

JWT Release Notes 2.1.32

[2015-06-17] Released [Jira Workflow Toolbox 2.1.32](#)

New features

- Functions for **version management** have been added to the parser:

FUNCTION	RETURNED VALUE
unreleasedVersions() : <i>string list</i>	Returns a string list with unreleased version names of current issue's project. Returned versions may be archived.
unreleasedVersions(string projects) : <i>string list</i>	Returns a string list with unreleased version names of projects in argument projects . Returned versions may be archived. Arguments projects is a comma separated list of project keys or project names .
releasedVersions() : <i>string list</i>	Returns a string list with released version names of current issue's project. Returned versions may be archived.
releasedVersions(string projects) : <i>string list</i>	Returns a string list with released version names of projects in argument projects . Returned versions may be archived. Arguments projects is a comma separated list of project keys or project names .
archivedVersions() : <i>string list</i>	Returns a string list with released version names of current issue's project. Returned versions may be archived.
archivedVersions(string projects) : <i>string list</i>	Returns a string list with released version names of projects in argument projects . Returned versions may either released or unreleased. Arguments projects is a comma separated list of project keys or project names .
latestReleasedVersion() : <i>string list</i>	Returns string with the name of the latest released version in current issue's project.
latestReleasedVersion(string projects) : <i>string list</i>	Returns string with the name of the latest released version among projects in argument projects . Returned versions may either released or unreleased. Arguments projects is a comma separated list of project keys or project names .
earliestUnreleasedVersion() : <i>string list</i>	Returns string with the name of the earliest unreleased version in current issue's project.
earliestUnreleasedVersion(string projects) : <i>string list</i>	Returns string with the name of the earliest unreleased version among projects in argument projects . Returned versions may either released or unreleased. Arguments projects is a comma separated list of project keys or project names .
count(number n, number list l) : <i>number</i>	Returns the number of times n appears in l . Example: <code>count(1, [1, 1, 2, 2, 1, 0])</code> returns 3.
count(string s, string list l) : <i>number</i>	Returns the number of times s appears in l . Example: <code>count("blue", ["blue", "blue", "red", "red", "blue", "green"])</code> returns 3.

Examples of Parser expressions

- Functions for **extracting text fragments** and **finding and replacing** literal substrings have been added to the parser:

FUNCTION	RETURNED VALUE
findPattern(string s, string regexp) : <i>string list</i>	Returns a string list with all substrings in argument s matching regular expression in string argument regexp .
findPatternIgnoreCase(string s, string regexp) : <i>string list</i>	Returns a string list with all substrings in argument s matching regular expression in string argument regexp . Evaluation of the regular expression is carried out in ignoring case mode.
findReplaceAll(string s, string find, string replacement) : <i>string list</i>	Returns a string with content of argument s where every occurrence of substring find has been replaced with string replacement .
findReplaceAllIgnoreCase(string s, string find, string replacement) : <i>string list</i>	Returns a string with content of argument s where every occurrence of substring find , ignoring the case, has been replaced with string replacement .
findReplaceFirst(string s, string find, string replacement) : <i>string list</i>	Returns a string with content of argument s where first occurrence of substring find has been replaced with string replacement .
findReplaceFirstIgnoreCase(string s, string find, string replacement) : <i>string list</i>	Returns a string with content of argument s where first occurrence of substring find , ignoring the case, has been replaced with string replacement .

- Functions for getting user names from user's **emails** or **full names** has been added to the parser:

FUNCTION	RETURNED VALUE
fullNameToUser(string fullName) : <i>string</i>	Returns a string with the name of a user whose full name is equal to argument fullName . Returned value is a string with a user name .
usersWithEmail(string email) : <i>string list</i>	Returns a string list with the user names of those users with emails equal to argument email . In case that only one user is expected, function first(string list) can be used to extract a string with its user name.

Improvements

- Added a new configuration parameter called "**Run as**" for selecting the user which will run a feature of the plugin. This parameter has been added to the following features:
 - Condition based on JQL query
 - Validation based on JQL query
 - Copy a parsed text to a field
 - Mathematical and date-time expression calculator
 - Set a field as a function of other fields
 - Create issue link
 - Break issue link
 - Write field on linked issues or subtasks
 - Read fields from linked issues or subtasks
 - Write field on issues returned by JQL query
 - [Read Field From Issues Returned By JQL Query](#)
 - Clone issue links
- Added new configuration parameter called "**Don't overwrite target field if it's already set**" to the following features:
 - Copy a parsed text to a field
 - Mathematical and date-time expression calculator
 - Set a field as a function of other fields
 - Write field on linked issues or subtasks
 - Write field on issues returned by JQL query
- **Optional prefixes** has been added to "[Set a field as a function of other fields](#)" post-function. These optional prefixes are single characters that can precede setting rules for changing somehow its behavior:
 - **a** : makes value part in setting rules to be parsed in **advanced parsing mode**.
 - **i** : in type 1 setting rules, makes regular expression to be evaluated in **ignore case mode**.
 - **l** : in type 1 setting rules, makes regular expression to be treated as a **literal string**.
- Conditional operator **?** : now also supports usage with **boolean arguments**.
- UI improvement thanks to the replacement of basic textarea editor with new advanced editor. Improved features are:
 - Log Work post-function.
 - Create issue link
 - Break issue link
 - Condition based on JQL query
 - Validator based on JQL query

Bug fixes

- Javascript error introduced in version **2.1.31** on "[Set a field as a function of other fields](#)" post-function, that makes automatic enabling/disabling of checkbox "**Full evaluation**" depending on selected "**Target field**" stopped working.
- Minor javascript bug in "**Condition based on JQL query**" and "**Validation based on JQL query**".
- Parse function **toStringList()** doesn't behave correctly with multi-valued fields containing options with **comma** characters.
- Setting multi-valued fields were options have comma characters were not possible. Now comma can be escaped with **'\'** character.
- Uncaught exception in "**Condition/Validation on linked issues**", "**Condition/Validation on subtasks**", "**Condition/Validation on a list of issue keys**", "**Read/Write field on linked issues or subtasks**" was configured with a **bad formed expressions** and when **uninitialized fields are encountered** in parameter "**Filtering by field values**".