



Atlassian Confluence Builder

Sphinx Documentation Generation / Publisher

Provided by Sphinx Confluence Builder Contributors

2026-06-13

3.2.0.dev0

CONTENTS

1	Introduction	1
2	Installing	4
3	Installing (virtualenv)	9
4	Tutorial	13
5	Configuration	18
6	Builders	61
7	Directives	62
8	Roles	76
9	Events	84
10	Features	87
11	Tips	96
12	Guides	101
13	Advanced	117
14	Changelog	126
	Index	175

INTRODUCTION

Atlassian® Confluence® Builder for Sphinx is a community provided extension to help build Confluence supported format files (e.g. storage format) and publish them to a Confluence instance.

- License: BSD-2-Clause
- Confluence Cloud / Data Center 9.0+
- Python 3.10+
- Requests 2.30.0+
- Sphinx 8.0+

The community and development for this extension can all be found at this project's GitHub repository:

Atlassian Confluence Builder for Sphinx — GitHub
<https://github.com/sphinx-contrib/confluencebuilder>

1.1 Demonstration

Published validation examples can be found on the following Confluence instance:

Atlassian Confluence Builder for Sphinx — Online Demo on Confluence Cloud
<https://sphinxcontrib-confluencebuilder.atlassian.net/>

1.2 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>

INSTALLING

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`.

2.1 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:

2.1.1 Arch

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

2.1.2 Fedora

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

2.1.3 Ubuntu

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

2.2 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>
```

2.3 Windows

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

Python — Downloads

<https://www.python.org/downloads/>

When invoking the installer, it is recommended to select the option to “Add Python to PATH”; however, users can explicitly 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>
```

2.4 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.

3.1 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:

3.1.1 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>
```

3.1.2 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>
```

3.1.3 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>
```

3.2 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>
```

3.3 Windows (virtualenv)

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

Python — Downloads
<https://www.python.org/downloads/>

When invoking the installer, it is recommended to select the option to “Add Python to PATH”; however, users can explicitly 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.

4.1 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](#).

4.1.1 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](#).

4.1.2 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:

```
extensions = []
```

After preparing these files, continue by [configuring for this extension](#) as follows.

4.2 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 Data Center)  
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).

4.3 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.

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](#).

4.4 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 -M confluence . _build -E -a
(or)
python -m sphinx -M confluence . _build -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

All configurations introduced by this extension are listed below. These may be used in a documentation project's [Sphinx configuration](#) (`conf.py`). This extension may take advantage of a subset of Sphinx-defined configurations as well when preparing documents.

Added 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 to options provided below and will not work for other configuration options provided by Sphinx or other Sphinx extensions.

- [Example configurations](#)
- [Essential configuration](#)
- [Generic configuration](#)
- [Publishing configuration](#)
- [Advanced publishing configuration](#)
- [Advanced processing configuration](#)
- [Third-party related options](#)
- [Other options](#)
- [Deprecated options](#)

5.1 Example configurations

The following shows example configurations (`conf.py`) for Confluence generation and publishing:

5.1.1 Example Cloud Configuration

```
extensions = [  
    'sphinxcontrib.confluencebuilder',  
]  
confluence_publish = True  
confluence_space_key = 'MYPRJ'  
confluence_server_url = 'https://example.atlassian.net/'  
confluence_parent_page = 'Documentation'  
confluence_server_user = 'myawesomeuser@example.com'
```

With use of a token environment variable:

```
export CONFLUENCE_API_TOKEN=TOKEN  
(or)  
set CONFLUENCE_API_TOKEN=TOKEN
```

5.1.2 Example Cloud (Scoped Token) Configuration

```
extensions = [  
    'sphinxcontrib.confluencebuilder',  
]  
confluence_publish = True  
confluence_space_key = 'MYPRJ'  
confluence_server_url = 'https://example.atlassian.net/'  
confluence_parent_page = 'Documentation'  
confluence_server_user = 'myawesomeuser@example.com'  
confluence_api_token_scoped = True
```

With use of a token environment variable:

```
export CONFLUENCE_API_TOKEN=TOKEN  
(or)  
set CONFLUENCE_API_TOKEN=TOKEN
```

5.1.3 Example Cloud (Custom Domain) Configuration

```
extensions = [  
    'sphinxcontrib.confluencebuilder',  
]  
confluence_publish = True  
confluence_space_key = 'MYPRJ'  
confluence_server_url = 'https://wiki.example.com/'  
confluence_parent_page = 'Documentation'  
confluence_server_user = 'myawesomeuser@example.com'  
confluence_cloud = True
```

With use of a token environment variable:

```
export CONFLUENCE_API_TOKEN=TOKEN  
(or)  
set CONFLUENCE_API_TOKEN=TOKEN
```

5.1.4 Example Data Center Configuration

```
extensions = [  
    'sphinxcontrib.confluencebuilder',  
]  
confluence_publish = True  
confluence_space_key = 'MYPRJ'  
confluence_server_url = 'https://wiki.example.com/'  
confluence_parent_page = 'Documentation'
```

With use of a token environment variable:

```
export CONFLUENCE_PUBLISH_TOKEN=TOKEN  
(or)  
set CONFLUENCE_PUBLISH_TOKEN=TOKEN
```

5.2 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
```

Re-publishing supports checking for page changes. Pages that have not changed will not be updated (see `confluence_publish_force`).

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 Confluence Data Center, 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

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'
```

Added in version 1.7.

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 Data Center, this value will most likely be the username value.

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

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

confluence_api_token

Tip

Use this option for Confluence Cloud.

Note

If using a scoped API token, users are required to also enable `confluence_api_token_scoped` option or ensure their `confluence_server_url` configuration points to a modern Confluence API endpoint (“api.atlassian.com”) for their site.

View [required scope permission](#).

Caution

It is never recommended to store an API token 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 a personal access token (PAT), use the `confluence_publish_token` option instead.

The API token value used to authenticate with the Confluence instance. Set this option to an API token for the configured username value (see [API tokens](#)):

```
confluence_api_token = 'YDYDD3qVvKV0FbkErSxaQ2olmy...AMGwaPe8=02381T9A'
```

Added in version 2.6.

confluence_api_token_scoped

Tip

Use this option for Confluence Cloud when using an API scoped token.

Enable this option to automatically convert a Confluence Cloud Wiki URL to a Confluence Cloud API endpoint URL. When using an API scoped token, Confluence expects an alternative API endpoint for calls. For example,

```
https://example.atlassian.net/wiki/
```

The following is needed:

```
https://api.atlassian.com/ex/confluence/550e8400-e29b-41d4-a716-446655440000/
```

The GUID for a space can be determined by venturing to:

```
https://example.atlassian.net/_edge/tenant_info
```

Users may opt to configuring `confluence_server_url` to the newer API endpoint for scoped tokens. However, enabling this option can perform the translation for users who prefer to configure the Wiki endpoint. By default, this option is not enabled with a value of `False`.

```
confluence_api_token_scoped = True
```

Added in version 2.15.

confluence_publish_token **Tip**

Use this option for Confluence Data Center.

 **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'
```

Added in version 1.8.

confluence_server_pass

Warning

It is not recommended to use this option when authenticating with an API token or a personal access token.

Note

Functionally, this option is the same as `confluence_api_token`. It is recommended to use the API token variant solely for naming convention. Only limited cases can use a password value for publication over API tokens or personal access tokens (specifically, cases using Confluence Data Center). Unless users are expected to interact directly with their Confluence instance with user passwords, users should instead use either one of the following options instead:

- `confluence_api_token`
- `confluence_publish_token`

Caution

It is never recommended to store a raw password into a committed/shared repository holding documentation. If desired, this extension provides a method for prompting for a password (see `confluence_ask_password`).

Future versions *may* deprecate this option.

The password value used to authenticate with the Confluence instance. This value expects the plaintext password for the configured username value:

```
confluence_server_pass = 'myawesomepassword'
```

confluence_cloud

Tip

Users only need to explicitly set this if using a Confluence Cloud instance hosted with a custom domain name.

A boolean value to whether or not to treat the configured `confluence_server_url` configuration as an Atlassian Cloud site. Cloud detection is implicitly performed to help determine which APIs and default editor to use. However, some environments may be utilizing custom domains that point to Atlassian Cloud services which can make it non-trivial to determine if the site is Cloud-based or not. This option can be used to explicitly configure if the site being published on is a Cloud-based instance. By default, automatic cloud detection occurs with a value of `None`.

```
confluence_cloud = True
```

See also `confluence_editor`.

Added in version 3.0.

5.3 Generic configuration

`confluence_add_secnumbers`

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`.

Added in version 1.2.

`confluence_code_block_theme`

Note

This option is only supported on Confluence Data Center.

Specifies the color scheme to use when displaying a Confluence code block macro.

```
confluence_code_block_theme = 'Midnight'
```

For configuring the theme on individual code blocks, see [class hints](#).

Added in version 2.2.

`confluence_default_alignment`

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'
```

Added in version 1.3.

`confluence_default_table_width`

Configure the default width to apply for all tables created. Accepts a width value where units are accepted (e.g. `100px`) or unitless to assume a pixel width. Percentages can be provided but experience may vary between Confluence editor versions.

```
confluence_default_table_width = 2000
(or)
confluence_default_table_width = '100em'
```

See also the [Confluence table width directive](#).

Added in version 2.15.

`confluence_disable_env_conf`

A boolean value to configure whether to ignore environment-provided configuration options. This extension will fallback on environment variables if an option is not set in a configuration file or on the command line. If a user never wants to pull options from the environment, this option can be set to `True`.

```
confluence_disable_env_conf = True
```

Added in version 2.7.

confluence_domain_indices

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',
]
```

Added in version 1.7.

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

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](#).

Added in version 1.9.

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

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](#).

Added in version 1.9.

confluence_include_search

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
```

Added in version 1.7.

confluence_page_generation_notice

This option can be set with a boolean value to whether or not to generate a message at the top of each document that the page has been automatically generated.

```
confluence_page_generation_notice = True
```

Alternatively, users may set a custom message to display.

```
confluence_page_generation_notice = 'My awesome message.'
```

By default, this notice is disabled with a value of `False`.

Added in version 1.7.

Changed in version 2.5: Accept a string for custom notice.

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 enabled with a value of `True`.

```
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.

Changed in version 2.0: Option is enabled by default.

confluence_prev_next_buttons_location

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'
```

Added in version 1.2.

confluence_secnumber_suffix

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

```
confluence_secnumber_suffix = '. '
```

See also [confluence_add_secnumbers](#).

Added in version 1.2.

confluence_sourcelink

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`
- `codeberg`
- `github`
- `gitlab`

Options to set for these types are as follows:

Option	Description
<code>owner</code>	The owner (group or user) of a project. <i>(required)</i>
<code>repo</code>	The name of the repository. <i>(required)</i>
<code>container</code>	The folder inside the repository which is holding the documentation. This will vary per project, for example, this may be <code>Documentation/</code> or <code>doc/</code> . If the documentation resides in the root of the repository, this option can be omitted or set to an empty string.
<code>version</code>	The version of the sources to list. This is typically set to either a branch (e.g. <code>main</code>) or tag value. For Codeberg, also include the version type. For example, <code>branch/main</code> or <code>tag/1.0</code> . <i>(required)</i>
<code>view</code>	The view mode to configure. By default, this value is set to <code>blob</code> for GitHub/GitLab and <code>view</code> for Bitbucket. GitHub/GitLab users may wish to change this to <code>edit</code> to create a link directly to the editing view for a specific document.
<code>host</code>	The hostname value to override. This option is useful for instances where a custom domain may be configured for an organization.
<code>protocol</code>	The protocol value to override (defaults to <code>https</code>).

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',
}
```

Added in version 1.7.

confluence_use_index

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
```

Added in version 1.7.

singleconfluence_toctree

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
```

Added in version 1.7.

5.4 Publishing configuration

confluence_append_labels

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](#)

Added in version 1.3.

confluence_api_mode

Configures the API mode to use for REST requests. Certain Confluence instances support a newer version of REST APIs (e.g. Confluence Cloud). This extension will attempt to use an appropriate API mode for a configuration set. However, a user can override the operating API mode based on preference or when handling situations where this extension cannot automatically determine the best API mode to use. Values accepted are either `v1` or `v2`.

```
confluence_api_mode = 'v2'
```

By default, if a Confluence Cloud configuration is detected, this extension will use `v2`. For all other cases, the default is `v1`.

Added in version 2.5.

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

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
```

Added in version 1.2.

confluence_cleanup_archive**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 Data Center with this feature will most likely result in an “Unsupported Confluence API call” error (500).

Attention

Confluence’s archiving API is marked as experimental by Atlassian 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_cleanup_search_mode`
- `confluence_publish_dryrun`

Added in version 1.9.

confluence_cleanup_from_root

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`

Added in version 1.9.

confluence_cleanup_purge

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 publish in the request. For example, if an original request publishes ten documents and purges excess documents, a following publish attempt with only one of the documents will purge the other nine pages.

See also:

- `confluence_cleanup_archive`
- `confluence_cleanup_from_root`
- `confluence_cleanup_search_mode`
- `confluence_publish_dryrun`

Added in version 1.9.

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 disabled with a value of `True`.

```
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](#).

Changed in version 2.6: Option is enabled by default.

confluence_full_width

A boolean value to whether to publish pages using the full width of a page. By default, page widths will use their default/existing page widths with a value of `None`. Specifying this option to `True` will ensure any new/updated page will attempt to use the full width of a page; likewise, specifying this option to `False` will ensure any new/updated page will attempt to use a smaller width.

```
confluence_full_width = True
```

For per-document overrides, please see the [confluence_metadata](#).

Added in version 2.0.

Changed in version 2.1: Support added for Confluence's v1 editor.

confluence_global_labels

Note

If removing global labels for a documentation set that already has been published, user may need to publish once with the [confluence_publish_force](#) option to help clear old labels.

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 see the [confluence_metadata](#). See also [confluence_append_labels](#).

Added in version 1.3.

confluence_parent_page

Note

This option cannot be used with [confluence_publish_root](#).

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`.

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

`confluence_publish_dryrun`

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

```
confluence_publish_dryrun = True
```

See also [Confluence Spaces and Unique Page Names](#).

Added in version 1.3.

`confluence_publish_postfix`

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 `docname` (name and location of the document source based from the project root) and the value of `confluence_publish_postfix_hash_modifier`. 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_postfix_hash_modifier`

Added in version 1.2.

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

confluence_publish_prefix

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

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

An example publish prefix is as follows:

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

By default, no prefix is used. See also:

- `confluence_ignore_titlefix_on_index`
- `confluence_publish_postfix`

confluence_publish_root

Note

This option cannot be used with `confluence_parent_page`.

The page identifier to publish the root document to. The root identifier value is used to find an existing page on the configured Confluence instance. When found, the root document of the documentation set being published will replace the content of the page found on the Confluence instance. If the root page cannot be found, the publish attempt will stop with an error message.

```
confluence_publish_root = 123456
```

See also `confluence_parent_page`.

Added in version 1.5.

confluence_root_homepage

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

```
confluence_root_homepage = False
```

Added in version 1.6.

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, it is 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

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](#).

Added in version 1.3.

5.5 Advanced publishing configuration

confluence_additional_mime_types

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',
]
```

Added in version 1.3.

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`

Changed in version 2.3: Support relative paths.

confluence_cleanup_search_mode

Warning

The `direct` search mode may not work on Confluence Data Center instances. For these cases, Confluence may report the following error:

(Not Implemented; 500)

Page children is currently only supported for direct children.

Configures the search mode used for finding descendant pages to be cleaned up (when configured for archiving/purging legacy pages). By default, this extension will search Confluence for known descendants for the root page:

```
confluence_cleanup_search_mode = 'search'
```

However, in some cases, the provided list of descendants may be incorrect (due to the Confluence version used, the state of Confluence's ancestors table caching, etc.). This configuration can be used to tweak how this extension searches for descendants, if a user experiences issues with the default method of searching. Supported modes are as follows:

- **direct**: Query known descendants from a page's cache.
- **search** (*default*): Search for descendants using Confluence's CQL capability.

Users can also postfix `-aggressive` (e.g. `search-aggressive`) on a mode to perform a recursive search for descendants ensure all descendants are found. Note that an aggressive search will increase the amount of API calls to a configured Confluence instance. See also:

- [confluence_cleanup_archive](#)
- [confluence_cleanup_purge](#)

Added in version 2.1.

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_autogen_title

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

```
confluence_disable_autogen_title = True
```

See also:

- `confluence_remove_title`
- `confluence_title_overrides`

confluence_disable_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

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`

Added in version 1.3.

confluence_page_search_mode

Note

This option is only supported on Confluence Data Center.

Configures the mode which pages will be fetched from Confluence. For Confluence Data Center instances, there may be performance issues when attempting to query `content/` API ([CONFSERVER-57639](#)). Select environments may opt to disable this endpoint in attempt to avoid performance issues, which in turn prevents this extension from fetching page content. To support these environments, users can configure this extension to use an alternative mode for fetching page content.

```
confluence_page_search_mode = 'search'
```

Supported modes are as follows:

- `content` (*default*): Pages will be fetched using the `content/` API.
- `search`: Pages will be fetched using the `content/search/` API.

Added in version 2.6.

confluence_parent_override_transform**Note**

Using this option may have unexpected results when using certain features of this extension. For example, users with purging enabled may not have pages with parent-ID overrides purged.

A function to override the parent page to publish a document under. This option is available for advanced users needing to tailor specific parent pages for individual documents. A provided transform is invoked with the document name and the expected parent page (numerical identifier) the document will be published under. A configuration can tweak the identifier used when publishing.

```
def my_publish_override(docname, parent_id):
    if docname == 'special-doc':
        return 123456

    return parent_id

confluence_parent_override_transform = my_publish_override
```

This extension will not check the validity of the identifiers set. If a provided page identifier does not exist or the publishing user does not have access to the parent page, the publication will fail with an error provided by Confluence.

See also [confluence_parent_page](#).

Added in version 2.2.

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

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`.

Added in version 1.3.

Changed in version 2.0: An empty allow list will no longer publish any documents.

Changed in version 2.3: Support relative paths.

confluence_publish_debug**Warning**

Enabling certain debugging options may reveal information such as authentication details in printed logs. Take this into consideration when sharing any debug logs with other users or enabling this option when part of a CI/CD script, job or workflow.

Configures the ability to enable certain debugging messages for requests made to a Confluence instance. This can be helpful for users attempting to debug their connection to a Confluence instance. By default, no debugging is enabled.

Available options are as follows:

- **all**: Enable all debugging options.
- **deprecated**: Log warnings when a deprecated API call is used (*for development purposes*).
- **headers**: Log requests and responses, including their headers.
- **headers-and-data**: Log header data along with request/response bodies.
- **urllib3**: Enable urllib3 library debugging messages.

An example debugging configuration is as follows:

```
confluence_publish_debug = 'headers'
```

Added in version 1.8.

Changed in version 2.5: Switched from boolean to string for setting new debugging options.

Changed in version 2.6: Introduce the **headers-and-data** option.

confluence_publish_delay

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
```

Added in version 1.8.

confluence_publish_denylist

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](#).

Added in version 1.3.

Changed in version 2.3: Support relative paths.

confluence_publish_force

A boolean value on whether or not to force publish page updates even if no changes are detected on the Confluence instance. When a page is published by this extension, a hash of the page will be stored on the Confluence page. This hash can be referred to later by hosts using this extension, by querying the hash and comparing it against a locally prepared page update. If hashes match, no attempt will be made to update the specific page. If users are experiencing issues with this check, they may force publishing by configuring this option to `True`. By default, this option is disabled with a value of `False`.

```
confluence_publish_force = True
```

See also [confluence_asset_override](#).

Added in version 2.1.

confluence_publish_headers

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

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

Added in version 1.5.

confluence_publish_intersphinx

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
```

Added in version 1.9.

confluence_publish_onlynew

Tip

Users considering this option may wish to use [confluence_publish_trample](#) instead.

A publish event 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
```

See also [confluence_publish_trample](#).

Added in version 1.3.

confluence_publish_orphan

Whether to permit the publishing of orphan pages to a Confluence space. This option must be explicitly set to `False` if a user wishes to not publish orphan pages for their documentation. By default, the value is set to `True`.

```
confluence_publish_orphan = True
```

See also [confluence_publish_orphan_container](#).

Added in version 2.1.

confluence_publish_orphan_container

The identifier of the page to hold orphan pages. The parent page associated to an orphan page can vary per configuration. When a user configures for a parent page/root, orphan pages will be placed under the respective parent page/root configuration. If no parent page/root is configured, orphan pages will not be associated with a parent page.

Users can override where orphan pages are placed by using this option. By specifying a page identifier, orphan pages will be placed under the configured container page. Users can also provide a special value of `0` to indicate to always publish with no parent page.

```
confluence_publish_orphan_container = 123456
```

See also [confluence_publish_orphan](#).

Added in version 2.1.

confluence_publish_postfix_hash_modifier

The hash for [confluence_publish_postfix](#) is computed from the docname (name and location of the document source based from the project root) and the value given here. The default value is:

```
conf.confluence_publish_postfix_hash_modifier = (
    str(conf.project)
    + str(conf.confluence_parent_page)
```

(continues on next page)

(continued from previous page)

```
+ str(conf.confluence_publish_root)
)
```

Modify this option if there is a need to generated hashes that are independent of:

- `project`
- `confluence_parent_page`
- `confluence_publish_root`

This can be helpful when a project name or the title of the parent page have changed but there is a desire to retain the hash values. Thereby the page contents may be updated without breaking links from other Confluence pages.

See also:

- `confluence_publish_postfix`

Added in version 3.0.

confluence_publish_override_api_prefix

Allows a user to override the path-prefix value used for API requests. API paths are commonly prefixed, such as `rest/api/` for API v1 and `api/v2/` for API v2. However, if a user is interacting with a Confluence instance which system administrators have configured non-standard locations for API endpoints, requests made by this extension will fail.

To support custom API endpoint paths, this option can be used to indicate what prefix to use, if any. By default, this extension operates with an API prefix configuration matching the following:

```
confluence_publish_override_api_prefix = {
    'v1': 'rest/api/',
    'v2': 'api/v2/',
}
```

Users may define a dictionary using `confluence_api_mode` values for keys, followed by a prefix override for their environment. For example, to disable prefixes for any API v1 request, the following may be used:

```
confluence_publish_override_api_prefix = {
    'v1': '',
}
```

Added in version 2.5.

confluence_publish_retry_attempts

Allows a user to override how many retry attempts are permitted when a single API request fails. By default, this extension uses a maximum retry of two, allowing a single request to be attempted three times for select failure events.

Failure scenarios that are retried on include all 500-series errors, as well as couple of observed/reported corner cases reported by Confluence instances during the life-cycle of this extension.

```
confluence_publish_retry_attempts = 2
```

See also [confluence_publish_retry_duration](#).

Added in version 2.10.

confluence_publish_retry_duration

The duration (in seconds) to wait between API retry events on select failures. By default, the duration waited is four seconds.

```
confluence_publish_retry_duration = 4
```

See also [confluence_publish_retry_attempts](#).

Added in version 2.10.

confluence_publish_skip_commented_pages

Note

There is no support to keep inlined comments on updated pages at this time.

Indicates to skip updates on pages which have inlined comments. By default, pages will always be updated with a value of `False`.

If this option is set to `True`, if a page to be update is detected to have an embedded comment reference, a warning will be generated and the page will not be updated.

```
confluence_publish_skip_commented_pages = True
```

If this option is set to `ignored-resolved`, the page will be checked for embedded comment references. In the event that comments are found and all comments are resolved, the page will be updated/overridden as if no comments were there. However, if any unresolved comments are detected, a warning will be generated and the page will not be updated.

```
confluence_publish_skip_commented_pages = 'ignored-resolved'
```

Users may be interested in turning [warnings into errors](#) to report a non-zero exit status for these warnings.

See also [suppress_warnings_config](#).

Added in version 2.13.

Changed in version 3.1: Support the `ignored-resolved` value.

confluence_publish_trample

Since [page names are unique for a given space](#), publishers need to take into account page name conflicts for documents that already exist on a space versus documents attempted to be published. By default, this extension will attempt to detect and stop a publish event if it has detected a page being updated/moved that is not part of the published documentation's hierarchy. This default is enforced with a value of `False`.

If a publisher wishes to always force renaming/replacing pages that are found outside the hierarchy, this option can be set to `True`.

```
confluence_publish_trample = True
```

See also:

- [Confluence Spaces and Unique Page Names](#)

- `confluence_publish_onlynew`

Added in version 2.15.

confluence_request_session_override

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
```

Added in version 1.7.

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 see [Requests – Authentication](#). By default, no custom authentication handler is provided to generated REST API requests. An example OAuth 1 is as follows:

```
from requests_oauthlib import OAuth1

...

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

confluence_server_cookies

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

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

Added in version 1.2.

confluence_title_overrides

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`

Added in version 1.3.

confluence_version_comment

Note

Confluence Data Center does not support setting a version comment for the first/new page revision.

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

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

Added in version 1.8.

Changed in version 2.1: Support comments for first/new pages on Confluence Cloud.

5.6 Advanced processing configuration

confluence_file_suffix

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

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

confluence_html_macro

This option will configure the HTML macro type for the `confluence_html` directive. By default, the `html` macro identifier is set.

```
confluence_html_macro = 'html'
```

See also the [HTML directive](#).

Added in version 2.7.

confluence_jira_servers

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](#)

Added in version 1.2.

confluence_lang_overrides

A dictionary to override literal block-based directive language values to a Confluence supported code block macro language values. The default mapping accepts [Pygments documented language types](#) to [Confluence-supported syntax highlight languages](#).

```
confluence_lang_overrides = {
    'rs': 'rust',
    'rust': 'rust',
}
```

In the event that a language entry is missing or returns a `None` value, the provided language type will be transform to a default language type as if this transform was not provided.

Added in version 2.6.

confluence_latex_macro**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, see 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:

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

Added in version 1.8.

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'
```

confluence_mathjax

Important

This option relies on additional configuration of a Confluence instance or additional directives such as using an HTML macro. Both of these approaches may not be available in a default Confluence configuration. Confluence Cloud may require a custom marketplace add-on. Confluence Data Center can require a system administrator configuring site support for MathJax; or environments that have a Confluence HTML macro enabled, users can attempt to include MathJax library support via the use of a [confluence_html](#) directive.

While this extension aims to support the capability of generating raw content that MathJax could render, the complete solution of using MathJax on a Confluence instance is considered unsupported.

Generate math content into a raw format that can be rendered on a Confluence instance that has been configured to support [MathJax](#).

```
confluence_mathjax = True
```

See also:

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

Added in version 2.10.

confluence_manifest_data

A manifest file (`scb-manifest.json`) is generated after each run into the output directory. This information includes built pages as well as attachments for these pages. Each page/attachment provides a path to where the content resides. However, if a user wishes to include this data into the manifest, this option can be used to Base64-encode page/attachment data into the manifest. By default, this is disabled:

```
confluence_manifest_data = True
```

Added in version 2.10.

confluence_mentions

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](#)

Added in version 1.9.

`confluence_navdocs_transform`

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`.

Added in version 1.7.

`confluence_permit_raw_html`

Caution

Using this option is considered unsupported. This extension will allow users to directly publish HTML content defined in a document, but there is no guarantees that the content will render as expected, or even be able to be published to a configured Confluence instance.

Configure whether to permit the use of raw HTML content in generated documents. While Confluence renders pages through a website, content is stored using a “storage” format, which only supports a subset of HTML. Confluence may filter out or reject the publication of pages with certain HTML content.

Some documentation may rely on HTML-specific content, and if this HTML content is not too complex, this may be renderable on a Confluence instance. Users wanting to allow this can enable this option to have HTML content directly injected on pages:

```
confluence_permit_raw_html = True
```

Or even placed inside an HTML-supported macro (if such a macro is available for the target Confluence instance):

```
confluence_permit_raw_html = 'html'
```

Using this option is not supported. Content may be automatically stripped when published into Confluence, content may not render as expected (e.g. styles can be ignored, JavaScript will not function) or Confluence may reject the publication of Confluence document (i.e. failing to upload a page).

See also `confluence_html` directive.

Added in version 2.2.

`confluence_tab_macro`

⚠ Attention

This feature is considered experimental. Inlined tabs are supported through third-party Sphinx extensions as well as only supported on Confluence instances using third-party marketplace extensions. The combination is less than ideal for providing consistent results and performing testing. This may be used by advanced users who can take advantage of their instance's configurations to utilize inlined tabs.

This configuration is used when attempting to build macros for rendering inlined tabs on a page. It is used to define what third-party macro can be used to render inlined tabs. This configuration also defines what parameter values dictate the name of a tab and well as which tab is considered to be the first/primary tab for a set.

```
confluence_tab_macro = {
    'macro-name': 'macro-name',
    'primary-id': 'id-for-primary-parameter',
    'primary-value': 'value-to-primary-parameter',
    'title-id': 'id-for-title-parameter',
}
```

The following configurations are known to function for the listed marketplace extensions:

Marketplace Application	Configuration
Mosaic: Content Formatting Macros & Templates for Confluence	<pre>confluence_tab_macro = { 'macro-name': 'cfm-tabs-page', 'primary-id': 'primaryTab', 'primary-value': 'true', 'title-id': 'tabsPageTitle', }</pre>

Added in version 2.14.

`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`

5.7 Third-party related options

Note

The configurations in this section are specific to supporting [third-party extensions](#); results may vary.

`confluence_mermaid_html_macro`

Warning

This option relies on an HTML macro which is not available in a default Confluence configuration. Using this option is only useful for users that have instances where a system administrator has enabled their use.

Note

This option will most likely require additional configuration to function. Setting this option only produces HTML macros with Mermaid content but does not automatically include the JavaScript required to process this content.

There can be various ways an instance/page can be configured to include Mermaid JS support. For example, adding the following content on the page planning to render diagrams:

```
.. confluence_html::  
  
   <script type="module">  
     import mermaid from 'https://cdn.jsdelivr.net/npm/mermaid@11/dist/mermaid.esm.min.mjs';  
   </script>
```

When using the `sphinxcontrib-mermaid` extension, this option can be used pass raw Mermaid figures into an HTML macro.

```
confluence_mermaid_html_macro = True
```

See also `confluence_html_macro`.

Added in version 2.7.

5.8 Other options

suppress_warnings

This extension supports suppressing warnings using Sphinx's `suppress_warnings` configuration. The following includes additional warning types that may be suppressed:

- `confluence` – All warnings
- `confluence.deprecated` – Configuration deprecated warnings
- `confluence.deprecated_develop` – Development deprecated warnings
- `confluence.inline-comment` – Inlined comment warnings
- `confluence.unsupported_code_lang` – Unsupported code language

For example:

```
suppress_warnings = [  
    'confluence.unsupported_code_lang',  
]
```

Added in version 2.1.

5.9 Deprecated options

`confluence_asset_force_standalone`

This option has been dropped as the shared assets feature is no longer supported. As of v2.15, a page will always host its own assets.

Added in version 1.3.

Deprecated since version 2.15.

Removed in version 3.0.

`confluence_editor`

Important

- This option should not be used for Confluence Data Center.
- This option is a deprecated hint at what editor style to use on Confluence Cloud. In the year 2026, Atlassian [deprecates their older editor](#). This extension as of its v3 release will use the v1 editor for Confluence Data Center and the v2 editor for Confluence Cloud. Users may attempt to use this option to tweak their output, but results may vary. It is expected over time that this option will do nothing when Atlassian finalizes its editor support across its platforms.
- If a page is published with a v2 editor, an attempt to re-publish with a v1 editor style may be ignored in Confluence Cloud. In such situations, users are recommended to delete the pages on Confluence and then retry the publish attempt with this extension.

A string value to indicate which [Confluence editor](#) to target. The following editor values are supported:

- v1: Use Confluence’s older editor (default on Data Center).
- v2: Use Confluence’s newer editor “fabric” (default on Cloud).

A user can choose which version of the editor to build and published documentation with. This extension may adjust how content is generated based on which editor is selected.

```
confluence_editor = 'v1'
```

See also [confluence_cloud](#).

Added in version 2.0.

Changed in version 3.0: Defaults to v2 for Confluence Cloud.

Deprecated since version 3.0.

`confluence_lang_transform`

This option has been replaced by [confluence_lang_overrides](#).

Deprecated since version 2.6.

Removed in version 3.0.

`confluence_master_homepage`

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

Deprecated since version 1.6.

Removed in version 3.0.

confluence_parent_page_id_check

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`.

Deprecated since version 1.9.

Removed in version 3.0.

confluence_publish_disable_api_prefix

This option has been replaced by `confluence_publish_override_api_prefix`.

Deprecated since version 2.5.

Removed in version 3.0.

confluence_publish_subset

This option has been renamed to `confluence_publish_allowlist`.

Deprecated since version 1.3.

Removed in version 3.0.

confluence_purge_from_master

This option has been renamed to `confluence_purge_from_root`, and has since been replaced with `confluence_cleanup_from_root`.

Deprecated since version 1.6.

Removed in version 3.0.

confluence_purge_from_root

This option has been renamed to `confluence_cleanup_from_root`.

Deprecated since version 1.9.

Removed in version 3.0.

confluence_space_name

This option has been renamed to `confluence_space_key`.

Deprecated since version 1.7.

Removed in version 3.0.

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 -M confluence . _build -E -a
```

singleconfluence

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 -M singleconfluence . _build -E -a
```

Added in version 1.3.

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
```

Alternatively, users may be interested in a third-party extension which can help support the use of Confluence directives in other builders:

sphinx-confluencebuilder-bridge

<https://github.com/adamtheturtle/sphinx-confluencebuilder-bridge>

7.1 Common

.. confluence_excerpt::

The `confluence_excerpt` directive allows a user to define a Confluence [Excerpt Macro](#) to help build snippets of content to be shared for other pages. For example:

```
.. confluence_excerpt::

    This content is reusable.
```

This directive supports the following options:

:atlassian-macro-output-type: output type (block, inline)

When this option is configured to `inline` (default), the rendered excerpt macro will be presented inlined with other content in the page (no additional line breaks). The `block` type can be used to help render an excerpt macro as a block-level element.

```
.. confluence_excerpt::
    :atlassian-macro-output-type: block

    This content is reusable.
```

:hidden: flag (boolean)

Whether the macro's content should be rendered on the page that defines it. Valid values are `true` or `false` (default).

```
.. confluence_excerpt::
    :hidden: true

    This content is reusable.
```

:name: value (string)

A name value to apply to the macros "name" field. This can be used to name a specific excerpt on a page, which can be explicitly mapped to with an `excerpt-include` when attempting to define multiple excerpts on the same page. If only a single excerpt is created, an `excerpt-include` will implicitly use the first excerpt on a page.

```
.. confluence_excerpt::
    :name: my-excerpt

    This content is reusable.
```

Added in version 2.0.

.. confluence_excerpt_include:: [ref]

The `confluence_excerpt_include` directive allows a user to define a Confluence [Excerpt Include Macro](#) to help include snippets of content provided by excerpt macro definitions. An include macro requires an explicit reference to the page which holds the excerpt content to load. Users can specify an exclamation-prefixed document name, referring to a local Sphinx documentation file to load an excerpt from. For example:

```
.. confluence_excerpt_include:: !my-excerpt-docname
```

Users may also specify a known document title name that may be managed outside their Sphinx project set. For example:

```
.. confluence_excerpt_include:: Custom excerpt
```

To target a page on a different space, the space name can be prefixed before the document title:

```
.. confluence_excerpt_include:: MYSPACE:Custom excerpt
```

This directive supports the following options:

:name: value (string)

The explicit name of the excerpt to use for a given page. If no name is provided, the excerpt-include macro will use the first available excerpt on the target page.

```
.. confluence_excerpt_include:: !my-excerpt-docname
   :name: my-excerpt
```

:npanel: flag (boolean)

Whether the macro's content should remove the panel around the excerpted content. Valid values are `true` or `false` (default).

```
.. confluence_excerpt_include:: !my-excerpt-docname
   :npanel: true
```

Added in version 2.0.

.. confluence_expand::

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.
```

See also [Collapsed content](#).

Added in version 1.3.

.. `confluence_html::`

Warning

The `HTML Macro` is disabled by default on Confluence instances. Using this directive is only useful for users that have instances where a system administrator has enabled their use.

The `confluence_html` directive allows a user to define a Confluence `HTML Macro` to render HTML content on a page. For example:

```
.. confluence_html::

    <h1>Header</h1>

    This is an <strong>example</strong>.
```

See also `confluence_permit_raw_html`.

Added in version 2.7.

.. `confluence_metadata::`

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

:editor: value (v1, v2)

Deprecated since version 2.14: Support for page-specific editor overrides will be removed in 2026 as Confluence Cloud removes support for the v1 editor.

Pages are published with the editor type configured through the `confluence_editor` option. However, users can override the editor for a specific page using the `editor` metadata option.

```
.. confluence_metadata::
    :editor: v2
```

See also `confluence_editor`.

:full-width: flag (boolean)

Pages are published with the full-width appearance configured through the `confluence_full_width` option. However, users can override the appearance for a specific page using the `full-width` metadata option.

```
.. confluence_metadata::
    :full-width: false
```

See also `confluence_full_width`.

:guid: value (string)

Assign a user-managed GUID value for a page. This value can be shared in a subset of `events` generated by this extension.

: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](#).

Added in version 1.3.

Changed in version 2.2: Added editor and full-width support.

Changed in version 2.8: Added guid support.

.. confluence_newline::

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

```
.. confluence_newline::
```

Added in version 1.7.

.. confluence_panel::

The `confluence_panel` directive allows a user to define a Confluence [Panel Macro](#) to help format content in a customizable colored panel on a page. For example:

```
.. confluence_panel::
```

This content is captured inside the panel macro.

This directive supports the following options:

:bg-color: value (string)

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

```
.. confluence_panel::
   :bg-color: #00ff00
```

This content is captured inside the panel macro.

:border-color: value (string)

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

```
.. confluence_panel::
   :border-color: #ff0000
```

This content is captured inside the panel macro.

:border-style: value (string)

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

```
.. confluence_panel::
   :border-style: solid
```

This content is captured inside the panel macro.

:border-width: value (string)

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

```
.. confluence_panel::
   :border-width: 42
```

This content is captured inside the panel macro.

:title: value (string)

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

```
.. confluence_panel::
   :title: This is my title
```

This content is captured inside the panel macro.

:title-bg-color: value (string)

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

```
.. confluence_panel::
   :title: my title
   :title-bg-color: #0000ff
```

This content is captured inside the panel macro.

:title-color: value (string)

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

```
.. confluence_panel::
   :title: my title
   :title-color: #ff00ff
```

This content is captured inside the panel macro.

Added in version 2.13.

.. confluence_table_width::

The `confluence_table_width` directive allows a user to define widths for tables defined after the directive. For example:

```
.. confluence_table_width:: 800px
```

```
+-----+-----+-----+-----+
| Header row, column 1 | Header 2 | Header 3 | Header 4 |
| (header rows optional) |         |         |         |
+-----+-----+-----+-----+
| body row 1, column 1 | column 2 | column 3 | column 4 |
+-----+-----+-----+-----+
| body row 2           | ...     | ...     |         |
+-----+-----+-----+-----+
```

To reset the table width for additional tables, a value `unset` can be provided.

```
.. confluence_table_width:: 800px
```

Product	Price	Availability
Widget	\$10	In stock
Gadget	\$15	Out of stock

```
.. confluence_table_width:: unset
```

Name	Age	City
Alice	30	New York
Bob	25	London

See also the `confluence_default_table_width` option.

Added in version 2.15.

.. `confluence_toc::`

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
```

Added in version 1.9.

.. confluence_viewpdf::

The `confluence_viewpdf` directive allows a user to define a Confluence [PDF Macro](#) to help inline PDF files in a page. For example:

```
.. confluence_viewpdf:: example.pdf
```

Added in version 2.14.

7.2 Jira

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

.. jira:: [jql]

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`. 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
```

Added in version 1.2.

```
.. jira_issue:: [issue-id]
```

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`. 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
```

Added in version 1.2.

See also [Jira roles](#).

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` 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::`

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!
```

Added in version 1.8.

See also [LaTeX roles](#).

7.4 Smart links

Note

Smart links will only render when using Confluence Cloud.

.. `confluence_doc::`

The `confluence_doc` directive allows a user to define a link to a document that is styled with a card appearance. The directive accepts the name of a document in an absolute or relative fashion (in the same manner as Sphinx's `:doc:` role). For example:

```
.. confluence_doc:: my-other-page
```

This directive supports the following options:

`:card:` card type (`block`, `embed`)

When using `block`, a smart link card is generated for the document link. The card will provide a summary of the target document. When using `embed`, the contents of the document will rendered on the page.

```
.. confluence_doc:: my-other-page
   :card: block
```

`:layout:` layout type (`align-start`, `align-end`, `center`, `wrap-left`, `wrap-right`)

Note

This option is only applicable when using an embed card.

Specifies how an embedded card will be laid out on a page. Embedded cards will default to 100% width. Therefore, to take advantage of certain layout capabilities, users should also assign an appropriate width as well.

```
.. confluence_doc:: my-other-page
   :card: embed
   :layout: align-end
   :width: 20
```

`:width:` value (integer)

Note

This option is only applicable when using an embed card.

Specifies the width to apply for an embedded card. The width is a value from 0 to 100 (e.g. a value of 80 for 80% of the page).

```
.. confluence_doc:: my-other-page
   :card: embed
   :width: 50
```

Added in version 2.1.

.. confluence_link::

The `confluence_link` directive allows a user to define a link to a page that is styled with a card appearance. The directive accepts a URL. How Confluence renders the context of a link card will vary based on which link targets Confluence supports. For example:

```
.. confluence_link:: https://example.com
```

This directive supports the following options:

:card: card type (block, embed)

When using `block`, a smart link card is generated for the link. The card will provide a summary of the target link. When using `embed`, the contents of the link will rendered on the page.

```
.. confluence_link:: https://example.com
   :card: block
```

:layout: layout type (align-start, align-end, center, wrap-left, wrap-right)**Note**

This option is only applicable when using an embed card.

Specifies how an embedded card will be laid out on a page. Embedded cards will default to 100% width. Therefore, to take advantage of certain layout capabilities, users should also assign an appropriate width as well.

```
.. confluence_link:: https://example.com
   :card: embed
   :layout: align-end
   :width: 20
```

:width: value (integer)**Note**

This option is only applicable when using an embed card.

Specifies the width to apply for an embedded card. The width is a value from 0 to 100 (e.g. a value of 80 for 80% of the page).

```
.. confluence_link:: https://example.com
   :card: embed
   :width: 50
```

Added in version 2.1.

See also [smart link roles](#).

ROLES

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

8.1 Emoticon Macro

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

:confluence_emoticon:

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.
```

Added in version 1.9.

8.2 Jira

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

:jira:

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.
```

Added in version 1.7.

See also [Jira directives](#).

8.3 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` 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:`

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

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

Added in version 1.8.

See also [LaTeX directives](#).

8.4 Mentions

The following role can be used to help include [Confluence mentions](#) into generated Confluence documents.

:confluence_mention:

Warning

Confluence Cloud mentions should always use account identifiers; where Confluence Data Center mentions should use either usernames or user keys. Attempting to use Confluence Cloud account identifiers when publishing to a Confluence Data Center 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 Data Center, 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` option.

Added in version 1.9.

8.5 Smart links

Note

Smart links will only render when using Confluence Cloud.

Support for inlined smart links can be created using the following roles.

:confluence_doc:

The `confluence_doc` role allows a user to define an inlined link to a document that is styled with a card appearance. The role accepts the name of a document in an absolute or relative fashion (in the same manner as Sphinx's `:doc:` role). For example:

```
See :confluence_doc:`my-other-page` .
```

Added in version 2.1.

:confluence_link:

The `confluence_link` role allows a user to define an inlined link to a page that is styled with a card appearance. The role accepts a URL. How Confluence renders the context of a link card will vary based on which link targets Confluence supports. For example:

```
See :confluence_link:`https://example.com` .
```

Added in version 2.1.

See also [smart link directives](#).

8.6 Status Macro

The following role can be used to help include [Confluence status macro](#) into generated Confluence documents.

:confluence_status:

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 (if supported by the configured Confluence editor), adjust the color enclosure to square brackets:

```
:confluence_status: `PASSED [green]`
```

Added in version 1.9.

8.7 Strikethrough

The following role can be used to explicitly define strikethrough text into generated Confluence documents.

:confluence_strike:

The `confluence_strike` role allows a user to build inlined text that has been styled with a strikethrough. For example:

```
:confluence_strike: `My text`
```

Added in version 2.1.

See also [Strike content](#).

EVENTS

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

9.1 Generic events

confluence-publish-attachment (**app**, **docname**, **key**, **aid**, **meta**)

Parameters

- **app** – Sphinx
- **docname** – str of the document name
- **key** – str of the attachment key
- **aid** – int of the upload id of the published attachment
- **meta** – dict of additional metadata for this event

Emitted when this extension has completed the upload of an attachment.

Added in version 2.8.

confluence-publish-page (**app**, **docname**, **pid**, **meta**)

Parameters

- **app** – Sphinx
- **docname** – str of the document name
- **pid** – int of the upload id of the published page
- **meta** – dict of additional metadata for this event

Emitted when this extension has completed the upload of a document.

Added in version 2.8.

confluence-publish-point (**app**, **point_url**)

Parameters

- **app** – Sphinx
- **point_url** – str of the publish point URL

Emitted when this extension prints the “publish point” URL to the standard output stream.

Added in version 2.6.

9.2 Advanced events

`confluence-publish-override-pageid` (`app`, `docname`, `meta`)

Parameters

- `app` – Sphinx
- `docname` – str of the document name
- `meta` – dict of additional metadata for this event

Returns

`int` | None of the new page identifier

Emitted when this extension is about to determine the page identifier to publish a document. A configuration and register this event and return a new identifier to use for a page. If `None` is returned, the extension will operate in the same manner as if no override was provided.

Added in version 2.8.

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.

10.1 reStructuredText markup

Type	Notes
admonitions	Supported
bibliographic fields	Supported
block quotes	Supported
bullet lists	Supported
citations	Supported
compound paragraph	Supported
container	Supported
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	Limited support. When using Confluence Cloud, images cannot be inlined.
inline markup	Supported
list-table	Supported
literal blocks	Supported
math	Supported with additional system tools. Requires a LaTeX and dvisvgm installation. When using Confluence Cloud, images cannot be offset to be aligned in a paragraph.
parsed literal block	Limited support. When using Confluence Cloud, literal blocks cannot be parsed and the contents will be placed inside a code macro.
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 mostly ignored as they are not supported in a Confluence format document)

10.2 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; Confluence v1 editor only). When the <code>class</code> attribute contains <code>collapse</code> , the macro will be configured to be collapsed.
deprecated	Supported
download	Supported
glossary	Supported
hlist	Limited support. When using Confluence Cloud, the maximum columns for an hlist is three.
manpage	Supported
production list	Supported
toctree	Supported
versionadded	Supported
versionchanged	Supported

10.3 Markdown support

This extension can support the rendering of Markdown content with the use of Sphinx with the [MyST Parser](#). Most content can be translated to an applicable Confluence storage format. However, not all content produced by MyST Parser will work with this extension. Specifically, any directives that generate HTML content or users who add raw HTML content to documents are not accepted by default by this extension. This includes line breaks, strikethroughs and more. Users may attempt to use the `confluence_permit_raw_html` option to help workaround select use cases, but the use of the option is unsupported. When the option is enabled, published content may not render as expected or may not be able to be published.

10.4 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
<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.
<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>	Supported (Sphinx v8.1+)
<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.

10.5 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 shows 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-data-viewer	Limited support. Can render JSON data inside code blocks. Content has collapse all data or expand all data, but not parts of the data.
sphinx-design	Limited support. Single card support (include header, title, body and footer). No side-by-side cards, alignment or extended capabilities such as images or clickable support. Dropdowns will function except for pre-expanded directives. Support for some badges but limited to select colors supported by Confluence. No extended support for tabs, button and inline icons.
sphinx-diagrams	Supported
sphinx-gallery	Supported
sphinx-inline-tabs	Limited support. Requires a Confluence instance that supports inlined tabs through the use of a third-party macro. <code>confluence_tab_macro</code> must be configured.
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-tabs	Limited support. Requires a Confluence instance that supports inlined tabs through the use of a third-party macro. <code>confluence_tab_macro</code> must be configured. Features such as group tabs are not supported. May require an explicit registration for support with sphinx-tabs:
	<pre>sphinx_tabs_valid_builders = ['confluence', 'singleconfluence',]</pre>
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. Raw/HTML renders can be used for environments with HTML macro support.
sphinxcontrib-nwdiag	Limited support. PNGs only; cannot configure for SVG at this time.
sphinxcontrib-openapi	Supported
sphinxcontrib-plantuml	Supported. sphinxcontrib-plantuml provides its own support for this extension.
sphinxcontrib-programoutput	Supported

continues on next page

Table 1 – continued from previous page

Type	Notes
sphinxcontrib-seqdiag	Limited support. PNGs only; cannot configure for SVG at this time.
sphinxcontrib-svgbob	Supported
sphinxcontrib-video	Supported
sphinxcontrib-youtube	Supported

10.6 Other

If a feature or extension is not listed above, is not working as expected or has another concern, feel free to bring up an issue:

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

11.1 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` or `confluence_publish_postfix` options.

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

11.2 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](#)).

11.3 Connection troubleshooting

The majority of connection issues reported are typically configuration issues. For example, attempting to configure a Confluence Cloud API token using a configuration designed for a Confluence Data Center Personal Access Token (PAT).

Users may try to invoke a connection test to help debug connection issues. To invoke a connection test, run the following command inside the Sphinx project:

```
$ python -m sphinxcontrib.confluencebuilder connection-test
Fetching configuration information...
Running Sphinx v7.2.6
...
Connecting to Confluence instance... connected!
Fetching Confluence instance information... fetched!
Decoding information... decoded!
Parsing information... parsed!
  Type: confluence
  Version: 1000.0.0-395b9ccce521
  Build: 6452
```

11.4 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 ]-----
```

GUIDES

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

12.1 Class hints

The following outlines all available class-related hints supported by the Confluence builder extension.

12.1.1 collapse

See [Collapsed content](#) for more details.

12.1.2 confluence-theme-<theme>

Added in version 2.2.

When this class hint is applied to a literal/code block, this provides the ability to override the theme configured for an individual code-block macro. For example:

```
.. code-block:: python
   :class: confluence-theme-FadeToGrey

   def main():
       print 'Hello, world!'

   if __name__ == '__main__':
       main()
```

See also [confluence_code_block_theme](#).

12.1.3 strike

See [Strike content](#) for more details.

12.2 Collapsed content

There are a series of collapsed element capabilities supported by the Confluence builder extension. This guide will cover the available options that can be used.

12.2.1 Expand directive

Added in version 1.3.

This extension provides a `confluence_expand` directive to help collapse content. For example:

```
.. confluence_expand::

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

This directive will only work when using the Confluence builder. See also the [Confluence expand directive](#).

12.2.2 Class hints

Added in version 2.1: Support added for containers.

Added in version 2.0: Support added for code blocks.

This extension provides support for managing collapsed elements when a `collapse` class hint is provided. If a container is defined with the style configured, the contents of the directive will be collapsed:

```
.. container:: collapse

    This paragraph should be collapsed.
```

When using the code block directive, providing a collapse style hint will also collapse the code block:

```
.. code-block:: python
   :class: collapse

import example
example.method()
```

12.2.3 sphinx-toolbox's collapse directive

Added in version 1.7.

When using the `sphinx-toolbox` extension, the collapse directive should be able to manage collapsed content. For example:

```
.. collapse:: Details

    This paragraph should be collapsed.
```

For more details, please see `sphinx-toolbox`'s documentation on the [collapse directive](#).

12.3 Confluence Macros

The Confluence Builder extension may attempt to support various stock Confluence macros through the use of custom [directives](#) or [roles](#). However, not all macros will have a markup defined for them with this extension. Some macros may not have a graceful way to be used in documentation, a stock macro has yet to have a directive/role created or the macro is provided by a third-party vendor.

Note

Typically, third-party vendor macros are not added into this extension. Exceptions may be made for free addons. However, this extension’s main goal is to focus on supporting stock macros.

If a stock Confluence macro does not have a directive/role created for it, users are welcome to submit a pull request to support the macro or create an issue outlining a desire for one. For developers wishing to support a third-party macro, it is recommended that developers create a custom extension providing support for this macro that can integrate with the Confluence Builder extension.

That all being said, users looking to use macros that do not have a directive/role define can add any custom macro using the [raw](#) directive. Consider the following examples. A user can add a “[Cheese](#)” macro using the following in a `reStructuredText` document:

```
.. raw:: confluence_storage

    <ac:structured-macro ac:name="cheese"
      ac:macro-id="5d87ef61-1fcf-4d30-8d3b-82305e13233a" />
```

Or create an “[Info](#)” macro in a Markdown document:

```
```{raw} confluence_storage
<ac:structured-macro ac:name="info" ac:macro-id="ecc1aad7-f2f5-4b75-86d1-786370e6d499">
 <ac:rich-text-body>
 <p>Test</p>
 </ac:rich-text-body>
</ac:structured-macro>
```
```

Try adding either of these example into a document to observe their results.

After trying out adding a raw macro from above, follow the steps in the next section to help add a custom macro to a document.

12.3.1 Steps

To add a custom macro available on a Confluence instance, perform the following steps:

1. Creating a new page on the Confluence instance
2. Add the specific macro to be captured on the page and saving
3. Selecting the page’s option menu and selecting “View Storage format”
4. Look for the `ac:structured-macro` tags in the storage format
5. Copy the tags and place them into a raw-directive in the documentation

12.4 Configuring the default highlighter

Sphinx's default highlight language is `python`. For any documentation processed by this extension that does not have an explicit language type provided will fallback to Sphinx's default.

Users wishing to override the default highlight language can use the `confluence_lang_overrides` configuration. For example, adding the following option inside a project's `conf.py` file:

```
confluence_lang_overrides = {  
    'default': 'none',  
}
```

12.5 Math support

Changed in version 2.10: Limited support for generating MathJax-raw content.

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.

Added 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.

12.5.1 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.

12.5.2 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` 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 see 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>
```

12.5.3 MathJax integration

Important

While this extension aims to support the capability of generating raw content that MathJax could render, the complete solution of using MathJax on a Confluence instance is considered unsupported.

Stock Confluence instances do not include support to render [MathJax](#) content. However, if the instance has been configured to support the JavaScript engine, the Confluence builder extension can be configured to generate output that is compatible with MathJax.

Users can use the `confluence_mathjax` option to output all math content into a MathJax-compatible format:

```
confluence_mathjax = True
```

There are no system administrator guidelines on how to setup a Confluence instance to provide MathJax support.

Users on a Confluence Data Center instance that has enabled support for HTML macros may be able to use the `confluence_html` directive to enable MathJax support on their pages. For example, adding the following into a

project's `conf.py`:

```
rst_prolog = """  
.. confluence_html::  
    <script id="MathJax-script" async src="https://cdn.jsdelivr.net/npm/mathjax@3/es5/  
↪tex-mml-ctml.js"></script>  
"""
```

User experience may vary.

12.6 Configure access for a scoped token

Added in version 2.15.

Note

Scoped tokens only apply to Confluence Cloud.

12.6.1 Setup a scoped token

To create a scoped token, venture to a user's API token management page:

API Tokens

<https://id.atlassian.com/manage-profile/security/api-tokens>

On this page:

- Select the button `Create API token with scopes`.
- Name the API token and configure a desired expiry date.
- Select the Confluence application.
- Check each of the following scopes required for this extension:
 - `read:attachment:confluence`
 - `read:content-details:confluence`
 - `read:content.metadata:confluence`
 - `read:content.property:confluence`
 - `read:page:confluence`
 - `read:space:confluence`
 - `write:attachment:confluence`
 - `write:content.property:confluence`
 - `write:page:confluence`
 - `write:watcher:confluence`
 - `delete:attachment:confluence`
 - `delete:page:confluence`
- Review and create.

12.6.2 Configuring the extension

Using a scoped token requires at least one of the following two configuration changes. The easiest option is to hint to this extension that a scoped token is being used by configuring:

```
confluence_api_token_scoped = True
```

This should help automatically resolve a modern API endpoint for Confluence interaction. Alternatively, if wanting to explicitly configure the modern API endpoint, determine a space's Cloud identifier (if not already) by venturing to:

```
https://<SPACE>.atlassian.net/_edge/tenant_info
```

For example:

```
https://example.atlassian.net/_edge/tenant_info
```

This page should provide a GUID value which can be used in a modern Confluence API endpoint required for scoped tokens:

```
confluence_server_url = 'https://api.atlassian.com/ex/confluence/<GUID>/'
```

For example:

```
confluence_server_url = 'https://api.atlassian.com/ex/confluence/550e8400-e29b-41d4-a716-  
↪446655440000/'
```

12.7 Strike content

There are a series of strike element capabilities supported by the Confluence builder extension. This guide will cover the available options that can be used.

12.7.1 Strike role

Added in version 2.1.

This extension provides a `confluence_strike` role to help strike content. For example:

```
:confluence_strike: `My text`
```

This role will only work when using the Confluence builder. See also the [Confluence Strikethrough role](#).

12.7.2 Class hints

Added in version 1.7.

When this class hint is applied to inlined text, the text will be styled with a strikethrough style. Sphinx projects may define the following in a document to support the strikethrough of text in both the `html` and `confluence` builders (assuming HTML documents are also prepared to support the `strike` class):

```
.. role:: strike
   :class: strike

This is an :strike:`example`.
```

Projects may also move the role definition inside the project's `conf.py` inside `rst_prolog` as follows:

```
rst_prolog = """
.. role:: strike
   :class: strike
"""
```

Projects using [Markedly Structured Text - Parser](#) can also take advantage of the role defined in `rst_prolog` (above) and use the role as follows:

```
This is an {strike}`example`.
```

12.7.3 sphinxnotes-strike's strike role

Added in version 2.2.

When using the `sphinxnotes-strike` extension, a `strike` role should be able to manage striked content. For example:

```
:strike:`My text`
```

12.8 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.

Caution

It is never recommended to store an API token or raw password into a committed/shared repository holding documentation. This guide should help ensure this is not required for users.

- Confluence environment variables
- Configuration overrides
- Manual configuration processing

Documentation can be published automatically through a CI system. While users may wish to customize a publish location or tweak title postfixes for a job, in almost all cases, the most important options to configure are specific to authentication (i.e. how to properly use a token to publish content without embedded the token into source). The following will focus on how to properly forward a token into a Sphinx configuration to allow publishing. The same approach can be used for tailoring other Confluence configuration entries in a CI environment.

There are three approaches that can be used to help accept a CI managed publish token:

- 1) Defining an environment variable to forward into the extension.
- 2) Override a configuration option when invoking a Sphinx build.
- 3) Adding the ability to fetch configuration values from the environment or file with tweaks to a `conf.py` definition.

Before demonstrating these methods, please note which type of authentication is required for the target Confluence instance. For example, if authenticating with an API key (Confluence Cloud; see [API tokens](#)), users will need to configure both `confluence_server_user` and `confluence_api_token` options. However, if using a personal access token (see [Using Personal Access Tokens](#)), users will need to configure only the `confluence_publish_token` option.

12.8.1 Confluence environment variables

Added in version 1.9.

The Confluence builder extension accepts most configuration entries from the environment if the option is not already set in `conf.py`.

Confluence Cloud API Key

If using a Confluence Cloud API key, ensure the following variables are *not set* inside `conf.py`:

- `confluence_api_token`
- `confluence_publish_token`
- `confluence_server_pass`

The option `confluence_server_user` may be set if a user will only ever be published with a single API token. If the environment plans to use multiple tokens, ensure `confluence_server_user` is not set as well.

Next, if the CI environment supports defining custom CI variables, create a new entry for `CONFLUENCE_API_TOKEN`, holding the API token value to use when publishing. If the API token is stored in another manner that can be exposed when running a build, ensure the token is set into a `CONFLUENCE_API_TOKEN` environment variable before running Sphinx. For example:

```
$ export CONFLUENCE_API_TOKEN="<my-token-value>"
$ sphinx-build ...
Running Sphinx
...
```

Or, when using a Windows command line:

```
> set CONFLUENCE_API_TOKEN="<my-token-value>"
> sphinx-build ...
Running Sphinx
...
```

The same applies to `CONFLUENCE_SERVER_USER` if the username field needs to be set.

Confluence Data Center PAT

If using a PAT, ensure the following variables are *not set* inside `conf.py`:

- `confluence_api_token`
- `confluence_publish_token`
- `confluence_server_pass`
- `confluence_server_user`

Next, if the CI environment supports defining custom CI variables, create a new entry for `CONFLUENCE_PUBLISH_TOKEN`, holding the PAT value to use when publishing. If the PAT is stored in another manner that can be exposed when running a build, ensure the token is set into a `CONFLUENCE_PUBLISH_TOKEN` environment variable before running Sphinx. For example:

```
$ export CONFLUENCE_PUBLISH_TOKEN="<my-token-value>"
$ sphinx-build ...
Running Sphinx
...
```

Or, when using a Windows command line:

```
> set CONFLUENCE_PUBLISH_TOKEN="<my-token-value>"
> sphinx-build ...
Running Sphinx
...
```

12.8.2 Configuration overrides

Sphinx supports providing configuration overrides from the command line.

Confluence Cloud API Key

The following can be used to configure an API token for Confluence Cloud:

```
sphinx-build ... -Dconfluence_api_token="<my-token-value>"
```

Confluence Data Center PAT

For an environment using a PAT for Confluence Data Center, a PAT can be configured as follows:

```
sphinx-build ... -Dconfluence_publish_token="<my-token-value>"
```

12.8.3 Manual configuration processing

Users are free to use custom implementation inside their `conf.py` file to help manage their configuration in a CI environment. The following shows two examples that read an environment variable `SECRET_KEY` prepared in a CI environment to be used for authentication.

Confluence Cloud API Key

If using an API token, the following can be used:

```
import os

...

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

Confluence Data Center PAT

If using a personal access token, the following can be used:

```
import os

...

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

12.9 Single Sign-On (SSO) and OAuth Environments

A percentage of extension users may be trying to operate with a Confluence instance which deploys some sort of single sign-on (SSO) or OAuth capability. How an instance is configured and the means to interact can vary per organization, even per instance. There are no definitive steps for configuring this extension to guarantee a graceful setup. That being said, there should be a means to allow interaction when using this extension. This guide may be useful for users configuring this extension for the first time in a corporate environment.

12.9.1 Confluence Authentication Token

Users should be using either an API token (Confluence Cloud) or personal access token (PAT; Confluence Data Center) for authenticating with Confluence. This means that `confluence_api_token` (along with `confluence_server_user`) should be used in a Confluence Cloud setting, and `confluence_publish_token` should be used in a Confluence Data Center setting (e.g. corporate/self-hosted).

12.9.2 Initial Test

Once a token has been configured, it is recommended to perform a connection attempt. This is to help ensure a project configuration and extension is ready to perform interaction with a Confluence instance, as well as to perform a sanity check that additional SSO/OAuth configuration is required.

From the working directory of a project, perform a [connection test](#):

```
$ python -m sphinxcontrib.confluencebuilder connection-test
Fetching configuration information...
...
```

12.9.3 Configuring for Identity or Authorization Instance

This extension uses [Requests](#) for HTTP requests. Options are available to help allow a user tailor requests. These options can be used to inject additional token/session information in a request required for a front-end of a Confluence instance.

The following options may help:

- `confluence_publish_headers`: Used to set HTTP headers for requests made to a Confluence instance.
- `confluence_server_cookies`: Used to provide cookie data for requests made to a Confluence instance.
- `confluence_server_auth`: Used to provide a custom Requests-capable authentication handler for requests made to a Confluence instance.

Which option and values to use will vary. It is recommended, if available, to refer to any documentation provided by the Confluence instance about what setup is used and what header/cookie data they expect. If no such information is available, it is recommended to either create a ticket with the service desk associated with the Confluence instance or talk with a system administrator for the Confluence instance.

12.9.4 Additional information

- To help debug the network traffic between this extension and a Confluence instance, the `confluence_publish_debug` option may be used.
- Refer to [connection-issued issues](#) reported by users.

13.1 Extensions

The Sphinx Confluence Builder is powered by [Sphinx](#). There are several ways individual projects may extend their configuration or use other extensions to improve the processing and generation of their documentation.

For users new to Sphinx, it is recommended to read Sphinx's documentation on extensions:

Sphinx Extensions API

<https://www.sphinx-doc.org/en/master/extdev/index.html>

Developers or advanced users are recommended to use official Sphinx API calls for any modifications that are applicable when using the Sphinx Confluence Builder extension. Users are welcome to inspect this extension's code for any custom tweaks desired for their project.

The following shows an example modification for a project to support a custom node type that is not supported by this extension. Users can use [Sphinx.add_node](#) to help register support for custom nodes. The keyword to use for this extension is `confluence`. If another extension defines a node `custom_node`, the following shows some example code to start supporting this node with Sphinx Confluence Builder:

```
def setup(app):
    app.add_node(
        custom_node,
        confluence=(visit_custom_node, depart_custom_node),
    )

def visit_custom_node(self, node):
    self.body.append(' [pre] ')
    self.context.append(' [post] ')

def depart_custom_node(self, node):
    self.body.append(self.context.pop())
```

13.1.1 Confluence Storage Format Translator Helpers

When extending this extension or adding more advanced configurations, a series of helper calls are available for use. API calls may evolve over time as this extension is maintained. The following methods are available for a `ConfluenceStorageFormatTranslator` translator.

start_tag(*self*, *node*, *tag*, *suffix=None*, *empty=False*, ***kwargs*)

A helper used to return content to be appended to a document which initializes the start of a storage format element (i.e. generates a start tag). The element of type `tag` will be initialized. This method may use the provided `node` to tweak the final content. Non-empty tags should be used with a respective `end_tag()` call.

```
def visit_custom_node(self, node):
    self.body.append(self.start_tag(node, 'p'))
    self.context.append(self.end_tag(node))

def depart_custom_node(self, node):
    self.body.append(self.context.pop())
```

Parameters

- **node** – the node processing the start-tag
- **tag** – the type of tag
- **suffix** (*optional*) – the suffix to add (defaults to nothing)
- **empty** (*optional*) – tag will not hold child nodes (defaults to `False`)
- ****kwargs** (*optional*) – dictionary of attributes to include in the tag

Returns

the content

end_tag(*self*, *node*, *suffix=None*)

A helper used to return content to be appended to a document which finalizes a storage format element (i.e. generates an end tag). This method should be used to help close a `start_tag()` call (with the exception of when `start_tag()` is invoked with `empty=True`).

Parameters

- **node** – the node processing the end-tag
- **suffix** (*optional*) – the suffix to add (defaults to newline)

Returns

the content

start_ac_image(*self*, *node*, ***kwargs*)

A helper used to return content to be appended to a document which initializes the start of a storage format image element. The `ac:image` element will be initialized. This method may use provided `node` to tweak the final content. This call should be used with a respective `end_ac_image()` call.

Parameters

node – the node processing the image

Returns

the content

end_ac_image(*self*, *node*)

A helper used to return content to be appended to a document which finalizes a storage format image element. This method should be used to help close a `start_ac_image()` call.

Parameters

node – the node processing the image

Returns

the content

start_ac_link(*self*, *node*, *anchor=None*, *appearance=None*)

A helper used to return content to be appended to a document which initializes the start of a storage format link element of a specific `type`. The `ac:link` element will be initialized. This method may use provided `node` to tweak the final content. This call should be used with a respective `end_ac_link()` call.

Parameters

- **node** – the node processing the link
- **anchor** (*optional*) – the anchor value to use (defaults to `None`)
- **appearance** (*optional*) – card appearance to use (defaults to `None`)

Returns

the content

end_ac_link(*self*, *node*)

A helper used to return content to be appended to a document which finalizes a storage format link element. This method should be used to help close a `start_ac_link()` call.

Parameters

node – the node processing the link

Returns

the content

start_ac_link_body(*self*, *node*)

A helper used to return content to be appended to a document which initializes the start of a storage format link-body element. The `ac:link-body` element will be initialized. This method may use provided `node` to tweak the final content. This call should be used with a respective `end_ac_link_body()` call.

Parameters

node – the node processing the macro

Returns

the content

end_ac_link_body(*self*, *node*)

A helper used to return content to be appended to a document which finalizes a storage format link-body element. This method should be used to help close a `start_ac_link_body()` call.

Parameters

node – the node processing the macro

Returns

the content

start_ac_macro(*self*, *node*, *type_*, *empty=False*)

A helper used to return content to be appended to a document which initializes the start of a storage format macro element of a specific `type`. The `ac:structured-macro` element will be initialized. This method may use provided `node` to tweak the final content. This call should be used with a respective `end_ac_macro()` call.

Parameters

- **node** – the node processing the macro
- **type** – the type of macro

- **empty** (*optional*) – tag will not hold child nodes (defaults to False)

Returns

the content

build_ac_param(*self, node, name, value*)

A helper used to return content to be appended to a document which builds a complete storage format parameter element. The `ac:parameter` element will be built. This method may use provided `node` to tweak the final content.

Parameters

- **node** – the node processing the parameter
- **name** – the parameter name
- **value** – the value for the parameter

Returns

the content

end_ac_macro(*self, node, suffix=None*)

A helper used to return content to be appended to a document which finalizes a storage format macro element. This method should be used to help close a `start_ac_macro()` call (with the exception of when `start_ac_macro()` is invoked with `empty=True`).

Parameters

- **node** – the node processing the macro
- **suffix** (*optional*) – the suffix to add (defaults to newline)

Returns

the content

start_ac_plain_text_body_macro(*self, node*)

A helper used to return content to be appended to a document which initializes the start of a storage format plain-text-body element. The `ac:plain-text-body` element will be initialized. This method may use provided `node` to tweak the final content. This call should be used with a respective `end_ac_plain_text_body_macro()` call.

Parameters

node – the node processing the macro

Returns

the content

end_ac_plain_text_body_macro(*self, node*)

A helper used to return content to be appended to a document which finalizes a storage format plain-text-body element. This method should be used to help close a `start_ac_plain_text_body_macro()` call.

Parameters

node – the node processing the macro

Returns

the content

start_ac_plain_text_link_body_macro(*self, node*)

A helper used to return content to be appended to a document which initializes the start of a storage format plain-text-body element. The `ac:plain-text-body` element will be initialized. This method may use provided `node` to tweak the final content. This call should be used with a respective `end_ac_plain_text_link_body_macro()` call.

Parameters

node – the node processing the macro

Returns

the content

end_ac_plain_text_link_body_macro(*self, node*)

A helper used to return content to be appended to a document which finalizes a storage format plain-text-link-body element. This method should be used to help close a `start_ac_plain_text_link_body_macro()` call.

Parameters

node – the node processing the macro

Returns

the content

start_ac_rich_text_body_macro(*self, node*)

A helper used to return content to be appended to a document which initializes the start of a storage format rich-text-body element. The `ac:rich-text-body` element will be initialized. This method may use provided `node` to tweak the final content. This call should be used with a respective `end_ac_rich_text_body_macro()` call.

Parameters

node – the node processing the macro

Returns

the content

end_ac_rich_text_body_macro(*self, node*)

A helper used to return content to be appended to a document which finalizes a storage format rich-text-body element. This method should be used to help close a `start_ac_rich_text_body_macro()` call.

Parameters

node – the node processing the macro

Returns

the content

start_adf_content(*self, node*)

A helper used to return content to be appended to a document which initializes the start of a storage format adf-content element. The `ac:adf-content` element will be initialized. This method may use provided `node` to tweak the final content. This call should be used with a respective `end_adf_content()` call.

Parameters

node – the node processing the macro

Returns

the content

end_adf_content(*self, node, suffix=None*)

A helper used to return content to be appended to a document which finalizes a storage format adf-content element. This method should be used to help close a `start_adf_content()` call.

Parameters

- **node** – the node processing the macro
- **suffix** (*optional*) – the suffix to add (defaults to newline)

Returns

the content

start_adf_extension(*self, node*)

A helper used to return content to be appended to a document which initializes the start of a storage format adf-extension element. The `ac:adf-extension` element will be initialized. This call should be used with a respective `start_adf_extension()` call.

Parameters

node – the node processing the macro

Returns

the content

build_adf_attribute(*self, node, name, value*)

A helper used to return content to be appended to a document which builds a complete storage format parameter element. The `ac:adf-attribute` element will be built. This method may use provided `node` to tweak the final content.

Parameters

- **node** – the node processing the parameter
- **name** – the parameter name
- **value** – the value for the parameter

Returns

the content

end_adf_extension(*self, node, suffix=None*)

A helper used to return content to be appended to a document which finalizes a storage format adf-extension element. This method should be used to help close a `start_adf_extension()` call.

Parameters

- **node** – the node processing the macro
- **suffix** (*optional*) – the suffix to add (defaults to newline)

Returns

the content

start_adf_node(*self, node, type_, empty=False*)

A helper used to return content to be appended to a document which initializes the start of a storage format adf-node element of a specific `type_`. The `ac:adf-node` element will be initialized. This method may use provided `node` to tweak the final content. This call should be used with a respective `end_adf_node()` call.

Parameters

node – the node processing the macro

Returns

the content

end_adf_node(*self, node, suffix=None*)

A helper used to return content to be appended to a document which finalizes a storage format adf-node element. This method should be used to help close a `start_adf_node()` call (with the exception of when `start_adf_node()` is invoked with `empty=True`).

Parameters

- **node** – the node processing the macro
- **suffix** (*optional*) – the suffix to add (defaults to newline)

Returns

the content

start_ri_attachment(*self*, *node*, *filename*)

A helper used to return content to be appended to a document which initializes the start of a storage format attachment element. The `ri:attachment` element will be initialized. This method may use provided `node` to tweak the final content.

Parameters

- **node** – the node processing the attachment
- **filename** – the filename of the attachment

Returns

the content

end_ri_attachment(*self*, *node*)

A helper used to return content to be appended to a document which finalizes a storage format attachment element. This method should be used to help close a `start_ri_attachment()` call.

Parameters**node** – the node processing the attachment**Returns**

the content

escape_cdata(*self*, *data*)

A helper used to return content that has been properly escaped and can be directly placed inside a CDATA container.

Parameters**data** – the text**Returns**

the escaped text

13.2 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` capability.

For environments using an OAuth connector or scoped API tokens, the following scopes are required:

```
read:space:confluence
read:content.metadata:confluence
read:page:confluence
write:page:confluence
delete:page:confluence
read:attachment:confluence
read:content-details:confluence
write:attachment:confluence
delete:attachment:confluence
write:watcher:confluence
```

13.3 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
```

CHANGELOG

14.1 3.1 (2026-04-03)

- Handle Confluence dropped reference anchors on first publish
- Improve reported publish URL for setups using direct Cloud API endpoint
- Support auto-retry on immediate page update version conflicts
- Support new code types on Confluence Cloud (Handlebars, Markdown and TOML)
- Support updating with resolved comments when setup to skip commented pages

14.2 3.0 (2026-01-18)

- **(note)** Default editor for Confluence Cloud is now v2
- All deprecated options in 2.x and older have been removed
- Deprecated the `confluence_editor` option
- Introduce the `confluence_cloud` option
- Introduce the `confluence_publish_hash_modifier` option
- Updated the connection test to include more publish-related options

14.3 2.17.1 (2025-12-19)

- Fixed use of epigraphs and pull-quotes with optional attribution

14.4 2.17.0 (2025-12-18)

- Fixed regression where only an attachment update could cause a publish to fail
- Fixed regression where publishing nested pages may fail
- Support epigraphs and pull-quotes with optional attribution

14.5 2.16.0 (2025-11-30)

- **(note)** Dropped support for Python 3.9 (EOL)
- Fixed regression with sphinxcontrib-video support failing to process
- Fixed regression with the dry-run mode failing to run
- Improve the presentation in some line block usage scenarios in v2 editor

14.6 2.15.0 (2025-10-26)

- **(note)** Final release supporting Python 3.9 (EOL)
- Assets are always standalone (`confluence_asset_force_standalone` dropped)
- Fixed reference conflicts with footnotes/citations in `singleconfluence`
- Introduce the Confluence Table Width directive
- Improved support for API endpoint resolution when using scoped API tokens
- Improved support for Confluence Cloud detection when using scoped API tokens
- Introduce the `confluence_default_table_width` option
- Introduce the `confluence_publish_trampoline` option
- Properly handle late stage assets (e.g. `plantuml`) in POSIX parallel builds

14.7 2.14.0 (2025-07-26)

- Fixed `get_and_resolve_doctree` usage when using `singleconfluence` for future Sphinx releases
- Fixed certain section references when publishing on Confluence Cloud
- Introduce the Confluence PDF (macro) directive
- Limited support for inlined tabs capability
- Page-specific editor overrides is deprecated
- Support references to specific list items (and some other content types)

14.8 2.13.0 (2025-06-08)

- Fixed `render_math` usage required for future Sphinx releases
- Fixed broken references when using Emojis
- Generate a warning when inlined comments will be dropped
- Improve error messages on update attempts on restricted pages
- Introduce the Confluence Panel (macro) directive
- Support the ability to skip page updates on a page that has inlined comments

14.9 2.12.0 (2025-03-03)

- Correct regression in reporting API failures in failed page updates since v2.6
- Ensure API failures are reported after exhausting failed retry attempts
- Fixed issue where header debug logging not printing request headers
- Help rebuild page properties on rare duplicate property scenarios

14.10 2.11.0 (2025-02-18)

- Addressed path-related issue with graphviz/inheritance-diagrams for Sphinx v9
- Fixed extra newlines generated with `productionlist` when using Sphinx v8.2
- Publish retry events trigger on all `transient errors`
- Support `no-wrap` for math directives
- Support collapsible code blocks for v2 editor pages
- Support collapsible option for admonitions (unsightly)

14.11 2.10.1 (2025-02-02)

- Correct publish fail with key check mismatch on new Confluence Cloud spaces

14.12 2.10.0 (2025-02-02)

- Disable use of inlined images for v2 editor pages
- Fixed issue where MinGW getpass workaround could throw an exception
- Introduce the `confluence_publish_retry_attempts` option
- Introduce the `confluence_publish_retry_duration` option
- Support for generating and publish pages with raw MathJax content

14.13 2.9.0 (2024-11-11)

- Fixed issue with env-relative assets in sub-folders (e.g. download role)
- Fixed some section references between pages for v2 editor pages

14.14 2.8.0 (2024-10-14)

- Fixed issues where various links were broken with singleconfluence
- Improve support with Sphinx v8.1 `--fail-on-warning` changes
- Improve support with Sphinx v8.1 `sphinx.ext.linkcode` changes
- Introduce the `confluence-publish-attachment` event
- Introduce the `confluence-publish-override-pageid` event
- Introduce the `confluence-publish-page` event

14.15 2.7.1 (2024-08-26)

- Fixed project dependency for Sphinx 7.1

14.16 2.7.0 (2024-08-25)

- **(note)** `confluence_file_transform` support is dropped
- **(note)** `confluence_link_transform` support is dropped
- Improve support when using the `sphinx_toolbox` extension
- Improved support for newer Confluence Cloud code formats
- Introduce the Confluence HTML directive
- Introduce the `confluence_disable_env_conf` option
- Support inlined images for v2 editor pages
- Update publisher's user agent to `SphinxConfluenceBuilder/<version>`

14.17 2.6.1 (2024-07-27)

- Support Sphinx's 8.x switch to pathlib

14.18 2.6.0 (2024-07-14)

- **(note)** `confluence_disable_notifications` is now enabled by default
- **(note)** `confluence_file_transform` support is deprecated
- **(note)** `confluence_link_transform` support is deprecated
- Fixed inconsistencies with page-meta editor/full-width with singleconfluence
- Fixed issue in Confluence REST v2 API usage (Cloud) where page updates may fail when a page has no metadata
- Fixed issue using cleanup modes with Confluence Cloud with large attachment/page sets (250+) could block publishing
- Fixed issues where various links were broken with singleconfluence
- Fixed support when using the `sphinxcontrib-video` extension (v0.2.1+)
- Improve support when using the `sphinx-designs` extension
- Improve user feedback when operating with a cleaning option enabled
- Introduce the `confluence-publish-point` event
- Introduce the `confluence_api_token` option
- Introduce the `confluence_page_search_mode` option
- REST API calls will retry on select error events (e.g. 5xx errors)
- Replacing `confluence_lang_transform` with a new option `confluence_lang_overrides`
- Translator helpers are now public for advanced configurations/extensions

14.19 2.5.2 (2024-05-12)

- Fixed issue in Confluence REST v2 API usage (Cloud) where page updates may fail when a page has no metadata

14.20 2.5.1 (2024-04-06)

- Fixed issue in Confluence REST v2 API usage (Cloud) where edited pages on Confluence prevented this extension from publishing updates

14.21 2.5.0 (2024-03-31)

- Confluence card directives/roles can be processed with `linkcheck`
- Fixed issue where an exception could throw without cleanup options set
- Fixed issue where anchor links may fail to work for v2 editor pages
- Fixed issue where generated pages would use incorrect template with an explicit v1 editor
- Fixed issue where index/search pages on v1 do not respect width configuration
- Fixed regression where math label anchors would not be created in v1 editor
- Fixed regression where search pages would not be set with a proper space key
- Improve search box alignment on generated search pages
- Initial support for using Confluence REST v2 API (Cloud)
- Provide extended debugging options for publish events
- Replacing `confluence_publish_disable_api_prefix` with a new option `confluence_publish_override_api_prefix`
- Simplify autodocs rendering in v2 to make it somewhat usable
- Support custom page notice via `confluence_page_generation_notice`

14.22 2.4.0 (2023-12-14)

- Fixed anchor links between various editor versions and when using MyST
- Fixed issue when using markup in captions when using the v2 editor
- Fixed issue with an editor metadata override may led to unexpected page styles
- Fixed issue with dry-run reporting pages when configure root does not exist
- Fixed regression in processing metadata directives on a page
- Prevent undesired spacing when adding targets in paragraphs with v2 editor
- Support width hints in v2 editor for CSV tables

14.23 2.3.0 (2023-09-09)

- **(note)** Final release supporting Python 3.7 (EOL)
- Fixed issue when a project defines a relative CA certificate path
- Fixed issue when a project defines relative publish list files
- Fixed issue where cleanup would remove up-to-date pages on rebuilds
- Fixed triggering rebuilds when select configuration options change
- Fixed select directive/role issues with Sphinx 7.2.x

14.24 2.2.0 (2023-07-22)

- Fixed `confluence_full_width` issues on Confluence server/DC
- Fixed issue where using `confluence_full_width` breaks an editor selection
- Fixed re-publishing issues when certain options change (e.g. parent page)
- Improve support when using the `sphinxcontrib-video` extension
- Introduce the `confluence_permit_raw_html` option
- Provide quirk for CDATA issues on Confluence 8.0.x to 8.2.x
- Support configuring the theme on generated code block macros
- Support page-specific editor and full-width overrides
- Support page-specific parent identifier overrides when publishing

14.25 2.1.1 (2023-05-19)

- Ensure source distribution includes the LICENSE document
- Fixed issue where v2 editor may fail with object has no attribute error

14.26 2.1.0 (2023-05-15)

- Allow users to configure legacy page search mode for cleanup
- Fixed anchor page links with v2 editor
- Fixed document processing issues with Sphinx 6.1.x
- Improve rendering when using sphinxcontrib-needs extension
- Introduce Confluence Smart link directives/roles
- Introduce the Confluence strike role
- Perform an attachment re-upload attempt on an unexpected Confluence 503 error
- Provide fallback styling for code languages with a similar style
- Support Codeberg helper for source links
- Support `confluence_full_width` with v1 editor
- Support default-fallback when using `confluence_lang_transform`
- Support deployment with Python 3.12
- Support publishing only pages with content changes
- Support suppressing extension warnings using Sphinx's `suppress_warnings`
- Support the ability to configure where orphan pages are stored
- Support the ability to not publish orphan pages

14.27 2.0.0 (2023-01-02)

- Fixed issue publishing orphan pages when a publish root is configured
- Fixed issue where captioned code blocks may not be numbered with `numfig`
- Fixed issue where captioned tables were not be numbered with `numfig`
- Hierarchy mode is now enabled by default
- Improve look of quote-like directives
- Introduce the Confluence excerpt (macro) directives
- Support Confluence Cloud's "Fabric" (v2) editor
- Support collapsible code blocks
- Support for Python 2.7 has been dropped
- Support for `confluence_max_doc_depth` has been dropped
- Support no publishing with an empty `confluence_publish_allowlist`

14.28 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

14.29 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

14.30 1.7.1 (2021-11-30)

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

14.31 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

14.32 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

14.33 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

14.34 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)

14.35 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

14.36 1.2.0 (2020-01-03)

- **(note)** Sphinx v1.[6-7] support for this extension is deprecated
- **(note)** XML-RPC support for this extension is deprecated
- Fixed issue when using hierarchy on Sphinx 2.1+ (new citations domain)
- Fixed issue with document names with path separators for windows users
- Fixed issue with multi-line description signatures (e.g. C++ autodocs)
- Fixed issue with processing hidden toctrees
- Fixed issue with Unicode paths with `confluence_publish_subset` and Python 2.7
- Improved formatting for option list arguments
- Improved handling and feedback when configured with incorrect publish instance
- Improved name management for published assets
- Improved reference linking for Sphinx domains capability (meth, attr, etc.)
- Introduce a series of Jira directives
- Support `firstline` parameter in the code block macro
- Support base admonition directive
- Support Confluence 7 series newline management
- Support default alignment in Sphinx 2.1+
- Support document postfixes
- Support for generated image assets (asterisk marked)
- Support passthrough authentication handlers for REST calls
- Support previous/next navigation
- Support prompting for publish username
- Support `sphinx.ext.autosummary` extension
- Support `sphinx.ext.todo` extension
- Support the math directive
- Support toctree's numbered option
- Support users injecting cookie data (for authentication) into REST calls

14.37 1.1.0 (2019-03-16)

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

14.38 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

14.39 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

14.40 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

14.41 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

14.42 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

14.43 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

14.44 0.4.0 (2017-02-21)

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

14.45 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

14.46 0.2.0 (2016-07-13)

- Moved configuration to the Sphinx config

14.47 0.1.1 (2016-07-12)

- Added table support
- Fixed internal links

14.48 0.1.0 (2016-07-12)

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

Symbols

- `:absolute-url:` (*directive option*)
 - `confluence_toc` (*directive*), 68
- `:align:` (*directive option*)
 - `confluence_latex` (*directive*), 73
- `:atlassian-macro-output-type:` (*directive option*)
 - `confluence_excerpt` (*directive*), 63
- `:bg-color:` (*directive option*)
 - `confluence_panel` (*directive*), 66
- `:border-color:` (*directive option*)
 - `confluence_panel` (*directive*), 66
- `:border-style:` (*directive option*)
 - `confluence_panel` (*directive*), 66
- `:border-width:` (*directive option*)
 - `confluence_panel` (*directive*), 66
- `:card:` (*directive option*)
 - `confluence_doc` (*directive*), 74
 - `confluence_link` (*directive*), 75
- `:columns:` (*directive option*)
 - `jira` (*directive*), 71
- `:count:` (*directive option*)
 - `jira` (*directive*), 71
- `:editor:` (*directive option*)
 - `confluence_metadata` (*directive*), 65
- `:exclude:` (*directive option*)
 - `confluence_toc` (*directive*), 68
- `:full-width:` (*directive option*)
 - `confluence_metadata` (*directive*), 65
- `:guid:` (*directive option*)
 - `confluence_metadata` (*directive*), 65
- `:hidden:` (*directive option*)
 - `confluence_excerpt` (*directive*), 63
- `:include:` (*directive option*)
 - `confluence_toc` (*directive*), 68
- `:indent:` (*directive option*)
 - `confluence_toc` (*directive*), 68
- `:labels:` (*directive option*)
 - `confluence_metadata` (*directive*), 65
- `:layout:` (*directive option*)
 - `confluence_doc` (*directive*), 74
 - `confluence_link` (*directive*), 75
- `:max-level:` (*directive option*)
 - `confluence_toc` (*directive*), 69
- `:maximum-issues:` (*directive option*)
 - `jira` (*directive*), 71
- `:min-level:` (*directive option*)
 - `confluence_toc` (*directive*), 69
- `:name:` (*directive option*)
 - `confluence_excerpt` (*directive*), 63
 - `confluence_excerpt_include` (*directive*), 64
- `:nopanel:` (*directive option*)
 - `confluence_excerpt_include` (*directive*), 64
- `:outline:` (*directive option*)
 - `confluence_toc` (*directive*), 69
- `:printable:` (*directive option*)
 - `confluence_toc` (*directive*), 69
- `:separator:` (*directive option*)
 - `confluence_toc` (*directive*), 69
- `:server:` (*directive option*)
 - `jira` (*directive*), 71
 - `jira_issue` (*directive*), 72
- `:server-id:` (*directive option*)
 - `jira` (*directive*), 71
 - `jira_issue` (*directive*), 72
- `:server-name:` (*directive option*)
 - `jira` (*directive*), 71
 - `jira_issue` (*directive*), 72
- `:style:` (*directive option*)
 - `confluence_toc` (*directive*), 69
- `:title:` (*directive option*)
 - `confluence_expand` (*directive*), 64
 - `confluence_panel` (*directive*), 67
- `:title-bg-color:` (*directive option*)
 - `confluence_panel` (*directive*), 67
- `:title-color:` (*directive option*)
 - `confluence_panel` (*directive*), 67
- `:type:` (*directive option*)
 - `confluence_toc` (*directive*), 69
- `:width:` (*directive option*)
 - `confluence_doc` (*directive*), 74
 - `confluence_link` (*directive*), 75

B

Builder

confluence, 61
singleconfluence, 61

C

Class hints, 101

Collapsed, 102

Configuration value

confluence_add_secnumbers, 27
confluence_additional_mime_types, 40
confluence_api_mode, 33
confluence_api_token, 23
confluence_api_token_scoped, 24
confluence_append_labels, 33
confluence_ask_password, 33
confluence_ask_user, 33
confluence_asset_force_standalone, 59
confluence_asset_override, 40
confluence_ca_cert, 40
confluence_cleanup_archive, 34
confluence_cleanup_from_root, 34
confluence_cleanup_purge, 35
confluence_cleanup_search_mode, 40
confluence_client_cert, 41
confluence_client_cert_pass, 41
confluence_cloud, 26
confluence_code_block_theme, 27
confluence_default_alignment, 27
confluence_default_table_width, 27
confluence_disable_autogen_title, 42
confluence_disable_env_conf, 27
confluence_disable_notifications, 35
confluence_disable_ssl_validation, 42
confluence_domain_indices, 28
confluence_editor, 59
confluence_file_suffix, 52
confluence_footer_data, 28
confluence_footer_file, 28
confluence_full_width, 36
confluence_global_labels, 36
confluence_header_data, 28
confluence_header_file, 28
confluence_html_macro, 52
confluence_ignore_titlefix_on_index, 42
confluence_include_search, 29
confluence_jira_servers, 52
confluence_lang_overrides, 52
confluence_lang_transform, 59
confluence_latex_macro, 53
confluence_link_suffix, 53
confluence_manifest_data, 54
confluence_master_homepage, 59
confluence_mathjax, 54
confluence_mentions, 54
confluence_mermaid_html_macro, 57
confluence_navdocs_transform, 55
confluence_page_generation_notice, 29
confluence_page_hierarchy, 29
confluence_page_search_mode, 42
confluence_parent_override_transform, 43
confluence_parent_page, 36
confluence_parent_page_id_check, 59
confluence_permit_raw_html, 55
confluence_prev_next_buttons_location, 29
confluence_proxy, 43
confluence_publish, 21
confluence_publish_allowlist, 43
confluence_publish_debug, 44
confluence_publish_delay, 45
confluence_publish_denylist, 45
confluence_publish_disable_api_prefix, 60
confluence_publish_dryrun, 37
confluence_publish_force, 46
confluence_publish_headers, 46
confluence_publish_intersphinx, 46
confluence_publish_onlynew, 47
confluence_publish_orphan, 47
confluence_publish_orphan_container, 47
confluence_publish_override_api_prefix, 48
confluence_publish_postfix, 37
confluence_publish_postfix_hash_modifier, 47
confluence_publish_prefix, 38
confluence_publish_retry_attempts, 48
confluence_publish_retry_duration, 49
confluence_publish_root, 38
confluence_publish_skip_commented_pages, 49
confluence_publish_subset, 60
confluence_publish_token, 25
confluence_publish_trample, 49
confluence_purge_from_master, 60
confluence_purge_from_root, 60
confluence_remove_title, 56
confluence_request_session_override, 50
confluence_root_homepage, 38
confluence_secnumber_suffix, 30
confluence_server_auth, 50
confluence_server_cookies, 50
confluence_server_pass, 26
confluence_server_url, 21
confluence_server_user, 22
confluence_sourcelink, 30
confluence_space_key, 21
confluence_space_name, 60
confluence_tab_macro, 56
confluence_timeout, 38
confluence_title_overrides, 50

- confluence_use_index, 32
- confluence_version_comment, 51
- confluence_watch, 39
- singleconfluence_toctree, 32
- suppress_warnings, 58
- confluence
 - Builder, 61
- Confluence Macros, 103
- confluence_add_secnumbers
 - Configuration value, 27
- confluence_additional_mime_types
 - Configuration value, 40
- confluence_api_mode
 - Configuration value, 33
- confluence_api_token
 - Configuration value, 23
- confluence_api_token_scoped
 - Configuration value, 24
- confluence_append_labels
 - Configuration value, 33
- confluence_ask_password
 - Configuration value, 33
- confluence_ask_user
 - Configuration value, 33
- confluence_asset_force_standalone
 - Configuration value, 59
- confluence_asset_override
 - Configuration value, 40
- confluence_ca_cert
 - Configuration value, 40
- confluence_cleanup_archive
 - Configuration value, 34
- confluence_cleanup_from_root
 - Configuration value, 34
- confluence_cleanup_purge
 - Configuration value, 35
- confluence_cleanup_search_mode
 - Configuration value, 40
- confluence_client_cert
 - Configuration value, 41
- confluence_client_cert_pass
 - Configuration value, 41
- confluence_cloud
 - Configuration value, 26
- confluence_code_block_theme
 - Configuration value, 27
- confluence_default_alignment
 - Configuration value, 27
- confluence_default_table_width
 - Configuration value, 27
- confluence_disable_autogen_title
 - Configuration value, 42
- confluence_disable_env_conf
 - Configuration value, 27
- confluence_disable_notifications
 - Configuration value, 35
- confluence_disable_ssl_validation
 - Configuration value, 42
- confluence_doc (*directive*), 74
 - :card: (*directive option*), 74
 - :layout: (*directive option*), 74
 - :width: (*directive option*), 74
- confluence_doc (*role*), 81
- confluence_domain_indices
 - Configuration value, 28
- confluence_editor
 - Configuration value, 59
- confluence_emoticon (*role*), 77
- confluence_excerpt (*directive*), 63
 - :atlassian-macro-output-type: (*directive option*), 63
 - :hidden: (*directive option*), 63
 - :name: (*directive option*), 63
- confluence_excerpt_include (*directive*), 63
 - :name: (*directive option*), 64
 - :nopanel: (*directive option*), 64
- confluence_expand (*directive*), 64
 - :title: (*directive option*), 64
- confluence_file_suffix
 - Configuration value, 52
- confluence_footer_data
 - Configuration value, 28
- confluence_footer_file
 - Configuration value, 28
- confluence_full_width
 - Configuration value, 36
- confluence_global_labels
 - Configuration value, 36
- confluence_header_data
 - Configuration value, 28
- confluence_header_file
 - Configuration value, 28
- confluence_html (*directive*), 64
- confluence_html_macro
 - Configuration value, 52
- confluence_ignore_titlefix_on_index
 - Configuration value, 42
- confluence_include_search
 - Configuration value, 29
- confluence_jira_servers
 - Configuration value, 52
- confluence_lang_overrides
 - Configuration value, 52
- confluence_lang_transform
 - Configuration value, 59
- confluence_latex (*directive*), 73
 - :align: (*directive option*), 73
- confluence_latex (*role*), 79

- confluence_latex_macro
 - Configuration value, 53
- confluence_link (*directive*), 74
 - :card: (*directive option*), 75
 - :layout: (*directive option*), 75
 - :width: (*directive option*), 75
- confluence_link (*role*), 81
- confluence_link_suffix
 - Configuration value, 53
- confluence_manifest_data
 - Configuration value, 54
- confluence_master_homepage
 - Configuration value, 59
- confluence_mathjax
 - Configuration value, 54
- confluence_mention (*role*), 80
- confluence_mentions
 - Configuration value, 54
- confluence_mermaid_html_macro
 - Configuration value, 57
- confluence_metadata (*directive*), 65
 - :editor: (*directive option*), 65
 - :full-width: (*directive option*), 65
 - :guid: (*directive option*), 65
 - :labels: (*directive option*), 65
- confluence_navdocs_transform
 - Configuration value, 55
- confluence_newline (*directive*), 66
- confluence_page_generation_notice
 - Configuration value, 29
- confluence_page_hierarchy
 - Configuration value, 29
- confluence_page_search_mode
 - Configuration value, 42
- confluence_panel (*directive*), 66
 - :bg-color: (*directive option*), 66
 - :border-color: (*directive option*), 66
 - :border-style: (*directive option*), 66
 - :border-width: (*directive option*), 66
 - :title: (*directive option*), 67
 - :title-bg-color: (*directive option*), 67
 - :title-color: (*directive option*), 67
- confluence_parent_override_transform
 - Configuration value, 43
- confluence_parent_page
 - Configuration value, 36
- confluence_parent_page_id_check
 - Configuration value, 59
- confluence_permit_raw_html
 - Configuration value, 55
- confluence_prev_next_buttons_location
 - Configuration value, 29
- confluence_proxy
 - Configuration value, 43
- confluence_publish
 - Configuration value, 21
- confluence_publish_allowlist
 - Configuration value, 43
- confluence_publish_debug
 - Configuration value, 44
- confluence_publish_delay
 - Configuration value, 45
- confluence_publish_denylist
 - Configuration value, 45
- confluence_publish_disable_api_prefix
 - Configuration value, 60
- confluence_publish_dryrun
 - Configuration value, 37
- confluence_publish_force
 - Configuration value, 46
- confluence_publish_headers
 - Configuration value, 46
- confluence_publish_intersphinx
 - Configuration value, 46
- confluence_publish_onlynew
 - Configuration value, 47
- confluence_publish_orphan
 - Configuration value, 47
- confluence_publish_orphan_container
 - Configuration value, 47
- confluence_publish_override_api_prefix
 - Configuration value, 48
- confluence_publish_postfix
 - Configuration value, 37
- confluence_publish_postfix_hash_modifier
 - Configuration value, 47
- confluence_publish_prefix
 - Configuration value, 38
- confluence_publish_retry_attempts
 - Configuration value, 48
- confluence_publish_retry_duration
 - Configuration value, 49
- confluence_publish_root
 - Configuration value, 38
- confluence_publish_skip_commented_pages
 - Configuration value, 49
- confluence_publish_subset
 - Configuration value, 60
- confluence_publish_token
 - Configuration value, 25
- confluence_publish_trample
 - Configuration value, 49
- confluence_purge_from_master
 - Configuration value, 60
- confluence_purge_from_root
 - Configuration value, 60
- confluence_remove_title
 - Configuration value, 56

confluence_request_session_override
 Configuration value, 50
 confluence_root_homepage
 Configuration value, 38
 confluence_secnumber_suffix
 Configuration value, 30
 confluence_server_auth
 Configuration value, 50
 confluence_server_cookies
 Configuration value, 50
 confluence_server_pass
 Configuration value, 26
 confluence_server_url
 Configuration value, 21
 confluence_server_user
 Configuration value, 22
 confluence_sourcelink
 Configuration value, 30
 confluence_space_key
 Configuration value, 21
 confluence_space_name
 Configuration value, 60
 confluence_status (*role*), 82
 confluence_strike (*role*), 83
 confluence_tab_macro
 Configuration value, 56
 confluence_table_width (*directive*), 67
 confluence_timeout
 Configuration value, 38
 confluence_title_overrides
 Configuration value, 50
 confluence_toc (*directive*), 68
 :absolute-url: (*directive option*), 68
 :exclude: (*directive option*), 68
 :include: (*directive option*), 68
 :indent: (*directive option*), 68
 :max-level: (*directive option*), 69
 :min-level: (*directive option*), 69
 :outline: (*directive option*), 69
 :printable: (*directive option*), 69
 :separator: (*directive option*), 69
 :style: (*directive option*), 69
 :type: (*directive option*), 69
 confluence_use_index
 Configuration value, 32
 confluence_version_comment
 Configuration value, 51
 confluence_viewpdf (*directive*), 70
 confluence_watch
 Configuration value, 39
 confluence-publish-attachment (*app, docname,*
 key, aid, meta)
 Event, 85

confluence-publish-override-pageid (*app,*
 docname, meta)
 Event, 86
 confluence-publish-page (*app, docname, pid, meta*)
 Event, 85
 confluence-publish-point (*app, point_url*)
 Event, 85
 Continuous integration, 112

E

Emoticon Macro, 76

Event

confluence-publish-attachment (*app,*
 docname, key, aid, meta), 85
 confluence-publish-override-pageid (*app,*
 docname, meta), 86
 confluence-publish-page (*app, docname, pid,*
 meta), 85
 confluence-publish-point (*app, point_url*), 85

Excerpt Include Macro, 63

Excerpt Macro, 63

Expand Macro, 64

J

Jira

Adding a Jira table, 71
 Adding a single Jira link (*directive*), 72
 Adding a single Jira link (*role*), 78
 Configuring Jira servers, 52

jira (*directive*), 71

:columns: (*directive option*), 71
 :count: (*directive option*), 71
 :maximum-issues: (*directive option*), 71
 :server: (*directive option*), 71
 :server-id: (*directive option*), 71
 :server-name: (*directive option*), 71

jira (*role*), 78

jira_issue (*directive*), 72

:server: (*directive option*), 72
 :server-id: (*directive option*), 72
 :server-name: (*directive option*), 72

L

LaTeX Macro

Adding a LaTeX block (*directive*), 72
 Adding inlined LaTeX (*role*), 78

M

Macros

Confluence Macros, 103
 Emoticon Macro (*role*), 76
 Excerpt Include Macro (*directive*), 63
 Excerpt Macro (*directive*), 63

- Expand Macro (*directive*), 64
- Jira Macro (*directive*), 70
- Jira Macro (*role*), 77
- LaTeX Macro (*directive*), 72
- LaTeX Macro (*role*), 78
- Mentions Macro (*role*), 79
- Panel Macro (*directive*), 66
- PDF Macro (*directive*), 70
- Status Macro (*role*), 81

Math

- LaTeX macros, 107
- Math support, 105
- MathJax, 108
- Recommended options, 106

Mentions

- Configuration, 54
- Macro (*role*), 79

P

Page removal

- Automatically archiving pages, 34
- Automatically purging pages, 35
- Wiping a space, 124

Panel Macro, 66

PDF Macro, 70

S

Scoped Token, 109

singleconfluence

- Builder, 61

singleconfluence_toc tree

- Configuration value, 32

Smart links

- Directives, 73

- Roles, 80

Status Macro, 81

Strike, 111

Strikethrough (*role*), 82

suppress_warnings

- Configuration value, 58

W

Wiping a space, 124