Verify Approval in a Jira Workflow

It's smart to make Jira workflows as simple and flexible as possible. I like to give users multiple ways to transition issues between statuses and even let them skip statuses when needed. But sometimes skipping a status is undesirable or creates a compliance problem. Consider an *approval* status for example. **You'd certainly fail an audit if work was started on an issue or an issue was completed before it was approved.** Luckily, Innovalog's Jira Misc Workflow Extensions (JMWE) app has a validator to prevent it.

Use Case

Before work is started or an issue reaches its final workflow status, make sure it passes through the "Approval" status.

Requirements

You'll need the following:

- Access: Jira application administrator permissions (to install the app) and the ability to edit workflows
- Environment: Jira Server, Jira Data Center, or Jira Cloud
- Install: Install the JMWE app from the "Find new apps" page in your Jira instance. Apply a free trial or paid license on the "Manage apps" page.

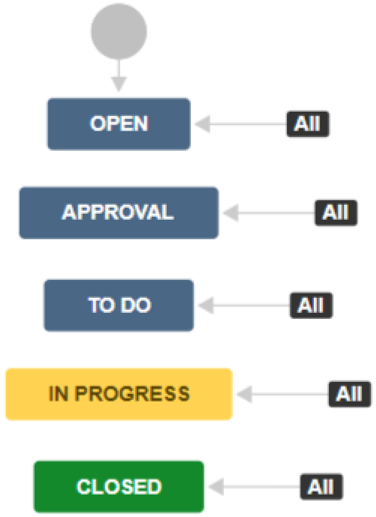
We'll use the following app features:

- Validator – Previous Status Validator

Set Up

Set up or create the following:

- Workflow: Create one simple workflow with an "Approval" status. Example: Open > Approval > To Do > In Progress > Closed
- Issues: Create one issue

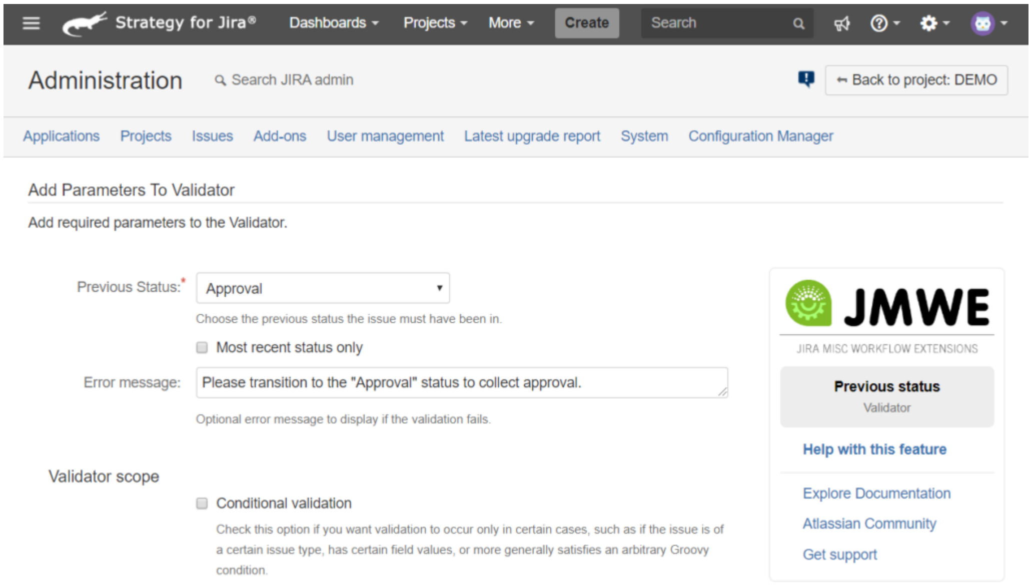


Simple workflow with an "Approval" status and global transitions

Implementation

Here's how to do it:

- Edit the workflow
- In text or diagram mode, add a validator to the "In Progress" transition
  - Select the transition leading to the "In Progress" status
  - Click the "Validators" tab (text mode) or link (diagram mode)
  - Click the "Add validator" link
  - Add the "Previous Status Validator (JMWE add-on)" validator
  - In the "Previous Status" field, select the "Approval" status
  - In the "Error message" field, enter the copy "*Please transition to the "Approval" status to collect approval.*"
  - Click the "Add" form submission button and publish the workflow



Add Parameters To Validator

Add required parameters to the Validator.

Previous Status:* Approval

Choose the previous status the issue must have been in.

□ Most recent status only

Error message: Please transition to the "Approval" status to collect approval.

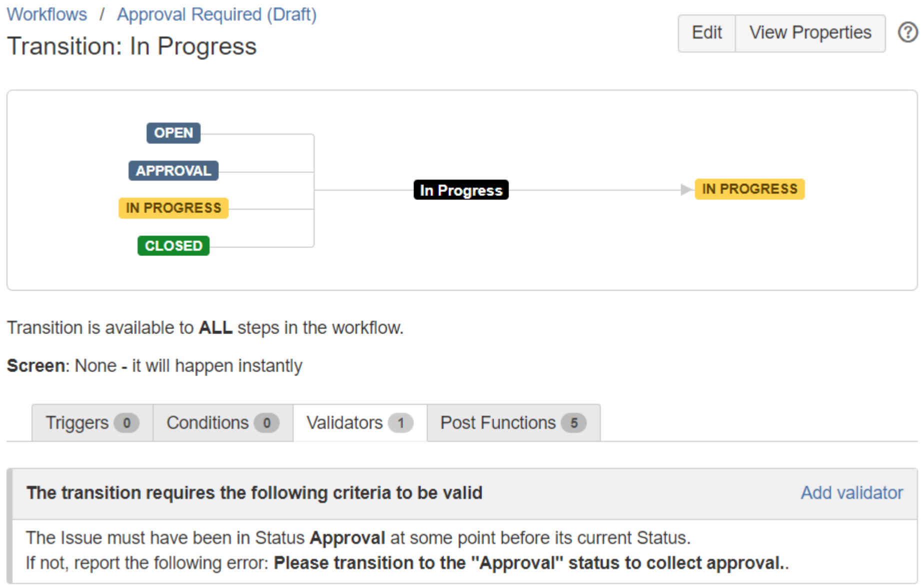
Optional error message to display if the validation fails.

Validator scope

□ Conditional validation

Check this option if you want validation to occur only in certain cases, such as if the issue is of a certain issue type, has certain field values, or more generally satisfies an arbitrary Groovy condition.

On the validator's setting page, the "Approval" status is selected and a custom error message is provided (optional).



Transition: In Progress

Transition is available to **ALL** steps in the workflow.

Screen: None - it will happen instantly

The transition requires the following criteria to be valid

The Issue must have been in Status **Approval** at some point before its current Status.
If not, report the following error: **Please transition to the "Approval" status to collect approval.**.

The "In Progress" global transition has one validator, requiring an issue to have previously transitioned through the "Approval" status.

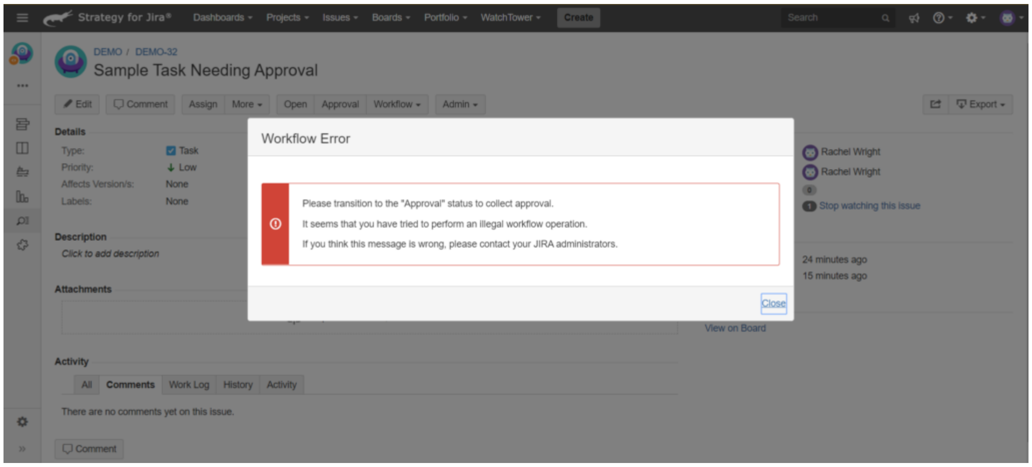
- Using the same steps above, add a "Previous Status Validator" to the "Closed" transition

With the two validators in place, issues may not skip the "Approval" workflow status. The validator checks the issue's transition history, to make sure it previously reached the "Approval" step.

Result

Test your work:

- Transition your sample issue from its initial status to the “In Progress” status
- The transition should fail and display an overlay with your custom error message



DEMO / DEMO-32

Sample Task Needing Approval

Details

Type: Task

Priority: Low

Affects Version/s: None

Labels: None

Description

Click to add description

Attachments

Activity

There are no comments yet on this issue.

Rachel Wright

Rachel Wright

0

1 Stop watching this issue

24 minutes ago

15 minutes ago

Workflow Error

Please transition to the "Approval" status to collect approval.

It seems that you have tried to perform an illegal workflow operation.

If you think this message is wrong, please contact your JIRA administrators.

A transition from the “Open” status to the “In Progress” status fails and a custom error message is displayed.

Bonus: Allow selected issues to bypass the “Approval” status

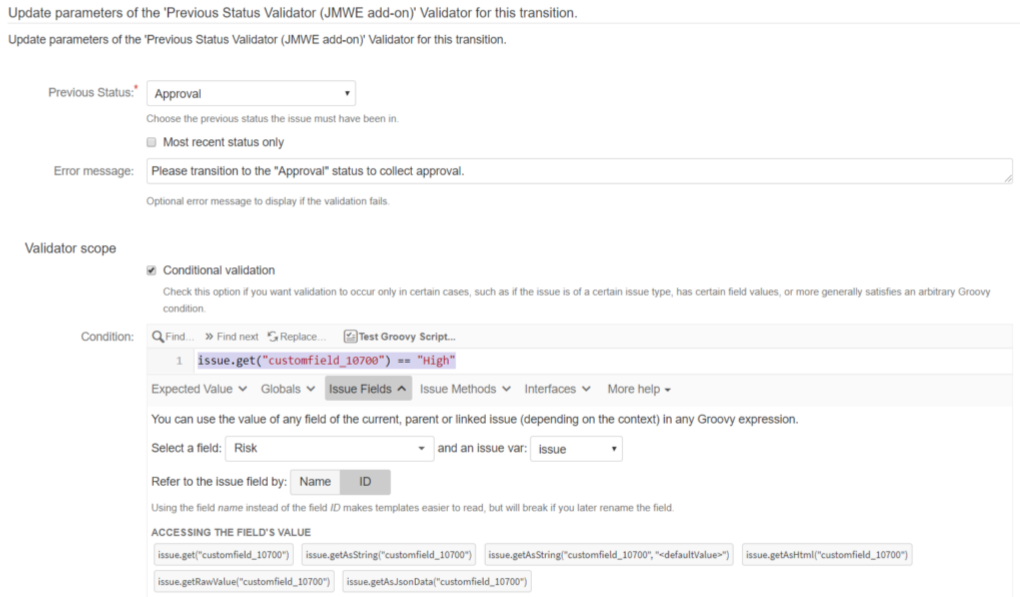
If an issue is small or low risk, you may want to conditionally bypass approval. An easy way to do this is by checking the value of a custom field before executing the transition validator.

Here’s how to do it:

- Create a custom “Select List” field called “Risk”
  - Create the selection values: “Low”, “Medium”, and “High”
- Add the custom field to your issue’s screen
- In the sample issue, set the “Risk” value to “Low” or “Medium”
- Edit one of the existing Previous Status Validators

- On the validator's settings page, click the "Conditional validation" checkbox under the "Validator scope" header
- Use the wizard to craft a simple Groovy script that checks the "Risk" field for a value of "High"
  - Click the "Issue Fields" button
  - In the "Select a field: " form field, chose the "Risk" custom field
  - Under the "ACCESSING THE FIELD'S VALUE" header, click the "*issue.get("customfield_10700") == "An option"*" button
    - Note: Your custom field ID will be different than the example
  - In the Groovy statement inserted above, change the "An option" copy to "High"
  - Click the "Update" form submission button and publish the workflow

This simple script allows issues with a Risk of "Low" or "Medium" to ignore the entire validator. Learn more about Groovy customizations here.



Update parameters of the 'Previous Status Validator (JMWE add-on)' Validator for this transition.

Update parameters of the 'Previous Status Validator (JMWE add-on)' Validator for this transition.

Previous Status:* Approval

Choose the previous status the issue must have been in.

□ Most recent status only

Error message: Please transition to the "Approval" status to collect approval.

Optional error message to display if the validation fails.

Validator scope

☑ Conditional validation

Check this option if you want validation to occur only in certain cases, such as if the issue is of a certain issue type, has certain field values, or more generally satisfies an arbitrary Groovy condition.

Condition:

```
issue.get("customfield_10700") == "High"
```

You can use the value of any field of the current, parent or linked issue (depending on the context) in any Groovy expression.

Select a field: Risk and an issue var: issue

Refer to the issue field by: Name ID

Using the field *name* instead of the field *ID* makes templates easier to read, but will break if you later rename the field.

ACCESSING THE FIELD'S VALUE

Find...
Find next
Replace...
Test Groovy Script...
Expected Value
Globals
Issue Fields
Issue Methods
Interfaces
More help
issue.get("customfield_10700")
issue.getAsString("customfield_10700")
issue.getAsString("customfield_10700", "<defaultValue>")
issue.getAsHtml("customfield_10700")
issue.getRawValue("customfield_10700")
issue.getAsJsonData("customfield_10700")

With this added condition the validator only runs if the "Risk" field's value is "High"

Q&A

Why is the "To Do" status needed in the sample workflow?

Two reasons:

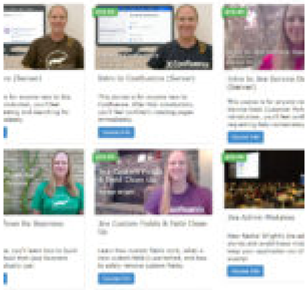
1. When an issue is approved, it doesn’t mean work automatically starts. There may be a review or assignment process that occurs before someone actually starts work on an approved issue. The “To Do” status helps signify that the issue is ready to work, but work has not yet started.
2. An issue must pass *through* the “Approval” status for the validator to function. Simply *reaching* the “Approval” status is not enough to indicate approval was collected.

Why did you use the “Previous Status *Validator*” instead of the “*Previous Status Condition*“?

Workflow conditions allow you to show or hide transitions. I wanted the “In Progress” and “Closed” transitions to display regardless of whether the issue reached the “Approval” status.

Still having trouble? Check the Jira log file, turn on error handling on the Jira Misc Workflow Extensions Global Configuration page, review the JMWE documentation, review answered questions in the Atlassian Community, or raise an Innovalog support request.

Need Workflow Help?



Take the “Jira Workflows for Business Teams” online course, get the Jira Strategy Admin Workbook, and check out the workflow materials in the Strategy for Jira store.

Related

Automatically Transition Jira Issues

June 11, 2019
In “Apps”

[System, Application & Project Admins: Who Does What in Jira?]

December 20, 2017
In “Content”

[Teaming Up to Improve Workflows]

June 10, 2019
In “Apps”

Author: Rachel Wright

Rachel Wright is an entrepreneur, Certified JIRA Administrator, and author of the JIRA Strategy Admin Workbook. She started using JIRA in 2011, became a JIRA administrator in 2013, and was certified in 2016. She is the owner and founder of Industry Templates, LLC, which helps companies grow, get organized, and develop their processes.

View all posts by Rachel Wright

Rachel Wright / January 9, 2020 / Apps / app, Innovalog, JMWE, workflows /

Strategy for Jira® / Proudly powered by WordPress