

Set security level based on groups and project roles the reporter or creator are in

On this page

- [Features used to implement the example](#)
- [Example: Set security level based on groups and project roles the reporter or creator are in](#)
- [Alternative Implementation](#)
- [Other examples of that functions](#)
- [Related Usage Examples](#)

Features used to implement the example

- [Copy parsed text to a field](#)
- [Set a field as a function of other fields](#)

Example: Set security level based on groups and project roles the reporter or creator are in

Is there a way to set the Issue of ticket when created based on a group the User is in? We want the default security to be open when an issue is created, but if a certain set of users create a ticket we want the security to be restricted.

Yes, you can set **Security Level** of an issue based on almost any circumstance you may imagine, and of course you can do it based on the **groups** or **project roles** one or more users are in.

To do it you can use post-functions [Copy parsed text to a field](#) using **advanced parsing mode**, or [Set a field as a function of other fields](#) using **type 2** setting rules.

I explain you how to do it both ways. Let's suppose we want to set issue security level to **"Restricted"** if **reporter** or **creator** are in project role **"Manager"** or group **"Bosses"**, otherwise security level will be set to **"Public"**. I'm supposing that **"Restricted"** and **"Public"** are security levels in the Security Level Scheme of my project:

Using [Copy parsed text to a field](#)

Target field:

Security level - [Security level]

Field to be written with the resulting parsed text.

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

Parsing Mode:

Basic

Advanced

Basic mode: Insert field codes anywhere in the text, and they will be replaced with corresponding field values. Field code formats are `%(nnnnn)`, and `%(nnnnn.i)` for Cascading Select fields (i = 0 for base level).

Advanced mode: Strings literals are written in double quotes (*"This is a string."*). Operator `*` is used to concatenate strings, and field codes are like in basic mode, e.g., `"Issue key is " + %(00015) + "`". More information at [parser syntax documentation](#).

Text to be parsed and then copied to target field: [Line 1 / Col 158] [Syntax Specification](#)

```
1 {isInGroup(%{00006}, "Bosses") OR isInGroup(%{00148}, "Bosses") OR isInRole(%{00006}, "Manager") OR isInRole(%{00148}, "Manager")} ? "Restricted" : "Public"
```

Text to be parsed in this example is:

```
(isInGroup(%{00006}, "Bosses") OR isInGroup(%{00148}, "Bosses") OR isInRole(%{00006}, "Manager") OR isInRole(%{00148}, "Manager")) ? "Restricted" : "Public"
```

Note that:

- `%{00006}` is field code for "Reporter"
- `%{00148}` is field code for "Creator"

Alternative Implementation

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

| | |
|--|---|
| Field to be checked for matching with type 1 setting rules: | Summary - [Text] <small>This field is only used by rules where conditional part is a regular expression written in brackets: <code>"(regular_expression)"value</code></small> |
| Target field to be set: | Security level - [Security level] <small>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.</small> <input type="checkbox"/> Don't overwrite target field if it's already set. |
| Setting rules: <small>There are two types of setting rules, and both types can be combined in the same post-function. Rule formats: - type 1: <code>"(regular_expression)"value</code> - type 2: <code>"[boolean_expression]"value</code> Write only one rule per line. <code>value</code> may be a parsed text or a mathematical or time formula, depending on the type of selected <i>Target field</i>. Regular expression syntax</small> | <pre>1 [isInGroup(%{00006}, "Bosses")]Restricted 2 [isInGroup(%{00148}, "Bosses")]Restricted 3 [isInRole(%{00006}, "Manager")]Restricted 4 [isInRole(%{00148}, "Manager")]Restricted 5 [true]Public</pre> |
| <input type="checkbox"/> Evaluate all the setting rules, not stopping at first match. Only for multi-valued and ephemeral target fields. [Line 5 / Col 14] | |

Setting rules used in this example are:

```
[isInGroup(%{00006}, "Bosses")]Restricted
[isInGroup(%{00148}, "Bosses")]Restricted
[isInRole(%{00006}, "Manager")]Restricted
[isInRole(%{00148}, "Manager")]Restricted
[true]Public
```

The difference between "Reporter" and "Creator" is that the creator is the user who **actually created the issue**, and because of that **can't be edited**, while the reporter is the user who **informed about the issue**, and **can be edited** if you have the necessary permissions to do it.

Other examples of that functions

Copy parsed text to a field

- Page: [Add all assignees of certain sub-task types to a "Multi-User Picker" custom field](#)
- Page: [Add and remove a single or a set of items from multi valued fields](#)
- Page: [Add current user to comment](#)
- Page: [Add or remove request participants](#)
- Page: [Add watchers from a part of the issue summary: "Summary_text - watcher1, watcher2, watcher3, ..."](#)
- Page: [Assign issue based on the value of a Cascading Select custom field](#)
- Page: [Assign issue to last user who executed a certain transition in the workflow](#)

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](#)
- [Set a date based on current date](#)
 - [example](#)
 - [post-function](#)
- [Setting the priority depending on the multiplication of custom fields](#)

Page: Automatically close resolved sub-tasks when parent issue is closed

Page: Automatically reopen parent issue when one of its sub-tasks is reopened

Page: Calculate the time elapsed between 2 transition executions

Page: Close parent issue when all sub-tasks are closed

Page: Combine the values of several Multi-User picker fields

Page: Compose a parsed text including the "full name" or a user selected in a User Picker custom field

Page: Compose dynamic text by inserting field values in a text template

Page: Copy issue labels to a custom field

Page: Copy the value of a user property into a user picker

Page: Create a comment in sub-tasks when parent transitions

Page: Execute transition in epic

Page: Getting the number of selected values in a custom field of type Multi Select

Page: Limit the number of hours a user can log per day

Page: Make a sub-task's status match parent issue's current status on creation

Page: Make parent issue progress through its workflow

Page: Moving story to "In Progress" when one of its sub-tasks is moved to "In Progress"

Page: Moving story to "Ready for QA" once all its sub-tasks are in "Ready for QA" status

Page: Parse Email addresses to watchers list

Page: Parsing text from last comment and appending it to issue's summary

Page: Remove versions selected in a version picker custom field

Page: Replace certain issue link types with different ones

Page: Restrict parent issue from closing if it has sub-tasks that were created during a given parent issue status

Page: Set a Select or Multi-Select field using regular expression to express the values to be assigned

Page: Set assignee depending on issue type

Page: Set field depending on time passed since issue creation

Page: Set priority for issues that have been in a certain status for longer than 24 hours

Page: Set security level based on groups and project roles the reporter or creator are in

Page: Transition linked issues in currently active sprint

Page: Transition only a sub-task among several ones

Page: Transition parent issue only when certain issue sub-task types are done

Page: Update Cascading Select custom field with a value of the field in parent issue

Page: Update checkboxes custom field if a file has been attached during a transition

Page: Validation on issue attachments

Page: Validation on MIME types of issue attachments

Page: Writing a comment to blocked issues when blocking issues are resolved

Set a field as a function of other fields

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

- 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
- Using project properties to calculate custom sequence numbers
 - example
 - post-function
 - calculated-field
 - project-properties
- 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
- 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
- Transition only a sub-task among several ones
 - example
 - post-function
 - sub-task
 - transition
 - outdated
- Transition sub-tasks when parent is transitioned
 - example
 - post-function
 - sub-task
 - transition
 - outdated

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