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

On this page

- Features used to implement the examples
- Example 1: Prevent a closed issue from being reopened after resting 1 week closed
- Example 2: Ensure that issues rest in certain status at least 24 hours
- Example 3: Enroute issues to different statuses depending on the time they rested in current status
- Other examples of that function
- Related Usage Examples

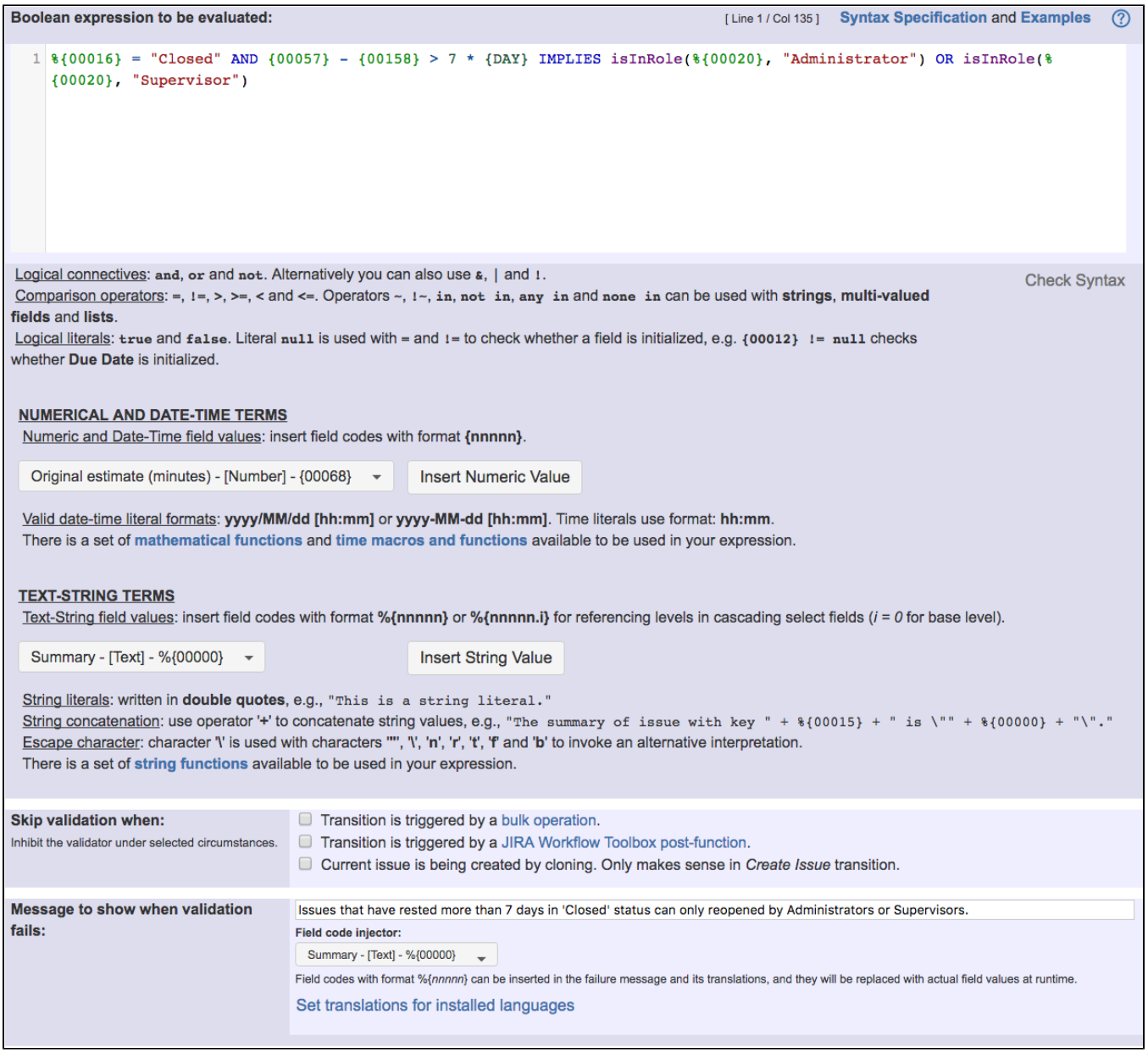
Features used to implement the examples

- **Boolean condition and validator with math. date-time or text-string terms**
- Virtual field **Current date and time**
- Virtual field **Date and time of latest status change**(available from version 2.1.22): subtracting it to **Current date and time** gives us the time current issue has rested in current status.
- Virtual field **Current user** and function **isInRole**(): used as backdoor for allowing transition being executed by certain roles, as **Administrator**.

Example 1: Prevent a closed issue from being reopened after resting 1 week closed

We want to prevent a certain issue from being reopened after 7 days closed, unless user has role "**Administrator**" or "**Supervisor**".

We implement this use case by inserting a validator **Boolean validator with math, date-time or text-string terms** in transition "**Reopen Issue**" with the following configuration:



Boolean expression used in the example is:

```
%{00016} = "Closed" AND {00057} - {00158} > 7 * {DAY} IMPLIES isInRole(%{00020}, "Administrator") OR isInRole(%{00020}, "Supervisor")
```

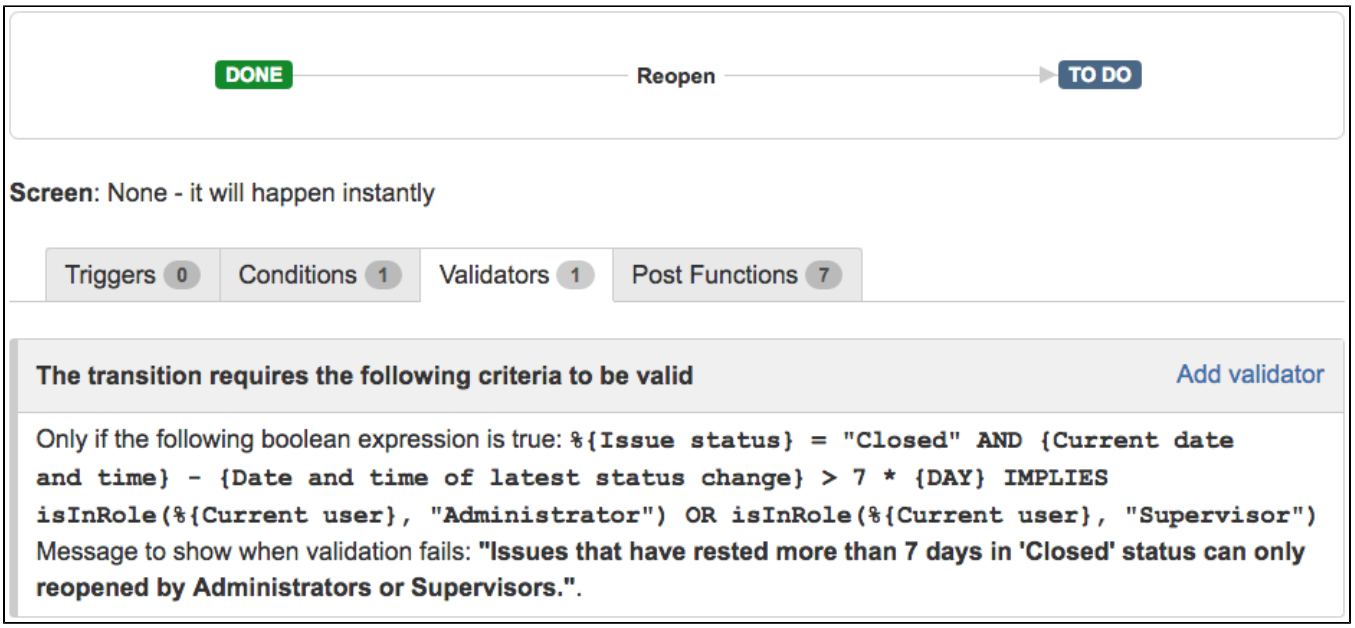
We are using logical connective **IMPLIES** for clarity. An equivalent expression using only primitive logical connectives is:

```
%{00016} != "Closed" OR {00057} - {00158} < 7 * {DAY} OR isInRole(%{00020}, "Administrator") OR isInRole(%{00020}, "Supervisor")
```

Note that:

- **%{00016}** is field code for **Issue status**
- **{00057}** is code for numerical value of field **Current date and time**
- **{00158}** is code for numerical value of field **Date and time of latest status change** (available from version 2.1.22)
- **%{00020}** is field code for **Current user**

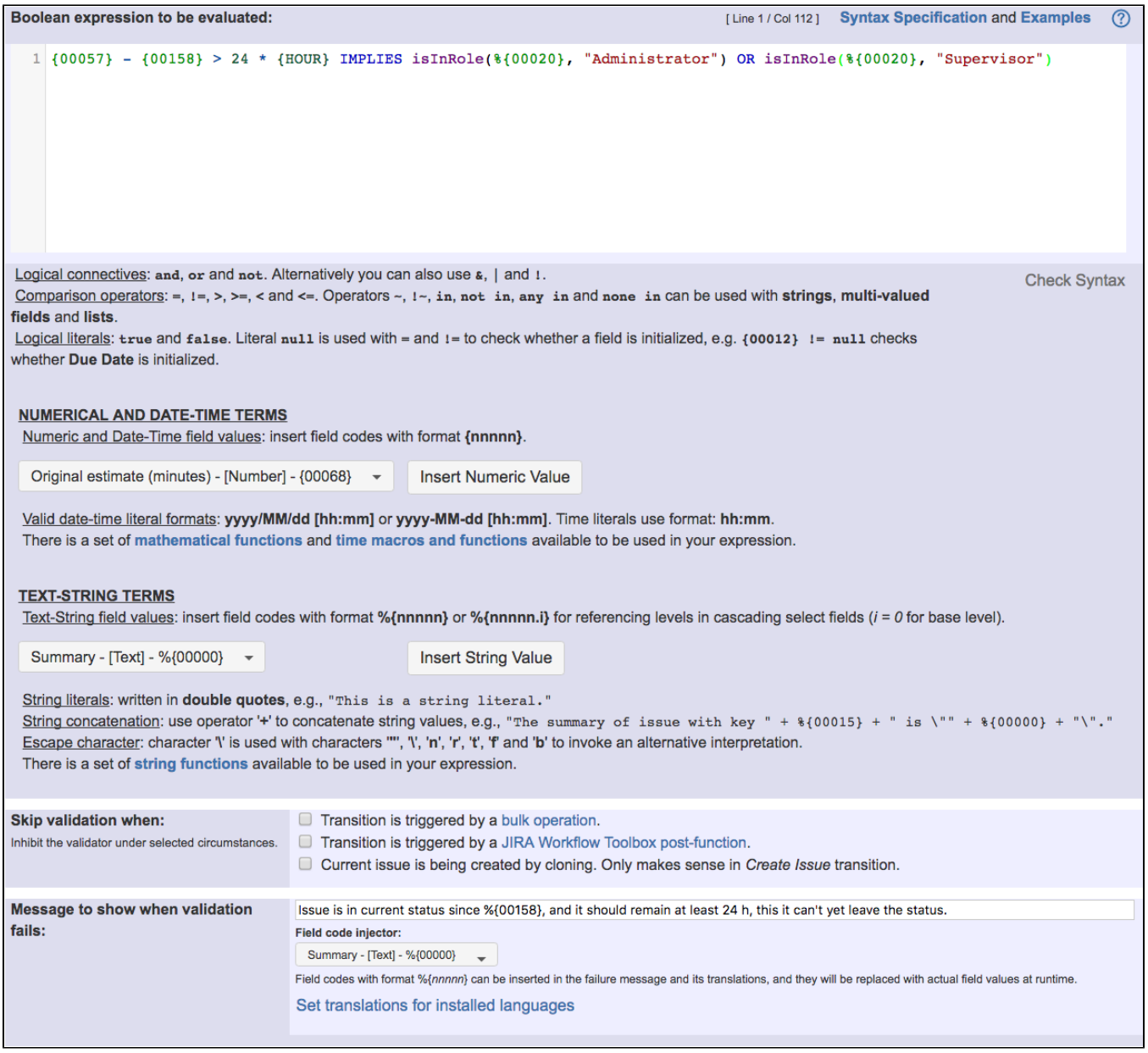
Once configured, transition "**Reopen Issue**" looks like this



Example 2: Ensure that issues rest in certain status at least 24 hours

As in the previous example, we allow users in "**Administrator**" or "**Supervisor**" project role to bypass this time restriction.

We insert **Boolean validator with math, date-time or text-string terms** in all the transitions with origin in the status we want to ensure 24 hours of permanence:



Boolean expression used in the example is:

```
{00057} - {00158} > 24 * {HOUR} IMPLIES isInRole(%{00020}, "Administrator") OR isInRole(%{00020}, "Supervisor")
```

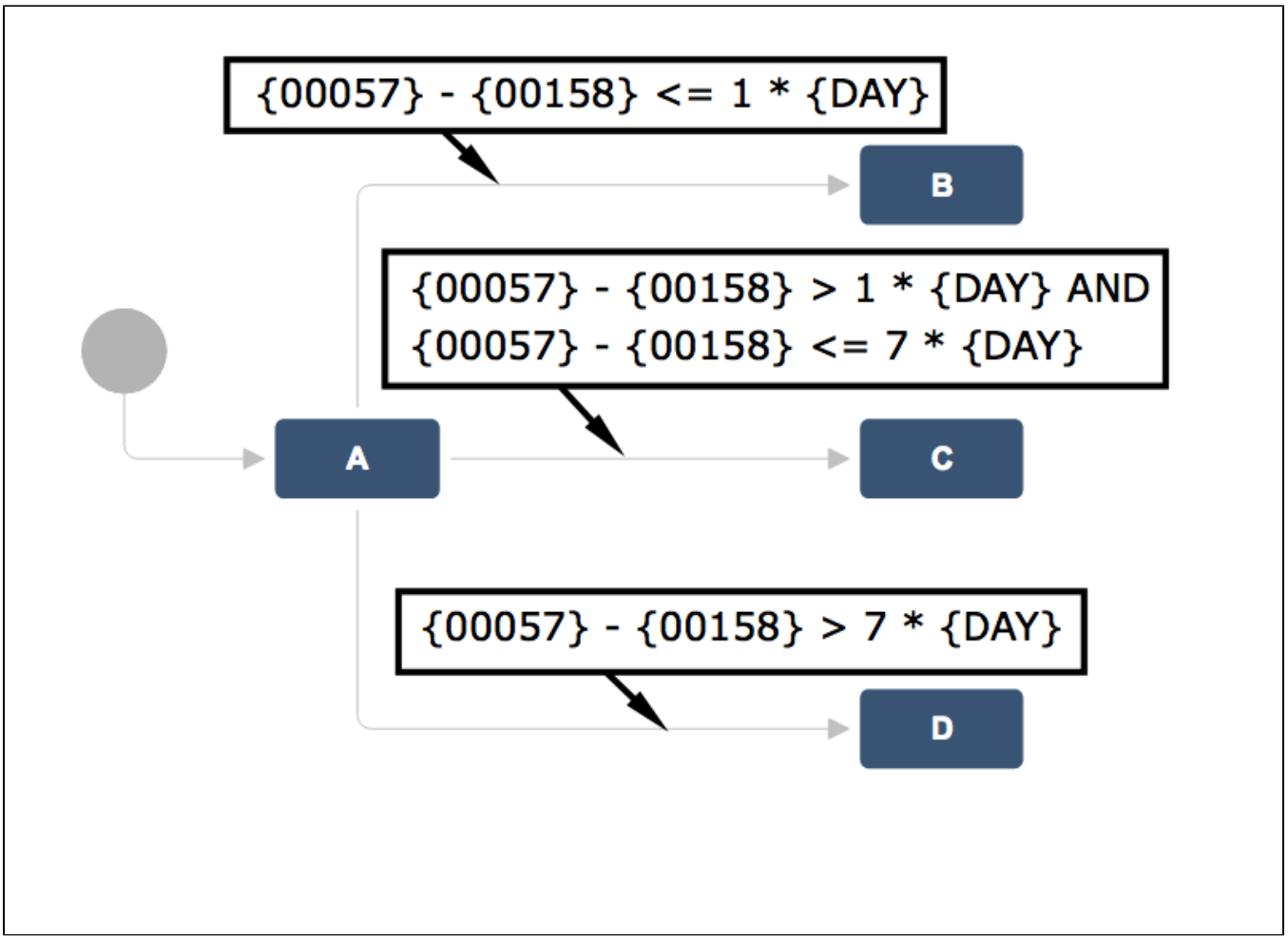
An equivalent expression is:

```
{00057} - {00158} <= 24 * {HOUR} OR isInRole(%{00020}, "Administrator") OR isInRole(%{00020}, "Supervisor")
```

Example 3: Enroute issues to different statuses depending on the time they rested in current status

We have different transitions with origin in a certain status **A**, and with different destination statuses (**B**, **C** and **D**).

We use **Boolean condition with math, date-time or text-string terms** in each of the 3 transitions **A -> B**, **A -> C** and **A -> D** with different boolean expressions, in order to show in UI only one of these transitions depending on the time the issue has passed in status **A**, while hiding the other two transitions:



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
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

Related Usage Examples

- Block or unblock a transition after an issue rested a specific time in a status
  - example
  - condition
  - validator
  - transition
- Block transition until all sub-tasks are in a specific status category
  - example
  - transition
  - condition
- Validation and condition based on time expressions
  - example
  - condition
  - validator
  - transition
- Set a condition in a global transition which only applies in a certain status
  - example
  - condition
  - transition
- Validation on sibling sub-tasks depending on issue type and status
  - example
  - validator
  - sub-task
  - transition
- Block a transition until all sub-tasks have certains fields populated
  - example

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

    - condition
    - validator
    - sub-task
    - transition
- Block an epic's transition depending on linked issues status and due date
    - example
    - validator
    - issue-links
    - transition
- Moving story to "In Progress" when one of its sub-tasks is moved to "In Progress" (Transition issues)
    - example
    - post-function
    - transition
- Transition sub-tasks when parent is transitioned
    - example
    - post-function
    - sub-task
    - transition
    - outdated
- Transition only a sub-task among several ones
    - example
    - post-function
    - sub-task
    - transition
    - outdated
- 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
- Change parent's status depending on sub-task's summary (Transition issues)
    - example
    - post-function
    - transition
- Automatically close resolved sub-tasks when parent issue is closed
    - example
    - post-function
    - sub-task
    - transition
    - outdated
- Change parent's status depending on sub-task's summary
    - example
    - post-function
    - sub-task
    - transition
    - outdated