Python 3 in DiFX

DiFX 2.7 is (coincidentally) is the version targeting good support for Python 3. Note that much of Python3 code can run under Python2.7. This page is meant to capture status information for utilities that have and have not been converted, and some tips relevant to DiFX code for the conversion.

Code that has been converted

Here the converted code is sorted alphabetically. Package name may be in parantheses.

- cpumon, difxdiagnosticmon, difxwatch, errormon, errormon2, getsmart, mk5mon, mk6mon, restartdifx, smartmon, statemon (difxmessage)
- (espresso)
- difxspeed, mk62v2d, oms2v2d, vlog (vex2difx)
- killjob, listcpus, listmedia, listmodules (mk5daemon)
- plotpcal2, showcal, makefits, gentcal (difx2fits)
- difxsniff (snifftools)
- joblist, jobdisks, jobstatus, startdifx, stopmpifxcorr (mpifxcorr)
- difxcopy, mvdifxhere (difxio)
- makefits, gentcal, showcal, plotpcal2 (difx2fits)

Code that needs converting

Here the code is sorted by priority rather than by package. The package name may be included in parentheses after the program name. Priority here is subjective; if you think something deserves more priority than something else, feel free to elevate it or make a request to difx-developers@nrao.edu to do so.

- genmachines, calcifMixed (mpifxcorr)

Coordination

The “python” package (https://svn.atnf.csiro.au/difx/libraries/python/) is used by some of the utilities. Coordination of the update to these utilities is required to prevent partial breakage. The affected utilities include:

- mk5control (mk5daemon)
- genmachines (mpifxcorr)

It seems like the dependencies are not too numerous…

High priority

Medium priority

Low priority

Some notes about converting from Python2 to Python3

There are many web pages describing what is needed to convert between Python2 code and Python3 code. Despite many of these pages, I (Walter) have not found a simple listing of the gotchas. So below I put some notes on challenges that I faced.

- print is now a function: print()
- integer division expecting (rounded) integer result requires use of double-/ rather than single-/
- network recv and recvfrom instructions return byte arrays that need to be converted to strings. This can be done with .decode('ascii') or .decode('utf-8') applied to the resultant string
- dictionaries no longer support the .has_key() method. Instead use “in”. E.g., “if key in dict:”
- dictionaries no longer support the .iteritems() method. Use now “.items()”.
- the .sort() method no longer works on dictionary.keys(). Instead, use “sorted()”. E.g., “sorted(dictionary.keys())”.
- “from string import …” does not work for functions strip(), split(), upper(), lower(). These functions can instead be used as methods of the strings you are working with
- Long integer numeric constants cannot be expressed as “0L” anymore
- Python3 is more picky about indentation. No equivalence between tabs and spaces.
- Octal values can no longer be input simply with leading “0”, such as 022, but now require “0o”, as in “0o22”
- “str” is now a reserved word, should not be used as a variable name

From:
https://www.atnf.csiro.au/vlbi/dokuwiki/ - **ATNF VLBI Wiki**

Permanent link:
https://www.atnf.csiro.au/vlbi/dokuwiki/doku.php/difx/python3

Last update: **2019/11/15 15:59**