Table of Contents

Access Control **1**
- An important control consideration 1
- Permissions settings of the webs on this Foswiki site 1
- Authentication vs. Access Control 1
- Users and groups 2
  - Managing Users 2
  - Managing Groups 2
  - The Super Admin Group 2
- Restricting Access 3
  - Controlling access to a Web 3
  - Controlling access to a topic 4
  - Controlling access to attachments 5
  - Controlling who can manage top-level webs 5
  - How Foswiki evaluates ALLOW/DENY settings 5
- Access control and INCLUDE 6
- Access control quick recipes 6
  - Obfuscating webs 6
  - Restrict Access to a whole Foswiki site 6
  - Authenticate all webs and restrict selected webs 7
  - Authenticate and restrict selected webs only 7
  - Hide control settings 7

Access Control

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

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

An important control consideration

Open, freeform editing is the essence of WikiCulture - what makes Foswiki 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 Foswiki, content is transparently preserved under **revision control**:
  - Edits can be undone by the administrator (per default a member of AdminGroup; 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).

Permissions settings of the webs on this Foswiki site

Web	Sitemap	VIEW		CHANGE		RENAME	
	Listed	DENY	ALLOW	DENY	ALLOW	DENY	ALLOW
System	on				AdminGroup		AdminGroup
Public	on						

Please Note:

- A blank in the the above table may mean *either* the corresponding control is absent or commented out *or* that it has been set to a null value. The two conditions have dramatically different and possibly opposed semantics.
- WikiGuest is the guest account - used by unauthenticated users.
- The web must not deny view to WikiGuest; otherwise, people will not be able to register.

Above table comes from SitePermissions

Authentication vs. Access Control

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

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 UserRegistration. 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 home page with the WikiName of the user is created in the Main web.
- The user is added to the WikiUsers topic.

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

Managing Groups

The following describes the standard Foswiki support for groups. Your local Foswiki may have an alternate group mapping manager installed. Check with your Wiki administrator if you are in doubt.

Groups are defined by group topics located in the **`Main`** web. To create a new group, visit WikiGroups 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 = Main.SomeUser, Main.OtherUser, Main.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 KasabianGroup topic write:

- **`Set ALLOWTOPICCHANGE = Main.KasabianGroup`**

Foswiki has strict formatting rules. Make sure you have three spaces, an asterisk, and an extra space in front of any access control rule.

The Super Admin Group

A number of Foswiki 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 `AdminGroup`. The system administrator may have chosen a different name for this group if your local Foswiki uses an alternate group mapping manager but for simplicity we will use the default name

AdminGroup in the rest of this topic.

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

- `Set GROUP = Main.ElizabethWindsor, Main.TonyBlair`

A member of the Super Admin Group has unrestricted access throughout the Foswiki, 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.

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 Wiki 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 >`

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.

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 >`

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 =`**
  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 you 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 Foswiki. A method to upgrade will be provided. Please read the release notes carefully when you upgrade.

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

Controlling access to attachments

Attachments are referred to directly, and are not normally indirected via Foswiki scripts. This means that the above instructions for access control will *not* apply to attachments. It is possible that someone may inadvertently publicise a URL that they expected to be access-controlled.

The easiest way to apply the same access control rules for attachments as apply to topics is to use the Apache `mod_rewrite` module, and configure your webserver to redirect accesses to attachments to the Foswiki `viewfile` script. For example,

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

RewriteEngine on
RewriteCond %{REQUEST_URI} !^/+foswiki/+pub/+System/+.+
RewriteRule ^/+foswiki/+pub/+([^/]+)((/+([^/]+))+)/+(.+) /foswiki/bin/viewfile/$1/$2?f
```

That way all the controls that apply to the topic also apply to attachments to the topic. Other types of webserver have similar support.

Images embedded in topics will load much slower since each image will be delivered by the `viewfile` script.

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 DefaultPreferences? 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 Foswiki evaluates ALLOW/DENY settings

When deciding whether to grant access, Foswiki 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

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`** setting 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.

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

Restrict Access to a whole Foswiki site

For a firewalled Foswiki, e.g. an intranet wiki or extranet wiki, you want to allow only invited people to access your Foswiki. In this case, enable user authentication with ApacheLogin and lock down access to the whole `bin` and `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.

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 UserAuthentication to be enabled.

1. The simple way is to add this to `WebPreferences` in all webs.
   - **`Set DENYWEBVIEW = WikiGuest?`**
2. **Restrict** view access to selected users and groups. Set one or both of these settings in its WebPreferences topic:
   - **`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.

In rare cases it may be required to authenticate the view script. This can in some cases have a dramatic performance hit because the webserver must re-authenticate for every page view.

1. Set `require valid-user` on your `view` script in .htaccess or the appropriate Apache .conf file. This looks like: `FilesMatch "(attach|edit|manage|rename|save|view|upload|mail|logon|.*auth).*"` (normally `view` is not in that list).

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 UserAuthentication to be enabled.

1. **Restrict** view access to selected users and groups. Set one or both of these settings 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

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
-->
```

Related Topics: AdminDocumentationCategory, UserAuthentication
Edit | Attach | Print version | History: %REVISIONS% | Backlinks | Raw View | More topic actions
Topic revision: r1 - 12 Sep 2009 - 04:10:01 - ProjectContributor

- . System

- Log In

- **Toolbox**
- Users
- Groups
- Index
- Search
- Changes
- Notifications
- RSS Feed
- Statistics
- Preferences

- **User Reference**
- BeginnersStartHere
- TextFormattingRules
- Macros
- FormattedSearch
- QuerySearch
- DocumentGraphics
- SkinBrowser
- InstalledPlugins

- **Admin Maintenance**
- Reference Manual
- AdminToolsCategory
- InterWikis
- ManagingWebs
- SiteTools
- DefaultPreferences
- WebPreferences

- **Categories**
- Admin Documentation
- Admin Tools
- Developer Doc
- User Documentation
- User Tools

- **Webs**
- Public
- System