instead")
    set(PYTHON_LIBRARIES ${ALT_PYTHON_LIBRARIES} PARENT_SCOPE)
  else()
    message(STATUS "----> There is inconsistency between versions of interpreter and library. Don't know how to handle,
compilation might fail.")
  endif()

  #endif()
endif()

if(NOT PYTHONLIBS_FOUND)
  if(APPLE)
    message(STATUS "----> There was a complain that no suitable Python library has been found. This is APPLE of course...
well, let's see... ")
    set(ALT_PYTHON_LIBRARIES "${ALT_PYTHON_PREFIX}/lib/libpython${ALT_PYTHON_LIBRARY_SUFFIX}.dylib")
    if(${PYTHON_LIBRARIES} STREQUAL ${ALT_PYTHON_LIBRARIES})
      message(STATUS "----> ... we found that the library is OK. Ignoring complain.")
      set(PYTHONLIBS_FOUND TRUE)
    else()
      if(EXISTS ${ALT_PYTHON_LIBRARIES})
        message(STATUS "----> ... there is another one which seems to be OK ${ALT_PYTHON_LIBRARIES}.")
        set(PYTHONLIBS_FOUND TRUE)
        set(PYTHON_LIBRARIES ${ALT_PYTHON_LIBRARIES} PARENT_SCOPE)
      endif()
    endif()
  endif()
endif()

if(PYTHONLIBS_FOUND)
  if(NOT WIN32)
    GET_FILENAME_COMPONENT(PyLibExtension ${PYTHON_LIBRARIES} EXT)
    if(${PyLibExtension} STREQUAL ".a")
      find_package(Threads)
      set(syslibs "-lm -ldl -lutil ${CMAKE_THREAD_LIBS_INIT} -rdynamic")
      message(STATUS "----> Static python library detected, adding ${syslibs}")
      set(PYTHON_LIBRARIES "${syslibs} ${PYTHON_LIBRARIES}" PARENT_SCOPE)
    endif()
  endif()
endif()

```

```

if(BUILD_DEBIAN)
  set(PYTHON_SITE_PACKAGES ${ALT_PYTHON_SITE_PACKAGES} PARENT_SCOPE)
endif()
endif()

if(PYTHONLIBS_FOUND)
  set(PYTHONLIBS_FOUND TRUE PARENT_SCOPE)
  message(STATUS "--> Python seems to be OK.")
else()
  message(FATAL_ERROR "No appropriate Python library has been found. Sorry.")
endif()

# set(ALT_PYTHONLIBS_FOUND ${ALT_PYTHONLIBS_FOUND} PARENT_SCOPE)
# set(ALT_PYTHON_VERSION_STRING ${ALT_PYTHON_VERSION_STRING} PARENT_SCOPE)
# set(ALT_PYTHON_INCLUDE_DIRS ${ALT_PYTHON_INCLUDE_DIRS} PARENT_SCOPE)
endifunction()

```

## History

**#1 - 27 Sep 2017 11:28 - pospelov**

- Status changed from New to Backlog

- Priority changed from Normal to High

The problem with mixed version of Python's interpreter and its libraries still exist. It appears on macmini, when simple

```
cmake -DCMAKE_PREFIX_PATH=/usr/local
```

leads to

```

-- Found PythonInterp: /usr/local/bin/python2.7 (found version "2.7.14")
-- Found Python interpreter version 2.7.14 at /usr/local/bin/python2.7
-- Found PythonLibs: /usr/lib/libpython2.7.dylib (found version "2.7.10")
-- Found Python libraries version 2.7.10 at /usr/lib/libpython2.7.dylib; includes at /usr/include/python2.7
-- Found NUMPY_INCLUDE_DIR='/usr/local/lib/python2.7/site-packages/numpy/core/include'.

```

Please note, that **Homebrew** version of Python (/usr/local) is mixed with MacOS libraries from /usr/lib. It doesn't cause any problem during the normal build, because version of interpreter is pretty close to the version of library. But with new development for embedded Python it become a problem, again.

The reason is a mess with numpy libraries installed on system and on Homebrew.

Current solution is to specify path to Python and libraries explicitly

```

cmake -DPYTHON_LIBRARY=/usr/local/Cellar/python/2.7.14/Frameworks/Python.framework/Versions/2.7/lib/libpython2.7.dylib
-DPYTHON_EXECUTABLE=/usr/local/bin/python2.7 -DCMAKE_PREFIX_PATH=/usr/local ../BornAgain

```

Then, interpreter and libraries are consistent.

```

-- Found PythonInterp: /usr/local/bin/python2.7 (found version "2.7.14")
-- Found Python interpreter version 2.7.14 at /usr/local/bin/python2.7
-- Found PythonLibs: /usr/local/Cellar/python/2.7.14/Frameworks/Python.framework/Versions/2.7/lib/libpython2.7.dylib (found version "2.7.14")
-- Found Python libraries version 2.7.14 at /usr/local/Cellar/python/2.7.14/Frameworks/Python.framework/Versions/2.7/lib/libpython2.7.dylib; includes
at /usr/local/Cellar/python/2.7.14/Frameworks/Python.framework/Versions/2.7/include/python2.7
-- Found NUMPY_INCLUDE_DIR='/usr/local/lib/python2.7/site-packages/numpy/core/include'.
-- Numpy headers found at /usr/local/lib/python2.7/site-packages/numpy/core/include, version 1.13.1.

```

## Required solution

- Get rid from cmake's FindPythonLibrary call and rely on FindPythonInterpreter to deduce libraries location, as it was 3 years ago.

**#2 - 06 Oct 2017 14:37 - pospelov**

- Status changed from Backlog to Sprint
- Assignee set to pospelov
- Target version set to Sprint 35

**#3 - 06 Oct 2017 15:03 - pospelov**

- Status changed from Sprint to Resolved