RPM Support - Issue #4962

Modules are incorrectly copied when their artifacts shadow the names of non-modular RPM dependencies

06/13/2019 08:12 PM - dalley

Status:	NEW	**Start date:**	
Priority:	Normal	**Due date:**	
Assignee:		**Estimated time:**	0:00 hour
Category:			
Sprint/Milestone:			
Severity:	3. High	**Sprint Candidate:**	No
Version:		**Tags:**	Pulp 2
Platform Release:		**QA Contact:**	bherring
Blocks Release:		**Complexity:**	
OS:		**Smash Test:**	
Backwards Incompatible:	No	**Verified:**	No
Triaged:	Yes	**Verification Required:**	Yes
Groomed:	No	**Sprint:**	

Description

When Pulp tries to copy a module like openmpi or mariadb from the Fedora modular repository (https://dl.fedoraproject.org/pub/fedora/linux/updates/30/Modular/x86_64/), using either the recursive or recursive-conservative depsolving strategies, it will additionally (and incorrectly) copy modules like perl, perl-bootstrap, and setools.

This happens because one of the artifacts in these modules (modules: openmpi and mariadb) depends on the perl package, which is meant to be provided by the normal Fedora repositories. However in this case there also exists a "perl" module, and libsolv tries to use artifacts from the "perl" module to satisfy the dependency. This should not happen because the openmpi and mariadb modules do not not declare a modular dependency in the "dependencies" section of their modulemd documents.

Dependency solving uses a libsolv "pool" that contains the entire contents of the source repository, and in conservative mode, also the entire contents of the target repository. We need to change the behavior of the depsolver such that such that all non-modular RPMs in the pool are "considered" during depsolving, but only the modular RPMs that are part of the module being copied or an explicitly listed modular dependency are "considered", and all others are excluded from depsolving.

I'm unsure if a mechanism to do hide certain units from the depsolver is exposed to us, see: https://github.com/openSUSE/libsolv/issues/258.

If not, it may be necessary to hack around the problem by being more selective about what module / modular artifact solvables we load into the pool to begin with.

```
---
document: modulemd
version: 2
data:
  name: openmpi
  stream: 4.0
  version: 3020190331170644
  context: a5b0195c
  arch: x86_64
  summary: Open Message Passing Interface
  description: >-
    Open MPI is an open source, freely available implementation of both the MPI-1
    and MPI-2 standards, combining technologies and resources from several other projects
    (FT-MPI, LA-MPI, LAM/MPI, and PACX-MPI) in order to build the best MPI library
    available.  A completely new MPI-2 compliant implementation, Open MPI offers advantages
    for system and software vendors, application developers, and computer science
    researchers. For more information, see http://www.open-mpi.org/ .
  license:
    module:
    - MIT
    content:
    - BSD and MIT and Romio
  dependencies:
  - buildrequires:
      platform: [f30]
    requires:
      platform: [f30]
  references:
    community: https://www.open-mpi.org/
    documentation: https://www.open-mpi.org/doc/v4.0/
    tracker: https://github.com/open-mpi/ompi/issues
  profiles:
    buildroot:
      rpms:
      - openmpi-devel
    default:
      rpms:
      - openmpi
  api:
    rpms:
    - openmpi
  components:
    rpms:
      openmpi:
        rationale: The core package
        ref: 4.0
        buildorder: 10
        arches: [aarch64, armv7hl, i686, ppc64le, s390x, x86_64]
  artifacts:
    rpms:
    - openmpi-0:4.0.1-1.module_f30+3806+d6645847.src
    - openmpi-0:4.0.1-1.module_f30+3806+d6645847.x86_64
    - openmpi-debuginfo-0:4.0.1-1.module_f30+3806+d6645847.x86_64
    - openmpi-debugsource-0:4.0.1-1.module_f30+3806+d6645847.x86_64
    - openmpi-devel-0:4.0.1-1.module_f30+3806+d6645847.x86_64
    - openmpi-devel-debuginfo-0:4.0.1-1.module_f30+3806+d6645847.x86_64
    - openmpi-java-0:4.0.1-1.module_f30+3806+d6645847.x86_64
    - openmpi-java-devel-0:4.0.1-1.module_f30+3806+d6645847.x86_64
    - python2-openmpi-0:4.0.1-1.module_f30+3806+d6645847.x86_64
    - python3-openmpi-0:4.0.1-1.module_f30+3806+d6645847.x86_64
...
```

See: no "dependencies" listed beyond the standard "platform"

The openmpi-devel package is the one that has a requires on /usr/bin/perl

Related issues:

Related to RPM Support - Issue #4693: Module Streams not copying correctly wi...	**NEW**
Blocked by Packaging - Task #5151: Apply new patches to libsolv, and possibly...	**CLOSED - CURRENTRELEASE**
Copied to RPM Support - Test #4995: Modules are incorrectly copied when their...	**CLOSED - COMPLETE**

History

#1 - 06/13/2019 08:15 PM - dalley

- Related to Issue #4693: Module Streams not copying correctly with recursive and recursive_conservative added

#2 - 06/13/2019 08:16 PM - dalley

- Description updated

#3 - 06/13/2019 08:42 PM - bherring

- QA Contact set to bherring

- Verification Required changed from No to Yes

#4 - 06/13/2019 09:10 PM - dalley

- Description updated

#5 - 06/13/2019 09:16 PM - dalley

- Description updated

#6 - 06/14/2019 09:33 PM - dalley

Part 1 of fix

https://github.com/pulp/pulp_rpm/pull/1375

#7 - 06/17/2019 04:19 PM - dalley

- Status changed from ASSIGNED to POST

PR: https://github.com/pulp/pulp_rpm/pull/1377

Still a work-in-progress

#8 - 06/19/2019 09:14 PM - bherring

- Copied to Test #4995: Modules are incorrectly copied when their artifacts shadow the names of non-modular RPM dependencies added

#9 - 06/21/2019 03:30 PM - ttereshc

- Sprint changed from Sprint 54 to Sprint 55

#10 - 06/24/2019 11:29 PM - rchan

Adding bz link.

#11 - 06/25/2019 01:18 PM - ttereshc

- Sprint/Milestone set to 2.20.0

#12 - 06/26/2019 06:04 PM - ttereshc

- Sprint/Milestone deleted (2.20.0)

- Platform Release deleted (2.20.0)

#13 - 06/27/2019 06:03 PM - dalley

Below is a conversation I had with Igor Gnatenko where he mentioned that the approach described above is flawed and suggested an alternative approach. That is what I will be implementing.

```
<ignatenkobrain> dalley: yes, you need to know about defaults
<ignatenkobrain> so when you are resolving deps, you need to take into account:
<ignatenkobrain> * non-modular content
<ignatenkobrain> * default module streams (even if they have lower version of non-modular content, they should
 be prefered)
<ignatenkobrain> * all combinations of you modular dependencies
<ignatenkobrain> dalley: so I think you do it wrong when you enable / disable some content. You should enable
everything by default and then set up libsolv dependencies in a proper way
<ignatenkobrain> and then handle some special cases
```

```
<dalley> ignatenkobrain, how do we do that? we have to load "perl-4:5.28.2-436.module_f30+4677+de2e7282.x86_64" (or whatever) into the depsolver, how do we tell it "this is a modular RPM, don't try to use it to satisfy the dependency of "mariadb-3:10.4.2-1.beta.module_f30+2924+d4dbc42b.x86_64"
<ignatenkobrain> dalley: from which module is that perl coming?
<ignatenkobrain> is it default stream of perl module?
<ignatenkobrain> if not, does mariadb module depend on it?
<dalley> mariadb doesn't depend on the perl module at all, but it does depend on perl, and "perl-4:5.28.2-436.module_f30+4677+de2e7282.x86_64" satisfies that so the depsolver tries to use it to satisfy the dep
<dalley> if it didn't exist at all, it would come back and say "we have a problem, perl is missing" and so we would fake it
<dalley> because DNF may know about all these extra repos, but Pulp doesn't
<ignatenkobrain> dalley: well, that's the thing... perl's module stream is not default
<ignatenkobrain> so it is in conflict
<ignatenkobrain> let me find some relevant code in fus
<ignatenkobrain> dalley: https://github.com/fedora-modularity/fus/blob/master/fus.c#L1290-L1318
<ignatenkobrain> dalley: so for default modules you need to add Provides: module-default() and then you can easily query default and non-default modules.
<ignatenkobrain> then you need to disable irrelevant non-default modules
<ignatenkobrain> which would make some constraints on packages from those modules to not be installable
<ignatenkobrain> and that will pull non-modular perl package :0
<ignatenkobrain> because that modular perl requires module(perl:5.26) or whatever, so if all providers of that capability are disabled, it won't be chosen to satisfy dependency
```

PR where future fixes will land is here: https://github.com/pulp/pulp_rpm/pull/1383

#14 - 07/12/2019 03:28 PM - dkliban@redhat.com

- Sprint changed from Sprint 55 to Sprint 56

#15 - 07/17/2019 09:41 PM - dalley

- Related to Task #5151: Apply new patches to libsolv, and possibly upgrade libsolv version added

#16 - 07/17/2019 09:42 PM - dalley

- Related to deleted (Task #5151: Apply new patches to libsolv, and possibly upgrade libsolv version)

#17 - 07/17/2019 09:42 PM - dalley

- Blocked by Task #5151: Apply new patches to libsolv, and possibly upgrade libsolv version added

#18 - 07/17/2019 09:43 PM - dalley

- Status changed from POST to ASSIGNED

#19 - 07/17/2019 11:02 PM - dalley

- Status changed from ASSIGNED to POST

https://github.com/pulp/pulp_rpm/pull/1408

#20 - 07/23/2019 03:59 AM - dalley

- Parent task set to #5063

#21 - 08/20/2019 04:42 PM - dalley

- Platform Release set to 2.21.0

#22 - 08/22/2019 02:51 PM - rchan

- Sprint changed from Sprint 56 to Sprint 57

#23 - 08/26/2019 04:32 PM - rchan

- Sprint changed from Sprint 57 to Sprint 58

#24 - 09/10/2019 08:24 PM - dalley

- Platform Release deleted (2.21.0)

#25 - 09/13/2019 04:00 PM - dalley

- Status changed from POST to NEW

- Assignee deleted (dalley)

On the backburner

#26 - 09/17/2019 09:02 PM - rchan

- Sprint deleted (Sprint 58)

#27 - 09/25/2019 10:53 PM - dalley

- Parent task deleted (#5063)