

Same-space publishing

Overview

With same-space publishing, each piece of content (page or blog post) can have concurrent draft and approved versions.



The same-space publishing capability is aimed at organisations using Confluence internally.

Activating same-space publishing

To use same-space publishing, just add the `final=true` parameter to one of the `{state}` macros in your workflow – this marks it as the published state.

The addition of the `final=true` parameter changes the behavior of Confluence to show a user different page versions depending on the Confluence permissions of the user.

The workflow published version of the content is the version created when a transition to the `final` state takes place.

Editing of content in the `final` state will create a new version *but the published version will remain as the version created on transition into this state*.

This is why it is good practice to transition out of the `final` state if the content is changed, for example by adding an `updated` transition to the `final` state.

```
{workflow:name=In-space Publishing}
  {state:Editing|submit=Review}
  {state}
  {state:
Review|approved=Published|rejected=Editing}
  {approval:Review|assignable=true}
  {state}
  {state:Published|final=true|updated=Editing}
  {state}
{workflow}
```



Subsequent drafts

The `updated` parameter can be applied to published states to ensure that any subsequent edits are pushed to a draft state.

In the example markup above, editing the published state content will result in the new version being in the editing state.

As the published version is created on a transition into the `final` state, if the `final` state content is edited and no transition occurs (for example, there is no `updated` transition present in the state) the version created will NOT BE the published version even though the content remains in the `final` state.

State colours

By default, draft (unpublished) states will have an  orange circle and published (`final=true`) state will have a  green circle.

For more information, see [Status Indicator Circles](#).

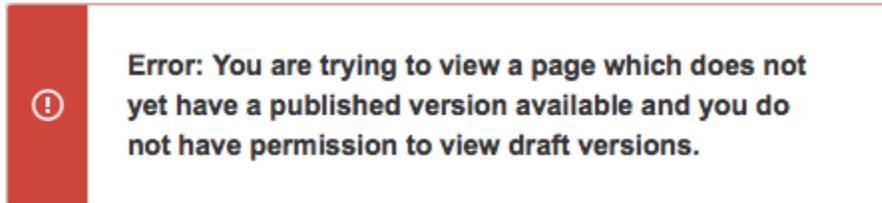
Note: Since v5.0.8, colours can be customized on state-by-state basis using workflow builder or via the editing the markup `color` parameter on the `{state}` macro.

Permissions

If a workflow has a published (`final=true`) state, then all other states in that workflow will be considered draft (unpublished) states.

-  **Editing** – draft
-  **Review** – draft
-  **Published** – published

By default, users who only have [View content permission](#) can only see the most recently published version of each page or blog post. If they try to view draft (unpublished) versions, they will see an error like this



Users with [Edit and Admin permissions](#) can see all versions of content, and can [switch between](#) the latest draft and published versions via the [Workflow Status Bar](#).

Note that the [URL for the published content](#) in a **final** state will be different to the URL for the content in the draft state.



Not for secure publishing

Same-space publishing is for when you want to direct users to a version of the content that is not currently in draft.



the user will not be directed in all cases to the content version that is not currently in draft - for example, if viewing on a mobile device



it is not a secure publishing option to hide draft content

While view only users might default to seeing the approved content the most up to date content is still visible through the page history or search index.

In addition, some mobile-based browsers can only display the latest version.

If there is a formal requirement (compliance; organizational policy) that draft updates are not visible to some users we recommend using [Different-space publishing](#) where Confluence permissions can hide the draft space content entirely.

Notifications

The `{set-message}` macro can differentiate between draft and published versions of content.

PDF and Word export

When exporting content as PDF or Word using native Confluence features, either as [individual pages](#) or an [entire space](#), the distinction between the **published** content vs. the **draft** content will be respected.

If the **final** state content is edited but there is no transition

- the content version in the **final** state will change
- the workflow published content will remain as the version created *when the last transition to the final state occurred*

To ensure that the current content in the **final** state is the workflow published version you should include an [updated transition](#) in the final state.

For more information, see: [Confluence PDF and Word Export](#).

App configuration

| Setting | Where | Notes |
|---------|-------|-------|
|---------|-------|-------|

| | | |
|--|---|---|
| Workflow Activity and Drafts Visibility | <ul style="list-style-type: none"> • Configuration - Space Tools • Configuration - Global | Determines which users can access the workflow information and tools on a page. If using same space publishing the users not in this group will see the approved version of the content by default. |
| Default View | <ul style="list-style-type: none"> • Configuration - Space Tools | Determines whether Published content is seen by default. |
| Published View Macro Transform | <ul style="list-style-type: none"> • Configuration - Space Tools • Configuration - Global | Determines whether <code>{include}</code> and <code>{excerpt-include}</code> macros display most recently Published version or, regardless of workflow state, the latest version of content.* |

* When using other macros including third party apps such as [ServiceRocket Scaffolding app](#) the data will be always be rendered from the latest version of the source content with the applied workflow irrespective of the **Published View Macro Transform** configuration. The only macros that display their `final` version content are the [Gliffy For Confluence macro](#) and the Confluence `excerpt-include` and `excerpt` macros.

See also

- [Different-space publishing](#)
- [Remote-space publishing](#)

[◀ Back to Publishing](#)