Support for filtering based on command line arguments?

Robert McLay

February 23, 2023

XALT: Outline

- Let's talk about how XALT might filter based on command line arguments
- A site would like to filter on where the python script is located.
- I can think of two ways that this might be done
- I am open to discussion on how this might be done
- I am assuming that command name must pass regular path filtering.
- Let's see about how a second filter might work.

Some issue to deal with

- Many commands take command line argument
- This includes python
- Below is a list of *some* of them (via zsh tab completion)
- Python: no options with values
- But may programs take options with values

```
% python3 -<tab>
option
-B   -- don't write .py[co] files on import
-E   -- ignore PYTHON* environment variables (such as PYTHONPATH)
-I   -- isolate Python from the user's environment
-O   -- optimize generated bytecode slightly
-OO  -- remove doc-strings in addition to the -O optimizations
-x   -- skip first line of source, allowing use of non-Unix forms of #!cmd
```

Command line parsing is a PITA

- This is difficult to get right the first time
- It much harder to maintain over time

Path foo is required

```
#!/bin/python3

from __future__ import print_function
import os, sys, re

def main():
  for i in sys.argv:
    fn = os.path.abspath(i)
    if (os.path.exists(fn)):
      print (i, fn)
    else:
      print (i)

if ( __name__ == '__main__'): main()
```

- running ./try.py or ../../abc/try.py doesn't expand to /full/path/try.py
- running python3 ../../abc/try.py also doesn't expand to /full/path/try.py
- Checking argments for abspath and existance will be required

Approach 1

```
int my_path_parser(const char * cmdName, const char* cmdlineA[])
{
  // return 1 for PKGS  Use for python if imports tracked
  // return 2 for KEEP
  // return 3 for SKIP
}
```

- Sites provide a function line the above
- XALT will provide a configure option to link in a library or a *.o file

Approach 2

```
path_cmd_patterns = [
['SKIP', r'python[0-9.]*;\/share\/.* ]
['PKGS', r'python[0-9.]*;\/other\/.* ]
   ...
   ]
```

- The command and the argument would be combined together with a semicolon
- Command line arguments would be ignored if started with a minus [-]
- All arguments would be abspath and checked for existance before being run through path_cmd_patterns
- Not perfect but reasonably safe
- Comments?

Future Topics?

- I'm looking for Topics.
- Next Meeting will be on March 16, 2023 at 10:00 am U.S. Central (15:00 UTC)