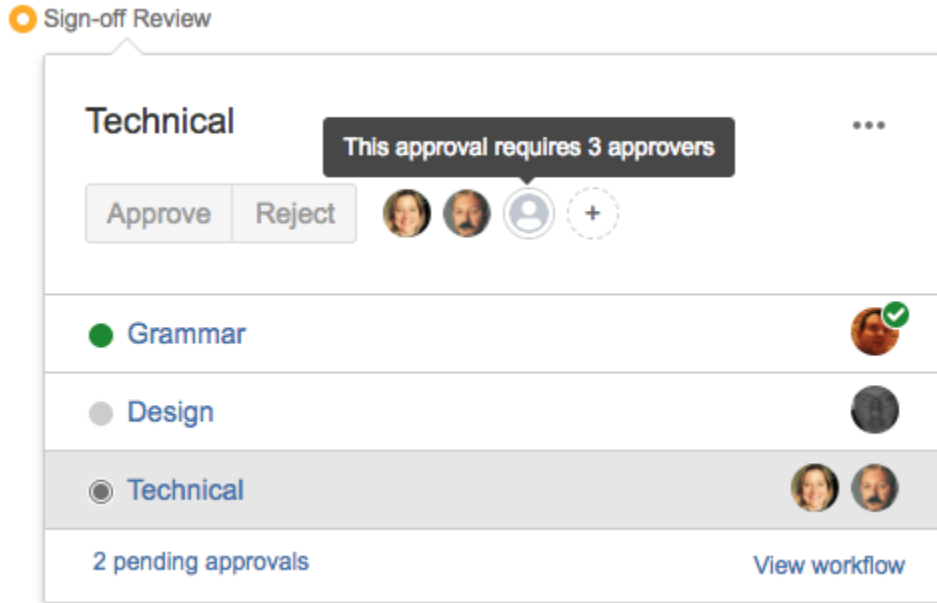


# Adding Multiple Reviews

## Overview

This example illustrates how to add multiple review groups in a single content review, set the minimum number of reviewers for each review, and also how to add dependencies between reviews.



## Parallel reviews

To add multiple reviews, put more than one `{approval}` macro in to a `{state}` macro body.

For example, to create the Grammar, Design and Technical review groups show in the screenshot above, the following markup is used:

```
{workflow:name=Parallel review}
  {state:Editing|submit=Sign-off Review}
  {state}
  {state:Sign-off Review|approved=Published|rejected=Editing}
    {approval:Technical}
    {approval:Grammar}
    {approval:Design}
  {state}
  {state:Published}
  {state}
{workflow}
```

You'll notice that the order in the [Workflow Popup](#) is often different to that specified in the workflow markup – we'll look at that in a moment.

Each of these reviews can be done in parallel, at the same time, and currently only one reviewer is required to complete each of the reviews. As soon as all the reviews agree to Approve or Reject, the transition to the relevant workflow state will occur.

## Reviewer settings

Each of the `{approval}` macros can use all the same features you'd expect from a basic content review. So, let's add some additional requirements on who can review, and how many reviewers are required:

```

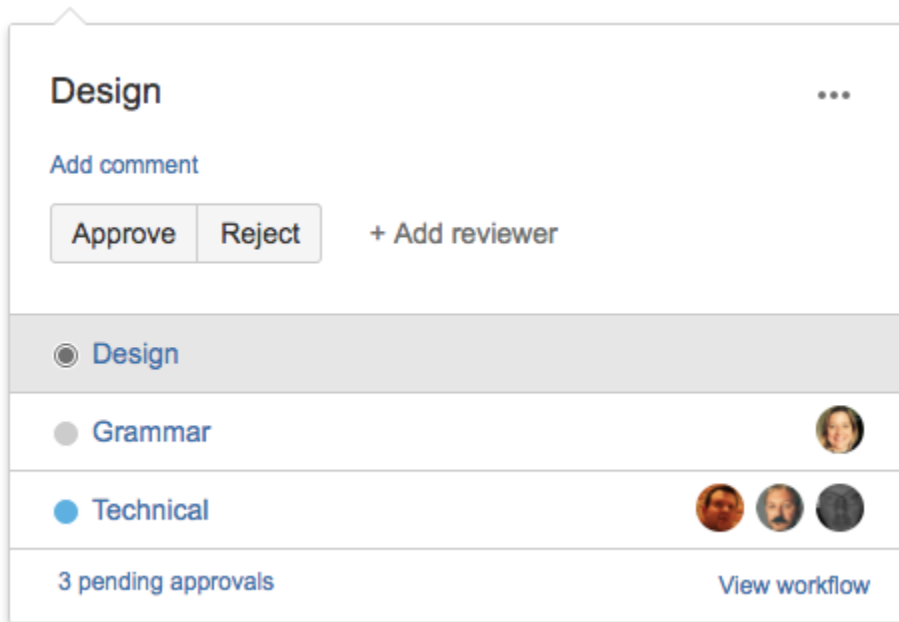
{workflow:name=Parallel review}
  {state:Editing|submit=Sign-off Review}
  {state}
  {state:Sign-off Review|approved=Published|rejected=Editing}
    {approval:Technical|group=tech-team|minimum=3}
    {approval:Grammar|user=johndoe,janedoe}
    {approval:Design|assignable=true}
  {state}
  {state:Published}
  {state}
{workflow}

```

We've used the **group** parameter to limit the Technical review to members of the **tech-team** Confluence user group. We've used the **minimum** parameter to indicate that at least 3 people are required to complete the review.

For the Grammar review, we've used the **user** parameter to name two individuals who are allowed to perform that review; as there's no other settings only one of them needs to complete the review, but both of them could still review if they wanted to.

And, for the Design review, we've used the **assignable=true** parameter to allow any Confluence user (members of the **confluence-users** user group) to be added to the review.



As you can see, the "+ Add Reviewer" link is added next to the Approve/Reject buttons.

## Review dependencies

There is often an order in which reviews must be performed, either due to process or compliance reasons, or simply to minimise reviewer workload. For example, you might want to delay the Design review until the Technical review is complete, as any technical errors in the content would likely need fixing prior to the Design review.

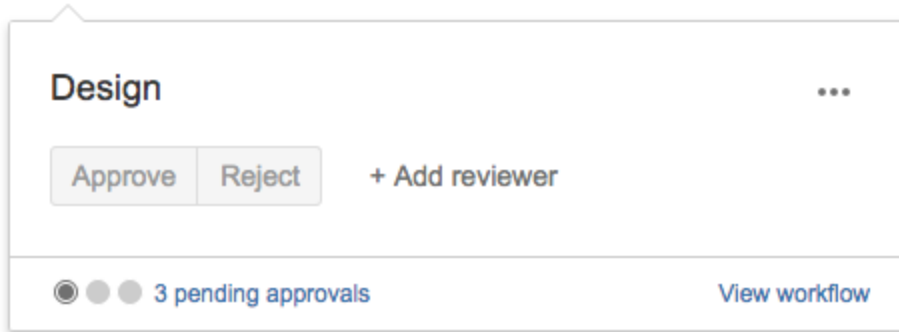
To achieve this, we can add the **hasapproval Condition** to the Design approval, like so:

```

{workflow:name=Parallel review}
  {state:Editing|submit=Sign-off Review}
  {state}
  {state:Sign-off Review|approved=Published|rejected=Editing}
    {approval:Technical|group=tech-team|minimum=3}
    {approval:Grammar|user=johndoe,janedoe}
    {approval:Design|assignable=true|hasapproval=Technical}
  {state}
  {state:Published}
  {state}
{workflow}

```

Now, if someone selects the Design review prior to the Technical review being completed, they'll see the Approve/Reject buttons are disabled:

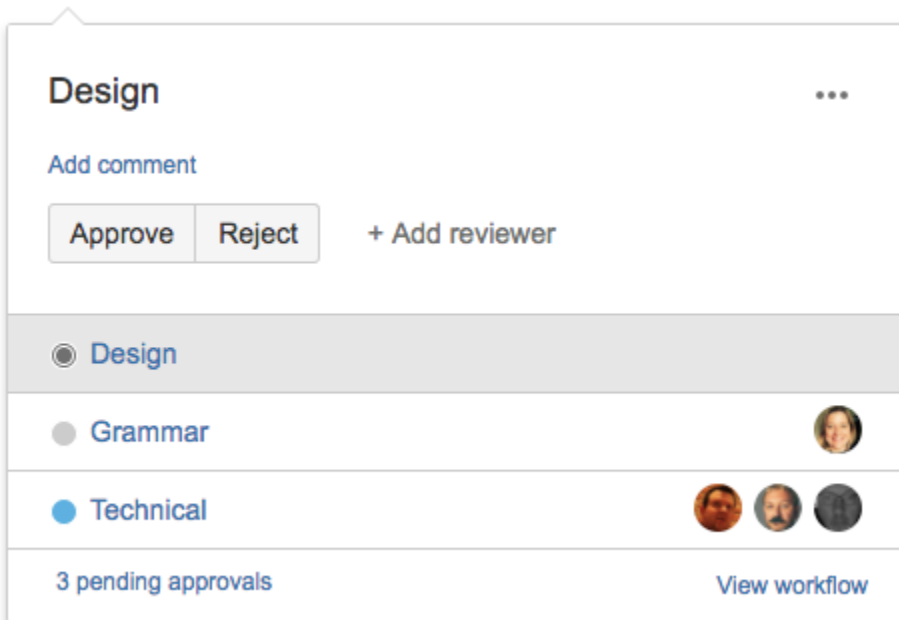


Note, however, the ability to add reviewers is still available so you don't have to wait for the Technical review to finish to specify who should participate in the Design review.

## Review display order

We mentioned earlier that the display order isn't set by the order of the `{approval}` macros in your workflow markup...

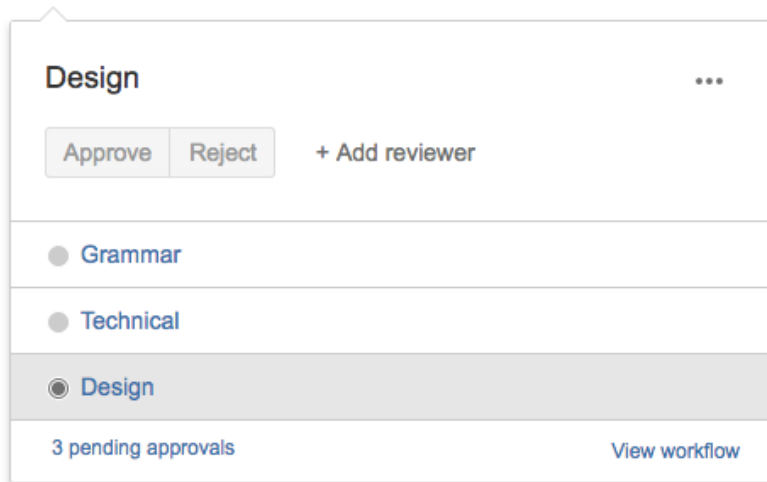
Initially, the reviews will be sorted alphabetically, even if some reviews are dependent on other reviews (our Design review is dependent on the Technical review). Assigning reviewers has no effect on the display order:



If you want to change the initial ordering of the list, you can use the `weight` parameter to do so:

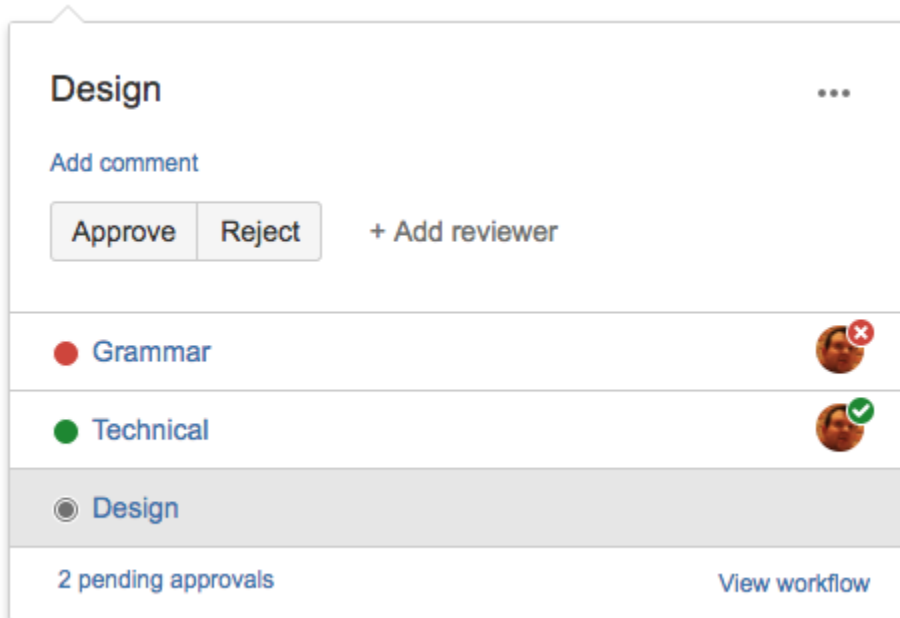
```
{workflow:name=Parallel review}
  {state:Editing|submit=Sign-off Review}
  {state}
  {state:Sign-off Review|approved=Published|rejected=Editing}
    {approval:Technical|group=tech-team|minimum=3}
    {approval:Grammar|user=johndoe, janedoe}
    {approval:Design|assignable=true|hasapproval=Technical|weight=100}
  {state}
  {state:Published}
  {state}
{workflow}
```

Reviews with heavier weights (bigger numbers) will sink to the bottom of the list, while lighter weights (smaller numbers) will float to the top of the list. In our case, we want to initially push the design review to the bottom of the list, because it's dependent on the Technical review, so we've given it a "heavy" weight of 100:



By default, reviews have a weight of 40. The reviews are sorted first by weight, then for any reviews that are the same weight, alphabetically.

As reviews are completed, regardless of the `weight` parameter, they will be pushed to the top of the list, in the order they are completed, leaving any incomplete reviews at the bottom of the list:



## See also

[Workflow Authoring Guide:](#)

- [Reviews](#)
- [Roles and Permissions](#)

[User Guide:](#)

- [Content reviews](#)

[Reporting Guide:](#)

- [Workflow Report Macro - Approvals](#)