

Redmine - Patch #34479

Fix possible race condition with parallel, identical file uploads

2020-12-21 03:33 - Jens Krämer

|   |             |                 |           |
|---|-------------|-----------------|-----------|
| Status:   | Closed      | Start date:     |           |
| Priority:   | Normal      | Due date:       |           |
| Assignee:   | Go MAEDA    | % Done:         | 0%        |
| Category:   | Attachments | Estimated time: | 0.00 hour |
| Target version:   | 4.1.2       |                 |           |
| <p><b>Description</b></p> <p>There is a race condition with multiple application processes when uploading multiple identical files at the same time, which may lead to attachments pointing to files that do not exist, if attachment creation and deduplication are executed in a particular order:</p> <ul style="list-style-type: none"><li>- Attachment A0 exists in the system.</li><li>- identical Attachment A1 is created in one process</li><li>- identical Attachment A2 is created at the same time in another application process</li><li>- for some reason, A2 is deduplicated before A1 and now points to A1's diskfile</li><li>- A1 is deduplicated and now points to A0's diskfile</li></ul> <p>Since during deduplication of A1, A1's original disk file is deleted, A2 now points to a non existent file.<br/>To fix that, an additional check whether the file really is not referenced anymore before deleting it is introduced.</p> <p>In plain Redmine usage scenarios this is really a very unlikely scenario, but we had it happen with our WebDAV integration where whole directory trees can be uploaded at once. Should the issue indeed have manifested itself somewhere else, it can easily be fixed by finding unreadable attachment records, using the digest to find an identical, readable attachment and correcting the disk location accordingly:</p> <pre>unreadables = Attachment.all.reject(&amp;:readable?) unreadables.each do  u    if readable = Attachment.where(filesize: u.filesize, digest: u.digest).detect(&amp;:readable?)     u.update_columns disk_directory: readable.disk_directory, disk_filename: readable.disk_filename   end end</pre> |             |                 |           |

Associated revisions

Revision 20812 - 2021-03-17 14:46 - Go MAEDA

Introduces an additional check before removing a deduplicated file (#34479).

Patch by Jens Krämer.

Revision 20813 - 2021-03-17 14:53 - Go MAEDA

Merged r20812 from trunk to 4.1-stable (#34479).

History

#1 - 2021-01-01 08:08 - Go MAEDA

- Target version set to Candidate for next major release

#2 - 2021-02-08 19:58 - Andriy Lesyuk

We had a situation, when several files with the same name were uploaded at the same time. All they got the same disk filename. As a result different container objects now point to the same file, which should actually belong to only one of them. Other files got lost (overridden).

#3 - 2021-02-09 09:16 - Go MAEDA

- Target version changed from Candidate for next major release to 4.1.2

Setting the target version to 4.1.2.

**#4 - 2021-03-15 23:55 - Marius BĂLTEANU**

All tests pass with the patch applied on the current trunk: <https://gitlab.com/redmine-org/redmine/-/pipelines/270891057>

**#5 - 2021-03-17 14:54 - Go MAEDA**

- *Status changed from New to Closed*
- *Assignee set to Go MAEDA*

Committed the patch. Thank you.

**#6 - 2021-07-07 07:27 - Jens Krämer**

Andriy Lesyuk wrote:

We had a situation, when several files with the same name were uploaded at the same time. All they got the same disk filename. As a result different container objects now point to the same file, which should actually belong to only one of them. Other files got lost (overridden).

see [#35539](#) for a patch fixing this.

**Files**

|   |         |            |             |
|---|---------|------------|-------------|
| 0001-introduces-an-additional-check-before-removing-a-ded.patch | 1.93 KB | 2020-12-21 | Jens Krämer |
|---|---------|------------|-------------|