RPM Support - Story #4162

Story # 4762 (CLOSED - CURRENTRELEASE): [Epic] As a user, I can copy content

Story # 6017 (CLOSED - CURRENTRELEASE): [Epic] As a user, when copying content, dependencies of that content are also copied

As a user, I have dependency solving when copying modules

11/15/2018 04:20 PM - milan

Status:	CLOSED - CURRENTRELEASE	**Start date:**	
Priority:	Normal	**Due date:**	
Assignee:	dalley	**% Done:**	100%
Category:		**Estimated time:**	0:00 hour
Sprint/Milestone:	Pulp RPM 3.3.0		
Platform Release:		**Tags:**	
Groomed:	No	**Sprint:**	
Sprint Candidate:	No	**Quarter:**	

Description

When modules are copied between repos, the following are also copied:

- RPM artifacts
- Other modules marked as dependencies
- The module default for that module, if one exists

Related issues:

Related to RPM Support - Story #3740: Implement modularity content dependency...	**CLOSED - CURRENTRELEASE**
Related to RPM Support - Test #4364: Implement modularity content dependency ...	**CLOSED - COMPLETE**

Associated revisions

Revision 9078fab4 - 03/10/2020 06:57 PM - dalley

Add dependency solving for modulemd and modulemd_defaults

- Copy modules that are dependencies of other modules
- Copy modules when their artifacts are copied
- Copy the modulemd_default if it exists when a module it references is copied

re: #4768 https://pulp.plan.io/issues/4768 closes: #4162 https://pulp.plan.io/issues/4162

History

#1 - 11/15/2018 04:21 PM - milan

- Related to Story #3740: Implement modularity content dependency solving added

#2 - 02/08/2019 05:42 AM - dalley

One thing I would like to change in the Pulp 3 implementation vs. Pulp 2 is to pull depsolving completely out of the plugin and into it's own package such as "pulp_solv". We can define a nice clean API that can perform the operations needed for Pulp's use cases with a minimum of hassle, and then unit test the crap out of it. Apart from being a massive help for testing, it'll also help prevent it from making a giant confusing mess like it kind of does in Pulp 2.

Potentially, we could even make it generic enough to be used by both the RPM and Debian plugins, and maybe some other platforms libsolv supports, assuming that libsolv is the best option for those other platforms (which I don't know to be the case).

#3 - 02/12/2019 05:59 PM - dalley

Another problem is the dependencies, such as libsolv and libmodulemd. These are typically shipped as system packages, and aren't available on all platforms.

A Libmodulemd oackage doesn't exist in the greater Debian ecosystem, nor does (as far as I can tell) a package providing Python bindings for libsolv.

Our options are:

- Use containers
- Put in the work to make those packages into pip-installable Python packages
- Tell RPM users to install the RPM packages, and Debian/Ubuntu/Mac users are SOL

#4 - 02/12/2019 06:09 PM - dalley

Also, via discussion with Justin Sherrill, we may need to modify our implementation *significantly* from what Pulp 2 is capable of.

The issue, as I understand it, is that Fedora and RHEL8/CentOS 8 will have separate module repos from the standard RPM repos, and that modules can depend on RPMs in the standard repo.

In that case, it would mean that to do recursive copy, you need to support multiple input repos (the repo where the module comes from, and the repo where the standard RPM deps come from), and correspondingly multiple output repos, so that the modules are copied from one module-only repo to another, and likewise the standard RPM dependencies are copied into the target standard RPM repo.

So whereas Pulp 2 can only do this:

(recursive module copy)
[rpms + modules] ----> modules + rpms -----> [rpms + modules]

Pulp 3 may need to do this, to properly support user cases:

(in one recursive module copy operation)
[modules] ----> modules -----> [modules]
[rpms] ------> rpms -----> [rpms]

(At least, one would hope we wouldn't need to implement this until Pulp 3. The timelines remain fuzzy.)

It remains an open question whether we can assume at most 2 source and 2 target repos, with matching pairs, or whether it needs to be even more general.

The amount of logic required here is another reason why we should break it off into a separate package to use as a toolkit.

#5 - 02/18/2019 08:01 PM - bherring

- Related to Test #4364: Implement modularity content dependency solving added

#6 - 04/26/2019 10:33 PM - bmbouter

- Tags deleted (Pulp 3)

#7 - 05/02/2019 06:35 PM - dalley

- Parent task set to #4762

#8 - 05/02/2019 09:33 PM - dalley

- Subject changed from Implement modularity content dependency solving to Implement modular RPM dependency solving

#9 - 05/25/2019 04:38 AM - dalley

- Subject changed from Implement modular RPM dependency solving to As a user, I have RPM dependency solving when copying modules

- Description updated

#10 - 01/17/2020 05:55 PM - dalley

- Parent task changed from #4762 to #6017

#11 - 02/26/2020 06:56 PM - dalley

- Status changed from NEW to ASSIGNED

- Assignee set to dalley

#12 - 02/26/2020 07:04 PM - dalley

- Status changed from ASSIGNED to MODIFIED

- % Done changed from 0 to 100

Applied in changeset 5d4b4f0e88f92115f8e81e61d6ef76aad20a7c16.

#13 - 02/27/2020 03:31 AM - dalley

- Subject changed from As a user, I have RPM dependency solving when copying modules to As a user, I have dependency solving when copying modules

- Description updated

#14 - 04/23/2020 10:04 PM - dalley

- Sprint/Milestone set to Pulp RPM 3.3.0

#15 - 04/23/2020 10:10 PM - dalley

- Status changed from MODIFIED to CLOSED - CURRENTRELEASE