

## Mercurial repositories display revision ID instead of changeset ID

2022/05/09 14:10 - Admin Redmine

ステータス:	Closed	開始日:	2009/08/10
優先度:	低め	期日:	
担当者:		進捗率:	0%
カテゴリ:	SCM_3	予定工数:	0.00時間
対象バージョン:	1.2.0_27	作業時間:	0.00時間
Redmineorg_URL:	https://www.redmine.org/issues/3724	status_id:	5
category_id:	3	tracker_id:	1
version_id:	27	plus1:	0
issue_org_id:	3724	affected_version:	
author_id:	7309	closed_on:	
assigned_to_id:	11192	affected_version_id:	
comments:	25		

**説明**

Redmine exposes Mercurial's revision IDs (one-up decimal number, similar to Subversion) as the primary means of referring to changesets, rather than the changeset ID (strings of hex digits, like Git). While this does work, it's explicitly discouraged in Mercurial's documentation.

From the "FAQ":

[http://mercurial.selenic.com/wiki/FAQ#FAQ.2BAC8-Terminology.What\\_are\\_revision\\_numbers.2C\\_changeset\\_IDs.2C\\_and\\_tags.3F](http://mercurial.selenic.com/wiki/FAQ#FAQ.2BAC8-Terminology.What_are_revision_numbers.2C_changeset_IDs.2C_and_tags.3F):

You should always use some form of changeset ID rather than the local revision number when discussing revisions with other Mercurial users as they may have different revision numbering on their system.

And the page on "revision numbers": <http://mercurial.selenic.com/wiki/RevisionNumber> says:

Pitfalls

Revision numbers referring to changesets are +very likely+ to be different in another copy of a repository. +Do not use them+ to talk about changesets with other people. Use the changeset ID instead.

(Emphasis theirs.)

The behavior of hgweb.cgi (the web frontend that comes with Mercurial) is to expose only the short-form changeset IDs. However it allows entering a revision ID into the URL as well. So the following URLs will bring up the same patch:

- <http://schismtracker.org/hg/rev/ff80faee224f>
- <http://schismtracker.org/hg/rev/1846>
- <http://schismtracker.org/hg/rev/20090805> (as hgweb allows entering tags as well -- I dunno if you want to get that elaborate :)

This is probably the sanest path to take with Redmine, in terms of keeping old URLs functioning and also to more or less parallel the way that Git repos are managed.

Issue #3567 might be tangentially related to this.

journals

Yeah this is a problem for us too. It's very fragile and something as simple as removing a changeset with screw up ALL the links. This really needs to be fixed.

The attached patch addresses your concern, and I'm all for moving away from the ambiguous revision numbers, but this sort of behavioural change probably requires a configuration option, since it will definitely break any existing r##-style references... comments?

Actually, there are some further changes to be made in  
app/models/repository/mercurial.rb

Peter Fern wrote:

The attached patch addresses your concern, and I'm all for moving away from the ambiguous revision numbers, but this sort of behavioural change probably requires a configuration option, since it will definitely break any existing r##-style references... comments?

Probably a conversion script of the r##-style references to hashes would be best. Thanks for doing this!

Just an update - I have a significant Mercurial patch near completion that addresses this issue, plus a number of others, please ignore my previous attachment.

Awesome. Look forward to it.

I don't see why it'd break r##, though - like I mentioned above, hg understands several kinds of references to changesets, so what's preventing redmine from allowing all three and just passing along whatever identifier to the client?

There are two keys in the Redmine database - revision and scmid, however Mercurial has 3 identifiers: revision, short changeset ID, and full changeset ID. Since revision is likely to get clobbered, and even the short changeset ID is not guaranteed to remain unique, I'd like to move to using the short ID as the revision, and the full ID as the scmid, since this is the only way we'll have a guaranteed unique identifier in the DB, and if the revision number gets routinely clobbered by merges, it's pretty useless anyway, since any references to it will now refer to some other commit.

Peter Fern wrote:

Just an update - I have a significant Mercurial patch near completion that addresses this issue, plus a number of others, please ignore my previous attachment.

If you're working on a big Mercurial update, you may want to take a look at issue r3421. Mercurial is more or less useless without it in Redmine.

Luke Hoersten wrote:

Peter Fern wrote:

Just an update - I have a significant Mercurial patch near completion that addresses this issue, plus a number of others, please ignore my previous attachment.

If you're working on a big Mercurial update, you may want to take a look at issue r3421. Mercurial is more or less useless without it in Redmine.

Sorry that's issue #3421.

I wouldn't say that - the only thing that doesn't work is file sizes... my patch incorporates branch and tag support like git, plus major performance improvements. I'll have it up in the next day or so. I've added a comment to #3421

Peter Fern wrote:

I wouldn't say that - the only thing that doesn't work is file sizes... my patch incorporates branch and tag support like git, plus major performance improvements. I'll have it up in the next day or so. I've added a comment to #3421

You understand what #3421 is fixing right? Redmine pulls the history from the repo but file access is attempted on the file system instead of using 'hg cat' to grab what's in the changeset specified.

I think you'll find that's incorrect, and looking at the patch, it doesn't modify the 'cat' method in any way other than to add a default identifier of 'tip'. The only file access the Mercurial code is to get file sizes, which I'd be happy to see resolved since it's the last functional hole in my patchset.

As a user of the Mercurial plugin, I want to add my \$.02 here.

Just as there is a reason for the globally-unique changeset hash id, there is also a reason for the integer revision id as well. I currently rely on Redmine as the source for access to the integer id, and removing it completely would break my build system implementation.

While a changeset hash is useful when discussing revisions between developers, it is unfortunately both completely unsuitable for use in build numbering of packages as well as being hostile to the human eye. Because of the necessity of numbering our builds with the revision number, in order to trace back any build package to the source code which created it, we simply make sure that our developers know of the mutability of the integer revision id and use the build repository as its authoritative source. This is simply the added cost of using a system like Mercurial, that you have to keep in mind something extra like the scope of the revision id.

It's not simply a matter of saying "suck it up, use the hash", either. When building .NET assemblies, the revision number must fit within a 16-bit integer. A hash can't possibly do this. Even if it were possible, however, I still wouldn't inflict hash numbers on our customers as a means of identifying a package.

So, whatever change you make, I would ask that both it preserve the integer revision id, as well as allowing r-style links in Redmine and being available in both the Repository view as well as the atom feed for the repository. Otherwise, Redmine's mercurial functionality will be significantly less useful to developers like me.

From the "Mercurial book":<http://hgbook.red-bean.com/read/a-tour-of-mercurial-the-basics.html> :

[...] the same number in two different clones of a repository may identify different changesets.

What that means is if you swap out the repo for a modified clone, potentially all your short numbers which you've "hardcoded" into the Redmine issues/comments would be invalidated.

I want my Redmine comments to refer to a specific changeset, not a specific clone's changeset. Perhaps if you're using Mercurial in a more centralized manor this is fine, but my team doesn't work that way.

Ted Lilley wrote:

As a user of the Mercurial plugin, I want to add my \$.02 here.

Just as there is a reason for the globally-unique changeset hash id, there is also a reason for the integer revision id as well. I currently rely on Redmine as the source for access to the integer id, and removing it completely would break my build system implementation.

While a changeset hash is useful when discussing revisions between developers, it is unfortunately both completely unsuitable for use in build numbering of packages as well as being hostile to the human eye. Because of the necessity of numbering our builds with the revision number, in order to trace back any build package to the source code which created it, we simply make sure that our developers know of the mutability of the integer revision id and use the build repository as its authoritative source. This is simply the added cost of using a system like Mercurial, that you have to keep in mind something extra like the scope of the revision id.

It's not simply a matter of saying "suck it up, use the hash", either. When building .NET assemblies, the revision number must fit within a 16-bit integer. A hash can't possibly do this. Even if it were possible, however, I still wouldn't inflict hash numbers on our customers as a means of identifying a package.

So, whatever change you make, I would ask that both it preserve the integer revision id, as well as allowing r-style links in Redmine and being available in both the Repository view as well as the atom feed for the repository. Otherwise, Redmine's mercurial functionality will be significantly less useful to developers like me.

---

Please see #4455 - Mercurial overhaul.

---

Luke Hoersten wrote:

From the "Mercurial book": <http://hgbook.red-bean.com/read/a-tour-of-mercurial-the-basics.html> :

[...] the same number in two different clones of a repository may identify different changesets.

What that means is if you swap out the repo for a modified clone, potentially all your short numbers which you've "hardcoded" into the Redmine issues/comments would be invalidated.

This is a very serious problem. But there is no way to resolve this problem simply. Because scmid of changeset table is not unique and revision is unique.

---

Luke Hoersten wrote:

What that means is if you swap out the repo for a modified clone, potentially all your short numbers which you've "hardcoded" into the Redmine issues/comments would be invalidated.

h2. 1

```
  /- B
A-----C

DB      hg
0 A     0 A
1 B     1 B
2 C     2 C
```

h2. 2

```
A--C

DB      hg
0 A     0 A
1 B     1 C
2 C
```

h2. 3

```
 /----D
A--C

DB      hg
0 A     0 A
1 B     1 C
2 C     2 D
```

h2. 4

```
 /-----B
 /----D
```

A---C

DB		hg	
0	A	0	A
1	B	1	C
2	C	2	D
3	B	3	B

---

I finish testing of last big feature of #4455.

This feature is "old rNN record on DB and wikis can not delete".

<http://github.com/marutosi/redmine/tree/hg-overhaul-0.9>

<http://github.com/marutosi/redmine/commit/30845a8c8ed51a71beb6e1d4df4dfbcf3a55c444>

---

I really hope, Mercurial will at least keep the option to use the revision number. That is in fact one of the big advantages compared to git, I am aware of the possibility to have same repo with different orders of changesets, but that doesn't matter, if you define server/buildsystem repo as master repo.

Please keep at least revision number for Mercurial as option.

## I also changed hgweb to show revision numbers instead short hashes.

Thank you for your feedback.

Yuya's #4455 new implementation keeps redmine original DB structure.

Please see <http://www.redmine.org/issues/4455#note-78> .

---

To Toshi: JFTR, the arguments of @assert\_equal@ are @expected, actual@

r4654 "trunk/test/unit/repository\_mercurial\_test.rb":

[http://www.redmine.org/projects/redmine/repository/revisions/4654/diff/trunk/test/unit/repository\\_mercurial\\_test.rb](http://www.redmine.org/projects/redmine/repository/revisions/4654/diff/trunk/test/unit/repository_mercurial_test.rb)

```
assert_equal @repository.find_changeset_by_name(r).revision, '2'
```

should be

```
assert_equal '2', @repository.find_changeset_by_name(r).revision
```

---

Yuya Nishihara wrote:

To Toshi: JFTR, the arguments of @assert\_equal@ are @expected, actual@

r4654 "trunk/test/unit/repository\_mercurial\_test.rb":

[http://www.redmine.org/projects/redmine/repository/revisions/4654/diff/trunk/test/unit/repository\\_mercurial\\_test.rb](http://www.redmine.org/projects/redmine/repository/revisions/4654/diff/trunk/test/unit/repository_mercurial_test.rb)

[...]

should be

[...]

I fixed and committed. Thanks.

---

I think there are two cases of revision number inconsistency.

- History editing (hg strip, hg rebase etc.)
- Replacing repository

If you did history editing, you need to delete repository from redmine and re-add it.

There are some feature issues of re-syncing #1273.

About replacing repository, I wrote at #6681 description.

I think Mercurial user should not use "rNN" in wiki.  
If someone wrote "rNN" in wiki, it is user's own risk.  
#6681 changed auto close issue text from "rNN" to "commit:XXX".

Redmine stores revision number and changeset id in database.  
To resolve this issue, we changed as following.

- Change revision text to Mercurial style "2:400bb8672109"
- Change revision link from revision number to changeset id
- Redmine URL accepts both of revision number and changeset id
- diff, cat and annotate accepts both of revision number and changeset id

h2. Before

!before-repository.png!  
!before-link.png!  
!before-annotate.png!

h2. After

!after-repository.png!  
!after-link-id.png!  
!after-link-revno.png!  
!after-annotate.png!

---

related\_issues

relates,Closed,6092,Truncate Git revision labels in Activity page/feed and allow configurable length  
relates,Closed,4924,mercurial reader disregard information from non-default branch  
relates,Closed,6681,Mercurial, Bazaar and Darcs auto close issue text should be commit id instead of revision number  
relates,Closed,4455,Mercurial overhaul  
relates,Closed,7518,Mercurial diff can be wrong if the previous changeset isn't the parent  
relates,New,8030,Bazaar integration doesn't notice new commits to repository  
relates,Closed,5501,Git: Mercurial: Adding visual merge/branch history to repository view

---

## 履歴

#1 - 2022/05/10 17:25 - Admin Redmine

- カテゴリ を SCM\_3 にセット

- 対象バージョン を 1.2.0\_27 にセット