Confluence Publisher Plugin

Plugin Information
View Confluence Publisher on the plugin site for more information.

ⓘ Older versions of this plugin may not be safe to use. Please review the following warnings before using an older version:

- CSRF vulnerability and missing permission checks

This plugin allows you to publish build artifacts as attachments to an Atlassian Confluence wiki page.

- Configuration
    - Global Configuration
    - Job Configuration
    - Editing page markup
- Changelog
    - Version 2.0.2 (July 30, 2018)
    - Version 1.8 (Jan 14, 2013)
    - Version 1.7.1 (Jun 25, 2012)
    - Version 1.6 (May 29, 2012)
    - Version 1.5 (Apr 27, 2012)
    - Version 1.4 (Jan 11, 2012)
    - Version 1.3 (Oct 17, 2011)
    - Version 1.2 (Oct 16, 2011)
    - Version 1.1.1 (Sep 22, 2011)
    - Version 1.1 (Jul 2, 2011)
    - Version 1.0.3 (Jun 19, 2011)
    - Version 1.0.2 (Jun 18, 2011)
    - Version 1.0.1 (Mar 24, 2011)
    - Version 1.0.0 (Feb 28, 2011)

Configuration

Global Configuration

First, configure one or more Confluence sites in the global system configuration. If authentication credentials are not supplied, the plugin will act as an anonymous user – as such your Confluence configuration must support anonymous attachments for that to work.

Job Configuration

With at least one site configured, enable the "Publish to Confluence" post-build action, and select from one of the pre-configured Confluence sites. Enter the Space and Page name where the attachment should end up (e.g., for this URL, Space name = "JENKINS", Page name = "Confluence+Publisher+Plugin").

If you have enabled "Archive the artifacts", and wish to automatically upload everything that was archived, check the box labeled "Attach archived artifacts to page". *Otherwise*, enter an Ant-style fileset pattern to specify which files should be uploaded.

Editing page markup

> ⚠️ Please note that as of Confluence v4.0, the page storage format is no longer supporting "wiki format". That makes it slightly more difficult to use simple hidden macros for the replacements. It is still possible, but at the moment it is a bit convoluted and fragile.

In the "Wiki Markup Replacements" section, select the Editor type from the "Add Replacement" dropdown menu. Available Editors include:

- **Entire Page**
  Entire page content is replaced. *No markers required.*
- **Prepend**
  Content is added to the very beginning of the page. *No markers required.*
- **Append**
  Content is added to the very end of the page. *No markers required.*
- **Before Token**
  Content is inserted before a configurable marker token. *Requires one marker.*
- **After Token**
  Content is inserted after a configurable marker token. *Requires one marker.*
- **Between Tokens**
  Content is inserted between two configurable start/end marker tokens. Existing content between the tokens is replaced. *Requires a start and end marker.*

Once an editor has been selected, choose the type of markup Generator to use. Available Generators:

- **Plain Text**
  Content is entered into a textarea field. The content should be valid wiki markup. Build variables will be expanded.
- **File Contents**
  Enter a filename to pull markup contents from. The file should exist relative to the workspace directory, and should contain plain-text markup. Build variables will be expanded.

You may add multiple markup replacement configurations. However, note that the token-related editors only expect to perform edits once, therefore each replacement configuration should use unique token markers.

If possible, it is recommended to configure two custom User Macros in the Confluence configuration (or have your Confluence administrator configure them):

Sample Confluence 3.x User Macros

- Single-marker macro (used for "Before Token" and "After Token" editors):
  - Name: jenkins-marker
  - `[  ]` Macro has a body (**uncheck** *the box*)
  - "Macro generates HTML markup"
  - Template: *blank*
- Start/End macro (used for the "Between Tokens" editor):
  - Name: jenkins-between
  - `[X]` Macro has a body
  - "Use unprocessed macro body"
  - "Macro generates wiki markup"
  - Template: *blank*

These two macros can then be used in the wiki markup, and will not interfere with normal rendering of the wiki content. Additionally, Confluence's "parameter" syntax for user macros allows adding arbitrary parameters to these macros, thus making it very simple to keep the tokens unique. Some examples:

- `{jenkins-marker:before|section=foo}`
- `{jenkins-marker:after|section=changelog}`
- `{jenkins-between:start|section=log}This will be replaced.{jenkins-between:end|section=log}`

If you choose not to use a confluence macro as the token, it will still work (e.g., you can something else, like "`%%INSERT_AFTER%%`"). However, that text may appear in the rendered markup. For that reason, using the Confluence macro is the recommended approach.

Sample Confluence 4.x User Macros

- Single-marker macro (used for "Before Token" and "After Token" editors):
  - Name: jenkins-marker

- ❌ No Macro Body
- Template:

```
## Macro title: Jenkins Token Marker
## Macro has a body: N
## Body processing: No macro body
## Output: None
##
## Developed by: Joe Hansche
## Date created: 10/17/2011
## Installed by: Joe Hansche

## Allows an "id" parameter in order to make multiple unique markers
## @param id
```

- Start/End macro (used for the "Between Tokens" editor):
  - Name: jenkins-between
  - ❌ Rendered
  - Template:

```
## Macro title: Jenkins Between-Tokens section
## Macro has a body: Y
## Body processing: Rendered
## Output: Piped input
##
## Developed by: Joe Hansche
## Date created: 10/17/2011
## Installed by: Joe Hansche

## Allows an "id" parameter in order to make multiple unique markers
## @param id
$body
```

It's then important to realize that you don't have the same control over the markers as you did in Confluence 3.x, and the marker you used previously will not work in the editor configuration. To know what text to use for the "marker" locations, you'll need to insert the macros in their proper locations, and then view the underlying storage format. These are the analogous examples that correspond to the same examples given above for the Confluence 3.x versions:

- `{jenkins-marker:id=before-foo}`
  - `<p><ac:macro ac:name="jenkins-marker"><ac:parameter ac:name="id">before-foo</ac:parameter></ac:macro></p>`

- `{jenkins-marker:id=after-changelog}`
  - `<p><ac:macro ac:name="jenkins-marker"><ac:parameter ac:name="id">after-changelog</ac:parameter></ac:macro></p>`

- `{jenkins-between:id=log}` (this won't actually render the same way, and will instead show a block section that has some body being rendered)
  - `<ac:macro ac:name="jenkins-between"><ac:parameter ac:name="id">log</ac:parameter><ac:parameter ac:name="atlassian-macro-output-type">BLOCK</ac:parameter><ac:rich-text-body>`

Here's an example of what it might look like while in **Edit mode**:

Changelog

Version 2.0.2 (July 30, 2018)

- Fix security issue

Version 1.8 (Jan 14, 2013)

- JENKINS-15472 Create the page in Confluence at build-time if the page didn't already exist.

Version 1.7.1 (Jun 25, 2012)

- JENKINS-14205 Fix logic error checking for "start" token for the "Between Markers" editor.
- *Note*: The 1.7 tag failed to release properly from the Maven Release plugin, which is why this version is actually released as "1.7.1".

Version 1.6 (May 29, 2012)

- JENKINS-13896 Fix StringIndexOutOfBoundsException when the "between" ending marker exists in multiple locations.

Version 1.5 (Apr 27, 2012)

- JENKINS-13569 Fix NPE when "Attach archived artifacts" is enabled, but the job's "Archive the artifacts" option is **not** enabled.

Version 1.4 (Jan 11, 2012)

- JENKINS-12253 Introduces "even if unstable" option in the job config.
- JENKINS-12253 Also introduces a ${BUILD_RESULT} build-time environment variable.
- JENKINS-12254 Allow Space and Page names to be specified by build-time environment variables (e.g., dynamic from a parameterized build).

Version 1.3 (Oct 17, 2011)

- JENKINS-11276 Fixes content editing in Confluence 4.0.
- Adds a new "Replace entire page" editor, which was previously only possible with the "Replace between tokens" editor, and placing the start/end tokens at the top/bottom of the page content.

Version 1.2 (Oct 16, 2011)

- JENKINS-11276 Fixes file attachments when working against a Confluence 4.0 server. Also fails gracefully if content editors are configured for a 4.0 server. Both features still work properly on Confluence 3.x, but content editing is disabled for version 4.0+.

Version 1.1.1 (Sep 22, 2011)

- Fix a Content-Type issue that prevented attachments to Confluence, by defaulting to application/octet-stream if unable to figure out the content type automatically.

Version 1.1 (Jul 2, 2011)

- Implements Confluence page wiki markup editing.

Version 1.0.3 (Jun 19, 2011)

- Fix bug introduced in 1.0.2 for Confluence installations hosted at the root (e.g., http://confluence.example.com/). 1.0.2 works for non-root installations (e.g., http://www.example.com/confluence/). This release should fix both cases.

Version 1.0.2 (Jun 18, 2011)

- **Bad Release!** See version 1.0.3
- GH-1 - Fixing incorrect confluence RPC URL creation in Utils.

Version 1.0.1 (Mar 24, 2011)

- Rerelease 1.0.0 to properly set required Jenkins version

Version 1.0.0 (Feb 28, 2011)

- Initial release