Subject: Re: Command line arguments
Posted by Michael Wallace on Fri, 11 Feb 2005 21:17:59 GMT
View Forum Message <> Reply to Message

> I would like to compile the program once outside the loop and then pass
> the file using an argument instead grabbing the file from an environment
> variable. Can I do something like this?
>
> print IDL "mentor, $filepath\n";

Yes.

Maybe I don't understand what you're doing, but why are you 1.) explicitly compiling the command your going to run instead of letting IDL automatically compile it? and 2.) why don't you write a wrapper in IDL itself that handles the looping and you just call the wrapper the one time instead of calling the same IDL command multiple times?

My Perl is rusty, but a while back I wrote a Python wrapper to simulate command line arguments. I think I posted it in the newsgroup some time ago, but I couldn't find the old thread.

The first argument to the program below is the name of the IDL command to execute. The remainder of the arguments will be fed to the IDL command named in the first argument. I have defined my IDL programs that need to be run on the command line to include the keyword ARGS. The additional arguments are passed into the IDL command using this keyword. By doing this, I can use this one wrapper for all command line programs, but the argument list is not bound to something specific.

If I want to quickly do something and the program I want to run doesn't have an ARGS keyword, I can just put the entire IDL command I want to run in the first argument. Just quote the entire command so that it's interpreted as a single argument.

-Mike

```
#!/usr/bin/env python

import os
import sys

# Usage statement
usage = "usage: %s idlprog args" %os.path.basename(sys.argv[0])

# Check that the name of the IDL program was provided
```

```
if len(sys.argv) < 2:
    print usage
else:
    fd = os.popen('idl', 'w')

    # If extra arguments are given, pass them via the ARGS keyword
    if len(sys.argv) < 3:
        fd.write(sys.argv[1])
    else:
        fd.write(sys.argv[1] + ', ARGS = ' + `sys.argv[2:]`)

    fd.close()
```

---