

## FreeNAS - Bug #6143

### Need background task manager for certain long-running tasks

09/20/2014 09:12 AM - Jordan Hubbard

<b>Status:</b> Closed	
<b>Priority:</b> Nice to have	
<b>Assignee:</b> Suraj Ravichandran	
<b>Category:</b> Middleware	
<b>Target version:</b> N/A	
<b>Seen in:</b> 9.3-M3	<b>Needs Merging:</b> Yes
<b>Severity:</b> New	<b>Needs Automation:</b> No
<b>Reason for Closing:</b>	<b>Support Suite Ticket:</b> n/a
<b>Reason for Blocked:</b>	<b>Hardware Configuration:</b>
<b>Needs QA:</b> Yes	<b>ChangeLog Required:</b> No
<b>Needs Doc:</b> Yes	
<b>Description</b> Just to name a few off the top of my head: 1. Importing data from another volume 2. Downloading a jail template in order to create a jail 3. Downloading a plugin 4. Any of the current tasks in the tasks menu  What makes all of this especially interesting is that some of these things are already in the background (like #4) but provide <b>no progress info</b> and others, like 1-3, are synchronously in the foreground and block the UI. In all cases, the tasks manager needs to be able to handle clients connecting, asking for something to be done, and disconnecting again before that "something" is completed but possibly reconnecting again periodically to ask for status information. Or, as in the case of #4, there may be two connections for the same task - one which is blocking, waiting for the task to complete so that it can move to the next stage, and another (the UI) which is simply connecting for progress info but may disconnect before completion. In other words, this is completely asynchronous and the task manager is the only one who really <b>must</b> monitor all of its tasks all the way from start to finish and also save their status when they complete, just in case someone checks in later to ask for the progress of a task which has completed and exited.	
<b>Related issues:</b> Is duplicate of FreeNAS - Feature #4795: Rsync Progress <b>Closed: Duplicate 09/12/2014</b> Is duplicate of FreeNAS - Feature #3194: Rsync Monitor/Session GUI <b>Closed</b>	

### History

#### #1 - 09/21/2014 12:02 AM - Suraj Ravichandran

- Status changed from Unscreened to Screened

#### #2 - 09/25/2014 06:00 AM - William Grzybowski

Since the beginning I thought being async was a job for the NMOS.

The whole GUI/middleware is very bad designed to be completely Sync. Changing those things to be async at this point (1 week away from BETA) would require extensive rework in the request handling from the UI side depending on the level of complexity where are willing to take, since the workflow imposed in the UI implementation expects that models part of the generic code path will always do stuff once the request is returned.

Suraj has been asking me several aspects of how to make this work and what should be changed to support this feature and I can't really tell him since there are some questions open (or I wasn't able to catch between the lines in this ticket description):

- Say you are importing data and dismiss the dialog. Are you going to allow the user to start another import, simultaneously?
- How would you check the progress of things? \* If the "Add jail" feature is not allowed to be concurrent this is easy, if there is there is a task running just show the progress if you click again in the "Add Jail", but this is ghetto. Do we need some sort of separate UI to show the current status of running tasks?
- What about conflicting tasks? Adding a jail and installing a plugin at the same time would probably make things go haywire.

**#3 - 09/27/2014 05:19 AM - Jordan Hubbard**

So, just to clarify some of the angst above. This is not an early attempt to make everything async - I agree that this would be counter-productive in 9.3 given all the other stuff that wouldn't work. This is just an attempt to take **certain** operations and push them into the background, the list I initially gave being more of an "idea list" than an actual specific list. Creating a jail or plugin, maybe that's indeed not a good target for this. Doing a file import, however, seems like it would be since that currently can block the UI for an unspecific (huge) amount of time.

As to how to "check in" with the UI after leaving it, I think Jakob already showed a pretty good way how this could work with his backup feature, which I think would be another good candidate for this daemon in a 9.3 software update. Even if this daemon just does one thing in 9.3, it would be a start down that path that will teach us some useful things in advance of 10.0, which will be async everywhere.

**#4 - 10/02/2014 01:39 PM - Josh Paetzel**

- Target version changed from 9.3-BETA to 49

**#5 - 10/10/2014 06:48 AM - William Grzybowski**

- Is duplicate of Feature #4795: Rsync Progress added

**#6 - 10/10/2014 06:49 AM - William Grzybowski**

- Is duplicate of Feature #3194: Rsync Monitor/Session GUI added

**#7 - 01/12/2015 04:53 PM - Suraj Ravichandran**

- Status changed from Screened to Closed

Option 1 is done and the rest is taken care (being worked on in 10)

**#8 - 07/19/2017 08:18 AM - Kris Moore**

- Target version changed from 49 to N/A