RPM Support - Issue #8944

Cannot sync RHEL6 repos from cdn.redhat.com

06/22/2021 12:37 AM - sskracic@redhat.com

Status:	CLOSED - CURRENTRELEASE	**Start date:**	
Priority:	Normal	**Due date:**	
Assignee:	dalley	**Estimated time:**	0:00 hour
Category:			
Sprint/Milestone:			
Severity:	4. Urgent	**Groomed:**	No
Version:	Master	**Sprint Candidate:**	No
Platform Release:		**Tags:**	
OS:		**Sprint:**	Sprint 99
Triaged:	Yes	**Quarter:**	

Description

- a containerized pulp based on pulp-ci-centos image with:

pulpcore-3.13.0 pulp-rpm-3.13.0

I am not sure whether it's a Pulp bug or a CDN data error, but RHEL6 i386 and x86_64 repo syncing repeatedly fails with this error. The repositories are synced with mirror=True and the distribution is set to point to repository (so that the latest repo is always distributed).

```
Repo: Red Hat Enterprise Linux 6 Server from RHUI (RPMs) (6Server-i386)
  Start Time:     2021-06-21 22:24:04
  Finish Time:    2021-06-21 22:28:18
  Elapsed Time:   0:04:13.581718
  Result:         Error
  Exception:      Package id from primary metadata (42003b3621ebf25f9b5fe25b0d2c4f2024958a9d), does not match package id from filelists, other metadata (dcacfd1c930cba9ee112c1055e7a1e94bedd7f3a)
  Traceback:        File "/usr/local/lib/python3.6/site-packages/rq/worker.py", line 1013, in perform_job
    rv  job.perform()
  File "/usr/local/lib/python3.6/site-packages/rq/job.py", line 709, in perform
    self._result = self._execute()
  File "/usr/local/lib/python3.6/site-packages/rq/job.py", line 732, in _execute
    result = self.func(*self.args, **self.kwargs)
  File "/src/pulp-rpm/pulp_rpm/app/tasks/synchronizing.py", line 395, in synchronize
    version = dv.create()
  File "/src/pulpcore/pulpcore/plugin/stages/declarative_version.py", line 149, in create
    loop.run_until_complete(pipeline)
  File "/usr/lib64/python3.6/asyncio/base_events.py", line 484, in run_until_complete
    return future.result()
  File "/src/pulpcore/pulpcore/plugin/stages/api.py", line 225, in create_pipeline
    await asyncio.gather(*futures)
  File "/src/pulpcore/pulpcore/plugin/stages/api.py", line 43, in __call__
    await self.run()
  File "/src/pulp-rpm/pulp_rpm/app/tasks/synchronizing.py", line 674, in run
    await self.parse_repository_metadata(repomd, repomd_files, file_extension)
  File "/src/pulp-rpm/pulp_rpm/app/tasks/synchronizing.py", line 719, in parse_repository_metadata
    file_extension=file_extension,
  File "/src/pulp-rpm/pulp_rpm/app/tasks/synchronizing.py", line 987, in parse_packages
    ).format(pkgid, pkgid_extra)
```

Related issues:

Related to RPM Support - Backport #8962: Backport 8944 to 3.13.1	**CLOSED - CURRENTRELEASE**

Associated revisions

Revision 8393a606 - 06/23/2021 09:35 PM - dalley

Handle duplicate packages in upstream repos

Some repositories have some packages listed in the metadata twice. This shouldn't happen (but it does). createrepo_c deduplicates by virtue of parsing everything into a dict keyed by pkgid, but the iterative parser does not. This eventually results in a mismatch once the iterative parser comes across a package that the createrepo_c primary parser already handled. So we keep a list of pkgid's we've already written out in order to skip them once the iterative parser hits them a 2nd (or 3rd, etc.) time.

closes: #8944 https://pulp.plan.io/issues/8944

Revision 7f4c0dbb - 06/23/2021 09:39 PM - dalley

Handle duplicate packages in upstream repos

Some repositories have some packages listed in the metadata twice. This shouldn't happen (but it does). createrepo_c deduplicates by virtue of parsing everything into a dict keyed by pkgid, but the iterative parser does not. This eventually results in a mismatch once the iterative parser comes across a package that the createrepo_c primary parser already handled. So we keep a list of pkgid's we've already written out in order to skip them once the iterative parser hits them a 2nd (or 3rd, etc.) time.

closes: #8944 https://pulp.plan.io/issues/8944

History

#1 - 06/22/2021 12:37 AM - sskracic@redhat.com

- Description updated

#2 - 06/22/2021 12:21 PM - ipanova@redhat.com

dalley seems like the concern of the package order is a problem

#3 - 06/22/2021 10:08 PM - dalley

- Status changed from NEW to ASSIGNED

- Assignee set to dalley

- Triaged changed from No to Yes

- Sprint set to Sprint 99

T

#4 - 06/22/2021 10:13 PM - ggainey

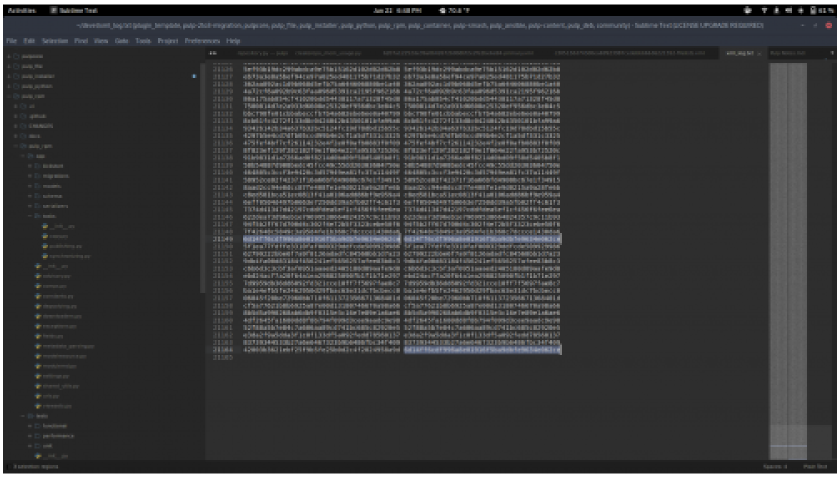
- Private changed from Yes to No

#5 - 06/23/2021 12:58 AM - dalley

- File Screenshot from 2021-06-22 18-48-27.png added

This is pretty strange. It seems like there are a bunch of packages listed multiple times in the XML with the same pkgids.

Here's a list of the order in which Pulp processed the package IDs. On the left is the primary pkgid and on the right is the filelists + other pkgid



The same package (6d14ff6cdf996a8e01916f5ba9dbfe9634e062ce) is also listed a second time in primary.xml, but in a different place.

It's also not the only package with multiple entries. The one listed right above (83739344533b27a6e6467323b9b6486fbc34f409) also has multiple entries. So does (e3da2f9a5dda3f1c0f133df5a092fedd78560137).

For all of these these the actual metadata is the same between the two entries, including the location_href. I can't think of any rational reason for that. And especially not for some of the entries to be in different orders. You would think that even if the same pkgid is present multiple times, the packages would still be in the same order?

Weird. If this is something that can "sometimes happen" then perhaps we need to have some fallback behavior, but I think we should also bring this up to EXD.

#6 - 06/23/2021 03:56 PM - dalley

I think the problem is not "out of order packages". Since primary.xml is parsed into a dict keyed by pkgid, it automatically throws away duplicates. So then when we iterate the dictionary in-order, the 2nd instance of the package is ignored, but that isn't the case with the iterative parser. So the disjoint parsing methods is the cause of the out of ordering problem, which is caused by duplicate package entries in the repo.

As a workaround we could keep a list of pkgids "seen" by the iterative parser and try to skip over them. I've also started a discussion with EXD about it.

#7 - 06/23/2021 09:36 PM - dalley

- Status changed from ASSIGNED to MODIFIED

Applied in changeset 8393a60695dd28a38d515cd7376396734626ae16.

#8 - 06/23/2021 10:53 PM - dalley

- Related to Backport #8962: Backport 8944 to 3.13.1 added

#9 - 06/24/2021 01:44 AM - dalley

- Status changed from MODIFIED to CLOSED - CURRENTRELEASE

Files

Screenshot from 2021-06-22 18-48-27.png	432 KB	06/22/2021	dalley