

Triage Jira Service Desk email requests (Move issues)

On this page

- [Features used to implement the example](#)
- [Example: Triage Jira Service Desk email requests](#)
- [Other examples of that function](#)
- [Related Usage Examples](#)

Features used to implement the example

- [Move issues](#)
- [Transition issues](#)
- [Transition is triggered by Jira Workflow Toolbox post-function](#)

Example: Triage Jira Service Desk email requests

The use of email requests in Jira Service Desk has one decisive disadvantage: only one request type (and therefore only one issue type) can be configured on a per email channel or per project basis.

In this example, a triage mechanism is configured for issues created via email, that are configured as request type **Email request** and issue type **IT Help**. If the summary contains the **keyword "Incident"** the request gets moved to issue type **Incident** and request type **Email incident**.

The recommended approach consists of two steps:

Add a global reflexive transition (from any status to itself) named **Incident Triage**.

In order to hide this transition from Jira users, add a [Transition is triggered by Jira Workflow Toolbox post-function](#) condition. Additionally, add a Move issues post function, with the following options:

- Project Retain the project
- Issue type Selected issue type Incident
- Status Selected status Open **⚠ must be a valid status present in the Incident's workflow**
- Additional fields, e.g.
 - Priority Standard Blocker
 - Customer Request Type Standard **Email incident**

i Additional fields aren't required, but at least **Customer Request Type** should definitely be set to ensure customers retain request visibility.

Once configured, the transition will look like this:

The following will be processed after the transition occurs

1. Move **current issue** to

Project:

Will be **retained**

Issue type:

Task

Status:

Open

Target fields and Source values:

Target Field	Type of Value	Source Value
Priority	Standard	Blocker
Customer Request Type	Standard	Email incident (Move Issues)

This feature will be run as **Jira Workflow Toolbox**.

by JWT

All the remaining post functions in this transitions are superfluous for this example and therefore can be deleted.

Next, add a [Transition issues](#) to the Create issue transition of the workflow attached to email requests - in this example IT Help - with the following options:


- Target issue Current issue
- Execute transition Incident Triage
- Conditional execution
`%{nnnnn} = "move/b52566f1-6353-404a-98de-04e9a2a351a1" AND %{00000} ~ "Incident"`

The field code `nnnnn` has to be replaced with the corresponding one for **Customer Request Type (Customer Request Type Custom Field)**. To receive the exact value to match against, the easiest way would be to set this request type on an issue and evaluate with the help of our [Expression Parser Test](#) page.

 In the latest versions of Jira Service Desk, those request types consist of the project key, a forward slash and a hash value.

Once configured, the transition will look like this:

The following will be processed after the transition occurs
1. Creates the issue originally.
2. Transition issues <u>Target issues:</u> Current issue <u>Action to be performed:</u> Execute Transition 'Incident Triage (961)' <u>Additional configuration:</u> Post-function will only be executed if the following boolean expression is satisfied: <code>%{Customer Request Type} = "move/b52566f1-6353-404a-98de-04e9a2a351a1" AND %{Summary} ~ "Incident"</code> This feature will be run as Jira Workflow Toolbox . 
3. Set issue status to the linked status of the destination workflow step.
4. Re-index an issue to keep indexes in sync with the database.
5. Fire a Issue Created event that can be processed by the listeners.

 Ensure that **Transition issues** is placed after the **Create the issue originally** post function.

Once those two transitions are configured, requests created via email with the **keyword "Incident"** in its summary - which is the email subject - will be moved to the new issue and request type.

Other examples of that function

Page: [Automatic work log with start and stop work transitions](#)
 Page: [Automatically log work time when the user uses a "Stop Progress" transition](#)
 Page: [Calculate the time elapsed between 2 transition executions](#)
 Page: [Getting the number of selected values in a custom field of type Multi Select](#)
 Page: [Implement a form with a series of questions and calculate a numeric value based on the answers](#)
 Page: [Increment a field or set to 1 if it's not set](#)
 Page: [Set "Date-Time Picker" custom field with current date-time](#)
 Page: [Set "Due date" 6 natural days \(or work days\) earlier than a "Date Picker" custom field](#)
 Page: [Set "Due date" to a specific day of next week no matter of date of creation this week](#)
 Page: [Set "Due date" with certain time offset from current date](#)
 Page: [Set "Total time spent" to "Current date and time - date and time of last update"](#)
 Page: [Set a custom field "Urgency" depending on a combined value of issue's priority and "Impact" custom field](#)
 Page: [Sum "Time Spent" in all sub-tasks of issues linked with issue link types "LinkA", "LinkB", "LinkC"](#)
 Page: [Triage Jira Service Desk email requests \(Move issues\)](#)
 Page: [Using project properties to calculate custom sequence numbers](#)

Related Usage Examples

- [Automatically log work time when the user uses a "Stop Progress" transition](#)
 - [example](#)
 - [post-function](#)
 - [custom-field](#)
 - [work-log](#)
- [Limit the number of hours a user can log per day](#)
 - [example](#)
 - [validator](#)
 - [post-function](#)
 - [work-log](#)
- [Make "Time Spent" field required when there is no time logged in the issue](#)
 - [example](#)
 - [validator](#)
 - [work-log](#)
- [Limit valid dates for work logs](#)
 - [example](#)
 - [validator](#)
 - [work-log](#)
- [Log absence time on another issue](#)
 - [example](#)
 - [post-function](#)
 - [work-log](#)
- [Set "Total time spent" to "Current date and time - date and time of last update"](#)
 - [example](#)
 - [post-function](#)
 - [work-log](#)
- [Sum "Time Spent" in all sub-tasks of issues linked with issue link types "LinkA", "LinkB", "LinkC"](#)
 - [example](#)
 - [post-function](#)
 - [issue-links](#)
 - [sub-task](#)
 - [work-log](#)
- [Automatic work log with start and stop work transitions](#)
 - [example](#)
 - [post-function](#)

- work-log
- Sum sub-task's "Time Spent" (work logs) and add it to a certain linked issue
 - example
 - post-function
 - issue-links
 - sub-task
 - work-log
- Validate that multi-user picker custom field A does not contain any user in multi-user picker custom field B
 - example
 - validator
 - custom-field
- Validation on the value of a Cascading Select field
 - example
 - validator
 - custom-field
- Make different fields mandatory depending on the value of a Select List custom field
 - example
 - validator
 - custom-field
- Validate compatible values selection among dependent custom fields
 - example
 - validator
 - custom-field
- Make a custom field mandatory when priority is "Critical" or "Blocker" and issue type is "Incident"
 - example
 - validator
 - custom-field
- Enforce a field (Select List) to be set when another field (Radio Button) has a certain value (works with any kind of field type)
 - example
 - validator
 - custom-field