

Table of Contents

| | |
|---|----------|
| Foswiki Skins..... | 1 |
| Overview..... | 1 |
| Changing the default skin..... | 1 |
| Defining Skins..... | 1 |
| Partial customisation, or adding in new features to an existing skin..... | 3 |
| Settings in Skins..... | 3 |
| Using Cascading Style Sheets..... | 4 |
| Skin parts..... | 4 |
| The "Go" Box and Navigation Box..... | 4 |
| FLASHNOTE Notifications..... | 4 |
| Attachment Tables..... | 5 |
| Packaging and Publishing Skins..... | 5 |
| Browsing Installed Skins..... | 6 |
| Activating Skins..... | 6 |
| Hard-Coded Skins..... | 6 |

Foswiki Skins

Skins overlay regular templates to give different looks and feels to Foswiki screens.

Overview

Foswiki uses skin templates as the basis of all the screens it uses to interact with users. Each screen has an associated template file that contains the basic layout of the screen. This is then filled in by the code to generate what you see in the browser.

Foswiki ships with a default set of template files that give a very basic, CSS-themable, look-and-feel. Foswiki also includes support for skins that can be selected to give different, more sophisticated, look and feel. A default Foswiki installation will usually start up with the PatternSkin already selected. Skins may also be defined by third parties and loaded into a Foswiki installation to give more options. To see how Foswiki looks when **no** skin is selected, view this topic with a non-existent skin.

Topic text is not affected by the choice of skin, though a skin can be defined to use a CSS (Cascading Style Sheet), which can sometimes give a radically different appearance to the text.

Relevant links

- Foswiki:Support.Extension - Index page for support resources related to skins.
- Foswiki:Extensions.SkinPackage - list of all contributed skin packages

See **other types of extensions**: ContributedAddOns, Contribs, Plugins

Changing the default skin

Wiki by default ships with the skin PatternSkin activated. You can set the skin for the whole site, a single web or topic, or for each user individually, by setting the SKIN setting to the name of a skin. If the skin you select doesn't exist, then Wiki will pick up the default templates.

Defining Skins

You may want to define your own skin, for example to comply with corporate web guidelines, or because you have a aesthetic vision that you want to share. There are a couple of places you can start doing this.

The skin templates used for skins are located in the `templates` directory and are named according to the skin: `<scriptname>.<skin>.tmpl`. Skin files may also be defined in Foswiki topics - see SkinTemplates for details.

To start creating a new skin, copy the default skin templates (like `view.tmpl`), or copy an existing skin to use as a base for your own skin. You should only need to copy the files you intend to customise, as Foswiki can be configured to fall back to another skin if a template is not defined in your skin. A custom skin can be as small as one file! Name your files as described above (for example `view.myskin.tmpl`).

If you use PatternSkin as your starting point, and you want to modify the layout, colors or even the templates to suit your own needs, have a look first at the topics PatternSkinCustomization and PatternSkinCssCookbook. These topics also provide practical instructions how to create custom skin template files.

⚠ Note: Don't call your skin `text` or `rss` as these two skin names have reserved meanings, see below at hard-coded meanings.


The following template files are used for Foswiki screens, and are referenced in the Foswiki core code. If a skin doesn't define its own version of a template file, then Foswiki will fall back to the next skin in the skin path, or finally, to the default version of the template file.

(Certain template files are expected to provide certain TMPL:DEFs - these are listed in sub-bullets)

- `addform` - used to select a new form for a topic
- `attachagain` - used when refreshing an existing attachment
- `attachnew` - used when attaching a new file to a topic
- `attachtables` - defines the format of attachments at the bottom of the standard topic view
 - ◆ `ATTACH:files:footer`, `ATTACH:files:header`, `ATTACH:files:row`,
`ATTACH:versions:footer`, `ATTACH:versions:header`,
`ATTACH:versions:row`
- `changeform` - used to change the form in a topic
- `changes` - used by the `changes` script
- `edit` - used for the edit screen
- `form`
- `formtables` - used to defined the format of forms
 - ◆ `FORM:display:footer`, `FORM:display:header`, `FORM:display:row`
- `login` - used for login in when using the `TemplateLoginManager`
 - ◆ `LOG_IN`, `LOG_IN_BANNER`, `LOG_OUT`, `LOGGED_IN_BANNER`, `NEW_USER_NOTE`,
`UNRECOGNISED_USER`
- `moveattachment` - used when moving an attachment
- `oopsaccessdenied` - used to format Access Denied messages
 - ◆ `no_such_topic`, `no_such_web`, `only_group`, `topic_access`
- `oopsattention` - used to format Attention messages
 - ◆ `already_exists`, `bad_email`, `bad_ver_code`, `bad_wikiname`,
`base_web_missing`, `confirm`, `created_web`, `delete_err`,
`invalid_web_color`, `invalid_web_name`, `in_a_group`, `mandatory_field`,
`merge_notice`, `missing_action`, `missing_fields`, `move_err`,
`missing_action`, `no_form_def`, `no_users_to_reset`, `not_a_user`,
`oversized_upload`, `password_changed`, `password_mismatch`,
`problem_adding`, `remove_user_done`, `rename_err`, `rename_not_wikiword`,
`rename_topic_exists`, `rename_web_err`, `rename_web_exists`,
`rename_web_prerequisites`, `reset_bad`, `reset_ok`, `save_error`,
`send_mail_error`, `thanks`, `topic_exists`, `unrecognized_action`,
`upload_name_changed`, `web_creation_error`, `web_exists`, `web_missing`,
`wrong_password`, `zero_size_upload`
- `oopschangelanguage` - used to prompt for a new language when internationalisation is enabled
- `oopsgeneric` - a basic dialog for user information; provides "ok" button only
- `oopslanguagechanged` - used to confirm a new language when internationalisation is enabled
- `oopsleaseconflict` - used to format lease Conflict messages
 - ◆ `lease_active`, `lease_old`
- `preview` - used for previewing edited topics before saving
- `rdiff` - used for viewing topic differences
- `registernotify` - used by the user registration system
- `registernotifyadmin` - used by the user registration system
- `rename` - used when renaming a topic
- `renameconfirm` - used when renaming a topic
- `renamedelete` - used when renaming a topic
- `renameweb` - used when renaming a web

- `renamewebconfirm` - used when renaming a web
- `renamewebdelete` - used when renaming a web
- `searchbookview` - used to format inline search results in book view
- `searchformat` - used to format inline search results
- `search` - used by the `search` CGI script
- `settings`
- `view` - used by the `view` CGI script
- `viewprint` - used to create the printable view

`foswiki.tmpl` is a master template conventionally used by other templates, but not used directly by code.

 **Note:** Make sure templates do not end with a newline. Any newline will expand to an empty `<p />` in the generated html. It will produce invalid html, and may break the page layout.

Partial customisation, or adding in new features to an existing skin

You can use recursion in the `TMPL:INCLUDE` chain (eg `view.tmpl` contains `%TMPL:INCLUDE{"foswiki"}%`, the templating system will include the next SKIN in the skin path. For example, to create a customisation of pattern skin, where you *only* want to remove the edit & WYSIWYG buttons from view page, you create only a `view.yourlocal.tmpl`:

```
%TMPL:INCLUDE{"view"}%
%TMPL:DEF{"edit_topic_link"}%%TMPL:END%
%TMPL:DEF{"edit_wysiwyg_link"}%%TMPL:END%
```

and then set `SKIN=yourlocal,pattern`

Settings in Skins

You can use template directives, ordinary macros, and other predefined settings in your skins. Some commonly used macros in skins:

| Macro: | Expanded to: |
|------------------------------------|--|
| <code>%WEBLOGONAME%</code> | Filename of web logo |
| <code>%WEBLOGOIMG%</code> | Image URL of web logo |
| <code>%WEBLOGOURL%</code> | Link of web logo |
| <code>%WEBLOGOALT%</code> | Alt text of web logo |
| <code>%WIKILOGOURL%</code> | Link of page logo |
| <code>%WIKILOGOIMG%</code> | Image URL of page logo |
| <code>%WIKILOGOALT%</code> | Alt text of page logo |
| <code>%WEBBGCOLOR%</code> | Web-specific background color, defined in the WebPreferences |
| <code>%WIKITOOLNAME%</code> | The name of your Foswiki site |
| <code>%SCRIPTURL%</code> | The script URL of Foswiki |
| <code>%SCRIPTURLPATH%</code> | The script URL path |
| <code>%SCRIPTSUFFIX%</code> | The script suffix, ex: <code>.pl</code> , <code>.cgi</code> |
| <code>%WEB%</code> | The name of the current web. |
| <code>%TOPIC%</code> | The name of the current topic. |
| <code>%WEBTOPICLIST%</code> | Common links of current web, defined in the WebPreferences. It includes a Go box |
| <code>%TEXT%</code> | The topic text, e.g. the content that can be edited |
| <code>%META{"form"}%</code> | DataForm, if any |
| <code>%META{"attachments"}%</code> | FileAttachment table |
| <code>%META{"parent"}%</code> | The topic parent |

| Macro: | Expanded to: |
|--------------------|--|
| %EDITTOPIC% | Edit link |
| %REVTITLE% | The revision title, if any, ex: (r1.6) |
| %REVINFO% | Revision info, ex: r1.6 - 24 Dec 2002 - 08:12 GMT - WikiGuest |
| %WEBCOPYRIGHT% | Copyright notice, defined in the WebPreferences |
| %BROADCASTMESSAGE% | Broadcast message at the beginning of your view template, can be used to alert users of scheduled downtimes; can be set in SitePreferences |

Using Cascading Style Sheets

CSS files are generally attachments to the skin topic that are included in the the skin templates - in the case of PatternSkin in the template `css.pattern.tmpl`.

- General documentation of CSS classes: CascadingStyleSheets
- To see how CSS is used in the default Foswiki skin, see: PatternSkin
- If you write a complete new skin, this is the syntax to use in a template file:

```
<style type='text/css' media='all'>@import url('%PUBURLPATH%/%SYSTEMWEB%/MySkin/mystyle.cs
```

Skin parts

The "Go" Box and Navigation Box

The default skins include a "Go" box, also called "Jump" box, to jump to a topic.

The box also understands URLs, e.g. you can type `http://www.google.com/` to jump to an external web site. The feature is handy if you build a skin that has a select box of frequently used links, like Intranet home, employee database, sales database and such. A little JavaScript gets into action on the `onchange` method of the select tag to fill the selected URL into the "Go" box field, then submits the form.

Here is an example form that has a select box and the "Go" box for illustration purposes. You need to have JavaScript enabled for this to work:

Bare bones header, for demo only

Navigate:

Jump:

Note: Redirect to a URL only works if it is enabled in `configure` (Miscellaneous, `{AllowRedirectUrl}`).

FLASHNOTE Notifications

PatternSkin has a notification message display using the variable `FLASHNOTE`. For example:

- Set `FLASHNOTE` = Skins documentation

See the alert at the top of this topic.

While this feature is not yet used by the system, it might be a good idea to already prepare your skin.

Attachment Tables

Controlling the look and feel of attachment tables is a little bit more complex than for the rest of a skin. By default, the attachment table is a standard Foswiki table, and the look is controlled in the same way as other tables. In a very few cases you may want to change the *content* of the table as well.

The format of standard attachment tables is defined through the use of special *template directives* which by default, are defined in the `attachtables.tmpl` template using the `%TMPL:DEF` directive syntax described in SkinTemplates. These macros are:

| Macro | Description |
|------------------------------------|--|
| <code>ATTACH:files:header</code> | Standard title bar |
| <code>ATTACH:files:row</code> | Standard row |
| <code>ATTACH:files:footer</code> | Footer for all screens |
| <code>ATTACH:files:header:A</code> | Title bar for upload screens, with attributes column |
| <code>ATTACH:files:row:A</code> | Row for upload screen |
| <code>ATTACH:files:footer:A</code> | Footer for all screens |

The format of tables of file versions in the Upload screen can also be changed, using the macros:


| Macro | Description |
|-------------------------------------|--|
| <code>ATTACH:versions:footer</code> | Footer for versions table on upload screen |
| <code>ATTACH:versions:header</code> | Header for versions table on upload screen |
| <code>ATTACH:versions:row</code> | Row format for versions table on upload screen |

The `ATTACH:row` macros are expanded for each file in the attachment table, using the following special tags:

| Tag | Description |
|--------------------------|--|
| <code>%A_URL%</code> | viewfile URL that will recover the file |
| <code>%A_REV%</code> | Revision of this file |
| <code>%A_ICON%</code> | A file icon suitable for representing the attachment content |
| <code>%A_FILE%</code> | The name of the file. To get the 'pub' url of the file, use <code>%PUBURL%/%WEB%/%TOPIC%/%A_FILE%</code> |
| <code>%A_SIZE%</code> | The size of the file |
| <code>%A_DATE%</code> | The date the file was uploaded |
| <code>%A_USER%</code> | The user who uploaded it |
| <code>%A_COMMENT%</code> | The comment they put in when uploading it |
| <code>%A_ATTRS%</code> | The attributes of the file as seen on the upload screen e.g "h" for a hidden file |

Packaging and Publishing Skins

See Foswiki:Extensions/SkinPackagingHowTo and Foswiki:Extensions/SkinDeveloperFAQ

For your own skin you are encouraged to show a small 80x31 pixel  FOSWIKI logo at the bottom of your skin:

```
<a href="http://foswiki.org/"></a>
```

The standard Wiki skins show the logo in the `%WEBCOPYRIGHT%`.

Browsing Installed Skins

You can try out all installed skins in the SkinBrowser.

Activating Skins

Foswiki uses a *skin search path*, which lets you combine skins additively. The skin path is defined using a combination of macros and URL parameters.

Foswiki works by asking for a template for a particular function - for example, 'view'. The detail of how templates are searched for is described in SkinTemplates, but in summary, the templates directory is searched for a file called `view.skin.tpl`, where *skin* is the name of the skin e.g. `pattern`. If no template is found, then the fallback is to use `view.tpl`. Each skin on the path is searched for in turn. For example, if you have set the skin path to `local,pattern` then `view.local.tpl` will be searched for first, then `view.pattern.tpl` and finally `view.tpl`.

The basic skin is defined by a `SKIN` setting:

- Set `SKIN = catskin, bearskin`

You can also add a parameter to the URL, such as `?skin=catskin,bearskin`:

- <https://wiki.iac.ethz.ch/System/Skins?skin=catskin,bearskin>

Setting `SKIN` (or the `?skin` parameter in the URL) replaces the existing skin path setting, for the current page only. You can also *extend* the existing skin path as well, using *covers*.

- Set `COVER = ruskin`

This pushes a different skin to the front of the skin search path (so for our example above, that final skin path will be `ruskin, catskin, bearskin`). There is also an equivalent `cover` URL parameter. The difference between setting `SKIN` vs. `COVER` is that if the chosen template is not found (e.g., for included templates), `SKIN` will fall back onto the next skin in line, or the default skin, if only one skin was present, while `COVER` will always fall back onto the current skin.

An example would be invoking the printable mode, which is achieved by applying `?cover=print`. The `view.print.tpl` simply invokes the `viewprint` template for the current skin which then can appropriately include all other used templates for the current skin. Where the printable mode be applied by using `SKIN`, all skins would have the same printable appearance.

The full skin path is built up as follows: `SKIN` setting (or `?skin` if it is set), then `COVER` setting is added, then `?cover`.

Hard-Coded Skins

The `text` skin is reserved for Foswiki internal use.










Skin names starting with `rss` also have a special meaning; if one or more of the skins in the skin path starts with 'rss' then 8-bit characters will be encoded as XML entities in the output, and the `content-type` header will be forced to `text/xml`.

Related Topics: SkinBrowser, AdminDocumentationCategory, DeveloperDocumentationCategory

- ☐ 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

-
-



Copyright © by the contributing authors. All material on this site is the property of the contributing authors.

Ideas, requests, problems regarding Wiki? Send feedback