

# Validation on MIME types of issue attachments

## On this page

- [Features used to implement the example](#)
- [Example: Validation on MIME types of issue attachments](#)
- [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 on MIME types of issue attachments

You may also find it useful knowing that you can also validate the type of files attached.

Let us suppose we want to validate that our issue has the following files attached:

- At least 1 .jar file.
- At least 3 images (any format).
- No more than 5 .txt file
- Exactly 1 XML file.

To implement this validation we will use [Boolean validator with math, date-time or text-string terms](#) with the following configuration:

Boolean expression to be evaluated:

Syntax Specification

1

```
matches(%{00072}, "(.*application/java-archive.*){1,}") AND matches(%{00072}, "(.*text/plain.*){0,5}") AND matches(%{00072}, "(.*text/xml.*){1}") AND matches(%{00072}, "(.*image/.*){3,}")
```

Logical connectives: **or**, **and** 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.

**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.

Message to show when validation fails:

The issue should contain at least 1 .jar file and 3 images, no more than 5 .txt files and 1 .xml file.

Boolean expression used is: `matches(%{00072}, "(.*application/java-archive.*){1,}") AND matches(%{00072}, "(.*text/plain.*){0,5}") AND matches(%{00072}, "(.*text/xml.*){1}") AND matches(%{00072}, "(.*image/.*){3,}")`

Note that:

- `%{00072}` is field code for "**Attachments with details**"

To design this configuration, We have previously used [Copy a parsed text to a field](#) post-function in order to see the value that field "**Attachments with details**" takes when we attach different file types. Then it is easy to write a regular expressions for matching desired values.

## Other examples of that function

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

## Related Usage Examples

- [Block or unblock a transition after an issue rested a specific time in a status](#)
  - [example](#)
  - [condition](#)
  - [validator](#)
  - [transition](#)
- [Validation on sibling sub-tasks depending on issue type and status](#)
  - [example](#)
  - [validator](#)
  - [sub-task](#)
  - [transition](#)
- [Validation on MIME types of issue attachments](#)
  - [example](#)
  - [validator](#)
- [Validation on the value of a Cascading Select field](#)
  - [example](#)
  - [validator](#)
  - [custom-field](#)
- [Validation based on the value of a date type project property](#)
  - [example](#)
  - [validator](#)
- [Validation and condition based on time expressions](#)
  - [example](#)
  - [condition](#)
  - [validator](#)
  - [transition](#)
- [Validate only issue links created in transition screen](#)
  - [example](#)
  - [validator](#)
  - [issue-links](#)
- [Require issue link when resolving as duplicate](#)
  - [example](#)
  - [validator](#)
  - [issue-links](#)
- [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](#)
- [Reject duplicated file names in attachments](#)
  - [example](#)
  - [validator](#)
- [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](#)

- 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