Sphinx Confluence Builder Documentation

Anthony Shaw and Contributors

Feb 01, 2020

Contents

Sphinx extension to build Confluence storage format files and optionally publish them to a Confluence instance.

- License: BSD-2-Clause

- Confluence Cloud or Server 6.7+ with Python 2.7 or 3.5+

- Home: https://github.com/sphinx-contrib/confluencebuilder

- Documentation: https://sphinxcontrib-confluencebuilder.readthedocs.io

Contents

documentation contents

1.1 installing

Atlassian Confluence Builder for Sphinx depends on:

- Python 2.7 or 3.5+
- Requests 2.14.0 or later
- Sphinx 1.6.3 or later
- Confluence Cloud or Server 6.7+

The recommended method of installation is using pip.

```
pip install sphinxcontrib-confluencebuilder
```

To verify the package has been installed, if desired, the following command can be used:

```
python -m sphinxcontrib.confluencebuilder --version
```

1.1.1 fresh quick-start

The following provides a series of steps to assist in preparing a new environment to use this package. This quick-start will aim to use the most recent version of Python.

Note: If the installation process fails with the following error "AttributeError: '_NamespacePath' object has no attribute 'sort'", try upgrading the setuptools module:

```
python -m pip install --upgrade setuptools
```

linux

While the use of Python/pip is almost consistent between Linux distributions, the following are a series of helpful steps to install this package under specific distributions of Linux. From a terminal, invoke the following commands:

arch

```
$ sudo pacman -Sy
$ sudo pacman -S python-pip
$ (optional) sudo pacman -S python-virtualenv
$ (optional) virtualenv sphinx-venv
$ (optional) source sphinx-venv/bin/activate
$ pip install sphinxcontrib-confluencebuilder
$ python -m sphinxcontrib.confluencebuilder --version
sphinxcontrib.confluencebuilder <version>
```

centos

```
$ sudo yum install epel-release
$ sudo yum install python-pip
$ (optional) sudo yum install python-virtualenv
$ (optional) virtualenv sphinx-venv
$ (optional) source sphinx-venv/bin/activate
$ pip install sphinxcontrib-confluencebuilder
$ python -m sphinxcontrib.confluencebuilder --version
sphinxcontrib.confluencebuilder <version>
```

fedora

```
$ sudo dnf install python-pip
$ (optional) sudo dnf install python-virtualenv
$ (optional) virtualenv sphinx-venv
$ (optional) source sphinx-venv/bin/activate
$ pip install sphinxcontrib-confluencebuilder
$ python -m sphinxcontrib.confluencebuilder --version
sphinxcontrib.confluencebuilder <version>
```

ubuntu

```
$ sudo apt-get update
$ sudo apt-get install python-pip
$ (optional) sudo dnf install python-virtualenv
$ (optional) virtualenv sphinx-venv
$ (optional) source sphinx-venv/bin/activate
$ pip install sphinxcontrib-confluencebuilder
$ python -m sphinxcontrib.confluencebuilder --version
sphinxcontrib.confluencebuilder <version>
```

os x

From a terminal, invoke the following commands:

```
$ sudo easy_install pip
$ (optional) pip install virtualenv
$ (optional) virtualenv sphinx-venv
$ (optional) source sphinx-venv/bin/activate
$ pip install sphinxcontrib-confluencebuilder
$ python -m sphinxcontrib.confluencebuilder --version
sphinxcontrib.confluencebuilder <version>
```

windows

If not already installed, download the most recent version of Python:

> Python - Downloads
>
> https://www.python.org/downloads/

When invoking the installer, it is recommended to select the option to "Add Python to PATH"; however, users can explicitly invoked Python from an absolute path (the remainder of these steps will assume Python is available in the path).

While optional, it is recommended to install `virtualenv` first. Open a Windows command prompt as an administrator. Invoke the following:

```
(optional) pip install virtualenv
```

The command prompt started as an administrator can be closed.

Open a Windows command prompt (administrator mode is not required). Invoke the following:

```
(optional) virtualenv sphinx-venv
(optional) source sphinx-venv\Scripts\activate.bat
python -m pip install sphinxcontrib-confluencebuilder
python -m sphinxcontrib.confluencebuilder --version
```

1.1.2 master

To install the bleeding edge sources, the following pip command can be used:

```
pip install \
    git+https://github.com/sphinx-contrib/confluencebuilder.git
```

1.2 tutorial

After *installing* Atlassian Confluence Builder for Sphinx, a Sphinx project can be configured to support the `confluence` builder. The following tutorial will provide a series of steps which will:

- Enables a builder to generate Confluence-compatible markup documentation.
- Enables a builder to publish to a Confluence instance.

- *new documentation*
  - *quick-start*
  - *minimalistic*
- *existing documentation*

1.2.1 new documentation

If starting without an existing Sphinx-based documentation, one can create a minimalistic Sphinx configuration or use Sphinx's quick-start utility.

quick-start

If opting for the quick-start utility, open a terminal to the location where documentation should be generated and invoke the following:

```
sphinx-quickstart
    (or)
python -m sphinx.cmd.quickstart
    (or)
python -m sphinx.quickstart
```

After completing the quick-start, `conf.py` can be tweaked as desired. Continue preparing this project's configuration by consulting the *existing documentation* steps (below).

minimalistic

For a minimalistic setup, create a new folder for the new documentation and configuration to be used. Create a document named `index.rst` with the following content:

```
my documentation
================

This is a test document.
```

Next, create a configuration file with the following information:

```
# -*- coding: utf-8 -*-

extensions = ['sphinxcontrib.confluencebuilder']
```

After preparing these assets, consult the *existing documentation* steps (below) to complete the configuration.

1.2.2 existing documentation

Enable this extension's builder by adding the extension to the target project's Sphinx configuration (`conf.py`):

```
extensions = ['sphinxcontrib.confluencebuilder']
```

Next, include a series of publish-related settings to the configuration file:

```
confluence_publish = True
confluence_space_name = 'TEST'
# (for confluence cloud)
confluence_server_url = 'https://example.atlassian.net/wiki/'
confluence_server_user = 'myawesomeuser@example.com'
confluence_server_pass = 'myapikey'
# (or for confluence server)
confluence_server_url = 'https://intranet-wiki.example.com/'
confluence_server_user = 'myawesomeuser'
confluence_server_pass = 'mypassword'
```

Make appropriate changes to the above configuration for the environment being targeted.

Tip: For more information on the above or additional configuration options, consult *all configuration options*.

If one wishes to publish documents as children of a parent page inside a space, the configuration `confluence_parent_page` (*jump*) should be supplied with the name of the page to append published documents. If omitted, the builder will publish documents in the root of the space. For example:

```
confluence_parent_page = 'MyDocumentation'
```

To process and publish the documentation set, invoke Sphinx with the `confluence` builder to perform building/publishing:

```
make confluence
    (or)
sphinx-build -b confluence . _build/confluence -E -a
    (or)
python -m sphinx -b confluence . _build/confluence -E -a
```

Documentation of the project should now be published to the Confluence site.

1.3 configuration

The following is an example of simple configuration for Confluence generation and publishing:

```
extensions = ['sphinxcontrib.confluencebuilder']
confluence_publish = True
confluence_space_name = 'TEST'
confluence_parent_page = 'Documentation'
confluence_server_url = 'https://intranet-wiki.example.com/'
confluence_server_user = 'username'
confluence_server_pass = 'password'
```

All Atlassian Confluence Builder configurations are prefixed with `confluence_`. View the entire list of configuration options below.

- *essential configuration*
  - *confluence_publish*
  - *confluence_server_pass*

1.3.1 essential configuration

confluence_publish

A boolean that decides whether or not to allow publishing. This option must be explicitly set to `True` if one wishes to publish content. By default, the value is set to `False`.

```
confluence_publish = True
```

confluence_server_pass

The password value used to authenticate with the Confluence instance. If using Confluence Cloud, it is recommended to use an API token for the configured username value (see API tokens):

```
confluence_server_pass = 'vsUsrSZ6Z4kmrQMapSXBYkJh'
```

If API tokens are not being used, the plain password for the configured username value should be used:

```
confluence_server_pass = 'myawesomepassword'
```

Caution: It is never recommended to store an API token or raw password into a committed/shared repository holding documentation. A documentation's configuration can modified various ways with Python to pull an authentication token for a publishing event (reading from a local file, acquiring a password from `getpass`, etc.). If desired, this extension provides a method for prompting for a password (see `confluence_ask_password`).

confluence_server_url

Warning: XML-RPC API has been deprecated by Atlassian as well as in this extension. XML-RPC API support will be removed from this extension in v1.3.

The URL for Confluence. The URL should be prefixed with `https://` or `http://` (depending on the URL target). The target API folder should not be included in the URL (for example, excluding `/rest/api/` or `/rpc/xmlrpc/`). For a Confluence Cloud instance, an example URL configuration is as follows:

```
confluence_server_url = 'https://example.atlassian.net/wiki/'
```

For a Confluence Server instance, an example URL configuration, if the instance's REST API is `https://intranet-wiki.example.com/rest/api/` or XML-RPC API is at `https://intranet-wiki.example.com/rpc/xmlrpc`, should be as follows:

```
confluence_server_url = 'https://intranet-wiki.example.com/'
```

confluence_server_user

The username value used to authenticate with the Confluence instance. If using Confluence Cloud, this value will most likely be the account's E-mail address. If using Confluence instance, this value will most likely be the username value.

```
confluence_server_user = 'myawesomeuser@example.com'
    (or)
confluence_server_user = 'myawesomeuser'
```

confluence_space_name

Key of the space in Confluence to be used to publish generated documents to.

```
confluence_space_name = 'MyAwesomeSpace'
```

Note that the space name can be **case-sensitive** in most (if not all) versions of Confluence.

1.3.2 generic configuration

confluence_add_secnumbers

New in version 1.2.

Add section numbers to page and section titles if `doctree` uses the `:numbered:` option. By default, this is enabled:

```
confluence_add_secnumbers = True
```

See also *confluence_publish_prefix*.

confluence_header_file

The name of the file to use header data. If provided, the contents found inside the header file will be added to the start of all generated documents. The file path provided should be relative to the build environment's source directory. For example:

```
confluence_header_file = 'assets/header.tpl'
```

See also *confluence_footer_file*.

---

confluence_footer_file

The name of the file to use footer data. If provided, the contents found inside the footer file will be added at the end of all generated documents. The file path provided should be relative to the build environment's source directory. For example:

```
confluence_footer_file = 'assets/footer.tpl'
```

See also *confluence_header_file*.

---

confluence_max_doc_depth

An integer value, if provided, to indicate the maximum depth permitted for a nested child page before its contents is inlined with a parent. The root of all pages is typically the configured master_doc. The root page is considered to be at a depth of zero. By defining a value of `0`, all child pages of the root document will be merged into a single document. By default, the maximum document depth is disabled with a value of `None`.

```
confluence_max_doc_depth = 2
```

---

confluence_page_hierarchy

A boolean value to whether or not nest pages in a hierarchical ordered. The root of all pages is typically the configured master_doc. If a master_doc instance contains a toctree, listed documents will become child pages of the master_doc. This cycle continues for child pages with their own toctree markups. By default, the hierarchy mode is disabled with a value of `False`.

```
confluence_page_hierarchy = False
```

Note that even if hierarchy mode is enabled, the configured master_doc page and other published pages that are not defined in the complete toctree, these documents will still be published based off the configured (or unconfigured) *confluence_parent_page* setting.

---

confluence_prev_next_buttons_location

New in version 1.2.

A string value to where to include previous/next buttons (if any) based on the detected order of documents to be included in processing. Values accepted are either `bottom`, `both`, `top` or `None`. By default, no previous/next links are generated with a value of `None`.

```
confluence_prev_next_buttons_location = 'top'
```

confluence_secnumber_suffix

New in version 1.2.

The suffix to put after section numbers, before section name.

```
confluence_secnumber_suffix = '. '
```

See also *confluence_add_secnumbers*.

1.3.3 publishing configuration

confluence_ask_password

> **Warning:** User's running Cygwin/MinGW may need to invoke with `winpty` to allow this feature to work.

Provides an override for an interactive shell to request publishing documents using an API key or password provided from the shell environment. While a password is typically defined in the option `confluence_server_pass` (either directly set/fetched from the project's `config.py` or passed via a command line argument `-D confluence_server_pass=password`), select environments may wish to provide a way to provide an authentication token without needing to modify documentation sources or having a visible password value in the interactive session requesting the publish event. By default, this option is disabled with a value of `False`.

```
confluence_ask_password = False
```

A user can request for a password prompt by invoking build event by passing the define through the command line:

```
sphinx-build [options] -D confluence_ask_password=1 <srcdir> <outdir>
```

Note that some shell sessions may not be able to pull the password value properly from the user. For example, Cygwin/MinGW may not be able to accept a password unless invoked with `winpty`.

confluence_ask_user

New in version 1.2.

Provides an override for an interactive shell to request publishing documents using a user provided from the shell environment. While a user is typically defined in the option `confluence_server_user`, select environments may wish to provide a way to provide a user without needing to modify documentation sources. By default, this option is disabled with a value of `False`.

```
confluence_ask_user = False
```

---

confluence_disable_autogen_title

A boolean value to explicitly disable the automatic generation of titles for documents which do not have a title set. When this extension processes a set of documents to publish, a document needs a title value to know which Confluence page to create/update. In the event where a title value cannot be extracted from a document, a title value will be automatically generated for the document. For automatically generated titles, the value will always be prefixed with `autogen-`. For users who wish to ignore pages which have no title, this option can be set to `True`. By default, this option is set to `False`.

```
confluence_disable_autogen_title = True
```

---

confluence_disable_notifications

A boolean value which explicitly disable any page update notifications (i.e. treats page updates from a publish request as minor updates). By default, notifications are enabled with a value of `False`.

```
confluence_disable_notifications = True
```

---

confluence_master_homepage

A boolean value to whether or not force the configured space's homepage to be set to the page defined by the Sphinx configuration's master_doc. By default, the master_doc configuration is ignored with a value of `False`.

```
confluence_master_homepage = False
```

---

confluence_parent_page

The root page found inside the configured space (*confluence_space_name*) where published pages will be a descendant of. The parent page value is used to match with the title of an existing page. If this option is not provided, pages will be published to the root of the configured space. If the parent page cannot be found, the publish attempt will stop with an error message. For example, the following will publish documentation under the `MyAwesomeDocs` page:

```
confluence_parent_page = 'MyAwesomeDocs'
```

If a parent page is not set, consider using the *confluence_master_homepage* option as well. Note that the page's name can be case-sensitive in most (if not all) versions of Confluence.

confluence_publish_postfix

New in version 1.2.

If set, the postfix value is added to the title of all published documents. In Confluence, page names need to be unique for a space. A postfix can be set to either:

- Add a unique naming schema to generated/published documents in a space which has manually created pages; or,
- Allow multiple published sets of documentation, each with their own postfix value.

An example publish postfix is as follows:

```
confluence_publish_postfix = '-postfix'
```

By default, no postfix is used. See also *confluence_publish_prefix*.

confluence_publish_prefix

If set, the prefix value is added to the title of all published documents. In Confluence, page names need to be unique for a space. A prefix can be set to either:

- Add a unique naming schema to generated/published documents in a space which has manually created pages; or,
- Allow multiple published sets of documentation, each with their own prefix value.

An example publish prefix is as follows:

```
confluence_publish_prefix = 'prefix-'
```

By default, no prefix is used. See also *confluence_publish_postfix*.

confluence_purge

> **Warning:** Publishing individual/subset of documents with this option may lead to unexpected results.

A boolean value to whether or not purge legacy pages detected in a space or parent page. By default, this value is set to `False` to indicate that no pages will be removed. If this configuration is set to `True`, detected pages in Confluence that do not match the set of published documents will be automatically removed. If the option *confluence_parent_page* is set, only pages which are a descendant of the configured parent page can be removed; otherwise, all pages in the configured space could be removed.

```
confluence_purge = False
```

While this capability is useful for updating a series of pages, it may lead to unexpected results when attempting to publish a single-page update. The purge operation will remove all pages that are not publish in the request. For example, if an original request publishes ten documents and purges excess documents, a following publish attempt with only one of the documents will purge the other nine pages.

---

confluence_purge_from_master

A boolean value to which indicates that any purging attempt should be done from the root of a published master_doc page (instead of a configured parent page; i.e. `confluence_parent_page`). In specific publishing scenarios, a user may wish to publish multiple documentation sets based off a single parent/container page. To prevent any purging between multiple documentation sets, this option can be set to `True`. When generating legacy pages to be removed, this extension will only attempt to populate legacy pages based off the children of the master_doc page. This option still requires `confluence_purge` to be set to `True` before taking effect.

```
confluence_purge_from_master = False
```

---

confluence_timeout

Force a timeout (in seconds) for network interaction. The timeout used by this extension is not explicitly configured (i.e. managed by Requests and other implementations). By default, assume that any network interaction will not timeout. Since the target Confluence instance is most likely to be found on an external server, is it recommended to explicitly configure a timeout value based on the environment being used. For example, to configure a timeout of ten seconds, the following can be used:

```
confluence_timeout = 10
```

1.3.4 advanced publishing configuration

confluence_asset_override

Provides an override for asset publishing to allow a user publishing to either force re-publishing assets or disable asset publishing. This extension will attempt to publish assets (images, downloads, etc.) to pages via Confluence's attachment feature. Attachments are assigned a comment value with a hash value of a published asset. If another publishing event occurs, the hash value is checked before attempting to re-publish an asset. In unique scenarios, are use may wish to override this ability. By configuring this option to `True`, this extension will always publish asset files (whether or not an attachment with a matching hash exists). By configuring this option to `False`, no assets will be published by this extension. By default, this automatic asset publishing occurs with a value of `None`.

```
confluence_asset_override = None
```

---

confluence_ca_cert

Provide a CA certificate to use for server certificate authentication. The value for this option can either be a file of a certificate or a path pointing to an OpenSSL-prepared directory. If configured to use REST API (default), refer to the Requests SSL Cert Verification documentation (`verify`) for information. If configured to use the XML-RPC API, refer to Python's TLS/SSL wrapper for socket object (`cafile` or `capath`) for more information. If server verification is explicitly disabled (see `confluence_disable_ssl_validation`), this option is ignored. By default, this option is ignored with a value of `None`.

```
confluence_ca_cert = 'ca.crt'
```

confluence_client_cert

Provide a client certificate to use for two-way TLS/SSL authentication. The value for this option can either be a file (containing a certificate and private key) or as a tuple where both certificate and private keys are explicitly provided. If a private key is protected with a passphrase, a user publishing a documentation set will be prompted for a password (see also `confluence_client_cert_pass`). By default, this option is ignored with a value of `None`.

```
confluence_client_cert = 'cert_and_key.pem'
# or
confluence_client_cert = ('client.cert', 'client.key')
```

confluence_client_cert_pass

Provide a passphrase for `confluence_client_cert`. This prevents a user from being prompted to enter a passphrase for a private key when publishing. If a configured private key is not protected by a passphrase, this value will be ignored. By default, this option is ignored with a value of `None`.

```
confluence_client_cert_pass = 'passphrase'
```

confluence_disable_rest

Deprecated since version 1.2: It is not recommended to use this option with a value of `True` as the XML-RPC API has been deprecated by Atlassian as well as in this extension. This configuration option will have no effect in v1.3. Only use if required.

A boolean value to explicitly disable any REST API calls. This extension has the ability to publish using either Confluence's REST or XML-RPC API calls. When publishing, this extension will first attempt to publish using REST and fallback to using XML-RPC. If the target Confluence instance cannot use REST for publishing, it is recommended to set the option to `True` to always use XML-RPC instead. By default, this option is set to `False`.

```
confluence_disable_rest = False
```

confluence_disable_ssl_validation

> **Warning:** It is not recommended to use this option.

A boolean value to explicitly disable verification of server SSL certificates when making a publish request. By default, this option is set to `False`.

```
confluence_disable_ssl_validation = False
```

---

confluence_disable_xmlrpc

Deprecated since version 1.2: XML-RPC API has been deprecated by Atlassian as well as in this extension. XML-RPC API support will be removed from this extension in v1.3.

A boolean value to explicitly disable any XML-RPC API calls. This extension has the ability to publish using either Confluence's REST or XML-RPC API calls. When publishing, this extension will first attempt to publish using REST and fallback to using XML-RPC. If the target Confluence instance supports REST or has XML-RPC explicitly disabled, it is recommended to set this option to `True`. By default, this option is set to `False`.

```
confluence_disable_xmlrpc = False
```

---

confluence_parent_page_id_check

The page identifier check for *confluence_parent_page*. By providing an identifier of the parent page, both the parent page's name and identifier must match before this extension will publish any content to a Confluence instance. This serves as a sanity-check configuration for the cautious.

```
confluence_parent_page_id_check = 1
```

---

confluence_proxy

Deprecated since version 1.2: This proxy configuration only applies to XML-RPC API which has been deprecated by Atlassian as well as in this extension. Setting this option in v1.3 of this extension will have no effect.

Provide the proxy needed to be used to interact with the Confluence instance over the network. At this time, the proxy configuration only applies to XML-RPC calls (REST calls use the Requests library which will use system-defined proxy configuration).

```
confluence_proxy = 'myawesomeproxy:8080'
```

---

confluence_publish_subset

Note: If `confluence_publish_subset` is configured, this option disables *confluence_purge*.

Provides the ability for a publisher to explicitly list a subset of documents to be published to a Confluence instance. When a user invokes sphinx-build, a user has the ability to process all documents (by default) or specifying individual filenames which use the provide files and detected dependencies. If the Sphinx-detected set of documents to process contain undesired documents to publish, `confluence_publish_subset` can be used to override this. Defined document names should be a relative file path without the file extension. For example:

```
confluence_publish_subset = ['index', 'foo/bar']
```

A user can force a publishing subset through the command line:

```
sphinx-build [options] -D confluence_publish_subset=index,foo/bar \
    <srcdir> <outdir> index.rst foo/bar.rst
```

By default, this option is ignored with a value of `[]`. See also *manage publishing a document subset*.

confluence_server_auth

An authentication handler which can be directly provided to a REST API request. REST calls in this extension use the Requests library, which provide various methods for a client to perform authentication. While this extension already provided simple authentication support (via `confluence_server_user` and `confluence_server_pass`), a publisher may need to configure an advanced authentication handler to support a target Confluence instance.

Note that this extension does not define custom authentication handlers. This configuration is a passthrough option only. For more details on various ways to use authentication handlers, please consult Requests – Authentication. By default, no custom authentication handler is provided to generated REST API requests (if any).

```
from requests_oauthlib import OAuth1

...

confluence_server_auth = OAuth1(client_key,
    client_secret=client_secret,
    resource_owner_key=resource_owner_key,
    resource_owner_secret=resource_owner_secret)
```

confluence_server_cookies

A dictionary value which allows a user to pass key-value cookie information for authentication purposes. This is useful for users who need to authenticate with a single sign-on (SSO) provider to access a target Confluence instance. By default, no cookies are set with a value of `None`.

```
confluence_server_cookies = {
    'SESSION_ID': '<session id string>',
    'U_ID': '<username>'
}
```

1.3.5 advanced processing configuration

confluence_file_suffix

The file name suffix to use for all generated files. By default, all generated files will use the extension `.conf` (see *confluence_file_transform*).

```
confluence_file_suffix = '.conf'
```

---

confluence_file_transform

A function to override the translation of a document name to a filename. The provided function is used to perform translations for both Sphinx's get_outdated_docs and write_doc methods. The default translation will be the combination of "docname + *confluence_file_suffix*".

---

confluence_jira_servers

New in version 1.2.

Provides a dictionary of named JIRA servers to reference when using the `jira` or `jira_issue` directives. In a typical Confluence environment which is linked with a JIRA instance, users do not need to take advantage of this configuration – Confluence should automatically be able to link to respectively JIRA issues or map JIRA query languages with a configured JIRA instance. In select cases where an instance has more than one JIRA instance attached, a user may need to explicitly reference a JIRA instance to properly render a JIRA macro. JIRA-related directives have the ability to reference JIRA instances, with a combination of a UUID and name; for example:

```
.. jira_issue:: TEST-151
    :server-id: d005bcc2-ca4e-4065-8ce8-49ff5ac5857d
    :server-name: MyAwesomeJiraServer
```

It may be tedious for some projects to add this information in each document. As an alternative, a configuration can define JIRA instance information inside a configuration option as follows:

```
confluence_jira_servers = {
    'server-1': {
        'id': '<UUID of JIRA Instance>',
        'name': '<Name of JIRA Instance>'
    }
}
```

With the above option defined in a project's configuration, the following can be used instance inside a document:

```
.. jira_issue:: TEST-151
    :server: server-1
```

---

confluence_lang_transform

A function to override the translation of literal block-based directive language values to Confluence-support code block macro language values. The default translation accepts Pygments documented language types to Confluence-supported syntax highlight languages.

```python
def my_language_translation(lang):
    return 'default'

confluence_lang_transform = my_language_translation
```

confluence_link_suffix

The suffix name to use for generated links to files. By default, all generated links will use the value defined by *confluence_file_suffix* (see *confluence_link_transform*).

```
confluence_link_suffix = '.conf'
```

confluence_link_transform

A function to override the translation of a document name to a (partial) URI. The provided function is used to perform translations for both Sphinx's get_relative_uri method. The default translation will be the combination of "docname + *confluence_link_suffix*".

confluence_remove_title

A boolean value to whether or not automatically remove the title section from all published pages. In Confluence, page names are already presented at the top. With this option enabled, this reduces having two leading headers with the document's title. In some cases, a user may wish to not remove titles when custom prefixes or other custom modifications are in play. By default, this option is enabled with a value of True.

```
confluence_remove_title = True
```

1.4 markup

The following outlines the reStructuredText/Sphinx markup supported by this extension. The intent of this extension is to support as much markup as possible that can be rendered on a Confluence instance. Below will identify markup that has been tested, planned to be implemented in the future or is not compatible with Confluence.

1.4.1 standard

type	status	notes
admonitions	supported	• reStructuredText Admonitions
bibliographic fields	supported	• reStructuredText Bibliographic Fields
block quotes	supported	• reStructuredText Block Quotes
bullet lists	supported	• reStructuredText Bullet Lists
centered	unsupported	• Sphinx Centered
citations	supported	• reStructuredText Citations
code	supported	• Sphinx Code Markup • code-block options `emphasize-lines` and `lines` as well as highlight option `linenothreshold` are ignored due to incompatibilities with Confluence's code block macro. • Highlighting is limited to languages defined by the Confluence instance (see code block macro). • Pending work to validate/improve code block options `caption`, `encoding` and `pyobject`.
compound paragraph	supported	• reStructuredText Compound Paragraph
container	prospect	• reStructuredText Container
definition lists	supported	• reStructuredText Definition Lists

Table 1 – continued from previous page

type	status	notes
deprecated	supported	• Sphinx Deprecated
download	supported	• Sphinx Download
enumerated lists	supported	• reStructuredText Enumerated Lists • Only auto-enumerator lists (#) are supported.
epigraph	supported	• reStructuredText Epigraph
footnotes	supported	• reStructuredText Footnotes
glossary	supported	• Sphinx Glossary
highlights	supported	• reStructuredText Highlights
hlist	unsupported	• Sphinx Horizontal List
hyperlink targets	supported	• reStructuredText Hyperlink Targets
images	supported	• reStructuredText Images
list table	supported	• reStructuredText List Table • Argument `title` not yet supported. • Options not supported: `align`, `header-rows`, `stub-columns` and `widths`.
literal blocks	supported	• reStructuredText Literal Blocks
manpage	supported	• Sphinx Manpage

Continued on next page

Table 1 – continued from previous page

type	status	notes
math	supported	- reStructuredText Math - Requires a LaTeX and dvipng/dvisvgm installation.
parsed literal block	supported	reStructuredText Parsed Literal Block
option lists	supported	reStructuredText Option Lists
production list	supported	Sphinx Production List
pull-quote	supported	reStructuredText Pull-Quote
raw	supported	reStructuredText Raw Data Pass-Through
rubric	supported	Sphinx Rubric
sections	supported	reStructuredText Sections
tables	supported	reStructuredText Tables
toctree	supported	- Sphinx TOC Tree Markup - Argument `caption` not yet supported.
transitions	supported	reStructuredText Transitions
versionadded	supported	Sphinx Version Added
versionchanged	supported	Sphinx Version Changed

(note: directive options "class" and "name" are ignored)

1.4.2 extensions

The following extensions are supported:

- sphinx.ext.autodoc
- sphinx.ext.autosummary
- sphinx.ext.todo

1.4.3 other

If a markup type and/or extension is not listed in the above, is not working as expected or brings up another concern, feel free to bring up an issue:

> Atlassian Confluence Builder for Confluence - Issues
> https://github.com/sphinx-contrib/confluencebuilder/issues

1.5 directives

The following outlines additional directives supported by this extension.

- *jira*
- *jira_issue*

1.5.1 jira

The `jira` directive allows a user to build a JIRA macro to be configured with a provided JQL query. For example:

```
.. jira:: project = "TEST"
   :maximum-issues: 10
```

This directive supports the following options:

- `columns` *(optional)* – A comma-separated list of columns to use when displaying the macro to show in the JIRA table. For example: `key`, `summary`, `updated`, `status`, `resolution`
- `count` *(optional)* – Whether the macro should display a table or just the number of issues. Valid values are `true` or `false`.
- `maximum_issues` *(optional)* – The maximum number of issues a `jira` directive will display. By default, Confluence defaults to `20`.
- `server` *(optional)* – Indicates a named JIRA server provided via `confluence_jira_servers` (*jump*). When set, options `server-id` and `server-name` cannot be set.
- `server-id` *(optional)* – The UUID of the JIRA server to link with. When set, the option `server-name` needs to be set and the option `server` cannot be set.
- `server-name` *(optional)* – The name of the JIRA server to link with. When set, the option `server-id` needs to be set and the option `server` cannot be set.

1.5.2 jira_issue

The `jira_issue` directive allows a user to build a JIRA macro to be configured with a provided JIRA key. For example:

```
.. jira_issue:: TEST-123
```

This directive supports the following options:

- server *(optional)* – Indicates a named JIRA server provided via confluence_jira_servers (*jump*). When set, options server-id and server-name cannot be set.

- server-id *(optional)* – The UUID of the JIRA server to link with. When set, the option server-name needs to be set and the option server cannot be set.

- server-name *(optional)* – The name of the JIRA server to link with. When set, the option server-id needs to be set and the option server cannot be set.

1.6 tips

1.6.1 confluence spaces and unique page names

An important consideration when using this extension is that Confluence has a requirement to having unique page names for a given space. When this extension parses a document's title value, the title is used as either a creation point or an update point (i.e. if the page name does not exist, it will be created; if the page name does exist, it will be updated).

One must be cautious when mixing a space with manually prepared content and published content from this extension. Consider the following use case.

A space MyAwesomeSpace already exists with the following content:

- MyHome
- About
- Tutorials
- See Also

A user may desire to publish a series of Sphinx documentation into a "container" by, so the page "Documentation" is made:

- MyHome
- About
- **Documentation**
- Tutorials
- See Also

If the Sphinx documentation contains a page named "About", unexpected events may occur to new users after publishing for the first time. One might expect the following to be published:

- MyHome

- About

- Documentation

    - About (new)

    - Installing (new)

    - User Guide (new)

    - Other (new)

- Tutorials

- See Also

However, since Confluence only supports a single "About" page for a space, the original "About" page is updated with new content from the documentation set and is moved as a child of the container page:

- MyHome

- Documentation

    - About (**updated and moved**)

    - Installing (new)

    - User Guide (new)

    - Other (new)

- Tutorials

- See Also

Users needing to restrict the extension from possibly mangling manually prepared content can use the `confluence_publish_prefix` (*jump*) or `confluence_publish_postfix` (*jump*) options.

1.6.2 setting a publishing timeout

By default, this extension does not define any timeouts for a publish event. It is recommended to provide a timeout value based on the environment being used (see `confluence_timeout`; *jump*).

1.6.3 disable xml-rpc

When publishing, the default configuration will attempt to first publish using REST API. If publishing fails, the extension automatically attempts to publish using the XML-RPC API. This is solely for compatibility reasons (i.e. environments which for some reason cannot support the REST API). If in an environment where access to a Confluence instance is limited by the network, one may not desire two failed attempts when publishing a documentation set. To disable attempts to publish using the XML-RPC API, see `confluence_disable_xmlrpc` (*jump*).

1.6.4 manage publishing a document subset

Users have the ability to publish a subset of processed documents by using the `confluence_publish_subset` (*jump*) option. This can be useful for large documentation sets where a user may wish to only publish an update for one or more documents (instead of the entire set). While subset publishing can be commonly used by setting the

`confluence_publish_subset` option in a command line build, this may not be ideal for some environments. The following is a code snippet, which when included in a project's `conf.py` file, will provide a means for a user to specify a file outlining which documents are desired:

```python
subset_path = os.getenv('PUBLISH_SUBSET')
if subset_path and os.path.isfile(subset_path):
    with open(subset_path) as f:
        confluence_publish_subset = [line
                                     for raw_line in f
                                     for line in [raw_line.strip()]
                                     if line and not line.startswith('#')]
```

Individual documents can be added into the file defined by the environment variable `PUBLISH_SUBSET` per line. In this snippet, blank lines and lines commented out with the # character are ignored.

1.6.5 asking for help

Having trouble or concerns using this extension? Do not hesitate to bring up an issue:

> Atlassian Confluence Builder for Confluence - Issues
>
> https://github.com/sphinx-contrib/confluencebuilder/issues

1.7 documentation

Documentation for Atlassian Confluence Builder is made with Sphinx and distributed on Read the Docs. Sources of the documentation can be found inside this extension's repository `doc` folder:

> Atlassian Confluence Builder for Confluence
>
> https://github.com/sphinx-contrib/confluencebuilder

The documentation theme used by default is set to `sphinx_rtd_theme`. If locally generating documentation, the theme can be installed on systems using the following command:

```
pip install sphinx_rtd_theme
```

1.8 changelog

1.8.1 1.2.0 (2020-01-03)

- (note) sphinx v1.[6-7] support for this extension is deprecated

- (note) xml-rpc support for this extension is deprecated

- fixed issue when using hierarchy on sphinx 2.1+ (new citations domain)

- fixed issue with document names with path separators for windows users

- fixed issue with multi-line description signatures (e.g. c++ autodocs)

- fixed issue with processing hidden toctrees

- fixed issue with unicode paths with confluence_publish_subset and python 2.7

- improved formatting for option list arguments

- improved handling and feedback when configured with incorrect publish instance
- improved name management for published assets
- improved reference linking for sphinx domains capability (meth, attr, etc.)
- introduce a series of jira directives
- support 'firstline' parameter in the code block macro
- support base admonition directive
- support confluence 7 series newline management
- support default alignment in sphinx 2.1+
- support document postfixes
- support for generated image assets (asterisk marked)
- support passthrough authentication handlers for rest calls
- support previous/next navigation
- support prompting for publish username
- support sphinx.ext.autosummary extension
- support sphinx.ext.todo extension
- support the math directive
- support toctree's numbered option
- support users injecting cookie data (for authentication) into rest calls

1.8.2 1.1.0 (2019-03-16)

- repackaged release (see sphinx-contrib/confluencebuilder#192)

1.8.3 1.0.0 (2019-03-14)

- all confluence-based macros can be restricted by the user
- block quotes with attribution are styled with confluence quotes
- citations/footnotes now have back references
- enumerated lists now support various styling types
- fixed issue with enumerated lists breaking build on older sphinx versions
- fixed issue with relative-provided header/footer assets
- fixed issues where table-of-contents may generate broken links
- improve support with interaction with other extensions
- improved paragraph indentation
- initial autodoc support
- nested tables and spanning cells are now supported
- provide option for a caller to request a password for publishing documents
- storage format support (two-pass publishing no longer needed)

- support for sass/yaml language types

- support parsed literal content

- support publishing subset of documents

- support the download directive

- support the image/figure directives

- support the manpage role

1.8.4 0.9.0 (2018-06-02)

- fixed a series of content escaping issues

- fixed an issue when purging content would remove just-published pages

- fixed detailed configuration errors from being hidden

- improve proxy support for xml-rpc on various python versions

- improve support for various confluence url configurations

- improve support in handling literal block languages

- support automatic title generation for documents (if missing)

- support linenothreshold option for hightlight directive

- support maximum page depth (nesting documents)

- support the raw directive

- support two-way ssl connections

1.8.5 0.8.0 (2017-12-05)

- fix case where first-publish with 'confluence_master_homepage' fails to configure the space's homepage

- support page hierarchy

- improve pypi cover notes

1.8.6 0.7.0 (2017-11-30)

- cap headers/sections to six levels for improved visualization

- fixed rest publishing for encoding issues and python 3.x (< 3.6) issues

- improve markup for:

  - body element lists

  - citations

  - definitions

  - footnotes

  - inline literals

  - literal block (code)

- rubric

- seealso

- table

- versionmodified

- re-work generated document references/targets (reference to section names)

- sanitize output to prevent confluence errors for certain characters

- support indentations markup

- support master_doc option to configure space's homepage

- support removing document titles from page outputs

- support silent page updates

1.8.7 0.6.0 (2017-04-23)

- cleanup module's structure, versions and other minor files

- drop 'confluence' pypi package (embedded xml-rpc support added)

- improve hyperlink and cross-referencing arbitrary locations/documents support

- improve proxy support

- re-support python 3.x series

- support anonymous publishing

- support rest api

1.8.8 0.5.0 (2017-03-31)

- header/footer support

- purging support

- use macros for admonitions

- (note) known issues with python 3.3, 3.4, 3.5 or 3.6 (see sphinx-contrib/confluencebuilder#10)

1.8.9 0.4.0 (2017-02-21)

- move from 'confluence' pypi package to 'confluence' pypi package (required for publishing). see: https://github.com/pycontribs/confluence

1.8.10 0.3.0 (2017-01-22)

- adding travis ci, tox and initial unit testing

- module now depends on `future`

- providing initial support for python 3

1.8.11 0.2.0 (2016-07-13)

- moved configuration to the sphinx config

1.8.12 0.1.1 (2016-07-12)

- added table support
- fixed internal links

1.8.13 0.1.0 (2016-07-12)

- added lists, bullets, formatted text
- added headings and titles