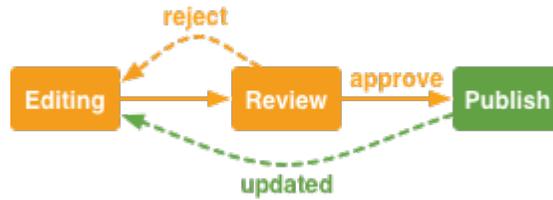


Lesson 4 - Reviews



Overview

Concepts: Reviews, Roles and Permissions

If you've completed [Lesson 3 - Transitions](#), you're probably stuck in the **Review** state of our workflow with no way out. That's because as soon as you specify a different type of transition, the default transitions are removed. But why did the `submit` transition in the **Editing** state work, yet the `approved` and `rejected` transitions in the **Review** state failed?

It's because the `submit` transition is very basic, and it automatically adds an option to transition to the target state. Review transitions, however, are a bit more advanced – in addition to specifying the target states with the `approved` and `rejected` parameters, we also need to define the review itself, and, optionally, who can take part in it.

Admin state override

As we're stuck in the **Review** state, this would be a good time to let you know about the admin state override. It will save you lots of time while developing workflows.

A **Space, Confluence or System administrator** can force a workflow in to a different state.

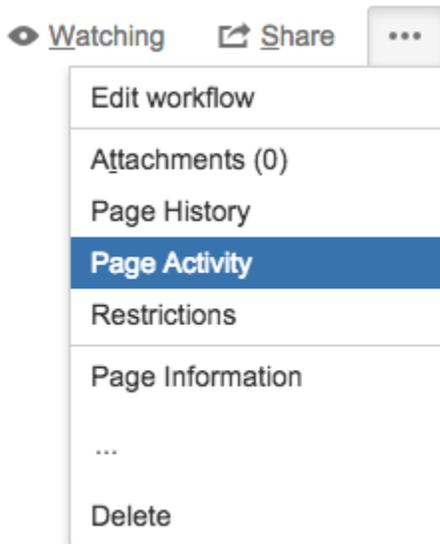
Now would be a good time to try that. You can find out how in the [Administrator state override](#) example.

Note: If you don't have administrator privileges in the space where you're testing the workflow, you can make yourself an administrator of the workflow by adding the `adminusers` parameter to the `{workflow}` macro.

Activity report

If you use the admin state override, it will be logged in the [Activity Report](#). Most interactions with the workflow, or the content it applies to (including commenting), will be tracked in the Activity Report.

If you haven't already done so, take a look now. Open the **Page Tools** menu and choose **Activity Report**.



Once you're done, choose **View Page** at the top of the screen to return to your page.

Adding a review

Now, let's fix our **Review** state. To do that we're going to need to add an `{approval}` macro inside the **Review** state:

```
{workflow:name=Simple Content Production Process}
  {state:Editing|taskable=true|submit=Review}
  {state}
  {state:Review|taskable=true|approved=Published|rejected=Editing}
    {approval:Technical Review}
  {state}
  {state:Published|final=true|taskable=true|updated=Editing}
  {state}
{workflow}
```

As you can see, we've added the `{approval}` macro, and given it a name, using its *First Unnamed Parameter*, of **Technical Review**.

More importantly, we've added **Approve** and **Reject** buttons, and it should be obvious what they will do.

Note that, unlike the other macros we've seen so far, the `{approval}` macro can't contain other macros, because it has no *Macro body*. That's why it doesn't need a closing tag.

Test it

Try testing your workflow again via the [Workflow Popup](#).

You should now see **Approve** and **Reject** buttons in the **Review** state, and clicking them should work as expected.

Also, did you notice that the heading text in the workflow popup changed to **Technical Review**?

And what about the **Published** state? Did you test that editing the page content will automatically take you back to the **Editing** state?