

## Arvados - Bug #4726

### [Crunch] run-command should show helpful error message when task.foreach argument is not a list

12/05/2014 03:33 PM - Bryan Cosca

<b>Status:</b>	New	<b>Start date:</b>	12/05/2014
<b>Priority:</b>	Normal	<b>Due date:</b>	
<b>Assigned To:</b>		<b>% Done:</b>	0%
<b>Category:</b>	Crunch	<b>Estimated time:</b>	0.00 hour
<b>Target version:</b>	Arvados Future Sprints		
<b>Description</b>			
<p>A common convention I would do with run-command version 83a9390a05bbffc2e4ea95dd693af3ab3547fa12 from job <a href="#">qr1hi-8i9sb-fqbtg1lcocoefe0</a> is:</p>			
<pre>"command": [ "\$\$(outputname)" ], "outputname": {   "value": {     "list": "iterator",     "index": "0",     "command": "OUTPUT=\$(basename \$(basename \$(iterator))).sam"   },   "iterator": {     "value": {       "group": "input_dir",       "regex": "(.*)_[0-9]_*.sam"     }   } }</pre>			
<p>In "command" of output name, I use some sort of command to play with the name of iterator. Iterator is the name of a file from a collection where task.foreach is taking the name of each file. This convention worked perfectly fine.</p>			
<p>Now, with run-command version ac21f0d45a76294aaca0c0c0fdf06eb72d03368d, I try the same convention in job <a href="#">qr1hi-8i9sb-i7vykyao0wg0f9a</a></p>			
<pre>"command": [ "\$\$(outputname)" ], "outputname": {   "list": "iterator",   "index": "0",   "command": "\$\$(basename \$(basename \$(iterator)))" }, "iterator": {   "group": "input_dir",   "regex": "(.*)fastq.gz" },</pre>			
<p>But get the error:</p>			
<pre>run-command: caught exception 12/5/2014 10:03:50 AM compute29 1 task-print 0 Traceback (most recent call last): 12/5/2014 10:03:50 AM compute29 1 task-print 0 File "/tmp/crunch-src/crunch_scripts/run-command", line 291, in &lt;module&gt; 12/5/2014 10:03:50 AM compute29 1 task-print 0 recursive_foreach(jobp, jobp["task.foreach"]) 12/5/2014 10:03:50 AM compute29 1 task-print 0 File "/tmp/crunch-src/crunch_scripts/run-command", line 262, in recursive_foreach 12/5/2014 10:03:50 AM compute29 1 task-print 0 recursive_foreach(params, fvars) 12/5/2014 10:03:50 AM compute29 1 task-print 0 File "/tmp/crunch-src/crunch_scripts/run-command", line 255, in recursive_foreach 12/5/2014 10:03:50 AM compute29 1 task-print 0 items = get_items(params, params[var]) 12/5/2014 10:03:50 AM compute29 1 task-print 0 File "/tmp/crunch-src/crunch_scripts/run-command", line 235, in get_items 12/5/2014 10:03:50 AM compute29 1 task-print 0 mode = os.stat(value).st_mode</pre>			

12/5/2014 10:03:50 AM compute29 1 task-print 0 OSError: [Errno 2] No such file or directory: 'SRR064287\_1'

So maybe command is trying to look for iterator in the input collection now?

I've got a workaround by just doing the command within script\_parameters, but I just wanted to note this behavior.

## History

---

### #1 - 12/05/2014 03:34 PM - Bryan Cosca

- Description updated

### #2 - 12/05/2014 03:40 PM - Radhika Chippada

- Target version set to Bug Triage

### #3 - 12/11/2014 04:10 PM - Brett Smith

- Subject changed from Run-command -- Command within a variable does not work anymore to [Crunch] run-command feature regression: Command within a variable does not work anymore

- Category set to Crunch

### #4 - 12/18/2014 10:10 PM - Brett Smith

- Subject changed from [Crunch] run-command feature regression: Command within a variable does not work anymore to [Crunch] run-command crashes when task.foreach argument is not a list

Bryan,

This job failed because of the way task.foreach was specified. task.foreach expects all of its arguments to refer to lists, and it generates tasks based on the Cartesian product of their elements. Because of this, when it resolves the first outputname to SRR064287\_1, it says, "Oh, this isn't a list, it's a string. This must refer to a directory with files I should iterate, or a file that contains a list of things to iterate." So it tries to find that thing on the filesystem, fails, and crashes.

If I'm following right, you're trying to do something different: you don't want a Cartesian product, but you just want pairs of input files (in iterator) to output names. This would be best accomplished by pointing task.foreach at the input list, then wrapping a couple of the command arguments in foreach so that it generates both the input name (passed in directly) and the output name (using your command). Hopefully your current versions are giving you that. In the meantime, we'll see about improving the error reporting in this case, so things are less confusing next time it happens.

### #5 - 12/18/2014 10:10 PM - Brett Smith

- Story points set to 0.5

### #6 - 12/19/2014 06:13 PM - Tom Clegg

- Subject changed from [Crunch] run-command crashes when task.foreach argument is not a list to [Crunch] run-command should show helpful error message when task.foreach argument is not a list

### #7 - 12/19/2014 06:13 PM - Tom Clegg

- Target version changed from Bug Triage to Arvados Future Sprints