

Automatically resolve an epic when all its stories are resolved

On this page

- [Features used to implement the example](#)
- [Example: Automatically resolve an epic when all its stories are resolved](#)
- [Other examples of that functions](#)
- [Related Usage Examples](#)

Features used to implement the example

- [Write field on linked issues or sub-tasks](#)
- [Block or hide a transition for an issue depending on its issue links](#)















We add [Write field on linked issues or sub-tasks](#) to **"Resolve Issue"** transition on Story's workflow, in order to execute transition **"Resolved Issue"** on Epic issue. To do it we write the name of the transition into virtual field **"Execute transition (delayed execution)"**. We use a boolean expression in order to check that the rest of the stories are already resolved or closed.

We add [Blocking or hiding a transition depending on its issue links](#) to **"Resolve Issue"** transition in **Epic's** workflow, in order to prevent **Epic** from being manually resolved while some of its Stories are still **"Open"** or **"In Progress"**.

Example: Automatically resolve an epic when all its stories are resolved

We are going to see how to implement a mechanism in our workflows, in order to auto-transition **"Epic"** to **Resolved** status when all its **"Stories"** become **Resolved** or **Closed**. Let's see in screenshots how to configure the different features.

We add [Block or hide a transition for an issue depending on its issue links](#) to **"Resolved Issue"** transition in **"Epic's"** workflow:

Issue link types:	<div><div><input type="checkbox"/> is blocked by</div><div><input type="checkbox"/> blocks</div><div><input type="checkbox"/> is cloned by</div><div><input type="checkbox"/> clones</div><div><input type="checkbox"/> is duplicated by</div><div><input type="checkbox"/> duplicates</div><div><input type="checkbox"/> has Epic</div><div><input checked="" type="checkbox"/> is Epic of</div><div><input type="checkbox"/> relates to</div><div><input type="checkbox"/> relates to</div></div> <div>Selected issue link types will be allowed. Nevertheless, if you didn't select any issue link type, and checked "Allow unselected issue link types", there won't be applied any filter by issue link type, i.e., every issue link type will be allowed.</div>
Issue types for linked issues:	<div><div><input type="checkbox"/>  Bug</div><div><input type="checkbox"/>  Epic</div><div><input type="checkbox"/>  New Feature</div><div><input checked="" type="checkbox"/>  Story</div><div><input type="checkbox"/>  Task</div><div><input type="checkbox"/>  Improvement</div><div><input type="checkbox"/>  Sub-task</div></div> <div>Linked issues of selected issue types will be allowed. Nevertheless, if you didn't select any issue type, and checked "Allow unselected issue types", there won't be applied any filter by issue type, i.e., every issue type will be allowed.</div>
Statuses for linked issues:	<div><div><input type="checkbox"/>  Open</div><div><input type="checkbox"/>  In Progress</div><div><input type="checkbox"/>  Reopened</div><div><input checked="" type="checkbox"/>  Resolved</div><div><input checked="" type="checkbox"/>  Closed</div><div><input type="checkbox"/>  To Do</div><div><input checked="" type="checkbox"/>  Done</div></div> <div>Linked issues in selected statuses will be allowed. Nevertheless, if you didn't select any issue status, and checked "Allow unselected statuses", there won't be applied any filter by status, i.e., every issue status will be allowed.</div>

Resolutions for linked issues:	<div><input type="checkbox"/> UNRESOLVED, i.e. no resolution value</div> <div><input type="checkbox"/> Fixed</div> <div><input type="checkbox"/> Won't Fix</div> <div><input type="checkbox"/> Duplicate</div> <div><input type="checkbox"/> Incomplete</div> <div><input type="checkbox"/> Cannot Reproduce</div> <div><input type="checkbox"/> Done</div> <div><input type="checkbox"/> Won't Do</div> <div>Linked issues with selected resolutions will be allowed. Nevertheless, if you didn't select any issue resolution, and checked "Allow unselected resolutions", there won't be applied any filter by resolution, i.e., every issue resolution will be allowed.</div>
Linked issues must belong to:	<div><div><input checked="" type="radio"/> any project</div><div><input type="radio"/> current project</div><div><input type="radio"/> any but current project</div><div><input type="radio"/> projects introduced:</div></div> <div><div></div><div>Introduce a comma separated list of project keys. If you write field codes (format <code>{nnnnn}</code>) they will be replaced with its values.</div><div><input type="checkbox"/> ignore links to issues in other projects</div></div>
Filtering by field values: Optional boolean expression that should be satisfied by linked issues. (Syntax Specification)	<div><div>1</div><div>Leave field empty for no filtering. [Line 1 / Col 1]</div><div><div>Logical connectives: <code>or</code>, <code>and</code> and <code>not</code>. Alternatively you can also use <code> </code>, <code>&</code> and <code>!</code>.</div><div>Comparison operators: <code>=</code>, <code>!=</code>, <code>></code>, <code>>=</code>, <code><</code> and <code><=</code>. Operators <code>~</code>, <code>!~</code>, <code>in</code>, <code>not in</code>, <code>any in</code> and <code>none in</code> can be used with strings, multi-valued fields and lists.</div><div>Logical literals: true and false. Literal null is used with <code>"="</code> and <code>"!="</code> to check whether a field is initialized, e.g. <code>{00012} != null</code> checks whether Due Date is initialized.</div><div>String Field Code Injector:<div>Summary - [Text] - <code>{00000}</code> ▾<div>Field Code for Current IssueField Code for Linked Issues</div></div><div>Numeric/Date Field Code Injector:<div>Original estimate (minutes) - [Number] - <code>{00068}</code> ▾<div>Field Code for Current IssueField Code for Linked Issues</div></div><div>Example 1: boolean condition <code>{00012} <= ^{00012}</code> will require that linked issues have <i>Due Date</i> equal or later than current issue's <i>Due Date</i>. Example 2: boolean condition <code>%{00074} ~ ^%{00074} AND ^%{00017} in ["Blocker", "Critical"]</code> will require that linked issues have <i>Fixed versions</i> contained in current issue's <i>Fixed versions</i> and <i>Priority</i> is <i>Blocker</i> or <i>Critical</i>.</div></div></div></div></div>
Minimum required number of issue links:	<div><div>0</div><div>Minimum number of linked issues required to satisfy selected filtering conditions (issue link type, issue type, status, resolution, project and field values).</div></div>
Maximum allowed number of issue links:	<div><div>1000</div><div>Maximum number of linked issues allowed to satisfy selected filtering conditions (issue link type, issue type, status, resolution, project and field values).</div></div>

Allow unselected issue link types:	<input checked="" type="checkbox"/> Unselected issue link types will be ignored, i.e., they will be allowed regardless of linked issues' issue type, status, resolution, field values and number. Nevertheless, if none of the issue link types are selected, checking this option will make the condition behave as if you had selected every issue link type.
Allow unselected issue types:	<input checked="" type="checkbox"/> Linked issues in unselected issue types will be ignored, i.e., they will be allowed regardless of their status, resolution, field values and number. Nevertheless, if none of the issue types are selected, checking this option will make the condition behave as if you had selected every issue type.
Allow unselected statuses:	<input type="checkbox"/> Linked issues in unselected statuses will be ignored, i.e., they will be allowed regardless of their resolution, field values and number. Nevertheless, if none of the statuses are selected, checking this option will make the condition behave as if you had selected every status.
Allow unselected resolutions:	<input checked="" type="checkbox"/> Linked issues in unselected resolutions will be ignored, i.e., they will be allowed regardless of their field values and number. Nevertheless, if none of the resolutions are selected, checking this option will make the condition behave as if you had selected every resolution.
Allow unsatisfied condition on field values:	<input type="checkbox"/> Linked issues not satisfying filter by field values will be ignored, i.e., they will be allowed regardless of their number.
Message to show when validation fails:	Epic can't be resolved while it has stories not resolved. Set translations for installed languages

Once configured, transition "**Resolved Issue**" in **Epic's** workflow will look like this:

OPEN

IN PROGRESS

REOPENED

Resolve Issue

RESOLVED

Screen: [Resolve Issue Screen](#)

Triggers 0

Conditions 1

Validators 1

Post Functions 6

The transition requires the following criteria to be valid

Add validator

Validation on linked issue:

At least 0 and no more than 1000 issue links with the following characteristics:

- Issue link types: is **Epic** of.
- Linked issue's issue type: **Story**.
- Linked issue's status: **Done**, **Closed** or **Resolved**.
- Linked issue's resolution: **any**
- Linked issue belongs to **any project**.

About the rest of issue links:










- Unselected issue link types are **allowed**.
- Unselected issue types are **allowed**.
- Unselected statuses are **not allowed**.

Message to show when validation fails: **"Epic can't be resolved while it has stories not resolved."**

We add [Write field on linked issues or sub-tasks](#) post-function to "**Resolve Issue**" transition in "**Story's** workflow:















Source value that will be written into target field:	<div><div>Select a source type:</div><div><div><input type="radio"/> field in current issue</div><div><input checked="" type="radio"/> parsed text (basic mode)</div><div><input type="radio"/> parsed text (advanced mode)</div><div><input type="radio"/> math or date-time</div></div><div>expression</div><div><div>1</div><div>Resolve Issue</div></div><div>Field codes with format <code>{nnnnn}</code> will be replaced with the corresponding field values. With Cascading Select fields use <code>{nnnnn.0}</code> and <code>{nnnnn.1}</code> for referencing base level and child levels respectively.<div>Check Syntax</div></div><div><div>String Field Code Injector:</div><div>Summary - [Text] - %{00000}</div><div>Field Code for Current IssueField Code for Linked Issues / Subtasks</div></div><div><div>Numeric/Date-Time Field Code Injector:</div><div>Original estimate (minutes) - [Number] - {00068}</div><div>Field Code for Current IssueField Code for Linked Issues / Subtasks</div></div></div>
Target field that will be set in linked issues or subtasks:	<div><div>Execute transition (delayed execution) - [Workflow transition]</div><div><input type="checkbox"/> Don't overwrite target field if it's already set.</div></div>
Filtering by issue link type:	<div><div><input type="checkbox"/> is blocked by</div><div><input type="checkbox"/> blocks</div><div><input type="checkbox"/> is cloned by</div><div><input type="checkbox"/> clones</div><div><input type="checkbox"/> is duplicated by</div><div><input type="checkbox"/> duplicates</div><div><input checked="" type="checkbox"/> has Epic</div><div><input type="checkbox"/> is Epic of</div><div><input type="checkbox"/> is caused by</div><div><input type="checkbox"/> causes</div><div><input type="checkbox"/> relates to</div><div><input type="checkbox"/> relates to</div></div> <div>Only issues linked to current issue by selected issue link types will be written.</div>
Write also subtasks fulfilling condition on issue type, status and project:	<div><div><input type="checkbox"/></div><div>This option only makes sense when current issue itself is not a subtask.</div></div>
Write also sibling subtasks fulfilling condition on issue type, status and project:	<div><div><input type="checkbox"/></div><div>Sibling subtasks are understood as subtasks with the same parent as current issue. This option only makes sense when current issue is itself a subtask.</div></div>

Filtering linked issues or subtasks by issue type:

- ☒  Epic
- ☐  Story
- ☐  Test Plan
- ☐  Bug
- ☐  New Feature
- ☐  Task
- ☐  Improvement
- ☐  QA Sub-task
- ☐  Sub-task

Selected issue types will be written, but if you don't select any, it won't be applied any filter by issue type. In that case all the issue types will be written.

Filtering linked issues or subtasks by status:

- ☒  Open
- ☒  In Progress
- ☒  Reopened
- ☐  Resolved
- ☐  Closed
- ☒  To Do
- ☐  Done
- ☐  Acceptance
- ☐  Fail
- ☐  Pass
- ☐  Retest
- ☐  Active
- ☐  Inactive
- ☐  Cancelled

Selected statuses will be written, but if you don't select any, it won't be applied any filter by status. In that case issues in any status will be written.

Linked issues or subtasks belong to:

- ☒ any project
- ☐ current project
- ☐ any but current project

1

Filtering by field values:

Optional boolean expression that should be satisfied by linked issues and subtasks. (Syntax Specification)

Leave field empty for no filtering.

[Line 1 / Col 1]

Logical connectives: or, and and not. Alternatively you can also use |, & and !.

Comparison operators: =, !=, >, >=, < and <=. Operators !~, in, not in, any in and none in can be used with strings, multi-valued fields and lists.

Logical literals: true and false. Literal null is used with = and != to check whether a field is initialized, e.g. {00012} != null checks whether Due Date is initialized.

String Field Code Injector:

Summary - [Text] - %{00000} ▾

Field Code for Current Issue

Field Code for Linked Issues / Subtasks

Numeric/Date Field Code Injector:

Original estimate (minutes) - [Number] - {00068} ▾

Field Code for Current Issue

Field Code for Linked Issues / Subtasks

Example 1: {00012} <= ^{00012} will require that linked issues and subtasks have Due Date equal or later than current issue's Due Date.

Example 2: %{00074} ~ ^{00074} AND ^{00017} in ["Blocker", "Critical"] will require that linked issues and subtasks have Fixed versions contained in current issue's Fixed versions and Priority is Blocker or Critical.

Write linked issues and subtasks recursively:

☐

Issues and subtasks transitively linked will also be written, provided they fulfill stated filtering conditions. Issues are written recursively without depth limit, but each selected issue is written only once.

Conditional execution:

Optional boolean expression that should be satisfied in order to actually execute the post-function. (Syntax Specification)

1

Leave the field empty for executing the post-function unconditionally.

Collection of Examples

[Line 1 / Col 1]

Logical connectives: and, or and not. Alternatively you can also use &, | and !.

Comparison operators: =, !=, >, >=, < and <=. Operators in, not in, any in, none in, ~ and !~ can be used with strings, multi-valued fields and lists.

Logical literals: true and false. Literal null is used with = and != to check whether a field is initialized, e.g. {00012} != null checks whether Due Date is initialized.

String Field Code Injector:

Summary - [Text] - %{00000} ▾

Numeric/Date Field Code Injector:

Original estimate (minutes) - [Number] - {00068} ▾

Run as:

Select the user that will be used to execute this feature. JIRA will apply restrictions according to the permissions, project roles and groups of the selected user.

Current user ▾

User defined by a field.

Input a specific user.

Note that: if you are using Jira **7.0 or higher** with versions of **Jira Workflow Toolbox** older than Release Notes for **2.2.8**, you should input the following boolean expression in parameter **Filtering by field values**:

```
count(filterByStatus(filterByIssueType(linkedIssues("is Epic of"), "Story"), "Done, Closed, Resolved")) = count(filterByIssueType(linkedIssues("is Epic of"), "Story")) - 1
```

Once configured, transition **"Resolved Issue"** in **Story's** workflow will look like this:

OPEN

IN PROGRESS

REOPENED

Resolve Issue

RESOLVED

Screen: [Resolve Issue Screen](#)

Triggers 0

Conditions 1

Validators 0

Post Functions 6

The following will be processed after the transition occurs

Add post function

- Text parsed in **basic** mode **Resolve Issue** will be copied to field **Execute transition (delayed execution)** in linked issues or subtasks filtering by:
 - Inward issue link types: **has Epic**.
 - Outward issue link types: **none**
 - Subtasks won't be written.**
 - Sibling subtasks won't be written.**
 - Issue types: **Epic**.
 - Statuses: **To Do, Open, Reopened and In Progress**.
 - Linked issues or subtasks may belong to **any** project.
 - This feature will be run as user in field **Current user**.

This same configuration should be added to "**Close Issue**" and "**Done**" transitions in **Story's workflow**, in case direct closing is allowed for stories.

Other examples of that functions

Write field on linked issues or sub-tasks

Page: [Add and remove a single or a set of items from multi valued fields](#)
Page: [Automatically become watcher of every issue blocking an issue assigned to you](#)
Page: [Automatically close resolved sub-tasks when parent issue is closed](#)
Page: [Automatically resolve an epic when all its stories are resolved](#)
Page: [Compose dynamic text by inserting field values in a text template](#)
Page: [Copy "Due date" into a date type custom field in a linked issue if it's greater than current issue's "Due date"](#)
Page: [Copy attachments from one issue to another](#)
Page: [Create a comment in sub-tasks when parent transitions](#)
Page: [Creating a Jira Service Desk internal comment](#)
Page: [Creating a Jira Service Desk internal comment on linked issues](#)
Page: [Execute transition in epic](#)
Page: [Make linked issues, sub-tasks and JQL selected issues progress through its workflows](#)
Page: [Moving sub-tasks to "Open" status when parent issue moves to "In Progress"](#)
Page: [Sum sub-task's "Time Spent" \(work logs\) and add it to a certain linked issue](#)
Page: [Transition sub-tasks when parent is transitioned](#)

Block or hide a transition for an issue depending on its issue links

Page: [Automatically resolve an epic when all its stories are resolved](#)

Related Usage Examples

- Block or unblock a transition after an issue rested a specific time in a status
 - example
 - condition
 - validator
 - transition
- Block transition until all sub-tasks are in a specific status category
 - example
 - transition
 - condition
- Validation on sibling sub-tasks depending on issue type and status
 - example
 - validator
 - sub-task
 - transition
- Validation and condition based on time expressions
 - example
 - condition
 - validator
 - transition
- Set a condition in a global transition which only applies in a certain status
 - example
 - condition
 - transition
- Block a transition until all sub-tasks have certain fields populated

Page: Make linked issues, sub-tasks and JQL selected issues progress through its workflows

- example
 - condition
 - validator
 - sub-task
 - transition
- Block an epic's transition depending on linked issues status and due date
 - example
 - validator
 - issue-links
 - transition
- Moving story to "In Progress" when one of its sub-tasks is moved to "In Progress" (Transition issues)
 - example
 - post-function
 - transition
- Transition sub-tasks when parent is transitioned
 - example
 - post-function
 - sub-task
 - transition
 - outdated
- Transition only a sub-task among several ones
 - example
 - post-function
 - sub-task
 - transition
 - outdated
- Moving sub-tasks to "Open" status when parent issue moves to "In Progress"
 - example
 - post-function
 - sub-task
 - transition
 - outdated
- Moving story to "Ready for QA" once all its sub-tasks are in "Ready for QA" status
 - example
 - post-function
 - sub-task
 - transition
 - outdated
- Change parent's status depending on sub-task's summary (Transition issues)
 - example
 - post-function
 - transition
- Change parent's status depending on sub-task's summary
 - example
 - post-function
 - sub-task
 - transition
 - outdated
- Moving story to "In Progress" when one of its sub-tasks is moved to "In Progress"
 - example
 - post-function
 - sub-task
 - transition
 - outdated