

# Prioritize the issues globally

## On this page

[Prerequisites](#) | [Configuration](#) | [Related use cases](#)

## Use case

Create a hierarchy of issues by prioritizing them via a *Number Field* and a couple of post functions in the *Create* transition and a global reflexive transition that will update the hierarchy after a new value has been modified or introduced.



## Prerequisites

1

Create a custom *Number Field* called **Customer Prioritization** and add it to the **Create/View/Edit screens**.

2

Create a new **global reflexive transition** (from any issues to itself) and add it a custom screen that includes the **Customer Prioritization** field.



## Configuration

1

Add an [Update or copy field values](#) post function to the *Create* transition in the second position.

1

### Target issue\*

Select **Set target issues manually (parser expression)** and add the following expression:

```
issuesFromJQL("issuekey != " + {%00015} + " and \"Customer Prioritization\" >= " + {issue.cfCusPrio} + " order by \"Customer Prioritization\" ASC")
```

This expression will select all of the existing issues whose **Customer Prioritization** field value is greater than or equal to the existing one excluding the one from the current issue.

2

### Field\*

Select the **Customer Prioritization** field and add the following expression.

```
{nnnnn} + indexOf(^%{00015}, issuesFromJQL("issuekey != " + %{00015} + " and \"Customer Prioritization\" >= " + {issue.cfCusPrio} + " order by \"Customer Prioritization\" ASC"))
```

This expression will add to the number introduced in the screen for **Customer Prioritization** the index of the issues in the issue list obtained before in every of the issues according to the issue list in the *Target issue* field.

3

### Conditional execution

```
%{nnnnn} != null
```

The post function will only be executed if the field has a value.

2

Add an [Update or copy field values](#) post function to global reflexive transition previously created.

1

### Target issue\*

Select **Set target issues manually (parser expression)** and add the following expression:

```
(previousValue({issue.cfCusPrio}) = null ? issuesFromJQL("issuekey != " + %{00015} + " and \"Customer Prioritization\" >= " + {nnnnn} + " order by \"Customer Prioritization\" ASC") : ({issue.cfCusPrio} < previousValue({issue.cfCusPrio}) ? issuesFromJQL("issuekey != " + %{00015} + " and \"Customer Prioritization\" >= " + {issue.cfCusPrio} + " and \"Customer Prioritization\" <= " + previousValue({issue.cfCusPrio}) + " order by \"Customer Prioritization\" ASC") : issuesFromJQL("issuekey != " + %{00015} + " and \"Customer Prioritization\" >= " + previousValue({issue.cfCusPrio}) + " and \"Customer Prioritization\" <= " + {issue.cfCusPrio} + " order by \"Customer Prioritization\" DESC"))
```

This expression will cover three possible situations:

Case 1 - There was no previous value in the custom field *Customer Prioritization*: The expression will fetch the issues as in the create transition.

Case 2 - The value entered in the custom field *Customer Prioritization* is smaller than the previous value: The expression will fetch all issues whose *Customer Prioritization* value is greater than or equal to the entered value and smaller than or equal to the previous value in an ascending order.

Case 3 - The entered value is greater than the previous value: The expression will fetch all issues whose *Customer Prioritization* value is greater than or equal to the previous value and less than or equal to the entered value in a descending order.

2

### Field\*

Select the **Customer Prioritization** field and add the following expression.

```
(previousValue({issue.cfCusPrio}) = null ? {issue.cfCusPrio} + indexOf(^%
{00015}, issuesFromJQL("issuekey != " + %{00015} + " and \"Customer
Prioritization\" >= " + {issue.cfCusPrio} + " order by \"Customer
Prioritization\"")) : ({issue.cfCusPrio} < previousValue({issue.cfCusPrio}) ? ^
{issue.cfCusPrio} + 1 : ^{issue.cfCusPrio} - 1))
```

Depending on the previous cases, a number will be added or deducted from the *Customer Prioritization* value of each issue of the issue list.



## Related use cases

| Use case   | JWT feature | Workflow function   | Parser functions   | Label |
|--|-------------|---|--|-------|
| Add days skipping weekends and holidays to a Date Picker field |             | Update or copy field values                               | addTime()  |       |
| Add or remove request participants                             |             | Update or copy field values                               |  |       |
| Add watchers ignoring inactive users                           |             | Update or copy field values                               | usersInGroup() isActive() toString() filterByPredicate() |       |
| Assign important issues to the project lead                    |             | Update or copy field values                               |  |       |
| Assign issue to current user                                   |             | Update or copy field values                               |  |       |
| Automatically fill an insight custom field after a transition  |             | Update or copy field values                               |  |       |
| Change the assignee to the next evaluator                      |             | Update or copy field values                               | first()<br>toStringList()                                |       |
| Create an internal Service Management comment on linked issues |             | Update or copy field values                               |  |       |
| Keep parent issue's priority in sync                           |             | Update or copy field values                               |  |       |
| Keep track of important status updates                         |             | Update or copy field values<br>Fields required or changed | dateTimeToString()                                       |       |

|   |   |                             |  |
|---|---|-----------------------------|--|
| Match several values of a list                        |    | Update or copy field values | toString() distinct() filterByPredicate()  |
| Prioritize the issues globally                        |    | Update or copy field values | indexOf() previousValue() issuesFromJQL()  |
| Set a date on the same week day on alternate weeks    |    | Update or copy field values | addDays() nextDayOfTheWeek() modulus() weekOfTheYear() datePart()                      |
| Set a Date Picker field to the nth day of the month   |    | Update or copy field values | dayOfTheMonth() lastDayOfTheMonth() addDays()  |
| Set assignee based on a former assignee               |    | Update or copy field values | previousValue()  |
| Set Due Date with latest value among sub-tasks        |    | Update or copy field values | fieldValue()<br>max()<br>siblingSubtasks()   |
| Set fix version based on its start and release date   |    | Update or copy field values | toString() textOnStringList() unreleasedVersions() startDates() releaseDates() first() |
| Set the next fix version                              |    | Update or copy field values | floor() toNumber() substring() length()  |
| Shorten the summary to a maximum number of characters |  | Update or copy field values | substring() length()   |

If you still have questions, feel free to refer to our [support](#) team.