JWT Release Notes 2.2.42

[2017-09-04] **Released Jira Workflow Toolbox 2.2.42**

New features

- New virtual field "**Available target statuses**" which returns a comma separated list with the names of the target statuses of all the transitions with origin in current issue status.
- **Jira Workflow Toolbox** parser now provides **case-insensitive** comparison operators for **strings** and **string lists**:

Case ignoring comparison operators

The following comparison operators are applicable to **String** and **String List** types. This operators have the peculiarity that ignores the case of the characters.

Operator	Meaning	Examples (all examples return `true`)

`=~`	equal to	`"HELLO" =~ "Hello"`
		`"up" =~ "UP"`
		`["blue", "red", "green"] =~ ["Blue", "RED", "Green"]`

`!=~`	not equal to	`" HELLO" !=~ "Hello"`
		`"up" !=~ "down"`
		`("up" !=~ "UP") = false`
		`["blue", "red"] !=~ ["Blue", "green"]`
		`["blue", "red"] !=~ ["Red", "BLUE"]`
		`(["blue", "red", "green"] !=~ ["Blue", "RED", "Green"]) = false`

`~~`	contains	`"Hello World!" ~~ "world"`, checks whether a string contains a substring. `"A small step for a man" ~~ "STEP"`, checks whether a string contains a substring. `["one", "two", "three"] ~~ ["TWO", "One"]`, checks whether a string list contains all the elements of another string list.
`!~~`	doesn't contain	`"Hello World!" !~~ "bye"`, checks whether a string doesn't contain a substring. `"A small step for a man" !~~ "big"`, checks whether a string doesn't contain a substring. `["one", "two", "three"] !~~ ["Four"]`, checks whether a string list doesn't contain one element of another string list. `(["one", "two", "three"] !~~ ["TWO"]) = false`
`in~`	is contained in	`"world" in~ "Hello World!"`, checks whether a substring is contained in another string. `"STEP" in~ "A small step for a man"`, checks whether a substring is contained in another string. `["TWO", "One"] in~ ["one", "two", "three"]`, checks whether all the elements of a string list are contained in another string list.
`not in~`	isn't contained in	`"bye" not in~ "Hello World!"`, checks whether a substring is not contained in another string. `"big" not in~ "A small step for a man"`, checks whether a substring is not contained in another string. `["Four"] not in~ ["one", "two", "three"]`, checks whether any of the elements of a string list are not contained in another string list. `(["TWO"] not in~ ["one", "two", "three"]) = false`
`any in~`	some element is in	`["blue", "violet"] any in~ ["Blue", "Red", "Green"]` `["Five", "One"] any in~ ["FOUR", "FIVE", "SIX"]`
`none in~`	no single element is in	`["Orange"] any in~ ["red", "blue", "green"]` `(["orange"] any in~ ["Red", "Orange"]) = false`

Applicable data types

Comparison operator	Boolean	Number	String	Number list	String list	Issue list	Multi-valued fields
`=~`	-	-	X	-	X	-	-
`!=~`	-	-	X	-	X	-	-
`~~`	-	-	X	-	X	-	-
`!~~`	-	-	X	-	X	-	-
`in~`	-	-	X	-	X	-	-
`not in~`	-	-	X	-	X	-	-
`any in~`	-	-	-	-	X	-	-
`none in~`	-	-	-	-	X	-	-

Improvements

- **Issue #775** - Virtual field "**Target status**", when used by conditions, returned a comma separated list with the names of the target statuses of all the transitions with origin in current issue status, but when used by validations and conditions only returned target status of current issue. Now, it behaves in conditions as in validations and post-functions.

WARNING!: If you are currently using field "**Target status**" in **conditions** (not validators nor post-functions), you should replace it with field "**Available target statuses**" in order to keep the exact same behavior.

- Support for comments in **Expression Parser**: Now we can insert one-line comments in our expressions (text, boolean, issue list and math/time) using the following syntax: `# This is a comment`.

Bug fixes

- **Issue #749** - Can't set components in parent issue from sub-tasks workflow
- Parse error when inserting a comment at the end of a **Schedule** definition.
- Post-function **Send an email** failed to replace the value of custom fields in the email's subject an body when used in "**Create Issue**" transition, and the issue creation was triggered by an email.
- **Issue #777** - Function `fieldHistory()` failed with certain fields when used in **Calculated Fields**.
- **Issue #774** - Thread synchronization problem that may cause some functionalities to stop working under heavy load.