Table of Contents

TWiki Access Control

Restricting read and write access to topics and webs, by Users and groups

TWiki Access Control allows you restrict access to single topics and entire webs, by individual user and by user Groups. Access control, combined with TWikiUserAuthentication, lets you easily create and manage an extremely flexible, fine-grained privilege system.

Tip: TWiki:TWiki.TWikiAccessControlSupplement on TWiki.org has additional documentation on access control.

An Important Control Consideration

Open, freeform editing is the essence of WikiCulture - what makes TWiki different and often more effective than other collaboration tools. For that reason, it is strongly recommended that decisions to restrict read or write access to a web or a topic are made with great care - the more restrictions, the less Wiki in the mix. Experience shows that *unrestricted write access* works very well because:

- **Peer influence** is enough to ensure that only relevant content is posted.
- **Peer editing** - the ability for anyone to rearrange all content on a page - keeps topics focused.
- In TWiki, content is transparently preserved under **revision control**:
  - Edits can be undone by the administrator (per default a member of TWikiAdminGroup; see #ManagingGroups).
  - Users are encouraged to edit and refactor (condense a long topic), since there's a safety net.

As a **collaboration guideline**:

- Create broad-based Groups (for more and varied input), and...
- Avoid creating view-only Users (if you can read it, you should be able to contribute to it).

Authentication vs. Access Control

Authentication: Identifies who a user is based on a login procedure. See TWikiUserAuthentication.

Access control: Restrict access to content based on users and groups once a user is identified.

Users and Groups

Access control is based on the familiar concept of Users and Groups. Users are defined by their WikiNames. They can then be organized in unlimited combinations by inclusion in one or more user Groups. For convenience, Groups can also be included in other Groups.

Managing Users

A user can create an account in TWikiRegistration. The following actions are performed:

- WikiName and encrypted password are recorded using the password manager if authentication is enabled.
- A confirmation e-mail is sent to the user.

- A user profile page with the WikiName of the user is created in the Main web.
- The user is added to the TWikiUsers topic.

The default visitor name is TWikiGuest. This is the non-authenticated user.

Authorisation via E-groups

At CERN, we are using authorisation via LDAP and E-groups.

To use an E-group instead of a regular TWiki group, please simply write the E-group name in lower-case, withouth `@cern.ch` at the end, like in the examples below:

- For a dynamic group with members of the CERN IT PES group: `it-dep-pes`
- E-group with members in the Alice experiment: `alice-member`

The corresponding TWiki access control settings for E-groups would be:

- **`Set ALLOWTOPICCHANGE = it-dep-pes`**
- **`Set ALLOWTOPICCHANGE =  alice-member`**

See TheLHCexperimentsWebs for more information and CERN access control examples.

Managing Groups

The following describes the standard TWiki support for groups. At CERN, we are using authorisation via LDAP and E-groups instead. Use of standard TWiki groups is now depricated, although access control based on them works until further notice.

TWiki Groups are defined by group topics located in the **`Main`** web. To create a new group, visit TWikiGroups and enter the name of the new group ending in **`Group`** into the "new group" form field. This will create a new group topic with two important settings:

- **`Set GROUP = < list of Users and/or Groups >`**
- **`Set ALLOWTOPICCHANGE = < list of Users and/or Groups >`**

The GROUP setting is a comma-separated list of users and/or other groups. Example:

- **`Set GROUP = SomeUser, OtherUser, SomeGroup`**

The ALLOWTOPICCHANGE setting defines who is allowed to change the group topic; it is a comma delimited list of users and groups. You typically want to restrict that to the members of the group itself, so it should contain the name of the topic. This prevents users not in the group from editing the topic to give themselves or others access. For example, for the MarketingGroup topic write:

- **`Set ALLOWTOPICCHANGE = MarketingGroup`** or

Note: TWiki has strict formatting rules. Make sure you have a real bullet. (In raw edit it is three or six spaces, an asterisk, and an extra space in front of any access control rule.)

The Super Admin Group

A number of TWiki functions (for example, renaming webs) are only available to administrators. Administrators are simply users who belong to the **SuperAdminGroup**. This is a standard user group, the

name of which is defined by {SuperAdminGroup} setting in configure. The default name of this group is the `TWikiAdminGroup`. The system administrator may have chosen a different name for this group if your local TWiki uses an alternate group mapping manager but for simplicity we will use the default name TWikiAdminGroup in the rest of this topic.

You can create new administrators simply by adding them to the TWikiAdminGroup topic. For example,

- **`Set GROUP = RobertCailliau, TimBernersLee`**

A member of the Super Admin Group has unrestricted access throughout the TWiki, so only trusted staff should be added to this group.

Restricting Access

You can define who is allowed to read or write to a web or a topic. Note that some plugins may not respect access permissions.

- Restricting VIEW blocks viewing and searching of content. When you restric VIEW to a topic or web, this also restricts INCLUDE and Formatted SEARCH from showing the content of the topics.
- Restricting CHANGE blocks creating new topics, changing topics or attaching files.
- Restricting RENAME prevents renaming of topics within a web.

Note that there is an important distinction between CHANGE access and RENAME access. A user can CHANGE a topic, but thanks to version control their changes cannot be lost (the history of the topic before the change is recorded). However if a topic or web is renamed, that history may be lost. Typically a site will only give RENAME access to administrators and content owners.

Controlling access to a Web

You can define restrictions on who is allowed to view a TWiki web. You can restrict access to certain webs to selected Users and Groups, by:

- **authenticating all webs and restricting selected webs:** Topic access in all webs is authenticated, and selected webs have restricted access.
- **authenticating and restricting selected webs only:** Provide unrestricted viewing access to open webs, with authentication and restriction only on selected webs.

- You can define these settings in the WebPreferences topic, preferable towards the end of the topic:
  - **`Set DENYWEBVIEW = < comma-delimited list of Users and Groups >`**
  - **`Set ALLOWWEBVIEW = < comma-delimited list of Users and Groups >`**
  - **`Set DENYWEBCHANGE = < comma-delimited list of Users and Groups >`**
  - **`Set ALLOWWEBCHANGE = < comma-delimited list of Users and Groups >`**
  - **`Set DENYWEBRENAME = < comma-delimited list of Users and Groups >`**
  - **`Set ALLOWWEBRENAME = < comma-delimited list of Users and Groups >`**

For example, set this to restrict a web to be viewable only by the MarketingGroup:

- **`Set ALLOWWEBVIEW = Main.MarketingGroup`**

If your site allows hierarchical webs, then access to sub-webs is determined from the access controls of the parent web, plus the access controls in the sub-web. So, if the parent web has **`ALLOWWEBVIEW`** set, this will also apply to the subweb. Also note that you will need to ensure that the parent web's **`FINALPREFERENCES`** does not include the access control settings listed above. Otherwise you will not be able override the parent web's

access control settings in sub-webs.

Creation and renaming of sub-webs is controlled by the WEBCHANGE setting on the parent web (or ROOTCHANGE for root webs). Renaming is additionally restricted by the setting of WEBRENAME in the web itself.

Note: If you restrict access to the Main, make sure to add the `TWikiRegistrationAgent` so that users can register. Example:

- **`Set ALLOWWEBCHANGE = TWikiAdminGroup, TWikiRegistrationAgent`**

Note: For Web level access rights Setting any of these settings to an empty value has the same effect as not setting them at all. Please note that the documentation of TWiki 4.0 and earlier versions of TWiki 4.1 did not reflect the actual implementation, e.g. an empty ALLOWWEBVIEW does *not* prevent anyone from viewing the web, and an an empty DENYWEBVIEW does *not* allow all to view the web.

Controlling access to a Topic

- You can define these settings in any topic, preferable towards the end of the topic:
  - **`Set DENYTOPICVIEW = < comma-delimited list of Users and Groups >`**
  - **`Set ALLOWTOPICVIEW = < comma-delimited list of Users and Groups >`**
  - **`Set DENYTOPICCHANGE = < comma-delimited list of Users and Groups >`**
  - **`Set ALLOWTOPICCHANGE = < comma-delimited list of Users and Groups >`**
  - **`Set DENYTOPICRENAME = < comma-delimited list of Users and Groups >`**
  - **`Set ALLOWTOPICRENAME = < comma-delimited list of Users and Groups >`**

For example, set this to restrict a topic to be viewable only by the MarketingExecGroup:

- **`Set ALLOWTOPICVIEW = Main.MarketingExecGroup`**

Remember when opening up access to specific topics within a restricted web that other topics in the web - for example, the WebLeftBar - may also be accessed when viewing the topics. The message you get when you are denied access should tell you what topic you were not permitted to access.

Be careful with empty values for any of these.

- **`Set ALLOWTOPICVIEW =`**
  This means the same as not setting it at all. (This was documented wrong in versions 4.0.X, 4.1.0 and 4.1.1)

- **`Set DENYTOPICVIEW =`**
  Since TWiki 4.0 this means *do not deny anyone the right to view this topic*. If DENYTOPICVIEW is set to an empty value anyone has access even if ALLOWTOPICVIEW or ALLOWWEBVIEW is defined. This allows to have very restrictive default access rights to an entire web and still allow individual topics to have more open access.

The same rules apply to ALLOWTOPICCHANGE/DENYTOPICCHANGE and APPLYTOPICRENAME/DENYTOPICRENAME. Setting ALLOWTOPICCHANGE or ALLOWTOPICRENAME to en empty value means the same as not defining it. Setting DENYTOPICCHANGE or DENYTOPICRENAME to an empty value means that anyone can edit or rename the topic.

 If the same setting is defined multiple times the last one overrides the previous. They are not OR'ed together.

The setting to an empty has caused confusion and great debate and it has been decided that the empty setting syntax will be replaced by something which is easier to understand in a later version of TWiki. A method to upgrade will be provided. Please read the release notes carefully when you upgrade.

See "How TWiki evaluates ALLOW/DENY settings" below for more on how ALLOW and DENY interacts.

Securing File Attachments

By default, TWiki does not secure file attachments. Without making the following changes to the twiki.conf file, it is possible for anyone who has access to the server to gain access to an attachment if they know the attachment's fully qualified path, even though access to the topic associated with the attachment is secured. This is because attachments are referred to directly by Apache, and are not by default delivered via TWiki scripts. This means that the above instructions for controlling to topics do ***not*** apply to attachments unless you make the changes as described below.

An effective way to secure attachments is to apply the same access control settings to attachments as those applied to topics. This security enhancement can be accomplished by instructing the webserver via Apache's `mod_rewrite` module to redirect accesses to attachments via the TWiki `viewfile` script, which honors the TWiki access controls settings to topics.

The preferred method to secure attachments is by editing the `twiki.conf` file to include:

```
ScriptAlias /twiki/bin/ /filesystem/path/to/twiki/bin/
Alias /twiki/pub/       /filesystem/path/to/twiki/pub/

RewriteEngine on
RewriteCond %{REQUEST_URI} !^/+twiki/+pub/+(TWiki|Sandbox)/+.+
RewriteRule ^/+twiki/+pub/+(.*)$  /twiki/bin/viewfile/$1 [L,PT]
```

Notes:

- You can use TWiki:TWiki/ApacheConfigGenerator to generate the Apache config file for TWiki.
- You will need to restart your Apache server after this change.
- Images embedded in topics will load slower since attached images will also be delivered by the `viewfile` script. The TWiki web and Sandbox web are excluded for performance reasons.
- As an alternative to editing the `twiki.conf` file used by Apache, you can make the same change directly to the `.htaccess` file in the `/twiki/bin` directory.
- The `viewfile` script sets the mime type based upon file name suffix. Unknown types are served as text/plain which can result in corrupt files.

Controlling who can manage top-level webs

Top level webs are a special case, because they don't have a parent web with a WebPreferences. So there has to be a special control just for the root level.

- You can define these settings in the Main.TWikiPreferences topic, preferable towards the end of the topic:
  - **Set DENYROOTCHANGE = < comma-delimited list of Users and Groups >**
  - **Set ALLOWROOTCHANGE = < comma-delimited list of Users and Groups >**

Note that you do **not** require `ROOTCHANGE` access to rename an existing top-level web. You just need `WEBCHANGE` in the web itself.

How TWiki evaluates ALLOW/DENY settings

When deciding whether to grant access, TWiki evaluates the following rules in order (read from the top of the list; if the logic arrives at **PERMITTED** or **DENIED** that applies immediately and no more rules are applied). You need to read the rules bearing in mind that VIEW, CHANGE and RENAME access may be granted/denied separately.

1. If the user is an administrator
   - access is **PERMITTED**.
2. If DENYTOPIC is set to a list of wikinames
   - people in the list will be **DENIED**.
3. If DENYTOPIC is set to *empty* (i.e. `Set DENYTOPIC =`)
   - access is **PERMITTED** *i.e* no-one is denied access to this topic.
     Attention: Use this with caution. This is ***deprecated*** and will likely change in the next release.
4. If ALLOWTOPIC is set
   1. people in the list are **PERMITTED**
   2. everyone else is **DENIED**
5. If DENYWEB is set to a list of wikinames
   - people in the list are **DENIED** access
6. If ALLOWWEB is set to a list of wikinames
   - people in the list will be **PERMITTED**
   - everyone else will be **DENIED**
7. If you got this far, access is **PERMITTED**

Access control and INCLUDE

ALLOWTOPICVIEW and ALLOWTOPICCHANGE only applies to the topic in which the settings are defined. If a topic A includes another topic B, topic A does not inherit the access rights of the included topic B.

Examples: Topic A includes topic B

- If the included topic B has ALLOWTOPICCHANGE set to block editing for a user, it does not prevent editing the including topic A.
- If the included topic B has ALLOWTOPICVIEW set to block view for a user, the user can still view topic A but he cannot see the included topic B. He will see a message *No permission to view B*

Access Control quick recipes

Restrict Access to Whole TWiki Site

For a firewalled TWiki, e.g. an intranet wiki or extranet wiki, you want to allow only invited people to access your TWiki. In this case, enable user authentication with ApacheLogin and lock down access to the whole `twiki/bin` and `twiki/pub` directories to all but valid users. In the Apache `.htaccess` file or the appropriate `.conf` file, replace the `<FilesMatch "(attach|edit|...` section with this:

```
<FilesMatch ".*">
       require valid-user
</FilesMatch>
```

If needed, you can further restrict access to selected webs with ALLOWWEBVIEW and other access control

settings.

Note: With this configuration, someone with access to the site needs to register new users.

Authenticate all Webs and Restrict Selected Webs

Use the following setup to authenticate users for topic viewing in all webs and to restrict access to selected webs. Requires TWikiUserAuthentication to be enabled.

1. Set `require valid-user` on your `view` script in .htaccess or the appropriate Apache .conf file. As of 4.x, this looks like: `FilesMatch "(attach|edit|manage|rename|save|view|upload|mail|logon|.*auth).*"` (normally `view` is not in that list).
2. **Restrict** view access to selected Users and Groups. Set one or both of these variables in its WebPreferences topic:
   - **`Set DENYWEBVIEW = < list of Users and Groups >`**
   - **`Set ALLOWWEBVIEW = < list of Users and Groups >`**
   - ***Note:*** `DENYWEBVIEW` is evaluated before `ALLOWWEBVIEW`. Access is denied if the authenticated person is in the `DENYWEBVIEW` list, or not in the `ALLOWWEBVIEW` list. Access is granted if `DENYWEBVIEW` and `ALLOWWEBVIEW` are not defined.
3. If you still want public users to be able to register automatically follow TWiki:TWiki.RegisterOnViewRestrictedSite.

Authenticate and Restrict Selected Webs Only

Use the following setup to provide unrestricted viewing access to open webs, with authentication only on selected webs. Requires TWikiUserAuthentication to be enabled.

1. **Restrict** view access to selected Users and Groups. Set one or both of these variables in its WebPreferences topic:
   - **`Set DENYWEBVIEW = < list of Users and Groups >`**
   - **`Set ALLOWWEBVIEW = < list of Users and Groups >`**
   - ***Note:*** `DENYWEBVIEW` is evaluated before `ALLOWWEBVIEW`. Access is denied if the authenticated person is in the `DENYWEBVIEW` list, or not in the `ALLOWWEBVIEW` list. Access is granted if `DENYWEBVIEW` and `ALLOWWEBVIEW` are not defined.

Hide Control Settings

Tip: To hide access control settings from normal browser viewing, you can put them into the *topic preference settings* by clicking the link `Edit topic preference settings` under `More topic actions` menu. Preferences set in this manner are not visible in the topic text, but take effect nevertheless. Access control settings added as topic preference settings are stored in the topic meta data and they override settings defined in the topic text.

Alternatively, place them in HTML comment markers, but this exposes the access setting during ordinary editing.

```
<!--
   * Set DENYTOPICCHANGE = Main.SomeGroup
-->
```

Obfuscating Webs

Another way of hiding webs is to keep them hidden by not publishing the URL and by preventing the **`all webs`** search option from accessing obfuscated webs. Do so by enabling the **`NOSEARCHALL`** variable in WebPreferences:

- **`Set NOSEARCHALL = on`**

This setup can be useful to hide a new web until content its ready for deployment, or to hide view access restricted webs.

Note: Obfuscating a web without view access control is **very** insecure, as anyone who knows the URL can access the web.

Read-only Skin Mode

It is possible to turn the PatternSkin and TopMenuSkin into read-only mode by removing the edit and attach controls (links and buttons). This is mainly useful if you have TWiki application pages or dashboards where you do not want regular users to change content. The read-only skin mode is not a replacement for access control; you can use it in addition to access control. Details at PatternSkinCustomization#ReadOnlySkinMode.

Note: Administrators may check the access on Webs under SitePermissions

Related Topics: AdminDocumentationCategory, TWikiUserAuthentication, TWiki:TWiki.TWikiAccessControlSupplement

-- ***Contributors:*** TWiki:Main.PeterThoeny, TWiki:Main.MikeMannix, TWiki:Main.CrawfordCurrie

---

This topic: TWiki21Nov > TWikiAccessControl
Topic revision: r40 - 2014-01-31 - NilsHoeimyr