

Validation based on the value of a date type project property

On this page

- [Features used to implement the example](#)
- [Example: Validation based on the value of a date type project property](#)
- [Other examples of that function](#)
- [Related Usage Examples](#)

Features used to implement the example

- [Boolean validator with math, date-time or text-string terms](#)

Example: Validation based on the value of a date type project property

Actually I've a validation in a transaction based on a projectProperty. The validation is this:

```
%{11312} >= projectProperty("finifacturacion")
```

Where %{11312} is a selection **date custom field** and in the project I've defined the property: {finifacturacion=01/01/2015}.

I would like to use the value of this property in the validation message. How can I do that? I have tried the following string but it does not work:

```
No es posible modificar los datos de facturación. La fecha tope para hacerlo era el + projectProperty("finifac  
turacion").
```

I have included parser function **stringToDate**(string_date, timezone). This function returns the numeric value for string containing a date-time value for a certain timezone. To implement your validation you should use [Boolean validator with math, date-time or text-string terms](#) with the following configuration:

Boolean expression to be evaluated:

[Line 1 / Col 87] [Syntax Specification and Examples](#)

```
1 {11500} >= stringToDate(projectProperty("finifacturacion"), LOCAL)
```

Logical connectives: and, or 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.

CHECK SYNTAX

NUMERICAL AND DATE-TIME TERMS

Numeric and Date-Time field values: insert field codes with format {nnnnn}.

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

Insert Numeric Value

Valid date-time literal formats: yyyy/MM/dd [hh:mm] or yyyy-MM-dd [hh:mm]. Time literals use format: hh:mm.

There is a set of [mathematical functions](#) and [time macros and functions](#) available to be used in your expression.

TEXT-STRING TERMS

Text-String field values: insert field codes with format %{nnnnn} or %{nnnnn.i} for referencing levels in cascading select fields (i = 0 for base level).

Summary - [Text] - %{00000} ▾

Insert String Value

String literals: written in **double quotes**, e.g., "This is a string literal."

String concatenation: use operator '+' to concatenate string values, e.g., "The summary of issue with key " + %{00015} + " is \" + %{00000} + "\"."

Escape character: character \ is used with characters "", \, 'n', 'r', 't', 'f' and 'b' to invoke an alternative interpretation.

There is a set of [string functions](#) available to be used in your expression.

Skip validation when:
Inhibit the validator under selected circumstances.

☐ Transition is triggered by a [bulk operation](#).
☐ Transition is triggered by a [JIRA Workflow Toolbox post-function](#).
☐ Current issue is being created by cloning. Only makes sense in *Create Issue* transition.

Message to show when validation fails:

Field code injector:

Summary - [Text] - %{00000} ▾

Field codes with format %(nnnnn) can be inserted in the failure message and its translations, and they will be replaced with actual field values at runtime.

[Set translations for installed languages](#)

Boolean expression used is: {11500} >= stringToDate(projectProperty("finifacturacion"), LOCAL)

Note that:

- {11500} is code for numeric value of certain **Date Picker** or **Date-Time Picker** custom field. This field code depends on each particular Jira instance.
- In this example I have used timezone **LOCAL**, since I suppose that the literal date-time value in project property **finifacturacion** should be interpreted in Jira server's local timezone configuration.
- Valid string date-time formats for project properties are: **yyyy/MM/dd HH:mm**, **yyyy-MM-dd HH:mm**, **yyyy/MM/dd** and **yyyy-MM-dd**, also formats relative to current time like in JQL queries: "w" (weeks), "d" (days), "h" (hours) or "m" (minutes), or format defined at property system **jira.date.time.picker.java.format**. In your particular case, you should write your project property like this: **{finifacturacion=2015/01/01}**

Other examples of that function

Related Usage Examples

Page: [Block a transition until all sub-tasks have certain fields populated](#)

- [Block or unblock a transition after an issue rested a specific time in a status](#)

Page: Block an epic's transition depending on linked issues status and due date

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

Page: Block or unblock a transition after an issue rested a specific time in a status

Page: Block transition until all sub-tasks are in a specific status category

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

Page: Enforce a field (Select List) to be set when another field (Radio Button) has a certain value (works with any kind of field type)

Page: Ensure that all issues linked with a certain issue link type have "Due Date" field set

Page: If field A is populated then, field B must also be populated

Page: Limit issue creation per role and issue type

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

Page: Limit valid dates for work logs

Page: Make "Time Spent" field required when there is no time logged in the issue

Page: Make a custom field mandatory when priority is "Critical" or "Blocker" and issue type is "Incident"

Page: Make attachment mandatory depending on the value of certain custom field

Page: Make different fields mandatory depending on the value of a Select List custom field

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

Page: Make parent issue progress through its workflow

Page: Prevent issue creation if another issue with same field value already exists

Page: Reject duplicated file names in attachments

Page: Require at least one sub-task in status "Resolved" or "Closed" when "Testing required" is selected in Check-Box custom field

Page: Require issue link when resolving as duplicate

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

Page: Restrict sub-task type creation depending on parent issue status

Page: Restrict sub-task type creation depending on parent issue type

Page: Set a condition in a global transition which only applies in a certain status

Page: Validate a custom field "Story Points" has been given a value in Fibonacci sequence

Page: Validate compatible values selection among dependent custom fields

Page: Validate only issue links created in transition screen

Page: Validate that multi-user picker custom field A does not contain any user in multi-user picker custom field B

Page: Validation and condition based on time expressions

Page: Validation based on the value of a date type project property

Page: Validation on issue attachments

Page: Validation on MIME types of issue attachments

Page: Validation on sibling sub-tasks depending on issue type and status

Page: Validation on the value of a Cascading Select field

- example
 - condition
 - validator
 - transition
- Validation based on the value of a date type project property
 - example
 - validator
- Validation and condition based on time expressions
 - example
 - condition
 - validator
 - transition
- Validation on sibling sub-tasks depending on issue type and status
 - example
 - validator
 - sub-task
 - transition
- Validate only issue links created in transition screen
 - example
 - validator
 - issue-links
- Require issue link when resolving as duplicate
 - example
 - validator
 - issue-links
- Validate compatible values selection among dependent custom fields
 - example
 - validator
 - custom-field
- Restrict sub-task type creation depending on parent issue status
 - example
 - validator
 - sub-task
- Require at least one sub-task in status "Resolved" or "Closed" when "Testing required" is selected in Check-Box custom field
 - example
 - validator
 - sub-task
- Validation on MIME types of issue attachments
 - example
 - validator
- Validate a custom field "Story Points" has been given a value in Fibonacci sequence
 - example
 - validator
 - custom-field
- Restrict sub-task type creation depending on parent issue type
 - example
 - validator
 - sub-task
- 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
- Block a transition until all sub-tasks have certain fields populated
 - example
 - condition
 - validator
 - sub-task
 - transition
- If field A is populated then, field B must also be populated
 - example
 - validator
- 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
- Ensure that all issues linked with a certain issue link type have "Due Date" field set
 - example
 - validator
 - issue-links
- Make different fields mandatory depending on the value of a Select List custom field
 - example
 - validator
 - custom-field
- Reject duplicated file names in attachments
 - example
 - validator
- 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
- Block an epic's transition depending on linked issues status and due date
 - example
 - validator
 - issue-links
 - transition
- Limit valid dates for work logs
 - example
 - validator
 - work-log
- Make attachment mandatory depending on the value of certain custom field
 - example
 - validator
 - custom-field
- Close parent issue when all sub-tasks are closed
 - example
 - condition
 - validator
 - post-function
 - sub-task
 - transition
- Make certain custom field required in resolve screen only if the resolution was set to "Fixed"
 - example
 - validator
 - custom-field
- Restrict parent issue from closing if it has sub-tasks that were created during a given parent issue status
 - example
 - validator
 - post-function
 - sub-task
 - transition
- Prevent addition of new sub-tasks if the parent issue is in "Resolved" or "Closed" status
 - example
 - validator
 - sub-task
- Prevent issue from moving forward if it's dependent on non-accepted tickets
 - example
 - validator
 - issue-links
 - transition
- Enforce linked issues in a specific project to be "Closed" before closing issue
 - example
 - validator
 - issue-links
 - transition
- Proceed with a task only when all sub-tasks are completed
 - example

- condition
 - validator
 - sub-task
 - transition
- Block or hide a transition for an issue depending on its issue links
 - example
 - validator
 - issue-links
 - transition
- Make "Affects Version/s" mandatory when issue resolution is "Fixed"
 - example
 - validator
- Prevent transitioning when there is a blocking issue
 - example
 - validator
 - issue-links
 - sub-task
 - transition
- Validation on issue attachments
 - example
 - validator
- Prevent issue creation if another issue with same field value already exists
 - example
 - validator
 - custom-field
- Prevent issue from being "Closed" if blocking issues aren't yet closed
 - example
 - validator
 - issue-links
 - transition
- Block creation of issue type X if it has not been linked with link type Y to issue type Z on the "Create Issue" screen
 - example
 - validator
 - issue-links
- Make a custom field mandatory when Priority is "Critical" or "Blocker"
 - example
 - validator
 - custom-field
- Limit issue creation per role and issue type
 - example
 - validator
- Transition parent issue only when certain issue sub-task types are done
 - example
 - validator
 - sub-task
 - transition
- Enforce certain type of sub-tasks to be "Resolved" to allow executing a transition
 - example
 - validator
 - sub-task
 - transition
- Prevent issue from being closed if it has links of type "is blocked by" to open issues
 - example
 - condition
 - validator
 - issue-links
 - transition
- Automatically resolve an epic when all its stories are resolved
 - example
 - condition
 - validator
 - post-function
 - transition
- Make linked issues, sub-tasks and JQL selected issues progress through its workflows
 - example
 - condition
 - validator
 - post-function
 - issue-links

- sub-task
- transition