

## GammaLib - Bug #4196

### Initial GammaLib compile failed on Mac OS 10.11

01/12/2023 10:33 AM - Knödlseeder Jürgen

<b>Status:</b>	Closed	<b>Start date:</b>	01/12/2023
<b>Priority:</b>	Normal	<b>Due date:</b>	
<b>Assigned To:</b>	Knödlseeder Jürgen	<b>% Done:</b>	100%
<b>Category:</b>		<b>Estimated time:</b>	0.00 hour
<b>Target version:</b>	2.1.0		

#### Description

When building GammaLib the first time on Mac OS 10.11 it failed with the following issue:

```
+ make installcheck
Making installcheck in include
...
Making installcheck in test
Traceback (most recent call last):
  File "<string>", line 1, in <module>
ImportError: No module named gammalib
*** gammalib unit test failure!
make[1]: *** [installcheck-local] Error 1
make: *** [installcheck-recursive] Error 1
Build step 'Exécuter un script shell' marked build as failure
```

#### History

##### #1 - 01/12/2023 11:24 AM - Knödlseeder Jürgen

- Status changed from New to In Progress

- % Done changed from 0 to 10

Checking what is available on the disk:

```
$ ssh -Y -p 2211 jenkins@ctamac.irap.omp.eu
Password:
$ ls -l jenkins/install/os/gammalib/lib/
-rwxr-xr-x 1 jenkins staff 7268540 12 jan 02:36 libgamma.9.dylib
-rw-r--r-- 1 jenkins staff 113925688 12 jan 02:36 libgamma.a
lrwxr-xr-x 1 jenkins staff 16 12 jan 02:36 libgamma.dylib -> libgamma.9.dylib
-rwxr-xr-x 1 jenkins staff 977 12 jan 02:36 libgamma.la
drwxr-xr-x 3 jenkins staff 102 12 jan 02:36 pkgconfig
drwxr-xr-x 3 jenkins staff 102 12 jan 02:36 python2.7
$ ls -l jenkins/workspace/gammalib-nightly-os/label/
total 0
drwxr-xr-x 48 jenkins staff 1632 12 jan 02:36 macosx11
```

Repeating in the command line

```
$ make installcheck
Making installcheck in include
...
Making installcheck in test
Traceback (most recent call last):
  File "<string>", line 1, in <module>
ImportError: No module named gammalib
*** gammalib unit test failure!
make[1]: *** [installcheck-local] Error 1
```

```
make: *** [installcheck-recursive] Error 1
```

reproduced the issue.

Reproducing by hand what is done by the Makefile gives:

```
CTAs-Mac:test jenkins$ export GAMMALIB=/Users/jenkins/jenkins/install/os/gammalib
CTAs-Mac:test jenkins$ ./Users/jenkins/jenkins/install/os/gammalib/bin/gammalib-init.sh
CTAs-Mac:test jenkins$ python
Python 2.7.10 (default, Oct 23 2015, 19:19:21)
[GCC 4.2.1 Compatible Apple LLVM 7.0.0 (clang-700.0.59.5)] on darwin
Type "help", "copyright", "credits" or "license" for more information.
>>> import gammalib
Traceback (most recent call last):
  File "<stdin>", line 1, in <module>
ImportError: No module named gammalib
```

The issue is maybe related to a bad PYTHONPATH in the gammalib-setup script:

```
$ echo $PYTHONPATH
/lib/python2.7/site-packages:/lib/python2.7/site-packages
$ more /Users/jenkins/jenkins/install/os/gammalib/bin/gammalib-setup
prefix="/Users/jenkins/jenkins/install/os/gammalib"
exec_prefix="${prefix}"
gammalib_bin="${exec_prefix}/bin"
gammalib_lib="${exec_prefix}/lib"
uname_system=`uname`
python_dir="${PYTHON_PREFIX}/lib/python2.7/site-packages"
python_exec_dir="${PYTHON_EXEC_PREFIX}/lib/python2.7/site-packages"
```

On my Mac I see the following

```
prefix="/usr/local/gamma"
exec_prefix="${prefix}"
gammalib_bin="${exec_prefix}/bin"
gammalib_lib="${exec_prefix}/lib"
uname_system=`uname`
python_dir="${prefix}/lib/python2.7/site-packages"
python_exec_dir="${exec_prefix}/lib/python2.7/site-packages"
```

Note that in the gammalib-setup script the following instructions are found:

```
python_dir="@pythondir@"
python_exec_dir="@pyexecdir@"
```

In the Makefile I see

```
...
PYTHON_EXEC_PREFIX = ${exec_prefix}
...
PYTHON_PREFIX = ${prefix}
...
pyexecdir = ${PYTHON_EXEC_PREFIX}/lib/python2.7/site-packages
pythondir = ${PYTHON_PREFIX}/lib/python2.7/site-packages
```

and for some reason this does not seem to be resolved.

Note that the issue is seen already in the ./configure step:

```
checking for python... /usr/bin/python
checking for python version... 2.7
checking for python platform... darwin
```

```
checking for GNU default python prefix... ${prefix}
checking for GNU default python exec_prefix... ${exec_prefix}
checking for python script directory (pythondir)... ${PYTHON_PREFIX}/lib/python2.7/site-packages
checking for python extension module directory (pyexecdir)... ${PYTHON_EXEC_PREFIX}/lib/python2.7/site-packages
checking for the distutils Python package... yes
checking for /System/Library/Frameworks/Python.framework/Versions/2.7/include/python2.7/Python.h... yes
```

while on my Mac I see

```
checking for python... /usr/bin/python
checking for python version... 2.7
checking for python platform... darwin
checking for python script directory... ${prefix}/lib/python2.7/site-packages
checking for python extension module directory... ${exec_prefix}/lib/python2.7/site-packages
checking for the distutils Python package... yes
checking /System/Library/Frameworks/Python.framework/Versions/2.7/include/python2.7/Python.h usability... yes
checking /System/Library/Frameworks/Python.framework/Versions/2.7/include/python2.7/Python.h presence... yes
checking for /System/Library/Frameworks/Python.framework/Versions/2.7/include/python2.7/Python.h... yes
```

\*Note the additional checks for "GNU default python prefix" and "GNU default python exec\_prefix" that were not there before."

In the previous Mac OS 10.11 configuration the following was shown in the configure step:

```
checking for python... /usr/bin/python
checking for python version... 2.7
checking for python platform... darwin
checking for python script directory... ${prefix}/lib/python2.7/site-packages
checking for python extension module directory... ${exec_prefix}/lib/python2.7/site-packages
checking for the distutils Python package... yes
checking /System/Library/Frameworks/Python.framework/Versions/2.7/include/python2.7/Python.h usability... yes
checking /System/Library/Frameworks/Python.framework/Versions/2.7/include/python2.7/Python.h presence... yes
checking for /System/Library/Frameworks/Python.framework/Versions/2.7/include/python2.7/Python.h... yes
```

**Note that the "GNU default" lines were not there.**

Let's see if this persists the next time. There is maybe a difference in the AM\_PATH\_PYTHON macro that I installed through the Homebrew (see also [https://www.gnu.org/software/automake/manual/html\\_node/Python.html](https://www.gnu.org/software/automake/manual/html_node/Python.html)).

## #2 - 01/12/2023 11:41 AM - Knödseder Jürgen

- % Done changed from 10 to 20

The python.m4 script differs:

```
$ more /usr/local/share/aclocal-1.16/python.m4
## ----- *- Autoconf *-
## Python file handling
## From Andrew Dalke
## Updated by James Henstridge and other contributors.
## -----
## Copyright (C) 1999-2021 Free Software Foundation, Inc.
```

while on my Mac it is

```
$ more /usr/local/share/aclocal-1.16/python.m4
## ----- *- Autoconf *-
## Python file handling
## From Andrew Dalke
## Updated by James Henstridge
## -----
## Copyright (C) 1999-2020 Free Software Foundation, Inc.
```

There was apparently a change in the way how things are done. On my Mac I see

```
dnl Use the values of $prefix and $exec_prefix for the corresponding
dnl values of PYTHON_PREFIX and PYTHON_EXEC_PREFIX. These are made
dnl distinct variables so they can be overridden if need be. However,
dnl general consensus is that you shouldn't need this ability.
```

```
AC_SUBST([PYTHON_PREFIX], [${prefix}])
AC_SUBST([PYTHON_EXEC_PREFIX], [${exec_prefix}])
```

while on Mac OS 10.11 there is a much more complicated code

```
dnl If --with-python-sys-prefix is given, use the values of sys.prefix
dnl and sys.exec_prefix for the corresponding values of PYTHON_PREFIX
dnl and PYTHON_EXEC_PREFIX. Otherwise, use the GNU $prefix and
dnl ${exec_prefix} variables.
```

```
dnl
dnl The two are made distinct variables so they can be overridden if
dnl need be, although general consensus is that you shouldn't need
dnl this separation.
```

```
dnl
dnl Also allow directly setting the prefixes via configure options,
dnl overriding any default.
```

```
dnl
if test "x$prefix" = xNONE; then
  am_usable_prefix=$ac_default_prefix
else
  am_usable_prefix=$prefix
fi
```

```
# Allow user to request using sys.* values from Python,
# instead of the GNU $prefix values.
AC_ARG_WITH([python-sys-prefix],
[AS_HELP_STRING([--with-python-sys-prefix],
[use Python's sys.prefix and sys.exec_prefix values])],
[am_use_python_sys=:],
[am_use_python_sys=false])
```

```
# Allow user to override whatever the default Python prefix is.
AC_ARG_WITH([python_prefix],
[AS_HELP_STRING([--with-python_prefix],
[override the default PYTHON_PREFIX])],
```

```

[am_python_prefix_subst=$withval
am_cv_python_prefix=$withval
AC_MSG_CHECKING([for explicit $am_display_PYTHON prefix])
AC_MSG_RESULT([$am_cv_python_prefix]),
[
if $am_use_python_sys; then
# using python sys.prefix value, not GNU
AC_CACHE_CHECK([for python default $am_display_PYTHON prefix],
[am_cv_python_prefix],
[am_cv_python_prefix=`$PYTHON -c "import sys; sys.stdout.write(sys.prefix)"]])

dnl If sys.prefix is a subdir of $prefix, replace the literal value of
dnl $prefix with a variable reference so it can be overridden.
case $am_cv_python_prefix in
$am_usable_prefix*)
am_strip_prefix=`echo "$am_usable_prefix" | sed 's|.|.|g'`
am_python_prefix_subst=`echo "$am_cv_python_prefix" | sed "s,^$am_strip_prefix,\\${prefix},"`
;;
*)
am_python_prefix_subst=$am_cv_python_prefix
;;
esac
else # using GNU prefix value, not python sys.prefix
am_python_prefix_subst=${prefix}
am_python_prefix=$am_python_prefix_subst
AC_MSG_CHECKING([for GNU default $am_display_PYTHON prefix])
AC_MSG_RESULT([$am_python_prefix])
fi])
# Substituting python_prefix_subst value.
AC_SUBST([PYTHON_PREFIX], [$am_python_prefix_subst])

# emacs-page Now do it all over again for Python exec_prefix, but with yet
# another conditional: fall back to regular prefix if that was specified.
AC_ARG_WITH([python_exec_prefix],
[AS_HELP_STRING([--with-python_exec_prefix],
[override the default PYTHON_EXEC_PREFIX])],
[am_python_exec_prefix_subst=$withval
am_cv_python_exec_prefix=$withval
AC_MSG_CHECKING([for explicit $am_display_PYTHON exec_prefix])
AC_MSG_RESULT([$am_cv_python_exec_prefix]),
[
# no explicit --with-python_exec_prefix, but if
# --with-python_prefix was given, use its value for python_exec_prefix too.
AS_IF([test -n "$with_python_prefix"],
[am_python_exec_prefix_subst=$with_python_prefix
am_cv_python_exec_prefix=$with_python_prefix
AC_MSG_CHECKING([for python_prefix-given $am_display_PYTHON exec_prefix])
AC_MSG_RESULT([$am_cv_python_exec_prefix]),
[
# Set am__usable_exec_prefix whether using GNU or Python values,
# since we use that variable for pyexecdir.
if test "x$exec_prefix" = xNONE; then
am__usable_exec_prefix=$am_usable_prefix
else
am__usable_exec_prefix=$exec_prefix
fi
#
if $am_use_python_sys; then # using python sys.exec_prefix, not GNU
AC_CACHE_CHECK([for python default $am_display_PYTHON exec_prefix],
[am_cv_python_exec_prefix],
[am_cv_python_exec_prefix=`$PYTHON -c "import sys; sys.stdout.write(sys.exec_prefix)"]])
dnl If sys.exec_prefix is a subdir of $exec_prefix, replace the
dnl literal value of $exec_prefix with a variable reference so it can
dnl be overridden.
case $am_cv_python_exec_prefix in
$am_usable_exec_prefix*)
am_strip_prefix=`echo "$am_usable_exec_prefix" | sed 's|.|.|g'`
am_python_exec_prefix_subst=`echo "$am_cv_python_exec_prefix" | sed "s,^$am_strip_prefix,\\${exec_prefix},"`
;;
*)
am_python_exec_prefix_subst=$am_cv_python_exec_prefix
;;
esac
else # using GNU $exec_prefix, not python sys.exec_prefix
am_python_exec_prefix_subst=${exec_prefix}

```

```
am_python_exec_prefix=$am_python_exec_prefix_subst
AC_MSG_CHECKING([for GNU default $am_display_PYTHON exec_prefix])
AC_MSG_RESULT([$am_python_exec_prefix
fi]))
# Substituting python_exec_prefix_subst.
AC_SUBST([PYTHON_EXEC_PREFIX], [$am_python_exec_prefix_subst])
```

### #3 - 01/12/2023 11:57 AM - Knödseder Jürgen

Maybe the issues comes from this change in /usr/local/share/aclocal-1.16/python.m4. On Mac OS 10.11 I see

```
case $am_py_prefix in
  /usr/System*) ;;
  *) am_cv_python_pythondir="\${PYTHON_PREFIX}/lib/python$PYTHON_VERSION/site-packages"
    ;;
esac
...
case $am_py_exec_prefix in
  /usr/System*) ;;
  *) am_cv_python_pyexecdir="\${PYTHON_EXEC_PREFIX}/lib/python$PYTHON_VERSION/site-packages"
    ;;
esac
```

while on my Mac I see

```
case $am_py_prefix in
  /usr/System*) ;;
  *)
    am_cv_python_pythondir=$PYTHON_PREFIX/lib/python$PYTHON_VERSION/site-packages
    ;;
esac
...
case $am_py_exec_prefix in
  /usr/System*) ;;
  *)
    am_cv_python_pyexecdir=$PYTHON_EXEC_PREFIX/lib/python$PYTHON_VERSION/site-packages
    ;;
esac
```

I guess that in the first version (on Mac OS 10.11) there is no expansion of the \$PYTHON\_PREFIX variable.

**#4 - 01/12/2023 12:17 PM - Knödseder Jürgen**

- % Done changed from 20 to 50

I added

```
PYTHON_PREFIX="${prefix}"  
PYTHON_EXEC_PREFIX="${exec_prefix}"
```

to src/gammlib-setup.in so that the initial setting is enforced, even for the new python.m4 script.

**#5 - 01/12/2023 10:53 PM - Knödseder Jürgen**

I added

```
PYTHON_PREFIX="${prefix}"  
PYTHON_EXEC_PREFIX="${exec_prefix}"
```

to src/ctools-setup.in

**#6 - 01/12/2023 10:54 PM - Knödseder Jürgen**

- Project changed from Infrastructure to GammaLib

- Target version set to 2.1.0

- % Done changed from 50 to 100

This fixed the problem.

**#7 - 01/16/2023 09:38 AM - Knödseder Jürgen**

- Status changed from In Progress to Closed