

Spécifications

Mantis BT

Mantis BT

Creating groups in Mantis

Version 1.00 du 3 août 2009

État : Travail

Method and tools

Creating groups in Mantis

Mantis BT / Mantis BT

Method and tools

Le 03/08/2009

Historique :

Version	Date	Origine de la mise à jour	Rédigée par	Validée par
1.00	20/07/2009	Creation	S. Veyret	

Préambule :

This document has been written in English by a non-Anglophone. Therefore, there may be some language errors, or some parts you do not perfectly understand. In case of any doubt, do not hesitate to contact the author and ask him to clear some points.

Sommaire :

1. INTRODUCTION	5
2. EXISTING	5
3. PRINCIPLE	5
4. ADMINISTRATORS	6
5. MIGRATION	6
6. GROUPS	7
7. USERS	7
8. PROJECTS	7

1. INTRODUCTION

When managing hundreds of projects and users, as we are doing in our company, we soon need a lot of time for the management of Mantis. There are a lot of users who need different rights on different projects depending on the team they belong to. Teams and projects are not always linked together.

In our company, for example, we have what we call “solutions”, each of them is composed of a set of projects. One project can be used by several solutions. There are solution-qualification teams which members should be able to report on all solution projects. Adding a member to such a team requires that each manager of each concerned project add reporter rights to this new member.

A time-saving solution would be to create the notion of user groups in Mantis. The purpose of this document is to describe what exactly is expected by this new feature.

2. EXISTING

The current way for managing user rights is to use one of the pre-defined access levels. These levels can be customized using the “config_inc.php” file, but the concept of “levels” does not always match the user needs. Indeed, we may need that a category of users can execute a command A but not B, whereas another category can execute B but not A. We cannot say then that one category has a higher level than the other.

3. PRINCIPLE

A new object type named “group” is created and access levels are removed. A non-removable group named “ADMINISTRATOR” is created for all Mantis administrators.

A group contains a name and links to:

- A list of users and groups who are group managers. They represent users who can manage the members of this group and rename the group. One value of this list can be “[self]” and would mean that every member of this list is also manager (this is the case for the “ADMINISTRATOR” group).
- Users and groups who are members of the group. All permissions given to the current group are also recursively given to its members.

At every point of the product where we need to grant rights to anyone, and for each specific type of right, we should find a list in which users and groups can be added (or removed).



At some of these points, the list may contain special values like “[self]” (in the group page to indicate the members of the current group), “[author]” (to indicate the author of the issue, the note, the new, etc.), “[assignee]” (to indicate the person to which the issue is assigned), “[nobody]” or “[everybody]”.

It is still possible to create a hierarchy of groups by including groups of higher privilege in the groups of lower privilege.

4. ADMINISTRATORS

People of the “ADMINISTRATOR” group have special privileges.

- They can create or remove groups.
- An administrator can also put himself in any group and then can take all rights to do any action.
- Administrators can access a special page (“manage rights”) used to set rights on different actions.

The “manage rights” page defines which people or groups can do what actions. The actions we would find there can be: create, modify or remove news, users, custom fields or profiles, change configuration, create, delete or modify each field of projects and everything which currently defines an access level in the `config_inc.php` file.

Concerning projects, this page should also allow setting default values for every action linked to the project, like manage project level news, versions, categories, used custom fields, view, create, modify, correct (be assigned to), delete an issue, and every available status of issues. It should also be possible to set notifications the same way, i.e. define which user or group will be notified when a new issue is created in the project, or when changes are done to issues.

5. MIGRATION

While migrating from an older version of Mantis, a few changes need to be done:

- In the “`constant_inc.php`” file, access level constants are replaced with strings of the same name (e.g. `define('VIEWER', 'VIEWER');`).
- In the “`config_defaults_inc.php`” file, remove the “`access_levels_enum_string`” variable.
- Create a group for each access level, containing all users with this level and every group created for higher access level :
 - Create the “ADMINISTRATOR” group and put all administrators in it.
 - Create the “MANAGERS” group and put all managers and “ADMINISTRATOR” group in it.
 - And so on...
- Change database to remove access level for the users.
- Create a table with more precise access rights for project, fill in this table using the currently set access rights and remove the current access rights.

6. GROUPS

New pages should be created for the group management. These pages could look like the pages for user management. We should have one page with the list of all groups, and one page with the details for a given group.

This last page could be used to change the group name, set the managers of the group (set to “[self]” and not editable in the “ADMINISTRATOR” group), and set members of the group. All these elements are only editable by group managers and administrators.

7. USERS

Before the “Add user to project” box, the “manage_user_edit_page” must contain a “Add user to group” box, built the same way, which allow to set the user as member of groups.

8. PROJECTS

When created, projects are defined with the default values set in the “manage rights” page. They, of course, can be modified. Managing access to a given project may seem more complex because of all possible options instead of current access levels. But actually, if groups are well used, these accesses should be done once when creating the project. Managing access should only be done by managing group members.