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

On this page

Features used to implement the example

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

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

We have 3 custom fields

1. "**Development Type**" with values "**Core**" and "**Custom**"
2. "**Story Points"**
3. "**Hours**"

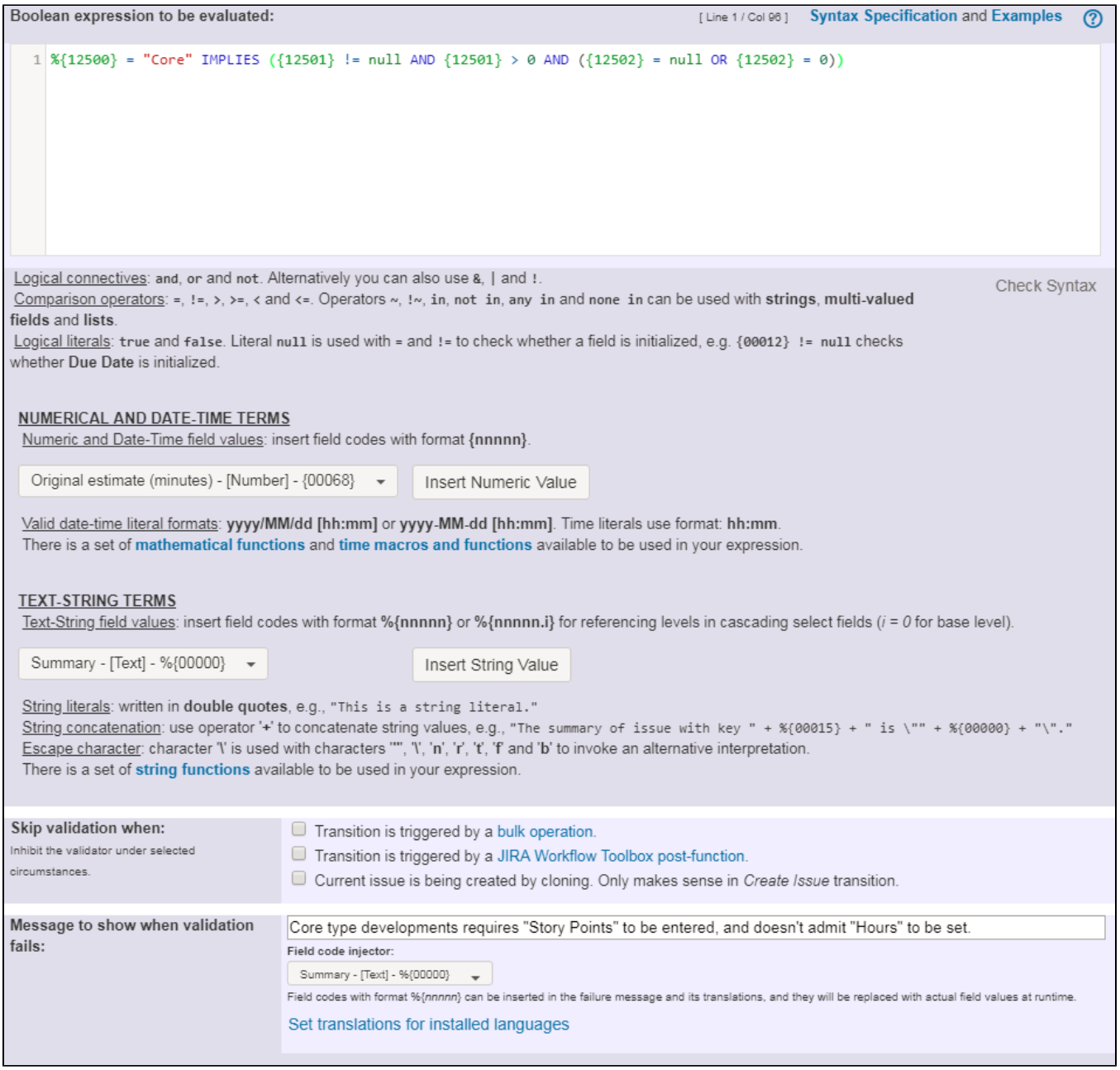
So the requirement is, if Development Type = "**Core**", the Story Points field would be displayed and if Development Type = "**Custom**", the Hours field would be displayed.

It is possible to set 2 validations in your transition, in order to ensure that:

1. When "**Development Type** = Core" then field "**Story Points**" will be **set** and given a value **higher than 0**, and field "**Hours**" is **not set** or **zero**
2. When "**Development Type** = Custom" then field "**Hours**" will be **set** and given a value **higher than 0**, and field "**Story Points**" is **not set** or **zero**

To do it, you just need to use **Boolean validator with math, date-time or text-string terms** with the following configurations:

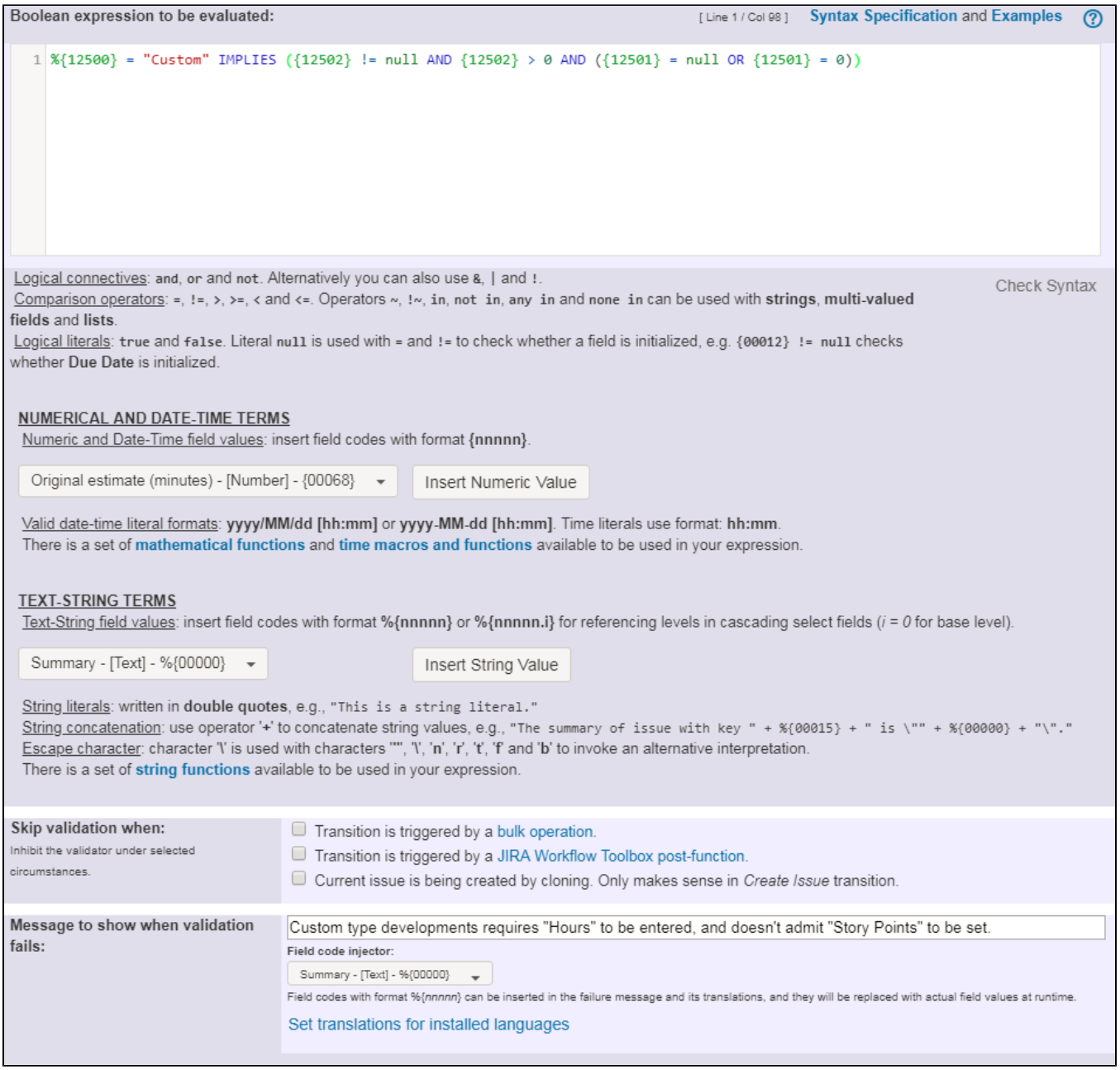
When "**Development Type** = Core" then field "**Story Points**" will be **set** and entered a value **higher than 0**, and field "**Hours**" is **not set** or **zero**:



Expression used is: **`%{12500} = "Core" IMPLIES ({12501} != null AND {12501} > 0 AND ({12502} = null OR {12502} = 0))`**

Other variation of the usage example

When "**Development Type** = Custom" then field "**Hours**" will be **set** and given a value **higher than 0**, and field "**Story Points**" is **not set** or **zero**:

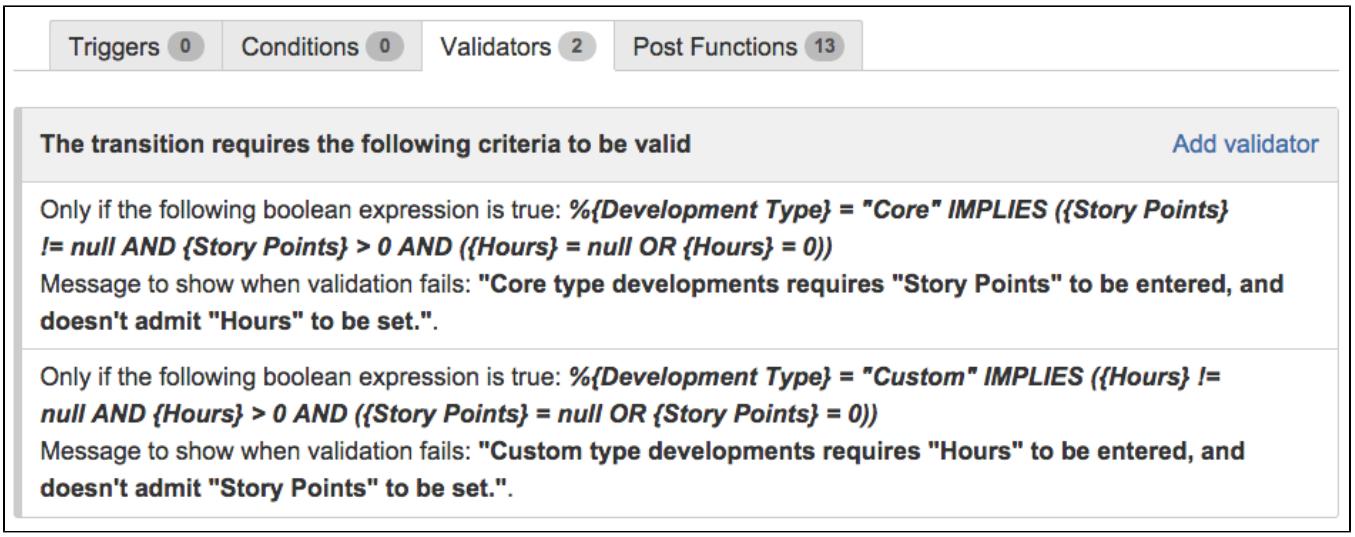


Expression used is: **`%{12500} = "Custom" IMPLIES ({12502} != null AND {12502} > 0 AND ({12501} = null OR {12501} = 0))`**

Note that in the particular JIRA instance I used to build this example:

- **%{12500}** is field code for string value of field **"Development Type"**
- **{12501}** is field code for numeric value of field "**Story Points"**
- **{12502}** is field code for numeric value of field "**Hours"**

Once configured, your transition will look like this:



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
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

Related Usage Examples

- Validate compatible values selection among dependent custom fields
  - example
  - validator
  - custom-field
- Validate a custom field "Story Points" has been given a value in Fibonacci sequence
  - example
  - validator
  - custom-field
- 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
- Make different fields mandatory depending on the value of a Select List custom field
  - example
  - validator
  - custom-field
- 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
- Make attachment mandatory depending on the value of certain custom field
  - example
  - validator
  - custom-field
- Create a dynamic set of sub-tasks based on checkbox selection with unique summaries
  - example
  - post-function
  - custom-field
  - sub-task
- Total of all story points in an epic
  - example
  - custom-field
  - calculated-field

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

- Show timeliness of an issue based on two date pickers
  - example
  - custom-field
  - calculated-field
- Add and remove a single or a set of items from multi valued fields
  - example
  - post-function
  - custom-field
  - issue-links
  - sub-task
- Highest value of a custom field among linked issues
  - example
  - custom-field
  - calculated-field
- Google Maps location from address
  - example
  - calculated-field
  - custom-field
- Make certain custom field required in resolve screen only if the resolution was set to "Fixed"
  - example
  - validator
  - custom-field