
Subject: Why is there a hardcoded \$PATH in config.sh?
Posted by [Andreas Herten](#) on Fri, 09 Aug 2013 11:56:10 GMT
[View Forum Message](#) <> [Reply to Message](#)

Every time I source the config.sh file from my build directory, my \$PATH environment variable is set to the status that it had while compiling.

Is there a particular reason for this?
Why does PandaRoot not extend the \$PATH, like every other program does?

For once, I think this is quite illogical. Why should PandaRoot be the program that takes care about what's in this variable?
It's quite a hassle to always remember that everytime you change your \$PATH, there's always this one shell script where you have to put the changes in (when you finally have found out that this was the reason it wasn't working before...).
Also it might be a point for PandaRoot beginners and occasional users to stumble over.

Subject: Re: Why is there a hardcoded \$PATH in config.sh?
Posted by [Stefano Spataro](#) on Fri, 09 Aug 2013 13:24:01 GMT
[View Forum Message](#) <> [Reply to Message](#)

In reality it is just the opposite. Since the beginners could have another ROOT installed in some root folder, then the PATH is reset in order to avoid possible conflicts. If you use pandaroot, then you have to use ONLY pandaroot
At least, this is what I remember.

Subject: Re: Why is there a hardcoded \$PATH in config.sh?
Posted by [Andreas Herten](#) on Fri, 09 Aug 2013 13:43:28 GMT
[View Forum Message](#) <> [Reply to Message](#)

Wouldn't the better solution to that problem be to prepend the \$PATH variable with PandaRoot's ROOT dir? Like

```
export PATH="$SIMPATH/bin:$PATH"
```

By the way, the line in my config.sh looks like this

```
export  
PATH="/home/ikp1/herten/fairsoft/extpkg/bin:/private/cuda-5.0/bin:/home/ikp1/herten/bin:  
/usr/lib64/mpi/gcc/openmpi/bin:./home/ikp1/herten:/usr/local_64/bin:/usr/local/bin:/usr/bin:/bin:  
/usr/bin/X11:/usr/X11R6/bin:/usr/games:/opt/bin:/usr/lib/mit/bin:/usr/lib/mit/sbin:  
/usr/lib/qt3/bin:/home/ikp1/herten/programme/sublime-text-2"
```

(with no spaces of course)

Subject: Re: Why is there a hardcoded \$PATH in config.sh?

Posted by [Florian Uhlig](#) on Fri, 09 Aug 2013 19:55:19 GMT

[View Forum Message](#) <> [Reply to Message](#)

Hi Andreas,

the reason why we have this strict settings is that we had many problems with wrong environment settings. As Stefano already pointed out people have several different Root or Geant versions.

To find problems with mismatches of libraries took so much time that we decide to have a very strict config file.

You are one of very few users which complain about this issue, so I think for most of the users this strict settings are okay. If you don't like this strict settings the best way is to change the config.sh file and add \$PATH at the end of the of the "export PATH=" line. The disadvantage is that in case you switch between several versions of pandaroot, your PATH variable will become longer and longer.

Ciao

Florian

Subject: Re: Why is there a hardcoded \$PATH in config.sh?

Posted by [Andreas Herten](#) on Mon, 12 Aug 2013 08:01:57 GMT

[View Forum Message](#) <> [Reply to Message](#)

Hey Florian,

Ok, debugging experience is a very good reason for doing it this way. I accept . And will think about modifying my own config.sh.

Just for curiosity, another question:

Wouldn't the way I suggested defining \$PATH in the previous answer also solve any wrongly set variable? Prepending the PandaRoot-specific paths like

```
export PATH="$SIMPATH/bin:OTHERPATHS/:$PATH"
```

For ROOT / GEANT executables would then be searched first in the PandaRoot directories.

(PandaRoot version switching and \$PATH lengthening would be a problem. That's true. There would be a script needed to strip the old PandaRoot/bin directory out of \$PATH before adding it again (maybe based on the previous \$SIMPATH value)...)

See you,
-Andreas

Subject: Re: Why is there a hardcoded \$PATH in config.sh?

Posted by [Florian Uhlig](#) on Tue, 13 Aug 2013 11:57:38 GMT

[View Forum Message](#) <> [Reply to Message](#)

Hi Andreas,

there is a option USE_PATH_INFO which you can use when running cmake. If the option is true \$PATH is added in the config file.

The default setting is to add the explicit list of directories.

To use this option pass it on the command line

```
cmake <source_dir> -DUSE_PATH_INFO=TRUE
```

Hope this solves your problem.

Ciao

Florian

Subject: Re: Why is there a hardcoded \$PATH in config.sh?

Posted by [Andreas Herten](#) on Tue, 13 Aug 2013 12:59:47 GMT

[View Forum Message](#) <> [Reply to Message](#)

Thanks Florian! It works .

(Btw, interesting svn commit message (your string "\$PATH" got extended to the content of your \$PATH variable) .)
