Wikiprint Book

Title: Fine grained permissions

Subject: eDokumenty - elektroniczny system obiegu dokumentów, workflow i CRM - TracFineGrainedPermissions

Version: 1

Date: 12/15/25 20:21:05

Table of Contents

Fine grained permissions	3 3 3 3
Permission Policies	
AuthzPolicy	
mod_authz_svn-like permission policy	
Trac Configuration	4
Subversion Configuration	4

Fine grained permissions

Before Trac 0.11, it was only possible to define fine-grained permissions checks on the repository browser sub-system.

Since 0.11, there's a general mechanism in place that allows custom permission policy plugins to grant or deny any action on any kind of Trac resources, even at the level of specific versions of such resources.

Permission Policies

AuthzPolicy

An example policy based on an Authz-style system has been added. See <u>trunk/sample-plugins/permissions/authz_policy.py</u> for details. (See also <u>trunk/sample-plugins/permissions</u> for more samples.)

- Install <a>!ConfigObj (required).
- · Copy authz_policy.py into your plugins directory.
- Put a <u>Bauthzpolicy.conf</u> file somewhere (preferably on a secured location on the server, not readable for others than the webuser.
 Update your trac.ini:

```
[trac]
...
permission_policies = AuthzPolicy, DefaultPermissionPolicy, LegacyAttachmentPolicy

[authz_policy]
authz_file = /some/trac/env/conf/authzpolicy.conf

[components]
...
authz_policy = enabled
```

Note that the order in which permission policies are specified is quite critical, as policies will be examined in the sequence provided.

A policy will return either True, False or None for a given permission check. Only if the return value is None will the *next* permission policy be consulted. If no policy explicitly grants the permission, the final result will be False (i.e. no permission).

For example, if the authz_file contains:

```
[wiki:WikiStart@*]
* = WIKI_VIEW

[wiki:PrivatePage@*]
john = WIKI_VIEW
* =
```

and the default permissions are set like this:

```
john WIKI_VIEW
jack WIKI_VIEW
# anonymous has no WIKI_VIEW
```

Then:

- · All versions of WikiStart will be viewable by everybody (including anonymous)
- PrivatePage will be viewable only by john
- other pages will be viewable only by john and jack

mod_authz_svn-like permission policy

At the time of this writing, the old fine grained permissions system from Trac 0.10 and before used for restricting access to the repository has not yet been converted to a permission policy component, but from the user point of view, this makes little if no differences.

That kind of fine-grained permission control needs a definition file, which is the one used by Subversion's mod_authz_svn. More information about this file format and about its usage in Subversion is available in the Subversion Book (Per-Directory Access Control).

Example:

```
[/]
* = r

[/branches/calc/bug-142]
harry = rw
sally = r

[/branches/calc/bug-142/secret]
harry =
```

- I = Everyone has read access by default
- Ibranches/calc/bug-142 = harry has read/write access, sally read only
- Ibranches/calc/bug-142/secret = harry has no access, sally has read access (inherited as a sub folder permission)

Trac Configuration

To activate fine grained permissions you <u>must</u> specify the authz_file option in the [trac] section of trac.ini. If this option is set to null or not specified the permissions will not be used.

```
[trac]
authz_file = /path/to/svnaccessfile
```

 $if you want to support the use of the \ [\textit{modulename:/some/path}] \ syntax \ within the \ authz_file, \ add$

```
authz_module_name = modulename
```

 $where \textit{ module name} \textit{ refers to the same repository indicated by the \textit{ repository_dir entry in the [trac] section.} \\$

Note: Usernames inside the Authz file $\underline{\text{must}}$ be the same as those used inside trac.

Subversion Configuration

The same access file is typically applied to the corresponding Subversion repository using an Apache directive like this:

```
<Location /repos>
DAV svn
SVNParentPath /usr/local/svn

# our access control policy
AuthzSVNAccessFile /path/to/svnaccessfile
</Location>
```

For information about how to restrict access to entire projects in a multiple project environment see @wiki:TracMultipleProjectsSVNAccess

See also: TracPermissions