mwlib Documentation

Release 0.15

Volker Haas, Ralf Schmitt, Johannes Beigel

January 20, 2016

Contents

Contents:

CHAPTER 1

Getting started

mwlib provides a library for parsing MediaWiki articles and converting them to different output formats.

The collection extension is a MediaWiki extensions enabling users to collect articles and generate PDF files from those.

Both components are used by wikipedia's 'Print/export' feature.

If you're running a low-traffic public mediawiki installation, you only have to install the collection extension. You'll have to use the public render server run by pediapress GmbH. Please read Collection Extension for MediaWiki.

If you need to run your own render server instance, you'll have to install mwlib and mwlib.rl first. Please read Installation of mwlib.

CHAPTER 2

Contact/Need help

If you need help with mwlib or the Collection extension you can either browse the mwlib mailing list or subscribe to it via mail.

The developers can also be found on IRC in the #pediapress channel

2.1 Dev Support

Need help with architectural advice or design validation? Get development help from our core team!

Want more information about our Development Support? Let us know how to get a hold of you and we'll be in touch soon.

contact us

contact@brainbot.com / +49 (0) 6131 2116391

2.2 Production Support

Have mwlib live in production and looking for SLA-based support? In production? Get reliable support from team that built mwlib.

We'd love to talk to you about our Production Support. Let us know how to get a hold of you and we'll be in touch soon.

contact us

contact@brainbot.com / +49 (0) 6131 2116391

CHAPTER 3

Installation of mwlib

If you're running Ubuntu 10.04 or a similar system, and you just want to copy and paste some commands, please read *Installation Instructions for Ubuntu 10.04 LTS*

Microsoft Windows is *not* supported.

3.1 Basic Prerequisites

You need to have a C compiler, a C++ compiler, make and the python development headers installed. mwlib will work with python 2.6 and 2.7. It will *not* work with python versions >= 3 or < 2.6. mwlib requires a recent UNIX-like operating system.

mwlib requires the python imaging library (pil) and the python lxml package. In order to compile pil from source the libjpeg, zlib, freetype and lcms header files and libraries must be present on the system. Compiling lxml requires the libxslt and libxml2 header files and libraries.

mwlib is split into multiple namespace packages, that each provide different functionality:

mwlib core functionality; provides a parser

mwlib.rl generates PDF files from mediawiki articles. This is what is being used on wikipedia in order to generate PDF output.

mwlib.zim generate ZIM files from mediawiki articles

3.2 Installation of mwlib with pip/easy_install

We recommend that you use a virtualenv for installation. If you don't use a virtualenv for installation, the commands below must probably be run as root.

Installation of mwlib can be done with:

```
$ pip install -i http://pypi.pediapress.com/simple/ mwlib
```

Make sure the output of the last command contains:

```
...
--- JPEG support available
--- ZLIB (PNG/ZIP) support available
--- FREETYPE2 support available
...
```

This will install mwlib and it's dependencies. The "-i http://pypi.pediapress.com/simple/" command line arguments instruct pip to use our private pypi server. It contains known "good versions" of mwlib dependencies and bugfixes for the greenlet package.

3.3 Installation of mwlib.rl with pip/easy_install

The following command installs the mwlib.rl package:

```
pip install -i http://pypi.pediapress.com/simple/ mwlib.rl
```

If you want to render right-to-left texts, you must also install the pyfribidi package:

```
pip install -i http://pypi.pediapress.com/simple/ pyfribidi
```

3.4 Testing the installation

Use the following two commands to test the installation:

```
mw-zip -c :en -o test.zip Acdc Number
mw-render -c test.zip -o test.pdf -w rl
```

Open test.pdf in your PDF viewer of choice and make sure that the result looks reasonable.

3.5 Optional Dependencies

mwlib uses a set of external programs in order to handle certain mediawiki formats. You may have to install some or all of the following programs depending on your needs:

- imagemagick
- texvc
- latex
- blahtexml

3.6 Installation Instructions for Ubuntu 10.04 LTS

The following commands can be used to install mwlib on Ubuntu 10.04 LTS. Run the following as root:

```
apt-get install -y gcc g++ make python python-dev python-virtualenv   \
  libjpeg-dev libz-dev libfreetype6-dev liblcms-dev                    \
  libxml2-dev libxslt-dev                                              \
  ocaml-nox git-core                                                   \
  python-imaging python-lxml                                           \
  texlive-latex-recommended ploticus dvipng imagemagick                \
  pdftk
```

After that switch to a user account and run:

```
virtualenv --distribute --no-site-packages ~/pp
export PATH=~/pp/bin:$PATH
hash -r
export PIP_INDEX_URL=http://pypi.pediapress.com/simple/
pip install pyfribidi mwlib mwlib.rl
```

Install texvc:

```
git clone https://github.com/pediapress/texvc
cd texvc; make; make install PREFIX=~/pp
```

Then *test the installation*.

3.7 Development version

The source code is managed via git and hosted on github. Please visit pediapress's profile on github to get an overview of what's available and for further instruction on how to checkout the repositories.

You will also need to install cython, re2c and gettext if you plan to build from the git repositories.

CHAPTER 4

Running a renderserver

4.1 Overview

Running a renderserver consists in running multiple programs [1]. Unless you have some special requirements, you should be able to start a working renderserver by running the following commands:

```
$ nserve
$ mw-qserve
$ nslave --cachedir ~/cache/
$ postman
```

These programs have the following purposes:

nserve nserve is a HTTP server. The Collection extension is talking to that program directly. nserve uses at least one mw-qserve instance in order to distribute and manage jobs.

mw-qserve mw-qserve is a job queue server used to distribute and manage jobs. You should start one mw-qserve instance for each machine that is supposed to render pdf files. Unless you're operating the Wikipedia installation, one machine should suffice.

nslave nslave pulls new jobs from exactly one mw-qserve instance and calls the mw-zip and mw-render programs in order to download article collections and convert them to different output formats. nslave uses a cache directory to store the generated documents. nslave also starts an internal http server serving the content of the cache directory.

postman postman uploads zip collections to pediapress in case someone likes to order printed books. You should start one instance for each mw-qserve instance.

None of the programs has the ability to run as a daemon. We recommend using runit for process supervision. daemontools is similar solution. Another alternative is to use supervisor.

4.2 nserve usage

nserve understands the following options:

`--port=PORT`

specify port to listen on. Default is to listen on port 8899 on any interface.

`--qserve=HOST:PORT` register qserve instance running on host HOST listening on port PORT

[1] In mwlib prior to version 0.13 it was possible to get away with running a single `mw-serve` program or even running no program at all by using the mwlib.cgi script. These programs have been removed in favor of the new tools, which provide the ability to scale an installation.

Any additional arguments are interpreted as additional qserve instances to register.

The following command starts nserve listening on port 8000 using two qserve instances:

```
nserve --port 8000 example1:14311 example2
```

4.3 mw-qserve usage

mw-qserve understands the following options:

-p PORT specify port to listen on. Default is to listen on port 14311

-i INTERFACE specify interface to listen on. Default is to listen on any interface.

4.4 nslave usage

nslave understands the following options:

--cachedir=CACHEDIR

specify cachedir to use. this is where nslave will store generated documents.

--serve-files-port port on which to start the http server (default is 8898)

--url=URL specify url under which the cache directory is being served. The default is to compute this value dynamically.

--numprocs=NUMPROCS allow up to NUMPROCS parallel jobs to be executed

4.5 postman usage

postman understands the following options:

--cachedir=CACHDIR specify cachedir to use. use the same value as specified when calling nslave

CHAPTER 5

command line tools

5.1 Common Options

This section contains a description of options that are accepted by more than one command.

`-h, --help`

Show usage information and exit.

`-c, --config=CONFIG`

The value for this option describes the source of MediaWiki articles and images for the command and can be of one of the following types:

- A "base URL" of a MediaWiki installation. A base URL is the URL up to, but not including the `index.php`/`api.php` part.

  This URL can differ from the prefix seen in "pretty" article URLs. For example the article *Physics* in the English Wikipedia has the URL http://en.wikipedia.org/wiki/Physics, but the base URL is http://en.wikipedia.org/w/.

  If you've set up your own MediaWiki you probably know what your base URL should be, but if you're using a different MediaWiki, you can see the base URL if add a query string to the URL, e.g. by clicking on the edit link or by looking at an older revision of an article.

  This value for `--config` corresponds to `type=mwapi` in a configuration file (see `docs/configfiles.txt`), i.e. articles and images are fetched with the MediaWiki API. Specifying the URL directly as value for `--config` is usually the quicker way to achieve exactly the same result.

  This requires MediaWiki 1.11 or later.

- A shortcut for a base URL. Currently there are the following shortcuts:
  - ":en" – http://en.wikipedia.org/w/, i.e. the English Wikipedia
  - ":de" – http://en.wikipedia.org/w/, i.e. the German Wikipedia
- A filename of a ZIP file generated with the *the mw-zip Command*.
- A filename of a configuration file (see `docs/configfiles.txt`).

`-m, --metabook=METABOOK`

Description of the article collection to be rendered in JSON format. This is used by the Collection extension to transfer this information to `mw-serve` which in turn passes the information to `mw-render` and `mw-zip`.

`--collectionpage=COLLECTIONPAGE`

Title of a saved article collection (using the Collection extension)

`-x, --no-images`

If given, no images are included in the output document.

`-i, --imagesize=IMAGESIZE`

Maximum size (which can be either width or height, whichever is greater) of images. If images exceed this maximum size, they're scaled down.

`-o, --output=OUTPUT`

Write output to given file.

`-l, --logfile=LOGFILE`

Log output to the given file.

`--login=USERNAME:PASSWORD[:DOMAIN]`

For MediaWikis that restrict the viewing of pages, login with given USERNAME, PASSWORD and optionally DOMAIN.

Currently this is only supported for mwapidb, i.e. when the –config argument is a base URL or shortcut, or when `type=mwapi` in the configuration file.

`--title`

Specify a title for the article collection. This is e.g. used by some writers to produce a title page. This title overrides titles contained in ZIP files or metabook files.

`--subtitle`

Specify a subtitle for the article collection. This is e.g. used by some writers to produce a title page (note that subtitle might require a tilte). This subtitle overrides subtitles contained in ZIP files or metabook files.

5.2 The `mw-render` Command

Render MediaWiki articles to one of several output formats like PDF or OpenDocument Text.

5.2.1 Usage

```
mw-render [OPTIONS] [ARTICLETITLE...]
```

5.2.2 Specific Options

`-w, --writer`

Name of the writer to produce the output. The list of available writers can be seen with `mw-render --list-writers`.

`--list-writers`

List the available writers.

`-W, --writer-options`

Writer specific options in a ";" separated list (depending on your shell, quoting with "..." or '...' might be needed). Each item in that list can either be a single option or an option=value pair. To list the available writer options use `mw-render --writer-info WRITERNAME`.

`--writer-info=WRITER`

Show available options and some additional information about the given writer.

`-s, --status-file=STATUS_FILE`

Write status/progress information in JSON format to this file. The file is continuously updated during the execution of `mw-render`.

`-e, --error-file=ERROR_FILE`

If an error occurs, write the error message to this file. If no error occurs this file is not written/created.

`--keep-zip=FILENAME`

Do not remove the (otherwise temporary) ZIP file, but save it under FILENAME.

5.3 The `mw-zip` Command

Generate a ZIP file containing

- articles,
- images,
- templates and
- additional meta information (especially if `--metabook` is given, see *Common Options*) like name and URL of the MediaWiki, licensing information and title, subtitle and the hierarchical structure of the article collection.

5.3.1 Usage

```
mw-zip [OPTIONS] [ARTICLETITLE...]
```

5.3.2 Specific Options

`-p, --posturl=POSTURL`

Upload the ZIP file with an HTTP POST request to the given URL.

`-g , --getposturl`

Retrieve the POSTURL from PediaPress and open the upload page in the web browser.

5.4 The `mw-post` Command

Send a ZIP file generated with *the mw-zip command* to a given or an automatically retrieved URL via HTTP POST request.

5.4.1 Usage

```
mw-post [OPTIONS]
```

5.4.2 Specific Options

`-i, --input=INPUT`

Filename of ZIP file.

`-p, --posturl=POSTURL`

Upload the ZIP file with an HTTP POST request to the given URL.

`-g , --getposturl`

Retrieve the POSTURL from PediaPress and open the upload page in the web browser.

5.5 The `mw-serve-ctl` command

`--purge-cache=HOURS`

Remove all cached files in –cache-dir that haven't been touched for the last HOURS hours. This is meant to be run as a cron job.

`--clean-up`

Report errors for processes that have died irregularly.

CHAPTER 6

Internals

The following section describes some of the internals of mwlib. Only read this if you plan to extend mwlib's functionality.

6.1 Writers

A *writer* in mwlib generates output from a collection of MediaWiki articles in some writer-specific format.

6.1.1 The writer function

Essentially a writer is just a Python function with the following signature:

```
def writer(env, output, status_callback, **kwargs): pass
```

Note that the function doesn't necessarily have to be called "writer".

The `env` argument is an `mwlib.wiki.Environment` instance which always has the `wiki` attribute set to the configured `WikiDB` instance and the `metabook` attribute set to a filled-in `mwlib.metabook.MetaBook` instance. If images are used, the `images` attribute of the `env` object is set to the configure `ImageDB` instance.

The `output` argument is a filename of a file in which the writer should write its output.

The `status_callback` argument is a callable with the following signature:

```
def status_callback(status=None, progress=None, article=None): pass
```

which should be called from time to time to update the status/progress information. `status` should be set to a short, English description of what's happening (e.g. "parsing", "rendering"), `progress` should be an integer value between 0 and 100 indicating the percentage of progress (actually you don't have to worry about setting it to 0 at the start and to 100 at the end, this is done by `mw-render`) and `article` should be the unicode string of the currently processed article. All parameters are optional, so you can pass only one or two of the parameters to `status_callback()` and the other parameters will keep their previous value.

The return value of the writer function is not used: If the function returns, this is treated as success. To indicate failure, the writer must raise an exception. Use the `WriterError` exception defined in `mwlib.writerbase` (or a subclass thereof) and instantiate it with a human readable English error message if you want the message to be written to the error file specified with the `--error-file` option of `mw-render`. For all other exceptions, the traceback is written to the error file.

Your writer function can define additional keyword arguments (indicated by the "`**kwargs`" above) that can be passed to the writer with the `--writer-options` argument of the `mw-render` command (see below). If the user

specified a writer option with `option=value`, the kwarg `option` gets passed the string `"value"`, if she specified a writer option just with `option`, the kwarg `option` gets passed the value `True`. All writer options should be optional and documented using the options attribute on the writer object (see below).

6.1.2 Attributes

Optionally – and preferably – this function object has the following additional attributes:

```
writer.description = 'Some short description'
writer.content_type = 'Content-Type of the output'
writer.file_extension = 'File extension for documents'
writer.options = {
    'foo: {
        'help': 'help text for "switch" foo',
    },
    'bar': {
        'param': 'PARAM',
        'help': 'help text for option bar with parameter PARAM',
    }
}
```

For example the writer "odf" (defined in `mwlib.odfwriter`) sets the attributes to these values:

```
writer.description = 'OpenDocument Text'
writer.content_type = 'application/vnd.oasis.opendocument.text'
writer.file_extension = 'odt'
```

and the writer "rl" from mwlib.rl (defined in `mwlib.rl.rlwriter`) sets the attributes to these values:

```
writer.description = 'PDF documents (using ReportLab)'
writer.content_type = 'application/pdf'
writer.file_extension = 'pdf'
writer.options = {
    'coverimage': {
        'param': 'FILENAME',
        'help': 'filename of an image for the cover page',
    }
}
```

The description is used when the list of writers is displayed with `mw-render --list-writers`, all information is displayed with `mw-render --writer-info SOMEWRITER`. The content type and file extension are written to a file, if one is specified with the `--status-file` argument of `mw-render`.

6.1.3 Publishing the writer

Writers are made available as plugins using setuptools entry points. They have a name and must belong to the entry point group "mwlib.writers". To publish writers in your distribution, add all included writers to the entry group by passing the entry_points kwarg to the call to `setuptools.setup()` in your `setup.py` file:

```
setup(
    ...
    entry_points = {
        'mwlib.writers': [
            'foo = somepackage.foo:writer',
            'bar = somepackage.barbaz:bar_writer',
            'baz = somepackage.barbaz:baz_writer',
        ],
```

```
    },
    ...
)
```

6.1.4 Using writers

From the command line, writers can be used with the `mw-render` command. Called with just the `--list-writers` option, `mw-render` lists the available writers together with their description. A name of an available writer can then be passed with the `--writer` option to produce output with that writer. For example this will use the ODF writer (named "odf") to produce a document in the OpenOffice Text format:

```
$ mw-render --config :en --writer odf --output test.odt Test
```

Additional options for the writer can be specified with the `--writer-options` argument, whose value is a ";" separated list of keywords or "key=value" pairs.

6.2 Metabooks

A Metabook describes a collection of articles and chapters together with some metadata like title or version. The actual data (e.g. the wikitext of articles) is not contained in the Metabook.

The Metabook is a simple dictionary containing lists, integers, strings (which are Unicode-safe; they are represented as unicode in Python) and other dictionaries. When read from/written to a file or sent over the network, it"s serialized in JSON format.

6.2.1 Metabook Types

Every dictionary contained in the Metabook (and the Metabook dicionary itself) has a type. The different types are described below. The Metabook dictionary itself has type "collection".

6.2.2 Collection

type (string):

Fixed value "collection"

version (integer):

Protocol version, 1 for now

title (string, optional):

Title of the collection

subtitle (string, optional):

Subtitle of the collection

editor (string, optional):

Editor of the collection

items (list of *article* and/or *chapter* objects, can be empty):

Chapters and top-level articles contained in the collection

licenses (list of *license* objects):

List of licenses for articles in this collection

6.2.3 License

type (string)

Fixed value “license”

name (string)

Name of license

mw_license_url (string, optional)

URL to license text in wikitext format

mw_rights_page (string, optional)

Title of article containing license text

mw_rights_icon (string, optional)

URL of license icon

mw_rights_url (string, optional)

URL to license text in any format

mw_rights_text (string, optional)

Name and possibly a short description of the license

6.2.4 Article

type (string):

Fixed value “article”

content_type (string):

Fixed value “text/x-wiki”

title (string):

Title of this article

displaytitle (string, optional):

Title to be used in rendered output instead of the real title

revision (string, optional):

Revision of article, i.e. oldid for MediaWiki. If omitted, the latest revision is used.

timestamp (integer, optional):

UNIX timestamp (seconds since 1970-1-1) of the revision of this article

url (string):

URL to article in source wiki

authors (list of strings):

list of principal authors

source-url (string)

URL of source wiki. This URL is the key to an item in the sources dictionary in the content.json object of the ZIP file.

6.2.5 Chapter

type (string):

Fixed value "chapter"

title (string):

Title of this chapter

items (list of *article* objects, can be empty):

List of articles contained in this chapter

6.2.6 Source

type (string)

Fixed value "source"

system (string):

Fixed value "MediaWiki" for now

url (string, optional):

"home" URL of source, e.g. "http://en.wikipedia.org/wiki/Main_Page" (same as key for this entry)

name (string):

Unique name of source, e.g. "Wikipedia (en)"

language (string)

2-character ISO code of language, e.g. "en"

interwikimap (dictionary mapping prefixes to *interwiki* objects, optional)

Describes interwikimap for this wiki, cf. http://en.wikipedia.org/w/api.php?action=query&meta=siteinfo&siprop=interwikimap

6.2.7 Interwiki

Interwiki entries can describe language links and interwiki links

type (string)

Fixed value "interwiki"

prefix (string)

Prefix is MediaWiki links, i.e. the part before the ":". This is the key in the interwikimap attribute of a *source* object.

url (string)

URL template, the string "$1" gets replaced with the link target (w/out prefx)

local (bool, optional)

True if the interwiki link is a “local” one

language (string, optional)

Name of the language, if this interwiki describes language links

6.2.8 Example

Given in JSON notation:

```
{
    "type": "collection",
    "version": 1,
    "title": "This is the Collection Title",
    "subtitle": "An optional subtitle",
    "editor": "Jane Doe",
    "items": [
        {
            "type": "article",
            "title": "Top-level Article",
            "content_type": "text/x-wiki"
        },
        {
            "type": "chapter",
            "title": "First Chapter",
            "items": [
                {
                    "type": "article",
                    "title": "First Article in Chapter",
                    "revision": "1234",
                    "timestamp": 122331212312,
                    "content_type": "text/x-wiki"
                    "source-url": "http://en.wikipedia.org/wiki/Main_Page",
                },
                {
                    "type": "article",
                    "title": "Second Article in Chapter",
                    "content_type": "text/x-wiki"
                    "source-url": "http://en.wikipedia.org/wiki/Main_Page",
                }
            ]
        },
    ],
    "licenses": [
        {
            "type": "license",
            "name": "GFDL",
            "mw_license_url": "http://en.wikipedia.org/wiki/Wikipedia:Text_of_the_GNU_Free_Documentat
        }
    ]
}
```

CHAPTER 7

Collection Extension for MediaWiki

7.1 About the *Collection* Extension

The *Collection* extension for MediaWiki allows users to collect articles and generate downloadable version in different formats (PDF, OpenDocument Text etc.) for article collections and single articles.

The extension has been developed for and tested with MediaWiki version 1.14 and later. Some features may not be available with older MediaWikis that don't have the MediaWiki API enabled.

The extension is being developed under the GNU General Public License by PediaPress GmbH in close collaboration with Wikimedia Foundation and the Commonwealth of Learning.

7.2 Prerequisites

If you use a render server the MediaWiki API must be enabled (i.e. just don't override the default value of `true` for `$wgEnableApi` in your `LocalSettings.php`).

7.2.1 Install PHP with cURL support

Currently Collection extension needs PHP with cURL support, see http://php.net/curl

7.3 Installation and Configuration of the Collection Extension

- For *MediaWiki* versions up to and including **1.18**: Download the Collection extension matching your *MediaWiki* version from http://www.mediawiki.org/wiki/Special:ExtensionDistributor/Collection and unpack it into your mediawiki extensions directory:

```
cd /srv/http/wiki/extensions
tar -xzf Collection-MW1.18-113990.tar.gz -C /var/www/mediawiki/extensions
```

- For *MediaWiki* versions **1.19 and newer**: You can checkout the newest code of the *Collection* extension from the Git repository into the `extensions` directory of your *MediaWiki* installation:

```
cd extensions/
git clone https://gerrit.wikimedia.org/r/mediawiki/extensions/Collection
```

- Put this line in your `LocalSettings.php`:

```
require_once("$IP/extensions/Collection/Collection.php");
```

If you intend to use the public render server, you're now ready to go.

7.3.1 Install and Setup a Render Server

Rendering and ZIP file generation is done by a server, which can run separately from the MediaWiki installation and can be shared by different MediaWikis.

If you have a low-traffic MediaWiki you can use the public render server running at http://tools.pediapress.com/mw-serve/. In this case, just keep the configuration variable $wgCollectionMWServeURL (see below) at its default value.

Your MediaWiki must be accessible from the render server, i.e. if your MediaWiki is behind a firewall you cannot use the public render server.

If you can't use the public render server, you'll have to *install mwlib* and *run your own render server*. See http://mwlib.readthedocs.org/ for more information.

Finally you'll have to set `$wgCollectionMWServeURL` in your `LocalSetting.php`:

`$wgCollectionMWServeURL` (string)

Set this to the URL of a render server (see above).

The default is `http://tools.pediapress.com/mw-serve/`, the public render server hosted by PediaPress.

7.3.2 Password protected wikis

Password protected wikis require some more information. You'll have to set the `$wgCollectionMWServeCredentials` variable.

`$wgCollectionMWServeCredentials` (string)

Set this to a string of the form "USERNAME:PASSWORD" (or "USERNAME:PASSWORD:DOMAIN" if you're using LDAP), if the MediaWiki requires to be logged in to view articles. The render server will then login with these credentials using MediaWiki API before doing other requests.

SECURITY NOTICE: If the MediaWiki and the render server communicate over an insecure channel (for example on an unencrypted channel over the internet), please DO NOT USE THIS SETTING, as the credentials will be exposed to eavesdropping!

7.3.3 Advanced Settings

The following variables can be set in `LocalSetting.php`. Most people do not have to change them:

`$wgCollectionMWServeCert` (string) Filename of a SSL certificate in PEM format for the mw-serve render server. This needs to be used for self-signed certificates, otherwise cURL will throw an error. The default is null, i.e. no certificate.

`$wgCollectionFormats` An array mapping names of mwlib writers to the name of the produced format. The default value is:

```
array(
    'rl' => 'PDF',
)
```

i.e. only PDF enabled. If you want to add OpenDocument Text in addition to PDF you can set $wgCollectionFormats to something like this:

```
$wgCollectionFormats = array(
    'rl' => 'PDF',
    'odf' => 'ODT',
);
```

On the public render server tools.pediapress.com, currently the following writers are available:

- docbook: DocBook XML
- odf: OpenDocument Text
- rl: PDF
- xhtml: XHTML 1.0 Transitional

If you're using your own render server, the list of available writers can be listed with the following mwlib command:

```
$ mw-render --list-writers
```

$wgCollectionContentTypeToFilename (array) An array matching content types to filenames for downloaded documents. The default is:

```
$wgCollectionContentTypeToFilename = array(
        'application/pdf' => 'collection.pdf',
        'application/vnd.oasis.opendocument.text' => 'collection.odt',
);
```

$wgCollectionPortletFormats (array) An array containing formats (keys in $wgCollectionFormats) that shall be displayed as "Download as XYZ" links in the "Print/export" portlet. The default value is:

```
array( 'rl' );
```

i.e. there's one link "Download as PDF".

$wgCollectionHierarchyDelimiter (string or null) If not null, treat wiki pages whose title contains the configured delimiter as subpages.

For example, to treat article [[Foo/Bar]] as subpage of article [[Foo]] set this variable to "/". This makes sense e.g. on wikibooks.org, but it's questionable on wikipedia.org (cf. [[AC/DC]]).

The (only) effect is that the display title for subpages in collections is set to the title of the (deepest) subpage. For example, the title of article [[Foo/Bar]] will be displayed/rendered as "Bar".

The defaul value is null, which means that no hierarchy is assumed.

$wgCollectionArticleNamespaces (array) List of namespace numbers for pages which can be added to a collection. Category pages (NS_CATEGORY) are always an exception (all articles in a category are added, not the category page itself). Default is:

```
array(
  NS_MAIN,
  NS_TALK,
  NS_USER,
  NS_USER_TALK,
  NS_PROJECT,
  NS_PROJECT_TALK,
  NS_MEDIAWIKI,
  NS_MEDIAWIKI_TALK,
  100,
```

```
    101,
    102,
    103,
    104,
    105,
    106,
    107,
    108,
    109,
    110,
    111,
);
```

$wgCommunityCollectionNamespace (integer) Namespace for "community collections", i.e. the namespace where non-personal article collection pages are saved.

> Note: This configuration setting is only used if the system message Coll-community_book_prefix has not been set (see below).

Default is `NS_PROJECT`.

$wgCollectionMaxArticles (integer) Maximum number of articles allowed in a collection.

Default is 500.

$wgCollectionLicenseName (string or null) License name for articles in this MediaWiki. If set to `null` the localized version of the word "License" is used.

Default is null.

$wgCollectionLicenseURL (string or null) HTTP URL of an article containing the full license text in wikitext format for articles in this MediaWiki. E.g.

```
$wgCollectionLicenseURL = 'http://en.wikipedia.org/w/index.php?title=Wikipedia:Text_of_the_GNU_F
```

for the GFDL. If set to null, the standard MediaWiki variables $wgRightsPage, $wgRightsUrl and $wgRightsText are used for license information.

If your MediaWiki contains articles with different licenses, make sure that each article contains the name of the license and set $wgCollectionLicenseURL to an article that contains all needed licenses.

$wgCollectionPODPartners (array or false) Array of parameters needed to define print on demand providers:

```
$wgCollectionPODPartners = array(
        'pediapress' => array(
                'name' => 'PediaPress',
                'url' => 'http://pediapress.com/',
                'posturl' => 'http://pediapress.com/api/collections/',
                'infopagetitle' => 'coll-order_info_article',
        ),
);
```

(This is the default.)

`name`, `url` and `posturl` are mandatory parameters to display information on the list of available providers.

If `infopagetitle` is present, it will be interpreted as the MediaWiki message that contains the name of the short information on particular provider. For example, it can be `coll-order_info_mypress` and if the message contains `Help:Books/MyPress order information`, a contents of this page will be used. The message itself can be localized for different languages.

Setting `$wgCollectionPODPartners` to false disables ordering interface altogether.

`$wgEnableWriteAPI`

If you want to let users save their collections as wiki pages, make sure $wgEnableWriteAPI is set to true, i.e. put this line in your `LocalSettings.php`:

```
$wgEnableWriteAPI = true;
```

(This is the default.)

There are two MediaWiki rights that are checked, before users are allowed to save collections: To be able to save collection pages under the User namespace, users must have the right 'collectionsaveasuserpage'; to be able to save collection pages under the community namespace (see $wgCommunityCollectionNamespace), users must have the right 'collectionsaveascommunitypage'. For example, if all logged-in users shall be allowed to save collection pages under the User namespace, but only autoconfirmed users, shall be allowed to save collection pages under the community namespace, add this to your LocalSettings.php:

```
$wgGroupPermissions['user']['collectionsaveasuserpage'] = true;
$wgGroupPermissions['autoconfirmed']['collectionsaveascommunitypage'] = true;
```

You may also want to configure some of the following:

- As the current collection of articles is stored in the session, the session timeout should be set to some sensible value (at least a few hours, maybe one day). Adjust session.cookie_lifetime and session.gc_maxlifetime in your `php.ini` accordingly.
- Add a help page (for example `Help:Books` for wikis in English language).

  A repository of help pages in different languages can be found on Meta-Wiki.

  The name of the help page is stored in the system message Coll-helppage and can be adjusted by editing the wiki page [[MediaWiki:Coll-helppage]].
- Add a template [[Template:saved_book]] which is transcluded on top of saved collection pages. An example for such a template can be found on the English Wikipedia: http://en.wikipedia.org/wiki/Template:Saved_book

  The name of the template can be adjusted via the system message Coll-savedbook_template, i.e. by editing [[MediaWiki:Coll-savedbook_template]].
- To enable ZENO and Okawix export, uncomment the corresponding lines in `$wgCollectionFormats` (file Collection.php). These exports are devoted to the Wikimedia projects and their mirrors.

  They cannot be used on other wikis since they get data and search engine indexes from the cache of wikiwix.com.

7.4 Customization via System Messages

There are several system messages, which can be adjusted for a MediaWiki installation. They can be changed by editing the wiki page [[MediaWiki:SYSTEMMESSAGENAME]], where SYSTEMMESSAGENAME is the name of the system message.

- `Coll-helppage`: The name of the help page (see above).

  The default for English language is `Help:Books`, and there exist translations for lots of different languages.
- `Coll-user_book_prefix`: Prefix for titles of "user books" (i.e. books for personal use, as opposed to "community books"). If the system message is empty or '-' (the default), the title of user book pages is constructed as User:USERNAME/Books/BOOKTITLE. If the system message is set and its content is PREFIX, the title of user book pages is constructed by directly concatenating PREFIX and the BOOKTITLE, i.e. there's no implicitly inserted '/' inbetween!

- `Coll-community_book_prefix`: Prefix for titles of "community books" (cf. "user books" above). If the system message is empty or '-' (the default), the title of community pages is constructed as NAMESPACE:Books/BOOKTITLE, where NAMESPACE depends on the value of $wgCommunityCollectionNamespace (see above). If the system message is set and its content is PREFIX, the title of community book pages is constructed by directly concatenating PREFIX and BOOKTITLE, i.e. there's no implicitly inserted '/' inbetween. Thus it's possible to define a custom namespace 'Book' and set the system message to 'Book:' to produce community book page titles Book:BOOKTITLE.
- `Coll-savedbook_template`: The name of the template (w/out the Template: prefix) included at the top of saved book pages (see above).

  The default is: `saved_book`, and there exist translations for lots of different languages.
- `Coll-bookscategory`: Name of a category (w/out the Category: prefix) to which all saved book pages should be added (optional, set to an empty value or "-" to turn that feature off).
- `Coll-book_creator_text_article`: The name of a wiki page which is transcluded on the "Start book creator" page (the page which is shown when a user clicks on "Create a book").

  The default is: `{{MediaWiki:Coll-helppage}}/Book creator text` i.e. a subpage of the configured help page named "Book creator text"
- `Coll-suggest_enabled`: If set to 1, the suggestion tool is enabled. Any other value will disable the suggestion tool.

  The default is: '1', i.e. the suggestion tool is enabled.
- `Coll-order_info_article`: The name of a wiki page which is included on the Special:Book page to show order information for printed books.

  The default value is: `{{MediaWiki:Coll-helppage}}/PediaPress order information` i.e. a subpage of the configured help page named "PediaPress order information".

  This wiki page is used only if included in the `$wgCollectionPODPartners` configuration.
- `Coll-rendering_page_info_text_article`: The name of a wiki page with additional informations to be displayed when single pages are being rendered.
- `Coll-rendering_collection_info_text_article`: The name of a wiki page with additional informations to be displayed when collections are being rendered.

CHAPTER 8

Changelog

8.1 mwlib

8.1.1 2014-02-19 mwlib 0.15.15

- catch IOError when reading image sizes
- prevent race conditions in rlwriter.toc by using a tmp directory

8.1.2 2014-01-13 mwlib 0.15.14

- set user agent from environment variable MWLIB_USER_AGENT

8.1.3 2014-01-09 mwlib 0.15.13

- add –disable-all-writers argument to nserve
- add note about professional support
- adapt bot filtering a bit

8.1.4 2013-11-11 mwlib 0.15.12

- workaround ‘first run’ tox issue
- fix tests with new wsgi_intercept 0.6 and require that version
- Match IPv6 addresses as anonymous users
- handle __NOGLOSSARY__ magicword
- Fix #37

8.1.5 2013-08-09 mwlib 0.15.11

- fix possible problems on solaris containers in init_tmp_cleaner
- don’t waste people’s lifetime in init_tmp_cleaner
- fix xnet tests

- use os.urandom in utils.uid
- generate junitxml files
- remove empty reference nodes

8.1.6 2013-07-04 mwlib 0.15.10

- add some tests for purge_cache
- don't step into nested directories in purge_cache
- catch errors while examining directory in purge_cache
- only log error if it's not ENOENT in purge_cache
- Add –serve-files-address parameter to nslave.
- make make-manifest useable as pre-commit hook
- is_good_baseurl(): eliminate some false positives

8.1.7 2013-07-02 mwlib 0.15.9

- set timeout for makezip in postman
- remove more template blacklisting/template exclusion handling code
- get rid of template blacklisting/print templates in nslave and postman
- mention that template blacklisting and print templates do not work anymore.
- use tox 1.5's whitelist_externals in order to suppress warnings
- fix imports in test_nserve.py and move it to tests directory

8.1.8 2013-04-23 mwlib 0.15.8

- do not install pil in tox testenv
- install Pillow
- also fetch used images when fetching 'redirected revisions'

8.1.9 2013-04-23 mwlib 0.15.7

- remove explicitly positioned nodes regardless of nesting level. fix bug where children were skipped and not removed

8.1.10 2013-03-26 mwlib 0.15.6

- fix redirect handling when fetching by articles by revision

8.1.11 2013-03-26 mwlib 0.15.5

- fix redirect handling

8.1.12 2013-03-26 mwlib 0.15.4

- fix missing img attribute translation
- remove duplicate coordinates

8.1.13 2013-03-12 mwlib 0.15.3

- fix nserve, nslave, postman

8.1.14 2013-03-12 mwlib 0.15.2

- use post request when posting text to action=expandtemplates

8.1.15 2013-03-12 mwlib 0.15.1

- fix mw-serve-ctl

8.1.16 2013-03-12 mwlib 0.15.0

Note: you'll have to adapt your start scripts, some programs have been renamed!

Note: Unfortunately the 'template blacklisting' and 'print templates' functionality had to be removed in order to support the scribunto extension. The documentation has not been updated and may still mention those features.

- nslave.py, nserve.py, postman.py have been renamed to nslave, nserve and postman
- require python 2.6, python 2.5 isn't supported anymore
- fetch expanded articles
- force pyparsing < 2
- remove open street maps used in wikivoyage - they can't be rendered currently
- fix for missing revid attribute
- fix and improve wikivoyage tagextensions
- allow item lists in div
- transform single-col, single-row table into div, even if it is an "infobox"
- tweak region lists for wikivoyage
- fix bug for article http://en.wikivoyage.org/wiki/Africa (and possibly more from wikivoyage)
- quick hack to expand the {{REVISIONID}}

8.1.17 2012-12-04 mwlib 0.14.3

- prefer UTF-8 locales for use in formatnum

8.1.18 2012-12-03 mwlib 0.14.2

- remove byte order mark (bom) in _do_request
- return unicode from formatnum
- improve table border code
- add noprint css class “rellink”

8.1.19 2012-09-24 mwlib 0.14.1

- implement locale aware formatnum
- implement wikipedia’s braindamaged scientific notation
- adapt single col splitting heuristics of treecleaner

8.1.20 2012-06-18 mwlib 0.14.0

- get rid of the _Version class, up version to 0.14.0
- install scripts via plain old distutils instead of “console_scripts” entry point
- remove cdbwiki
- remove mwlib.xfail, use pytest.mark.xfail instead
- expect setuptools or distribute to be installed
- remove some problematic dependencies in PP_MAINTAINER mode

8.1.21 2012-06-18 mwlib 0.13.11

- skip checkpil if PP_MAINTAINER is set
- relax simplejson requirement a bit
- fix content disposition header when filenames contain commas
- make it easier to test the content disposition logic

8.1.22 2012-06-17 mwlib 0.13.10

- fix handling of filenames with spaces

8.1.23 2012-06-17 mwlib 0.13.9

- use filenames derived from content for downloads
- synchronize documentation with MediaWiki

8.1.24 2012-06-11 mwlib 0.13.8

- do not embed apipkg anymore
- make sure temp files are removed even if mw-render is killed

8.1.25 2012-05-08 mwlib 0.13.7

- unconditionally require simplejson
- workaround a inspect module bug
- fix pypi url used by tox
- improve transformSingleColTables in treecleaner
- expose DumpParser's redirect-ignoring functionality as an optional boolean command-line flag to mw-buildcdb

8.1.26 2012-03-07 mwlib 0.13.6

- make mw-zip -gg post test.pediapress.com
- implement protocol relative urls in named links

8.1.27 2012-02-29 mwlib 0.13.5

- simplify the brain-damaged iferror_rx regular expression, fixes #10
- support syntaxhighlight nodes

8.1.28 2012-02-15 mwlib 0.13.4

- require qserve >= 0.2.7 in order to be compatible with the latest gevent
- move our custom argument parser to mwlib
- prefer simplejson to json
- allow nserve to listen on a specific interface with -i/–interface
- fix styleutils: limit rgb values to [0,1]
- remove mw-watch in setup.py

8.1.29 2012-01-12 release 0.13.3

- fix pagename when expanding <pages> tag
- handle the case where NAMESPACE is called as a template
- get rid of lxml warnings

8.1.30 2012-01-11 release 0.13.2

- add support for adding spacing for cjk text
- add initial support for the pages tag
- protect page-break info from removal in divs and spans

8.1.31 2011-12-13 release 0.13.1

- replaced mw-serve with nserve.py
- removed CGI support
- removed lots of obsolete code
- updated documentation, available online at http://mwlib.readthedocs.org

8.1.32 2011-10-24 release 0.12.17

- handle siteinfo without "magicwords" key in templ.parser
- use gevent instead of twisted in mw-zip/mw-render
- show memory usage in mw-zip
- use sqlite3dmb to store html
- fix directionality of math nodes for RTL documents

8.1.33 2011-08-31 release 0.12.16

- remove xhtmlwriter
- remove docbookwriter
- fix_wikipedia_siteinfo for kdb, ltg and xmf
- remove zipwiki
- implement safesubst
- match noinclude and onlyinclude tags with whitespace
- bail out when running setup.py with an unsupported python version

8.1.34 2011-08-12 release 0.12.15

- require lxml.
- dont switch fonts for direction switch chars lrm/rlm
- set teletype style by css
- fix rtl direction check bug
- quick fix in order to support the kbd tag.
- fix switch statements with localized #default case.
- dont remove direction switching nodes

- resolve aliases when expanding templates.
- support localized parser functions.
- make tests work with latest py.test 2.1.
- add support for css direction switching
- Code and Var nodes now use teletype style
- be more verbose when collection params can not be retrieved
- fix subpage links (bugzilla #28055)
- fix for https://bugzilla.wikimedia.org/show_bug.cgi?id=29354
- dont die on treecleaner errors
- remove paragraphs from galleries
- add license templates
- get rid of some more parsing calls
- cache img display info in licensehandler
- speed up getting template args (for licensehandling)
- always show full text of contributors of images
- fix for getAllDisplayText
- add nofilter to licensehandling
- make licensechecker less fragile to bad config format
- improve image license handling
- improve stats for licensechecker
- add custom element to metabook
- dont throw away collapsible boxes. fixes: #935
- decrease api_request_limit
- limit max. simultaneous img downloads to 15
- moar categories. less whitespace. untangle revision/category fetching
- increase standard resolution of images
- fix getting html with revisions
- clean up after fixNesting
- fetch extension images
- prevent adding same api url twice
- retry failed img downloads
- workaround for missing descriptionurl
- fix: descriptionurl returned from api seems be “false” sometimes.
- fix for #925. make syntaxhighlighting work again
- fix for #755
- support older mediawikis

- add lower bound on word splitting hints
- mwlib.refine: parse <caption> tags inside tables
- be more generous when trying to detect see also
- fix for “See Also “Section removal
- fix #905: remove See also sections.
- remove edit links
- magics.py: handle second argument to fullurl magic function.
- convert tiff images to png
- fix for infobox detection
- handle Abbreviation node in xhtmlwriter
- add Abbreviation node
- improve table splitting

8.1.35 2010-10-29 release 0.12.14

- magics.py: fix NS magic function.
- refine/core.py: do not parse links if link target would contain newlines.
- setup.py: require lockfile==0.8.
- add xr formatting in #time
- replace mwlib.async with qserve package.
- move fontswitcher to writer dir
- remove collapsible elements
- fix for #830
- move gallery nodes out of tables.
- handle overflow:auto crap
- fix for reference handling
- better handling for references nodes.
- fix for ReferenceLists
- fix whitespace handling and implicit newlines in template arguments. fixes http://code.pediapress.com/wiki/ticket/877.
- Add support for more PageMagic as per http://meta.wikimedia.org/wiki/Help:Magic_words
- Fix PageMagic to consider page as argument
- fetch parsed html from mediawiki and store it as parsed_html.json. We store the raw result from mediawiki since it’s not clear what’s really needed.
- make mwapi work for non query actions.

8.1.36 2010-7-16 release 0.12.13

- omit passwords from error file
- make login work with latest mediawiki.
- use content_type, not content-type in metabooks
- filter crap from ref node names
- try to set GDFONTPATH to some sane value. call EasyTimeline with font argument.
- do not scale easytimeline images after rendering rather scale then in EasyTimeline.pl
- update EasyTimeline to 1.13
- another fix for nested references
- fix for broken tables
- make #IFEXIST handle images
- add treecleaner method to avoid large cells
- fix img alignment
- fix nesting of section with same level
- do not let tablemode get negative.
- fix #815
- call fix_wikipedia_siteinfo based on contents of server (instead of sitename)
- workaround for broken interwikimap. fixes #807
- handle the case, where the
 ends up in a new paragraph. fixes #804
- move the poem tag implementation to mwlib.refine.core and make it expand templates
- add #ifeq node. fixes #800
- fix for images with spaces in file extensions
- fix and test for #795
- pull tables out of DefinitionDescriptions
- add getVerticalAlign to styleutils
- remove tables from image captions
- remove –clean-cache option to mw-serve
- allow floats as –purge-cache argument
- workaround for buggy lockfile module.
- implement DISPLAYTITLE
- generate higher resolution timelines
- handle abbr and hiero tags
- make sure print_template_pattern is written to nfo.json, when getting it as part of the collection params
- relax odfpy requirement a bit
- make hash-mark only links work again
- remove empty images

8.1.37 2009-12-16 release 0.12.12

- dont remove sections containing only images.
- improve handling of galleries
- fix use of uninitialized last variable
- do not 'split' links when expanding templates
- quick workaround for http://code.pediapress.com/wiki/ticket/754

8.1.38 2009-12-8 release 0.12.11

- *beware* python 2.4 is not supported anymore
- parse paragraphs before spans
- parse named urls before links.
- fix urllinks inside links
- fix named urls inside double brackets
- avoid splitting up Reference nodes.
- parse lines/lists before span.
- add getScripts method. improve rtl compat. for fontswitching
- do not replace uniq strings with their content when preprocessing gallery tags. fixes e.g. ref tags inside gallery tags.
- run template expansion for each line in gallery tags
- handle mhr, ace, ckb, mwl interwiki links
- add clearStyles method
- add another condition to avoid single col tables in border-boxes
- refactor node style handling
- remove fixInfoBoxes from treecleaner
- fix for identifiying image license information
- handle closing ul/ol tags inside enumerations
- correctly determine text alignment of node.
- fix for image only table check
- add code for simple rpc servers/clients based on the gevent library.
- add flag for split itemlists
- do not blacklist articles
- add upper limit for font sizes

8.1.39 2009-10-20 release 0.12.10

- fix race condition when fetching siteinfo
- introduce flag to suppress automatic escaping when cleaning text
- sent error mails only once
- add 'pageby', 'uml', 'graphviz', 'categorytree', 'summary' to list of tags to ignore

8.1.40 2009-10-13 release 0.12.9

- fix #709
- allow higher resolution in math formulas
- fetch collection parameters and use them (template exclusion category,...)
- fix #699
- fix <ref> inside table caption
- refactor filequeue
- adjust table splitting parameter
- move invisible, named references out of table nodes
- fix late #if
- fix bug with inputboxes
- fix parsing of collection pages: titles/subtitles may but do not need to have spaces
- use new default license URL
- fix race condition in mw-serve/mw-watch

8.1.41 2009-9-25 release 0.12.8

- fix argument handling in mw-serve Previously it had been possible to overwrite any file by passing arguments containing newlines to mw-serve.

8.1.42 2009-9-23 release 0.12.7

- ensure that files extracted from zip files end up in the destination directory.

8.1.43 2009-9-15 release 0.12.6

- fix for reference nodes
- allow most characters in urls
- fix for setting content-length in response
- fix problem with blacklisted templates creating preformatted nodes (#630)
- do not split preformatted nodes on non-empty whitespace only lines
- do not create preformatted nodes inside li tags

- pull garbage out of table rows. fix #17.
- dont remove empty spans if an explicit size is given.
- uncomment fix_wikipedia_siteinfo and add pnb as interwiki link
- remove mwxml writer.
- add mw-version program

8.1.44 2009-9-8 release 0.12.5

- fix missing page case in get_page when looking for redirects
- some minor bugfixes

8.1.45 2009-8-25 release 0.12.3

- better compatibility with older mediawiki installations

8.1.46 2009-8-18 release 0.12.2

- fix status callbacks to pod partner

8.1.47 2009-8-17 release 0.12.1

- added mw-client and mw-check-service
- mw-serve-ctl can now send report mails
- fixes for race conditions in mwlib.filequeue (mw-watch)
- lots of other improvements...

8.1.48 2009-5-6 release 0.11.2

- fixes

8.1.49 2009-5-5 release 0.11.1

- merge of the nuwiki branch: better, faster resource fetching with twisted_api, new ZIP file format with nuwiki

8.1.50 2009-4-21 release 0.10.4

- fix chapter handling
- fix bad #tag params

8.1.51 2009-4-17 release 0.10.3

- fix issue with self-closing tags
- fix issue with "disappearing" table rows

8.1.52 2009-4-15 release 0.10.2

- fix for getURL() method in zipwiki

8.1.53 2009-4-9 release 0.10.1

- the parser has been completely rewritten (mwlib.refine)
- fix bug in recorddb.py: do not overwrite articles
- removed mwapidb.WikiDB.getTemplatesForArticle() which was broken and wasn't used.

8.1.54 2009-3-5 release 0.9.13

- normalize template names when checking against blacklist
- make NAMESPACE magic work for non-main namespaces
- make NS template work

8.1.55 2009-03-02 release 0.9.12

- fix template expansion bug with non self-closing ref tags containing equal signs

8.1.56 2009-2-25 release 0.9.11

- added –print-template-pattern
- fix bug in LOCALURLE with non-ascii characters (#473)
- fix 'upright' image modifier handling (#459)
- allow star inside URLs (#483)
- allow whitespace in image width modifiers (#475)

8.1.57 2009-2-19 release 0.9.10

- do not call check() in zipcreator: better some missing articles than an error message

8.1.58 2009-2-18 release 0.9.8

- localize image modifiers
- fix bug in serve with forced rendering
- fix bug in writerbase when no URL is returned
- return only unqiue image contributors, sorted
- #expr with whitespace only argument now returns the empty string instead of marking the result as an error.
- added mw-serve-ctl command line tool (#447)
- mwapidb: omit title in URLs with oldid
- mwapidb: added getTemplatesForArticle()
- zipcreator: check articles and sources to prevent broken ZIP files
- mwapidb: do query continuation to find out all authors (#420)
- serve: use a deterministic checksum for metabooks (#451)

8.1.59 2009-2-9 release 0.9.7

- fix bug in #expr parsing
- fix bug in localised namespace handling/#ifexist
- fix bug in redirect handling together with specific revision in mwapidb

8.1.60 2009-2-3 release 0.9.6

- mwapidb: return authors alphabetically sorted (#420)
- zipcreator: fixed classname from DummyScheduler to DummyJobScheduler; this bug broke the –no-threads option
- serve: if rendering is forced, don't re-use ZIP file (#432)
- options: remove default value "Print" from –print-template-prefix
- mapidb: expand local* functions, add them to source dictionary
- expander: fix memory leak in template parser (#439)
- expander: better noinclude, includeonly handling (#426)
- expander: #iferror now uses a regular expression (#435)
- expander: workaround dateutils bug (resulting in a TypeError: unsupported operand type(s) for +=: 'NoneType' and 'int')

8.1.61 2009-1-26 release 0.9.5

- initial release

CHAPTER 9

Indices and tables

- genindex
- search