Redmine - Defect #42172

`format_hours` method produces incorrect output for negative time values when `Setting.timespan_format` is "minutes"

2025-01-27 16:00 - Steven Jones

Status:	Closed		**Start date:**	
Priority:	Normal		**Due date:**	
Assignee:	Go MAEDA		**% Done:**	0%
Category:	Time tracking		**Estimated time:**	0.00 hour
Target version:	6.0.4			
Resolution:	Fixed		**Affected version:**	6.0.2

Description

Following on from [#36897](#) we've spotted a strange formatting issue with time that is negative.

In our use-case we have some 'contracts' that are assigned a project and say have 40 hours a month.

As we log time against a project, we use-up this time assigned to the support contract. So suppose we have done 25 hours of work, then we want to display the obvious calculation:

Start of month	40:00
Used time	25:00
Balance	15:00

We use the format_hours method to display this.

When we've done more than 40 hours of work however, this causes issues, specifically, assume that we've done 40 hours 15 minutes of work in a month, the calculation with Redmine 6.0.2 (and our custom code) looks like this:

Start of month	40:00
Used time	40:15
Balance	-1:45

Which to my eyes looks like we've done 1 hour 45 minutes over the contract.

This is because of the semantics of Ruby modulo arithmetic:

- https://stackoverflow.com/q/16073932/2557659
- https://en.wikipedia.org/wiki/Modulo

And this is purely because of the formatting of the number.

If we changed the display of Time spans to fractional our table looks like this:

Start of month	40.00
Used time	40.25
Balance	-0.25

Which makes more sense to me.

I think the fix is pretty simple, I think that one could add a simple line like this:

```
# If the timespan is negative, perform the calculation on the positive value.
return "-" + format_hours(-hours) if hours < 0
```

To the top of the method.

And then my table would look like this:

Start of month	40:00
Used time	40:15
Balance	-0:15

Which makes much more sense, I think, for a time span.

Related issues:

Related to Redmine - Defect #36897: The minutes part of a time entry is displ...	**Closed**

Associated revisions

Revision 23482 - 2025-01-30 23:11 - Go MAEDA

`format_hours` method produces incorrect output for negative time values when `Setting.timespan_format` is "minutes" (#42172).

Patch by Go MAEDA (user:maeda).

Revision 23483 - 2025-02-01 07:11 - Go MAEDA

Merged r23482 from trunk to 6.0-stable (#42172).

History

#1 - 2025-01-29 02:08 - Go MAEDA

- Status changed from New to Confirmed

#2 - 2025-01-29 02:22 - Go MAEDA

- Related to Defect #36897: The minutes part of a time entry is displayed as 60 instead of being carried over added

#3 - 2025-01-29 02:46 - Go MAEDA

- File 42172.patch added

- Target version set to 6.0.4

Thank you for reporting the issue.

The attached patch fixes the behavior of Redmine::I18n.format_hours when it is given a negative value. Previously, negative values were not handled correctly, resulting in incorrect outputs such as -1:45 for -0.25 hours. This patch ensures that negative values are formatted as expected, for example, -0:15 for -0.25 hours.

#4 - 2025-01-29 11:11 - Steven Jones

I can confirm that this patch fixes the issue, thanks!

#5 - 2025-01-30 07:37 - Go MAEDA

- Subject changed from Negative time can be formatted incorrectly to `format_hours` method produces incorrect output for negative time values when Setting.timespan_format is "minutes"

#6 - 2025-01-30 08:05 - Go MAEDA

- Subject changed from `format_hours` method produces incorrect output for negative time values when Setting.timespan_format is "minutes" to `format_hours` method produces incorrect output for negative time values when `Setting.timespan_format` is "minutes"

#7 - 2025-01-30 23:12 - Go MAEDA

- Status changed from Confirmed to Resolved

- Assignee set to Go MAEDA

- Resolution set to Fixed

Committed the fix in r23482.

#8 - 2025-02-01 07:12 - Go MAEDA

- Status changed from Resolved to Closed

Merged the fix into the 6.0-stable branch in [r23483](#).

Files

42172.patch	1.43 KB	2025-01-29	Go MAEDA

- Status changed from Resolved to Closed

Merged the fix into the 6.0-stable branch in [r23483](#).

Files

42172.patch	1.43 KB	2025-01-29	Go MAEDA