



# **Atlassian Confluence Builder**

**Sphinx Documentation Generation / Publisher**

Provided by Sphinx Confluence Builder Contributors

2021-05-26

1.5.0

# CONTENTS

<b>1</b>	<b>Installing</b>	<b>1</b>
<b>2</b>	<b>Installing (virtualenv)</b>	<b>4</b>
<b>3</b>	<b>Tutorial</b>	<b>7</b>
<b>4</b>	<b>Configuration</b>	<b>10</b>
<b>5</b>	<b>Builders</b>	<b>25</b>
<b>6</b>	<b>Directives</b>	<b>26</b>
<b>7</b>	<b>Compatibilities</b>	<b>29</b>
<b>8</b>	<b>Tips</b>	<b>34</b>
<b>9</b>	<b>Changelog</b>	<b>37</b>
<b>10</b>	<b>Indices and tables</b>	<b>44</b>
	<b>Index</b>	<b>45</b>

## INSTALLING

Atlassian Confluence Builder for Sphinx 1.5.0 depends on:

- [Python](#) 2.7 or 3.6+
- [Requests](#) 2.14.0 or later
- [Sphinx](#) 1.8 or 2.4+
- [Confluence](#) Cloud or Server 6.13+

The recommended method of installation is using [pip](#).

```
pip install sphinxcontrib-confluencebuilder
(or)
python -m pip install sphinxcontrib-confluencebuilder
```

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

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

For new users, following provides a series of steps to assist in preparing a new environment to use this package. For users wishing to use virtualenv, please consult the instructions in [install-virtualenv](#).

---

**Note:** If the installation process fails with the following error “AttributeError: ‘\_NamespacePath’ object has no attribute ‘sort’”, try upgrading the [setuptools](#) module:

```
pip install --upgrade setuptools
(or)
python -m pip install --upgrade setuptools
```

---

## 1.1 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:

### 1.1.1 Arch

```
$ sudo pacman -Sy
$ sudo pacman -S python-pip
$ pip install sphinxcontrib-confluencebuilder
$ python -m sphinxcontrib.confluencebuilder --version
sphinxcontrib.confluencebuilder <version>
```

### 1.1.2 CentOS

```
$ sudo yum install epel-release
$ sudo yum install python-pip
$ pip install sphinxcontrib-confluencebuilder
$ python -m sphinxcontrib.confluencebuilder --version
sphinxcontrib.confluencebuilder <version>
```

### 1.1.3 Fedora

```
$ sudo dnf install python-pip
$ pip install sphinxcontrib-confluencebuilder
$ python -m sphinxcontrib.confluencebuilder --version
sphinxcontrib.confluencebuilder <version>
```

### 1.1.4 Ubuntu

```
$ sudo apt-get update
$ sudo apt-get install python-pip
$ pip install sphinxcontrib-confluencebuilder
$ python -m sphinxcontrib.confluencebuilder --version
sphinxcontrib.confluencebuilder <version>
```

## 1.2 OS X

From a terminal, invoke the following commands:

```
$ sudo easy_install pip
$ pip install sphinxcontrib-confluencebuilder
$ python -m sphinxcontrib.confluencebuilder --version
sphinxcontrib.confluencebuilder <version>
```

## 1.3 Windows

If not already installed, download the most recent version of [Python](https://www.python.org/downloads/):

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 invoke Python from an absolute path. The remainder of these steps will assume Python is available in the path.

Open a Windows command prompt. Invoke the following:

```
> python -m pip install sphinxcontrib-confluencebuilder
> python -m sphinxcontrib.confluencebuilder --version
sphinxcontrib.confluencebuilder <version>
```

## 1.4 Development installation

To install the bleeding edge sources, the following [pip](#) command can be used:

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

**orphan**

## INSTALLING (VIRTUALENV)

For users wishing to use `virtualenv` for their environments, the following outlines the recommended commands to invoke for various environments. For users who do not desire to use `virtualenv`, please see [Installing](#) instead.

### 2.1 Linux (virtualenv)

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:

#### 2.1.1 Arch (virtualenv)

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

#### 2.1.2 CentOS (virtualenv)

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

### 2.1.3 Fedora (virtualenv)

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

### 2.1.4 Ubuntu (virtualenv)

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

## 2.2 OS X (virtualenv)

From a terminal, invoke the following commands:

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

## 2.3 Windows (virtualenv)

If not already installed, download the most recent version of [Python](https://www.python.org/downloads/):

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 invoke Python from an absolute path. The remainder of these steps will assume Python is available in the path.

Open a Windows command prompt as an administrator. Invoke the following to install `virtualenv`:

```
> 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:

```
> virtualenv sphinx-venv
> source sphinx-venv\Scripts\activate.bat
> python -m pip install sphinxcontrib-confluencebuilder
> python -m sphinxcontrib.confluencebuilder --version
sphinxcontrib.confluencebuilder <version>
```



## TUTORIAL

---

**Note:** Advanced users of Sphinx can skip this section and view *configuration options* available to prepare their documentation.

---

After *installing* Atlassian Confluence Builder for Sphinx, a Sphinx project can be configured to use supported *builders*. The following tutorial will provide a series of steps which will:

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

### 3.1 New documentation

If a user is starting a new Sphinx-based documentation, the following steps can be used to create a new minimalistic Sphinx configuration or use Sphinx's quick-start utility. If attempting to use this extension for existing documentation, start *configuring for this extension*.

#### 3.1.1 Quick-start

If opting for the quick-start utility, open a terminal to the location where documentation should be generated (typically, an empty directory) 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 the documentation by *configuring for this extension*.

### 3.1.2 Minimalistic

For a minimalistic setup, create a new folder for the new documentation and configuration to be used. This is done by first creating a document named `index.rst` with the following content:

```
My documentation
```

```
=====
```

```
This is a test document.
```

Next, create a configuration file `conf.py` with the following information:

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

```
extensions = []
```

After preparing these files, continue by *configuring for this extension* as follows.

## 3.2 Configuring to use this extension

Enable this extension by registering the extension in 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'
confluence_ask_password = True
# (for Confluence Cloud)
confluence_server_url = 'https://example.atlassian.net/wiki/'
confluence_server_user = 'myawesomeuser@example.com'
# (or, for Confluence Server)
confluence_server_url = 'https://intranet-wiki.example.com/'
confluence_server_user = 'myawesomeuser'
```

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

**Note:** The configuration of the space name (`confluence_space_name`) is case-sensitive. Ensure the value matches the case found on the Confluence instances (typically, uppercase).

### 3.3 Recommended configurations

By default, this extension will publish any documents to the root of a configured space. It can be common for most users to want to publish a documentation set as children of an already existing page. To take advantage of this feature, a user will want to define a `confluence_parent_page` option in their configuration file. For example:

```
confluence_parent_page = 'MyDocumentation'
```

When publishing a documentation set, the above configuration will tell this extension to publish all documents under the `MyDocumentation` page.

By default, all documents published to a Confluence instance will be stored either in the root of the space or a configured parent space (as mentioned above). For larger documentation sets which include multiple nested documents, it may be desired to have individual documents published as children of other published documents. Configuring the `confluence_page_hierarchy` option will allow a user to enable hierarchy support. For example:

```
confluence_page_hierarchy = True
```

For first time users, they may wish to sanity check what content will be published before publishing for the first time to a Confluence instance. A user can perform a dryrun by configuring the `confluence_publish_dryrun` option in the project's configuration file. For example:

```
confluence_publish_dryrun = True
```

For more information on the above or additional configuration options, see [all configuration options](#).

### 3.4 Building/publishing documentation

To process and publish the documentation set, invoke Sphinx with the `confluence` builder (or a desired *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.

For users who set the dryrun option above (`confluence_publish_dryrun`), they may inspect the output of the run to confirm what the publish event will perform. If the desired result is observed, a user can remove the dryrun option and re-invoke the build/publish command to publish onto the configured Confluence instance.

## CONFIGURATION

The following is an example of simple configuration (`config.py`) 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 = 'myawesomeuser'
confluence_ask_password = True
confluence_page_hierarchy = True
```

All configurations introduced by this extension are prefixed with `confluence_`. This extension may take advantage of a subset of [Sphinx configurations](#) as well when preparing documents. View the entire list of configuration options below.

- *Essential configuration*
- *Generic configuration*
- *Publishing configuration*
- *Advanced publishing configuration*
- *Advanced processing configuration*
- *Deprecated options*

### 4.1 Essential configuration

#### **confluence\_publish**

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

```
confluence_publish = True
```

#### **confluence\_server\_url**

The URL for the Confluence instance to publish to. 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 (i.e. excluding `rest/api/`). 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/`, should be as follows:

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

### confluence\_space\_name

---

**Note:** The space name is **case-sensitive**.

---

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

```
confluence_space_name = 'MyAwesomeSpace'
```

### 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 Server, this value will most likely be the username value.

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

### confluence\_server\_pass

**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 be modified various ways with Python to pull an authentication token for a publishing event such as *reading from an environment variable*, reading from a local file or acquiring a password from `getpass`. If desired, this extension provides a method for prompting for a password (see *confluence\_ask\_password*).

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 = 'vsU5rSZ6Z4kmrQMapSXBYkJh'
```

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

```
confluence_server_pass = 'myawesomepassword'
```

## 4.2 Generic configuration

### **confluence\_add\_secnumbers**

New in version 1.2.

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

```
confluence_add_secnumbers = True
```

See also [confluence\\_publish\\_prefix](#).

### **confluence\_default\_alignment**

New in version 1.3.

Explicitly set which alignment type to use when a default alignment value is detected. As of Sphinx 2.0+, the default alignment is set to `center`. Legacy versions of Sphinx had a default alignment of `left`. By default, this extension will use a Sphinx-defined default alignment unless explicitly set by this configuration value. Accepted values are `left`, `center` or `right`.

```
confluence_default_alignment = 'left'
```

### **confluence\_header\_file**

The name of the file to use header data. If provided, the raw 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 raw 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**

A positive 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 default, the maximum document depth is disabled with a value of `None`.

```
confluence_max_doc_depth = 2
```

---

**Important:** This feature still supports a document depth of `0`, where all child pages of the root document will be merged into a single document. Setting this value to `0` is considered deprecated. An alternative approach to generating a single document page is to use the `singleconfluence` [builder](#) instead.

---

**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, hierarchy mode is disabled with a value of `False`.

```
confluence_page_hierarchy = True
```

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 and uploaded to either the configured `confluence_parent_page` or in the root of the space.

**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`.

## 4.3 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 a 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 an alternative means), select environments may wish to provide a way to accept 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 a shell environment. While a user is typically defined in the option `confluence_server_user`, select environments may wish to provide a way to accept a username 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
```

See also:

- [`confluence\_remove\_title`](#)
- [`confluence\_title\_overrides`](#)

**confluence\_disable\_notifications**

A boolean value which explicitly disables 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
```

Note that even if this option is set, there may be some scenarios where a notification will be generated for other users when a page is created or removed, depending on how other users may be watching a space.

See also [`confluence\_watch`](#).

**confluence\_global\_labels**

New in version 1.3.

Defines a list of labels to apply to each document being published. When a publish event either adds a new page or updates an existing page, the labels defined in this option will be added/set on the page. For example:

```
confluence_global_labels = ['label-a', 'label-b']
```

For per-document labels, please consult the [`confluence\_metadata` directive](#). See also [`confluence\_append\_labels`](#).

**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**


---

**Note:** This option cannot be used with [`confluence\_publish\_root`](#).

---



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, new 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.

See also *confluence\_publish\_root*.

### **confluence\_publish\_postfix**

New in version 1.2.

If set, a 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\_ignore\_titlefix\_on\_index*
- *confluence\_publish\_prefix*

### **confluence\_publish\_prefix**

If set, a 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\_ignore\_titlefix\_on\_index*
- *confluence\_publish\_postfix*

### **confluence\_publish\_root**

---

**Note:** This option cannot be used with *confluence\_parent\_page*.

---

The page identifier to publish the root document to. The root identifier value is used to find an existing page on the configured Confluence instance. When found, the root document of the documentation set being published

will replace the content of the page found on the Confluence instance. If the root page cannot be found, the publish attempt will stop with an error message.

```
confluence_publish_root = 123456
```

See also [confluence\\_parent\\_page](#).

### 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 flagged 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.

See also:

- [confluence\\_publish\\_dryrun](#)
- [confluence\\_purge\\_from\\_master](#)

### 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 requires [confluence\\_purge](#) to be set to `True` before taking effect. If [confluence\\_publish\\_root](#) is set, this option is implicitly enabled.

```
confluence_purge_from_master = False
```

See also [confluence\\_purge](#).

### confluence\_title\_overrides

New in version 1.3.

Allows a user to override the title value for a specific document. When documents are parsed for title values, the first title element's content will be used as the publish page's title. Select documents may not include a title and are ignored; or, documents may conflict with each other but there is a desire to keep them the same name in reStructuredText form. With `confluence_title_overrides`, a user can define a dictionary which will map a given docname to a title value instead of the title element (if any) found in the respective document. By default, documents will give assigned titles values based off the first detected title element with a value of `None`.

```
confluence_title_overrides = {
    'index': 'Index Override',
}
```

See also:

- *Confluence Spaces and Unique Page Names*
- `confluence_disable_autogen_title`
- `confluence_publish_postfix`
- `confluence_publish_prefix`
- `confluence_remove_title`

#### **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`). 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
```

#### **confluence\_watch**

New in version 1.3.

Indicate whether or not the user publishing content will automatically watch pages for changes. In Confluence, when creating a new page or updating an existing page, the editing user will automatically watch the page. Notifications on automatically published content is typically not relevant to publishers through this extension, especially if the content is volatile. If a publisher wishes to be keep informed on notification for published pages, this option can be set to `True`. By default, watching is disabled with a value of `False`.

```
confluence_watch = False
```

See also `confluence_disable_notifications`.

## 4.4 Advanced publishing configuration

#### **confluence\_append\_labels**

New in version 1.3.

Allows a user to decide how to manage labels for an updated page. When a page update contains new labels to set, they can either be stacked on existing labels or replaced. In the event that a publisher wishes to replace any existing labels that are set on published pages, this option can be set to `False`. By default, labels are always appended with a value of `True`.

```
confluence_append_labels = True
```

See also:

- `confluence_global_labels`
- `confluence_metadata` *directive*

#### **confluence\_asset\_force\_standalone**

New in version 1.3.

Provides an override to always publish individual assets (images, downloads, etc.) on each individual document which uses them. This extension will attempt to minimize the amount of publishing of shared assets on multiple documents by only hosting an asset in a single document. For example, if two documents use the same image, the image will be hosted on the root document of a set and each document will reference the attachment on the root

page. A user may wish to override this feature. By configuring this option to `True`, this extension will publish asset files as an attachment for each document which may use the asset. By default, this extension will attempt to host shared assets on a single document with a value of `False`.

```
confluence_asset_force_standalone = True
```

#### **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, a user 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. Refer to the [Requests SSL Cert Verification](#) documentation (`verify`) for more information. If server verification is explicitly disabled, this option is ignored. By default, this option is ignored with a value of `None`.

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

See also:

- [confluence\\_client\\_cert\\_pass](#)
- [confluence\\_client\\_cert](#)
- [confluence\\_disable\\_ssl\\_validation](#)

#### **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')
```

See also:

- [confluence\\_ca\\_cert](#)
- [confluence\\_client\\_cert\\_pass](#)
- [confluence\\_disable\\_ssl\\_validation](#)

#### **confluence\_client\_cert\_pass**

**Caution:** It is never recommended to store a certificate's passphrase into a committed/shared repository holding documentation.

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_ca_cert`
- `confluence_client_cert`
- `confluence_disable_ssl_validation`

#### **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_ca_cert`
- `confluence_client_cert`
- `confluence_client_cert_pass`

#### **confluence\_ignore\_titlefix\_on\_index**

New in version 1.3.

When configured to add a prefix or postfix onto the titles of published documents, a user may not want to have any title modifications on the index page. To prevent modifying an index page's title, this option can be set to `True`. By default, this option is set to `False`.

```
confluence_ignore_titlefix_on_index = True
```

See also:

- `confluence_publish_postfix`
- `confluence_publish_prefix`

#### **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 = 123456
```

See also `confluence_parent_page`.

**confluence\_proxy**

REST calls use the [Requests](#) library, which will use system-defined proxy configuration; however, a user can override the system-defined proxy by providing a proxy server using this configuration.

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

**confluence\_publish\_allowlist**

New in version 1.3.

---

**Note:** Using this option will disable the [confluence\\_purge](#) option.

---

Defines a list 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 contains undesired documents to publish, [confluence\\_publish\\_allowlist](#) can be used to override this. This option accepts either a list of relative path document names (without an extension) or a filename which contains a list of document names.

For example, a user can specify documents in a list to allow for publishing:

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

Alternatively, a user can specify a filename such as following:

```
confluence_publish_allowlist = 'allowed-docs.txt'
```

Which could contain a list of documents to allow:

```
index
foo/bar
```

A user can configured an allowed list of documents through the command line:

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

By default, this option is ignored with a value of None.

See also [confluence\\_publish\\_denylist](#).

**confluence\_publish\_denylist**

New in version 1.3.

---

**Note:** Using this option will disable the [confluence\\_purge](#) option.

---

Defines a list of documents to not 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\\_denylist](#) can be used to override this. This option accepts either a list of relative path document names (without an extension) or a filename which contains a list of document names.

For example, a user can specify documents in a list to deny for publishing:

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

Alternatively, a user can specify a filename such as following:

```
confluence_publish_denylist = 'denied-docs.txt'
```

Which could contain a list of documents to allow:

```
index
foo/bar
```

A user can configured a denied list of documents through the command line:

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

By default, this option is ignored with a value of `None`.

See also [confluence\\_publish\\_allowlist](#).

### **confluence\_publish\_dryrun**

New in version 1.3.

When a user wishes to start managing a new document set for publishing, there maybe concerns about conflicts with existing content. When the dry run feature is enabled to `True`, a publish event will not edit or remove any existing content. Instead, the extension will inform the user which pages will be created, whether or not pages will be moved and whether or not pages/attachments will be removed. By default, the dry run feature is disabled with a value of `False`.

```
confluence_publish_dryrun = True
```

See also *Confluence Spaces and Unique Page Names*.

### **confluence\_publish\_headers**

A dictionary value which allows a user to pass key-value header information. This is useful for users who need to interact with a Confluence instance which expects (in a reverse proxy or the instance itself) specific header information to be set. By default, no custom header entries are added with a value of `None`.

```
confluence_publish_headers = {
    'CUSTOM_HEADER': '<some-value>',
}
```

### **confluence\_publish\_onlynew**

New in version 1.3.

A publish event will from this extension will typically upload new pages or update existing pages on future attempts. In select cases, a user may not wish to modify existing pages and only permit adding new content to a Confluence space. To achieve this, a user can enable an “only-new” flag which prevents the modification of existing content. This includes the restriction of updating existing pages/attachments as well as deleting content. By default, the only-new feature is disabled with a value of `False`.

```
confluence_publish_onlynew = True
```

**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 provides 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. An example OAuth 1 is as follows:

```
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>'
}
```

## 4.5 Advanced processing configuration

**confluence\_additional\_mime\_types**

New in version 1.3.

Candidate selection for images will only support the internally managed list of MIME types supported by a default Confluence instance. A custom installation or future installations of a Confluence instance may support newer MIME types not explicitly managed by this extension. This configuration provides a user the option to register additional MIME types to consider for image candidates.

```
confluence_additional_mime_types = [
    'image/tiff',
]
```

**confluence\_file\_suffix**

The file name suffix to use for all generated files. By default, all generated files will use the extension `.conf`.

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

See also [confluence\\_file\\_transform](#).

**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 supported code block macro language values. The default translation accepts [Pygments documented language types](#) to Confluence-supported syntax highlight languages.

```
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`.

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

See also `confluence_link_transform`.

### **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`

See also:

- *confluence\_disable\_autogen\_title*
- *confluence\_title\_overrides*

## 4.6 Deprecated options

### **confluence\_publish\_subset**

Changed in version 1.3.

This option has been renamed to *confluence\_publish\_allowlist*.

## BUILDERS

The following outlines the Sphinx builders provided by this extension.

### **confluence**

The **confluence** builder allows a user to process a Sphinx supported documentation set to generate a Confluence supported representation. Individual documents will generate Confluence supported documents, which in turn can be published to a configured Confluence instance:

```
sphinx-build -b confluence . _build/confluence -E -a
```

### **singleconfluence**

New in version 1.3.

The **singleconfluence** builder allows a user to process a Sphinx supported documentation set to generate a single document in a Confluence supported representation. The generated document can in turn be published to a configured Confluence instance:

```
sphinx-build -b singleconfluence . _build/singleconfluence -E -a
```

## DIRECTIVES

The following outlines additional [directives](#) supported by this extension.

Directives listed below are only supported when using this extension. For users with documentation that is built with multiple builders, the following can be used to restrict these directives to supported builders by using the `:only:` directive. For example:

```
.. only:: builder_confluence or builder_singleconfluence

  .. confluence_metadata::
     :labels: label-test
```

### 6.1 Common

#### **.. confluence\_expand::**

New in version 1.3.

The `confluence_expand` directive allows a user to define a Confluence [Expand Macro](#) to help manage the visibility of content on a page. For example:

```
.. confluence_expand::

  This content is captured inside the expand macro.
```

This directive supports the following options:

#### **:title: value (string)**

A string value to apply to the macros “title” field.

```
.. confluence_expand::
   :title: View more details...

  This content is captured inside the expand macro.
```

#### **.. confluence\_metadata::**

New in version 1.3.

The `confluence_metadata` directive allows a user to define metadata information to be added during a publish event. This directive supports the following options:

#### **:labels: value (space separated strings)**

A space-separated list of label strings to apply to a page. The following example will result in the labels `label-a` and `label-b` being added to the document which defines this directive.

```
.. confluence_metadata::
   :labels: label-a label-b
```

See also `confluence_global_labels` ([ref](#)).

## 6.2 JIRA

The following directives can be used to help include JIRA macros into generated Confluence documents.

.. **jira::** [jql]

New in version 1.2.

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

```
.. jira:: project = "TEST"
```

This directive supports the following options:

**:columns:** value (comma separated numbers)

A comma-separated list of columns to use when displaying the macro to show in the JIRA table.

```
.. jira:: project = "TEST"
   :columns: key,summary,updated,status,resolution
```

**:count:** flag (boolean)

Whether the macro should display a table or just the number of issues. Valid values are `true` or `false`.

```
.. jira:: project = "TEST"
   :count: true
```

**:maximum\_issues:** count (number)

The maximum number of issues a `jira` directive will display. By default, Confluence defaults to 20.

```
.. jira:: project = "TEST"
   :maximum_issues: 10
```

**:server:** instance (string)

Indicates a named JIRA server provided via `confluence_jira_servers` ([ref](#)). When set, options `server-id` and `server-name` cannot be set.

```
.. jira:: project = "TEST"
   :server: server-1
```

**:server-id:** uuid (string)

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.

```
.. jira:: project = "TEST"
   :server-id: d005bcc2-ca4e-4065-8ce8-49ff5ac5857d
   :server-name: MyAwesomeJiraServer
```

**:server-name:** name (string)

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.

```
.. jira:: project = "TEST"
   :server-id: d005bcc2-ca4e-4065-8ce8-49ff5ac5857d
   :server-name: MyAwesomeJiraServer
```

.. **jira\_issue::** [issue-id]

New in version 1.2.

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: instance (string)**

Indicates a named JIRA server provided via `confluence_jira_servers` ([ref](#)). When set, options `server-id` and `server-name` cannot be set.

```
.. jira_issue:: TEST-123
   :server: server-1
```

**:server-id: uuid (string)**

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.

```
.. jira_issue:: TEST-123
   :server-id: d005bcc2-ca4e-4065-8ce8-49ff5ac5857d
   :server-name: MyAwesomeJiraServer
```

**:server-name: name (string)**

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.

```
.. jira_issue:: TEST-123
   :server-id: d005bcc2-ca4e-4065-8ce8-49ff5ac5857d
   :server-name: MyAwesomeJiraServer
```

## **COMPATIBILITIES**

The following outlines the `reStructuredText/Sphinx` markup, configuration entries and more supported by this extension. The intent of this extension is to support various standard Sphinx features that can be rendered on a Confluence instance. Below will identify features that have been tested, planned to be implemented in the future or is not compatible with Confluence.

## 7.1 reStructuredText markup

Type	Notes
admonitions	Supported
bibliographic fields	Supported
block quotes	Supported
bullet lists	Supported
citations	Supported
compound paragraph	Supported
container	Unsupported. Confluence markup does not permit the use of the <code>class</code> attribute for tags.
csv-table	Supported
definition lists	Supported
enumerated lists	Limited support. Only auto-enumerator lists (#) are supported. Using other types of enumerated lists will be handled as auto-enumerators. This is a limitation in the Confluence markup.
epigraph	Supported
footnotes	Supported
highlights	Supported
hyperlink targets	Supported
images	Supported
inline markup	Supported
list-table	Supported
literal blocks	Supported
math	Supported with additional system tools. Requires a LaTeX and dvipng/dvisvgm installation.
parsed literal block	Supported
option lists	Supported
pull-quote	Supported
raw	Supported. Use the <code>confluence_storage</code> format type to inject raw data into a document.
rubric	Supported
sections	Supported
tables	Supported
transitions	Supported

(note: directive options “`class`” and “`name`” are ignored as they are not supported in a Confluence format document)



## 7.2 Sphinx markup

Type	Notes
<code>centered</code>	Supported
<code>code</code>	Limited support. Supported languages (for highlighting) are limited to the languages supported by Confluence's <code>code block macro</code> . This applies to a language defined in a <code>code-block</code> directive or set through a <code>highlight</code> directive. The <code>code-block</code> directive contain the options <code>emphasize-lines</code> and <code>lines</code> which are not supported in the Confluence markup. The code block macro only supports a simple line numbers (configurable with the <code>linenos</code> option).
<code>deprecated</code>	Supported
<code>download</code>	Supported
<code>glossary</code>	Supported
<code>hlist</code>	Supported
<code>manpage</code>	Supported
<code>production list</code>	Supported
<code>toctree</code>	Supported
<code>versionadded</code>	Supported
<code>versionchanged</code>	Supported

*(note: directive options “class” and “name” are ignored as they are not supported in a Confluence format document)*

## 7.3 Miscellaneous

This extension currently does not support the generation of indexed content (e.g. `genindex` or `modindex`).

This extension does not include support for generating a “Search” document as Confluence provides its own advanced search capabilities.

## 7.4 Extensions

**Note:** Atlassian Confluence Builder for Sphinx does not plan to directly support external extensions provided outside of Sphinx's main source repository. However, changes are considered towards this extension's processing and API to make it flexible for other extensions to integrate.

Developers wishing to integrate an extension with Atlassian Confluence Builder can either attempt to add implementation inside the extension itself (if permitted and rational), or create a new extension which can bind both desired extensions together (e.g. `sphinx-confluence-nbsphinx-test`).

This extension will attempt to support any extension that is integrated into Sphinx's repository. The following extensions are currently supported:

Type	Notes
<code>sphinx.ext.autodoc</code>	Supported*. While support for autodocs has been included in this extension, only a limited amount of examples and testing has been done to verify its capabilities. If an issue is observed when using an autodoc feature, please confirm an expected result using an <code>html</code> build then report the issue.
<code>sphinx.ext.autosectionlabel</code>	Supported
<code>sphinx.ext.autosummary</code>	Supported
<code>sphinx.ext.coverage</code>	N/A
<code>sphinx.ext.doctest</code>	Supported
<code>sphinx.ext.duration</code>	Supported
<code>sphinx.ext.extlinks</code>	Supported
<code>sphinx.ext.githubpages</code>	N/A
<code>sphinx.ext.graphviz</code>	Supported
<code>sphinx.ext.ifconfig</code>	Supported
<code>sphinx.ext.imgconverter</code>	N/A
<code>sphinx.ext.imgmath</code>	Supported
<code>sphinx.ext.inheritance_diagram</code>	Supported
<code>sphinx.ext.intersphinx</code>	Limited support. Users should have no issues when using external modules via <code>intersphinx_mapping</code> . When building with publishing enabled, this extension will generate an <code>objects.inv</code> inventory file which can be used by other documentation sets. At this time, only document names and standard references are supported. A generated inventory file is not published to a Confluence instance.
<code>sphinx.ext.jsmath</code>	Unsupported. Confluence does not support the injection of JavaScript into a page in most scenarios.
<code>sphinx.ext.linkcode</code>	Unsupported. This extension only supports injecting references for the <code>html</code> builder.
<code>sphinx.ext.mathjax</code>	Unsupported. Confluence does not support the injection of JavaScript into a page in most scenarios.
<code>sphinx.ext.napoleon</code>	Supported
<code>sphinx.ext.todo</code>	Supported
<code>sphinx.ext.viewcode</code>	Unsupported. This extension only supports injecting references for the <code>html</code> builder.

## 7.5 Other

If a feature and/or extension is not listed 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>

## 8.1 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).

A user 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” page, so the page “Documentation” is made:

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

If the Sphinx documentation contains a page named “About”, an unexpected event may occur for new users after publishing for the first time. A user 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` ([ref](#)) or `confluence_publish_postfix` ([ref](#)) options.

See also the [dry run capability](#) and the [title overrides capability](#).

## 8.2 Recommended options for math

The following are recommended options to use when using `sphinx.ext.imgmath`:

```
imgmath_font_size = 14
imgmath_use_preview = True
imgmath_image_format = 'svg'
```

## 8.3 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`; [ref](#)).

## 8.4 Publishing with a CI secret key

---

**Note:** If running in a `tox/virtualenv` setup, ensure any environment variables used are configured to be passed through to the virtual environment.

---

For users performing automatic publishing through a CI system, they may wish to authenticate their publish event with a secret key. A common approach to applying a secret key is through an environment variable. For example:

```
import os

...

confluence_server_pass = os.getenv('SECRET_KEY')
```

The above will read an environment variable `SECRET_KEY` prepared by a CI script which will be set on the `confluence_server_pass` ([ref](#)) configuration.

## 8.5 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>

For issues when using this extension, generating a report and including this content in an issue may be helpful towards finding a solution. To generate a report, run the following command from the documentation directory:

```
$ python -m sphinxcontrib.confluencebuilder report
...
Confluence builder report has been generated.
Please copy the following text for the GitHub issue:

-----[ cut here ]-----
(system)
...
(configuration)
...
(confluence instance)
...
-----[ cut here ]-----
```

## CHANGELOG

### 9.1 1.5.0 (2021-05-25)

- fixed issue where this extension conflicts with docutils's translator attribs
- fixed issue with :doc:<> references when using singlebuilder
- fixed issue with alignment changes in newer sphinx
- fixed issue with caption/title changes in newer docutils/sphinx
- fixed issue with singlebuilder processing assets
- fixed issue with table width hints using pixels instead of percentages
- improvements for getpass handling in msystem-environments
- support an explicit root page publishing option
- support custom headers for rest calls

### 9.2 1.4.0 (2021-01-17)

- fixed issue where a meta node directive would fail the writer stage
- fixed issue where intersphinx would fail in python 2.7
- fixed issue where not all math directive content would be accepted
- fixed/improved handling of configuration options from command line
- support for math visual depth adjustments (line alignment)
- support for numerical figures and references to these figures
- support late image/download processing (for third-party extensions)

## 9.3 1.3.0 (2020-12-31)

- **(note)** support for sphinx v1.[6-7] has been dropped
- **(note)** support for xml-rpc has been dropped
- conflicting titles will be automatically adjusted to prevent publishing issues
- enable page-specific title overrides via `confluence_title_overrides`
- ensure configured title postfix is not trimmed in long titles
- extend language mappings for supported storage format language types
- fixed a series of scenarios where titles/missing images will fail a build
- fixed indentation to consistent offset for newer confluence instances
- fixed issue when building heading which reference another document
- fixed issue when processing a download role with a url
- fixed issue where an anchor target may not generate a proper link
- fixed issue where ask options would fail in python 2.7
- fixed issue where ask options would prompt when not publishing
- fixed issue where autosummary registration may fail
- fixed issue where default alignment did not apply to a figure's legend
- fixed issue where empty pages could not be published
- fixed issue where links to headers which contain a link would fail
- fixed issue where literal-marked includes would fail to publish
- fixed issue where registering this extension caused issues with other builders
- fixed issue where todo entries would render when disabled in configuration
- fixed issue with previous-next links not generated for nested pages
- improved built references by including title (alt) data if set
- improved code macros rendering a title value when a caption is set
- improved emphasis handling for autodocs content
- improved figure/section numbering
- improved handling unknown code languages to none-styled (instead of python)
- improved previous-next button visualization
- improved publishing when dealing with changing page title casing
- introduce the `expand` directive
- introduce the `report` command line feature
- introduce the `wipe` command line feature
- promote `confluence_storage` over `confluence` for raw type
- support `:stub-columns:` option in a list-table directive
- support disabling titlefix on an index page



- support for assigning confluence labels for pages
- support for both allow and deny lists for published documents
- support for centered directive
- support for graphviz extension
- support for hlist directive
- support for inheritance-diagram extension
- support image candidate detection of extra image types for custom instances
- support publish dry runs
- support single-page builder
- support the `:backlinks:` option for contents directive
- support the generation of an inventory file (for intersphinx)
- support users overriding default alignment
- support users to force standalone hosting of shared assets
- support width hints for tables

## 9.4 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

## 9.5 1.1.0 (2019-03-16)

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

## 9.6 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

## 9.7 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 highlight directive
- support maximum page depth (nesting documents)
- support the raw directive
- support two-way ssl connections

## 9.8 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

## 9.9 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

## 9.10 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

## 9.11 0.5.0 (2017-03-31)

- (note) known issues with python 3.3, 3.4, 3.5 or 3.6 (see [sphinx-contrib/confluencebuilder#10](#))
- header/footer support
- purging support
- use macros for admonitions

## 9.12 0.4.0 (2017-02-21)

- move from `Confluence` pypi package to a `confluence` pypi package (required for publishing to pypi; see [pycontribs/confluence](#))

## 9.13 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

## 9.14 0.2.0 (2016-07-13)

- moved configuration to the sphinx config

## 9.15 0.1.1 (2016-07-12)

- added table support
- fixed internal links

## 9.16 0.1.0 (2016-07-12)

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

## INDICES AND TABLES

## Symbols

- `:columns:` (*directive option*)
  - `jira` (*directive*), 27
- `:count:` (*directive option*)
  - `jira` (*directive*), 27
- `:labels:` (*directive option*)
  - `confluence_metadata` (*directive*), 26
- `:maximum_issues:` (*directive option*)
  - `jira` (*directive*), 27
- `:server:` (*directive option*)
  - `jira` (*directive*), 27
  - `jira_issue` (*directive*), 28
- `:server-id:` (*directive option*)
  - `jira` (*directive*), 27
  - `jira_issue` (*directive*), 28
- `:server-name:` (*directive option*)
  - `jira` (*directive*), 27
  - `jira_issue` (*directive*), 28
- `:title:` (*directive option*)
  - `confluence_expand` (*directive*), 26

## B

builder

- `confluence`, 25
- `singleconfluence`, 25

## C

configuration value

- `confluence_add_secnumbers`, 12
- `confluence_additional_mime_types`, 22
- `confluence_append_labels`, 17
- `confluence_ask_password`, 13
- `confluence_ask_user`, 13
- `confluence_asset_force_standalone`, 17
- `confluence_asset_override`, 18
- `confluence_ca_cert`, 18
- `confluence_client_cert`, 18
- `confluence_client_cert_pass`, 18
- `confluence_default_alignment`, 12
- `confluence_disable_autogen_title`, 14
- `confluence_disable_notifications`, 14
- `confluence_disable_ssl_validation`, 19

- `confluence_file_suffix`, 22
- `confluence_file_transform`, 22
- `confluence_footer_file`, 12
- `confluence_global_labels`, 14
- `confluence_header_file`, 12
- `confluence_ignore_titlefix_on_index`, 19
- `confluence_jira_servers`, 23
- `confluence_lang_transform`, 23
- `confluence_link_suffix`, 23
- `confluence_link_transform`, 23
- `confluence_master_homepage`, 14
- `confluence_max_doc_depth`, 12
- `confluence_page_hierarchy`, 12
- `confluence_parent_page`, 14
- `confluence_parent_page_id_check`, 19
- `confluence_prev_next_buttons_location`, 13
- `confluence_proxy`, 19
- `confluence_publish`, 10
- `confluence_publish_allowlist`, 20
- `confluence_publish_denylist`, 20
- `confluence_publish_dryrun`, 21
- `confluence_publish_headers`, 21
- `confluence_publish_onlynew`, 21
- `confluence_publish_postfix`, 15
- `confluence_publish_prefix`, 15
- `confluence_publish_root`, 15
- `confluence_publish_subset`, 24
- `confluence_purge`, 16
- `confluence_purge_from_master`, 16
- `confluence_remove_title`, 23
- `confluence_secnumber_suffix`, 13
- `confluence_server_auth`, 21
- `confluence_server_cookies`, 22
- `confluence_server_pass`, 11
- `confluence_server_url`, 10
- `confluence_server_user`, 11
- `confluence_space_name`, 11
- `confluence_timeout`, 17
- `confluence_title_overrides`, 16
- `confluence_watch`, 17

confluence

- builder, 25

---

confluence_add_secnumbers	confluence_metadata ( <i>directive</i> ), 26
configuration value, 12	:labels: ( <i>directive option</i> ), 26
confluence_additional_mime_types	confluence_page_hierarchy
configuration value, 22	configuration value, 12
confluence_append_labels	confluence_parent_page
configuration value, 17	configuration value, 14
confluence_ask_password	confluence_parent_page_id_check
configuration value, 13	configuration value, 19
confluence_ask_user	confluence_prev_next_buttons_location
configuration value, 13	configuration value, 13
confluence_asset_force_standalone	confluence_proxy
configuration value, 17	configuration value, 19
confluence_asset_override	confluence_publish
configuration value, 18	configuration value, 10
confluence_ca_cert	confluence_publish_allowlist
configuration value, 18	configuration value, 20
confluence_client_cert	confluence_publish_denylist
configuration value, 18	configuration value, 20
confluence_client_cert_pass	confluence_publish_dryrun
configuration value, 18	configuration value, 21
confluence_default_alignment	confluence_publish_headers
configuration value, 12	configuration value, 21
confluence_disable_autogen_title	confluence_publish_onlynew
configuration value, 14	configuration value, 21
confluence_disable_notifications	confluence_publish_postfix
configuration value, 14	configuration value, 15
confluence_disable_ssl_validation	confluence_publish_prefix
configuration value, 19	configuration value, 15
confluence_expand ( <i>directive</i> ), 26	confluence_publish_root
:title: ( <i>directive option</i> ), 26	configuration value, 15
confluence_file_suffix	confluence_publish_subset
configuration value, 22	configuration value, 24
confluence_file_transform	confluence_purge
configuration value, 22	configuration value, 16
confluence_footer_file	confluence_purge_from_master
configuration value, 12	configuration value, 16
confluence_global_labels	confluence_remove_title
configuration value, 14	configuration value, 23
confluence_header_file	confluence_secnumber_suffix
configuration value, 12	configuration value, 13
confluence_ignore_titlefix_on_index	confluence_server_auth
configuration value, 19	configuration value, 21
confluence_jira_servers	confluence_server_cookies
configuration value, 23	configuration value, 22
confluence_lang_transform	confluence_server_pass
configuration value, 23	configuration value, 11
confluence_link_suffix	confluence_server_url
configuration value, 23	configuration value, 10
confluence_link_transform	confluence_server_user
configuration value, 23	configuration value, 11
confluence_master_homepage	confluence_space_name
configuration value, 14	configuration value, 11
confluence_max_doc_depth	confluence_timeout
configuration value, 12	configuration value, 17



confluence\_title\_overrides  
    configuration value, [16](#)  
confluence\_watch  
    configuration value, [17](#)

## J

jira (*directive*), [27](#)  
    :columns: (*directive option*), [27](#)  
    :count: (*directive option*), [27](#)  
    :maximum\_issues: (*directive option*), [27](#)  
    :server: (*directive option*), [27](#)  
    :server-id: (*directive option*), [27](#)  
    :server-name: (*directive option*), [27](#)  
jira\_issue (*directive*), [28](#)  
    :server: (*directive option*), [28](#)  
    :server-id: (*directive option*), [28](#)  
    :server-name: (*directive option*), [28](#)

## S

singleconfluence  
    builder, [25](#)