Redmine - Patch #25215
Re-use existing identical disk files for new attachments

2017-02-28 04:11 - Jens Krämer

Status:	Closed	**Start date:**	
Priority:	Normal	**Due date:**	
Assignee:	Jean-Philippe Lang	**% Done:**	0%
Category:	Attachments	**Estimated time:**	0.00 hour
Target version:	3.4.0		

Description

Uploaded files can already be associated with multiple attachment records through Attachment#copy. This patch adds an after_create hook to change the disk_filename and disk_directory attributes of a newly created attachment to point to an already existing, identical (filesize and digest) diskfile if one exists.

Database locks are used to guard against the deletion of the older attachment while the reference of the new attachment is changed.

Usefulness of this feature (i.e. how much space is saved) will vary a lot depending on external circumstances of course (one large scale setup we maintain at Planio saved around 15% / 60GB). Since the possiblity to have 1:n relationships of disk files to attachments already exists it just seems logical to make use of it for new attachments as well.

Related issues:

Related to Redmine - Feature # 19289: Exclude attachments from incoming email...	**New**
Related to Redmine - Feature # 23510: Reuse an exist attachment	**Closed**
Duplicated by Redmine - Feature # 15257: Attachment deduplication	**Closed**
Blocked by Redmine - Patch # 25240: Use SHA256 for attachment digest computation	**Closed**
Blocked by Redmine - Patch # 25590: prevent deadlocks in attachment deduplica...	**Closed**

Associated revisions

Revision 16458 - 2017-04-03 13:54 - Jean-Philippe Lang

Reuse existing identical disk files for new attachments (#25215).

Patch by Jens Kraemer.

Revision 16459 - 2017-04-03 13:55 - Jean-Philippe Lang

Adds file equality check to deduplication hook (#25215).

Patch by Jens Kraemer.

Revision 16460 - 2017-04-03 14:13 - Jean-Philippe Lang

Delete the file after the change is committed (#25215).

Revision 16462 - 2017-04-03 14:51 - Jean-Philippe Lang

Adds a test for when the file comparison fails (#25215).

History

#1 - 2017-02-28 04:26 - Jan from Planio www.plan.io

- Target version set to Candidate for next minor release

#2 - 2017-02-28 10:43 - Go MAEDA

- Duplicated by Feature #15257: Attachment deduplication added

#3 - 2017-02-28 10:45 - Go MAEDA

- Related to Feature #19289: Exclude attachments from incoming emails based on file content or file hash added

#4 - 2017-02-28 10:51 - Go MAEDA

- Related to Feature #23510: Reuse an exist attachment added

#5 - 2017-02-28 21:46 - Jean-Philippe Lang

- Status changed from New to Needs feedback

IMO, MD5 is too weak for this purpose and this could lead to potential vulnerabilities. The first that comes to my mind: attacker generates a malicious file and a legitimate file with the same MD5, he first uploads the malicious file then send the legitimate one to a user X who will eventually upload it => people downloading the later from user X will actually get the malicious file.
We should implement this after upgrading the digest to a safer hash function.
Please let me know what you think.

#6 - 2017-03-01 02:17 - Jens Krämer

Good point. Since we're also comparing file sizes, an attacker would have to create a collision while also matching the file sizes. Still not impossible I guess. Changing the hash function would make malicious collision creation harder, but still not impossible theoretically.

Maybe adding a real byte by byte comparison for the case of matching filesize / md5 would be the better way? Uploads take a while any way so the added computation time might not weigh in too much.

#7 - 2017-03-01 03:59 - Go MAEDA

What do you think about migrating to SHA-1? It is as fast as MD5 and much safer.

One problem is that migrating a existing Redmine instance may take a long time if it stores many large files.

#8 - 2017-03-01 11:08 - Jean-Philippe Lang

Jens Krämer wrote:

> *Good point. Since we're also comparing file sizes, an attacker would have to create a collision while also matching the file sizes. Still not impossible I guess.*

MD5 collission with the same input size can be created in seconds with a standard computer. The file size comparaison does not make it safer.

> *Changing the hash function would make malicious collision creation harder, but still not impossible theoretically.*

Not just harder, but practically impossible. Unlike MD5, there's no known way to easily generate SHA256 collisions.

> *Maybe adding a real byte by byte comparison for the case of matching filesize / md5 would be the better way? Uploads take a while any way so the added computation time might not weigh in too much.*

That would be the safer option indeed. It also guarantees that the original file is not broken/missing, which is a good thing IMO before discarding the uploaded file. Replacing the MD5 with a safer hash function does make sense anyway.

#9 - 2017-03-01 11:12 - Jean-Philippe Lang

Go MAEDA wrote:

> *What do you think about migrating to SHA-1? It is as fast as MD5 and much safer.*

SHA-1? No. Maybe SHA256 or SHA512 instead.

> *One problem is that migrating a existing Redmine instance may take a long time if it stores many large files.*

Yes, this should not be done during a db migration for this reason. We can use a new hash function for new files and provide a task that updates the existing hashes.

#10 - 2017-03-02 04:30 - Jens Krämer

I gave the upgrade to SHA256 a try in #25240. Independent of that I'll add the bytewise comparison to this feature here next.

#11 - 2017-03-02 04:47 - Go MAEDA

- Blocked by Patch #25240: Use SHA256 for attachment digest computation added

#12 - 2017-03-02 08:36 - Jens Krämer

- File 0002-adds-file-equality-check-to-deduplication-hook.patch added

Additional patch adding a FileUtils.identical? check to compare actual file contents before removing the duplicate.

#13 - 2017-03-03 21:42 - Jean-Philippe Lang

- Target version changed from Candidate for next minor release to 3.4.0

#14 - 2017-03-08 11:57 - Hugo García

OK.

#15 - 2017-04-03 14:20 - Jean-Philippe Lang

- Status changed from Needs feedback to Closed

- Assignee set to Jean-Philippe Lang

Patches are committed, thanks. I think it's safer to delete the duplicate file after the change is committed to the DB (r16460).

#16 - 2017-04-12 08:54 - Go MAEDA

- Related to Patch #25590: prevent deadlocks in attachment deduplication added

#17 - 2017-04-12 09:53 - Go MAEDA

- Related to deleted (Patch #25590: prevent deadlocks in attachment deduplication)

#18 - 2017-04-12 09:54 - Go MAEDA

- Blocked by Patch #25590: prevent deadlocks in attachment deduplication added

#19 - 2017-04-12 09:56 - Go MAEDA

- Status changed from Closed to Reopened

A fix for this patch was submitted as #25590.

#20 - 2017-04-13 14:10 - Jean-Philippe Lang

- Status changed from Reopened to Closed

Fix committed.

#21 - 2017-06-25 14:11 - Jean-Philippe Lang

- Subject changed from re-use existing identical disk files for new attachments to Re-use existing identical disk files for new attachments

Files

0001-reuse-existing-identical-disk-files-for-new-attachme.patch	2.89 KB	2017-02-28	Jens Krämer
0002-adds-file-equality-check-to-deduplication-hook.patch	1.24 KB	2017-03-02	Jens Krämer