

Store the status name in the issue history

2022/05/09 17:17 - Admin Redmine

ステータス:	New	開始日:	2022/05/09
優先度:	通常	期日:	
担当者:		進捗率:	0%
カテゴリ:	Issues_2	予定工数:	0.00時間
対象バージョン:		作業時間:	0.00時間
Redmineorg_URL:	https://www.redmine.org/issues/16748	status_id:	1
category_id:	2	tracker_id:	2
version_id:	0	plus1:	0
issue_org_id:	16748	affected_version:	
author_id:	54354	closed_on:	
assigned_to_id:	0	affected_version_id:	
comments:	2		

説明

Currently, @journal_details@ table only stores the @id@ of the standard issue fields in @old_value@ and @value@.

This leads to the Issue History displaying the current value of the @id@, and if that @id@ is deleted, the @id@ itself.

For example:

- I change the Status from @New@ to @Accepted@. ** Issue History indicates: @Status changed from New to [Accepted@](#).
- The Admin then change the name of the status to @Just-Before-Accepted@. ** Issue History indicates: @Status changed from New to [Just-Before-Accepted@](#).
- Some time later, issue is closed and this status is deleted. ** Issue History indicates: @Status changed from New to [2@](#).

Displaying of the current name, or id, breaks the audit trail and the context under which the original action was taken.

One way of addressing this is to add @old_id@ and @new_id@ fields to capture the field relations (which if useful for linking, etc) and use the @old_value@ and @value@ fields for displaying.

journals

Note that this affects all standard list/enumeration fields, not just Status.

I've confirmed the behaviour (value changes to current value, or to @id@ when deleted)

- Assignee (deleted Group => @id@, deleted User => @Anonymous@),
- Category,
- Priority,
- Target version.

(I have only checked Issue fields, not Document, etc.)

履歴

#1 - 2022/05/10 17:11 - Admin Redmine

- カテゴリ を Issues_2 にセット