

# approval macro

[Where to use it?](#) | [Parameters](#) | [Reviewers / Assignees](#) | [User Interface](#) | [Deprecated Parameters](#) | [App configuration](#) | [Example](#) | [Events](#) | [Related macros](#) | [All examples](#) | [See also](#)

## Overview

Use the `{approval}` macro to add content-focussed reviews to workflow states.

Content has to be approved before it can progress to the next state, or it can be rejected to send it back for more edits. Once all reviewers agree – either on Approve or Reject – the workflow will transition to the state defined by the `approved` or `rejected` parameter on the parent `{state}` macro.

## Where to use it?

Optional.

One or more `{approval}` macros can be put in a `{state}` macro.

Within a `{state}`, each `{approval}` must be uniquely named. Can not be used if the `{state}` has a `{state-selection}` macro.

## Parameters

Parameter	Required	Default	Notes	Ver
<i>unnamed first parameter</i>			Name of the content review. <ul style="list-style-type: none"><li>Within any <code>{state}</code> macro, each <code>{approval}</code> macro must be given a unique name</li><li>The name can use any character set supported by your Confluence server – except some <a href="#">reserved characters</a>.<ul style="list-style-type: none"><li><a href="#">Adding Multiple Reviews</a></li></ul></li></ul>	
<code>description</code>			The description for the approvals: <ul style="list-style-type: none"><li>Plain text</li><li>Can use any character set supported by your Confluence server – except some <a href="#">reserved characters</a></li></ul>	5.7+
<code>label</code>			Requires that the content has the specified label for the approval to be active. <ul style="list-style-type: none"><li>only one label can be used</li><li>the label cannot contain <code>!, #, &amp;, (, ), *, ,, , , ;, &lt;, &gt;, ?, @, [, ], ^</code></li><li>the label cannot be negated</li></ul> If the content does not have the specified label: <ul style="list-style-type: none"><li>the Approve/Reject buttons are disabled.</li><li>reviewers cannot be manually assigned to the approval if the approval includes <code>assignable=true</code></li><li>if the approval is one of several approvals in a state (<a href="#">multiple reviews</a>) it is not required for the transition</li></ul> If the addition of the specified label is a requirement for the the approval to be undertaken then use the <code>hasLabel</code> condition instead. See: <a href="#">Conditions</a> .	
<a href="#">Conditions</a>			Optionally define additional requirements which must be met prior to content approval – see: <a href="#">Conditions</a> . If one or more conditions are not met, the Approve/Reject buttons will be disabled. However, reviewer assignment will still be possible. <ul style="list-style-type: none"><li><a href="#">Adding Multiple Reviews</a></li><li><a href="#">Communardo Metadata for Confluence app</a></li><li><a href="#">Make tasks disable a review</a></li></ul>	

## Reviewers / Assignees

Specify if, who and how reviewers are assigned to the content review.

<sup>†</sup> Parameters `selectedapprover`, `selectedapprovers` and `(group and/or user)` are mutually exclusive. You can use the parameters `group` and `user` together.

<sup>‡</sup> Prefixing the list with `&` (ampersand) will auto-assign all users and require them to review (this feature is incompatible with the `assignable` parameter).

If there are 15 or fewer possible assignees (as specified by the `user`, `group`, `selectedapprover` or `selectedapprovers` parameters, then they will all be listed by default on the [Assigning reviewers](#) screen without the need to perform a search.

The default number (15) of possible assignees listed on the [Assigning reviewers](#) screen can be changed in [Advanced Global Configuration](#).

<b>assignable</b>	false	<p>Can the user assign reviewers?</p> <ul style="list-style-type: none"> <li><b>true</b> – Yes, they can assign reviewers</li> <li><b>false</b> – No</li> </ul> <p>Note: Assignees must have permission to edit the content and also be member of <a href="#">confluence-users group</a> as a minimum. Other parameters relating to reviewers may enforce further restrictions.</p> <ul style="list-style-type: none"> <li><a href="#">Adding Multiple Reviews</a></li> <li><a href="#">Advanced different-space publishing</a></li> <li><a href="#">Advanced remote-space publishing</a></li> <li><a href="#">Assignment Examples</a></li> <li><a href="#">State expiry dates</a></li> </ul>	3.2+
<b>credentials</b>	0	<p>To what extent do reviewers need to confirm their identity when reviewing?</p> <ul style="list-style-type: none"> <li><b>0</b> – <b>must be logged in</b> (member of <a href="#">confluence-users group</a>)</li> <li><b>1</b> – additionally, must confirm <b>Password / Signing Token</b></li> <li><b>2</b> – additionally, must confirm <b>Username</b></li> </ul> <p>The requirements stack, so a value of 2 would mean: Logged in + Password / Signing Token + Username.</p> <p>The method of authentication can be configured globally. The options above for the credential password can be set to be based on either:</p> <ul style="list-style-type: none"> <li>a user's Confluence credentials</li> <li>a signing token generated by a third-party app.</li> </ul> <ul style="list-style-type: none"> <li><a href="#">Reviewer Authentication</a></li> <li><a href="#">E-Signatures</a></li> <li><a href="#">E-Signatures Configuration - Global</a></li> </ul>	
<b>exclude</b>		<p>List users who are not allowed to review.</p> <ul style="list-style-type: none"> <li>A comma-separated list of usernames.</li> <li>Supports <a href="#">Value References</a> (for example: <code>@author@</code> to prevent content author from reviewing)</li> </ul> <p>Note: While excluded users are not able to be assigned or Approve/Reject, they can still assign other users as reviewers where applicable (eg. if the <code>assignable=true</code> is used).</p> <ul style="list-style-type: none"> <li><a href="#">Assignment Examples</a></li> </ul>	4.16+
<b>group<sup>†</sup></b>		<p>Reviewers must be a member of the specified user group(s):</p> <ul style="list-style-type: none"> <li>A comma-separated list of group names<sup>†</sup></li> <li>Supports <a href="#">Value References</a></li> </ul> <p>Note: Users who do not have <a href="#">Edit content permission</a> will not be allowed to review.</p> <ul style="list-style-type: none"> <li><a href="#">Adding Multiple Reviews</a></li> <li><a href="#">Assignment Examples</a></li> </ul>	
<b>minimum</b>	0 	<p>Define the minimum number of reviewers required for the content review to be approved.</p> <ul style="list-style-type: none"> <li>Default is at least one user must approve</li> <li>If you need more users, then specify number 2</li> <li>Supports <a href="#">Value References</a> (v4.18+)</li> </ul> <p>If more than the minimum number of users are assigned as reviewers then all assignees must undertake and agree the approval.</p> <ul style="list-style-type: none"> <li><a href="#">Adding Multiple Reviews</a></li> <li><a href="#">Assignment Examples</a></li> <li><a href="#">Fast-tracked Rejections and Approvals</a></li> </ul> <p> If using the <code>selectedapprover</code> or <code>selectedapprovers</code> parameters, the <code>minimum</code> defaults to 1.</p>	
<b>rememberassignees</b>	false	<p>Remember the assignees for reviews with this name?</p> <ul style="list-style-type: none"> <li><b>true</b> – any subsequent reviews in the workflow that have the same name as this review (as defined by <code>unnamed first parameter</code>) will automatically have the same reviewers assigned ("sticky assignees")</li> <li><b>false</b> – don't remember assignees</li> </ul> <ul style="list-style-type: none"> <li><a href="#">State expiry dates</a></li> </ul>	4.14+
<b>roles</b>	false	<p>Enable approval roles feature</p> <ul style="list-style-type: none"> <li><b>true</b> – <a href="#">approval roles</a> are enabled, users assigning a review will be able to record the role that the reviewer is performing</li> <li><b>false</b> – approval roles are disabled</li> </ul>	6.4+

<code>selectedapprover<sup>†</sup></code>			<p>A single reviewer must be assigned from the list:</p> <ul style="list-style-type: none"> <li>• A comma-separated list of usernames and/or group names</li> <li>• Group names are treated as "a list of usernames"</li> <li>• Supports <a href="#">Value References</a></li> </ul> <p>Note: Users who do not have <a href="#">Edit content permission</a> will not be allowed to review.</p> <ul style="list-style-type: none"> <li>• <a href="#">Assignment Examples</a></li> </ul>	
<code>selectedapprovers<sup>†</sup></code>			<p>One or more reviewers must be assigned from the list:</p> <ul style="list-style-type: none"> <li>• A comma-separated list of usernames and/or group names<sup>†</sup></li> <li>• Group names are treated as "a list of usernames"</li> <li>• Supports <a href="#">Value References</a></li> </ul> <p>Note: Users who do not have <a href="#">Edit content permission</a> will not be allowed to review.</p> <ul style="list-style-type: none"> <li>• <a href="#">Assignment Examples</a></li> </ul>	
<code>user<sup>†</sup></code>			<p>Reviewers must be in the specified list:</p> <ul style="list-style-type: none"> <li>• A comma-separated list of usernames<sup>†</sup></li> <li>• Supports <a href="#">Value References</a></li> </ul> <p>Note: Users who do not have <a href="#">Edit content permission</a> will not be allowed to review.</p> <ul style="list-style-type: none"> <li>• <a href="#">Adding Multiple Reviews</a></li> <li>• <a href="#">Assignment Examples</a></li> <li>• <a href="#">Communardo Metadata for Confluence app</a></li> </ul>	
<code>allowedassigners</code>			<p>Assigners must be in the specified list:</p> <ul style="list-style-type: none"> <li>• A comma-separated list of usernames</li> <li>• Supports <a href="#">Value References</a></li> </ul> <p>Note</p> <ul style="list-style-type: none"> <li>• Users who do not have <a href="#">Edit content permission</a> will not be allowed to assign. It can be used along with the <code>allowedassigngroups</code> parameter.</li> <li>• Adding this parameter does not disable the review for users who have <a href="#">Edit content permission</a> until a reviewer has been assigned. To prevent the review from being undertaken before assignment of a reviewer, use this parameter in conjunction with one of <code>selectedapprover</code> or <code>selectedapprovers</code> parameters.</li> </ul> <ul style="list-style-type: none"> <li>• <a href="#">Assignment Examples</a></li> </ul>	
<code>allowedassigngroups</code>			<p>Assigners must be a member of the specified user group(s):</p> <ul style="list-style-type: none"> <li>• A comma-separated list of group names</li> <li>• Supports <a href="#">Value References</a></li> </ul> <p>Note</p> <ul style="list-style-type: none"> <li>• Users who do not have <a href="#">Edit content permission</a> will not be allowed to assign. It can be used along with the <code>allowedassigngroups</code> parameter.</li> <li>• Adding this parameter does not disable the review for users who have <a href="#">Edit content permission</a> until a reviewer has been assigned. To prevent the review from being undertaken before assignment of a reviewer, use this parameter in conjunction with one of <code>selectedapprover</code> or <code>selectedapprovers</code> parameters.</li> </ul> <ul style="list-style-type: none"> <li>• <a href="#">Assignment Examples</a></li> </ul>	

## User Interface

These parameters are used to customise the user interface of the [Workflow Popup](#).

<code>approvelabel</code>	Approve	<p>Set the caption of the Approve button.</p> <ul style="list-style-type: none"> <li>• <a href="#">Advanced different-space publishing</a></li> <li>• <a href="#">Advanced remote-space publishing</a></li> <li>• <a href="#">Customise Approval Buttons</a></li> </ul>	5.1+
<code>rejectlabel</code>	Reject	<p>Set the caption of the Reject button.</p> <ul style="list-style-type: none"> <li>• <a href="#">Advanced different-space publishing</a></li> <li>• <a href="#">Advanced remote-space publishing</a></li> <li>• <a href="#">Customise Approval Buttons</a></li> </ul>	5.1+

<code>weight</code>		40	<p>When using multiple <code>{approval}</code> macros inside a single <code>{state}</code>, you can use this parameter to set the <i>initial</i> display order.</p> <ul style="list-style-type: none"> <li>• Number 1</li> <li>• Valid values in range of 1 - 2<sup>32</sup></li> <li>• For best results, use multiples of 40.</li> <li>• Items of the same weight will be ordered alphabetically A-Z</li> <li>• Completed reviews will bubble to the top of the list, in order of completion</li> </ul> <ul style="list-style-type: none"> <li>• <a href="#">Adding Multiple Reviews</a></li> </ul>
---------------------	--	----	--

## Deprecated Parameters

The following parameters should no longer be used.

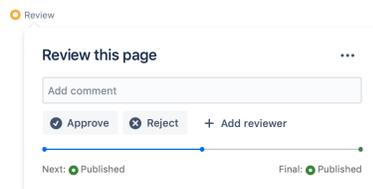
<code>final</code>		<b>PUBLISHED</b>	Use the <code>final</code> parameter on the published <code>{state}</code> macro instead – see <a href="#">Publishing</a> .
--------------------	--	------------------	---

## App configuration

Setting	Where	Notes
<b>Approval – Assignee</b>	<ul style="list-style-type: none"> <li>• <a href="#">Notifications - Space Tools</a></li> <li>• <a href="#">Notifications - Global</a></li> </ul>	While enabled, assignees will receive email notifications about events during the content review.
<b>Approval Comment</b>	<ul style="list-style-type: none"> <li>• <a href="#">Configuration - Global</a></li> </ul>	While enabled, reviewers will be forced to comment when they Approve or Reject during content reviews.
<b>Page Update Reset Approval</b>	<ul style="list-style-type: none"> <li>• <a href="#">Configuration - Space Tools</a></li> <li>• <a href="#">Configuration - Global</a></li> </ul>	While enabled, if the content is edited whilst it is under active review then any approvals made during that active review will be reset.
<b>Workflow Activity and Drafts Visibility</b>	<ul style="list-style-type: none"> <li>• <a href="#">Configuration - Space Tools</a></li> <li>• <a href="#">Configuration - Global</a></li> </ul>	Who can see and interact with the <a href="#">Workflow Popup</a>

## Example

```
{workflow:name=Example}
  {state:
Editing|submit=Review|colour=#ffa000}
  {state}
  {state:
Review|approved=Published|rejected=Editing}
  {approval:Review this page|assignable=true|roles=true}
  {state}
  {state:
Published|final=true|updated=Editing}
  {state}
{workflow}
```

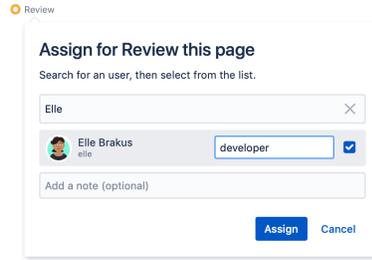


When `roles=true` is added to the approval macro, users assigned to the content review can be assigned a role.

A role can be created when adding the assignee or chosen from a dropdown menu listing of previously added roles for other assignees.

Roles for assignees can be enabled editing each individual approval in workflow builder.

- Enable Roles**  
Allows to add roles to reviewers



## Events

Content reviews generate the following [Events](#):

- `approvalunassigned`
- `newsapproved`
- `newsapprovalassigned`
- `newsrejected`
- `pageapproved`
- `pageapprovalassigned`
- `pagerejected`

## Related macros

The following [Macros](#) are related to [Reviews](#):

- `{approve-page}`
- `{approve-children}`
- `{reject-page}`
- `{reject-children}`
- `{state}` – `approved` and `rejected` parameters

## All examples

- [Adding Multiple Reviews](#) — Add multiple reviews to a content review, set assignee requirements and review dependencies
- [Advanced different-space publishing](#)
- [Advanced remote-space publishing](#)
- [Assignment Examples](#) — Define who can take part in, or be assigned to, a content review.
- [Blog Post Events](#) — Workflow events associated with blog posts
- [Communardo Metadata for Confluence app](#) — Using the Communardo metadata as value references in a workflow - including use in a custom condition for a trigger macro
- [Customise Approval Buttons](#) — How to change captions of the Approve and Reject buttons created by the `{approval}` macro.
- [Fast-tracked Rejections and Approvals](#) — How to require everyone to Approve, but only need one person to Reject
- [Make tasks disable a review](#) — Two ways to use tasks to disable a review. The first way uses a precursor task state where the tasks can be set and completed. The second example uses a [Value Reference](#) as a [condition](#) to check that all tasks are complete. For the [value reference](#) method, you will **need** the [Reporting for Confluence app](#) in addition to Comala Document Management.
- [Reviewer Authentication](#) — This example shows how to authenticate reviewers during a content review.
- [State expiry dates](#) — Using state expiry dates, defining them with metadata, and making them editable

## See also

### [Workflow Authoring Guide](#)

- [Reviews](#)
- [States](#)
- [Transitions](#)
- [Value References](#) – [Attribute References](#), [Event references](#)

### [User Guide](#)

- [Content reviews](#)

### [Reporting Guide](#)

- [Workflow Report Macro - Approvals](#)

## Configuration - Global

- [Advanced Configuration - Global](#)