

Table of Contents

TWiki Templates.....	1
Overview.....	1
Master Templates.....	1
How Template Directives Work.....	1
Finding Templates.....	2
TMPL:INCLUDE recursion for piecewise customisation, or mixing in new features.....	3
Default master template.....	3
Template Topics.....	4
Variable Expansion.....	4
1. Default variable expansion.....	4
2. Preventing variable expansion.....	5
3. Control over variable expansion.....	5
Specifying a Form.....	6
Automatically Generated Topic Names.....	6
Template Topics in Action.....	6
Using Absolute vs Relative URLs in Templates.....	7

TWiki Templates


Definition of the templates used to render all HTML pages displayed in TWiki

Overview

Templates are plain text with embedded *template directives* that tell TWiki how to compose blocks of text together, to create something new.

There are two types of template:

- **Master Templates:** Define the HTML used to display TWiki pages.
- **Template Topics:** Define default text when you create a new topic

 **Tip:** TWiki:TWiki.TWikiTemplatesSupplement[?] on TWiki.org has supplemental documentation on TWiki templates.



Master Templates

TWiki uses master templates when composing the output from all actions, like topic view, edit, and preview. This allows you to change the look and feel of all pages by editing just a few template files.

Master templates are also used in the definition of TWikiSkins.

Master templates are stored as text files with the extension `.tmpl`. They are usually HTML with embedded *template directives*. The directives are expanded when TWiki wants to generate a user interface screen.

How Template Directives Work

- Directives are of the form `%TMPL:<key>%` and `%TMPL:<key>{"attr"}%`.
- Directives:
 - ◆ `%TMPL:INCLUDE{"file"}%`: Includes a template file. The file is found as described below.
 - ◆ `%TMPL:DEF{"block"}%`: Define a block. **All** text between this and the next `%TMPL:END%` directive is removed and saved for later use with `%TMPL:P`.
 - ◆ `%TMPL:END%`: Ends a block definition.
 - ◆ `%TMPL:P{"var"}%`: Includes a previously defined block.
 - ◆ `%{... }%`: is a comment.
- Two-pass processing lets you use a variable before or after declaring it.
- Templates and TWikiSkins work transparently and interchangeably. For example, you can create a skin that overloads only the `twiki.tmpl` master template, like `twiki.print.tmpl`, that redefines the header and footer.
-  Use of template directives is optional: templates work without them.
-  **NOTE:** Template directives work only for templates: they do not get processed in normal topic text.

`TMPL:P` also supports simple parameters. For example, given the definition `%TMPL:DEF{"x"}%
x%P%z%TMPL:END%` then `%TMPL:P{"x" P="y"}%` will expand to `xyz`.

Note that parameters can simply be ignored; for example, `%TMPL:P{"x"}%` will expand to `x%P%z`.

Any alphanumeric characters can be used in parameter names. You are highly recommended to use parameter names that cannot be confused with TWikiVariables.

Note that three parameter names, `context`, `then` and `else` are **reserved**. They are used to support a limited form of "if" condition that you can use to select which of two templates to use, based on a *context identifier*:

```
%TMPL:DEF{"link_inactive"}%<input type="button" disabled value="Link">%TMPL:END%
%TMPL:DEF{"link_active"}%<input type="button" onclick="link()" value="Link" />%TMPL:END%
%TMPL:P{context="inactive" then="inactive_link" else="active_link"}% for %CONTEXT%
```

When the "inactive" context is set, then this will expand the "link_inactive" template; otherwise it will expand the "link_active" template. See IfStatements for details of supported context identifiers.

Finding Templates



The master templates shipped with a twiki release are stored in the `twiki/templates` directory. As an example, `twiki/templates/view.tmpl` is the default template file for the `twiki/bin/view` script.

You can save templates in other directories as long as they are listed in the `{TemplatePath}` configuration setting. The `{TemplatePath}` is defined in the Miscellaneous section of the configure page.

You can also save templates in user topics (*IF* there is no possible template match in the `templates` directory). The `{TemplatePath}` configuration setting defines which topics will be accepted as templates.

Templates that are included with an explicit `.tmpl` extension are looked for only in the `templates/` directory. For instance `%TMPL:INCLUDE{"example.tmpl"}%` will only return `templates/example.tmpl`, regardless of `{TemplatePath}` and `SKIN` settings.

The out-of-the-box setting of `{TemplatePath}` supports the following search order to determine which template file or topic to use for a particular script or `%TMPL:INCLUDE{"script"}%` statement. The *skin path* is set as described in TWikiSkins.

1. `templates/web/script.skin.tmpl` for each skin on the skin path
 - ◆  this usage is supported **for compatibility only** and is **deprecated**. Store web-specific templates in TWiki topics instead.
2. `templates/script.skin.tmpl` for each skin on the skin path
3. `templates/web/script.tmpl`
 - ◆  this usage is supported **for compatibility only** and is **deprecated**. Store web-specific templates in TWiki topics instead.
4. `templates/script.tmpl`
5. The TWiki topic `aweb.atopic` if the template name can be parsed into `aweb.atopic`
6. The TWiki topic `web.SkinSkinScriptTemplate` for each skin on the skin path
7. The TWiki topic `web.ScriptTemplate`
8. The TWiki topic `%SYSTEMWEB%.SkinSkinScriptTemplate` for each skin on the skin path
9. The TWiki topic `%SYSTEMWEB%.ScriptTemplate`

Legend:

- `script` refers to the script name, e.g `view`, `edit`
- `Script` refers to the same, but with the first character capitalized, e.g `View`
- `skin` refers to a skin name, e.g `dragon`, `pattern`. All skins are checked at each stage, in the order they appear in the skin path.
- `Skin` refers to the same, but with the first character capitalized, e.g `Dragon`
- `web` refers to the current web


For example, the `example` template file will be searched for in the following places, when the current web is `Thisweb` and the skin path is `print, pattern`:

1. `templates/Thisweb/example.print.tpl` *deprecated; don't rely on it*
2. `templates/Thisweb/example.pattern.tpl` *deprecated; don't rely on it*
3. `templates/example.print.tpl`
4. `templates/example.pattern.tpl`
5. `templates/Thisweb/example.tpl` *deprecated; don't rely on it*
6. `templates/example.tpl`
7. `Thisweb.PrintSkinExampleTemplate`
8. `Thisweb.PatternSkinExampleTemplate`
9. `Thisweb.ExampleTemplate`
10. `TWiki.PrintSkinExampleTemplate`
11. `TWiki.PatternSkinExampleTemplate`
12. `TWiki.ExampleTemplate`

Template names are usually derived from the name of the currently executing script; however it is also possible to override these settings in the `view` and `edit` scripts, for example when a topic-specific template is required. Two preference variables can be used to override the templates used:

- `VIEW_TEMPLATE` sets the template to be used for viewing a topic
- `EDIT_TEMPLATE` sets the template for editing a topic.

If these preferences are set locally (using *Local* instead of *Set*) for a topic, in `WebPreferences`, in `Main.TWikiPreferences`, or `TWiki.TWikiPreferences` (using *Set*), the indicated templates will be chosen for `view` and `edit` respectively. The template search order is as specified above.

 **Tip:** If you want to override existing templates, without having to worry that your changes will get overwritten by the next TWiki update, change the `{TemplatePath}` so that another directory, such as the `%USERSWEB%` appears at the front. You can then put your own templates into that directory or web and these will override the standard templates. (Note that such will increase the lookup time for templates by searching your directory first.)

TMPL:INCLUDE recursion for piecwise customisation, or mixing in new features

If there is recursion in the `TMPL:INCLUDE` chain (eg `twiki.classic.tpl` contains `%TMPL:INCLUDE{"twiki"}%`, the templating system will include the next `twiki.SKIN` in the skin path. For example, to create a customisation of pattern skin, where you *only* want to over-ride the breadcrumbs for the view script, you can create only a `view.yourlocal.tpl`:

```
%TMPL:INCLUDE{"view"}%
%TMPL:DEF{"breadcrumb"}% We don't want any crumbs %TMPL:END%
```

and then set `SKIN=yourlocal,pattern`

The default `{TemplatePath}` will not give you the desired result if you put these statements in the topic `Thisweb.YourlocalSkinViewTemplate`. The default `{TemplatePath}` will resolve the request to the `template/view.pattern.tpl`, before it gets to the `Thisweb.YourlocalSkinViewTemplate` resolution. You can make it work by prefixing the `{TemplatePath}` with: `$web.YourlocalSkin$nameTemplate`.

Default master template

`twiki.tpl` is the default master template. It defines the following sections.

Template variable:	Defines:
--------------------	----------

%TMPL:DEF{"sep"}%	" " separator
%TMPL:DEF{"htmldoctype"}%	Start of all HTML pages
%TMPL:DEF{"standardheader"}%	Standard header (ex: view, index, search)
%TMPL:DEF{"simpleheader"}%	Simple header with reduced links (ex: edit, attach, oops)
%TMPL:DEF{"standardfooter"}%	Footer, excluding revision and copyright parts

Template Topics

The second type of template in TWiki are template topics. Template topics define the default text for new topics. There are four types of template topic:

Topic Name:	What it is:
WebTopicViewTemplate	Alert page shown when you try to view a nonexistent topic. This page is usually used as a prompt to help you create a new topic.
WebTopicNonWikiTemplate	Alert page shown when you try to view a nonexistent topic with a non-WikiName. Again, this page is used as a prompt to help you create the new topic.
WebTopicEditTemplate	Default text used in a new topic.
<MyCustomNamed>Template	Whenever you create a topic ending in the word "Template", it is automatically added to the list of available templates in the "Use Template" drop down field on the WebCreateNewTopic page.

When you create a new topic using the `edit` script, TWiki locates a topic to use as a content template according to the following search order:

1. A topic name specified by the `templatetopic` CGI parameter
 - ◆ if no web is specified, the current web is searched first and then the TWiki web
2. WebTopicEditTemplate in the current web
3. WebTopicEditTemplate in the Main web
4. WebTopicEditTemplate in the TWiki web

Variable Expansion

TWikiVariables located in template topics get expanded as follows when a new topic is created.

1. Default variable expansion

The following variables used in a template topic automatically get expanded when new topic is created based on it:

Variable:	Description:
%DATE%	Signature format date. See VarDATE
%GMTIME%	Date/time. See VarGMTIME
%GMTIME{...}%	Formatted date/time. See VarGMTIME2
%NOP%	A no-operation variable that gets removed. Useful to prevent a SEARCH from hitting an edit template topic; also useful to escape a variable, such as %URLPA%NOP%RAM{...}% escaping URLPARAM
%STARTSECTION{type="templateonly"}% ... %ENDSECTION{type="templateonly"}%	Text that gets removed when a new topic based on the template is created. See notes below.
%SERVERTIME%	Date/time. See VarSERVERTIME

<code>%SERVERTIME{...}%</code>	Formatted date/time. See <code>VarSERVERTIME2</code>
<code>%USERNAME%</code>	Login name of user who is instantiating the new topic, e.g. <code>guest</code>
<code>%URLPARAM{"name"}%</code>	Value of a named URL parameter. See <code>VarURLPARAM</code> .
<code>%WIKINAME%</code>	WikiName of user who is instantiating the new topic, e.g. <code>TWikiGuest</code>
<code>%WIKIUSERNAME%</code>	User name of user who is instantiating the new topic, e.g. <code>Main.TWikiGuest</code>

2. Preventing variable expansion

In a template topic, embed text that you *do not* want expanded inside a

`%STARTSECTION{type="templateonly"}% ... %ENDSECTION{type="templateonly"}%` section. For example, you might want to write this in the template topic:

```
%STARTSECTION{type="templateonly"}%
This template can only be changed by:
  * Set ALLOWTOPICCHANGE = Main.TWikiAdminGroup
%ENDSECTION{type="templateonly"}%
```

This will restrict who can edit the template topic, but will be removed when a new topic based on that template topic is created.

`%NOP%` can be used to prevent expansion of TWiki variables that would otherwise be expanded during topic creation. For example, escape `%SERVERTIME%` with `%SER%NOP%VERTIME%`.

3. Control over variable expansion

You can forcefully expand TWikiVariables by placing them inside a `type="expandvariables"` section in the template topic, such as:

```
%STARTSECTION{ type="expandvariables" }% ... %ENDSECTION{ type="expandvariables" }%
```

Example:

If you have the following content in a template topic:

```
%STARTSECTION{ type="expandvariables" }%
  * %SYSTEMWEB%.ATasteOfTWiki - view a short introductory presentation on TWiki for beginners
  * %SYSTEMWEB%.WelcomeGuest - starting points on TWiki
  * %SYSTEMWEB%.TWikiUsersGuide - complete TWiki documentation
  * Sandbox.%HOMETOPIC% - try out TWiki on your own
  * Sandbox.%TOPIC%Sandbox - just for me
%ENDSECTION{ type="expandvariables" }%
```

you will get this raw text in new topics based on that template topic:

```
* TWiki.ATasteOfTWiki - view a short introductory presentation on TWiki for beginners
* TWiki.WelcomeGuest - starting points on TWiki
* TWiki.TWikiUsersGuide - complete TWiki documentation
* Sandbox.WebHome - try out TWiki on your own
* Sandbox.JimmyNeutronSandbox - just for me
```

Specifying a Form

When you create a new topic based on a template, you often want the new topic to have a form attached to it. You can attach a form to the template topic, in which case it will be copied into the new topic.

Sometimes this isn't quite what you want, as it copies all the existing data from the template topic into the new topic. To avoid this and use the default values specified in the form definition instead, you can use the `formtemplate` CGI parameter to the `edit` script to specify the name of a form to attach.

See TWikiScripts for information about all the other parameters to `edit`.

Automatically Generated Topic Names

For TWiki applications it is useful to be able to automatically generate unique topicnames, such as BugID0001, BugID0002, etc. You can add `AUTOINC<n>` to the topic name in the edit and save scripts, and it will be replaced with an auto-incremented number on topic save. `<n>` is a number starting from 0, and may include leading zeros. Leading zeros are used to zero-pad numbers so that auto-incremented topic names can sort properly. Deleted topics are not re-used to ensure uniqueness of topic names. That is, the auto-incremented number is always higher than the existing ones, even if there are gaps in the number sequence.

Examples:

- BugAUTOINC0 - creates topic names Bug0, Bug1, Bug2, ... (does not sort properly)
- ItemAUTOINC0000 - creates topic names Item0000, Item0001, Item0002, ... (sorts properly up to 9999)
- DocIDAUTOINC10001 - start with DocID10001, DocID10002, ... (sorts properly up to 99999; auto-links)

Example link to create a new topic:

```
[[%SCRIPTURLPATH{edit}%/%WEB%/BugIDAUTOINC00001?templatetopic=BugTemplate;topicparent=%TOPIC%;t=%
```

Template Topics in Action

Here is an example for creating new topics (in the Sandbox web) based on a specific template topic and form:

- New example topic:

The above form asks for a topic name. A hidden input tag named `templatetopic` specifies ExampleTopicTemplate as the template topic to use. Here is the HTML source of the form:

```
<form name="new" action="%SCRIPTURLPATH{edit}%/Sandbox/">
  * New example topic:
  <input type="text" name="topic" value="ExampleTopicAUTOINC0001" size="30" />
  <input type="hidden" name="templatetopic" value="ExampleTopicTemplate" />
  <input type="hidden" name="topicparent" value="%TOPIC%" />
  <input type="hidden" name="onlywikiname" value="on" />
  <input type="hidden" name="onlynewtopic" value="on" />
  <input type="submit" class="twikiSubmit" value="Create" />
</form>
```

Note: You can create a topic in one step, without going through the edit screen. To do that, specify the save script instead of the `edit` script in the form action. When you specify the save script you *have* to use the "post" method. Example:

```
<form name="new" action="%SCRIPTURLPATH{save}%/Sandbox/" method="post">
  ...
</form>
```

T The `edit` and `save` scripts understand many more parameters, see [TWikiScripts#edit](#) and [TWikiScripts#save](#) for details.

TIP: You can use the `%WIKIUSERNAME%` and `%DATE%` variables in your topic templates to include the signature of the person creating a new topic. The variables are expanded into fixed text when a new topic is created. The standard signature is:

```
-- %WIKIUSERNAME% - %DATE%
```

Using Absolute vs Relative URLs in Templates

When you use TWikiVariables such as `%PUBURL%` and `%PUBURLPATH%` in templates you should be aware that using `%PUBURL%` instead of `%PUBURLPATH%` puts absolute URLs in the produced HTML. This means that when a user saves a TWiki page in HTML and emails the file to someone outside a company firewall, the receiver has a severe problem viewing it. It is therefore recommended always to use the `%PUBURLPATH%` to refer to images, CSS, Javascript files etc so links become relative. This way browsers just give up right away and show a usable html file.

Related Topics: [TWikiSkins](#), [TWikiForms](#), [TWikiScripts](#), [DeveloperDocumentationCategory](#), [AdminDocumentationCategory](#)

-- **Contributors:** [TWiki:Main.PeterThoeny](#), [TWiki:Main.MikeMannix](#), [TWiki:Main.DavidLeBlanc](#), [TWiki:Main.CrawfordCurrie](#), [TWiki:Main.SopanShewale](#)

This topic: [TWiki21Nov > TWikiTemplates](#)

Topic revision: r35 - 2010-09-20 - [TWikiContributor](#)



Copyright &© 2008-2022 by the contributing authors. All material on this collaboration platform is the property of the contributing authors.

or Ideas, requests, problems regarding TWiki? use [Discourse](#) or [Send feedback](#)