the Jira guy
YEAH, I'M "THAT" GUY

Your source for tips, tricks, and how-to's
to get the most from your Atlassian tools

Jira App Review

Jira Misc Workflow Extensions (JMWE)

By Rodney Nissen

January 2021

PRODUCT VALIDATION

INTRODUCTION

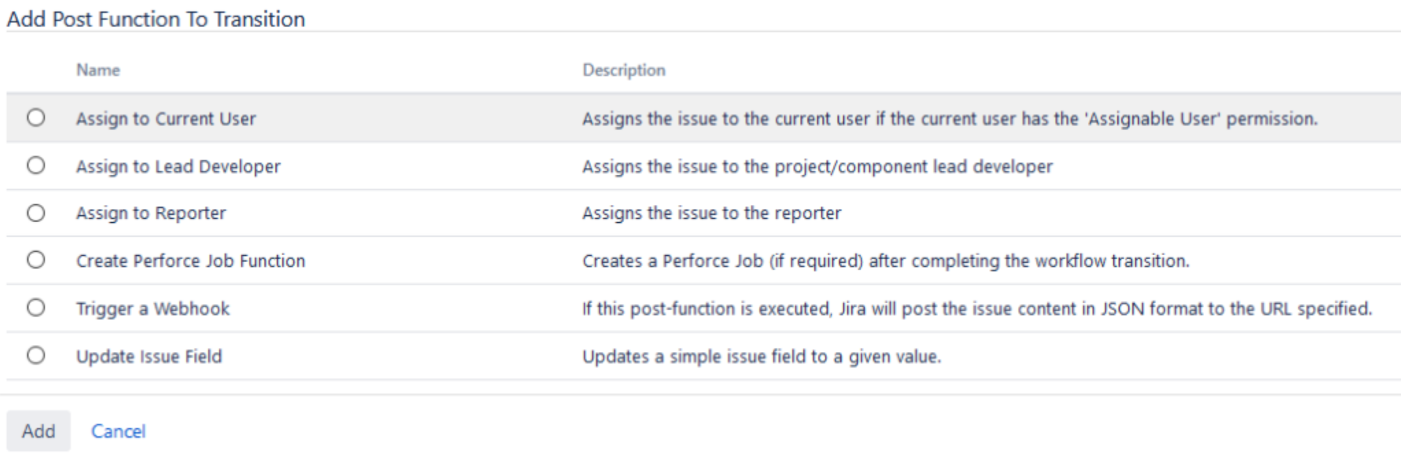
When it comes to the various Jira workflow apps, I consider them like various wrenches. They all essentially do something similar, but each one's unique design makes it suited to a slightly different application or use case. Below, I'll take a more in-depth look at what Jira Misc Workflow Extensions (JMWE) by Innovalog is, what problems it solves, what makes it unique.

Problem Statement and Solution

One of the most prominent concepts I had to get my head around as a Jira Admin is people work in Time. What do I mean by that? Well, People don't usually live off points alone. When estimating work, we do it either directly in person-hours or use some metric like Velocity to convert points, T-shirt sizes, or whatever value we use to a unit of time.

So the most important item to understand when setting up any automation is, "When does this thing happen?" And one of the most natural points in time for an Automation to execute is during a Status change. At this point, someone is already doing something to the issue, so if values need to change, this is likely where it should be.

Considering we are already in a Status change, otherwise known as a Transition, using a Transition Post Function is where I would first look to execute this Automation. However, the default list of Post Functions is a bit...puny.



Add Post Function To Transition

	Name	Description
○	Assign to Current User	Assigns the issue to the current user if the current user has the 'Assignable User' permission.
○	Assign to Lead Developer	Assigns the issue to the project/component lead developer
○	Assign to Reporter	Assigns the issue to the reporter
○	Create Perforce Job Function	Creates a Perforce Job (if required) after completing the workflow transition.
○	Trigger a Webhook	If this post-function is executed, Jira will post the issue content in JSON format to the URL specified.
○	Update Issue Field	Updates a simple issue field to a given value.

What!? That's it!?

Without any apps, there are six post functions in Jira. And the one that looks like it would be the most useful, "Update Issue Field," only works on built-in Fields, not any Custom fields! Conditions are a bit better off, with some twelve options. But poor Validators only have two options, both based on whether you or someone else has permissions.

This lack of options is the problem that JMWE tries to solve. It gives you many more options for post functions, conditions, and validators, letting you automate aspects of your workflow. They even have a method to create functionality if you need something they don't have! But we'll cover that in a moment.

Now to be clear, JMWE was my first workflow App. I inherited my first Jira instance, and the Admin before me had kept a well-maintained system. In it, they had installed JMWE, and it took me an embarrassingly long time to figure out Jira didn't come with all the nifty workflow functions out of the box. But given it was my first Workflow App, JMWE holds a special place in my heart.

So given that, why don't we look at what features JMWE offers and what you can do with them.

JMWE Features

30+configurable conditions, validators & post-functions

This feature is the meat-and-potatoes of the App, in my opinion. If you are getting this App, these configurable workflow objects will be what makes you choose this over other similar Apps.

I like the analogy of them being different kinds of wrenches. Because workflow Apps all have slightly different functions, they are suitable for different situations.

When I'm trying to narrow down which one to recommend for purchase, I look to the specific use-case I am trying to solve (a.k.a. "What's the ask?"), then figure out what post functions, validators, and conditions are available to solve it. That will usually help me figure out which one is appropriate for my case. So, what's available with JMWE? A lot!

Conditions:

Server/Data Center

- Current Status Condition
- Hide transition
- Previous Status Condition
- Related Issues Condition
- Related Issues Status Condition
- **Scripted (Groovy) Condition**
- Separation of Duties Condition

Cloud

- Current Status Condition
- Linked Issues Condition
- Linked Issue Status Contition
- **Build-your-own (Scripted) Condition**

Validators:

Server/Data Center

- Comment Required Validator
- Field has been modified Validator
- Field has single value Validator
- Fields Required Validator
- Parent Status Validator
- Previous Status Validator
- Related Issues Status Validator
- Related Issues Validator
- **Scripted (Groovy) Validator**

Cloud

- Field Required Validator
- Linked Issues Validator
- Linked Issues Status Validator
- **Build-your-own (scripted) Validator**

Post Functions:

Server/Data Center

- Add field value to parent issue
- Assign to last role member
- Assign to role member
- Clear field(s)
- Clear field(s) of related issues
- Comment issue
- Comment related issues
- Copy field value from parent issue
- Copy field value from related issues
- Copy field value to parent issue
- Copy field value to related issues
- Copy value from field to field
- Create / Clone issue(s)
- Display message to user
- Email issue
- Increase value of field
- Link issues to the current issue
- Return to Previous Status
- **Scripted (Groovy) operation on issue**
- Set field value
- Set field value from User Property value
- Set field value of related issues
- Set issue security level based on current user's project role
- Set issue, user or project Entity Property value
- Transition current issue
- Transition parent issue
- Transition related issues
- Unlink issues from the current issue

Cloud

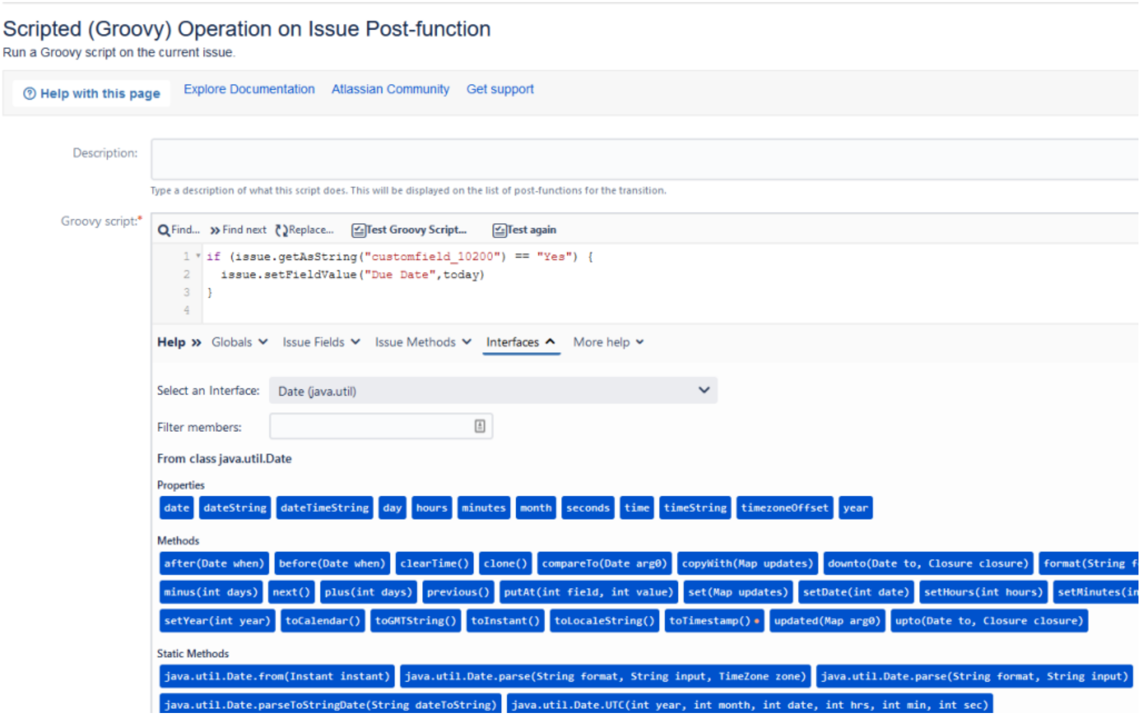
- Assign to last role member
- Assign to role member
- **Build-your-own (scripted) Post-function**
- Clear fields
- Clear fields of linked issues
- Comment issue(s)
- Comment linked issues
- Copy comments to related issues
- Copy field value from linked issues
- Copy field value from parent issue
- Copy field value to linked issues
- Copy field value to parent issue
- Copy value from field to field
- Create issue(s)
- Delete issue
- Display Message to User
- Email issue
- Increase value of field
- Link issues to the current issue
- Sequence of post-functions
- Set entity property value
- Set field value
- Set field value from User Entity Property value
- Set field value of linked issues
- Shared Action post-functions
- Transition issue(s)
- Transition linked issues
- Transition parent issue
- Unlink issues from the current issue

Build-your-own (scripted) extensions with JMWE's simple editor & tester

So, I alluded to these in my list above, but this is a new feature to me. This feature allows you to create Groovy scripts that run as either a validator, condition, or post function. What can these do? Well, that depends on how good your Groovy skills are. But this does take the App from being an otherwise curated collection of post functions to something you can fit to your needs.

Now, I've made no secret that my Groovy skills are non-existent. However, the editor they provide tries to help you out as much as possible, meaning that even a novice like me can get in there and start experimenting and working. Then it's just a matter of my favorite programming methodology: Change something and see what happens.

Now, I feel it important to mention a note here about the Cloud version of this feature. Unfortunately, we do not have access to a Groovy editor. Instead, we get access to Nunjucks and Jira Expressions. However, the experience is very similar otherwise, so other than language we are good to go!



Scripted (Groovy) Operation on Issue Post-function

Run a Groovy script on the current issue.

Description:

Type a description of what this script does. This will be displayed on the list of post-functions for the transition.

Groovy script:*

```
if (issue.getAsString("customfield_10200") == "Yes") {
  issue.setFieldValue("Due Date",today)
}
```

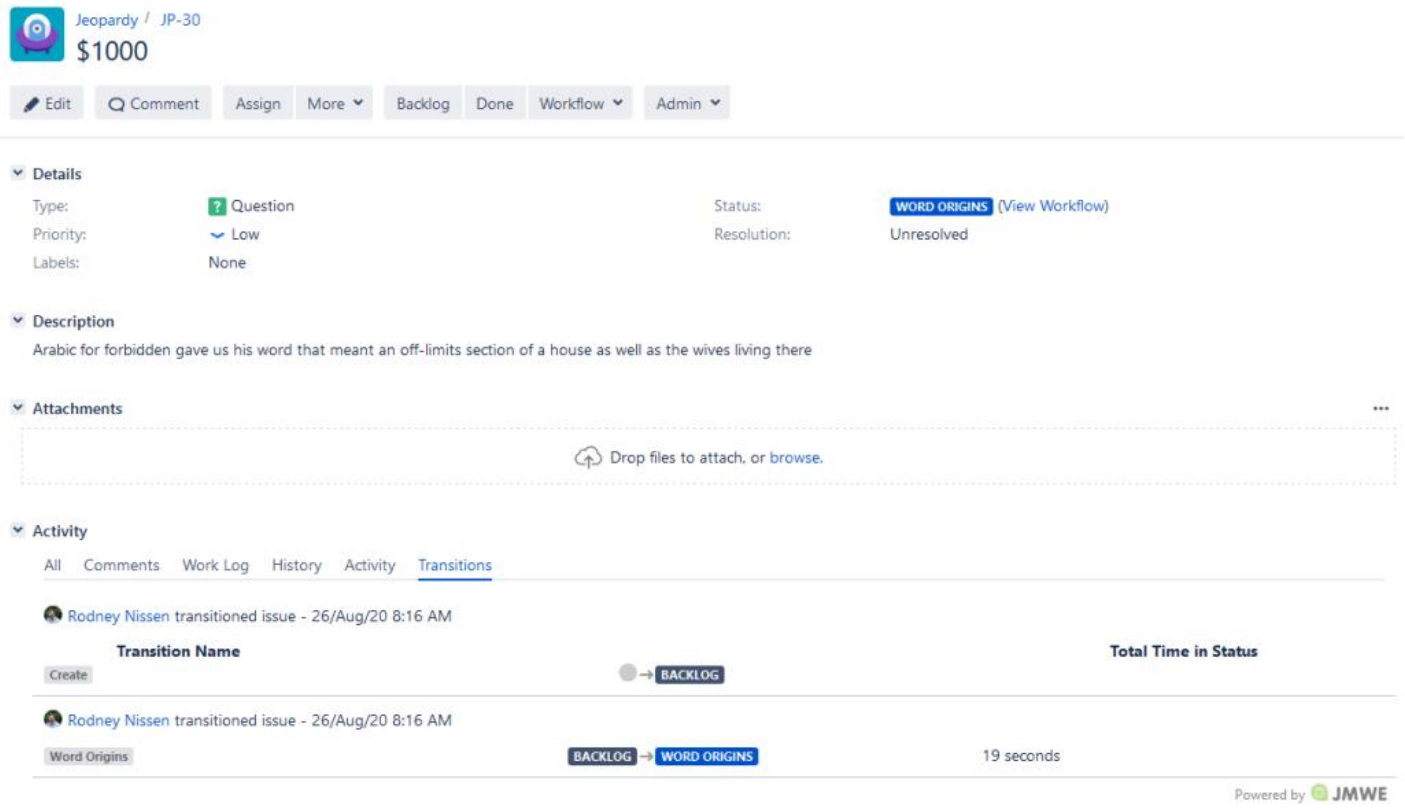
Select an Interface: Date (java.util)

Filter members:

From class java.util.Date

View Issue Transitions tab

This feature creates a new tab on your issues page called “Transitions”. Here you can see precisely when different transitions occurred, who did it, and when that transition took place. I know this is a top question, and digging through the “History” tab can be a bit of a chore. I love how this looks. While not a deal-maker in and of itself, it’s something to make the App that much more appealing.



$1000

Type: Question

Priority: Low

Labels: None

Status: WORD ORIGINS (View Workflow)

Resolution: Unresolved

Arabic for forbidden gave us his word that meant an off-limits section of a house as well as the wives living there

Drop files to attach, or browse.

Rodney Nissen transitioned issue - 26/Aug/20 8:16 AM

Transition Name		Total Time in Status
Create	→ BACKLOG	

Rodney Nissen transitioned issue - 26/Aug/20 8:16 AM

Word Origins	BACKLOG → WORD ORIGINS	19 seconds

Shared Actions – reuse post-function configurations

This. This feature. There are times that I’ll have to use a post function in several parts of the workflow. And nothing changes on the post function, other than it’s on a different transition.

Having the ability to set it once and share it amongst transition would save me a lot of time in these scenarios. And anything that saves me time makes me a happy Jira Admin.

Unfortunately, this feature appears to only be for Cloud at the moment, but I'm hoping they'll consider porting it over to the Server and Data Center Versions – because I neeeeed it!

Scheduled Actions

This feature is another Cloud-Only feature at the moment, but I figured I'd cover it as well. As I stated at the start of the piece, Automations of any sort have to have a time component figured out. While a status change as a trigger makes sense a lot of the time, it's not universal. So the Cloud version of JMWE gives you another option: Just trigger it off time. Hence the scheduled action. I've had times where I had an even that would need to be run every first of the month for reporting purposes. This feature would help out with just that.

ANALYSIS

What this App does Well

While I see this App is branching out, it doesn't forget what it does well and sticks to that. Sometimes you need a specialty knife designed for a specific purpose and is just really good at that purpose. It has a varied and broad set of Post Functions, Conditions, and Validators. And if you manage to find a use case it doesn't cover, you have the option to script your own. And while it has its own extra functionality, like the "Transitions" view on Issues, it doesn't forget what it's here to do.

What this App could work on

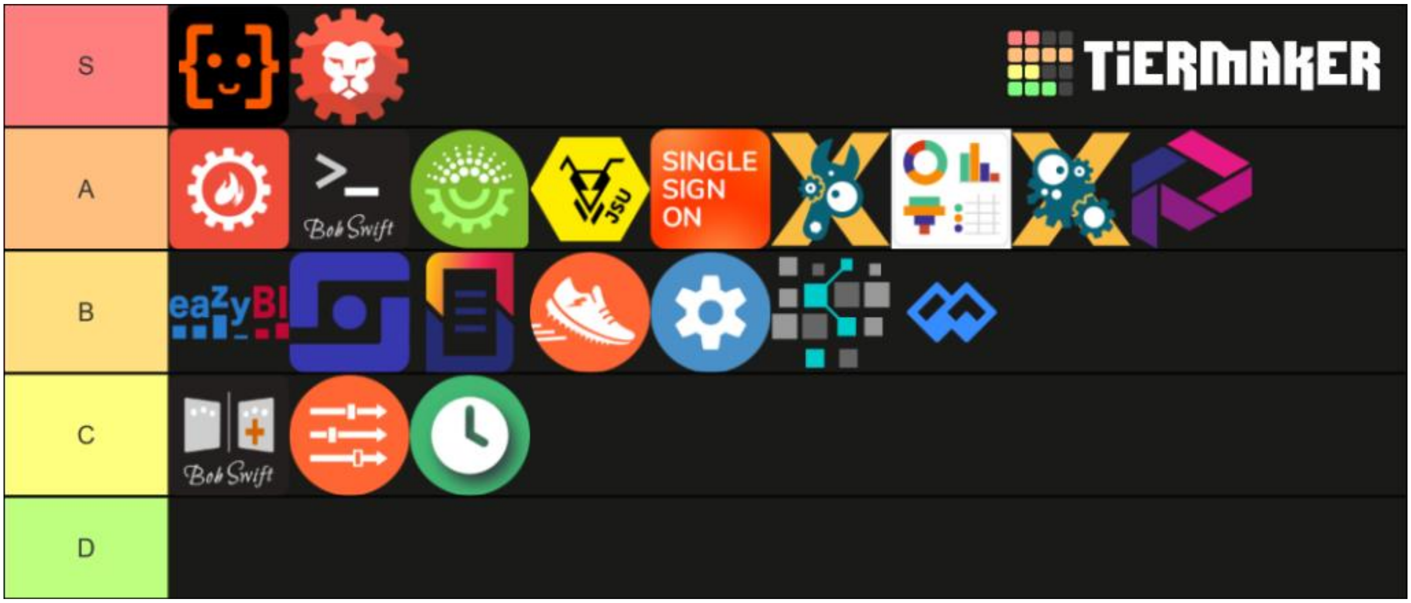
Well, that's easy enough. As I stated above, the Cloud version of this App has several features that I'd love to see in the Server/DC version as well. I know that full feature parity may never be possible, but I would love to see the Shared Actions in my Server instance.

Would I recommend this App?

Well, Yes! I already have once, after all. Look, every instance can use a Workflow App of one flavor or another. And while I don't believe in a "One size fits all" approach, you definitely can't go wrong with JMWE.

JMWE's Tier Rank

Honestly, this is the most straightforward ranking I've had – because I've already done it! This App, if it's in your mix, is a must-have. It's not a groundbreaking game-changer, but it does fill an almost vital niche. Given that, I have no problem giving it an "A" Rank....again.



For more information, visit Innovalog.com/jmwe.

ABOUT THE AUTHOR

Rodney Nissen is an experienced Atlassian Administrator and Consultant, but he started out right where you might be. He interviewed to be a Linux Administrator, and day one was given a JIRA Instance saying “Congrats, this is yours too!” He now has experience working with both Software Development and ITSM teams, as well as a creating a variety of non-standard business workflows and processes within JIRA.

Rodney currently holds an Atlassian Certified Master status, backed up by the following Certifications:

- Atlassian Certified Jira Administrator
- Atlassian Certified Confluence Administrator
- Atlassian Certified System Administrator
- Atlassian Certified in Agile Development with Jira Software

ABOUT PRODUCT VALIDATION REPORTS

The Product Validation Report is produced by TheJiraGuy.com to deliver independent validation of vendor product capability and provides insights on how the product compliments customer's native Atlassian Jira deployments. This report should not be treated as a replacement for the evaluation process that you would normally conduct before making a purchasing decision, but it will provide independent information on the product to help you with your decision. This report includes opinions of the reviewer, and TheJiraGuy.com is not responsible for any issues that may arise during your usage of the product. This research was commissioned by the vendor. Visit https://thejiraguy.com for other Product Validation Reports.