Reject duplicated file names in attachments

On this page

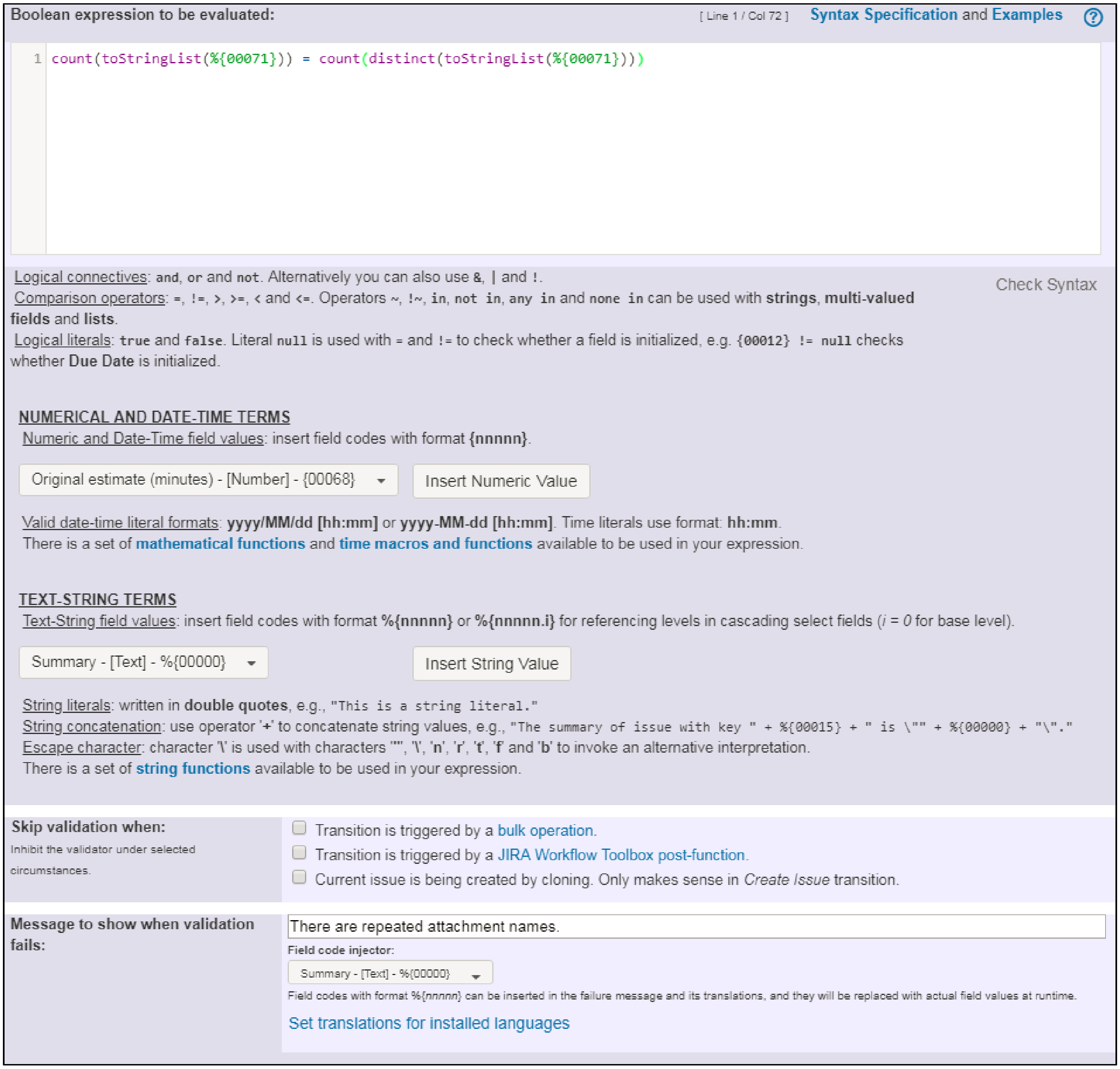
Features used to implement the example

- **Boolean validator with math, date-time or text-string terms**

Example: Reject duplicated file names in attachments

How to configure a validation for rejecting attachments with repeated file names?

Use **Boolean validator with math, date-time or text-string terms** with the following configuration:



Boolean expression to be evaluated: [Line 1 / Col 72] **Syntax Specification** and **Examples** ?

```
count(toStringList(%{00071})) = count(distinct(toStringList(%{00071})))
```

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.

NUMERICAL AND DATE-TIME TERMS
Numeric and Date-Time field values: insert field codes with format **{nnnnn}**.

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

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}

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:

There are repeated attachment names.

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.

Boolean expression is: **`count(toStringList(%{00071})) = count(distinct(toStringList(%{00071})))`**

Note that:

- **%{00071}** is field code for "**Attachments**"

Other examples of that function

Page: Block a transition until all sub-tasks have certains 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

Related Usage Examples

- Block or unblock a transition after an issue rested a specific time in a status
    - example
    - condition
    - validator
    - transition

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

- 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
- 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
- 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 certains 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