Doing Automation

Jira can be very efficient, but out of the box no real automation exists..

Interfacing

These days several other ways exist:

Script/Type	Comments
Automation App	Almost standard now-a-days... (and You get it automatically in cloud) Some Apps, like JSU App and Script Runner App offer parts of automation
Atlassian CLI	Can be "integrated via shell calls from script, powershell and other programming languages. As it is java based, it is virtually platform independant, the main downside is that is is slow.
JIRA JellyScript	See Atlassian (old) samples at: https://confluence.atlassian.com/display/JIRA/Jelly+Escalation DEPRECATED
Script Runner App	See some samples at library.adaptavist.com
REST API	Interfacing with JIRA through webservices - REST, The SOAP and XML-RCP is deprecated

Scheduling

The internal JIRA Scheduler implemented in Services really sucks - only setting is a "pr. minute" count called delay, and this:

- Resets when JIRA is restarted
- Is relative from JIRA Start, so You can't schedule "at midnight"

Hence, use another Scheduler if the timing must be right, like tweaking my Making a free JIRA Scheduler or buy TheScheduler Plugin

Or create Your own....

Hiding Auto Transitions

Problem

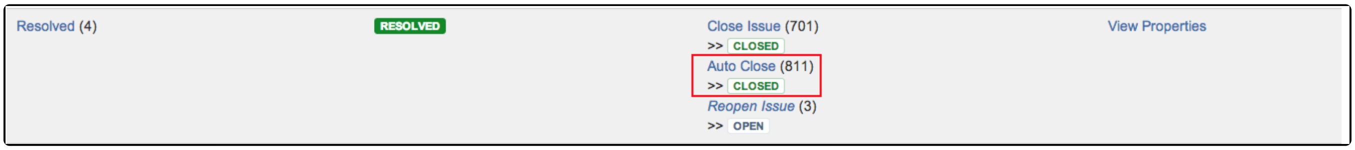
Often, You want the Automation and the User to perform the same transitions, or the Automation to do extra "invisible" stuff, - This can be tricky, also because any validations not handled in the transitionscript will cause a fail:

```
2014-05-05 22:46:30,674 QuartzScheduler_Worker-3 INFO ServiceRunner    CloseResolvedIssues [jira.jelly.service.
JellyService] JellyService.run
2014-05-05 22:46:30,803 QuartzScheduler_Worker-3 ERROR anonymous    CloseResolvedIssues [jira.jelly.service.
JellyService] file:/pack/jira-jelly/support-closeresolved.xml:48:125: <jira:TransitionWorkflow> The following
problems were found:
customfield_12922: ITIL Tag is required.
org.apache.commons.jelly.JellyTagException: file:/pack/jira-jelly/support-closeresolved.xml:48:125: <jira:
TransitionWorkflow> The following problems were found:
customfield_12922: ITIL Tag is required.
        at com.atlassian.jira.jelly.tag.JellyUtils.processErrorCollection(JellyUtils.java:24)
        at com.atlassian.jira.jelly.tag.issue.TransitionWorkflow.doTag(TransitionWorkflow.java:133)
        at org.apache.commons.jelly.impl.TagScript.run(TagScript.java:247)
```

Solution

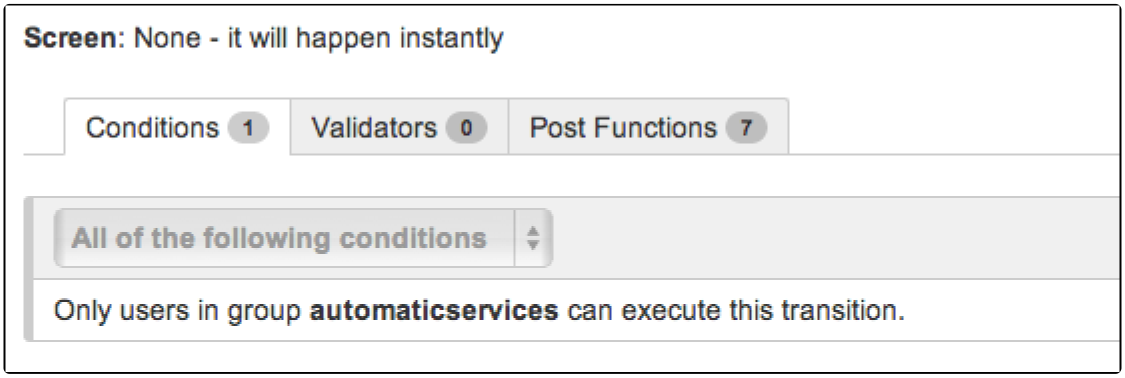
Create an "automationservices" group

Create a Transition that can only be done by users in that group:



Resolved (4)	RESOLVED	Close Issue (701) >> CLOSED Auto Close (811) >> CLOSED Reopen Issue (3) >> OPEN	View Properties

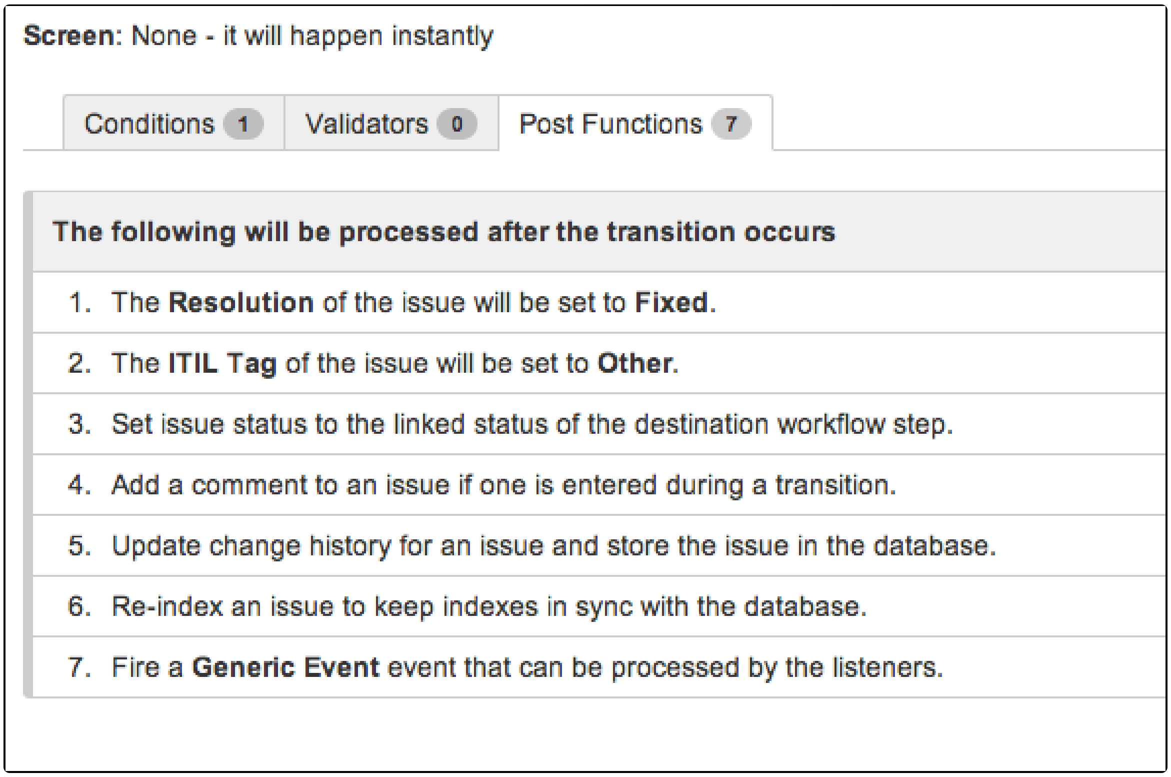
Hidden:



Screen: None - it will happen instantly

Only users in group **automaticservices** can execute this transition.

Setting fields correctly:



Screen: None - it will happen instantly

The following will be processed after the transition occurs

1. The **Resolution** of the issue will be set to **Fixed**.
2. The **ITIL Tag** of the issue will be set to **Other**.
3. Set issue status to the linked status of the destination workflow step.
4. Add a comment to an issue if one is entered during a transition.
5. Update change history for an issue and store the issue in the database.
6. Re-index an issue to keep indexes in sync with the database.
7. Fire a **Generic Event** event that can be processed by the listeners.

If You encounter this error in the logs:

```
2014-06-10 08:01:43,716 QuartzScheduler_Worker-2 WARN automaticservices    CloseinactiveSupportIssues
[jelly.tags.log.WarnTag] Inactivating issue MYPROJECT-151
2014-06-10 08:01:43,825 QuartzScheduler_Worker-2 ERROR anonymous    CloseinactiveSupportIssues [jira.
jelly.service.JellyService] file:/pack/jira-jelly/support-closeinactivate.xml:48:145: <jira:
TransitionWorkflow> Field 'resolution' can not be set on action with no screen
org.apache.commons.jelly.JellyTagException: file:/pack/jira-jelly/support-closeinactivate.xml:48:145:
<jira:TransitionWorkflow> Field 'resolution' can not be set on action with no screen
```

This is typically due to the script is a copy/similarity of https://confluence.atlassian.com/display/JIRA/Jelly+Escalation and the line:

```
<jira:TransitionWorkflow key="${issue.key}" user="${workflowUser}" workflowAction="${workflowStep}"
comment="${comment}" resolution="Customer Timeout"/>
```

Setting a field on the Jelly "jira:TransitionWorkflow" requires a screen with the field, therefore - set it in the Post Function

A real-life example:

I created an onComment Listener in Groovy, that transistet Issue when a Comment came in, a code snippet is:

```
workflowTransitionUtil.setUsername(currentuser);
workflowTransitionUtil.setAction (transition)
// Validate and transition issue
workflowTransitionUtil.validate();
workflowTransitionUtil.progress();
```

This code is valid, but in the "Post Function" of the Transition was:

```
Fire a Issue Commented event that can be processed by the listeners.
```

Hence, the script re-fired itself, which is not good...