

## Katello - Bug #4826

### Investigate if we can speed up reposit enablement in the Red Hat repositories page

03/24/2014 11:46 PM - Mike McCune

<b>Status:</b> Closed	
<b>Priority:</b> High	
<b>Assignee:</b> Ivan Necas	
<b>Category:</b> Web UI	
<b>Target version:</b> Katello 2.0	
<b>Difficulty:</b> medium	<b>Fixed in Releases:</b>
<b>Triaged:</b> Yes	<b>Found in Releases:</b>
<b>Bugzilla link:</b>	<b>Red Hat JIRA:</b>
<b>Pull request:</b>	
<b>Description</b>	
0) Make sure you have a manifest imported	
1) Navigate to the Content -> Red Hat Repositories page	
2) If you select a group of repositories, say:	
<ul style="list-style-type: none"><li>• Red Hat Enterprise Linux 6 Server (RPMs)</li></ul>	
it can take upwards of 1-2 minutes for the subset of only around 10-15 repos to show up	
Would like to see if we can spend some time looking at the code that detects the set of repos underneath that list to see if we can speed it up.	

#### Associated revisions

##### Revision 2c27b9a7 - 04/15/2014 12:36 PM - Ivan Necas

Refs #4826 - Update cli to support new reposit approach

##### Revision f0581b72 - 04/24/2014 09:23 AM - Ivan Necas

Fixes #4826 - rework reposit sets to not create repositories on repo set enable

The previous way of reposit enable was creating disabled repositories when the reposit was enabled. These repositories were not usable till enabled and it takes quite long to have them created, provided that most of the disabled repositories will not be used anyway.

It also meant we needed to keep the repos in sync with the CDN, once the reposit was enabled.

With the new approach, there is no notion of enabled/disabled repository set. Instead, one can get the list of available repositories for the repository set and enable/disable subset of the available repositories. Enable available repository means creation of the repository, disable means deletion. Therefore, no need for having special flag tracking the enablement status on the repository itself.

##### Revision e972c028 - 04/24/2014 09:23 AM - Ivan Necas

Fixes #4826 - fix rubocop

##### Revision c1dae0c7 - 04/24/2014 09:23 AM - Ivan Necas

Fixes #4826 - address review comments

##### Revision fc4c745f - 04/25/2014 01:11 PM - Ivan Necas

Merge pull request #3980 from iNecas/reposets-rework

Fixes #4826 - rework reposets to not create repositories on repo set enable

## Revision 8814106e - 04/25/2014 01:11 PM - Ivan Necas

Merge pull request #143 from iNecas/reposet-enable

Refs #4826 - Update cli to support new reposet approach

## History

---

### #1 - 03/24/2014 11:47 PM - Mike McCune

- Description updated

### #2 - 03/25/2014 02:35 PM - Ivan Necas

I'm working on the Dynflowization of repo set enablement: the parallel execution of repo creations might probably help + we can try to identify the slow bits and address that ad-hoc.

### #3 - 03/25/2014 02:37 PM - Ivan Necas

- Assignee changed from Justin Sherrill to Ivan Necas

### #4 - 03/25/2014 05:11 PM - Ivan Necas

Dynflowized repo enable for reposet enable with 12 repos took 12 seconds against mirrored cdn, 25s against real one.

There are two slow things here:

1. loading the listing files from cdn (don't know yet how long it takes, will have more data tomorrow)
2. creating the repositories: one single repo is created in about 2 seconds. However, when we run the creating of 12 repos in parallel, the last one takes about 8 seconds to create because waiting in a queue for a while for repo publish.

With the second step, we could:

- a. skip the repo publishing and postpone it for syncing, as without syncing, the repo is not that much useful.
- b. increase the pulp concurrency options to be able to run more distribution publish in parallel

The a. option could also help with <http://projects.theforeman.org/issues/4724>

### #5 - 03/26/2014 02:59 PM - Ivan Necas

I've extracted the reposet enable code into more dynflow actions to have more numbers in hand:

1. load initial data to be able to start scanning: 2s
2. the scanning of cnd itself: 3.5s
3. create 12 repos locally in db: 16 s: majority of the time, it was the repository.save itself
4. create 12 repos in pulp in parallel (7 s total, in scale of 3-7s: most of the time it was the distributor publish stuff)

This gives us around 30s for repo set enable.

This is already boost comparing the old style, where the repo creation (both local db and pulp orchestration) was run in sequence, which gives us more than 60s for the task.

Possible optimization are:

1. run even the local repo creation in parallel (would give us around - 10s)
2. disable the metadata generation after repo is created (as we don't need that until some content appears there anyway) (would give us around -8s)
3. postpone the repo creation to the repo-enable phase (would give us -25s) it would be just about loading the listing files from cdn, which takes around 5 seconds. Later optimization could be

My favorite is the option 3. and now is good time to do this change, we need to add the support for repo-enable in new cli so we will touch the code anyway.

### #6 - 03/26/2014 05:08 PM - Ivan Necas

Having just the cdn scan for reposet enable checkbox takes us to something around 8 seconds, which is not that bad.

### #7 - 04/17/2014 12:24 PM - Ivan Necas

- Status changed from Assigned to Ready For Testing

### #8 - 04/25/2014 01:52 PM - Ivan Necas

- Status changed from Ready For Testing to Closed

- % Done changed from 0 to 100

Applied in changeset katello|commit:f0581b72a63bd9c4aca36f62c59523097fa338b9.

**#9 - 08/21/2014 02:37 PM - Eric Helms**

- translation missing: en.field\_release set to 13