Trac Permissions

Error: Macro TracGuideToc(None) failed

```
'NoneType' object has no attribute 'find'
```

Trac uses a simple, case sensitive, permission system to control what users can and can't access.

Permissions are managed using the trac-admin tool or the *General / Permissions* panel in the *Admin* tab of the web interface.

In addition to the default permission policy described in this page, it is possible to activate additional permission policies by enabling plugins and listing them in [trac] permission_policies. See TracFineGrainedPermissions for more details.

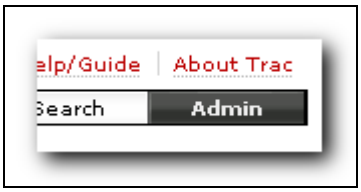
Non-authenticated users accessing the system are assigned the name *anonymous*. Assign permissions to the *anonymous* user to set privileges for anonymous/guest users. The parts of Trac that a user does not have privilege for will not be displayed in the navigation. In addition to these privileges, users can be granted additional individual rights in effect when authenticated and logged into the system. All logged in users belong to the virtual group *authenticated*, which inherits permissions from *anonymous*.

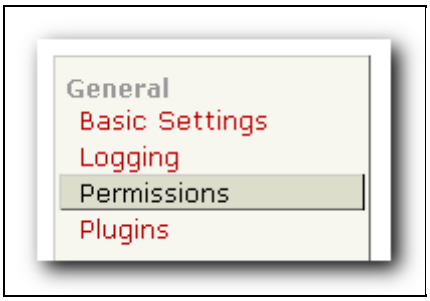
Graphical Admin Tab

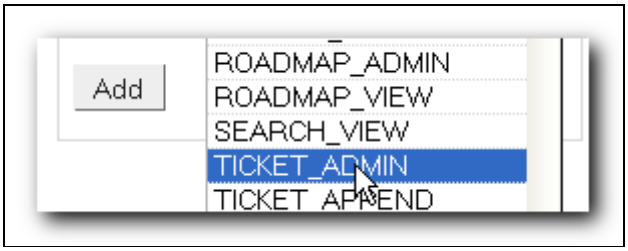
To access this tab, a user must have one of the following permissions: `TRAC_ADMIN`, `PERMISSION_ADMIN`, `PERMISSION_GRANT`, `PERMISSION_REVOKE`. The permissions can be granted using the `trac-admin` command (more on `trac-admin` below):

```
$ trac-admin /path/to/projenv permission add bob TRAC_ADMIN
```

Then, the user `bob` will be able to see the Admin tab, and can access the permissions menu. This menu will allow you to perform all the following actions, but from the browser rather than requiring root access to the server. **Use at least one lowercase character in user names, as all-uppercase names are reserved for permissions.**







From the graphical admin tab, users with `PERMISSION_GRANT` will only be allowed to grant permissions that they possess, and users with `PERMISSION_REVOKE` will only be allowed to revoke permissions that they possess. For example, a user cannot grant `MILESTONE_ADMIN` unless they have `PERMISSION_GRANT` and `MILESTONE_ADMIN`, and they cannot revoke `MILESTONE_ADMIN` unless they have `PERMISSION_REVOKE` and `MILESTONE_ADMIN`. `PERMISSION_ADMIN` just grants the user both `PERMISSION_GRANT` and `PERMISSION_REVOKE`, and users with `TRAC_ADMIN` can grant or revoke any permission.

Available Privileges

To enable all privileges for a user, use the `TRAC_ADMIN` permission. Having `TRAC_ADMIN` is like being `root` on a *NIX system: it will allow you to perform any operation.

Otherwise, individual privileges can be assigned to users for the various different functional areas of Trac (**note that the privilege names are case-sensitive**):

Repository Browser

`BROWSER_VIEW`	View directory listings in the repository browser
`FILE_VIEW`	View files in the repository browser
`CHANGESET_VIEW`	View repository check-ins
`LOG_VIEW`	View revision logs of files and directories in the repository browser

Ticket System

`TICKET_VIEW`	View existing tickets and perform ticket queries
`TICKET_CREATE`	Create new tickets
`TICKET_APPEND`	Add comments or attachments to tickets
`TICKET_CHGPROP`	Modify ticket properties (priority, assignment, keywords, etc.) with the following exceptions: edit description field, add/remove other users from cc field when logged in
`TICKET_MODIFY`	Includes both `TICKET_APPEND` and `TICKET_CHGPROP`, and in addition allows resolving tickets in the default workflow. Tickets can be assigned to users through a drop-down list when the list of possible owners has been restricted.
`TICKET_EDIT_CC`	Full modify cc field
`TICKET_EDIT_DESCRIPTION`	Modify description field
`TICKET_EDIT_COMMENT`	Modify another user's comments. Any user can modify their own comments by default.
`TICKET_BATCH_MODIFY`	Batch modify tickets

TICKET_ADMIN	All TICKET_* permissions, deletion of ticket attachments and modification of the reporter field, which grants ability to create a ticket on behalf of another user (it will appear that another user created the ticket). It also allows managing ticket properties through the web administration module.

Roadmap

MILESTONE_VIEW	View milestones and assign tickets to milestones.
MILESTONE_CREATE	Create new milestones
MILESTONE_MODIFY	Modify milestones
MILESTONE_DELETE	Delete milestones
MILESTONE_ADMIN	All MILESTONE_* permissions
ROADMAP_VIEW	View the roadmap page, which is not yet the same as MILESTONE_VIEW, see ?#4292
ROADMAP_ADMIN	to be removed with ?#3022, replaced by MILESTONE_ADMIN

Reports

REPORT_VIEW	View reports, i.e. the *View Tickets* link.
REPORT_SQL_VIEW	View the SQL query of a report
REPORT_CREATE	Create new reports
REPORT_MODIFY	Modify reports
REPORT_DELETE	Delete reports
REPORT_ADMIN	All REPORT_* permissions

Wiki System

WIKI_VIEW	View wiki pages
WIKI_CREATE	Create new wiki pages
WIKI_MODIFY	Modify wiki pages
WIKI_RENAME	Rename wiki pages
WIKI_DELETE	Delete wiki pages and attachments
WIKI_ADMIN	All WIKI_* permissions, plus the management of *readonly* pages.

Permissions

PERMISSION_GRANT	add/grant a permission
PERMISSION_REVOKE	remove/revoke a permission
PERMISSION_ADMIN	All PERMISSION_* permissions

Others

TIMELINE_VIEW	View the timeline page
SEARCH_VIEW	View and execute search queries
CONFIG_VIEW	Enables additional sections on *About Trac* that show the current configuration and the list of installed plugins
EMAIL_VIEW	Shows email addresses even if trac show_email_addresses configuration option is false

Granting Privileges

You grant privileges to users using trac-admin. The current set of privileges can be listed with the following command:

```
$ trac-admin /path/to/projenv permission list
```

This command will allow the user *bob* to delete reports:

```
$ trac-admin /path/to/projenv permission add bob REPORT_DELETE
```

The `permission add` command also accepts multiple privilege names:

```
$ trac-admin /path/to/projenv permission add bob REPORT_DELETE WIKI_CREATE
```

Or add all privileges:

```
$ trac-admin /path/to/projenv permission add bob TRAC_ADMIN
```

Permission Groups

There are two built-in groups, *authenticated* and *anonymous*. Any user who has not logged in is automatically in the *anonymous* group. Any user who has logged in is also in the *authenticated* group. The *authenticated* group inherits permissions from the *anonymous* group. For example, if the *anonymous* group has permission WIKI_MODIFY, it is not necessary to add the WIKI_MODIFY permission to the *authenticated* group as well.

Custom groups may be defined that inherit permissions from the two built-in groups.

Permissions can be grouped together to form roles such as *developer*, *admin*, etc.

```
$ trac-admin /path/to/projenv permission add developer WIKI_ADMIN
$ trac-admin /path/to/projenv permission add developer REPORT_ADMIN
$ trac-admin /path/to/projenv permission add developer TICKET_MODIFY
$ trac-admin /path/to/projenv permission add bob developer
$ trac-admin /path/to/projenv permission add john developer
```

Group membership can be checked by doing a `permission list` with no further arguments; the resulting output will include group memberships. **Use at least one lowercase character in group names, as all-uppercase names are reserved for permissions**.

Adding a New Group and Permissions

Permission groups can be created by assigning a user to a group you wish to create, then assign permissions to that group.

The following will add *bob* to the new group called *beta_testers* and then will assign WIKI_ADMIN permissions to that group. (Thus, *bob* will inherit the WIKI_ADMIN permission)

```
$ trac-admin /path/to/projenv permission add bob beta_testers
$ trac-admin /path/to/projenv permission add beta_testers WIKI_ADMIN
```

Removing Permissions

Permissions can be removed using the 'remove' command. For example:

This command will prevent the user *bob* from deleting reports:

```
$ trac-admin /path/to/projenv permission remove bob REPORT_DELETE
```

Just like `permission add`, this command accepts multiple privilege names.

You can also remove all privileges for a specific user:

```
$ trac-admin /path/to/projenv permission remove bob '*'
```

Or one privilege for all users:

```
$ trac-admin /path/to/projenv permission remove '*' REPORT_ADMIN
```

Creating New Privileges

To create custom permissions, for example to be used in a custom workflow, enable the optional ?tracopt.perm.config_perm_provider.ExtraPermissionsProvider component in the "Plugins" admin panel, and add the desired permissions to the `[extra-permissions]` section in your trac.ini. For more information, please refer to the documentation on the TracIni page after enabling the component.

Default Permissions

By default on a new Trac installation, the *anonymous* user will have *view* access to everything in Trac, but will not be able to create or modify anything. On the other hand, the *authenticated* users will have the permissions to *create and modify tickets and wiki pages*.

anonymous

```
BROWSER_VIEW
CHANGESET_VIEW
FILE_VIEW
LOG_VIEW
MILESTONE_VIEW
REPORT_SQL_VIEW
REPORT_VIEW
ROADMAP_VIEW
SEARCH_VIEW
TICKET_VIEW
TIMELINE_VIEW
WIKI_VIEW
```

authenticated

```
TICKET_CREATE
TICKET_MODIFY
WIKI_CREATE
WIKI_MODIFY
```

See also: TracAdmin, TracFineGrainedPermissions