# RhodeCode Control - Bug #5501

## rccontrol throwing rccontrol.lib.exceptions.SupervisorFailedToStart

12.10.2018 14:37 - dan m

| | | | |
|---|---|---|---|
| **Status:** | New | **Start date:** | 12.10.2018 |
| **Priority:** | Normal | **Due date:** | |
| **Assignee:** | | **% Done:** | 0% |
| **Category:** | | **Estimated time:** | 0.00 hour |
| **Target version:** | | | |
| **Sorting:** | | **Affected Version:** | |
| **Commit Number:** | | | |

**Description**

Attempting to run any command that invokes supervisord gives me the following STE:

```
$ rccontrol status community-1
RHODECODE CONTROL VERSION: 1.19.2
Starting supervisord.
Traceback (most recent call last):
File "/opt/rhodecode/store/i58g29lpsgfr7jxsirfx2xsjhddllrc5-python2.7-supervisor-3.3.4/bin/.supervisord-wrapped", line 12, in
sys.exit(main())
File
"/opt/rhodecode/store/i58g29lpsgfr7jxsirfx2xsjhddllrc5-python2.7-supervisor-3.3.4/lib/python2.7/site-packages/supervisor/supervisord.py", line 357, in main
go(options)
File
"/opt/rhodecode/store/i58g29lpsgfr7jxsirfx2xsjhddllrc5-python2.7-supervisor-3.3.4/lib/python2.7/site-packages/supervisor/supervisord.py", line 367, in go
d.main()
File
"/opt/rhodecode/store/i58g29lpsgfr7jxsirfx2xsjhddllrc5-python2.7-supervisor-3.3.4/lib/python2.7/site-packages/supervisor/supervisord.py", line 67, in main
self.options.set_rlimits_or_exit()
File
"/opt/rhodecode/store/i58g29lpsgfr7jxsirfx2xsjhddllrc5-python2.7-supervisor-3.3.4/lib/python2.7/site-packages/supervisor/options.py",
line 1398, in set_rlimits_or_exit
soft, hard = resource.getrlimit(res)
resource.error: (38, 'Function not implemented')
Supervisord did not start properly.
Traceback (most recent call last):
File
"/opt/rhodecode/store/hvmz2dqbl1ha75bnw4gh7kgy6wqk62jz-python2.7-rhodecode-control-1.19.2/bin/..rccontrol-wrapped-wrapped",
line 12, in
sys.exit(cmd())
File
"/opt/rhodecode/store/hvmz2dqbl1ha75bnw4gh7kgy6wqk62jz-python2.7-rhodecode-control-1.19.2/lib/python2.7/site-packages/rccontrol/main.py", line 54, in cmd
File "/opt/rhodecode/store/bbsdjfz28lj6g8s90wbj6i746nwxhh2s-python2.7-click-6.6/lib/python2.7/site-packages/click/core.py", line
716, in call
return self.main(*args, **kwargs)
File "/opt/rhodecode/store/bbsdjfz28lj6g8s90wbj6i746nwxhh2s-python2.7-click-6.6/lib/python2.7/site-packages/click/core.py", line
696, in main
rv = self.invoke(ctx)
File "/opt/rhodecode/store/bbsdjfz28lj6g8s90wbj6i746nwxhh2s-python2.7-click-6.6/lib/python2.7/site-packages/click/core.py", line
1060, in invoke
return _process_result(sub_ctx.command.invoke(sub_ctx))
File "/opt/rhodecode/store/bbsdjfz28lj6g8s90wbj6i746nwxhh2s-python2.7-click-6.6/lib/python2.7/site-packages/click/core.py", line
889, in invoke
return ctx.invoke(self.callback, **ctx.params)
File "/opt/rhodecode/store/bbsdjfz28lj6g8s90wbj6i746nwxhh2s-python2.7-click-6.6/lib/python2.7/site-packages/click/core.py", line
534, in invoke
```

```
return callback(*args, **kwargs)
File "/opt/rhodecode/store/bbsdjfz28lj6g8s90wbj6i746nwxhh2s-python2.7-click-6.6/lib/python2.7/site-packages/click/decorators.py", line 64, in new_func
return ctx.invoke(f, obj, *args[1:], **kwargs)
File "/opt/rhodecode/store/bbsdjfz28lj6g8s90wbj6i746nwxhh2s-python2.7-click-6.6/lib/python2.7/site-packages/click/core.py", line 534, in invoke
return callback(*args, **kwargs)
File "/opt/rhodecode/store/hvmz2dqbl1ha75bnw4gh7kgy6wqk62jz-python2.7-rhodecode-control-1.19.2/lib/python2.7/site-packages/rccontrol/status.py", line 21, in cmd
File "/opt/rhodecode/store/hvmz2dqbl1ha75bnw4gh7kgy6wqk62jz-python2.7-rhodecode-control-1.19.2/lib/python2.7/site-packages/rccontrol/lib/utils.py", line 431, in wrapper
File "/opt/rhodecode/store/hvmz2dqbl1ha75bnw4gh7kgy6wqk62jz-python2.7-rhodecode-control-1.19.2/lib/python2.7/site-packages/rccontrol/status.py", line 34, in main
File "/opt/rhodecode/store/hvmz2dqbl1ha75bnw4gh7kgy6wqk62jz-python2.7-rhodecode-control-1.19.2/lib/python2.7/site-packages/rccontrol/lib/context.py", line 1951, in status
File "/opt/rhodecode/store/hvmz2dqbl1ha75bnw4gh7kgy6wqk62jz-python2.7-rhodecode-control-1.19.2/lib/python2.7/site-packages/rccontrol/lib/context.py", line 1059, in get_process_status
File "/opt/rhodecode/store/hvmz2dqbl1ha75bnw4gh7kgy6wqk62jz-python2.7-rhodecode-control-1.19.2/lib/python2.7/site-packages/rccontrol/lib/context.py", line 1049, in get_process_info
File "/opt/rhodecode/store/hvmz2dqbl1ha75bnw4gh7kgy6wqk62jz-python2.7-rhodecode-control-1.19.2/lib/python2.7/site-packages/rccontrol/lib/context.py", line 1326, in xmlrpc
rccontrol.lib.exceptions.SupervisorFailedToStart
```

RC version: 1.19.2
What's installed: 1 CE instance, 1 VCSServer instance
OS: CentOs 6.10 (updates current up to the past week or so)

Comments to follow

## History

### #1 - 12.10.2018 15:18 - dan m

How I got to this point:

I checked out rccontrol and installed a CE and vcsserver instances. The setup of the instances, as well as the MySQL and nginx configs seemed to be ok. I've done the setup a few times for both EE and CE, and though I'm not an expert, there wasn't anything I ran into that stood out as an issue. No warnings or errors that I recall.

I was troubleshooting a login issue - The UI was accepting the admin login with the credentials I gave during setup, but not actually logging me into the UI. There were no auth errors in the logs, and as far as they were concerned, things seemed to be normal. At the time I suspected I had a proxy / prefix configuration issue as that was what I was working on before the login issue. During the course of troubleshooting, I wanted to trim down other variables so I start disabling other thridparty services on the server (this server runs evaluations of several products and occasionally I need to step back to make sure one is not interfering with another). To sanity check service setup, I rebooted the server. It was at that time that I started getting the supervisord error shown in the OP.

I didn't have too much to work with - this looks like a init / pre-run step, so there are no log entries that I see in the instance logs. I can say that I get this error whenever I try to use the status, upgrade, start, or stop command.

After some frustration, I moved /opt/rhodecode (store location), moved my ~/.rccontrol directories, and re-ran my instance setups using the same rccontrol. Again, no issues of note. I got back to my original state (running, but no login). No errors using rccontrol commands. I went off to troubleshoot a different thridparty service, and during that troubleshooting I ended up doing a reboot at some point. During that troubleshooting I noticed the rhodecode services weren't running, so I came back and the supervisord error was back. I repeated my steps and it seems to me that I can do a fresh install and have everything work, including installations, upgrades (at least the checking part was testable), starts and stop without any issues. Each time I rebooted I start to get the error again. I was hoping this would be something as simple as an environment variable not set or something like that, but I'm just not familiar enough rhodecode or python to troubleshoot it on my own at this point.

**#2 - 12.10.2018 15:21 - dan m**

I should note that during the re-installs I did not wipe the MySql databases. That was an oversight as I was trying to have a fresh state. Hopefully that was not impactful to my test. If I have time I could re-test if anyone thinks it would be useful.

**#3 - 13.10.2018 09:40 - Marcin Kuzminski [staff]**

This is an odd one.

Based on this error: soft, hard = resource.getrlimit(res)

I'm suspecting maybe selinux?

Installation doesn't really set any enviromental variables, so i doubt this would be it.

**#4 - 14.10.2018 23:24 - dan m**

> I'm suspecting maybe selinux?

I can give that a try tomorrow and report back. When I re-test, I plan to disable selinux, and then redo the same steps as the last paragraph in comment #1. If there are any other steps that I should add or any other data collection that would be useful, please let me know.

**#5 - 17.10.2018 15:38 - dan m**

I went back to setup a new test with selinux disabled, but as I was doing so, I realized it already was. I tried again (with a clean mysql db this time), and had the same issue. With some more investigation, I found this note about old repositories format: https://docs.rhodecode.com/RhodeCode-Enterprise/known-issues/known-issues.html

That is definitely something that would have been biting me, as I had test repos under a (probably very, very) old version of rhodecode. I repeated the same setup steps again, but this time with a completely fresh repo directory. I can't say for sure this is the trigger, but now I have a working supervisord, and a working UI (referenced in OP). I have not repeated any tests with the old repo folder, but I could if someone wants the datapoint from me. It does seem that the only way this could have been a problem for a running status or the tangential UI issue is if the community install had an issue that wasn't apparent to the user. That's just guestwork, but it would seem odd that the supervisord would crap out due to a runtime check - I would more expect community / enterprise to do such things. Again, I'm no expert on rhodecode or python.

I can provide more details, and am willing to re-run tests if anyone would like. I have a snapshot of the server I can go back if need be. Otherwise, this issue could probably be closed as my original is resolved for now.

**#6 - 17.10.2018 17:56 - Marcin Kuzminski [staff]**

Hi Dan,

Thanks for detailed explanation. Yes indeed it's very odd problem that supervisord fails, and not rhodecode itself.
Happy this is now resolved, and we'll keep this task for reference in case it ever happens again.

Best,

#### #7 - 23.10.2018 22:39 - Joseph Sible

I just ran into this problem myself, and I think I got to the bottom of it. Important to note is that this happens on CentOS 6.10, which even when fully updated runs the rather old kernel 2.6.32-754.6.3.el6.x86_64. Here's an easy way to reproduce the root cause of the problem on that system:

```
-bash-4.1$ /opt/rhodecode/store/6rjyfgq8yrnzwsc4x9b6mlwi6h9qhay6-python-2.7.15/bin/python2.7 -c 'import resource; print
resource.getrlimit(resource.RLIMIT_NPROC)'
Traceback (most recent call last):
File "<string>", line 1, in <module>
resource.error: (38, 'Function not implemented')
-bash-4.1$
```

And here it is with an older Python working fine from before the upgrade:

```
-bash-4.1$ /opt/rhodecode/store/g7qzsl3a8sb23daxgs3gin4j22dhs3sw-python-2.7.12/bin/python2.7 -c 'import resource; print
resource.getrlimit(resource.RLIMIT_NPROC)'
(1024, 22720)
-bash-4.1$
```

Running each through strace reveals these two different syscalls:

```
prlimit64(0, RLIMIT_NPROC, NULL, 0x7ffd089a54c0) = -1 ENOSYS (Function not implemented)
getrlimit(RLIMIT_NPROC, {rlim_cur=1024, rlim_max=22720}) = 0
```

The root cause is that CentOS's kernel is old enough that it doesn't have the prlimit64 syscall, but the new Python binary was built to require it. It can be fixed by rebuilding the Python binary against the oldest kernel that you still support.

#### #8 - 23.10.2018 22:57 - Marcin Kuzminski [staff]

Hi Joseph,

First, thanks for a detailed investigation. Surprised to see it was due to the python and packages updated. Unfortunately, we're having a limited capability of re-building the python against such old kernel. The build one comes via the NIX Package manager which affair has a 4.X kernel to build against.

we need to fix if this one can be really solved via packaging

#### #9 - 23.10.2018 23:10 - Joseph Sible

Okay, given that, I did a bit more investigation. It turns out that Python always just calls the getrlimit C function, and it's glibc that decides which of the two syscalls to use (so it was actually the new glibc used by the new Python that causes the problem). This turns out to be good news, as glibc makes it easy to build for an older kernel than the one you're running: just pass "--enable-kernel=2.6.32" as a ./configure parameter when building it. Is this solution possible with the way RhodeCode does packaging?

#### #10 - 23.10.2018 23:18 - Marcin Kuzminski [staff]

In 4.13+ glibc was updated indeed.

I believe we're able to pass extra compilation flags, but we need to check it.

How critical is that problem?

#### #11 - 23.10.2018 23:19 - Joseph Sible

It's a blocker. Once we upgrade, we can't start RhodeCode or otherwise use rccontrol at all, as everything we try to do with it immediately fails with that message.

#### #12 - 23.10.2018 23:27 - Marcin Kuzminski [staff]

Ok,

We're about to release 4.14 now, so we'll do this top pririty to check the compilation option for glibc

#### #13 - 23.10.2018 23:46 - Joseph Sible

So it turns out this won't be quite that easy. Glibc 2.26 dropped support for kernels older than 3.2. In https://github.com/NixOS/nixpkgs/commit/87acb2b9fdb990ccb9f94a061b9bc21e4d31e0e1 NixOS already patched around this check so that it isn't totally, 100% broken, but they didn't patch this particular syscall. Fixing this will either mean sticking to glibc 2.25 for now, or getting a patch to them that actually fixes it.

#### #14 - 24.10.2018 00:32 - Marcin Kuzminski [staff]

Based on this it might be a real problem.

We cannot support older glibc, one reason we did package update was to handle later glibc, since NEW operating systems started to produce problems with old glibc + python and few other libs needed an upgrade to fix some security problems.

potentially NIX could fix this, but it would take a long time until that lands...

What is your capability for a kernel upgrade on the machine?

#### #15 - 24.10.2018 00:49 - Marcin Kuzminski [staff]

I also think about potentially patching supervisord. The option to read/set limits is not really needed in RhodeCode context.

https://github.com/Supervisor/supervisor/blob/master/supervisor/options.py#L1427

Would it be possible you edit the
/opt/rhodecode/store/i58g29lpsgfr7jxsirfx2xsjhddllrc5-python2.7-supervisor-3.3.4/lib/python2.7/site-packages/supervisor/options.py

and change limits = [] before:
https://github.com/Supervisor/supervisor/blob/master/supervisor/options.py#L1420

**#16 - 24.10.2018 00:52 - Marcin Kuzminski [staff]**

Futhermore i see we

minfds='1024',
minprocs='200',

Inside supervisord.ini, need to check WHY, and maybe removing those from supervisord.ini would not read/set limits at all.

**#17 - 24.10.2018 14:16 - Joseph Sible**

Removing those lines from supervisord.ini doesn't fix the problem. Looking at the source of supervisord reveals that it calls getrlimit unconditionally, and the only way to make it not do so would be to patch it (the patch you suggested does indeed let me start). I'm still concerned, though, because Nix's patch to glibc didn't restore *any* of the functionality that was removed upstream (it just patched out their version check), and I'm expecting this syscall to just be the tip of the iceberg. If more problems of this nature appear, and downgrading glibc isn't an option, then the only two practical ways to move forward would be to patch glibc ourself (or get Nix to do it), or drop support for CentOS 6.

**#18 - 24.10.2018 15:01 - Marcin Kuzminski [staff]**

This is tricky one. we need to discuss this within the team

I read somewhere that glibc 2.27 would bring better old kernel support, but that's not proven info.
We might get lucky here, and just disabling this supervisord check would get us going good.

IMHO once we get the start phase things should be good as we don't do anything special during the operation itself.