Table of Contents

Comment Plugin **1**
- Features 1
- Syntax 1
  - Positioning the comment 2
    - Location relative to %COMMENT tag 2
    - Location relative to an anchor 2
    - Location relative to an arbitrary text string 2
  - Default templates 2
- Customisation 3
  - Defining custom templates 3
  - Customisation example 4
  - The PROMPT template 4
    - Providing attribute values 4
    - Special macros 5
    - Customisation example with custom form template 5
  - The OUTPUT template 6
- Settings 6
- Plugin Installation Instructions 7
- Plugin Info 7

Comment Plugin

Comment Plugin lets users quickly post comments to a page without an edit/preview/save cycle.

Related topics: CommentPluginTemplates, CommentPluginExamples

Features

Inserts an edit box into the page that allows users to type in and save comments. Comments can be made

- in different formats (as defined by a template),
- in both forward and reverse chronological order,
- signed or unsigned, dated or undated (as defined by a template),
- in other topics, or other positions within the current topic.

Syntax

Write `%COMMENT{`*`attributes`*`}%` anywhere in a topic.

- A `%COMMENT%` without parameters shows a simple text box.
- The following standard attributes are recognized

Name	Description	Default
`type`	This is the name of the template to use for this comment. Comment templates are defined in a Foswiki template - see Customisation, below. If this attribute is not defined, the type is whatever is defined by COMMENTPLUGIN_DEFAULT_TYPE, either in this topic or in your WebPreferences.	`below`
`default`	Default text to put into the textarea of the prompt.	
`target`	Name of the topic to add the comment to	the current topic
`location`	Regular expression specifying the comment location in the target topic. Read *carefully* the CommentPlugin documentation!	
`mode`	For compatibility with older versions only, synonymous with `type`	
`nonotify`	Set to "on" to disable change notification for target topics	`off`
`noform`	Set to "on" to disable the automatic form that encloses your comment block - *remember* to insert `<form>` tags yourself! See CommentPluginExamples#noform for an example.	`off`
`nopost`	Set to "on" to disable insertion of the posted text into the topic.	`off`
`remove`	Set to "on" to remove the comment prompt after the first time it is clicked.	`off`
`button`	Button label text	`Add comment`

(See also additional attributes)

Positioning the comment

%COMMENT supports several ways to specify *where* a comment should be inserted in the target topic. This is referred to as the *location* of the comment.

Location relative to %COMMENT tag

The default *location* is the %COMMENT tag itself. For example:

```
%COMMENT{type="below"}%
```

will add comments in the current topic, directly below the %COMMENT tag.

Location relative to an anchor

The target attribute may specify a web, and may also specify an anchor within the target topic; for example,

```
%COMMENT{type="above" target="%MAINWEB%.PersonalRemarks#InsertHere"}%
```

This uses a standard in-topic anchor as the insertion location. See TextFormattingRules for more about Foswiki anchors.

Location relative to an arbitrary text string

Getting more sophisticated, you can also specify a regular expression for the target location using the location parameter. The target topic is searched for the regular expression, and the comment inserted relative to the string that the search matched. For example,

```
%COMMENT{type="above" location="Flights of Fancy"}%
```

will place comments above the first occurence of the string Flights of Fancy in the current topic.

Warning of course, if a user's comment contains the string "Flights of Fancy" they may and up *changing the location* for the next comment! Also, if you use a tag in the location, then you've just inserted another tag in the page that contains the %COMMENT! So be very careful how you specify the RE for location. Note that the RE is matched using perl "multiple line" mode, so ^ and $ match the start of a line and the end of a line respectively.
Also note that you cannot have the text location=" just before the location.

I look forward to someone leveraging this feature to create - for example - threaded conversations using %COMMENT.

If you specify an anchor *and* a location, the anchor will be ignored.

Default templates

Templates are used to define the "comment style" i.e. how comments appear in the page. The default is to add comments in "Blog like" style using bulleted lists, with the most recent comment at the top, but many other styles are available such as tables or Wiki thread mode comments. It is easy to define your own customer

styles as well.

A set of default comment templates are shipped with the plugin - see also CommentPluginTemplates:

Template type	Description
`top`	Comments, signed and dated (server time), added at top of the topic (the anchor is ignored)
`bottom`	Comments, signed and dated (server time), added at end of the target topic (the anchor is ignored)
`above`	Comments, signed and dated (server time), added immediately before the target anchor, or the `%COMMENT` if no anchor is specified
`below`	Comments, signed and dated (server time), added immediately below the target anchor, or the `%COMMENT` if no anchor is specified
`belowthreadmode`	Comments, signed and dated, added recurse after comment box
`threadmode`	Wiki thread mode comment, signed and dated (server time)
`tableprepend`	Comments, signed and dated (server time), formatted as an HTML table row, added below the anchor (which must be in an HTML <table>)
`tableappend`	Comments, signed and dated (server time), formatted as an HTML table row, added above the anchor (which must be in an HTML <table>)
`action`	Action added to action table directly above comment box (see Plugin Installation Instructions below for important notes)
`table`	Tablerows adding on end
`toctalk`	Talk using TOC adding on end
`bookmark`	Create a list of annotated bookmarks
`return`	Post to a different topic and return

Your local installation may add more template types as well - see Customisation, below.

Customisation

Customisation of the comment plugin requires

- familiarity with HTML forms
- some familiarity with the Skin Templates.

To define a comment type, you have to provide two simple template definitions in the template file; one for the prompt box, and one for the generated output. If we have a template type "mytype", these are named `PROMPT:mytype` and `OUTPUT:mytype` respectively. See `comments.tmpl` in the templates directory for examples.

The plugin picks up these template definitions from a standard template file, `templates/comments.tmpl`. This allows different templates to be defined for different Foswiki skins.

Defining custom templates

By default, `templates/comments.tmpl` includes the topic CommentPluginTemplate, which contains all the shipped standard templates and in turn includes System.UserCommentsTemplate that can include non-standard customisations.

This allows for several levels of customisation:

1. To **override all default templates, everywhere**, change `comments.tmpl` to include a different topic (this customisation will be lost next time you upgrade, though).
2. To add **site-wide local template customisations**, add them to UserCommentsTemplate? (create if it does not exist yet). You can redefine the standard templates here if you want, and your definitions will override the standard definitions.
3. To override templates on a **web-by-web basis**, add a topic `UserCommentsTemplate` to the web (this will replace System.UserCommentsTemplate)
4. To override templates **for a specific skin**, add them to System.UserComments<Skin>Template (where <Skin> is the name of the skin with the first letter capitalised, e.g. Pattern)

You can also define a **comment template in a topic**, by passing the topic location with `templatetopic`. For example:

```
 %COMMENT{type="blogpost" templatetopic="BlogPostCommentTemplate" target="%TOPIC%" button="Add co
```

`templatetopic` accepts `topic` or `web.topic` syntax. See an example in CommentPluginExamples:templatetopic.

Templates are picked up by following the standard rules for locating template files. Note that you can use `%TMPL:INCLUDE` to include other files of templates.

Customisation example

Provide both a `PROMPT` and an `OUTPUT` definition:

```
%TMPL:DEF{PROMPT:myComment}%%TMPL:P{promptbox}%%TMPL:END%
%TMPL:DEF{OUTPUT:myComment}%%TMPL:P{outputoneliner}%%POS:TOP%
%TMPL:END%
```

Call your custom comment with:

```
%COMMENT{type="myComment"}%
```

The `PROMPT` template

The `PROMPT` template defines the contents of an HTML form that is used to capture the comment. This form invokes the comment generator when submitted. Parameters to the comment generator are defined using standard HTML input fields, such as `input`, `textarea` and `select`. The user enters values for these parameters, and these are then available when the `OUTPUT` template is expanded, in the form of `%URLPARAM%`s.

Only the input fields of the form need be defined. The plugin automatically generates the `<form>` and `</form>` tags, unless you specify `noform="on"`, in which case you have to provide them yourself. **Note** that you must define a "submit" button if you want the form to work!

Providing attribute values

If an attribute is given to the `%COMMENT` tag that is not one of the standard attributes, then that attribute is taken as the name of a parameter to be expanded in the `PROMPT` template. Expressions in the template of the form `%`*param*`|`*default*`%` (e.g. `%rows|3%`, `%button|Push me%`) are expanded to the values given in the

%COMMENT. For example, if the PROMPT template 'example' contains:

```
<textarea rows=%rows|3% cols="%cols|50%" value="%tval|Rubbish%">
```

and the %COMMENT tag is:

```
%COMMENT{type="example" cols="75"}%
```

then the template will be expanded as

```
<textarea rows="3" cols="75" value="Rubbish">
```

Special macros

As well as support for all the usual macros in templates, the following special macros are supported in the PROMPT definition:

Macro	Description
%DISABLED%	Set to 'disabled' when you cannot comment (e.g. in preview mode).
%MESSAGE%	The text specified by default. This may be overridden by a helpful message when the prompt is DISABLED.

EXPERT Note that when a comment is saved, the save script is invoked on the target topic, with a number of parameters provided by the comment form. Normally the CommentPlugin will provide these fields in the form, but experts can also provide the fields themselves in order to get finer control over what is submitted, or you might want to define your own HTML forms that do comment submission. The parameters that the CommentPlugin recognises are as follows:

CGI parameter	Description
comment_action	Must be save to get the CommentPlugin to perform
comment_type	Type of the OUTPUT template
comment_index	Zero-based index of the %COMMENT in the source topic. Used to place a post relative to an existing %COMMENT.
comment_anchor	Anchor taken from the target spec
comment_location	As passed to %COMMENT
comment_nonotify	As passed to %COMMENT
comment_remove	Zero-based index of a %COMMENT to remove from the target topic
comment_nopost	As passed to %COMMENT
comment_templatetopic	As passed to %COMMENT

Note that comment_location overrides comment_anchor, and both override comment_index. Example, shows an "I Approve" button that adds your approval signature to the end of the topic:

```
<form method="post" action="%SCRIPTURL{save}%/%WEB%/%TOPIC%">
<input type="submit" value="I Approve" />
<input type="hidden" name="comment_action" value="save" />
<input type="hidden" name="comment_type" value="bottom" />
<input type="hidden" name="comment" value="I Approve" />
</form>
```

Customisation example with custom form template

Write a custom form in a topic.

- In the form set the location of the prompt with `%COMMENTPROMPT%`; the prompt will be positioned here.
- In %COMMENT use parameter `noform="on"`
- In %COMMENT use parameter `templatetopic` to point to the topic with the form template

Example form:

```
%TMPL:DEF{FORM:example}%
<form method="post" action="%SCRIPTURL{save}%/%BASEWEB%/%BASETOPIC%" enctype="application/
<input type="hidden" name="redirectto" value="%BASEWEB%.%BASETOPIC%" />
%COMMENTPROMPT%
</form>
%TMPL:END%
```

Example comment:

```
%COMMENT{noform="on" type="example" templatetopic="Sandbox.CommentPluginTemplateExample" t
```

The `OUTPUT` template

The `OUTPUT` template defines the format for the text that actually gets embedded into the topic. All the usual macros are available in the `PROMPT` definition, but note that they get expanded *when the comment is inserted in the text*, so time, date and username will refer to the time and date when the comment was made, and the user who made it.

There are also four position tags that are used to indicate where the comment should be placed, relative to the `location` defined in the `%COMMENT` tag:

`%POS:TOP%`	If present, comments will be inserted **at the top of the topic** i.e. before any other text
`%POS:BOTTOM%`	If present, comments will be inserted **at the end of the topic** i.e. after all existing text
`%POS:BEFORE%`	If present, comments will be inserted **immediately before the `%COMMENT%` tag**
`%POS:AFTER%`	If present, comments will be inserted **immediately after the `%COMMENT%` tag**

Note that these position tags are obviously mutually exclusive. If you define more than one, the result is undefined. If none is present, the default is taken from the plugin setting `DEFAULT_TYPE`

`%COMMENTPROMPT%`	Use with a custom form. If present, the comment prompt will be positioned here.

All the usual macros that can be used in a topic template can also be used in an `OUTPUT` template.

Settings

Two preference settings are recognised by the CommentPlugin:

Preference	Default	Description
`%COMMENTPLUGIN_TEMPLATES%`	`comments`	Name of template file in the 'templates' directory that contains the comment templates. The default 'comments.tmpl' automatically includes user templates from CommentPluginTemplate, which in turn includes UserCommentsTemplate?.
`%COMMENTPLUGIN_DEFAULT_TYPE%`	`above`	Default template type

These can be set in %USERSWEB%.SitePreferences, in WebPreferences, or in individual topics.

#Installation

Plugin Installation Instructions

- This plugin is pre-installed in most releases. However if you need to upgrade the plugin for any reason:
- Download the archive file from the Extensions web (see below)
- Unpack the archive in your Foswiki installation directory.
    - ♦ You may need to correct file permissions
- Run **`CommentPlugin_installer`** to automatically check and install other modules that this module depends on, and enable the plugin.
- Alternatively,
    - ♦ Manually resolve the dependencies listed below. None
- Use `configure` to enable the plugin

Note that if you want to use the `action` template then you must also:

1. Install the Foswiki:Extensions/ActionTrackerPlugin;
2. Put the CommentPlugin **before** the ActionTrackerPlugin in the `{PluginsOrder}` configuration option (in `configure`)

Plugin Info

Another great extension from the **WikiRing** - working together to improve your wiki experience!

Plugin Author:	Foswiki:Main.CrawfordCurrie http://www.c-dot.co.uk inspired by the work of TWiki:Main.DavidWeller and TWiki:Main.PeterMasiar
Copyright:	© 2004, Foswiki:Main.CrawfordCurrie © 2004-2010 Foswiki:System.ProjectContributor
License:	GPL (GNU General Public License)
Release:	24 Feb 2010
Version:	6512 (2010-02-24)
Change History:	
24 Feb 2010	Foswikitask:Item8611 - block comments resulting from calls to saveTopic from within an afterSaveHandler
12 Sep 2009	Foswikitask:Item8269 - Targeting anchors broken by Item727 is fixed.
04 Jun 2009	Foswikitask:Item1668 - The action template used with ActionTrackerPlugin now uses new syntax ending with %ENDACTION. This makes each action item appear on a new line. Additionally new lines are now correctly saved as html br tags and not as html encoded br tag Foswikitask:Item1640 - Templates shipped with the plugin no longer encodes the user date entered when the date is saved so that it is possible for the user to use macros.
15 Jan 2008	Foswikitask:Item727 - data loss issue when missing anchor or location
13 Jan 2009	Foswikitask:Item736 - Make CommentPlugin aware of CompareRevisionsAddOn? bin script.
16 Dec 2008	Foswiki version
06 Mar 2002	initial commit
Home:	http://foswiki.org/Extensions/CommentPlugin
Support:	http://foswiki.org/Support/CommentPlugin

Related Topics: SitePreferences, Plugins

Topic revision: r1 - 04 Feb 2007 - 15:05:47 - 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

Copyright © by the contributing authors. All material on this site is the property of the contributing authors.
Ideas, requests, problems regarding Wiki? Send feedback