

RPM Support - Story #4458

Story # 5114 (CLOSED - CURRENTRELEASE): [Epic] Publish features

As a user, I can configure which checksum algorithm to use when creating metadata

02/25/2019 03:09 PM - ttereshc

Status:	CLOSED - CURRENTRELEASE	Start date:	
Priority:	Normal	Due date:	
Assignee:	fao89	% Done:	100%
Category:		Estimated time:	0:00 hour
Sprint/Milestone:	Pulp RPM 3.3.0	Tags:	Katello
Platform Release:		Sprint:	Sprint 70
Groomed:	Yes	Quarter:	
Sprint Candidate:	Yes		
Description			
<p>Support metadata_checksum_type and package_checksum_type options at publish time. These are one-time options and are passed as a parameter for a publication creation call. metadata_checksum_type affects all the repodata, including primary.xml, repomd.xml, etc. package_checksum_type affects package checksum type in all repo metadata files.</p> <p>Origin of checksum types and their precedence (at the top is the one which is used first if present and available):</p> <ul style="list-style-type: none">- configuration at publish time- a remote repo (store the checksum type from the remote source on the repository version model)- Pulp default (sha256) <p>DNF supports different package checksum types in one repo.</p> <p>Publication creation should never fail due to checksum type configuration. At least one checksum type is always available for a package and it should be used if the requested one is not available.</p>			
Related issues:			
Related to RPM Support - Test #4469: Test checksum algorithm when publishing		CLOSED - COMPLETE	

Associated revisions

Revision 6e453c93 - 04/15/2020 06:50 PM - Fabricio Aguiar

Enabled checksum selection when creating metadata

<https://pulp.plan.io/issues/4458> closes #4458

Revision 914750af - 04/17/2020 04:22 PM - Fabricio Aguiar

Fix publication checksum issue

<https://pulp.plan.io/issues/6505> fixes #4458

History

#1 - 02/25/2019 03:09 PM - ttereshc

- Description updated

#2 - 02/25/2019 07:35 PM - ttereshc

Everything should be easy and without any issues when every package in a repo is downloaded. This is the case for:

- immediate download policy, after sync is performed
- uploaded packages
- already downloaded packages copied from other repos

Other cases:

1. On_demand repo, no packages from other sources, first sync from the remote source.

- If the requested checksum type is different from what is available, publish with available checksum type + log a warning that the requested checksum type can't be used

2. Mixture of packages from different on_demand repos

- If on_demand repos have different checksum types, there is no way out -> fail

3. Mixture of downloaded content and in_demand.

- If on_demand content has the common checksum type, it should be used, otherwise fail.

4. On_demand repo, Nth sync, a checksum type in a remote repo changed

- Not sure what to do with it. We can publish using old checksum type but I guess expectation would be that the new checksum type is used.

Concerns:

- we don't have info about remote repo checksum type, we have checksum type info for each package in a RemoteArtifact.
- does it mean that we should inspect all RemoteArtifacts to figure out what the common checksum type is for on_demand content?

#3 - 02/25/2019 08:07 PM - ipanova@redhat.com

Case 1.

I wonder if we could just fail the publish with the helpful explanation message? This way we could be sure this kind of incompatibility is acknowledged by user and he'd update the checksum type or resync the repo with immediate policy.

Case 4.

I don't think we can somehow possibly be aware of the changes happened remotely unless we resync. We publish what we have synced down locally.

#4 - 02/25/2019 10:05 PM - kersom

- Related to Test #4469: Test checksum algorithm when publishing added

#5 - 03/06/2019 07:20 PM - jsherril@redhat.com

My two cents:

- checksum types on metadata are not a security feature
- Publishing a repository should work all the time, even if a consistent checksum type is not used
- Publishing should never fail (unless the user explicitly declares they want it to fail unless some condition is met)

In my experience, Katello's end users are met with pure confusion when any of the above error scenarios occur, which results in a failure. They didn't make any of the decisions which led to the failure, so they have no idea how to resolve it. Its almost always no fault of their own, and it results in a bad user experience.

My recommendation:

1. We have a setting 'metadata_checksum' which controls which checksum types the metadata are recorded with (in repomd.xml) - Optional
2. We have another setting 'preferred_package_checksum_type' (or something like that), which indicates the preference of the checksum type for the user.
3. If desired, we have 3rd setting 'strict_package_checksum_type', which does fail in the cases above, this seems less useful.

Going to your scenarios above:

2. If 'preferred_package_checksum_type' is set, generate with mixed checksum types, log a warning. If its not set, generate with whatever is upstream
3. Similar to 2
4. Yes, i'd expect the new checksum type

Your concerns:

Concerns:

- we don't have info about remote repo checksum type, we have checksum type info for each package in a RemoteArtifact.
- Could this be stored read-only in the remote, and the user could then pass that to the publisher?

#6 - 04/26/2019 10:31 PM - bmbouter

- Tags deleted (Pulp 3)

#7 - 01/27/2020 07:11 PM - ttereshc

- Description updated

Updated the description based on discussions with Katello and DNF.

#8 - 01/27/2020 07:15 PM - ttereshc

- Checklist item [] Support metadata_checksum_type option for publication added
Checklist item [] Support package_checksum_type option for publication added
Checklist item [] Store remote source metadata checksum type on the repository version added

#9 - 01/27/2020 08:07 PM - CodeHeeler

- Groomed changed from No to Yes

#10 - 01/27/2020 09:00 PM - CodeHeeler

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

#11 - 01/27/2020 09:00 PM - CodeHeeler

- Sprint set to Sprint 65

#12 - 02/07/2020 03:20 PM - rchan

- Sprint changed from Sprint 65 to Sprint 66

#13 - 02/21/2020 03:34 PM - rchan

- Sprint changed from Sprint 66 to Sprint 67

#14 - 02/25/2020 06:32 PM - ttereshc

- Sprint/Milestone set to Pulp 3.x RPM (Katello 3.16)
- Parent task set to #5114

#15 - 03/06/2020 04:09 PM - rchan

- Sprint changed from Sprint 67 to Sprint 68

#16 - 03/20/2020 04:21 PM - rchan

- Sprint changed from Sprint 68 to Sprint 69

#17 - 03/23/2020 01:33 PM - CodeHeeler

- Status changed from ASSIGNED to NEW
- Assignee deleted (CodeHeeler)

#18 - 03/23/2020 01:33 PM - CodeHeeler

- Sprint Candidate changed from No to Yes

#19 - 03/24/2020 04:01 PM - fao89

- Status changed from NEW to ASSIGNED
- Assignee set to fao89

#20 - 03/24/2020 10:33 PM - pulpbot

- Status changed from ASSIGNED to POST

PR: https://github.com/pulp/pulp_rpm/pull/1655

#21 - 04/03/2020 06:14 PM - rchan

- Sprint changed from Sprint 69 to Sprint 70

#22 - 04/15/2020 07:32 PM - Anonymous

- Status changed from POST to MODIFIED
- % Done changed from 0 to 100

Applied in changeset [6e453c934d2b07db6ba58fd97ecb58a21a1b460a](#).

#23 - 04/15/2020 09:05 PM - pulpbot

PR: https://github.com/pulp/pulp_rpm/pull/1674

#24 - 04/23/2020 10:04 PM - dalley

- *Sprint/Milestone changed from Pulp 3.x RPM (Katello 3.16) to Pulp RPM 3.3.0*

#25 - 04/23/2020 10:07 PM - dalley

- *Status changed from MODIFIED to CLOSED - CURRENTRELEASE*

#26 - 05/08/2020 07:49 PM - ggainey

- *Tags Katello added*

- *Tags deleted (Katello-P2)*