

# Set "Due date" to current date at issue creation if not initialized

## On this page

- [Features used to implement the example](#)
- [Example: Set "Due date" to current date at issue creation if not initialized](#)
- [Other examples of that function](#)
- [Related Usage Examples](#)

## Features used to implement the example

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

## Example: Set "Due date" to current date at issue creation if not initialized

We wanted to add a due date to new issues and managed to do this with a post-function: Field Due date will be assigned with the calculation result of the formula: `{Current date and time}`

It workes fine but we recently found out that it does allways set the due date to current day, even if you enter a different one in the create issue mask. So here's the question. Is it possible to alter the function to only set the due date if none is entered?

Yes, it is possible to do it. You simply have to use post-function [Set a field as a function of other fields](#) to evaluate whether "Due date" is set or not before assigning "Current date and time" to due date. Now I show you a screenshot of how to configure the post-function [Set a field as a function of other fields](#) to implement the solution:

Update parameters of the Set a field from a set of rules based on regular expressions Function for this transition.

Update parameters of the Set a field from a set of rules based on regular expressions Function for this transition.

Field to be checked for matching with the set of setting rules:

Due date

Setting rules:

Put only one rule per line.

Rule format: `{(regular_expression)'value}`

Regular expression syntax

{(^\$){00057}

Write **only one rule per line**. The rules will be processed in order. Once a rule is matched by the field under evaluation, its associated value will be parsed 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** or **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** requiere values of corresponding suitable types.

Rule format: `{(regular_expression)'value}`

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**.

Target field to be set:

Due date

Available fields:

Copy and paste field codes (e.g. `{00001}`) for issue Description) to the text fields that will be parsed. When parsed all the field codes in the text will be replaced with its value.

FIELD CODE	FIELD NAME	FIELD TYPE
<code>{00000}</code>	Summary	Text
<code>{00001}</code>	Description	Text
<code>{00003}</code>	Assignee	JIRA User

## Alternative expression

I'm in need of changing the due date to current date +30 days. You can do it with this post-function configuration:

## Update parameters of the Set a field from a set of rules based on regular expressions Function for this transition.

Update parameters of the Set a field from a set of rules based on regular expressions Function for this transition.

Field to be checked for matching with the set of setting rules:

Due date

### Setting rules:

Put only one rule per line.

Rule format: `{(regular_expression)/value}`

Regular expression syntax

(^\$)(00057)+30\*(DAY)

Write only one rule per line. The rules will be processed in order. Once a rule is matched by the field under evaluation, its associated value will be parsed 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** or **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.

Rule format: `{(regular_expression)/value}`

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**.

Target field to be set:

Due date

## 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](#)
- [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](#)
- [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