

JWT Release Notes 2.2.39

[2017-07-05] Released [Jira Workflow Toolbox 2.2.39](#)

New features

1) Support for doing time calculations on Custom Schedules

Now, you will be able to do any time calculations on the very particular work-schedule(s) of your company or organization.

You can define one or more custom schedules at **Administration > Add-ons > Jira WORKFLOW TOOLBOX > Schedule** using [Schedules Definition Grammar](#), which is a very powerful language capable of defining schedules as complex as you might imagine.

Then you have the following functions available to do time calculations on your custom schedules, using any feature of the plugin (conditions, validations, post-functions and calculated fields):

Function	Returned value
inSchedule (number time_instant , string schedule_name , string additional_terms , timeZone time_zone) : boolean	Similar to previous function, but with extra parameter additional_terms , which is a string containing extra Schedule Definition clauses that will be attached to schedule with name schedule_name . This function can be used to include personal holidays to an existing schedule. Example without additional terms: <code>inSchedule(2017/12/04 9:00, "my_schedule", LOCAL)</code> returns true . Example with additional terms: <code>inSchedule(2017/12/04 9:00, "my_schedule", LOCAL)</code> returns false .
timeDifference (number higher_instant , number lower_instant , string schedule_name , timeZone time_zone) : number	Returns the number of milliseconds elapsed from lower_instant to higher_instant within schedule with name schedule_name for time_zone timezone. Example: <code>timeDifference(2017/12/04 10:01, 2017/12/01 01:00, "my_schedule", LOCAL)</code> returns 8 * {HOUR} + 31 * {MINUTE} . Example: <code>timeDifference(2017/12/04 17:00, 2017/12/04 14:00, "my_schedule", LOCAL)</code> returns 2 * {HOUR} + 30 * {MINUTE} .
timeDifference (number higher_instant , number lower_instant , string schedule_name , string additional_terms , timeZone time_zone) : number	Similar to previous function, but with extra parameter additional_terms , which is a string containing extra Schedule Definition clauses that will be attached to schedule with name schedule_name . This function can be used to include personal holidays to an existing schedule. Example without additional terms: <code>timeDifference(2017/12/05 18:00, 2017/12/01 9:00, "my_schedule", LOCAL)</code> returns 25 * {HOUR} . Example with additional terms: <code>timeDifference(2017/12/05 18:00, 2017/12/01 9:00, "my_schedule", "2017/12/04 {;}", LOCAL)</code> returns 15 * {HOUR} .
addTime (number base_instant , number offset , string schedule_name , timeZone time_zone) : number	Returns the time instant resulting of adding offset milliseconds to base_instant within schedule with name schedule_name for time_zone timezone. Example: <code>addTime(2017/12/01 01:00, 8 * {HOUR} + 31 * {MINUTE}, "my_schedule", LOCAL)</code> returns 2017/12/04 10:01 . Example: <code>addTime(2017/12/04 14:00, 2 * {HOUR} + 30 * {MINUTE}, "my_schedule", LOCAL)</code> returns 2017/12/04 17:00 .
addTime (number base_instant , number offset , string schedule_name , string additional_terms , timeZone time_zone) : number	Similar to previous function, but with extra parameter additional_terms , which is a string containing extra Schedule Definition clauses that will be attached to schedule with name schedule_name . This function can be used to include personal holidays to an existing schedule. Example without additional terms: <code>addTime(2017/12/01 9:00, 25 * {HOUR}, "my_schedule", LOCAL)</code> returns 2017/12/05 18:00 . Example with additional terms: <code>addTime(2017/12/01 9:00, 15 * {HOUR}, "my_schedule", "2017/12/04 {;}", LOCAL)</code> returns 2017/12/05 18:00 .

In the examples above we have used schedule "my_schedule", which definition in [Schedules Definition Grammar](#) is:

```
1 MON-THU {
2     08:30 - 15:30,
3     16:00 - 19:30;
4 }
5
6 FRI {
7     08:00 - 15:00;
8 }
```

Note that 2017/12/01 is Friday, and 2017/12/04 is Monday.

A typical usage example is: [Automatic work log with start and stop work transitions](#)

2) Other new functions added to the parser:

Function	Returned value
<code>issueIDFromKey(string issue_key) : string</code>	Returns the internal ID of issue with key issue_key . This function also admits a comma separated list of issue keys, and returns a comma separated list of internal IDs . Example: <code>issueIDFromKey("CRM-1")</code> returns "10001".
<code>issueKeyFromID(string issue_ID) : string</code>	Returns the issue key of issue with internal ID issue_ID . This function also admits a comma separated list of issue IDs , and returns a comma separated list of issue keys . Example: <code>issueIDFromKey("10001")</code> returns "CRM-1".
<code>timeZone(string timeZone_name) : timeZone</code>	Returns the timeZone whose name is represented by string timeZone_name . This function is useful to obtain a timeZone from a string, like the value of a Project Property . Example: <code>timeZone("DST")</code> returns DST timeZone .

Bug fixes

- [Issue #636](#) - Function `latestReleasedVersion()` failed when there are version with no release date.