Using project properties to calculate custom sequence numbers

Features used to implement the example

- **Project Properties**
- **Read a project property**
- **Mathematical and date-time expression calculator**
- **Update issue fields**
- **Set or create project property**

Example: Using project properties to calculate custom sequence numbers

In this example, a custom identifier **"Approval Code"** is added to issues by using a project property and different custom fields.

The code that will be created is a combination of a specific number (sequence number), the fiscal year (custom field) and abbreviation of a department and will look like this:

> *003/2018-2019/DEV*

Configuration items needed are:

- Project property **"sequenceNumber"**
- Custom Text Field **"Department"**
- Custom Text Field **"Fiscal Year"**
- Custom Text Field **"Approval Code"**

Setting the project property

Setting a project property can be done via the project settings Details.

Details

Name* Development

Key* DEV ⓘ

URL

Project type* ◉ Software ▾ ⓘ

Project category None ⓘ

Avatar* select image

Description
{sequenceNumber=0}

{sequenceNumber=0}

Save details Cancel

A valid expression is: `{sequenceNumber=0}`

Adding multiple post functions

When executing the transition, the current value of **"sequenceNumber"** is stored in an ephemeral field with the **Read a project property** post function.

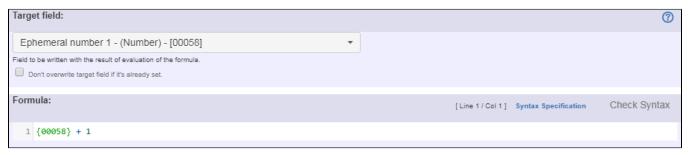| Project key:	
Leave this field **empty** for reading a property in current project.	**Field code injector:**
	Start typing to get the list of availabe fields ▾
	Field codes with format %{nnnnn} will be replaced with their corresponding value at runtime.

Name of project property:	
Project Properties are stored in project description with the following format: {property_name=property_value} e.g. {MaxNumberBugs=5}	sequenceNumber
	Field code injector:
	Start typing to get the list of availabe fields ▾
	Field codes with format %{nnnnn} will be replaced with their corresponding value at runtime.

Target field to store project property value:	
	Ephemeral number 1 - (Number) - [00058] ▾

Additional options:	
	☐ **Update issue immediately** after field writing. A specific entry will be created in issue history for this field writing.

Name of project property is: `sequenceNumber`

> ⓘ **Ephemeral fields** are temporary fields that can be used within a transition, e.g. to cache temporary values. They're cleared when the transition is completed, unless a Inhibit ephemeral fields auto-clearance post function is added to the transition.

Incrementing the **"sequenceNumber"** by one is done with a **Mathematical and date-time expression calculator** post function.

Target field:

Ephemeral number 1 - (Number) - [00058] ▾

Field to be written with the result of evaluation of the formula.

☐ Don't overwrite target field if it's already set.

Formula: [Line 1 / Col 1]   Syntax Specification   Check Syntax
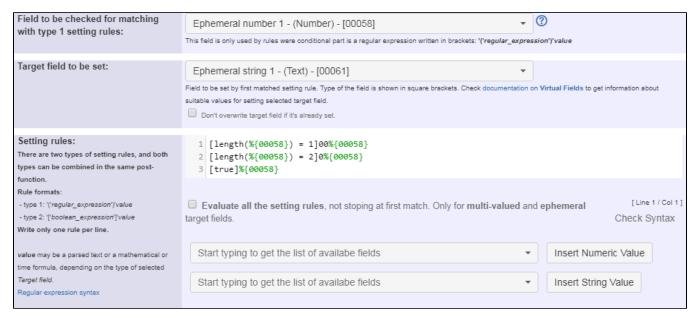
```
1  {00058} + 1
```

Formula is: `{00058} + 1`

Note that:

- **{00058}** is the field code for Ephermeral Number 1

Formatting the **"sequenceNumber"** is done with a **Set a field as a function of other fields** post function: This ensures that the sequence number is always a 3 digit number, e.g. "3" is formatted to "003" and "10" to "010".
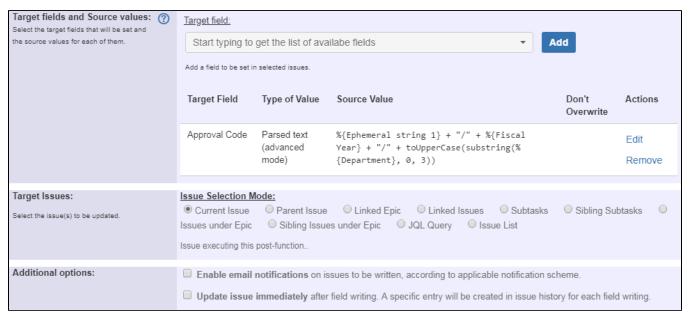
Field to be checked for matching with type 1 setting rules:	Ephemeral number 1 - (Number) - [00058] ▼ ⑦
	This field is only used by rules were conditional part is a regular expression written in brackets: '('regular_expression')'value

Target field to be set:	Ephemeral string 1 - (Text) - [00061] ▼
	Field to be set by first matched setting rule. Type of the field is shown in square brackets. Check documentation on **Virtual Fields** to get information about suitable values for setting selected target field.
	☐ Don't overwrite target field if it's already set.

Setting rules: There are two types of setting rules, and both types can be combined in the same post-function. Rule formats: - type 1: '('regular_expression')'value - type 2: '['boolean_expression']'value Write only one rule per line. value may be a parsed text or a mathematical or time formula, depending on the type of selected Target field. Regular expression syntax	1 [length(%{00058}) = 1]00%{00058} 2 [length(%{00058}) = 2]0%{00058} 3 [true]%{00058} ☐ **Evaluate all the setting rules**, not stoping at first match. Only for **multi-valued** and **ephemeral** target fields. [Line 1 / Col 1] Check Syntax Start typing to get the list of availabe fields ▼ Insert Numeric Value Start typing to get the list of availabe fields ▼ Insert String Value

Setting rules are:

> **[length(%{00058}) = 1]00%{00058}**
> **[length(%{00058}) = 2]0%{00058}**
> **[true]%{00058}**

Note that:

- **%{00058}** is the field code for Ephemeral number 1

Finally, the **"Approval Code"** is composed with an **Update issue fields** post function.

Target fields and Source values: ⑦ Select the target fields that will be set and the source values for each of them.	Target field: Start typing to get the list of availabe fields ▼ **Add** Add a field to be set in selected issues.

Target Field	Type of Value	Source Value	Don't Overwrite	Actions
Approval Code	Parsed text (advanced mode)	%{Ephemeral string 1} + "/" + %{Fiscal Year} + "/" + toUpperCase(substring(%{Department}, 0, 3))		Edit Remove

Target Issues: Select the issue(s) to be updated.	**Issue Selection Mode:** ⦿ Current Issue ◯ Parent Issue ◯ Linked Epic ◯ Linked Issues ◯ Subtasks ◯ Sibling Subtasks ◯ Issues under Epic ◯ Sibling Issues under Epic ◯ JQL Query ◯ Issue List Issue executing this post-function..

Additional options:	☐ **Enable email notifications** on issues to be written, according to applicable notification scheme. ☐ **Update issue immediately** after field writing. A specific entry will be created in issue history for each field writing.

Text to be parsed (in advanced mode) is: **%{00061} + "/" + %{10420} + "/" + toUpperCase(substring(%{10421}, 0, 3))**
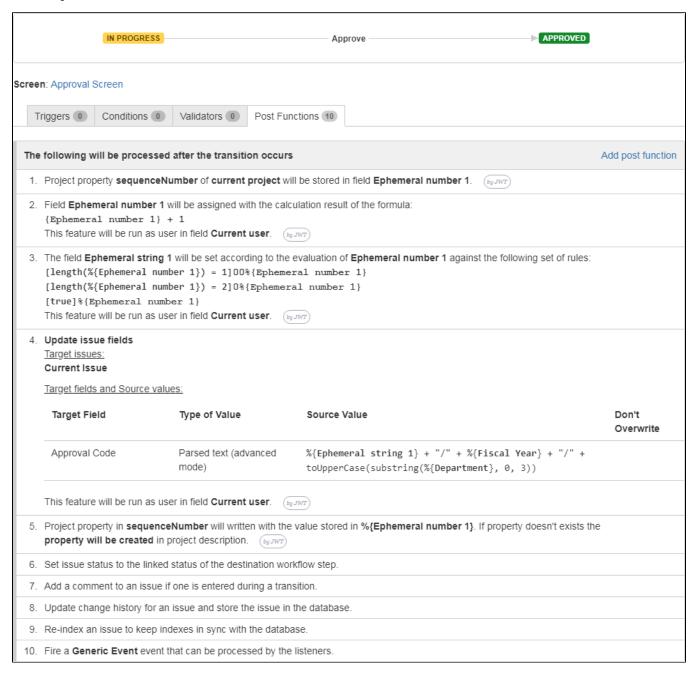
Note that:

- **%{00061}** is the field code for Ephermeral string 1
- **%{10420}** is the field code for custom text field "Fiscal Year" (this field code might differ on your instance)
- **%{10421}** is the field code for custom text field "Department" (this field code might differ on your instance)

In order to keep the **"sequenceNumber"** in sync, a final **Set or create project property** post function is added.

Property Name:		
Project Properties are stored in project description with the following format: {property_name=property_value} e.g. {MaxNumberBugs=5}	sequenceNumber	
	Field code injector:	
	Start typing to get the list of availabe fields ▾	
	Field codes with format %{nnnnn} will be replaced with their corresponding value at runtime.	

Value to be written:	
	%{00058}
	Field code injector:
	Start typing to get the list of availabe fields ▾
	Field codes with format %{nnnnn} will be replaced with their corresponding value at runtime.

Behavior if Project Property doesn't exist:	● Create the property and assign the value ○ Do nothing

Once configured, the transition will look like this:

IN PROGRESS ———————— Approve ————————► APPROVED

Screen: Approval Screen

Triggers 0   Conditions 0   Validators 0   Post Functions 10

The following will be processed after the transition occurs   Add post function

1. Project property **sequenceNumber** of **current project** will be stored in field **Ephemeral number 1**. (by JWT)

2. Field **Ephemeral number 1** will be assigned with the calculation result of the formula:
   `{Ephemeral number 1} + 1`
   This feature will be run as user in field **Current user**. (by JWT)

3. The field **Ephemeral string 1** will be set according to the evaluation of **Ephemeral number 1** against the following set of rules:
   `[length(%{Ephemeral number 1}) = 1]00%{Ephemeral number 1}`
   `[length(%{Ephemeral number 1}) = 2]0%{Ephemeral number 1}`
   `[true]%{Ephemeral number 1}`
   This feature will be run as user in field **Current user**. (by JWT)

4. **Update issue fields**
   Target issues:
   Current Issue

   Target fields and Source values:

Target Field	Type of Value	Source Value	Don't Overwrite
Approval Code	Parsed text (advanced mode)	%{Ephemeral string 1} + "/" + %{Fiscal Year} + "/" + toUpperCase(substring(%{Department}, 0, 3))	

   This feature will be run as user in field **Current user**. (by JWT)

5. Project property in **sequenceNumber** will written with the value stored in **%{Ephemeral number 1}**. If property doesn't exists the **property will be created** in project description. (by JWT)

6. Set issue status to the linked status of the destination workflow step.

7. Add a comment to an issue if one is entered during a transition.

8. Update change history for an issue and store the issue in the database.

9. Re-index an issue to keep indexes in sync with the database.

10. Fire a **Generic Event** event that can be processed by the listeners.

⊘

⊘ When executing this transition, the custom fields **"Fiscal Year"** and **"Department" must not be empty!** Ensuring this can be done by using a validator.

Other examples of that function

Page: Assign new issues to a different project role depending on field value in current issue
Page: Using project properties to calculate custom sequence numbers

Related Usage Examples

- Using project properties to calculate custom sequence numbers
  - example
  - post-function
  - calculated-field
  - project-properties
- Setting the priority depending on the multiplication of custom fields
  - example
  - calculated-field
  - post-function
- Number of open subtasks
  - example
  - calculated-field
- Total of all story points in an epic
  - example
  - custom-field
  - calculated-field
- Show timeliness of an issue based on two date pickers
  - example
  - custom-field
  - calculated-field
- Highest value of a custom field among linked issues
  - example
  - custom-field
  - calculated-field
- Predicted due date of an epic
  - example
  - calculated-field
- Number of issues in same version
  - example
  - calculated-field
- Number of linked bugs (with hyperlink)
  - example
  - calculated-field
- Google Maps location from address
  - example
  - calculated-field
  - custom-field
- Total remaining estimate in an epic
  - example
  - calculated-field
- Creating a Jira Service Desk internal comment
  - example
  - post-function
- Limit the number of hours a user can log per day
  - example
  - validator
  - post-function
  - work-log
- Set a date based on current date
  - example
  - post-function
- Parse Email adresses to watchers list
  - example
  - post-function