

Jira Workflow Toolbox



Jira Workflow Toolbox provides a rich set of **conditions**, **validators** and **post-functions** for designing complex workflows in Jira.

Main features

[Conditions / Validators](#)

[Post Functions](#)

[Custom Fields](#)

[Expression Parser](#)

[Scheduler](#)

Additional features

[Virtual Fields](#)

[Project Properties](#)

[Test your expressions](#)

Additional Information

[Marketplace Listing](#)

[Support](#)

[Version History](#)

[License FAQ](#)

News

Labeled content



[JWT Release Notes 2.6.2](#)

Decadis AG posted on Aug 16, 2019

[Direct download from Marketplace](#)

Bug fixes

- **Last comment's visibility restrictions** did not return the **Service Desk** visibility of a transition comment
- Value type **Standard** did not work properly when setting fields in the **Create issues** post function
- **Disabling and enabling** the app could produce a "**Service proxy has been destroyed**" error

Find more release notes in our [Version History section](#).



[JWT Release Notes 2.6.1](#)

Decadis AG posted on Jul 25, 2019

[Direct download from Marketplace](#)

Bug fixes

- Setting **Clear field value** as **value type** in distinct post functions did not work properly for version and component fields
- Post functions stopped executing when trying to access a **non-existent (e.g. deleted) custom field**
- Executing **Jira Workflow Toolbox** post functions via the **Workflow Enhancer for Jira** app did not work properly

Find more release notes in our [Version History section](#) .



JWT Release Notes 2.6.0

Decadis AG posted on Jul 15, 2019

[Direct download from Marketplace](#)

Improvements

- The **Move issues** post function now supports **parsed text** to be used to determine **target project**, **issue type** and **status** selection
- The **Move issues** post function now shows custom error message on the screen when issues **can't be moved due to incompatibility**
- The **expression parser** and the **field injector** now support the **Resolution Date**
- The **Send an Email** post function now supports mixed parsing mode when using personalized mode

New expressions in the parser

| Function | Returned Value |
|--|--|
| setBoolean (string variable_name , boolean value) : boolean | Creates a variable named variable_name for storing a boolean value, and assigns it a value , which is also returned in order to be used within an expression. Example: <code>setBoolean("myBoolean", true)</code> |
| getBoolean (string variable_name) : boolean | Returns the value stored in a boolean variable named variable_name , which was previously created using the setBoolean() function. Example: <code>getBoolean("myBoolean")</code> |
| setNumber (string variable_name , number value) : number | Creates a variable named variable_name for storing a number, and assigns it a value , which is also returned in order to be used within an expression. Example: <code>setNumber("myNumber", 100)</code> |
| getNumber (string variable_name) : number | Returns the value stored in a numeric variable named variable_name , which was previously created using the setNumber() function. Example: <code>getNumber("myNumber")</code> |
| setString (string variable_name , string value) : string | Creates a variable named variable_name for storing a string, and assigns it a value , which is also returned in order to be used within an expression. Example: <code>setString("myString", "Hello World!")</code> |
| getString (string variable_name) : string | Returns the value stored in string variable named variable_name , which was previously created using the setString() function. Example: <code>getString("myString")</code> |
| setNumberList (string variable_name , number list value) : number list | Creates a variable named variable_name for storing a number list, and assigns it a value , which is also returned in order to be used within an expression. Example: <code>setNumberList("myNumberList", [1, 2, 3])</code> |
| getNumberList (string variable_name) : number list | Returns the value stored in number list variable named variable_name , which was previously created using the setNumberList() function. Example: <code>getNumberList("myNumberList")</code> |
| setStringList (string variable_name , string list value) : string list | Creates a variable named variable_name for storing a string list, and assigns it a value , which is also returned in order to be used within an expression. Example: <code>setStringList("myStringList", ["Hello", "World"])</code> |

A-O

[calculated-field](#)
[condition](#)
[custom-field](#)
[example](#)
[faq](#)
[function](#)
[issue-links](#)
[issue-types](#)

P-U

[parser](#)
[post-function](#)
[project-properties](#)
[release-notes](#)
[schedule](#)
[sub-task](#)
[support](#)
[transition](#)

V-Z

[validator](#)
[virtual-fields](#)
[work-log](#)

Recently Updated Pages

[JWT Release Notes 2.6.2](#)
updated Aug 16, 2019
[view change](#)
[JWT Release Notes 2.6.0](#)
updated Aug 16, 2019
[view change](#)
[Functions to temporarily store and retrieve values](#)
updated Aug 02, 2019
[view change](#)
[Expression Parser](#)
updated Aug 02, 2019
[view change](#)
[JWT Release Notes 2.6.1](#)
updated Aug 01, 2019
[view change](#)
[Examples of Issue List expressions](#)
updated Aug 01, 2019
[view change](#)
[JWT Release Notes 2.5.0](#)
updated Jul 15, 2019
[view change](#)
[JWT 2.6](#)
updated Jul 15, 2019
[view change](#)
[Version History](#)
updated Jul 15, 2019
[view change](#)
[Move issues](#)
updated Jul 15, 2019
[view change](#)

| | |
|--|--|
| getStringList (string variable_name) : string list | Returns the value stored in string list variable named variable_name , which was previously created using the setStringList() function. Example: <code>getStringList("myStringList")</code> |
| setIssueList (string variable_name , issue list value) : issue list | Creates a variable named variable_name for storing an issue list, and assigns it a value , which is also returned in order to be used within an expression. Example: <code>setIssueList("myIssueList", ["KEY-1", "KEY-2"])</code> |
| getIssueList (string variable_name) : issue list | Returns the value stored in issue list variable named variable_name , which was previously created using <code>setIssueList()</code> function. Example: <code>getIssueList("myIssueList")</code> |
| weekOfTheYear (number t , number firstDayOfTheWeek , number minimalDaysInFirstWeek , time zone time_zone) : number | Returns the week of the year of the date-time t in a certain time_zone . The parameter firstDayOfTheWeek represents the first day of the week, e.g.: { SUNDAY } in the U.S., and { MONDAY } in Germany. The parameter minimalDaysInFirstWeek represents the minimal number of days required in the first week of the year, e.g., if the first week is defined as the one that contains the first day of the first month of the year, value 1 should be used. If the minimal number of days required must be a full week (e.g. all days of the week need to be in that year), value 7 should be used. Example: <code>weekOfTheYear(2023/01/03, {SUNDAY}, 1, LOCAL)</code> returns 1. Example: <code>weekOfTheYear(2023/01/03, {MONDAY}, 1, LOCAL)</code> returns 2. Example: <code>weekOfTheYear(2023/01/03, {MONDAY}, 7, LOCAL)</code> returns 1. |
| dayOfTheYear (number t , time zone time_zone) : number | Returns the day of the year of date-time t in a certain time_zone , e.g. for January 1st the value returned will be 1. Example: <code>dayOfTheYear(2019/02/01, LOCAL)</code> returns 32 |
| timeInValue (string field field , boolean expression predicate) : number | Returns the number of milliseconds a string field with code <code>{nnnnn}</code> of the current issue has had a value satisfying a boolean expression predicate , where the string value of the field with code <code>{nnnnn}</code> is represented by <code>^%</code> . Example: <code>timeInValue({00000}, ^% ~~ "ERROR" OR ^% ~~ "WARNING")</code> returns the number of milliseconds the field summary (field code <code>{00000}</code>) of the current issue has contained any of the words "ERROR" or "WARNING" , ignoring the case. Example: <code>timeInValue({00094}, count(toStringList(^%, ",") > 1)</code> returns the number of milliseconds the field components (field code <code>{00094}</code>) of the current issue has contained more than one selected component . Example: <code>timeInValue({00017}, ^% in ["Critical", "High"])</code> returns the number of milliseconds the field priority (field code <code>{00017}</code>) of the current issue has had a value of Critical or High . |
| timeInValue (number field field , boolean expression predicate) : number | Returns the number of milliseconds a number or date-time field with code <code>{nnnnn}</code> of the current issue has had a value satisfying a boolean expression predicate , where the numeric value of the field with code <code>{nnnnn}</code> is represented by <code>^</code> . Example: <code>timeInValue({00012}, ^ != null)</code> returns the number of milliseconds the field Due date (field code <code>{00012}</code>) of the current issue has had a value. Example: <code>timeInValue({10001}, ^ >= 5 AND ^ <= 10)</code> returns the number of milliseconds a hypothetical numeric field called Passengers (field code <code>{10001}</code>) of the current issue has remained between 5 and 10. Example: <code>timeInValue({10001}, modulus(^, 2) = 0)</code> returns the number of milliseconds a hypothetical numeric field called Passengers (field code <code>{10001}</code>) of the current issue has had an even value (2, 4, 6,...). |

| | |
|--|--|
| <p>timeInValue(string field field, issue list issues, boolean expression predicate): number</p> | <p>Returns the sum of milliseconds a string field with code <code>{nnnnn}</code> has had a value satisfying a boolean expression predicate in distinct issues, where the string value of the field with code <code>{nnnnn}</code> is represented by <code>^</code>.</p> <p>Example: <code>timeInValue({00000}, subtasks(), ^% ~~ "ERROR" OR ^% ~~ "WARNING")</code> returns the sum of milliseconds the field summary (field code <code>{00000}</code>) of all sub-tasks of the current issue have contained any of the words "ERROR" or "WARNING", ignoring the case.</p> <p>Example: <code>timeInValue({00094}, epic(), count(toStringList(^%, ",") > 1))</code> returns the number of milliseconds the field components (field code <code>{00094}</code>) in a linked Epic issue have contained more than one selected component.</p> <p>Example: <code>timeInValue({00017}, filterByIssueType(linkedIssues(), "Bug, New Feature"), ^% in ["Critical", "High"])</code> returns the sum of milliseconds all linked Bugs and New Features of the current issue have had a priority (field code <code>{00017}</code>) value of Critical or High.</p> |
| <p>timeInValue(number field field, issue list issues, boolean expression predicate): number</p> | <p>Returns the sum of milliseconds a number or date-time field with code <code>{nnnnn}</code> has had a value satisfying a boolean expression predicate in distinct issues, where the numeric value of the field with code <code>{nnnnn}</code> is represented by <code>^</code>.</p> <p>Example: <code>timeInValue({00012}, subtasks(), ^ != null)</code> returns the number of milliseconds the field Due Date (field code <code>{00012}</code>) of all sub-tasks of the current issue has had a value.</p> <p>Example: <code>timeInValue({10001}, epic(), ^ >= 5 AND ^ <= 10)</code> returns the number of milliseconds a hypothetical numeric field called Passengers (field code <code>{10001}</code>) of an Epic issue has had a value between 5 and 10.</p> <p>Example: <code>timeInValue({10001}, filterByIssueType(linkedIssues(), "Bug, New Feature"), modulus(^, 2) = 0)</code> returns the number of milliseconds a hypothetical numeric field called Passengers (field code <code>{10001}</code>) has had an even value in any linked Bug or New Feature.</p> |
| <p>timeInValue(string field field, boolean expression predicate, string schedule_name, time Zone time_zone): number</p> | <p>Returns the number of milliseconds a string field with code <code>{nnnnn}</code> of the current issue has had a value satisfying a boolean expression predicate, where the string value of the field with code <code>{nnnnn}</code> is represented by <code>^</code>. The time being calculated by this function is only counted during a defined schedule with name schedule_name for time zone time_zone.</p> <p>Example: <code>timeInValue({00000}, ^% ~~ "ERROR" OR ^% ~~ "WARNING", "my_schedule", LOCAL)</code> returns the number of milliseconds the field summary (field code <code>{00000}</code>) of the current issue has contained any of the words "ERROR" or "WARNING", ignoring the case, within a schedule named my_schedule for the server's default time_zone.</p> <p>Example: <code>timeInValue({00094}, count(toStringList(^%, ",") > 1, "my_schedule", LOCAL)</code> returns the number of milliseconds the field components (field code <code>{00094}</code>) of the current issue has contained more than one selected component, within a schedule named my_schedule for the server's default time_zone.</p> <p>Example: <code>timeInValue({00017}, ^% in ["Critical", "High"], "my_schedule", LOCAL)</code> returns the number of milliseconds the current issue has had a priority value of Critical or High (field code <code>{00017}</code>), within a schedule named my_schedule for the server's default time_zone.</p> |

| | |
|---|--|
| <p>timeInValue(number field field, boolean expression predicate, string schedule_name, time Zone time_zone) : number</p> | <p>Returns the number of milliseconds of a number or date-time field with code {nnnn} of the current issue has had a values satisfying a boolean expression predicate, where the numeric value of the field with code {nnnn} is represented by ^. The time being calculated by this function is only counted during a defined schedule with name schedule_name for timeZone time_zone.</p> <p>Example: <code>timeInValue({00012}, ^ != null, "my_schedule", LOCAL)</code> returns the number of milliseconds the field Due Date (field code {00012}) of the current issue has had a value, ignoring the case, within a schedule named my_schedule for the server's default time_zone.</p> <p>Example: <code>timeInValue({10001}, ^ >= 5 AND ^ <= 10, "my_schedule", LOCAL)</code> returns the number of milliseconds a hypothetical numeric field called Passengers (field code {10001}) of the current issue has had a value between 5 and 10, within a schedule named my_schedule for the server's default time_zone.</p> <p>Example: <code>timeInValue({10001}, modulus(^, 2) = 0, "my_schedule", LOCAL)</code> returns the number of milliseconds a hypothetical numeric field called Passengers (field code {10001}) in current issue has had an even value, within a schedule named my_schedule for the server's default time_zone.</p> |
| <p>timeInValue(string field field, issue list issues, boolean expression predicate, string schedule_name, time Zone time_zone) : number</p> | <p>Returns the sum of milliseconds a string field with code %{nnnnn} has had a value satisfying a boolean expression predicate in distinct issues, where the value of the field with code %{nnnnn} is represented by ^%. The time being calculated by this function is only counted during a defined schedule with name schedule_name for timeZone time_zone.</p> <p>Example: <code>timeInValue(%{00000}, subtasks(), ^% ~~ "ERROR" OR ^% ~~ "WARNING", "my_schedule", LOCAL)</code> returns the sum of milliseconds the fields summary (field code %{00000}) of all sub-tasks of the current issue have contained any of the words "ERROR" or "WARNING", ignoring the case, within a schedule named my_schedule for the server's default time_zone.</p> <p>Example: <code>timeInValue(%{00094}, epic(), count(toStringList(^%, ",") > 1, "my_schedule", LOCAL)</code> returns the number of milliseconds the field components (field code %{00094}) in the linked Epic issue has contained more than one selected component, within a schedule named my_schedule for the server's default time_zone.</p> <p>Example: <code>timeInValue(%{00017}, filterByIssueType(linkedIssues(), "Bug, New Feature"), ^% in ["Critical", "High"], "my_schedule", LOCAL)</code> returns the sum of milliseconds all linked Bugs and New Features of the current issue have had a priority (field code %{00017}) value of Critical or High, within a schedule named my_schedule for the server's default time_zone.</p> |
| <p>timeInValue(number field field, issue list issues, boolean expression predicate, string schedule_name, time Zone time_zone) : number</p> | <p>Returns the sum of milliseconds number or date-time field with code {nnnnn} has had a value satisfying a boolean expression predicate in distinct issues, where the numeric value of the field with code {nnnnn} is represented by ^. The time being calculated by this function is only counted during a defined schedule with name schedule_name for timeZone time_zone.</p> <p>Example: <code>timeInValue({00012}, subtasks(), ^ != null, "my_schedule", LOCAL)</code> returns the number of milliseconds the field Due date (field code {00012}) of all sub-tasks of the current issue have had a value, within a schedule named my_schedule for the server's default time_zone.</p> <p>Example: <code>timeInValue({10001}, epic(), ^ >= 5 AND ^ <= 10, "my_schedule", LOCAL)</code> returns the number of milliseconds a hypothetical numeric field called Passengers (field code {10001}) in the linked Epic issue has had a value between 5 and 10, within a schedule named my_schedule for the server's default time_zone.</p> <p>Example: <code>timeInValue({10001}, filterByIssueType(linkedIssues(), "Bug, New Feature"), modulus(^, 2) = 0, "my_schedule", LOCAL)</code> returns the number of milliseconds a hypothetical numeric field called Passengers (field code {10001}) has had an even value in any linked Bug or New Feature, within a schedule named my_schedule for the server's default time_zone.</p> |

| | |
|---|---|
| <p>fieldChangeTimes(string field field, boolean expression predicate) : number list</p> | <p>Returns the timestamps of when a string value of field with code <code>{nnnnn}</code> has changed satisfying a certain predicate that depends on the values of the field before and after the value change. The string value before the change is represented by <code>^0%</code>, and after the change by <code>^1%</code>. The timestamps are returned as a number list sorted in ascending order.</p> <p>Example: <code>fieldChangeTimes({00000}, ^0% !~~ "IMPORTANT" AND ^1% ~~ "IMPORTANT")</code> returns the list of timestamps when the word "IMPORTANT" has been added to the current issue's summary (field code <code>{00000}</code>) ignoring the case.</p> <p>Example: <code>fieldChangeTimes({00017}, ^0% = null AND ^1% != null)</code> returns the list of timestamps of when the issue's priority (field code <code>{00017}</code>) of the current issue has been set.</p> <p>Example: <code>fieldChangeTimes({00017}, ^0% not in ["Critical", "High"] AND ^1% in ["Critical", "High"])</code> returns the list of timestamps when current issue's priority (field code <code>{00017}</code>) has become Critical or High.</p> |
| <p>fieldChangeTimes(number field field, boolean expression predicate) : number list</p> | <p>Returns the timestamps of when a numeric / date-time value of field with code <code>{nnnn}</code> has changed satisfying a certain predicate that depends on the values of the field before and after the value change. The numeric value before the change is represented by <code>^0</code>, and after the change by <code>^1</code>. The timestamps are returned as a number list sorted in ascending order.</p> <p>Example: <code>fieldChangeTimes({00012}, ^0 < ^1)</code> returns the timestamps of when the Due date (field code <code>{00012}</code>) has been edited to a higher value.</p> <p>Example: <code>fieldChangeTimes({10001}, abs(^0 - ^1) / ^0 >= 0.25)</code> returns the timestamps of when a hypothetical numeric field called Passengers (field code <code>{10001}</code>) has been edited with a variation of at least 25% over its previous value.</p> |
| <p>fieldChangeTimes(string field field, issue list issues, boolean expression predicate) : number list</p> | <p>Returns the timestamps of when a string value of fields with code <code>{nnnnn}</code> in distinct parameter issues have changed satisfying certain predicate that depends on the values of the fields before and after the value change. The string value before the change is represented by <code>^0%</code>, and after the change by <code>^1%</code>. The timestamps are returned as a number list containing a sequence of sorted numeric values in ascending order for each parameter issue.</p> <p>Example: <code>fieldChangeTimes({00000}, subtasks(), ^0% !~~ "IMPORTANT" AND ^1% ~~ "IMPORTANT")</code> returns the list of timestamps of when the word "IMPORTANT" has been added to the summary (field code <code>{00000}</code>) of all current issue's sub-tasks, ignoring the case.</p> <p>Example: <code>fieldChangeTimes({00017}, epic(), ^0% = null AND ^1% != null)</code> returns the list of timestamps of when the issue priority (field code <code>{00017}</code>) of the current issue's epic has been set.</p> <p>Example: <code>fieldChangeTimes({00017}, linkedIssues("is blocked by"), ^0% not in ["Critical", "High"] AND ^1% in ["Critical", "High"])</code> returns the list of timestamps of when the priority (field code <code>{00017}</code>) in all blocking linked issues has become Critical or High.</p> |
| <p>fieldChangeTimes(number field field, issue list issues, boolean expression predicate) : number list</p> | <p>Returns the timestamps of when a numeric value of fields with code <code>{nnnnn}</code> in distinct parameter issues have changed satisfying a certain predicate that depends on the values of the fields before and after the value change. The numeric value before the change is represented by <code>^0</code>, and after the change by <code>^1</code>. The timestamps are returned as a number list containing a sequence of sorted numeric values in ascending order for each parameter issue.</p> <p>Example: <code>fieldChangeTimes({00012}, subtasks(), ^0 < ^1)</code> returns the timestamps of when the Due Date (field code <code>{00012}</code>) has been edited to a higher value in any of the current issue's sub-tasks.</p> <p>Example: <code>fieldChangeTimes({10001}, epic(), abs(^0 - ^1) / ^0 >= 0.25)</code> returns the timestamps when a hypothetical numeric field called Passengers (field code <code>{10001}</code>) in the current issue's epic has been edited with a variation of at least 25% over its previous value</p> |
| <p>lastFieldChangeTime(string field field) : number</p> | <p>Returns the timestamp of most recent value update of a field with code <code>{00000}</code>.</p> <p>Example: <code>lastFieldChangeTime({00000})</code> returns the timestamp of the last update of an issue's summary (field code <code>{00000}</code>).</p> |

| | |
|---|--|
| unreleasedVersionsBySequence() : string list | Returns a string list with the unreleased versions in the current project with the default order. Only non-archived versions are returned. The first version in the list is the lowest version in the version table. |
| releasedVersionsBySequence() : string list | Returns a string list with the released versions in the current project with the default order. Only non-archived versions are returned. The first version in the list is the lowest version in the version table. |

Bug fixes

- Using the **Move issues** post function on individual sub-tasks (without moving the parent) resulting resulted in **incompatible parent-child relationships**. Moving individual sub-tasks is **currently not available** ⚠
- Issues were not correctly moved to newly created Epics.
- Custom fields could be set to **values not available/valid in their respective field context** using the **Move issues** post function
- Setting the **"Sprint ID"** to a value **equal or greater than 1000** produced an error
- **Incorrect error messages** were logged when detecting **deleted custom fields**

Find more release notes in our [Version History section](#).

