

# Changing issue priority depending on issue description

## On this page

- [Features used to implement the example](#)
- [Example: Changing issue priority depending on issue description](#)
- [Other examples of that function](#)
- [Related Usage Examples](#)

## Features used to implement the example

- [Set a field as a function of other fields](#)

## Example: Changing issue priority depending on issue description

Is it possible to change the workflow to use a string in the issue description to alter the issue priority upon issue creation? Let's say we receive an e-mail with a string like this in it "**REPORTEDPRIORITY= 1**". Is it somehow possible to use this string from the description to change the issue priority to 1 with a post-function?

It's possible to do it simply by adding a post-function [Set a field as a function of other fields](#) in transition "**Create Issue**", setting field "**Priority**" depending on whether field "**Description**" matches one of a set of regular expressions. You can see exactly how to do it in the following two screenshots:

Field to be checked for matching with type 1 setting rules:

Description - [Text]

This field is only used by rules where conditional part is a regular expression written in brackets: '(*regular\_expression*)'*value*

Target field to be set:

Priority - [Issue priority]

Field to be set by first matched setting rule. Type of the field is shown in square brackets. Check [documentation on Virtual Fields](#) to get information about suitable values for setting selected target field.

☐ Don't overwrite target field if it's already set.

Setting rules:

There are two types of setting rules, and both types can be combined in the same post-function.

Rule formats:

- type 1: '(*regular\_expression*)'*value*

- type 2: '[*boolean\_expression*]'*value*

Write only one rule per line.

*value* may be a parsed text or a mathematical or time formula, depending on the type of selected *Target field*.

[Regular expression syntax](#)

1 ((.[\r\n])\*bREPORTEDPRIORITY\s\*=\s\*0\b(.[\r\n]))\*Blocker

2 ((.[\r\n])\*bREPORTEDPRIORITY\s\*=\s\*1\b(.[\r\n]))\*Critical

3 ((.[\r\n])\*bREPORTEDPRIORITY\s\*=\s\*2\b(.[\r\n]))\*Major

4 ((.[\r\n])\*bREPORTEDPRIORITY\s\*=\s\*3\b(.[\r\n]))\*Minor

5 ((.[\r\n])\*bREPORTEDPRIORITY\s\*=\s\*4\b(.[\r\n]))\*Trivial

☐ Evaluate all the setting rules, not stopping at first match. Only for multi-valued and ephemeral target fields.

[ Line 5 / Col 61 ]

Check Syntax

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

Insert Numeric Value

Summary - [Text] - %{00000}

Insert String Value

Write only one rule per line. The rules will be processed in order. Once a rule is matched, its associated value will be parsed or calculated and copied to selected target field, and the rest of the rules won't be processed. If selected target field is of type number, date, date and time, the associated value should be a number or a mathematical/time formula. Other types like *user*, *date*, *issue status*, *issue priority* and *issue resolution* require values of corresponding suitable types.

There are 2 kinds of rules which can be combined in the same post-function:

- type 1: uses format *optional\_prefixes* (*regular\_expression*)*value* . This setting rules check if a certain regular expression matches selected *field to be checked*.

- type 2: uses format *optional\_prefixes* [*boolean\_expression*]*value* . This is the most powerful of both types of rules, since you can use complex boolean expressions, including a combination of math, date, time and text-string terms with logical connectives OR, AND and NOT.

In the case of setting rules of type 1, both *regular expression* and *value* will be parsed like in post-function "Copy parsed text to a field", this way, by inclusion of field codes, you will be able to create dynamic regular expressions and assignable values.

Optional prefixes are single characters that can precede setting rules for changing somehow its behavior:

a : makes value part in setting rules to be parsed in advanced parsing mode.

i : in type 1 setting rules, makes *regular expression* to be evaluated in ignore case mode.

l : in type 1 setting rules, makes *regular expression* to be treated as a literal string.

Setting rules used were:

```
((.[\r\n])*bREPORTEDPRIORITY\s*=\s*0\b(.[\r\n]))*Blocker
((.[\r\n])*bREPORTEDPRIORITY\s*=\s*1\b(.[\r\n]))*Critical
((.[\r\n])*bREPORTEDPRIORITY\s*=\s*2\b(.[\r\n]))*Major
((.[\r\n])*bREPORTEDPRIORITY\s*=\s*3\b(.[\r\n]))*Minor
((.[\r\n])*bREPORTEDPRIORITY\s*=\s*4\b(.[\r\n]))*Trivial
```

Once configured, the transition looks like this:

The following will be processed after the transition occurs

[Add post function](#)

1. The field **Priority** will be set according to the evaluation of **Description** against the following set of rules:

```
((.[\r\n])*\bREPORTEDPRIORITY\s*=\s*0\b(.|[\r\n])*)Blocker
((.[\r\n])*\bREPORTEDPRIORITY\s*=\s*1\b(.|[\r\n])*)Critical
((.[\r\n])*\bREPORTEDPRIORITY\s*=\s*2\b(.|[\r\n])*)Major
((.[\r\n])*\bREPORTEDPRIORITY\s*=\s*3\b(.|[\r\n])*)Minor
((.[\r\n])*\bREPORTEDPRIORITY\s*=\s*4\b(.|[\r\n])*)Trivial
```

This feature will be run as user in field **Current user**.

by JWT

## Other examples of that function

Page: [Add watcher depending on security level](#)  
Page: [Add watchers based on issue type](#)  
Page: [Add watchers depending on the value of a custom field](#)  
Page: [Assign issue based on the value of a Cascading Select custom field](#)  
Page: [Assign issue to a specific user based on a specific custom field value](#)  
Page: [Assign issue to current user if assignee is empty](#)  
Page: [Assign issue to current user if the user is not member of a certain project role](#)  
Page: [Change assignee based on a custom field](#)  
Page: [Change parent's status depending on sub-task's summary](#)  
Page: [Changing issue priority depending on issue description](#)  
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: [Limit the number of hours a user can log per day](#)  
Page: [Make parent issue progress through its workflow](#)  
Page: [Rise priority if due date is less than 3 weeks away](#)  
Page: [Set "Due date" depending on the value of other fields, in case it's uninitialized](#)  
Page: [Set "Due date" to a specific day of next week no matter of date of creation this week](#)  
Page: [Set "Due date" to current date at issue creation if not initialized](#)  
Page: [Set a custom field "Urgency" depending on a combined value of issue's priority and "Impact" custom field](#)  
Page: [Set a date based on current date](#)  
Page: [Set a field based on reporter's email](#)  
Page: [Set a watcher at ticket creation depending on custom field's value](#)  
Page: [Set assignee depending on issue type](#)  
Page: [Set security level based on groups and project roles the reporter or creator are in](#)  
Page: [Set security level depending on reporter or creator](#)  
Page: [Set the assignee based on a condition](#)  
Page: [Set the value of a field of type "User Picker" depending on other field's value](#)  
Page: [Set watchers depending on the value of a custom field](#)  
Page: [Setting a custom field \(User Picker\) based on the value of another custom field \(Text Field\)](#)  
Page: [Setting a field's default value depending on another field](#)  
Page: [Setting the priority depending on the multiplication of custom fields](#)  
Page: [Transition an issue automatically depending on the value of a field](#)  
Page: [Unassign an issue when assigned to project leader](#)  
Page: [Update checkboxes custom field if a file has been attached during a transition](#)  
Page: [Using project properties to calculate custom sequence numbers](#)

## Related Usage Examples

- [Creating a Jira Service Desk internal comment](#)
  - [example](#)
  - [post-function](#)
- [Limit the number of hours a user can log per day](#)
  - [example](#)
  - [validator](#)
  - [post-function](#)
  - [work-log](#)
- [Using project properties to calculate custom sequence numbers](#)
  - [example](#)
  - [post-function](#)
  - [calculated-field](#)
  - [project-properties](#)
- [Set a date based on current date](#)
  - [example](#)
  - [post-function](#)
- [Setting the priority depending on the multiplication of custom fields](#)
  - [example](#)
  - [calculated-field](#)
  - [post-function](#)
- [Parse Email addresses to watchers list](#)
  - [example](#)
  - [post-function](#)
- [Set the assignee based on a condition](#)
  - [example](#)
  - [post-function](#)
- [Create a dynamic set of sub-tasks based on checkbox selection with unique summaries](#)
  - [example](#)
  - [post-function](#)
  - [custom-field](#)
  - [sub-task](#)
- [Create a static set of sub-tasks with unique summaries](#)
  - [example](#)
  - [post-function](#)
- [Triage Jira Service Desk email requests \(Move issues\)](#)
  - [example](#)
  - [post-function](#)
  - [move](#)
  - [transition-issue](#)
- [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