

## RPM Support - Issue #1823

### RPMs partially downloaded

04/06/2016 09:16 PM - mhrivnak

|  |                                  |
|--|----------------------------------|
| <b>Status:</b> CLOSED - CURRENTRELEASE | <b>Start date:</b>               |
| <b>Priority:</b> High                  | <b>Due date:</b>                 |
| <b>Assignee:</b> ttereshc              | <b>Estimated time:</b> 0:00 hour |
| <b>Category:</b>                       |                                  |
| <b>Sprint/Milestone:</b>               |                                  |
| <b>Severity:</b> 2. Medium             | <b>Groomed:</b> No               |
| <b>Version:</b> 2.6.5                  | <b>Sprint Candidate:</b> No      |
| <b>Platform Release:</b> 2.12.0        | <b>Tags:</b> Pulp 2              |
| <b>OS:</b>                             | <b>Sprint:</b> Sprint 7          |
| <b>Triaged:</b> Yes                    | <b>Quarter:</b>                  |

**Description**

There have been multiple reports of RPMs ending up in /var/lib/pulp/content/ with either 0 bytes, or partially downloaded. Looking at the 6.1 code, it is difficult to identify how that is possible.

You can see here that in pulp 2.6, an rpm does not get saved into the DB until after validation has happened, and the file has been moved into place without errors. Katello has assured us that they supply "validate: true" with each sync request, so validation should be happening.

[https://github.com/pulp/pulp\\_rpm/blob/2.6-release/plugins/pulp\\_rpm/plugins/importers/yum/listener.py#L85](https://github.com/pulp/pulp_rpm/blob/2.6-release/plugins/pulp_rpm/plugins/importers/yum/listener.py#L85)

And yet, users are seeing this happen, so we need to investigate further.

**Related issues:**

|  |                                |
|--|--------------------------------|
| Related to Docker Support - Issue #2142: Units created with 0-byte files when... | <b>CLOSED - CURRENTRELEASE</b> |
| Blocked by RPM Support - Issue #2190: Unit is associated with the repo before... | <b>CLOSED - CURRENTRELEASE</b> |

#### Associated revisions

##### Revision 96fd275c - 11/08/2016 10:43 PM - ttereshc

Verify RPM/SRPM/DRPM unit at its final location

closes #1823 <https://pulp.plan.io/issues/1823>

##### Revision a583b757 - 11/08/2016 10:48 PM - ttereshc

Verify size of a file at its final location

re #1823 <https://pulp.plan.io/issues/1823>

##### Revision a583b757 - 11/08/2016 10:48 PM - ttereshc

Verify size of a file at its final location

re #1823 <https://pulp.plan.io/issues/1823>

#### History

##### #1 - 04/08/2016 05:08 PM - mhrivnak

- Triaged changed from No to Yes

**#3 - 06/10/2016 03:23 PM - Anonymous**

- Groomed changed from No to Yes
- Sprint Candidate changed from No to Yes

**#4 - 06/10/2016 03:25 PM - Anonymous**

- Sprint/Milestone set to 22
- Groomed changed from Yes to No
- Sprint Candidate changed from Yes to No

**#5 - 07/05/2016 02:29 PM - mhrivnak**

- Sprint/Milestone changed from 22 to 23

**#6 - 07/14/2016 04:36 PM - ttereshc**

- Status changed from NEW to ASSIGNED
- Assignee set to ttereshc

**#7 - 07/25/2016 04:13 PM - mhrivnak**

- Sprint/Milestone changed from 23 to 24

**#8 - 07/27/2016 04:20 PM - ttereshc**

An attempt to do an audit of the code path for downloading RPMs (pulp 2.6-release, pulp\_rpm 2.6-release, nectar 1.3.3-1) results in the following. Principal suspicions so far:

- non-atomic `shutil.move` could silently collide with each other
- pre-move content validation may not notice corrupted content
- possible race condition on concurrent downloads

`shutil.move` may make the process of putting temp file into its final location non-atomic (between different FS) and thus unsafe unless proper locking is used.

[Copy to the final location](#) + concurrent downloads of the same RPM could be an issue.

Assuming that `shutil.move` can't silently fail there are few potential scenarios in which RPM file could be corrupted:

Scenario 1

- Upstream repo A and repo B which both contain the same RPM.
- RPM from both repos is synced nearly at the same time.
- Download of RPM from repo B started before download from repo A was finished (second download started because there is no reference for RPM in DB yet).
- Download from repo A succeeded, validation succeeded, file was moved to the final location and is not corrupted anyhow, reference to this RPM is saved into the DB.
- Download from repo B succeeded, validation succeeded, there was some problem during copy to the final location.
- Because the final location will be the same in both cases, the file could be corrupted at the end.

Scenario 2

- One repo, two nearly concurrent downloads of the same RPM from the same repo
- First download succeeded, validation succeeded but before it was copied to the final location second download started to write into the same file in the working directory.
- Possibly corrupted file is moved to the final location and saved into the DB.

Uncertainties:

- Both scenarios seem to me to be very unlikely but in most cases RPM were partially downloaded in case of some network problems, so the probability of the race conditions is getting higher.
- How to end up with concurrent downloads of the same RPM from the same repo?
- I do not think it is possible if there is only one feed specified for the repo. I followed the code path and all the locks look good to me.
- So [alternate content sources](#) are under more suspicion, [several queues, one for each source](#), are used in this case. More investigation needed.

Suggestion/workaround for both scenarios so far:

Validate file at its final location (copy file from working directory into the temp name into the final directory, then validate and atomically rename it to the final name)

Any thoughts? or issues in my logic?

**#9 - 08/09/2016 09:43 PM - mhrivnak**

- Related to Issue #2142: Units created with 0-byte files when sync runs out of disk space added

**#10 - 08/09/2016 09:55 PM - mhrivnak**

That all seems reasonable. I like the idea of validating it in the final location but with a different filename, and doing the atomic rename.

I added an association to issue [#2142](#), which may also reveal a possible cause. That one seems to be quite reproducible with docker content. I could not reproduce that behavior with an rpm sync, but it's possible that something has changed in the yum importer since 2.6 to explain it. Maybe whatever problem is present in the docker importer used to also affect the yum importer.

**#11 - 08/15/2016 08:18 PM - mhrivnak**

- Sprint/Milestone changed from 24 to 25

**#12 - 08/23/2016 12:13 PM - ttereshc**

- Blocked by Issue #2190: Unit is associated with the repo before it is copied to the final location added

**#13 - 09/02/2016 05:30 PM - ttereshc**

- Status changed from ASSIGNED to POST

<https://github.com/pulp/pulp/pull/2730>

[https://github.com/pulp/pulp\\_rpm/pull/974](https://github.com/pulp/pulp_rpm/pull/974)

**#14 - 11/14/2016 01:47 PM - ttereshc**

- Status changed from POST to MODIFIED

Applied in changeset [96fd275c2b8b3e6f1ce866a21af2ce47736c8f39](#).

**#15 - 11/14/2016 01:53 PM - ttereshc**

Now we are doing an additional validation during copy to the final location of the file, so there will be errors during sync when files are corrupted. However, since root cause of the corrupted files is still unclear, this approach does not fix the root issue.

**#16 - 11/21/2016 04:41 PM - semyers**

- Platform Release set to 2.12.0

**#17 - 01/18/2017 12:42 AM - semyers**

- Status changed from MODIFIED to 5

**#18 - 02/01/2017 09:56 PM - semyers**

- Status changed from 5 to CLOSED - CURRENTRELEASE

**#19 - 03/08/2018 08:08 PM - bmbouter**

- Sprint set to Sprint 7

**#20 - 03/08/2018 08:08 PM - bmbouter**

- Sprint/Milestone deleted (25)

**#21 - 04/15/2019 10:32 PM - bmbouter**

- Tags Pulp 2 added

