Trac Permissions

Error: Macro TracGuideToc(None) failed

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

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

Permission privileges are managed using the trac-admin tool.

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 the privileges 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".

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

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.) except description field
`TICKET_MODIFY`	Includes both `TICKET_APPEND` and `TICKET_CHGPROP`, and in addition allows resolving tickets
`TICKET_ADMIN`	All `TICKET_*` permissions, plus the deletion of ticket attachments and modification of the description field

Roadmap

`MILESTONE_VIEW`	View a milestone
`MILESTONE_CREATE`	Create a new milestone
`MILESTONE_MODIFY`	Modify existing milestones
`MILESTONE_DELETE`	Delete milestones

`MILESTONE_ADMIN`	All `MILESTONE_*` permissions
`ROADMAP_VIEW`	View the roadmap page

Reports

`REPORT_VIEW`	View reports
`REPORT_SQL_VIEW`	View the underlying SQL query of a report
`REPORT_CREATE`	Create new reports
`REPORT_MODIFY`	Modify existing reports
`REPORT_DELETE`	Delete reports
`REPORT_ADMIN`	All `REPORT_*` permissions

Wiki System

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

Others

`TIMELINE_VIEW`	View the timeline page
`SEARCH_VIEW`	View and execute search queries
`CONFIG_VIEW`	Enables additional pages on *About Trac* that show the current configuration or the list of installed plugins

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
```

Permission 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 lowercase for group names, as uppercase is reserved for permissions.

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
```

Default Permissions

Granting privileges to the special user *anonymous* can be used to control what an anonymous user can do before they have logged in.

In the same way, privileges granted to the special user *authenticated* will apply to any authenticated (logged in) user.

See also: TracAdmin, TracGuide and ?FineGrainedPermissions