

Strategy for Jira®

Worksheets, Templates & Materials for Jira

Automatically Transition Jira Issues

As a Jira administrator, I frequently see this problem: child issues are closed but nobody remembered to close the parent issue. Users have already moved on to the next thing and the parent issue sits in its incorrect status forever. Or maybe it's the opposite: the Epic is closed, but its Stories, Bugs, and Tasks were forgotten. When issue status does not reflect reality, reporting is a nightmare. All completed or unneeded issues should be transitioned to their final status. Why not take this off your user's "to do" list and automate that step? It's easy with to automatically transition issues Innovalog's [Jira Misc Workflow Extensions](#) (JMWE) app.

Use Case

When all child issues are closed, automatically transition the parent so users don't need to do it manually. Example: When all Sub-tasks are transitioned to the "Closed" status, automatically transition the Task issue to the "Closed" 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:

- Condition – [Hide transition](#)
- Condition – [Transition parent issue](#)
- Post Function – [Comment issue](#)

Set Up

Set up or create the following:

- Issue Types: A “Standard” issue type (Example: Task) and a “Sub-Task” issue type (Example: Sub-task). These are default Jira issue types and likely already exist.
- Workflows: Create two workflows – one for each issue type. For this example, we'll use two simple, 3 step (Open > In Progress > Closed) workflows.
- Issues: Create one “Task” issue and two “Sub-task” issues

Implementation

Here's how to do it:

- Edit the workflow for the Task issue
- In diagram mode, create a new global transition
 - Click “Add transition”
 - In the “From status” field, select “Any status”
 - In the “To status” field, select “Closed”
 - In the “Name” field, enter “Auto Close”
 - In the “Screen” field, leave the selection set to “None”

Add Transition

New Transition
Reuse a transition

From status* Any status

To status* Closed

Name* Auto Close

Description Automatically transition issue to the "Closed" status when all child issues are closed.

Screen None

Add
Cancel

Add Transition Screen in Diagram Mode

Now the Task workflow has two transitions to the “Closed” status: the manual one (labeled “Close”) and the one for automation (labeled “Auto Close”).

Workflows

Task ACTIVE USED BY 1 PROJECT

For Innovalog articles

i This workflow was last edited by you at 03/May/19 1:46 PM.

Diagram
Text
Export

Step Name (id)	Linked Status	Transitions (id)	Actions
Open (1)	OPEN	Start Progress (11) >> IN PROGRESS Auto Close (41) >> Closed	View Properties
In Progress (2)	IN PROGRESS	Close (21) >> CLOSED Auto Close (41) >> Closed	View Properties
Closed (3)	CLOSED	Reopen (31) >> OPEN Auto Close (41) >> Closed	View Properties

The “In Progress” status has the normal “Close” transition and the new “Auto Close” transition.

- In the “Auto Close” transition:
 - Add the “[Hide transition \(JMWE add-on\)](#)” Condition
 - NOTE: The JMWE function hides the transition from the user but allows both Java and the REST API to trigger it *without* needing to bypass Conditions. Atlassian’s similar “Hide transition from user” function requires you to bypass *all* Conditions, which may not be desired.
 - NOTE: This transition cannot have any Screens or required field Validators.

- Add a “Sub-Task Blocking Condition” and select the “Closed” status. This prevents the parent from transitioning until all children are closed.
 - Example behavior: *All sub-tasks must have one of the following statuses to allow parent issue transitions: Closed*
- Add a “[Comment issue \(JMWE add-on\)](#)” post function
 - Enter the desired comment copy in the “Comment text” field. Example comment: *“This issue was automatically closed when all Sub-tasks reached the “Closed” status.”*
- Add the “Update Issue Field” Post Function. Set the “Resolution” field to a generic value like “Done.” Since a user isn’t manually selecting a resolution this behavior is needed to fill the field.
- If you want email notifications sent when the issue is automatically closed, go to the “Post Functions” tab and change “Fire a **Generic Event** event that can be processed by the listeners.” to “Fire a **Issue Closed** event that can be processed by the listeners.”

Workflows / Task
Transition: Auto Close View Properties ⓘ

Description: Automatically transition issue to the “Closed” status when all child issues are closed.
Transition is available to **ALL** steps in the workflow.
Screen: None - it will happen instantly

Triggers 0 Conditions 2 Validators 0 Post Functions 7

All of the following conditions Add condition

Hide transition from the user (the transition can be triggered only programmatically or from a workflow function).

All sub-tasks must have one of the following statuses to allow parent issue transitions: **CLOSED**

The “Auto Close” transition has two Conditions, seven Post Functions, and no transition Screen.

- Edit the workflow for the Sub-task issue
- Add the “[Transition parent issue \(JMWE add-on\)](#)” post function to all close-type transitions.
 - On the transition settings page, click the “Transition Picker” button to select the correct transition. Alternatively, you can type the name of the transition (Example: “Close”) or enter its ID (Example: “41”).
 - **Tip:** I prefer to enter the transition’s ID as it’s less likely to change than its label.

- Click the “Add” button then click the second “Add” button at the bottom of the page.
- Move the new post function after the function called “*Update change history for an issue and store the issue in the database.*”

Workflows / Sub-task
Transition: Close View Properties ⓘ

Screen: Resolve Issue Screen

Triggers ⓘ Conditions ⓘ Validators ⓘ Post Functions ⓘ

The following will be processed after the transition occurs	Add post function
1. Set issue status to the linked status of the destination workflow step.	
2. Add a comment to an issue if one is entered during a transition.	
3. Update change history for an issue and store the issue in the database.	
4. The first applicable transition from this list will be triggered on the parent issue: <ul style="list-style-type: none"> ◦ 41 (Auto Close) of workflow Task 	
5. Re-index an issue to keep indexes in sync with the database.	
6. Fire a Issue Closed event that can be processed by the listeners.	

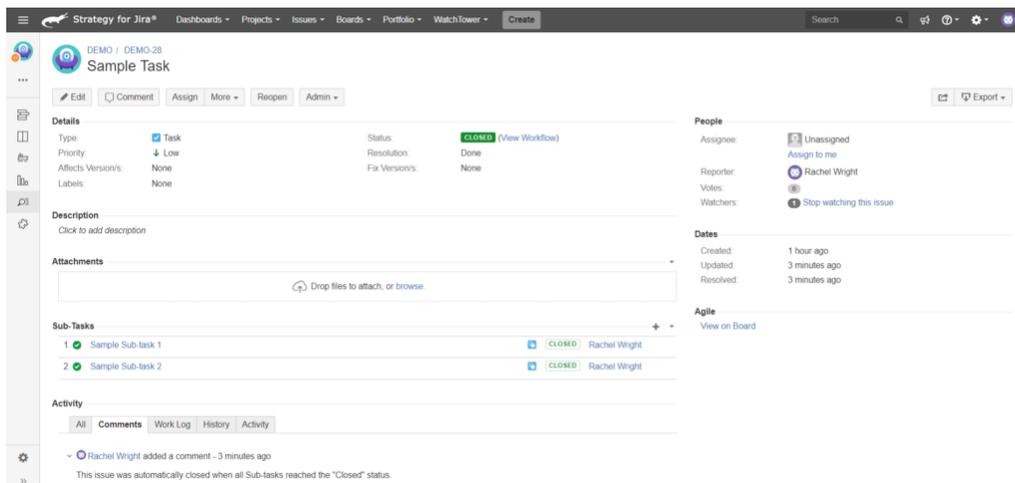
The “Transition parent issue” behavior is the forth Post Function in the “Close” transition.

Finally, publish your workflows and assign them to the “Task” and “Sub-task” issue types in your Jira project.

Result

Test your work:

- Transition one of your two Sub-task issues to its final “Closed” status
- Refresh your parent Task and verify there was no status change, no comment, and no resolution added
- Transition your second Sub-task issue to its final “Closed” status
- Refresh your parent Task. The Task should *automatically* transition to its final “Closed” status with a resolution and comment added.



The two Sub-task issues are in the “Closed” status. The parent Task issue was automatically transitioned to the “Closed” status, a Resolution of “Done” was set, and a comment was added.

Q&A

Why did you add a hidden “Auto Close” transition in the “Task” workflow?

The normal transition, labeled “Close” opens a transition screen for users to select a Resolution. I wanted to preserve a user’s ability to manually close the parent Task issue with the Resolution of their choice.

Why did you create a *global* “Auto Close” transition instead of a single transition?

I wanted to be able to transition the Task issue to the “Closed” status from any status. If you’re sure users will remember to transition the Task to the “In Progress” status, you could create a single transition, in the “In Progress” status, instead.

It’s not working as expected. What should I do?

- Make sure the “Transition parent issue” post function, in the Sub-task workflow, is positioned after the “Update change history...” behavior.
- Verify the “Transition parent issue” post function, in the Sub-task workflow, is connected to the correct “Auto Close” transition from the Task workflow.

- Make sure the “Auto Close” transition in the Task workflow does not have a transition Screen or required field Validators.

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 Users](#)” online course, get the [Jira Strategy Admin Workbook](#), and check out the workflow materials in the [Strategy for Jira store](#).

Related

[Best Practices for Creating a Custom Workflow](#)

January 8, 2018

In “Content”

[9 Tips for Getting Action in Jira](#)

July 5, 2018

In “Content”

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

December 20, 2017

In “Content”



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 / May 3, 2019 / Apps / app, Innovalog, JMWE, workflows /

Strategy for Jira® / Proudly powered by WordPress