



Atlassian Confluence Builder

Sphinx Documentation Generation / Publisher

Provided by Sphinx Confluence Builder Contributors

2022-08-21

1.9.0

CONTENTS

1	Installing	1
2	Installing (virtualenv)	4
3	Tutorial	7
4	Configuration	10
5	Builders	33
6	Directives	34
7	Roles	39
8	Features	42
9	Tips	48
10	Guides	51
11	Advanced	55
12	Changelog	56
13	Indices and tables	65
	Index	66

INSTALLING

Atlassian Confluence Builder for Sphinx 1.9.0 depends on:

- [Python](#) 2.7 or 3.7+
- [Requests](#) 2.14.0 or later
- [Sphinx](#) 1.8 or 4.1+
- [Confluence](#) Cloud or Server 7.8+

The recommended method of installing or upgrading is using [pip](#):

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

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

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

For new users, the following provides a series of steps to assist in preparing a new environment to use this package. For users wishing to use virtualenv, please see 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 -U setuptools
(or)
python -m pip install -U 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
```

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
> (or: python -m virtualenv sphinx-venv)
> 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_key = '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 key (`confluence_space_key`) 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_key = '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 listed below. This extension may take advantage of a subset of [Sphinx configurations](#) as well when preparing documents.

New in version 1.9: All options provided by this extension may be set from the running environment. For example, if `confluence_publish` is not explicitly set inside `conf.py` or provided via [Sphinx's command line](#), this extension may check the `CONFLUENCE_PUBLISH` environment option as a fallback. Note that this only applies options provided below and will not work for other configuration options provided by Sphinx or other Sphinx extensions.

- *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_key

New in version 1.7.

Note:

- Use the key value for the space, not the name of the space. For example, `MYAWESOMESPACE` instead of `My Awesome Space`.
- The space key is **case-sensitive** (typically uppercase).

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

```
confluence_space_key = 'MYAWESOMESPACE'
```

If attempting to publish to a user's personal space, the space's key will typically start with a tilde value followed by the space's identifier. For example:

```
confluence_space_key = '~123456789'
```

confluence_server_user

Note: If using a personal access token (PAT), this option does not need to set (see [confluence_publish_token](#)).

Note: If trying to use netrc authentication, support is provided by the [Requests](#) library¹. A user can default to using a configured netrc file by not setting a value for `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.

¹ <https://requests.readthedocs.io/en/latest/user/authentication/#netrc-authentication>

```
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*).

Note: If attempting to use a personal access token (PAT), use the *confluence_publish_token* option instead.

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 = 'vsUstrSZ6Z4kMrQMapSXBYkjh'
```

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

```
confluence_server_pass = 'myawesomepassword'
```

confluence_publish_token

New in version 1.8.

Caution: It is never recommended to store a personal access tokens (PAT) 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 token from `getpass`.

Note: If attempting to use an API token, use the *confluence_server_pass* option instead.

The personal access token value used to authenticate with the Confluence instance (see *Using Personal Access Tokens*):

```
confluence_publish_token = 'AbCdEfGhIjKlMnOpQrStUvWxY/z1234567890aBc'
```

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_domain_indices

New in version 1.7.

A boolean or list value to configure whether or not generate domain-specific indices. If configured to a value of `True`, all domain-specific indices generated when processing a documentation set will have a Confluence document created. If configured with a list of index names, any matching domain-index with a matching name will have a Confluence document created. By default, domain-specific indices are disabled with a value of `False`.

```
confluence_domain_indices = True
(or)
confluence_domain_indices = [
    'py-modindex',
]
```

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_header_data](#)

confluence_header_data

New in version 1.9.

Takes an optional dictionary. If this value is set then `confluence_header_file` is interpreted as a jinja2 template with these values passed in. If this value is not set then `confluence_header_file` is included verbatim.

See also [confluence_header_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_footer_data](#)

confluence_footer_data

New in version 1.9.

Takes an optional dictionary. If this value is set then `confluence_footer_file` is interpreted as a jinja2 template with these values passed in. If this value is not set then `confluence_footer_file` is included verbatim.

See also [confluence_header_file](#).

confluence_include_search

New in version 1.7.

A boolean value to configure whether or not generate a search page. If configured to a value of `True`, a search page will be created with a search macro configured to search on the configured space. If a `search` document is registered in a documentation's `toctree`, a search page will be generated and will replace the contents of the provided search page. To avoid the implicit enablement of this feature, the generation of a search page can be explicitly disabled by setting this value to `False`. By default, search page generation is automatically managed with a value of `None`.

```
confluence_include_search = True
```

confluence_max_doc_depth

Important: This feature is deprecated. If there is a desire to generate a single document page instead, consider using the `singleconfluence builder` instead.

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 `root_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
```

confluence_page_generation_notice

New in version 1.7.

A boolean value to whether or not to generate a message at the top of each document that the page has been automatically generated. By default, this notice is disabled with a value of `False`.

```
confluence_page_generation_notice = True
```

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 `root_doc`. If a `root_doc` instance contains a `toctree`, listed documents will become child pages of the

`root_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 `root_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.

Important: This feature will default to `True` in a v2.0 release. Users who do not want to use hierarchy mode should explicitly configure this to `False` in their configurations.

`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`.

`confluence_use_index`

New in version 1.7.

A boolean value to configure whether or not generate an index page. If configured to a value of `True`, an index page will be created. If a `genindex` document is registered in a documentation's `toctree`, index content will be generated and will replace the contents of the provided `genindex` page. To avoid the implicit enablement of this feature, the generation of an index page can be explicitly disabled by setting this value to `False`. By default, index generation is automatically managed with a value of `None`.

```
confluence_use_index = True
```

`singleconfluence_toctree`

New in version 1.7.

A boolean value to configure whether or not TOC trees will remain in place when building with a `singleconfluence` builder. By default, this option is disabled with a value of `False`.

```
singleconfluence_toctree = True
```

4.3 Publishing configuration

confluence_ask_password

Warning: User's running Cygwin/MinGW may need to invoke with `wintpy` 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 `wintpy`.

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_cleanup_archive

New in version 1.9.

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

Note: This option cannot be used with `confluence_cleanup_purge`.

Warning: Only Confluence Cloud identifies support for an archiving API. Attempting to Confluence server with this feature will most likely result in an “Unsupported Confluence API call” error (500).

Attention: Confluence’s archiving API is marked as experimental at the time of writing. This feature may experience issues over time until the API is flagged as stable (if ever).

A boolean value to whether to archive legacy pages detected in a space or parent page. By default, this value is set to `False` to indicate that no pages will be archived. If this configuration is set to `True`, detected pages in Confluence that do not match the set of published documents will be automatically archived. 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 archived.

```
confluence_cleanup_archive = 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 archive operation will archive all pages that are not publish in the request. For example, if an original request publishes ten documents and archives excess documents, a following publish attempt with only one of the documents will archive the other nine pages.

See also:

- `confluence_cleanup_from_root`
- `confluence_cleanup_purge`
- `confluence_publish_dryrun`

confluence_cleanup_from_root

New in version 1.9.

A boolean value to which indicates that any cleanup attempt should be done from the root of a published `root_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 cleanup 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 `root_doc` page. This option requires either `confluence_cleanup_archive` or `confluence_cleanup_purge` to be set to `True` before taking effect. If `confluence_publish_root` is set, this option is implicitly enabled.

```
confluence_cleanup_from_root = False
```

See also:

- `confluence_cleanup_archive`
- `confluence_cleanup_purge`

confluence_cleanup_purge

New in version 1.9.

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

Note: This option cannot be used with [confluence_cleanup_archive](#).

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_cleanup_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 published 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_cleanup_archive](#)
- [confluence_cleanup_from_root](#)
- [confluence_publish_dryrun](#)

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_root_homepage

New in version 1.6.

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 `root_doc`. By default, the `root_doc` configuration is ignored with a value of `False`.

```
confluence_root_homepage = False
```

confluence_parent_page

Note: This option cannot be used with `confluence_publish_root`.

Changed in version 1.9: Support added for accepting a page identifier.

The root page found inside the configured space (`confluence_space_key`) where published pages will be a descendant of. The parent page value is used to match either the title or page identifier 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'
```

Users wishing to publish against a parent page's identifier value can do so by using an integer value instead. For example:

```
confluence_parent_page = 123456
```

If a parent page is not set, consider using the `confluence_root_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.

Changed in version 1.9: Support for the `{hash}` placeholder.

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'
```

Postfixes can include placeholders. These placeholders are filled using the `format` method so formatting types can be used. For example:

```
confluence_publish_postfix = ' ({hash:.5})'
```

Supported placeholders:

- `{hash}` - Create a reproducible hash given the title and location based from the project root. Using this placeholder provides an option for allowing pages with the same title to be pushed to the same Confluence space without needing to manually add an index to the title.

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

New in version 1.5.

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_sourcelink

New in version 1.7.

Provides options to include a link to the documentation's sources at the top of each page. This can either be a generic URL or customized to link to individual documents in a repository.

An example of a simple link is as follows:

```
confluence_sourcelink = {
    'url': 'https://www.example.com/',
}
```

Templates for popular hosting services are available. Instead of defining a `url` option, the `type` option can instead be set to one of the following types:

- `bitbucket`
- `github`
- `gitlab`

Options to set for these types are as follows:

Option	Description
owner (required)	The owner (group or user) of a project.
repo (required)	The name of the repository.
container	The folder inside the repository which is holding the documentation. This will vary per project, for example, this may be Documentation/ or doc/. If the documentation resides in the root of the repository, this option can be omitted or set to an empty string.
version (required)	The version of the sources to list. This is typically set to either a branch (e.g. main) or tag value.
view	The view mode to configure. By default, this value is set to blob for GitHub/GitLab and view for Bitbucket. GitHub/GitLab users may wish to change this to edit to create a link directly to the editing view for a specific document.
host	The hostname value to override. This option is useful for instances where a custom domain may be configured for an organization.
protocol	The protocol value to override (defaults to https).

For example, a project hosted on GitHub can use the following:

```
confluence_sourcelink = {
    'type': 'github',
    'owner': 'sphinx-contrib',
    'repo': 'confluencebuilder',
    'container': 'doc/',
    'version': 'main',
    'view': 'edit',
}
```

For unique environments, the source URL can be customized through the url option. This option is treated as a format string which can be populated based on the configuration and individual documents being processed. An example is as follows:

```
confluence_sourcelink = {
    'url': 'https://git.example.com/mydocs/{page}{suffix}',
}
```

This configures a base URL, where page and suffix will be generated automatically. Any option provided in

the `confluence_sourcelink` dictionary will be forwarded to the format option. For example:

```
confluence_sourcelink = {
    'base': 'https://git.example.com/mydocs',
    'url': '{base}/{version}/{page}{suffix}',
    'version': 'main',
}
```

The `text` option can be used to override the name of the link observed at the top of the page:

```
confluence_sourcelink = {
    ...
    'text': 'Edit Source',
}
```

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_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_cleanup_archive` and `confluence_cleanup_purge` options.

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_debug

New in version 1.8.

A boolean value to whether or not to print debug requests made to a Confluence instance. This can be helpful for users attempting to debug their connection to a Confluence instance. By default, this option is disabled with a value of False.

```
confluence_publish_debug = True
```

confluence_publish_delay

New in version 1.8.

Force a delay (in seconds) for any API calls made to a Confluence instance. By default, API requests will be made to a Confluence instance as soon as possible (or until Confluence reports that the client should be rate limiting). A user can use this option to reduce how fast this extension may attempt to interact with the Confluence instance. For example, to delay each API request by almost a 1/4 of a second, the following can be used:

```
confluence_publish_delay = 0.25
```

confluence_publish_denylist

New in version 1.3.

Note: Using this option will disable the [confluence_cleanup_archive](#) and [confluence_cleanup_purge](#) options.

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_disable_api_prefix

A boolean value which explicitly disables the use of the `rest/api` in the Confluence publish URL. This can be useful for environments where the API endpoint for a Confluence instance is proxied through a non-standard location. By default, API prefixes are enabled with a value of `False`.

```
confluence_publish_disable_api_prefix = True
```

confluence_publish_dryrun

New in version 1.3.

When a user wishes to start managing a new document set for publishing, there may be 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_intersphinx

New in version 1.9.

A publish event will upload a generated intersphinx's inventory (*object.inv*) as an attachment to the configured [root_doc](#). Inventory files are typically small and should not cause issues for most users. However, if a user desired to not publish an inventory for their documentation, this option can be configured to `False`. By default, inventories are published with a value of `True`.

```
confluence_publish_intersphinx = True
```

confluence_publish_headers

New in version 1.5.

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_request_session_override

New in version 1.7.

A hook to manipulate a [Requests](#) session prepared by this extension. Allows users who wish to perform advanced configuration of a session for features which may not be supported by this extension.

```
def my_request_session_override(session):
    session.trust_env = False

confluence_request_session_override = my_request_session_override
```

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

New in version 1.2.

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>',
}
```

confluence_version_comment

New in version 1.8.

A string value to be added as a comment to Confluence's version history.

```
confluence_version_comment = 'Automatically generated.'
```

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
```

See also:

- [Jira directives](#)
- [Jira roles](#)

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_latex_macro

New in version 1.8.

Note: Confluence does not provide stock support for LaTeX macros.

The name of a LaTeX macro to use when wishing to render LaTeX content on a Confluence instance. Stock Confluence instances do not support LaTeX content by default. However, if an instance has installed a marketplace add-on that supports LaTeX, this option can be used to hint to render LaTeX content (such as mathematical notation) by configuring this option.

```
confluence_latex_macro = 'macro-name'
(or)
confluence_latex_macro = {
    'block-macro': 'block-macro-name',
    'inline-macro': 'inline-macro-name',
    'inline-macro-param': 'inline-macro-parameter', # (optional)
}
```

The name of a LaTeX macro will vary based on which add-on is installed. For a list of known macro names or steps to determine the name of a supported macro, consult the [macro table/instructions](#) found in the math guide.

If this option is not set, any LaTeX content processed in a document will instead be converted to images using dvipng/dvisvgm (see also [sphinx.ext.imgmath](#) for additional information).

See also:

- [LaTeX directives](#)
- [LaTeX roles](#)
- [Math support](#)

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_mentions

New in version 1.9.

Provides a dictionary of key-to-value mappings which can be used with `confluence_mention` roles. When defining mentions, documents can reference a user's account identifier, user key or username (depending on the Confluence instance being published to). This configuration can be used to swap the value mentioned in a document with a value specified in configuration. For example, with the following configuration:

```
confluence_mentions = {
    'myuser': '3c5369:fa8b5c24-17f8-4340-b73e-50d383307c59',
}
```

With a document such as follows:

```
For more information, contact :confluence_mention:`myuser`:
```

The value `myuser` will be replaced with the configured account identifier. This can be useful for when trying to manage multiple user's account identifiers when targeting a Confluence Cloud instance, as well as providing a quick-way to swap a generic contact role which may change over time.

See also:

- [Mention roles](#)

confluence_navdocs_transform

New in version 1.7.

A function to override the document list used for populating navigational buttons generated from a `confluence_prev_next_buttons_location` configuration. This can be helpful in advanced publishing cases where a user would like ignore or re-order select pages from navigation, or even reference pages outside of documentation list.

```
def my_navdocs_transform(builder, docnames):
    # override and return a new docnames list
    return docnames

confluence_navdocs_transform = my_navdocs_transform
```

See also `confluence_prev_next_buttons_location`.

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_master_homepage

Changed in version 1.6.

This option has been renamed to [confluence_root_homepage](#).

confluence_parent_page_id_check

Changed in version 1.9: The [confluence_parent_page](#) option now accepts both a page name and identifier.

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_publish_subset

Changed in version 1.3.

This option has been renamed to [confluence_publish_allowlist](#).

confluence_purge_from_master

Changed in version 1.6.

This option has been renamed to [confluence_purge_from_root](#), and has since been replaced with [confluence_cleanup_from_root](#).

confluence_purge_from_root

Changed in version 1.9.

This option has been renamed to [confluence_cleanup_from_root](#).

confluence_space_name

Changed in version 1.7.

This option has been renamed to [confluence_space_key](#).

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](#)).

.. confluence_newline::

New in version 1.7.

The `confluence_newline` directive supports the injection of a newline in a document where separation may be desired between inlined elements.

```
.. confluence_newline::
```

.. confluence_toc::

New in version 1.9.

The `confluence_toc` directive allows a user to define a Confluence [Table of Contents Macro](#). Users are typically recommended to use `reStructuredText`'s [Table of Contents](#) directive when generating local table of contents; and Confluence's Table of Contents macro is typically not a replacement of Sphinx's `toctree` directive. However, if a user wishes to take advantage of Confluence's TOC-specific macro capabilities, the following can be used:

```
.. confluence_toc::
```

This directive supports the following options:

:absolute-url: flag (boolean)

Whether the macro should generate full URLs for TOC entry links. Valid values are `true` or `false` (default).

```
.. confluence_toc::
    :absolute-url: true
```

:exclude: value (string)

Filter heading to exclude entries matching the provided value. The value should support a regular expressions string.

:include: value (string)

Filter heading to include entries matching the provided value. The value should support a regular expressions string.

:indent: value (string)

The indent to apply for header entries.

```
.. confluence_toc::
    :indent: 15px
```

:max-level: count (number)

Defines the lowest heading level to include in the table of contents.

```
.. confluence_toc::
    :max-level: 10
```

:min-level: count (number)

Defines the highest heading level to include in the table of contents.

```
.. confluence_toc::
   :min-level: 1
```

:outline: flag (boolean)

Whether the macro should include outline numbering for entries. Valid values are `true` or `false` (default).

```
.. confluence_toc::
   :outline: true
```

:printable: flag (boolean)

Whether the macro should render when a user prints a Confluence page. Valid values are `true` (default) or `false`.

```
.. confluence_toc::
   :printable: true
```

:separator: separator style type of the toc (brackets, braces, parens, <user-defined>)

When the type option is configured to flat, this option can configure the separator type applied between header entries. By default, the separator type is set to brackets.

```
.. confluence_toc::
   :separator: braces
```

:style: list style type of the toc (default, none, disc, circle, square, decimal, lower-alpha, lower-roman, upper-roman)

Configures how the table of contents will be style its list type. By default, the style type is set to default.

```
.. confluence_toc::
   :style: square
```

:type: outline type of the toc (flat or list)

Configures how the table of contents will be style its structure. Valid values are `flat` or `list` (default).

```
.. confluence_toc::
   :type: flat
```

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
```

See also *Jira roles*.

6.3 LaTeX

Note: LaTeX support requires `dvipng/dvisvgm` to be installed on system; however, if a Confluence instance supports a LaTeX macro, the `confluence_latex_macro` ([ref](#)) option can be used instead. For more information, please read *Math support*.

The following directive can be used to help add LaTeX content into a Confluence page.

.. confluence_latex::

New in version 1.8.

The `confluence_latex` directive allows a user to add LaTeX content into a document. For example:

```
.. confluence_latex::

   $\mathfrak{H}$ello world!
```

This directive supports the following options:

:align: "left", "center", or "right" (string)

The alignment to apply on the LaTeX content. By default, the value is set to `center`.

```
.. confluence_latex::
   :align: left

   $\mathfrak{H}$ello world!
```

See also *LaTeX roles*.

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

7.1 Emoticon Macro

The following role can be used to help include Confluence emoticon macros into generated Confluence documents.

:confluence_emoticon:

New in version 1.9.

The `confluence_emoticon` role allows a user to build inlined emoticon macros. For example:

```
:confluence_emoticon:`tick`: This is done.  
:confluence_emoticon:`cross`: This is incomplete.
```

7.2 Jira

The following role can be used to help include Jira macros into generated Confluence documents.

:jira:

New in version 1.7.

The `jira` role allows a user to build an inlined Jira macro to be configured with a provided Jira key. For example:

```
See :jira:`TEST-123` for more details.
```

See also [Jira directives](#).

7.3 LaTeX

Note: LaTeX support requires `dvipng`/`dvisvgm` to be installed on system; however, if a Confluence instance supports a LaTeX macro, the `confluence_latex_macro` ([ref](#)) option can be used instead. For more information, please read [Math support](#).

The following role can be used to help include LaTeX content into generated Confluence documents.

:confluence_latex:

New in version 1.8.

The `confluence_latex` role allows a user to build inlined LaTeX content. For example:

```
This is a :confluence_latex: `$\\mathfrak{t}$est`.
```

See also *LaTeX directives*.

7.4 Mentions

The following role can be used to help include *Confluence mentions* into generated Confluence documents.

:confluence_mention:

New in version 1.9.

Warning: Confluence Cloud mentions should always use account identifiers; where Confluence Server mentions should use either usernames or user keys. Attempting to use Confluence Cloud account identifiers when publishing to a Confluence server will most likely result in an “Unsupported Confluence API call” error (500).

The `confluence_mention` role allows a user to build inlined mentions. For Confluence Cloud instances, a mention to a specific user’s account identifier would be defined as follows:

```
See :confluence_mention: `3c5369:fa8b5c24-17f8-4340-b73e-50d383307c59`.
```

For Confluence Server instances, a mention to a specific user can either be set to the username value, or a user’s key value:

```
For more information, contact :confluence_mention: `myuser`.  
(or)  
Contact :confluence_mention: `b9aaf35e80441f415c3a3d3c53695d0e` for help.
```

A user mapping table can also be configured using the `confluence_mentions` (*ref*) option.

7.5 Status Macro

The following role can be used to help include *Confluence status macro* into generated Confluence documents.

:confluence_status:

New in version 1.9.

The `confluence_status` role allows a user to build inlined status macros. For example:

```
:confluence_status: `My Status`
```

The color of a status macro can be configured to a value supported by Confluence’s status macro. For example, to adjust the status value to a yellow color, the following can be used:

```
:confluence_status: `WARNING <yellow>`
```

To tweak the style of a status macro to an outlined variant, adjust the color enclosure to square brackets:

```
:confluence_status: `PASSED <green>`
```

FEATURES

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.

8.1 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)

8.2 Sphinx markup

Type	Notes
centered	Supported
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).
deprecated	Supported
download	Supported
glossary	Supported
hlist	Supported
manpage	Supported
production list	Supported
toctree	Supported
versionadded	Supported
versionchanged	Supported

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

8.3 Extensions

This extension will attempt to support any extension that is integrated in Sphinx's main source repository. The following section shows the status of each of these extensions:

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 have been tested. If a type of reference does not work or yields unexpected results, please report the issue with an example.
<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.

8.4 Extensions (Third-party)

Note: Support with third-party extensions (if any) may be limited. While an extension may work with other builders (such as the `html` builder), it may be implemented in a way which it cannot be integrated with this extension. In addition, some features that an extension may use (e.g. using JavaScript) cannot be used with a stock Confluence instance and therefore, cannot be supported.

Atlassian Confluence Builder for Sphinx does not guarantee support for third-party extensions found outside of [Sphinx's main source repository](#). Changes are considered towards this extension's processing and API to make it flexible for other extensions to integrate. Also, this extension *may* provide optional support for select third-party extensions (if permitted and rational), to help improve user experience.

Developers wishing to integrate a third-party extension with Atlassian Confluence Builder can either attempt to add implementation inside the third-party extension itself, propose non-intrusive changes to this extension or create a new extension which can bind both desired extensions together (e.g. [sphinx-confluence-nbsphinx-test](#)). Any changes directly submitted to be added into this extension's repository will be limited to the arbitrary release/development windows of this extension.

The following table show a most recent state of various third-party extensions interacting with this extension:

Type	Notes
mlx.traceability	Limited support. Formatting of content may not be as expected.
sphinx-diagrams	Supported
sphinx-gallery	Supported
sphinx-needs	Limited support. Formatting of content may not be as expected. The <code>needs_default_layout</code> option may need to be tailored specifically for a Confluence build.
sphinx-toolbox	Supported
sphinxcontrib-aafig	Supported. May require configuration of the <code>aafig_format</code> option.
sphinxcontrib-actdiag	Limited support. PNGs only; cannot configure for SVG at this time.
sphinxcontrib-blockdiag	Limited support. PNGs only; cannot configure for SVG at this time.
sphinxcontrib-drawio	Supported
sphinxcontrib-httpdomain	Supported
sphinxcontrib-kroki	Supported
sphinxcontrib-mermaid	Limited support. Requires a PNG/SVG configuration.
sphinxcontrib-nwdiag	Limited support. PNGs only; cannot configure for SVG at this time.
sphinxcontrib-openapi	Supported
sphinxcontrib-plantuml	Supported. <code>sphinxcontrib-plantuml</code> provides its own support for this extension.
sphinxcontrib-programoutput	Supported
sphinxcontrib-seqdiag	Limited support. PNGs only; cannot configure for SVG at this time.
sphinxcontrib-svgbob	Supported
sphinxcontrib-youtube	Supported

8.5 Other

If a feature 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>

9.1 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](#).

9.2 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`; [ref](#)).

9.3 Wiping a space

Danger: Pages removed from this call cannot be recovered without the help of an administrator for the Confluence space which pages will be removed from.

A command line argument `wipe` is available for users wishing to remove pages from a configured space. This can be useful for users who need to clear multiple pages which have been pushed through automation or if the Confluence instance does not support automatically deleting page children.

A wipe request can be started using the following:

```
$ python -m sphinxcontrib.confluencebuilder wipe --danger
...
Are you sure you want to continue? [y/N] y

      URL: https://intranet-wiki.example.com/
    Space: TEST
    Pages: All Pages
Total pages: 250

Are you sure you want to REMOVE these pages? [y/N] y

Removing pages.... done
```

If a user wishes to only remove child pages of a *configured parent page*, the option `--parent` can be used:

```
python -m sphinxcontrib.confluencebuilder wipe --danger --parent
```

9.4 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 ]-----
```

The following contains a series of guides which may help a user using the Confluence Builder extension in a Sphinx-enabled environment.

10.1 Math support

Changed in version 1.8: Support for using Confluence-supported LaTeX macros.

Changed in version 1.7: SVG-generated math images requires the `sphinx.ext.imgmath` extension to be explicitly registered to work.

Changed in version 1.4: Support both block and inlined math elements.

New in version 1.2: Initial support for math.

Mathematical notation is supported when using the Confluence builder extension. There are some limitations and special cases a user may wish to consider when trying to get the most out of their documentation/environment.

The recommended approach for rendering math in Confluence is using Sphinx's `sphinx.ext.imgmath` extension. This extension renders math via LaTeX and uses either `dvipng` or `dvisvgm` to create PNG or SVG images which are published to Confluence (which requires installing a LaTeX engine alongside the Sphinx installation, such as a TeX suite/MiKTeX installation).

Math can be defined using the `:math:` role or the `.. math::` directive. For example:

The Pythagorean theorem formula is `:math:`a^2 + b^2 = c^2``.

The mass-energy equivalence formula:

`.. math::`

$$E = mc^2$$

The former case will generate math content in an inlined fashion, where the latter will generate a math block. Math defined in this way is supported on Sphinx's standard builders with the help of other Sphinx-provided math extensions.

10.1.1 Recommended options for math

Ensure the `sphinx.ext.imgmath` extension is added to the documentation’s extension list:

```
extensions = [
    'sphinx.ext.imgmath',
    'sphinxcontrib.confluencebuilder',
]
```

If loading the `sphinx.ext.imgmath` extension conflicts with other builders, the extension can be optionally included using an approach such as follows (but may require additional changes depending on the environment):

```
import sys

extensions = [
    'sphinxcontrib.confluencebuilder',
]

if any('confluence' in arg for arg in sys.argv):
    extensions.append('sphinx.ext.imgmath')
```

The following are the recommended options to use:

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

This configures the generated font size commonly observed on Confluence instances, hints that images are generated into SVG images (for ideal image scaling) and attempt to process the “depth” of an image to better align content alongside text.

10.1.2 Support for LaTeX macros for math

Note: This section outlines how users can take advantage of LaTeX macros enabled on their Confluence instance. The Confluence builder extension only helps forward LaTeX-generated content directly into macros. How a LaTeX macro decides to handle/render LaTeX content is up to the provider supporting the macro.

Note: Not all LaTeX macros support block and inline macros (normally, just the former).

Note: Confluence builder will attempt to support numbered equations by adding a floating label alongside a rendered math block. The variety of LaTeX macros which exist and limitations in how LaTeX macros are structured in a page may prevent the ability to perfectly align these labels alongside the rendered math content.

A stock Confluence instance does not provide LaTeX support. This is the main reason why the Confluence builder extension promotes the use of `sphinx.ext.imgmath`. However, if a user’s Confluence instance supports a marketplace add-on which provides LaTeX macro support, math content can instead be injected into these macros instead.

To use a LaTeX macro, the `confluence_latex_macro` (*ref*) configuration option can be used. This option accepts either the name of a macro to use or a dictionary of macro options to consider (the dictionary is for more complex

configurations such as when attempting to support block-specific and inlined-specific macros). For example, to specify the macro to use for any LaTeX content, the following can be used:

```
confluence_latex_macro = 'macro-name'
```

If an environment supports a macro which supports block and inlined content in different macros, the following can be used:

```
confluence_latex_macro = {
    'block-macro': 'block-macro-name',
    'inline-macro': 'inline-macro-name',
    'inline-macro-param': 'inline-macro-parameter', # (optional)
}
```

LaTeX macro names

What macro names to use will vary based off which macro types are installed (if any). Please consult the following table for reported macro names:

Marketplace Application	Configuration
Content Formatting Macros for Confluence	<pre>confluence_latex_macro = 'latex-formatting'</pre>
LaTeX for Confluence	<pre>confluence_latex_macro = 'orah-latex'</pre>
LaTeX Math	<pre>confluence_latex_macro = { 'block-macro': 'mathblock', 'inline-macro': 'mathinline', 'inline-macro-param': 'body', }</pre>

If a Confluence environment supports a different macro type, a user can determine the name of the macro by:

1. Creating a new page on the Confluence instance
2. Adding a LaTeX macro on the page and saving
3. Selecting the page's option menu and selecting "View Storage format"
4. Look for the corresponding macro name inside an `ac:name` attribute (in this case, the macro's name is `my-latex-macro`):

```
<ac:structured-macro ac:name="my-latex-macro" ...>
  ...
</ac:structured-macro>
```

10.2 Publishing with CI

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

Tip: As of v1.9, this extension can automatically accept configuration options from the environment that have not been explicitly set in `conf.py` or provided by the command line. For example, to configure for authentication using a personal access token using the environment, do not add `confluence_publish_token` into `conf.py` and ensure the `CONFLUENCE_PUBLISH_TOKEN` environment variable is set.

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, if authenticating with an API key (see [API tokens](#)), the following can be used:

```
import os

...

confluence_server_user = 'api-key-uid'
confluence_server_pass = os.getenv('SECRET_KEY')
```

Or, if using a personal access token (see [Using Personal Access Tokens](#)), the following can be used:

```
import os

...

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

Both examples above will read an environment variable `SECRET_KEY` prepared in a CI environment to be used for authentication.

See `confluence_server_pass` ([ref](#)) and `confluence_publish_token` ([ref](#)) for more information.

11.1 Publishing permissions

An account used for authentication will have the best experience with the following [permissions](#) when attempting to publish to a configured space:

- Space – View
- Page – Add, Delete
- Attachments – Add, Delete

Delete permissions are only required for environments using the `confluence_cleanup_purge` ([ref](#)) capability.

For environments using an OAuth connector, the following scopes are required:

```
delete:content:confluence
read:content-details:confluence
write:attachment:confluence
write:content:confluence
write:watcher:confluence
```

CHANGELOG

12.1 1.9.0 (2022-08-21)

- **(note)** Final release supporting Python 2.7.
- Allow `confluence_ca_cert` to accept a CA-bundle path
- Default publish Intersphinx databases to root documents
- Fixed issue where code blocks may not properly render captions
- Fixed issue where dry-runs may fail with a non-existent root page
- Fixed issue where external references with ampersands would fail to publish
- Fixed issue where labels are missing on first-publish on Confluence server
- Fixed issue where title length limits were not properly enforced
- Improve support for loading local windows CA/root certificates for requests
- Introduce the Confluence emoticon (macro) role
- Introduce the Confluence status (macro) role
- Introduce the Confluence toc (macro) directive
- Introduce the Confluence user mention (macro) role
- Relax domain restrictions for Intersphinx generation
- Support `confluence_parent_page` to accept a page identifier
- Support archiving legacy pages (alternative to purging)
- Support configuring configuration options from environment
- Support document-specific reproducible hash injection in a postfix
- Support jinja2 templating on header/footer templates
- Support the ability to disable injecting `rest/api` in API endpoint url

12.2 1.8.0 (2022-03-27)

- **(note)** `confluence_max_doc_depth` support is deprecated (reminder)
- **(note)** (advanced mode) Macro disabling is deprecated
- Add configuration for personal access tokens (to avoid header entry changes)
- Fixed issue where Confluence cloud did not update homepage on a personal space
- Fixed issue where inline internal targets would create an empty link label
- Fixed issue where Jira directives/role could not be substituted
- Improve formatting when processing autodoc content (containers)
- Improve support when using the `sphinx-diagrams` extension
- Improve table-alignment for markdown
- Introduce the `latex` directive/role
- removed informational macro styling on figures
- Support rate limiting for publish requests
- Support setting the comment field in page update events
- Support users to render math through available marketplace latex macros

12.3 1.7.1 (2021-11-30)

- Fixed regression where `publish-root/dryrun` modes would fail with an exception

12.4 1.7.0 (2021-11-21)

- **(note)** `confluence_max_doc_depth` support is deprecated (reminder)
- `confluence_space_name` renamed to `confluence_space_key`
- Brackets will be wrapped around bottom footnote entries
- Fixed issue where links to numbered section would not work
- Fixed issue where publishing could fail without a proxy set for older requests
- Fixed issue where `report/wipe` commands would fail in Python 2.7
- Fixed regression in `timeout` option
- Improve dry-run reflecting new attachments to be published
- Improve indentations for line-block content
- Improve support for attached SVG images with length/scaling modifiers
- Improve support for non-pixel length units for images
- Improve support for SVG images without an XML declaration
- Improve support when publishing page updates converted to a new editor
- Improve support when using the `sphinx-gallery` extension

- Improve support when using the `sphinx_toolbox` extension
- Improve support when using the `sphinxcontrib-mermaid` extension
- Improve support when using the `sphinxcontrib-needs` extension
- Improve support when using the `sphinxcontrib-youtube` extension
- Improve user feedback on ancestor page update failures (500 errors)
- Improved support for dealing with unreconciled page detections
- Introduce the Jira role
- Introduce the newline directive
- Introduce the page generation notice option (notice for top of documents)
- Introduce the source link option (e.g. “Edit Source” link)
- Prevent issues with extension directives causing errors with other builders
- Provide a configuration hook to override requests session information
- Remove borders on footnote tables
- Support domain indices generation/processing
- Support for leaving resolved toctrees for singleconfluence
- Support genindex generation/processing
- Support search generation
- Support strikethrough through strike role
- Support the consideration of `confluence_parent_page` for wipe requests

12.5 1.6.0 (2021-09-26)

- **(note)** `confluence_max_doc_depth` support is deprecated
- `confluence_master_homepage` renamed to `confluence_root_homepage`
- `confluence_purge_from_master` renamed to `confluence_purge_from_root`
- Always publish without XSRF checks (minimize Confluence instance logging)
- Always strip control characters from content
- Ensure publish events use legacy editor
- Fixed issue where `sphinx.ext.imgmath` was forced for non-Confluence builds
- Fixed issue where rubrics were built as headers and not titled paragraphs
- Handle extensions providing Unicode attributes (e.g. `sphinxcontrib.drawio`)
- Improve formatting when processing markdown content
- Prevent exceptions where third-party extensions provide invalid image uris
- Support rendering explicit newline entries

12.6 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

12.7 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)

12.8 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

12.9 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++ autodoocs)
- 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

12.10 1.1.0 (2019-03-16)

- Repackaged release (see [sphinx-contrib/confluencebuilder#192](https://github.com/sphinx-contrib/confluencebuilder/pull/192))

12.11 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

12.12 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

12.13 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

12.14 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

12.15 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

12.16 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

12.17 0.4.0 (2017-02-21)

- Move from Confluence PyPI package to a confluence PyPI package (required for publishing to PyPI; see [pycontribs/confluence](#))

12.18 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

12.19 0.2.0 (2016-07-13)

- Moved configuration to the Sphinx config

12.20 0.1.1 (2016-07-12)

- Added table support
- Fixed internal links

12.21 0.1.0 (2016-07-12)

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

INDICES AND TABLES

Symbols

- `:absolute-url:` (*directive option*)
 - `confluence_toc` (*directive*), 35
- `:align:` (*directive option*)
 - `confluence_latex` (*directive*), 38
- `:columns:` (*directive option*)
 - `jira` (*directive*), 36
- `:count:` (*directive option*)
 - `jira` (*directive*), 37
- `:exclude:` (*directive option*)
 - `confluence_toc` (*directive*), 35
- `:include:` (*directive option*)
 - `confluence_toc` (*directive*), 35
- `:indent:` (*directive option*)
 - `confluence_toc` (*directive*), 35
- `:labels:` (*directive option*)
 - `confluence_metadata` (*directive*), 34
- `:max-level:` (*directive option*)
 - `confluence_toc` (*directive*), 35
- `:maximum_issues:` (*directive option*)
 - `jira` (*directive*), 37
- `:min-level:` (*directive option*)
 - `confluence_toc` (*directive*), 35
- `:outline:` (*directive option*)
 - `confluence_toc` (*directive*), 36
- `:printable:` (*directive option*)
 - `confluence_toc` (*directive*), 36
- `:separator:` (*directive option*)
 - `confluence_toc` (*directive*), 36
- `:server:` (*directive option*)
 - `jira` (*directive*), 37
 - `jira_issue` (*directive*), 37
- `:server-id:` (*directive option*)
 - `jira` (*directive*), 37
 - `jira_issue` (*directive*), 38
- `:server-name:` (*directive option*)
 - `jira` (*directive*), 37
 - `jira_issue` (*directive*), 38
- `:style:` (*directive option*)
 - `confluence_toc` (*directive*), 36
- `:title:` (*directive option*)
 - `confluence_expand` (*directive*), 34

- `:type:` (*directive option*)
 - `confluence_toc` (*directive*), 36

B

- builder
 - `confluence`, 33
 - `singleconfluence`, 33

C

- configuration value
 - `confluence_add_secnumbers`, 13
 - `confluence_additional_mime_types`, 29
 - `confluence_append_labels`, 23
 - `confluence_ask_password`, 16
 - `confluence_ask_user`, 16
 - `confluence_asset_force_standalone`, 23
 - `confluence_asset_override`, 23
 - `confluence_ca_cert`, 23
 - `confluence_cleanup_archive`, 16
 - `confluence_cleanup_from_root`, 17
 - `confluence_cleanup_purge`, 17
 - `confluence_client_cert`, 24
 - `confluence_client_cert_pass`, 24
 - `confluence_default_alignment`, 13
 - `confluence_disable_autogen_title`, 16
 - `confluence_disable_notifications`, 18
 - `confluence_disable_ssl_validation`, 24
 - `confluence_domain_indices`, 13
 - `confluence_file_suffix`, 29
 - `confluence_file_transform`, 29
 - `confluence_footer_data`, 14
 - `confluence_footer_file`, 13
 - `confluence_global_labels`, 18
 - `confluence_header_data`, 13
 - `confluence_header_file`, 13
 - `confluence_ignore_titlefix_on_index`, 25
 - `confluence_include_search`, 14
 - `confluence_jira_servers`, 29
 - `confluence_lang_transform`, 30
 - `confluence_latex_macro`, 30
 - `confluence_link_suffix`, 30
 - `confluence_link_transform`, 30

- confluence_master_homepage, 32
- confluence_max_doc_depth, 14
- confluence_mentions, 31
- confluence_navdocs_transform, 31
- confluence_page_generation_notice, 14
- confluence_page_hierarchy, 14
- confluence_parent_page, 19
- confluence_parent_page_id_check, 32
- confluence_prev_next_buttons_location, 15
- confluence_proxy, 25
- confluence_publish, 11
- confluence_publish_allowlist, 25
- confluence_publish_debug, 26
- confluence_publish_delay, 26
- confluence_publish_denylist, 26
- confluence_publish_disable_api_prefix, 27
- confluence_publish_dryrun, 27
- confluence_publish_headers, 27
- confluence_publish_intersphinx, 27
- confluence_publish_onlynew, 27
- confluence_publish_postfix, 19
- confluence_publish_prefix, 20
- confluence_publish_root, 20
- confluence_publish_subset, 32
- confluence_publish_token, 12
- confluence_purge_from_master, 32
- confluence_purge_from_root, 32
- confluence_remove_title, 31
- confluence_request_session_override, 28
- confluence_root_homepage, 18
- confluence_secnumber_suffix, 15
- confluence_server_auth, 28
- confluence_server_cookies, 28
- confluence_server_pass, 12
- confluence_server_url, 11
- confluence_server_user, 11
- confluence_sourcelink, 20
- confluence_space_key, 11
- confluence_space_name, 32
- confluence_timeout, 22
- confluence_title_overrides, 22
- confluence_use_index, 15
- confluence_version_comment, 28
- confluence_watch, 22
- singleconfluence_toc tree, 15
- confluence
 - builder, 33
- confluence_add_secnumbers
 - configuration value, 13
- confluence_additional_mime_types
 - configuration value, 29
- confluence_append_labels
 - configuration value, 23
- confluence_ask_password
 - configuration value, 16
- confluence_ask_user
 - configuration value, 16
- confluence_asset_force_standalone
 - configuration value, 23
- confluence_asset_override
 - configuration value, 23
- confluence_ca_cert
 - configuration value, 23
- confluence_cleanup_archive
 - configuration value, 16
- confluence_cleanup_from_root
 - configuration value, 17
- confluence_cleanup_purge
 - configuration value, 17
- confluence_client_cert
 - configuration value, 24
- confluence_client_cert_pass
 - configuration value, 24
- confluence_default_alignment
 - configuration value, 13
- confluence_disable_autogen_title
 - configuration value, 16
- confluence_disable_notifications
 - configuration value, 18
- confluence_disable_ssl_validation
 - configuration value, 24
- confluence_domain_indices
 - configuration value, 13
- confluence_emoticon (*role*), 39
- confluence_expand (*directive*), 34
 - :title: (*directive option*), 34
- confluence_file_suffix
 - configuration value, 29
- confluence_file_transform
 - configuration value, 29
- confluence_footer_data
 - configuration value, 14
- confluence_footer_file
 - configuration value, 13
- confluence_global_labels
 - configuration value, 18
- confluence_header_data
 - configuration value, 13
- confluence_header_file
 - configuration value, 13
- confluence_ignore_titlefix_on_index
 - configuration value, 25
- confluence_include_search
 - configuration value, 14
- confluence_jira_servers
 - configuration value, 29
- confluence_lang_transform
 - configuration value, 30

- confluence_latex (*directive*), 38
 - :align: (*directive option*), 38
- confluence_latex (*role*), 39
- confluence_latex_macro
 - configuration value, 30
- confluence_link_suffix
 - configuration value, 30
- confluence_link_transform
 - configuration value, 30
- confluence_master_homepage
 - configuration value, 32
- confluence_max_doc_depth
 - configuration value, 14
- confluence_mention (*role*), 40
- confluence_mentions
 - configuration value, 31
- confluence_metadata (*directive*), 34
 - :labels: (*directive option*), 34
- confluence_navdocs_transform
 - configuration value, 31
- confluence_newline (*directive*), 35
- confluence_page_generation_notice
 - configuration value, 14
- confluence_page_hierarchy
 - configuration value, 14
- confluence_parent_page
 - configuration value, 19
- confluence_parent_page_id_check
 - configuration value, 32
- confluence_prev_next_buttons_location
 - configuration value, 15
- confluence_proxy
 - configuration value, 25
- confluence_publish
 - configuration value, 11
- confluence_publish_allowlist
 - configuration value, 25
- confluence_publish_debug
 - configuration value, 26
- confluence_publish_delay
 - configuration value, 26
- confluence_publish_denylist
 - configuration value, 26
- confluence_publish_disable_api_prefix
 - configuration value, 27
- confluence_publish_dryrun
 - configuration value, 27
- confluence_publish_headers
 - configuration value, 27
- confluence_publish_intersphinx
 - configuration value, 27
- confluence_publish_onlynew
 - configuration value, 27
- confluence_publish_postfix
 - configuration value, 19
- confluence_publish_prefix
 - configuration value, 20
- confluence_publish_root
 - configuration value, 20
- confluence_publish_subset
 - configuration value, 32
- confluence_publish_token
 - configuration value, 12
- confluence_purge_from_master
 - configuration value, 32
- confluence_purge_from_root
 - configuration value, 32
- confluence_remove_title
 - configuration value, 31
- confluence_request_session_override
 - configuration value, 28
- confluence_root_homepage
 - configuration value, 18
- confluence_secnumber_suffix
 - configuration value, 15
- confluence_server_auth
 - configuration value, 28
- confluence_server_cookies
 - configuration value, 28
- confluence_server_pass
 - configuration value, 12
- confluence_server_url
 - configuration value, 11
- confluence_server_user
 - configuration value, 11
- confluence_sourcelink
 - configuration value, 20
- confluence_space_key
 - configuration value, 11
- confluence_space_name
 - configuration value, 32
- confluence_status (*role*), 40
- confluence_timeout
 - configuration value, 22
- confluence_title_overrides
 - configuration value, 22
- confluence_toc (*directive*), 35
 - :absolute-url: (*directive option*), 35
 - :exclude: (*directive option*), 35
 - :include: (*directive option*), 35
 - :indent: (*directive option*), 35
 - :max-level: (*directive option*), 35
 - :min-level: (*directive option*), 35
 - :outline: (*directive option*), 36
 - :printable: (*directive option*), 36
 - :separator: (*directive option*), 36
 - :style: (*directive option*), 36
 - :type: (*directive option*), 36

confluence_use_index
 configuration value, 15
 confluence_version_comment
 configuration value, 28
 confluence_watch
 configuration value, 22
 Continuous integration, 53

E

Emoticon Macro, 39
 Expand Macro, 34

J

Jira
 Adding a Jira table, 36
 Adding a single Jira link (*directive*), 37
 Adding a single Jira link (*role*), 39
 Configuring Jira servers, 29
 jira (*directive*), 36
 :columns: (*directive option*), 36
 :count: (*directive option*), 37
 :maximum_issues: (*directive option*), 37
 :server: (*directive option*), 37
 :server-id: (*directive option*), 37
 :server-name: (*directive option*), 37
 jira (*role*), 39
 jira_issue (*directive*), 37
 :server: (*directive option*), 37
 :server-id: (*directive option*), 38
 :server-name: (*directive option*), 38

L

LaTeX Macro
 Adding a LaTeX block (*directive*), 38
 Adding inlined LaTeX (*role*), 39

M

Macros
 Emoticon Macro (*role*), 39
 Expand Macro (*directive*), 34
 Jira Macro (*directive*), 36
 Jira Macro (*role*), 39
 LaTeX Macro (*directive*), 38
 LaTeX Macro (*role*), 39
 Mentions Macro (*role*), 40
 Status Macro (*role*), 40
 Math
 LaTeX macros, 52
 Math support, 51
 Recommended options, 51
 Mentions
 Configuration, 31
 Macro (*role*), 40

P

Page removal
 Automatically archiving pages, 16
 Automatically purging pages, 17
 Wiping a space, 49

S

singleconfluence
 builder, 33
 singleconfluence_toc tree
 configuration value, 15
 Status Macro, 40

W

Wiping a space, 49