



**PANDORA**FMS  
E N T E R P R I S E

Issue Management in JIRA

© Ártica Soluciones Tecnológicas 2005-2019

## INDEX

<b>Introduction</b>	<b>3</b>
<b>Compatibility Matrix</b>	<b>3</b>
<b>Prerequisites</b>	<b>3</b>
<b>Configuration</b>	<b>4</b>
Change the save path of the files	4
Storage in MySQL of the ID of the last issue	5
<b>Execution</b>	<b>5</b>
<b>Manual Execution</b>	<b>9</b>

## INTRODUCTION

Plugin for JIRA issue management. Revised January 2019.

JIRA is a tool for the administration of tasks in projects, tracking and resolution of issues and operational management of projects.

## COMPATIBILITY MATRIX

Developed for Atlassian Jira Project Management Software v7.13.0.

## PREREQUISITES

A user with management privileges is required on the JIRA web platform.

Connection with the Tentacle service associated to your Pandora FMS server is required (for local executions of the plugin) under the following conditions:

- If you have `as_server_plugin` enabled.
- If you have `agent_per_database` enabled.

The deployment of this plugin by binaries does not need any special requirement.

It requires PandoraFMS library::PluginTools.pm

This library is available with the installation of Pandora FMS OpenSource package.

You can download the latest version from:

[https://github.com/pandorafms/pandorafms/blob/develop/pandora\\_server/lib/PandoraFMS/PluginTools.pm](https://github.com/pandorafms/pandorafms/blob/develop/pandora_server/lib/PandoraFMS/PluginTools.pm)

## CONFIGURATION

The JIRA plugin does not require any default configuration. However, you can optionally specify parameters such as the path in which the issue files will be generated or the access to Pandora's MySQL database.

### Change the save path of the files

```
#=====
# Create a file containing Jira's issue ID in any path given
#=====
#
if [ -d "/usr/share/pandora_server/jira_ticket/" ]; then

    if [ -f "/usr/share/pandora_server/jira_ticket/$module_id.txt" ]; then
        touch "/usr/share/pandora_server/jira_ticket/$module_id.txt"
        echo $ticket_id > "/usr/share/pandora_server/jira_ticket/$module_id.txt"

    else
        touch "/usr/share/pandora_server/jira_ticket/$module_id.txt"
        echo $ticket_id > "/usr/share/pandora_server/jira_ticket/$module_id.txt"
        fi

else

    mkdir /usr/share/pandora_server/jira_ticket/
    chmod 755 /usr/share/pandora_server/jira_ticket/
    touch "/usr/share/pandora_server/jira_ticket/$module_id.txt"
    echo $ticket_id > "/usr/share/pandora_server/jira_ticket/$module_id.txt"

fi

#
```

It is possible to configure the desired path for the storage of the ID of the generated issues, although it is recommended to keep the default one.

## Storage in MySQL of the ID of the last issue

```
#####  
# Database access data  
#####  
  
#  
  
conf_file="/etc/pandora/pandora_server.conf"  
  
db_host=$(grep "^dbhost" $conf_file | awk '{print $2}' | tr -d '\n');  
db_user=$(grep "^dbuser" $conf_file | awk '{print $2}' | tr -d '\n');  
db_pass=$(grep "^dbpass" $conf_file | awk '{print $2}' | tr -d '\n');  
db_name=$(grep "^dbname" $conf_file | awk '{print $2}' | tr -d '\n');  
  
# Save last ticket id in MySQL  
  
#mysql -h $db_host -u $db_user -p$db_pass $db_name -e "update tagente_modulo set  
custom_id=$ticket_id where id_agente_modulo=$module_id ;" 2> /dev/null
```

You can save the ID of the last issue generated in the database by pointing the MySQL query to the desired table.

It is recommended not to use this option and to leave the management of the generated IDs to the file created by the plugin itself.

## EXECUTION

The JIRA issue management plugin has the following parameters associated with it:

**Host:**

-c [IP Address and connection point to JIRA]

**User:**

-u [JIRA Connection User]

**Password:**

-k [User Password]

**Project:**

-t [Acronyms of the project to which we want to associate the alerts]

**Summary:**

-d [Issue Indicator Title for JIRA]

**Description:**

-p [Brief description of the issue]

**Assignee:**

-a [JIRA user to whom the issue will be assigned]

**Type:**

-g [Type of issue in JIRA]

**Priority:**

-i [Priority level of the Issue]

**Security:**

-s [Security level of the issue]

**Optional:**

-o [Optional Custom Parameter]

**Module:**

-m [ID del módulo del agente en el que se ha producido la incidencia]

**Mode:**

-M [This parameter will be marked as "validate" only when the module has been recovered.]

JIRA has a multitude of parameters and configurations that adapt to the needs of each environment, so that if necessary the fields that were needed could be added or deleted in the code itself.

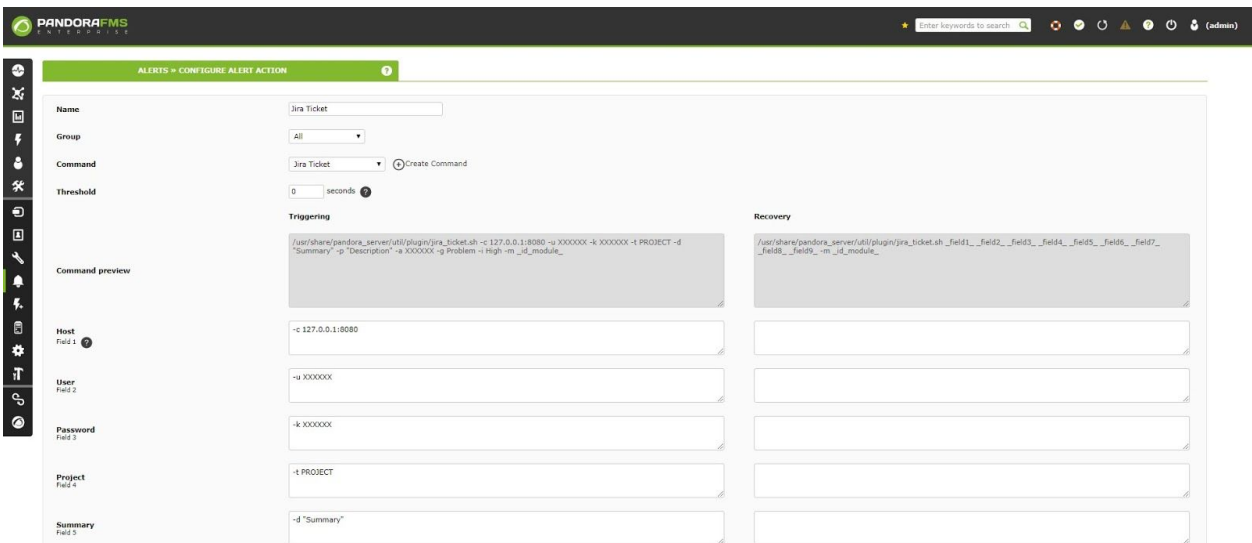
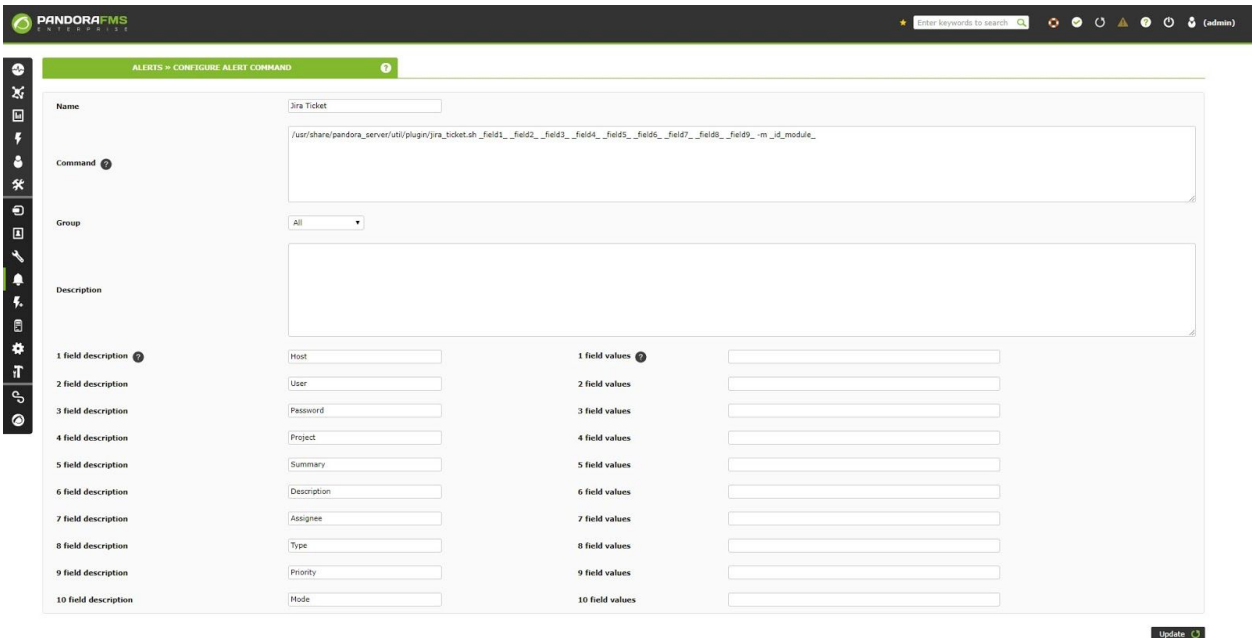
For the correct execution of the plugin it is recommended to associate it to the Pandora event alerts.

It will be necessary to create two independent processes for the plugin, one to generate the issues and another to manage their recovery.

The process that manages the creation of issues in JIRA will need all the parameters except "**Mode**" for the correct configuration of the tickets that will be generated in the platform.

The "**Optional**" parameter should be specified only when there are custom fields in the creation of tickets defining them in the code with "customfield\_[ID]".

The following images show images related to the creation of the necessary command and action for the event that will allow to create issue tickets in the JIRA web platform:



Description Field 4	-p "Description"	
Assignee Field 7	-a XXXXXXX	
Type Field 8	-g Problem	
Priority Field 9	-i High	
Mode Field 10		

[Update](#)

Pandora FMS v2.0RC-739 - Build 739 - MR 23  
Page generated at January 28, 2015, 5:07 pm  
By Jitka ST

The necessary parameters to close JIRA issues are **Host, User, Password, Module** and **Mode**. In the same way as with the previous step, you will need to create a specific action for the recovery of issues.

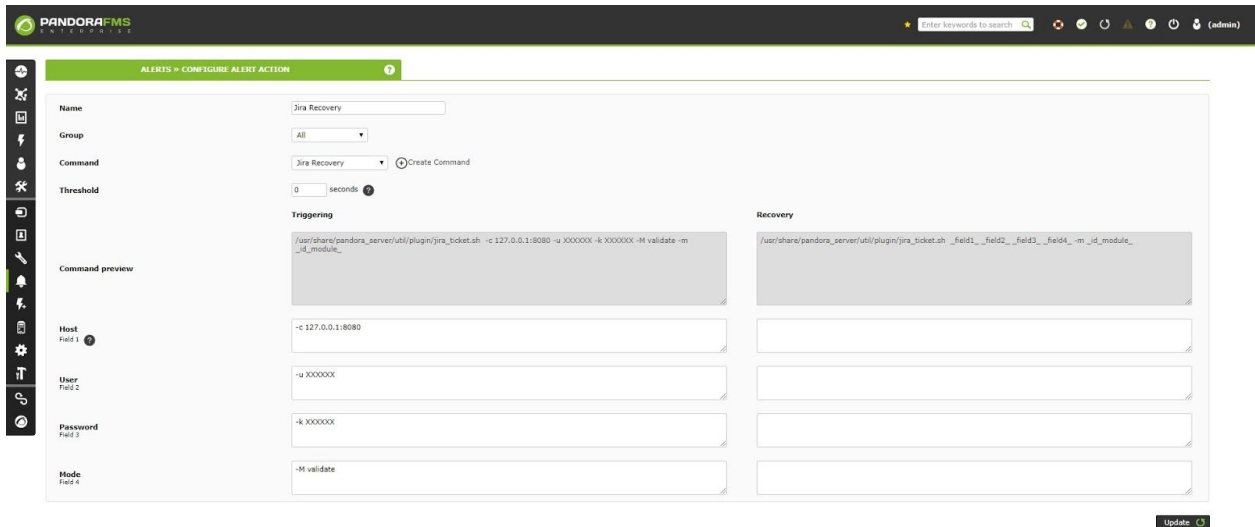
**PANDORAFMS** Enter keywords to search

ALERTS >> CONFIGURE ALERT COMMAND

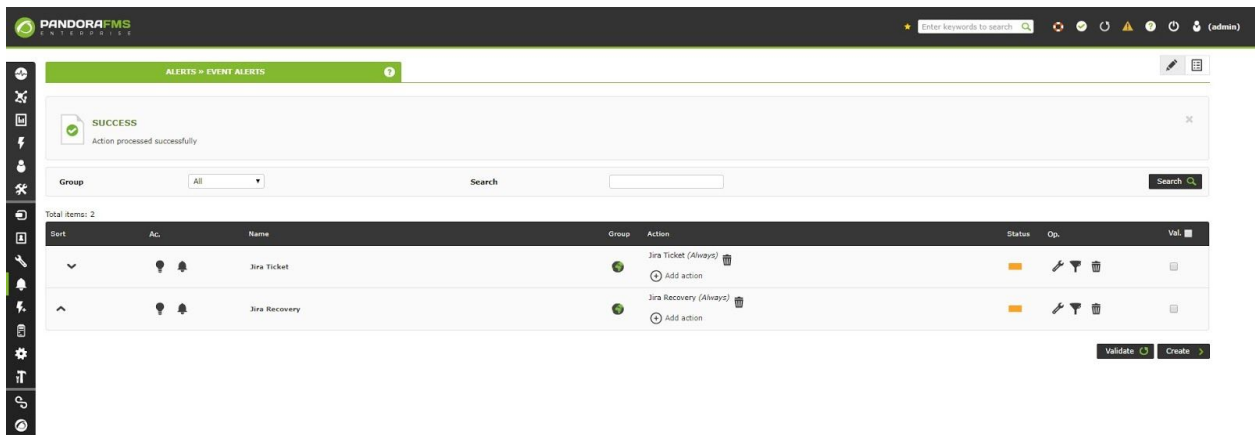
Name	Jira Recovery		
Command	/usr/share/pandora_server/uti/plugin/jira_ticket.sh _field1_ _field2_ _field3_ _field4_ -m _id_module_		
Group	All		
Description			
1 field description	Host	1 field values	
2 field description	User	2 field values	
3 field description	Password	3 field values	
4 field description	Mode	4 field values	
5 field description		5 field values	
6 field description		6 field values	
7 field description		7 field values	
8 field description		8 field values	
9 field description		9 field values	
10 field description		10 field values	

[Update](#)





Once the actions for the correct management of JIRA issues have been defined, two Pandora alert events must be created to supervise the operation of the modules in the agents we want to monitor with the plugin.



Once the alert events for the agents are created, they will start to supervise the operation of the associated modules, generating issue tickets in the JIRA web platform that will change their status to **"Completed"** as soon as the modules return to their correct operation.

## MANUAL EXECUTION

To execute the plugin manually, it will be necessary to execute the following command:

```
./jira_ticket.sh -c X.X.X.X:8080 -u XXXXXX -k XXXXXX -t PRO -d "Example" -p "Example" -a XXXXXX -g Issue -i High -m module
```

This command will generate a JIRA ticket with the specified values. If on the other hand you

want to change the status of the issue to "**Completed**" it will be necessary to execute the command as follows:

```
./jira_ticket.sh -c X.X.X.X:8080 -u XXXXXX -k XXXXXX -m module -M validate
```