Wikipublisher

Wikipublisher: User Guide

02 February 2006, at 12:52 PM

From the PmWiki+WikiPublisher

PDF2You: http://www.wikipublisher.org/wiki/index.php?n=Wikipublisher.UserGuide

Creative Commons Wiki Licence

This wiki is licensed under the new (beta) *Creative Commons Wiki Licence*[a] — an *Attribution-ShareAlike* licence.

By editing or posting to this wiki, you agree that your contributions will be licensed under the Creative Commons Wiki 0.5 (beta) licence, and you agree to waive attribution for your contributions with the understanding that copyrighted material made available from this wiki will be attributed to the wiki itself.

Printed materials are made available under the terms of the Creative Commons Wiki 0.5 (beta) licence. You may also choose to receive these under the terms of any other copyleft licence, such as the *GNU Free Documentation Licence*[b]. You may:

1. make and distribute *verbatim copies* of these pages, provided that the copyright notice and this permission notice are preserved on all copies
2. copy and distribute *modified versions* of these pages under the conditions for verbatim copies, provided that the entire resulting derived work is distributed under the terms of a permission notice identical to this one
3. copy and distribute *translations* of these pages into another language, under the above conditions for modified versions

---

[a] creativecommons.org/drafts/wiki_0.5

[b] www.gnu.org/copyleft/fdl.html

USER GUIDE

The following pages describe various aspects of using, administering and troubleshooting a Wikipublisher (w) installation.

Wikipublisher (w) is an extension to PmWiki (w) that supports the collaborative creation of print documents which draw their content from wiki web pages. In fact *any web page* able to be reformulated as wikibook XML can be composed into a print-friendly (w) document.

PmWiki sites with wikipublisher installed look and act like normal wiki sites, except that they have a pdf icon on every page. Some pages, such as those listing trails (w) and categories (w), may include a Typeset button; pressing this button composes the listed pages into a single rich pdf document. Search results are displayed as a form, allowing information seekers to select and typeset pages of interest as a single document.

Wikipublisher is distributed under the *General Public Licence*[1]. It is distributed in 2 parts:

- a script library which makes PmWiki generate output to the Wikibook DTD (w), instead of XHTML
- a pdfserver application which, when given wikibook XML content, composes this into pdf

---

[1] www.gnu.org/licenses/gpl.html

This page introduces the wikipublisher USER GUIDE. Or view the Slideshow (w), then create a personalised guide (w).The site is running pmwiki-2.0.13 and wikipublisher-2.0.beta9 (final beta).

Contents

1 Basic Features (w)

THE WIKIPUBLISHER (w) SYSTEM COMPOSES PRINT pages by re-interpreting wiki markup (w) for print in place of the web. Just as PmWiki (w) applies the selected web template (skin and accompanying style sheet) to compose a web page, so wikipublisher applies a suitable print template to compose a print document. Basic typesetting features include:

- generate a cover page, summary page (optional), table of contents page, list of tables (optional), and list of figures (optional)
- turn intra-document page links into page number references
- autonumber page headings based on trail page hierarchy depth (optional)
- headings are turned into pdf bookmarks
- table column width and cell text wrapping and alignment handled automatically
- substitute high resolution print images for low resolution web images, if available
- show links to external sites, such as *PmWiki.org*[2], as page footnotes
- reformat tool tips, such as TINFL (there is no free lunch), as parenthesised text
- hide or omit web-oriented constructs, such as revision marks, in the print version
- pages laid out for duplex printing — mirrored odd and even page headers and footers (optional)
- sticky notes become marginal notes
- widow[3] and orphan[4] control and headings always kept with their following paragraphs
- support a wide range of formatting options, including *emphasis*, `monospaced` and coloured text
- one-click reader control over various composition options, such as font style, watermark, and when to start a new page, with an options (w) form
- visually-impaired readers can choose a large print option

Form follows function

While document composition is automatic and the markup translation guarantees consistent output, authors still need to proofread the printed results to check that the output is suitable and correct.

User Guide (p i)

---

[2] www.pmwiki.org/wiki

[3] a *widow* is when the last line of a paragraph falls at the top of a page or the last word of a paragraph falls on a line by itself

[4] an *orphan* is when the first line of a paragraph falls at the bottom of a page

2 Composition options

2.1 Publish a Trail (w)

The following options are available for typesetting the pages on a trail.

The "typeset" directive

An author can turn any wiki trail (w) page into the table of contents for a page collection by adding a *typeset* directive to the trail page:

This adds a Typeset button to the trail page which, when pressed, assembles the pages on the trail into a single document, typeset for printing. The *options* let the author over-ride the default document composition settings, using *keyword=value* pairs. The following table lists available options.

Table 1: Available typesetting options

Keyword	**Values**	**Comment**
surtitle	text	appears on cover
title	text	appears on cover and in header
subtitle	text	appears on cover and in footer
cover	left center right	alignment of cover text
coverstyle	sans serif	font style for cover headings
watermark	text	appears on every page
duplex	on off	format pages to print 2 sided
justification	on off	text justified or ragged
sectionnewpage	on off	major section starts a new page
autonumber	0 1 2 3	depth of numbered page headings
headingstyle	sans serif	font style for heading text
bodystyle	sans serif	font style for body text
tablelist	on off	generate a list of tables
figurelist	on off	generate a list of figures
pagesize	A4 US	determine the output page size
cache	on off	cache a copy of the generated pdf (future)
draft	on	watermark=draft cache=off subtitle='draft version'

Wikipublisher favours writers over readers. If an author sets any typesetting options, when a reader presses the PDF options button the author-defined settings are preserved and may not be changed.

The "pubtrail" action

An author can let readers choose which of the trail pages to include in the typeset document by using the *pubtrail* page action:

```
[[UserGuide?action=pubtrail]]
```

. . . .

UserGuide?action=pubtrail (w)

This turns the trail into a form, with a check box against each page on the trail. A reader checks pages of interest, then presses the Typeset button.

The “traillinks” directive

An author can also create a formatted list of the stops on a trail by using the *traillinks* directive, such as for display in a sidebar page. This directive omits the publishing features:

```
(:traillinks UserGuide:)
```

. . . .

- Basic Features (p 1)
- Composition options
  - PUBLISH A TRAIL
  - Publish a Search (p 4)
  - Publish Linked Pages (p 5)
  - Publish a Form List (p 7)
  - Structure and Style (p 8)
  - Notices (p 9)
- Administration options
  - Install (p 11)
  - Register (p 11)
  - Configure (p 12)
  - Customise (p 12)
  - Recipes (p 13)
  - Install the Server (p 15)
- Troubleshooting
  - Skins (p 17)
  - Markups (p 18)
  - PDF Errors (p 18)
  - Wikibook DTD (p 18)
- Reference
  - FAQ (p 21)
  - Wiki Styles (p 23)
  - Markup Extensions (p 24)
  - Typesetting Conventions (p 26)

User Guide (p i)

2.2 Publish a Search (w)

All the search features are optional; some may not be available on a given wiki. If in doubt, consult your wiki administrator. On screen, the lists of links on this page are each presented as a “select and typeset” form; in print, they are presented as lists of links.

The “search” action

After entering a request in the search box and pressing *Go*, the information-seeker sees a list of qualified pages, with a checkbox against each pagename and a *Typeset* button at the bottom. Check the box against each page of interest and press the button. Wikipublisher (w) composes the chosen pages into a PDF. For example, searching for “trail”:

```
(:pagelist group=Wikipublisher ’trail’ fmt=custom list=normal:)
```

……

Wikipublisher Basic Features[w]

- FAQ[w]
- Home Page[w]
- Install[w]
- Publish a Search[w]
- Publish a Trail[w]
- Recipes[w]
- User Guide[w]

Search for category pages

For sites whose pages are tagged with categories (w), the category search results can be displayed with checkboxes and a *Typeset* button by adding *fmt=custom* to the *pagelist* directive. For example:

```
(:pagelist link=Category.Requirements fmt=custom list=normal:)
```

……

No matches.

User Guide (p i)

2.3 Publish Linked Pages (w)

The links-related actions are optional capabilities.

The “backlinks” action

Clicking on the pagename at the top of the page will cause PmWiki (w) to run a search for all the pages that link to the current page. It displays these in alphabetical order as a standard search results page, with check-boxes and a *Typeset* button. An author may also add this to a particular page. Like this:

```
[[{$FullName}?action=backlinks]]
```

. . . .

Wikipublisher.PublishLinkedPages?action=backlinks (w)

To see the form, click the link.

The “links” action

This is a complement to the *backlinks* action — it shows all the pages that the current page links to, in alphabetical order. This may be part of the site’s template or used in a specific page. Like this:

```
[[{$FullName}?action=links]]
```

. . . .

Wikipublisher.PublishLinkedPages?action=links (w)

To see the form, click the link.

The “linkslist” directive

A variant of the links action can be used as a directive to list all the links on a particular page, in alphabetical order, for example in a Sidebar. This variant omits the typeset capability.

`(:linkslist UserGuide:)`

....

Wikipublisher Basic Features^w^

- Configure^w^
- Customise^w^
- FAQ^w^
- Home Page^w^
- Install^w^
- Install the Server^w^
- Markup Extensions^w^
- Markups^?^
- Notices^w^
- PDF Errors^w^
- Publish a Form List^w^
- Publish a Search^w^
- Publish a Trail^w^
- Publish Linked Pages^w^
- Recipes^w^
- Register^w^
- Skins^w^
- Structure and Style^w^
- Typesetting Conventions^w^
- Wiki Styles^w^
- Wikibook DTD^w^

User Guide (p i)

2.4 Publish a Form List (w)

The wikiforms recipe included with the Wikipublisher (w) library adds `(:wikiform:)` and `(:wikilist:)` markup directives. The wikiform directive builds a page creation form using a FormTemplate; the wikilist directive generates a list of pages meeting the *fieldname=fieldvalue* selection criterion set out in the directive, such as a list of active issues (w). An information seeker can compose a typeset version of the pages listed.

To sort the list, click on one of the columns (by default, the list is sorted by page name — a 5 digit number). Clicking the column again reverses the sort order. Once the list is in the desired order, press Typeset. To customise the report layout, press the PDF options button.

Wikipublisher can typeset any page collection. The library also includes the wiki calendar recipe, enabling an information seeker to compose a typeset version of the calendar entries on display. Again, this includes a PDF options button.

User Guide (p i)

2.5 Structure and Style (w)

The typesetting engine, $\LaTeX$, is *structure-oriented* — that is, an author focuses on content and meaning, not presentation. When composing print output, the engine applies presentation rules derived from the document's structure. This separation of content from presentation means printed pages are set out in a consistent manner, no matter how many different authors are involved.

It is this separation which enables us to generate web and print output from the same source. However, PmWiki (w) also provides Wiki Styles (w), enabling authors to control the presentation of individual content elements, supplementing or over-riding any rules which a site administrator may have specified in the site's web style sheet. There are 3 kinds of styling:

- inline styles apply to parts of a block
- block styles apply to the current block element
- division styles apply to multiple block elements

While wiki markup elements are *media-neutral* and can be readily transformed to web or print, styles are *web-oriented* and hence can present a problem for the print channel. Specifically, we have to transform css-based web page styles into structure-based print page elements. Most of the time, authors don't need to worry about this — Wikipublisher (w) takes care of it.

To do this, it makes "reasonable assumptions":

- inline styles are mostly honoured
  authors need to use inline styles with some caution — what looks good on the web may not look good in print, and *vice versa*

- block styles are usually honoured, but may be ignored
  in general, it honours structural instructions (float this image right) but ignores formatting instructions (centre this heading)

- division styles are treated case-by-case
  these are usually interpreted as print side-bars and float to the top or bottom of the page; this is a problem if the division is bigger than a page

This area will continue to evolve and improve. At the moment, it's fair to say that the more authors use markup, the less they have to worry about what the print output looks like. The more they use styles, the more they have to check that the print effect is suitable.

Tip Authors may find it useful to think of wiki styles as *wiki tags* — ways to define the meaning of content elements beyond those which the standard wiki markups offer. It then becomes easier to associate suitable web and print presentation rules with those wiki tags.

To be continued…

User Guide (p i)

2.6 Notices (w)

Wikipublisher (w) automatically inserts some standard pages in all outputs, giving a way to include controlled text, such as legal notices, at the start and end of all printed documents produced. The pages are:

- For page collections
  - Site.Print Header (w)
  - Site.Print Footer (w)
- For individual pages
  - Site.Page Print Header? (not currently used)
  - Site.Page Print Footer (w)

If a group header or footer page exists, this takes precedence over the Site page. Thus Wikipublisher.Page Print Footer? (if it exists) will over-ride Site.Page Print Footer (w).

3 Administration options

3.1 Install (w)

These instructions apply to a clean PmWiki (w) install.

1. Download and decompress the library from Cookbook:PublishPDF
2. Move the cookbook/wikipublisher/ directory into cookbook/
3. Move the skin directories in pub/skins/ into pub/skins/
4. Move the image files in pub/images/ into pub/images/
5. Move the css files in pub/css/ into pub/css/
6. Add the following line at the *start* of local/config.php
   include_once("$FarmD/cookbook/wikipublisher/extensions.php");
7. Modify your skin to use the required format for search boxes
   Or, use the default latexpmwiki skin provided
   To use your own skin, be sure to set $Skin *after* loading wikipublisher.
8. Register your site at http://www.wikipublisher.org/wiki
9. Read the enclosed User Guide (w) and Issues (w) (pdf)
10. Test that everything is working:
    - typeset a single page
    - add to a wiki trail page
    - do a search and publish
    - do a backlinks search
    - do a categories search

User Guide (p i)

3.2 Register (w)

The Wikipublisher (w) pdf2you server, which composes xml output into pdf, only accepts composition requests from registered sites. This works the same way as PmWiki.Url Approvals (w): if the requesting site is on the list of registered addresses, its requests get processed. To register a new wikipublisher user site:

1. edit the Wikipublisher:AccessControl.ApprovedWikiUrls page
2. add the address of your site to the list
3. press Save
4. click on your address to return to your site
5. click the pdf icon to generate a pdf of your site's home page

And that's it! There is also a link to the *access control* page on www.wikipublisher.org

User Guide (p i)

3.3 Configure (w)

Wikipublisher (w) includes a wikilib.d/ directory containing Site templates for all the forms used to control the way pages are composed for print. These are installed automatically and do not affect either the wiki.d/ or PmWiki (w) wikilib.d/ directories. The form templates define layout metadata used by the typesetting engine. They are used to:

- determine the default values for layout of typeset pages
- generate the forms seen when a reader selects PDF options

There are 6 template pages:

1. Site.Print Template (w) defines options for typesetting a single wiki page
2. Site.Publish Template (w) defines options for the `(:typeset:)` directive
3. Site.Search Template (w) defines options for typesetting search results
4. Site.Calendar Template (w) defines options for typesetting `(:wikilog:)` entries
5. Site.Wikipublisher (w) defines the page through which a reader confirms all typesetting requests
6. Site.Wikipublisher Error (w) defines the page displayed if wikipublisher encounters an error

An administrator may set up group print, publish, search or calendar templates. The settings in a group template over-ride those in the Site template. An administrator may also edit the site templates, for example to translate the field prompts and setting options into a language other than English.

The template syntax for a form element uses an extension of the definition list markup:

Prompt text variable tool tip text (settings) [±]

Do not change the variable *name! Do not change the + or - at the end of the line!*

When changing the language of radio button settings, the administrator *must* preserve the metadata value passed to the typesetting engine. For example, (left,centre,*right) becomes (left|gauche;centre;*right|droit). The order variable on the Calendar Template (w) illustrates this capability.

User Guide (p i)

3.4 Customise (w)

The Wikipublisher (w) library supports all standard PmWiki (w) markup, plus a number of markup extensions. Many site administrators will have taken advantage of the flexibility PmWiki offers and created local extensions to the standard markup. In most cases, such extensions will be ignored when typesetting a page, *unless the extension author provides a wikibook-compliant alternate markup rule*.

The library also provides alternate link and page list formats that are compatible with the Wikibook DTD (p 18) and appropriate for print. Again, the administrator must make sure that local customisations do not over-write these with formats appropriate for XHTML.

For example, suppose an administrator has created markup for the <b> and <i> xhtml tags, using

```
Markup(“ ‘*” , ‘inline’, “ /’\*(.*?)\*’/” , ‘<b>$1</b>’);
Markup(“ ‘~” , ‘inline’, “ /’~(.*?)~’/” , ‘<i>$1</i>’);
```

To make marked-up text appear correctly in a pdf, the administrator writes the following rules *after* loading the wikipublisher library (which sets $format) and *before* setting the html rules (so the pdf rules take priority):

```
if ($format==‘pdf’) {
    Markup(“ ‘*” , ‘inline’, “ /’\*(.*?)\*’/” ,
        ‘<tbook:visual markup=“ bf” >$1</tbook:visual>’);
    Markup(“ ‘~” , ‘inline’, “ /’~(.*?)~’/” ,
        ‘<tbook:visual markup=“ it” >$1</tbook:visual>’);
}
```

The scripts stdmarkup.php and altmarkup.php in the cookbook/wikipublisher/latex/ directory contain lots of examples. Together, they provide print support for all the standard markups defined in PmWiki plus the Cookbook:MarkupExtensions.

User Guide (p i)

3.5 Recipes (w)

The Wikipublisher (w) script library embodies a number of related recipes from the PmWiki Cookbook, plus various PmWiki.Skins (w). Taken together, they turn a PmWiki (w) site into a rich print-on-demand publishing environment. This page describes what they do and which ones can be turned off.

Recipes

latex Translates wiki markup into wikibook xml. This group of scripts, held in the latex/ directory, completely replaces stdmarkup.php and wikistyles.php. It also provides wikibook equivalents for the markup extensions described below.

Required.

mailform If wikipublisher encounters an error, it generates a form with error details that is emailed to the designated support address. It is an extension of the Cookbook:MailForm recipe.

Required; loads only when an error occurs.

markup Extends standard PmWiki markup to encompass a variety of additional structures, such as smart quotes, footnotes, and drop capitals. Each markup extension (p 24) can be individually turned on or off.

Required.

pagetoc Automatically generates a table of contents for a wiki page. This web navigation aid turns into a table of contents in print and is omitted in published page collections. The pages in the User Guide (w) make extensive use of toc directives.

Optional; to disable, set $MarkupEnabled[‘pagetoc’] = 0;

pdf2you Handles the interface between the wiki and the pdf2you typesetting server. Puts the pdf icon on every page.

Required.

rename Adds action=rename to rename any wiki page. To handle rename across groups, it generates a list of all the links on a page. As a by-product, it allows us to typeset a page and all (or a selection of) the pages it links to.

Optional; to disable, set $MarkupEnabled['rename'] = 0;

search Adds support for typesetting selected pages found by searching the wiki. This includes category pages. It also preserves location — all search actions occur within the current page.

Optional; to disable, set $MarkupEnabled['search'] = 0;

titledictindex Adds support for publishing selected pages from a dictionary-style index page. This variant uses the markup extension to show a teaser paragraph for each page.

Optional; to disable, set $MarkupEnabled['titledictindex'] = 0;

wikiform Handles the forms interface for specifying pdf layout metadata, using the Site Template pages. Used to track issues (w), including typesetting of issue lists.

Required.

wikilog Provides a simple wiki calendar, with the ability to typeset the currently-displayed set of calendar entries. Used for the Release Log (w). Also allows wiki forms to turn date fields of the form yyyy-mm-dd into a date, eg Tuesday, 13 September 2005.

Optional; to disable, set $MarkupEnabled['wikilog'] = 0;

wikipublisher The engine that typesets individual pages, trail pages and search result collections. It creates output in either xhtml or wikibook xml. It also sets print-oriented versions of PmWiki format variables such as $LinkPageExistsFmt.

Required.

wpversion Tells us which version of the wikipublisher library the site is running (wikipublisher-2.0.beta9 (final beta)).

Required.

Skins

The wikipublisher skins are all required.

latexpmwiki This is based on the standard pmwiki skin, modified to support wikipublisher. It adds a backlinks action to the page name, adds a links on this page option, and ties the search box to a search action. Use this as the starting point for a wikipublisher-aware custom skin.

latexprint This is the skin that produces wikibook xml for a single wiki page.

latexpublish This is the skin that produces wikibook xml for a wiki page collection.

publish This is the skin used to produce xhtml from the print view of a single page or page collections (pdf option is unchecked). It replaces the standard pmwiki print skin.

Dependencies

Wikipublisher includes a selection of pdf icons and expects to find these in pub/images.

The markup extension requires the markup.css file and expects to find this in pub/css.

The wikilog extension requires the wikilog.css file and also expects to find this in pub/css.

User Guide (p i)

3.6 Install the Server (w)

Software requirements:

- perl
- libwww-perl (available via the CPAN shell as Bundle::LWP)
- $\LaTeX$
- *ImageMagick*[5] (Mac version available from entropy.ch)
- libxml utilities
- libxslt utilities (especially, xsltproc)

Installation:

1. Grab the latest Wikipublisher snapshot
2. Untar it where you want to install the server software
3. Move the mkpdf directory to your cgi-bin
4. In the mkpdf directory:
    - 4.1. Use either mkpdf-affinity.inc (from a Mac) or mkpdf-visit.inc (from a Debian Linux box) as a template for the mkpdf.inc configuration file
    - 4.2. Change $src to reference the tbook directory under where you unpacked wikipublisher
    - 4.3. Change the $pdflatex, $xsltproc, $xmllint and $tbrplent to point to where pdflatex, xsltproc, xmllint and tbrplent are installed
        - If your system has no tbrplent, say it’s in $src/tbrplent/tbrplent; then open up a terminal window, change to the $src/tbrplent directory, and run the command “g++ tbrplent.cc -o tbrplent”. This requires the g++ compiler to be installed.
    - 4.4. Change $url and $check_url to point to your local PmWiki (w) install.
5. Make sure the required perl modules are installed — the ones you may not have are libwww-perl (LWP) and perlmagick (which often comes with ImageMagick).
6. Copy (from the wikipublisher snapshot) pmwiki2/local/Access Control.Approved Wiki Urls (w).php to your pmwiki installation’s local directory.

---

[5] www.imagemagick.org

7. Edit the Approved Wiki Urls [(w)] page in your wiki to include a link to your Wiki (in the style of the Access Control.Approved Wiki Urls [(w)] page on this wiki).
8. Create an Access Control.Splash Screen [(w)] that the wikipublisher server can display while it is generating a PDF.
9. Edit your pmwiki local/config.php file:
    - 9.1. set the $WikiPublisherUrlFmt variable to point to the mkpdf.pl CGI script you installed
    - 9.2. set $MailFormAddresses[‘support’] to the email address of your wikipublisher support person
10. Try PDFing some pages!

4 Troubleshooting

4.1 Skins (w)

The wikipublisher (w) skin template uses a *css-based text logo* rather than an image to designate the site's home page. If an administrator has a statement in the local/config.php file to point to an image logo while using the wikipublisher skin, a browser will display the *address* of the image, rather than the image itself. By default, wikipublisher sets:

```
$PageLogoUrl =
    <span class='wp'>Wiki</span><span class='wpub'>Publisher</span>;
```

In contrast, the default PmWiki (w) skin sets

```
$PageLogoUrl = $PubDirUrl/skins/pmwiki/pmwiki-32.gif;
```

Using PmWiki's default value with the wikipublisher skin will *not* work correctly. It is the most common problem with new wikipublisher installations. To avoid this, start with the sample-wp-config.php file included in the wikipublisher distribution. This sets a suitable default value for the skin logo. An administrator can of course use a different skin, with either a text or graphic logo, if preferred.

Wikipublisher aims to place a pdf icon on a wiki page in a way that will work correctly with a range of skins. To display a pdf icon on every wiki page in PmWiki's browse mode, wikipublisher adds the variable $PageTypesetFmt to PmWiki's $HandleBrowseFmt variable, between $PageStartFmt and $PageRedirectFmt. An administrator can control the positioning of the pdf icon using css. The default placement is “float: right”.

An administrator may over-ride the default pdf icon placement as follows:

1. enter the following code *before* the line in local/config.php that includes the Wikipublisher library
   $PageTypesetFmt = ‘’;
2. place the following code at the point in a local skin where the pdf icon is to appear

```
<p class='myclass'>
  <a href='$PageUrl?action=print&ptype=print&format=makepdf'>
  <img src='$PubDirUrl/images/pdficon.gif' title='Typeset page' /></a>
  <sup>(<a href='$PageUrl?action=texprint'>options</a>)</sup></p>
```

If an administrator disables the pdf icon by default, an author can create a *typeset this page link* by writing:

```
[[{$Name}?action=print&ptype=print&format=makepdf | Typeset]]
```

....

Typeset (w)

A future version may provide a markup directive for this, if sufficient demand exists.

User Guide (p i)

4.2 Markups?

Page *Markups?* does not exist.

4.3 PDF Errors (w)

The Wikipublisher (w) engine has been tested using a wide range of source pages. However, you may occasionally encounter problems. These fall into 3 broad classes — unexpected features, cosmetic errors and fatal errors. If you encounter an error, please contact (w) us.

Unexpected features

It's not a bug, it's a feature. Sometimes, the typesetting engine doesn't do what the reader or author expects. For example:

- it formats definition lists differently to the default style of many web browsers
- images "float" rather than staying in place
- long verbatim text lines may not fit across the paper, whereas a browser can scroll

For this class of error, one person's feature is another person's bug. We welcome constructive feedback.

Cosmetic errors

The server returns a pdf, but it contains unexpected or undesirable presentation results. There are some known bugs or limitations; check the Issues (w) list. If the site administrator has created local markup customisations, these may not be recognised. Wikipublisher transforms wiki markup to print xml — any residual html tags, such as those a third party recipe may produce, are removed automatically, leaving plain text. In many cases, an author can "write around" a cosmetic error.

Fatal errors

The server does not return a pdf. Web browsers are fairly forgiving of html pages that are badly formed or invalid. A browser will usually display *something*, no matter how bad the html. However, the xslt processor which transforms the print xml to typesetting instructions requires well formed xml (correct nesting, correct opening and closing tags) to work correctly. It forgives some types of invalid xml (the output doesn't conform to the Wikibook DTD (p 18)), but not others. We give fatal errors the highest priority.

User Guide (p i)

4.4 Wikibook DTD (w)

The Wikipublisher (w) library instructs PmWiki (w) to generate xml output to a print-oriented DTD (document type definition). This appeared to be a more straightforward route than directly converting the xhtml for several reasons:

- xhtml doesn't directly support common print features, such as footnotes and page cross-references
- xhtml includes a number of constructs which are web-specific, such as forms and tool tips
- wiki markup is simple and largely (but not totally) media-agnostic — it primarily describes a page's content

Investigations led to the *tbook dtd*[6] project on Sourceforge. Mapping the tbook entities to wiki markup showed that most (but not all) wiki markups had tbook xml equivalents. The wikipublisher project extended the tbook dtd to support all core PmWiki markups plus a number of Markup Extensions (p 24). To avoid confusion, the new dtd is called "wikibook".

The wikibook dtd describes the *structure* of the page's content, not its presentation. Presentation is left to the typesetting engine and is controlled, to a degree, through <meta> tags and their attributes. This approach generally works well, but it has some limitations, many arising from the physical differences between a scrolling colour screen and a fixed-size sheet of paper.

Style information is partially supported. There are major differences between the css-based model used in xhtml (and implicit in PmWiki) and the structure-based approach used in the typesetting engine. Currently, there is no print equivalent to the rich style options found in xhtml. The approach taken is to translate those style options with recognisable equivalents (inline styles such as text colour blue), and ignore the rest (block styles such as place a red dotted border around this paragraph with a pale blue background).

Much, but not all, tabular material is supported. True tables are supported well. Headings, captions, cell alignment and text wrap all map to their print equivalents. Where tables are used as a way to control the presentation of complex material, it currently works less well, especially with very long cells. It treats complex cells as "minipages" and keeps the content together on a page. As a result, if the cell contains more than a page of content, some will fall off the end of the page.

Division styles have limited suport. The wikibook dtd allows multiple block tags to be wrapped in a <group> tag, which is treated as a minipage. This area is under development. At the moment, groups are treated on a case-by-case basis. Significant additional work will be needed to create a mechanism for translating css properties to their typesetting equivalents. This will require a separate project, but only if there is sufficient demand.

The wikibook dtd can be inspected at http://www.wikipublisher.org/dtd/wikibook.dtd.

User Guide (p i)

---

[6] tbookdtd.sourceforge.net

5 Reference

5.1 FAQ (w)

Wikipublisher (w) offers authors and administrators a range of features for controlling content and appearance of printed output.

Techniques for authors

How do I enter text that appears on screen but is hidden in print?

There are 2 markup extensions to support this

```
Normal text is printed

=>Right-aligned and hidden for printing

=<Left-aligned and hidden for printing
```

……

Normal text is printed

How do I insert a page break into printed output?

Page breaks are based on structure. For composing a single wiki page, the reader has the option of specifying that headings of a particular level always start on a new page. By default, single pages are typeset with no breaks. For composing page collections, the reader has the option of specifying that each wiki page starts a new page. By default, search and calendar page collections are typeset with no breaks. Trail pages are typeset with each major section (first level list items) starting a new page.

How can I omit one or more trail list items from the typeset output?

The composer takes advantage of the "invisible stop" markup extension (backtick-fullstop). In normal page display, this is simply removed. However, a number of services, including typeset and page table of contents, ignore any text after an invisible stop. If an author starts a list item with an invisible stop, the composer ignores it. As far as PmWiki (w) is concerned, such items are not on the trail. For example:

```
‘.* a list item with an invisible stop at the beginning
```

……

- a list item with an invisible stop at the beginning

Why are there a lot of blank pages in the output and how do I prevent these?

The composer is configured to typeset page collections for double-sided printing. This means major headings start on odd numbered (right-hand) pages. If the previous content page is also an odd numbered page, the typesetting engine inserts a blank page automatically. A reader can over-ride this by selecting duplex *off* on the options form. A writer can put `(:typeset duplex=off:)` on a trail (p 3) page. An administrator can configure (p 12) the Site Templates. The ideal solution is to use a printer that supports duplex printing.

How do I keep text and floating images together?

The composer "floats" images to the top of the next page if they don't fit and normally the text following the image will flow back above it. This can sometimes separate an image from its associated text. If a writer ends the paragraph immediately following an image with a line break ([[<<]]), Wikipublisher (w) will keep the image and paragraph together. This is particularly useful when used in combination with %lfloat% or %rfloat% style markup. Remember to use image alt text on all images — these are used either as captions (if none is provided) or as the entry in the generated list of figures.

Techniques for administrators

Will I be able to install the wikipublisher server locally?

The server-side composition software has a number of external dependencies: $\LaTeX$, *ImageMagick*[7], and an xslt processor. Installing (p 11) the Wikipublisher PmWiki script library is simple; installing the server-side (p 15) software is less so. Once the beta version is more stable, it can be made available to interested parties, under an open source licence.

Wikipublisher overrides my chosen skin. What has happened?

Wikipublisher sets a default skin — a version of the standard PmWiki skin modified to support the full set of Wikipublisher options. It is probably best to test a new Wikipublisher install using this skin. Once everything is working correctly, an administrator can restore the preferred skin by setting $Skin = 'yourskin'; in the local/config.php file *after* the line that includes the Wikipublisher extensions. If the skin you expect doesn't load, it is probably because config.php is setting the skin *before* it loads the Wikipublisher library.

I press Typeset and a pdf comes back. How is that done?

The PmWiki engine takes wiki markup and normally produces xhtml output, suitable for viewing through a web browser. The Wikipublisher library gives PmWiki an alternate set of markup translation rules, which cause it to produce output to the print-oriented Wikibook DTD (p 18). The xml is then passed through an xslt processor to transform it into *LATEX*[8] markup. Pdflatex composes this into a pdf file.

Does the wikipublisher server keep copies of my data?

No. Any temporary files are purged at the end of each typesetting request. Wikipublisher does not keep the email addresses of those who ask for pdf files to be emailed. The server keeps the standard activity log files. Administrators responsible for sensitive data may choose to run their own version of the pdf server software.

---

[7] www.imagemagick.org

[8] www.tug.org

Can I use wikipublisher with other web content management systems apart from PmWiki?

Certainly. The typesetting engine composes print-oriented xml into a pdf file. Any content management system able to be taught how to output print xml instead of xhtml can be used. PmWiki happens to provide a particularly easy to use way of creating print xml from wiki markup.

User Guide (p i)

5.2 Wiki Styles (w)

The PmWiki.Wiki Styles (w) introduction states:

> WIKISTYLES allow authors to modify the color and other styling attributes of a page's contents. A wikistyle is written using percent-signs, as in `%red%` or `%bgcolor=blue%`.

It does this by applying stylistic attributes to the HTML tags generated during markup translation, which the reader's Web browser then interprets.

Wikipublisher (w) aims to honour the author's stylistic intentions, but is constrained by what the typesetting engine can do and the intrinsic differences between print and Web. A style that is appropriate for a Web page may not be appropriate for print. The engine detects all applied styles and converts those it understands to their print equivalents, ignoring those it doesn't understand. The following table summarise the status of style support in wikipublisher-2.0.beta9 (final beta).

Wikipublisher style support

Supported now

```
center text
right-align text
text color
text background color
division color
division background color
division frames
comment
```

Partially supported

```
block background color (treated as text background color)
```

Future support

```
text floats
list bullet/number styles
```

Not supported

```
figure frames (sidecaps makes this hard)
external CSS styles
table styles
any other attributes
```

If there is a style that you think Wikipublisher ought to support, either raise an issue(w) or contact(w) the wikipublisher project. If you have pages that make heavy use of styles, be sure to test the typeset versions for unexpected or undesirable behaviour.

User Guide (p i)

5.3 Markup Extensions(w)

An administrator can turn extensions on and off

Wikipublisher(w) works best when all Markup Extensions are turned on. This is the installation (p 11) default. However, all extensions are optional, controlled through the $MarkupExtensionsFmt array. This is set to:

```
SDV($MarkupExtensionsFmt,
  array("inote abbr ‘A ‘. ‘- ‘s ‘: ‘f -d ... aquo mac ’/ ’@ ’; [^",
  "q&a A; {|} =| {= revisions ^!! fig :: para lazyweb spaced squo links"));
```

To enable a selected subset of these markups, copy the above 3 lines into local/config.php *before* including the Wikipublisher(w) library and remove the markups you don’t want.

To disable just a few markups, add one or more lines like the following *before* including the library:

```
   $MarkupExtensions[’extension’] = false;
```

For example, $MarkupExtensions[‘q&a’] = false; will disable the variant of Q: and A: markup, restoring PmWiki(w)’s default behaviour.

Available extensions

All extensions “do the right thing” when a page is composed for printing; most of them significantly improve the quality of printed pages.

Character markups

These extensions provide additional special characters.

Inline markups

These extensions provide additional text markups.

Table 2: List of character extensions

Extension	**Markup**	**Function**
‘.	`‘.`	invisible stop; removed on output, tells markups such as para to “stop”
‘-	`‘-`	backtick-hyphen makes an en dash – (meaning “to” as in A–B)
‘s	`‘`	backtick-space makes a non-breaking space
‘:	`‘:`	backtick-colon makes a mid-dot ·
‘f	`‘1/4 ‘1/2 ‘3/4`	makes a fraction ¼ ½ ¾
-d	`-- d-d +- -d`	make em dash —, digit en dash digit, plus or minus ±, and minus digit
...	`...`	ellipsis . . .
aquo	`<- -> <<...>> <...| |...>`	left and right arrows ← → and angle quotes « . . . » ‹ . . . | and | . . . ›
mac	{vowel}	macronised vowels Ā, ā, Ē, ē, etc
copy	`(tm) (r) (c)`	™ ® © symbols
squo	n/a	automatic smart quotes — highly recommended for pdf

Table 3: List of inline extensions

Extension	**Markup**	**Function**
‘/	`’/text/’`	<cite>text</cite> (often rendered in italics)
‘@	`’@text@’`	<kbd>text</kbd> (often rendered in monospace)
‘;	`’;text;’`	small caps text (if the browser supports this font variant)
[^	`[^footnote text^]` and `[^#^]`	swallow footnote text and list the footnotes, with links both ways
{|}	`{abbr|abbreviation}` and `{:term:definition}`	<abbr title=‘. . . ’> and <dfn title=‘. . . ’> tags (with print equivalents)
{=	`{=sticky note|colour=}`	a sticky note, colour (optional) is yellow, green, blue, purple, pink, grey
revisions	(:revisions:)	toggle between show and hide revision markups insert and

Block markups

These extensions provide additional paragraph markups.

Link markups

These extensions provide additional link markups.

Table 4: List of paragraph extensions

Extension	Markup	Function
q&a	Q: and A:	render Q: and A: markups with drop-caps
A;	D;rop caps; markup	Drop cap followed by small caps (looks stunning in pdf)
=|	`=<text`, `=|text` and `=>text`	left, centred and right aligned text (left and right aligned omitted in print)
^!!	`!run-in head!text`	run-in headings (if the browser supports this — looks stunning in pdf)
fig	=figure image caption	display an image and its caption (superseded by image | caption)
::	:term:definition:+more definition	multiple <dd> tags per <dt> tag (also tidy :: used for indenting) and * item:+more text for paragraphs in lists

Table 5: List of link extensions

Extension	Markup	Function
inote	`(:inote PageName:)`	inserts the contents of PageName formatted as a sticky note
abbr	n/a	prevents wikiwords with only one lower case letter, like PhD or CIOs
‘A	`Wiki‘Word`	another way to prevent wikiwords, like PhD
para	T[*#:] Name#id and `(:para Name#id action:)`	teaser markup and paragraph include markup (action = more, edit)
lazyweb	`www.affinity.co.nz`	lazy web links — omit the http://
spaced	n/a	tidy wikiword spacing and display wikiwords like TheReturnOfTheKing as The Return of the King
links	n/a	automatic tool tips on wiki links, if the page starts with a heading or definition; also disables links to the page you’re on, and highlights them

User Guide (p i)

5.4 Typesetting Conventions (w)

When coverting web pages to print, the typesetting engine automatically applies standard conventions for printed material. For a given input, it optimises the quality of the printed output and applies the rules of typesetting consistently to every page. This means authors can focus on content, rather than presentation. It also means authors do not need to be typesetting experts to produce professional-looking printed documents from their web page collections. The following are among the more common conventions followed:

- captions are placed above tables and below figures

- images and tables "float" to the top of the next page if there is insufficient room; the text following flows back around the floated object
- captions for images floated left or right on the web are on the right of the image on recto pages and the left on verso pages

While the typesetting engine maximises quality for a given input, authors may find the output unsatisfactory in some cases. The generally recommended solution is to adjust the input, rather than trying to change the output rules. This is because:

- unless we know the rules, we can't make an informed decision about when to break them — most people are not typesetting experts
- in a collaborative authoring environment, it is almost impossible to teach everyone the conventions
- computers are much better than people at applying conventions consistently

Those who are used to controlling the look of their outputs have to learn to relax, go with the flow and resist the temptation to fiddle. The wikipublisher authors do not claim to be typesetting experts; the pages look the way they do because that's the way the engine composes them. Authors coming from a word processing background may take some time to get used to this approach.

User Guide (p i)

Retrieved from http://www.wikipublisher.org/wiki/index.php?n=Wikipublisher.UserGuide

Page collection published on 02 February 2006, at 12:52 PM

Copyright © 2005–2006 the Wikipublisher wiki and its authors