

Bugzilla Workflow Project

1. Objectives

The aim of Bugzilla Workflow Project (named from now on as the Project) is to review, discuss and improve current bug reporting and maintaining procedures, in order to increase consistency, readability and ease of use for ReactOS bugzilla data. Thanks to efficiency, and possible automation, granted by Wiki listings of bugzilla, with customized search filters, it is possible to divide the bug list into smaller, logically sorted and properly described entries, with minimal amount of upkeep required. We hope to make Bugzilla more friendly, especially to project newcomers, future developers and all those that currently feel lost after generating full list of opened bugs.

2. Requirements

In order to combat current problems, we need to define a new way of marking the bug, with enough elasticity to allow future modifications to the schema. New marks should allow filtering from Wiki, present consistency and allow swift recall of bug status and type, even if displayed in sub-par conditions like email or mail-list report. On the other hand, they should not copy current bugzilla functionality (unless there is a reason to do so), create confusion, nor obscure bug readability.

3. Plan

The project will use tags as a main tool. Bug report's SUMMARY field will be used to store the tags followed by the bug summary itself. Each bug will utilize two tags, 1st and 2nd class (defined in following paragraph). Only one tag of each class may be used in the single bug report. Proper tagging must be observed throughout complete bug life cycle. The list of tags might be augmented, provided that Requirements are observed. Deleting tags should not be performed without a very good reason.

4. Definitions

- First class tags: PATCH, METABUG, TRANSLATION, REGRESSION, HACK, UNIMPLEMENTED, ARWINSS and TAG— written in capital letters, with observed spelling, first class tags should be placed at front, followed by colon and separated by a single space;
- HACK: whenever a not proper fix is applied to trunk, for any reason (like patching a nasty issue until proper fix is devised), a bugreport should be opened for this issue. The location of hack should be described, as well as any information that would be useful for solving it properly. This bug should be marked with HACK tag;
- UNIMPLEMENTED: this tag should be marking any bug that is caused by unimplemented API or functionality;
- TAG, used to give some direction to the ReactOS development/fixing process (manually tagging important bugs to improve ROS usability /perception);
- Second class tags: in this category, specific ROS components can be used (like gdi32, usetup, ntoskrnl), written whole in lowercase, or 3rd party program names, without version number with first capital letter, followed by lowercase. Second class tag should also end up with colon and be separated by a single space;

5. Workflow and case resolution

A bug is discovered, happening while starting Freeciv, which is present only in ARWINSS. Reporter creates an entry, with the example summary: "ARWINSS: Freeciv: crash when displaying main menu". The bug is verified by one of the testers and confirmed; alas it occurs that there is a specific reason of this crash – bug within GDI. The bug report is updated and summary should reflect new information gathered. The new title: "ARWINSS: gdi32: On main screen of Freeciv 2.2.1 pervert things happen with pointers" - now points to

different component of bug origin. Now, if someone would create a patch, fixing this issue, the Summary would change again: "PATCH: gdi32: On main screen of Freeciv 2.2.1 pervert things happen with pointers". What is the benefit of all this? Simple look at the title allows distinguishing the bug, as well as the state it is in. It will be automatically listed in different categories, based on static tag filtering: ARWINSS, gdi32, PATCHES. It will be easier to group bugs of similar origin, hence, easier also to spot duplicated reports. We hope this will also allow to fast-track 3rd party commits, both in patches and translations. Finally, it will lessen the amount of work needed for bug up keeping, provided that the proposed rules are strictly observed.