

Main Features

Minyaa Workflows provides new capacities to JIRA, using powerful of Workflows definitions ...

- Different Workflow Functions
- A new Workflow Editor (Text mode) with the support of more OSWorkflow features,
- An enhanced Auto-Transition Management
- and more ...

Align your Workflows to your Business

Stop to be limited, and use all possibilities allowed with JIRA Workflows

Workflows Concepts

Workflows is not only Steps and Transitions, but it is a little more sophisticate.

Workflow Editor

With just a few enhancements, the default Workflow Editor can highlight some additional concept.

Auto-Transition Management

Discover Auto-Transition Management and imagine how you will be able to improve your workflows.

- [Auto-Transition Concepts](#)
- [Auto-Transition Manager](#)
- [Auto-Transition Service](#)

Permission controlled by Workflow

Based on same mechanism as Permissions on Worklog <https://doc.alkaes.fr/display/ALKMT/Permissions+on+Worklog> in Minyaa Time , your workflow may provide more sophisticated permissions <https://doc.alkaes.fr/display/ALKMW/Permission+controlled+by+Workflow> for Delete or Edit actions.

- [Is Issue Deletable](#)
- [Is Issue Editable](#)
- [And evaluated by Conditions](#)

Progress Workflow by Customfield

Workflow Conditions (Minyaa Workflows)

Some useful Workflows Conditions

- [Evaluating Issue Context](#) — Conditions based on Issue Context (Attributes, ...)
 - [Block Linked Issues](#) — Condition to block a Transition depending on Status of Linked Issue.
 - [Date Reached](#) — Condition based on Date.
 - [Project Condition](#) — Condition on a specified Project.
 - [RegExp Pattern Condition](#) — Condition based on a Pattern of Regular Expression
- [Evaluating Issue Role](#) — Conditions based Issue Roles (Assignee, Reporter, ...)
 - [Allow Leader](#) — Condition for allowing only Project Leader.
 - [Allow Reporter](#) — Condition for allowing only Issue Reporter.
 - [Assigning Conditions](#) — Conditions for allowing only if the Issue is Assigned or Not assigned.
- [Evaluating something lese](#) — Conditions used for more complex cases
 - [Activity and Unactivity Condition](#) — Conditions based on Activity or Inactivity on a Issue.
 - [Flag Condition](#) — Condition to allow only a Flag is passed.

Workflow Validators (Minyaa Workflows)

Some useful Workflows Validators

- [Fibonacci Resolution](#) — Validator based on Comment / Resolution and Fibonacci Suite
- [On Resolution Comments](#) — Validator based on Comment
- [Selected Fix Version Count Validator](#) — Validator on Nb of selected Versions (Fix or Affected)

Workflow Post-Functions (Minyaa Workflows)



Older versions ...

For previous versions of Minyaa Workflow (Minyaa Suite 3.x), please refer to [Minyaa Workflow 3.5.x](#) .

All features ...

- [Align your Workflows to your Business](#)
- [Workflows Concepts](#)
- [Workflow Editor](#)
- [Auto-Transition Management](#)
 - [Auto-Transition Concepts](#)
 - [Auto-Transition Manager](#)
 - [Auto-Transition Service](#)
- [Permission controlled by Workflow](#)
 - [Is Issue Deletable](#)
 - [Is Issue Editable](#)
 - [And evaluated by Conditions](#)
- [Progress Workflow by Customfield](#)
- [Workflow Conditions \(Minyaa Workflows\)](#)
 - [Evaluating Issue Context](#)
 - [Block Linked Issues](#)
 - [Date Reached](#)
 - [Project Condition](#)
 - [RegExp Pattern Condition](#)
 - [Evaluating Issue Role](#)
 - [Allow Leader](#)
 - [Allow Reporter](#)
 - [Assigning Conditions](#)
 - [Evaluating something lese](#)
 - [Activity and Unactivity Condition](#)
 - [Flag Condition](#)
- [Workflow Validators \(Minyaa Workflows\)](#)
 - [Fibonacci Resolution](#)
 - [On Resolution Comments](#)
 - [Selected Fix Version Count Validator](#)

Some useful Workflows Post-Functions

- [Inherit Field values from Parent Issue](#)
 - [Inherit Priority](#) — Post-Function for inherit Priority from Parent Issue
 - [Inherit Component](#) — Post-Function for inherit Component from Parent Issue
 - [Inherit Fix or Affected Versions](#) — Post-Function for inherit Fix Version or Affected Version from Parent Issue
 - [Inherit CustomFields](#) — Post-Function for inherit Customfields from Parent Issue
- [Manage Auto-Transition](#)
 - [Auto-Transition \(Issues\)](#) — Post-Function to initiate auto-transition in linked issues
 - [Auto-Transition \(SubTask\)](#) — Post-Function to transition Sub-Task
 - [Flagged Auto Transition](#) — Post-Function to initiate a transition hidden by Flag Condition
- [Propagate Post-Functions](#)
 - [Propagate Any Fields](#) — Post-Function to propagate some of System or Custom Fields from the current Issue to linked Issues.
- [Miscellaneous ...](#)
 - [Back to Previous Step](#) — Post-Function allowing a transition Back To Previous Step.
 - [Move Issue Function](#) — Post-Function allowing in a transition to Move an Issue to another Project/Issue Type .

Statistic Transition Report

To know which transition occurred on a list of issues ... Run Report the [Statistic Transition Report](#) .

Issues Validation

In previous version of JIRA, it was possible implement an Issue Validation for Edit and Create operation, but with the evolution of In Line Editing and some other, it was no more possible to support this mechanism.

- [Workflow Post-Functions \(Minyaa Workflows\)](#)
 - [Inherit Field values from Parent Issue](#)
 - [Inherit Priority](#)
 - [Inherit Component](#)
 - [Inherit Fix or Affected Versions](#)
 - [Inherit CustomFields](#)
 - [Manage Auto-Transition](#)
 - [Auto-Transition \(Issues\)](#)
 - [Auto-Transition \(SubTask\)](#)
 - [Flagged Auto Transition](#)
 - [Propagate Post-Functions](#)
 - [Propagate Any Fields](#)
 - [Miscellaneous ...](#)
 - [Back to Previous Step](#)
 - [Move Issue Function](#)