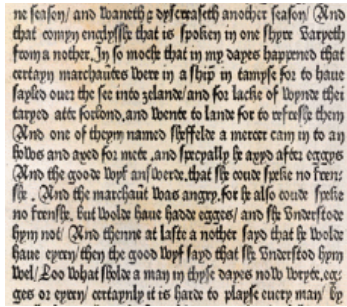


---+!! PublishContrib **Generates a static view of a web, as HTML files on disc, or as a PDF, or as a zip or tgz archive file, or by uploading directly to an FTP server.**

+

*Previously known as GenHTMLAddOn, and then PublishAddOn, this is the **original** publishing extension for TWiki.*

NOTE: This extension is designed to work with TWiki 4.0 and later. You can use revision 1 of the attached zip file if you want to use the extension with an earlier version of TWiki, but it is missing many features and bugfixes.



PublishContrib provides support for the generation of stand-alone HTML from a TWiki web. It will generate fully rendered versions of a set of TWiki pages together with any attached files.

When TWiki generates a view, it does so dynamically i.e. there's a CGI script that runs, processes some files, and generates HTML that is displayed by the browser. There are circumstances in which this may not be desirable or even possible. For example:

1. TWiki is used to create documentation which has to be bundled into a product release
2. Published versions of TWiki pages must be read-only
3. The TWiki server is inaccessible to the audience (e.g. on the other side of a corporate firewall)

Features

- All standard TWiki tags are interpreted
- Plugins are called
- Unresolved links to non-existent topics are silently ignored
- Topic links internal to the TWiki are translated to relative links
- Powerful support for choosing what content gets published
- Any links to the 'pub' areas of topics in the web are automatically resolved and the referenced files copied
- Any links to images outside the TWiki are resolved, and the image is stored in the output (requires LWP)
- Output in HTML or PDF. HTML can be compressed in different archive formats.
- Full support for hierarchical webs
- Multiple instances (e.g. dev, test, prod) can be specified
- Special output format specific templates (such as viewpdf) can be used
- Able to publish HTML and referenced files directly to a remote server via ftp * Complete history of what was published, and when!

Usage

Publish Form

The easiest way to publish a web is from this topic, by filling in the following form.

The output is generated in a directory designated during installation. The progress messages printed during documentation generation tell you exactly where the output is, and you can use the publishers control interface to manage your published output.

Publishing is a controlled process; before you can publish, you have to have VIEW access to the topics you want to publish, and CHANGE access to the publishing history topic.

You can also create a permanent topic in a web to help with the publishing process.

Choose what to publish

Web to publish

Comma-separated list of wildcard patterns that match the names of topics to **include**. Use * for all topics. You can specify order of topics using semicolon

Comma-separated list of wildcard patterns that match the names of topics to **exclude**. Leave blank to include all topics.

Name of a topic that contains a TWiki table, each row of which maps a topic names to the version of that topic to publish. Leave blank to publish the most recent version of each topic. The table can be generated by a %SEARCH or other TWiki variable. For example: | Web.TopicName | 1.33 |. If a topic does not appear in the table, the most recent version will be published.

A regular expression that will cause a topic to be **excluded** if the RE matches the topic content. Leave blank to include all topics.


Comma-separated list of Plugins to enable during publish. Leave blank to enable all plugins. You are recommended to disable any plugins that generate buttons in the output. The currently enabled plugins are: SpreadSheetPlugin, ActionTrackerPlugin, AttachmentListPlugin, AutoSectionsPlugin, BackupRestorePlugin, BibliographyPlugin, CalendarPlugin, CernLogoutPlugin, CernUserTopicPlugin, ChartPlugin, ChecklistPlugin, ColorPickerPlugin, CommentPlugin, DatePickerPlugin, DeleteRevisionsPlugin, DirectedGraphPlugin, DirectedGraphWebMapPlugin, DoxygenPlugin, EditTablePlugin, ExplicitNumberingPlugin, FilterPlugin, ForEachPlugin, GnuPlotPlugin, HeadlinesPlugin, HistoryPlugin, HolidaylistPlugin, HostnamePlugin, IfDefinedPlugin, ImageGalleryPlugin, ImagePlugin, InterwikiPlugin, JQueryPlugin, JiraPlugin, JqPlotPlugin, LatexModePlugin, LdapContribAdminPlugin, LdapNgPlugin, LogsStatisticsPlugin, MathModePlugin, NewsPlugin, OrphansPlugin, PerlDocPlugin, PreferencesPlugin, PublishWebPlugin, RedirectPlugin, RenderListPlugin, SetGetPlugin, ShareMePlugin, SlideShowPlugin, SmiliesPlugin, StopWikiWordLinkPlugin, SvnPlugin, SyntaxHighlightingPlugin, TWikiDrawPlugin, TWikiSheetPlugin, TablePlugin, TagMePlugin, TimeTablePlugin, TinyMCEPlugin, TopicDataHelperPlugin, TreeBrowserPlugin, TreePlugin, TwistyPlugin, VarCachePlugin, VotePlugin, WatchlistPlugin, WebPermissionsPlugin, WorkflowPlugin, WysiwygPlugin, XmlRpcPlugin

Output options

Select skin for published HTML The skin provides the template for how topics are published. See TWikiSkins for more

informations on skins.

You are recommended to pick `basic_publish`, or `plain`, or a `print` skin. Your installation may also offer a special `export` or `publish` skin.

 The `view` template is used to generate published pages, so `view.pattern.tmpl` is the template that will be used to generate the output. You can preview any topic in this skin simply by appending `?skin=pattern` to the end of the view URL. Note that the standard `VIEW_TEMPLATE` template override still works when publishing.

Output format

The output will be generated on the server, in the directory pointed to by the `{PublishContrib}{Dir}` configuration setting. You can manage the contents of this directory from the browser using the publishers control interface to the `publish` CGI script.

X The rendered data can get pretty big, and the process itself puts a heavy load on the server, especially when using compression on large webs.

Publishing history topic

This is where the history of your publishing is stored. Each time you publish, this topic is re-written with the log of the publishing process. You have to have "change" access to this topic. You can specify a topic in another web using the standard `Web.Topic` syntax.

Web options

Web options are only relevant for web output formats (`file` and `ftp`) The web formats generate a `sitemap.xml`, and can also generate `default.htm`, `index.html` and google site verification files.

Google file

generates the `HTML verification` file needed to verify your site claim. see Google webmaster tools

Default topic:

Name of topic to used to generate `default.htm`, `index.html`

Relative URL used in sitemap

the base URL that your published TWiki topics will reside at (if you are publishing to the root of your site, `/` is correct) see Google webmaster tools

FTP options

FTP options are only relevant if Output format is `ftp`

Destination FTP server

Set to blank to proof the output prior to uploading to your site. destination

Path to upload to on server

destination

FTP username

destination

The FTP output generator was written by TWiki:Main.SvenDowideit[?].

FTP Password

destination

Fast publish

Speed up the ftp publishing by only uploading modified files. This will store a (tiny) checksum (`.md5`) file on the server alongside each uploaded file which will be used to optimise future uploads. Recommended. fastupload

Other output generator options

Some output generators support extra options (e.g. for `pdf`, you can add `htmldoc` command-line parameters here, such as `--linkstyle underline`)

Wildcard Patterns

Wildcard patterns are well known to people who are used to command lines on computers, but may be unfamiliar to the Windows generation. A wildcard is a special string that you can put into a filename so that it matches a whole range of files:

String	What it does	Example	What the example matches
*	Matches any string, including an empty string.	*Cheese*	Every topic with "Cheese" somewhere in the name (but <i>not</i> "cheese")
?	Matches any single character.	Example1?	Example10 and Example 1X but <i>not</i> example1
[...]	Matches any one of the enclosed characters. A pair of characters separated by a hyphen denotes a range expression; any character that sorts between those two characters, inclusive, using the current locale's collating sequence and character set, is matched. If the first character following the [is a ^ then any character not enclosed is matched. A - may be matched by including it as the first or last character in the set. A] may be matched by including it as the first character in the set. Within [and], character classes can be specified using the syntax [:class:], where class is one of the following classes defined in the POSIX.2 standard: alnum, alpha, ascii, blank, cntrl, digit, graph, lower, print, punct, space, upper, word, xdigit. A character class matches any character belonging to that class. The word character class matches letters, digits, and the character _.	B[aeiou]g	Bag, Bog, Big, Beg, Bug

+#TopicOrder

Specifying topic order

You may want to specify the order of topics in a published file. You can do it by putting inclusion pattern lists separated by semicolon. Let's say you specify the following as inclusion pattern:

```
Abc*, Def*;Ghi*
```

Then, topics starting with Abc or Def are put into the output first. Next, topics starting with Def are put into the output.

More than one pattern lists may match a topic. Even in that case, a topic isn't included more than once in an output. For example, the following pattern yields an output having topics starting with Abcd first, then topics starting with Ab but not Abcd because topics starting with Abcd are already included.

```
Abcd*;Ab*
```

Regular Expressions

A perl regular expression. You can use a simple string here, which will be matched exactly, or you can read up on perl regular expressions on the web.

Using a Publish Topic (configtopic)

You can create a publish topic in a web that contains all the details needed to publish that web. This is just a topic with a series of standard TWiki variable settings (which correspond to the form parameters) in it. You can use the PublishWeb topic in this web as a template for your own topics.

Alternatively you can just take a copy of the form in this topic, paste it into your own topic, and change the defaults.

To use a publish topic, you must pass the `configtopic` parameter to the `publish` script set to the name of the topic to use to control publishing. You can specify a topic in another web using the standard `Web.Topic` syntax.

Publishing from the command line

TWiki-4 allows you to call any TWiki script from the command line, and the `publish` script is no exception. Just `cd` to the `bin` directory, and `perl -T publish`. Parameters are passed as `name=value` pairs, for example:

```
perl -T publish web=Book exclusions='Web*' format=file
perl -T publish web=Book inclusions=WebBook format=pdf genopt='--book --duplex --toclevels=5'
```

The available parameter names are shown in the example above, in the 'Name' column.

Controlling which parts of a topic get published

You can control what gets published from a topic using `%STARTPUBLISH%` and `%STOPPUBLISH%` control tags:

- If `%STARTPUBLISH%` is the first control tag seen in the file, everything before it will be ignored.
- Everything between `%STOPPUBLISH%` and the next `%STARTPUBLISH%` (or the end of the topic) will be ignored.
- `%STARTPUBLISH%` and `%STOPPUBLISH%` will be visible in the viewed topic, so you can easily see what will be published from the topic.

Note: the old `<nopublish>` tag is deprecated and should be replaced in topics

Another good trick is to set up a special "publishing" web. Create topics in the web that `%INCLUDE` the topics from **other** webs that you want to publish. You can use `STARTSECTION` and `ENDSECTION` to highlight what you want published. This way the "publishing" web gives you a view of exactly what will be in the published output, without the need for special publishing tags.

Publishing History

Every time a web is published, then the results of that publishing step are stored in a topic in the web. By default this topic is called `PublishContribHistory`, but you can choose another name (see the form, above). In order to publish a web, you have to be able to write to this topic. If you need to add access controls to the topic, then make sure you do that right at the beginning of the topic, or in the hidden preferences.

The history topics contains a list of all the parameters used, and the versions of the topics that were published, so it is very useful for tracking exactly what you publish. it is written every time you run `publish`.

Installation Instructions

Dependencies

Note: If you want to generate PDF files, you will need an installation of `htmldoc`. This program is available from <http://www.easysw.com/htmldoc/> for free, but you are **strongly** recommended to buy the commercial version. Your support for open-source projects helps make open-source software possible.

Note: You do not need to install anything on the browser to use this extension. The following instructions are for the administrator who installs the extension on the TWiki server.

- For an *automated installation*, run the configure script and follow "Find More Extensions" in the in the *Extensions* section.
 - ◆ See the installation supplement on TWiki.org for more information.
- Or, follow these *manual installation* steps:
 - ◆ Download the ZIP file from the extension home on twiki.org (see below).
 - ◆ Unzip `PublishContrib.zip` in your twiki installation directory.
 - ◆ Set the ownership of the extracted directories and files to the webserver user.
 - ◆ Install the dependencies (if any).

Run `configure` and complete the installation in the **PublishContrib** section. If you can't do this for some reason, these are the settings required in `LocalSite.cfg`:

<code>\$TWiki::cfg{PublishContrib}{Dir}</code>	File path to the directory where published files will be generated. you will normally want this to be visible via a URL, so the TWiki pub directory is a good choice.			
<code>\$TWiki::cfg{PublishContrib}{URL}</code>	URL path of the directory you defined above.			
<code>\$TWiki::cfg{PublishContrib}{PDFCmd}</code>	Template command-line for the PDF generator program - for example, <code>=htmldoc --webpage --links --linkstyle plain --outfile %FILE</code>	<code>F%</code> <code>%EXTRAS</code>	<code>U%</code> <code>%FILES</code>	<code>F%'</code> <code>=</code>

PDF output

1. install `htmldoc` from <http://www.easysw.com/htmldoc/>

Note that `htmldoc` can also be used to generate PostScript by using the `-t` option in the `Other` output generator options above. See the `htmldoc` man pages for details.

.tgz (tar) output

1. Install `Archive::Tar` and everything it depends on


.zip output

1. Install `Archive::Zip` and everything it depends on

Info

This add-on started as the [TWiki:Plugins/GenHTMLAddon](#), written by [TWiki:Main/CrawfordCurrie](#) at Motorola. It was then rewritten by [TWiki:Main/EricScouten](#), and then fixed and enhanced by [TWiki:Main/CrawfordCurrie](#) (<http://c-dot.co.uk>). It has been further extended by [TWiki:Main/SvenDowideit](#) and [TWiki:Main/MartinClever](#).

Authors:	TWiki:Main/CrawfordCurrie , TWiki:Main/EricScouten , TWiki:Main.SvenDowideit , TWiki:Main.MartinClever		
Dependencies:	Name	Version	Description
	File::Spec	>0	Required. Used to analyse URL paths.
	File::Copy	>0	Required. Used to move files around.
	File::Path	>0	Required to manipulate directories.
	File::Temp	>0	Required for temporary files.
	LWP	>0	Optional. Used to include images referenced by absolute URLs
	Archive::Zip	>=0	Optional. Required to generate .zip output
	Archive::Tar	>=0	Optional. Required to generate .tgz output
	Net::FTP	>0	Optional. Required for ftp publishing.
	htmldoc		Optional. Required to generate .pdf output
	Digest::MD5	>0	Optional. Required for fast upload to ftp servers.
Version:	24185 (2012-11-28)		
Change History:			
27 Oct 2008	TWikibug:Item5385 : Fixed doc for configtopic TWikibug:Item5388 : \$WEB and \$TOPIC were not correct in %IF statements TWikibug:Item5390 : remove comments from .css before processing for included resoures TWikibug:Item5706 : Improved FTP upload process for incrementally maintained webs TWikibug:Item6029 : expand config topic on load to support use of searches TWikibug:Item6030 : respect VIEW_TEMPLATE in published topics TWikibug:Item6092 : expand common tags in configtopic TWikibug:Item6110 : rename settings in config topic to avoid clashes with other plugins		
11 Dec 2007	TWikibug:Item5099 fixed		
10 Nov 2007	Tested on 4.2.0. TWikibug:Item4624 ., TWikibug:Item4625 : TWikibug:Item4830 : fixed. TWikibug:Item4825 : added a basic skin to avoid the confusion caused by text skin. TWikibug:Item4951 : added interface to allow management of output files		
13222	fixed ftp publish, added doco, and added enabled plugin selection funcitonality		
13064	TWikibug:Item3722 worked around core attaching URL params to internal URLs		
12961	TWikibug:Item3671 cannot publish without write access to history topic, so security now checked early. TWikibug:Item3619 Cleaned up error handling from writers. TWikibug:Item3675 added history topic to record changeset. Plus major refactoring of main class to get rid of some of the cruft that had built up from many authors. Item2726 : uses getExternalResource now, so should obey proxy settings (untested)		
12824	Added support for new internal api - no user changes		
12708	Added UI for FTP. Added .spec file. Fixed TWikibug:Item3515 and TWikibug:Item2725		
12028	Michael Daum - create a new TWiki object for every topic, don't reuse the current one (TWikibug:Item3139)		
10412	Correction to the correction for anchors.		
10345	Correction to support anchors in URLs properly		
10242	Martin Cleaver - changes to allow generation of viewprint and viewxxx when specified by TEMPLATE; multiple INSTANCE (dev/test/prod); (TWikibug:Item2269)		
10063	Bugfix TWikibug:Item2216		
10006			

	Crawford Currie - fixed problem where it was failing to remove <base> tags completely (TWikibug:Item2200 ↗)
9986	Crawford Currie - added doc on usage from command line, corrected sense of topicsearch filter (TWikibug:Item2120 ↗ , TWikibug:Item2121 ↗), renamed parameters (old ones are still valid), corrected handling of empty web refs (TWikibug:Item2128 ↗), deprecated nopublish html-style tag in favour of PublishWebPlugin-compatible style (though with richer semantics) (TWikibug:Item2196 ↗)
9823	Crawford Currie - added support for hierarchical webs, and inclusion of external images.
9773	Crawford Currie - added tgz and pdf support
9725	Michael Daum - fixed rewriting urls; fixed nested resources issue; creating a new prefs object for each topic
9713	Corrected form action so it uses up the right web preferences
9695	Michael Daum - recursively archive resources imported by css files; fixed several html errors in the PublishContrib and PublishWeb topics; removed hardcoded reference to print.pattern
8959	TWiki-4 version. Also supports publishing to a file area, making TWiki easier to use as a CMS (see also TWiki:Plugins/PublishWebPlugin ↗ , which does almost the same thing )
6276	TWikibug:Item196 ↗ - bugfix for HTTP_HOST, as described in the Dev topic for the contrib
5566	Changed interface to support wildcards, and lightened the plugin by replacing a lot of files with simpler ways of doing things.
5499	Added Compress::Zlib dependency, as requested by Brad Taylor
27 Apr 2005	1.301 Crawford Currie - fixed minor issues highlighted by Bruce Dillahunty and Scott Claridge
11 Apr 2005	1.3 Crawford Currie - reworked the interface and code to work better
13 October 2004	1.200 Crawford Currie - Cairo compatible
7 Jan 2003	1.1 Initial version
Home:	http://TWiki.org/cgi-bin/view/Plugins/PublishContrib ↗
Feedback:	http://TWiki.org/cgi-bin/view/Plugins/PublishContribDev ↗
Appraisal:	http://TWiki.org/cgi-bin/view/Plugins/PublishContribAppraisal ↗

Related Topics: TWikiPreferences, TWikiPlugins

Copyright

This code is a development of the Architectures and System Platforms group of Motorola Inc. and is protected by the following copyrights:

- Copyright © 2001 Motorola. All Rights Reserved.
- Copyright © 2002-2003, Eric Scouten.
- Copyright © 2004-2006 Crawford Currie <http://c-dot.co.uk>[↗](#)
- Copyright © 2006 Martin Cleaver <http://www.cleaver.org>[↗](#)


The 2005 functionality improvements were sponsored by Wind River Systems[↗](#)

The pdf and tgz output formats were made possible by Sabio Labs[↗](#)

License

As required for the publication of all extensions to TWiki, the software is published under the terms of the GNU General Public License.

This program is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See

the GNU General Public License for more details, published at <http://www.gnu.org/copyleft/gpl.html> 

This topic: TWiki > PublishContrib

Topic revision: r3 - 2012-11-02 - PeterJones



Copyright &© 2008-2019 by the contributing authors. All material on this collaboration platform is the property of the contributing authors.

Ideas, requests, problems regarding TWiki? Send feedback

Note: Please contribute updates to this topic on TWiki.org at TWiki:TWiki.PublishContrib