# Redmine - Feature #35889

## Textile and Markdown attachment rendering should support third-party formatters

2021-09-18 13:02 - Mischa The Evil

| | | | | |
|---|---|---|---|---|
| **Status:** | New | | **Start date:** | |
| **Priority:** | Normal | | **Due date:** | |
| **Assignee:** | | | **% Done:** | 0% |
| **Category:** | Attachments | | **Estimated time:** | 0.00 hour |
| **Target version:** | | | | |
| **Resolution:** | | | | |

### Description

As it is implemented currently with [#29752](), Textile and Markdown attachment rendering is supported <u>only</u> for the core formatters for Textile and Markdown.

- Textile attachment rendering currently relies on the *bundled* (custom) RedCloth3 library.
- Markdown attachment rendering currently relies on the *optional* Redcarpet (or CommonMark, post [#32424]()) libraries, with a fall-back to Redmine::WikiFormatting::NullFormatter.

Given the info above, currently:

- Textile attachment rendering always works, always relying on RedCloth3 (since RedCloth3 is bundled);
- Markdown attachment rendering **only** works with one of the optional Markdown core formatters installed (relying on Redcarpet or CommonMark) and based on Setting.text_formatting.

Redmine should also provide support for (custom) third-party Textile and Markdown formatters (e.g. RedCloth4 [via a formatter like provided by e.g. [redmine_redcloth](), kramdown, Maruku, RDiscount, etc.), based on what formatters (core and third-party) are actually available.
I think the easiest way to implement this is to let formatters provide a way to indicate that they provide support for either a Textile or Markdown flavor.

### Related issues:

| | | |
|---|---|---|
| Related to Redmine - Feature #29752: Render Textile and Markdown attachments ... | **Closed** | |
| Related to Redmine - Feature #15520: Markdown formatting | **Closed** | |
| Related to Redmine - Feature #32424: CommonMark Markdown Text Formatting | **Closed** | |
| Related to Redmine - Patch #2025: Pluggable wiki formatter | **Closed** | 2008-10-12 |

## History

#### #1 - 2021-09-18 13:03 - Mischa The Evil

*- Related to Feature #29752: Render Textile and Markdown attachments on the preview page added*

#### #2 - 2021-09-18 13:04 - Mischa The Evil

*- Related to Feature #15520: Markdown formatting added*

#### #3 - 2021-09-18 13:04 - Mischa The Evil

*- Related to Feature #32424: CommonMark Markdown Text Formatting added*

#### #4 - 2021-09-18 13:07 - Mischa The Evil

*- Related to Patch #2025: Pluggable wiki formatter added*

#### #5 - 2021-09-20 16:37 - Holger Just

Right of the hook, I think I'd rather prefer if we standardize on exactly one Markdown formatter. That way, we don't have to deal with the intricatities and potential (security) issues resulting from an overly generic integration. I'd also be fine with making this one markdown library a non-optional dependency.

With that being said, if you have a specific use-case in mind for having support for multiple optional markdown libraries, could you describe in more detail how this would be useful?

#### #6 - 2021-09-22 01:27 - Mischa The Evil

Holger Just wrote:

> Right of the hook, I think I'd rather prefer if we standardize on exactly one Markdown formatter. That way, we don't have to deal with the intricatities and potential (security) issues resulting from an overly generic integration.

I couldn't agree more in the context of what the Redmine core should provide.

Holger Just wrote:

> I'd also be fine with making this one markdown library a non-optional dependency.

I don't know if that is the best solution. For deployments that don't need the feature added with #29752 and don't use any Markdown formatters, it would mean that Redmine adds dependencies (thus adding to Redmine's attack surface) that will not be used.

Holger Just wrote:

> With that being said, if you have a specific use-case in mind for having support for multiple optional markdown libraries, could you describe in more detail how this would be useful?

Yes. I know of at least two Redmine user groups (± 100 deployments combined, all off-label usage [i.e. not (software) development]) that use a shared kramdown formatter for 10+ years. They all omit installing the Redcarpet gem required for the current core Markdown formatter, based on the assumption that not-installed software cannot act as an attack surface for their Redmine deployment.
Now, with this setup they are missing the functionality for Markdown attachments provided by #29752, while they do have a Markdown formatter available for use.

With all that being said, I just want to make clear that I don't propose to add any of the mentioned (Textile and Markdown) formatter alternatives to the Redmine core. Instead, what I wanted to convey with this issue is that given that the core provides support for custom formatters, the functionality provided by #29752 shouldn't be bound to the core formatters only.

Good to note is that this issue is less problematic for Textile because:

1. The only Ruby libraries for Textile parsing are Redmine-maintained RedCloth3 and upstream RedCloth 4.x.
2. Redmine provides the non-optional, Redmine-maintained RedCloth3 library as part of the core.

As such, within the context of #29752, the only issue with it is that it effectively only provides another Textile flavor; the functionality provided by #29752 still largely works when Redmine is using a custom Textile formatter that is based on RedCloth 4.