# JobSub - Bug #11709

# jobsub eats "

02/12/2016 11:26 AM - Dennis Box

| | | | |
|---|---|---|---|
| **Status:** | New | **Start date:** | 02/12/2016 |
| **Priority:** | Normal | **Due date:** | |
| **Assignee:** | Dennis Box | **% Done:** | 0% |
| **Category:** | | **Estimated time:** | 0.00 hour |
| **Target version:** | v1.3.3 | **Spent time:** | 0.00 hour |
| **First Occurred:** | | **Stakeholders:** | |
| **Occurs In:** | | | |

**Description**

see RITM0329060

Dear jobsub experts,

I have a script which takes a whole command (with its own arguments) as one parameter. It works, however, when I am running the script through jobsub_submit, " are eaten and my whole command is split and its parameters are treated as the main script parameter.

Lets say I have a script: runCommand.sh which can take a string as a parameter (my command to run), e.g.

runCommand.sh -a arg1 -b arg2 -c "myCommand -a arg3 -b arg4"

When I pass this to jobsub_submit, e.g.

jobsub_submit -G genie -M --OS=SL6 --resource-provides=usage_model=DEDICATED,OPPORTUNISTIC file://runCommand.sh -a arg1 -b arg2 -c "myCommand -a arg3 -b arg4"

then jobsub understand this like:

runCommand.sh -a arg1 -b arg2 -c myCommand -a arg3 -b arg4

so arg3 and arg4 are treated as runCommand.sh arguments. I tried escape \", but it did not help.

Is there a way to make this work?

Thanks,
Tomek

p.s. I sent this email to jobsub-support@fnal.gov (I found the email address here: https://cdcvs.fnal.gov/redmine/projects/fife/wiki/Submitting_jobs_via_jobsub), but apparently it does not really work?

## History

**#1 - 08/11/2017 06:19 PM - Dmitrii Torbunov**

Hello jobsub experts,

We also encounter this issue in the NOvA group. A quick look at the jobsub indicates that the problem is connected with the way the wrapper script is constructed -- **pylib/groupsettings/JobSettings.py** lines 569-570 show that script options are being constructed as a string, so any difference between opts = [ "opt 1" ] and opts = [ "opt", "1" ] is lost when converted into a string opts = "opt 1".

There is however a simple(but not fully correct) workaround -- when constructing script_args(pylib/groupsettings/JobSettings.py, lines 569-570) we can surround separate arguments by single(double) quotes, which will help bash to determine how to correctly parse arguments. I.e. we can replace

```
for x in settings['script_args']:
script_args = script_args+x+' '
```

by

```
for x in settings['script_args']:
script_args += "'" + x + "' "
```

or we may also safeguard and escape single quotes in the x, like:

```
for x in settings['script_args']:
script_args += "'" + x.replace("'", "'\\''") + "' "
```

Thank you,
Dmitrii

**#2 - 05/17/2019 12:55 PM - Dennis Box**

*- Target version set to v1.3.2*

**#3 - 04/10/2020 12:00 PM - Dennis Box**

*- Target version changed from v1.3.2 to v1.3.3*