



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

Pandora OTRS

© Ártica Soluciones Tecnológicas 2005-2018

## INDEX

<b>Introduction</b>	<b>3</b>
<b>Compatibility Matrix</b>	<b>3</b>
<b>Prerequisites</b>	<b>3</b>
<b>Creation of the GenericTicketConnector Web Service</b>	<b>4</b>
<b>Plugin Parameters</b>	<b>6</b>
[<operation>]	6
<connection_data>	7
<credentials>	8
<additional_parameters>	8
Full Syntax for Ticket Creation	10
Full Syntax for Ticket Update	11
Examples	11
<b>Configuration as Alert</b>	<b>12</b>
Alert Command	12
Alert Action	15
Module Alert	17
<b>Configuration as Event Response</b>	<b>18</b>

## INTRODUCTION

Plugin for Pandora FMS integration with OTRS ticketing tool.

It allows to create and update tickets in OTRS automatically from Pandora FMS through the OTRS "GenericTicketConnector" web service.

## COMPATIBILITY MATRIX

Developed for:

- OTRS 6.0.5

## PREREQUISITES

Required:

- Installation of the "GenericTicketConnector" web service in OTRS.
- Allow communication from the Pandora FMS server and console to the "GenericTicketConnector" web service.

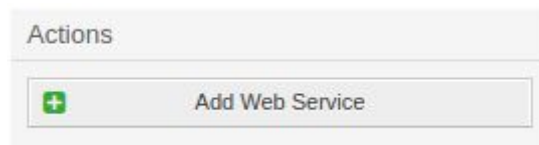
## CREATION OF THE GENERIC TICKET CONNECTOR WEB SERVICE

Along with the integration plugin, a file called "*GenericTicketConnector.yml*" is included, to easily create the required web service in OTRS.

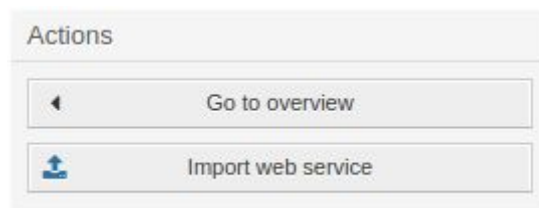
To that end, access the OTRS web portal and go to the section "*Admin > Processes & Automation > Web Services*".



When accessing, a button to add a new web service will appear at the left side of the screen.



After clicking on that button, a form will be displayed to add the web service, and to the left of the form another button to import it will be shown.



By clicking on the "Import web service" button, a window will be displayed where it is possible to look for the "GenericTicketConnector.yml" file in the system.

**Import web service** [X]

★ Configuration File:  GenericTic...ector.yml

The file must be a valid web service configuration  
YAML file.

Name:

Here you can specify a name for the webservice. If  
this field is empty, the name of the configuration file  
is used as name.

Once selected, clicking on the "Import" button will automatically fill out the form to create the web service. To finish creating the service, go to the bottom of the page and click "Save".

or

## PLUGIN PARAMETERS

The plugin has different parameters for creating and updating tickets in OTRS to specify the different fields that it has.

In a simplified way, the syntax for executing the plugin from the command line would be as follows:

```
pandora_otrs_ticket \  
  [<operation>] \  
  <connection_data> \  
  <credentials> \  
  [<additional_parameters>]
```

### [<operation>]

These are optional parameters that specify the operation performed by the plugin. It might be:

```
--Operation 'TicketCreate'
```

It makes the plugin create a new ticket in OTRS and return its ticket number.

```
--Operation 'TicketUpdate' <ticket_identifier>
```

It makes the plugin update an existing ticket, identified by a "*<ticket\_identifier>*". This identifier can be specified using the ticket ID or the ticket number with these options:

```
--TicketID '<ID>'
```

```
--TicketNumber '<Number>'
```

If no operation is specified, '*TicketCreate*' is used by default.

Therefore, if the operation is explicitly indicated, these would be the possibilities:

```
--Operation 'TicketCreate'
```

```
--Operation 'TicketUpdate' --TicketID '<ID>'
```

```
--Operation 'TicketUpdate' --TicketNumber '<Number>'
```

## <connection\_data>

It would be made up by a set of parameters needed for establishing the connection to the "GenericTicketConnector" web service. There are two options, indicating the complete URL to connect to the web server or let the plugin build it by itself.

```
--Url '<url>'
```

It indicates the full connection URL.

```
--Server '<server>' [--Ssl]
```

It indicates the IP or connection DNS name to the OTRS server, and from there it builds the complete connection URL. If the parameter "--Ssl" is also indicated, the URL will be built with HTTPS instead of HTTP.

*http://<server>/otrs/nph-genericinterface.pl/Webservice/GenericTicketConnector*

or

*https://<server>/otrs/nph-genericinterface.pl/Webservice/GenericTicketConnector*

Therefore there would be 3 possible configurations to indicate the connection data:

```
--Url '<url>'
```

```
--Server '<server>'
```

```
--Server '<server>' --Ssl
```

## <credentials>

Here the necessary parameters for authentication in OTRS to create and update tickets should be indicated. These 2 parameters must be indicated:

```
--UserLogin '<agent_username>'
```

It indicates an existent username in OTRS. This user will be the one that will create and update the tickets. Users available on the OTRS web portal can be checked in the menu "Admin> Users, Groups & Roles> Agents".

```
--Password '<agent_password>'
```

It indicates the authentication password of the given user.

Therefore, in this configuration, this should be indicated:

```
--UserLogin '<agent_username>' --Password '<agent_password>'
```

## <additional\_parameters>

It would include the rest of the parameters accepted by the plugin and that are not mandatory in all calls, although they may be for some cases:

```
--NameSpace '<name_space>'
```

If indicated, this namespace must match that of the SOAP call of the web service. By default it is "<http://www.otrs.org/TicketConnector/>".

```
--CustomerUser '<customer_username>'
```

It indicates the username of the client that creates the ticket. Users available on the OTRS web portal can be checked in the menu "Admin> Users, Groups & Roles> Customer User". It is required if the operation "TicketCreate" is used.

```
--Owner '<agent_username>'
```

It indicates the username of the ticket owner. Users available on the OTRS web portal can be checked in the menu "Admin> Users, Groups & Roles> Agents". With the operation "TicketCreate", it uses by default the indicated user for "--UserLogin".

```
--Title '<ticket_title>'
```

It indicates the ticket title. With the operation "TicketCreate", it uses the title "New ticket" by default.

```
--Subject '<article_subject>'
```

It indicates the subject of the article included in the ticket. With the operation "TicketCreate", it uses by default the value of the "--Title" option. With the "TicketUpdate" operation, it uses the subject "Update" by default.



`--Body '<article_body>'`

It indicates the content of the article included in the ticket. With the operation *"TicketCreate"*, it uses the content *"Ticket created by Pandora FMS"* by default. With the operation *"TicketUpdate"*, it uses the default content *"Ticket updated by Pandora FMS"*.

`--ContentType '<content_type_charset>'`

It indicates the codification of the content of the article included in the ticket. By default, it uses *"text/plain; charset=utf8"*.

`--Queue '<queue_name>'`

It indicates the name of the ticket queue. Queues available in the OTRS web portal can be checked in the menu *"Admin > Ticket Settings > Queues"*. Necessary if the operation *"TicketCreate"* is used.

`--Type '<type_name>'`

It indicates the ticket type name. Types available in the OTRS web portal can be checked in the menu *"Admin > Ticket Settings > Types"*.

`--Priority '<priority_name>'`

It indicates ticket priority. Priorities available in the OTRS web portal can be checked in the menu *"Admin > Ticket Settings > Priorities"*. With the *"TicketCreate"* operation, priority *"3 normal"* is used by default.

`--State '<state_name>'`

It indicates the ticket state. The state available in the OTRS web portal can be checked in the menu *"Admin > Ticket Settings > States"*. With the operation *"TicketCreate"*, the status *"new"* is used by default.

`--PendingTime '<minutes>'`

It indicates the ticket waiting time in minutes. It is only useful with status of the *"pending reminder"* or *"pending auto"* type. The status types available in the OTRS web portal can be checked in the menu *"Admin > Ticket Settings > States"*.

`--Service '<service_name>'`

It indicates the ticket service. Services available in the OTRS web portal can be checked in the menu *"Admin > Ticket Settings > Services"*. Required if the option *"--SLA"* is used.

`--SLA '<sla_name>'`

It indicates the ticket SLA. SLA available in the OTRS web portal can be checked in the menu *"Admin > Ticket Settings > Service Level Agreements"*. It needs for the ticket to have a valid service for the indicated SLA.

```
--TimeUnit '<time_unit>'
```

It indicates the time recorded for the ticket. The value "0" is used by default.

```
--DynamicField <field_name>=<value>'
```

It allows to indicate values for dynamic fields. This parameter can be used more than once in the same call to indicate different dynamic fields. The dynamic fields available on the OTRS web portal can be checked in the menu *"Admin> Processes & Automation> Dynamic Fields"*

```
--StoreID '<pandora_module_id>'
```

With the operation *"TicketCreate"*, it indicates the ID of a Pandora FMS module in which to store the ID of the created ticket. The ticket ID is stored in the *"Custom ID"* field of the Pandora FMS module.

```
--PandoraConf '<pandora_server_conf_file_path>'
```

It points out the path of the Pandora FMS server configuration file *"pandora\_server.conf"*. This file is used to take the connection data to the Pandora FMS database. Required if the option *"--StoreID"* is used.

## Full syntax for ticket creation

```
pandora_otrs_ticket
  [--Operation 'TicketCreate']
  [--Url '<url>' | --Server '<server>' [--Ssl]
  [--NameSpace '<name_space>']
  --UserLogin '<agent_username>'
  --Password '<agent_password>'
  --CustomerUser '<customer_username>'
  --Queue '<queue_name>'
  [--Owner '<agent_username>']
  [--Title '<ticket_title>']
  [--Subject '<article_subject>']
  [--Body '<article_body>']
  [--ContentType '<content_type_charset>']
  [--Type '<type_name>']
  [--Priority '<priority_name>']
  [--State '<state_name>' [--PendingTime '<minutes>']]
  [--Service '<service_name>' [--SLA '<sla_name>']]
  [--TimeUnit '<time_unit>']
  [--StoreID '<pandora_module_id>' --PandoraConf '<pandora_server_conf_file_path>']
  [--DynamicField <field_name>=<value>']
```

## Full syntax for ticket update

```
pandora_otrs_ticket
  --Operation 'TicketUpdate'
  --TicketID '<ID>' | --TicketNumber '<Number>'
  --Url '<url>' | --Server '<server>' [--Ssl]
  [--NameSpace '<name_space>']
  --UserLogin '<agent_username>'
  --Password '<agent_password>'
  [--CustomerUser '<customer_username>']
  [--Queue '<queue_name>']
  [--Owner '<agent_username>']
  [--Title '<ticket_title>']
  [--Subject '<article_subject>']
  [--Body '<aticle_body>']
  [--ContentType '<content_type_charset>']
  [--Type '<type_name>']
  [--Priority '<priority_name>']
  [--State '<state_name>']
  [--PendingTime '<minutes>']
  [--Service '<service_name>']
  [--SLA '<sla_name>']
  [--TimeUnit '<time_unit>']
  [--DynamicField <field_name>=<value>']
```

## Examples

- Create new ticket:

```
pandora_otrs_ticket \
  --Server '192.168.1.20' --Ssl \
  --UserLogin 'pandora' --Password 'P4nd0r4' \
  --CustomerUser 'Artica' \
  --Queue 'Office' \
  --Owner 'Systems' \
  --Title 'Office AP down' \
  --Body '7th floor AP is unreachable' \
  --Type 'Incident' \
  --Priority '4 high'
```

- Update the state of an existing ticket:

```
pandora_otrs_ticket \
  --Operation 'TicketUpdate' --TicketNumber '2019080910000228' \
  --Server '192.168.1.20' --Ssl \
  --UserLogin 'pandora' --Password 'P4nd0r4' \
  --Subject 'AP is up again' \
  --Body '7th floor AP is now working' \
  --State 'closed successful'
```

- Create a new ticket and store its ID in Pandora FMS:

```
pandora_otrs_ticket \
  --Server '192.168.1.20' --Ssl \
  --UserLogin 'pandora' --Password 'P4nd0r4' \
  --CustomerUser 'Artica' \
  --Queue 'Office' \
  --Owner 'Systems' \
  --Title 'Office AP down' \
  --Body '7th floor AP is unreachable' \
  --Type 'Incident' \
  --Priority '4 high' \
  --StoreID '23' --PandoraConf '/etc/pandora/pandora_server.conf'
```

## CONFIGURATION AS ALERT

### Alert command

First of all, the plugin must be located in a directory on the Pandora FMS server. For example:

**`/usr/share/pandora_server/util/pandora_otrs_ticket`**

To register the plugin as an alert command in Pandora FMS, access the "Alerts> Commands" menu and click on "Create".

Although different plugin parameters can be registered according to what suits each environment the best, in the following example a general configuration is proposed. This one stores the IDs of the created tickets in the "Custom ID" field of the corresponding module:

**Name:** Pandora OTRS

The fixed parameters for all executions (such as the IP of the OTRS server or user) are set directly in the definition of the command, while the rest of the variable parameters are defined as "*fieldN*" fields to be able to change their value in each case from the Pandora FMS console.

**Command:** `/usr/share/pandora_server/util/pandora_otrs_ticket --Server '192.168.70.133' --UserLogin 'pandora' --Password 'pandora' --PandoraConf '/etc/pandora/pandora_server.conf' --StoreID '_id_module_' --TicketNumber '_modulecustomid_' --Operation '_field1_' --Queue '_field2_' --State '_field3_' --Type '_field4_' --Priority '_field5_' --CustomerUser '_field6_' --Owner '_field7_' --Title '_field8_' --Subject '_field9_' --Body '_field10_'`

**Group:** All

**Description:** Create or update tickets in OTRS

For the "*fieldN*" fields, a configuration that creates forms with drop-downs when configuring alert actions is proposed:

**1 field description:** Operation

**1 field value:** TicketCreate,Create;TicketUpdate,Update

The 2 possible plugin operations are configured as a drop-downs.

**2 field description:** Queue

**2 field value:** Systems,Systems;Network,Network;Databases,Databases

Some of the existing queues in OTRS are configured as drop-downs.

NAME
Databases
Junk
Misc
Network
Pandora
Postmaster
Raw
Systems

**3 field description:** State

**3 field value:** new,New;closed successful,Closed successful

Some of the existing states in OTRS are configured as drop-downs.

NAME	TYPE
closed successful	closed
closed unsuccessful	closed
merged	merged
new	new
open	open
pending auto close+	pending auto
pending auto close-	pending auto
pending reminder	pending reminder
removed	removed

**4 field description:** Type

**4 field value:** Incident,Incident;Resources,Resources;Unclassified,Unclassified

Some of the existing ticket types in OTRS are configured as drop-downs.

NAME
Incident
Resources
Unclassified

**5 field description:** Priority

**5 field value:** 5 very high,Very high;4 high,High;3 normal,Normal;2 low,Low;1 very low,Very low

Some of the existing priorities in OTRS are configured as drop-downs.

NAME
1 very low
2 low
3 normal
4 high
5 very high

The remaining fields are left as user-filled text boxes when creating the alert action:

**6 field description:** Customer User

**7 field description:** Owner

**8 field description:** Title

**9 field description:** Subject

**10 field description:** Body

<b>Name</b>	Pandora OTRS			
<b>Command</b>	/usr/share/pandora_server/util/pandora_otrs_ticket--Server '192.168.70.133' --UserLogin 'pandora' --Password 'pandora' --PandoraConf /etc/pandora/pandora_server.conf --StoreID '_id_module_' --TicketNumber '_modulecustomid_' --Operation '_field1_' --Queue '_field2_' --State '_field3_' --Type '_field4_' --Priority '_field5_' --CustomerUser '_field6_' --Owner '_field7_' --Title '_field8_' --Subject '_field9_' --Body '_field10_'			
<b>Group</b>	All			
<b>Description</b>	Create or update tickets in OTRS			
<b>1 field description</b>	Operation	<b>1 field values</b>	TicketCreate,Create;TicketUpdate,Update	Hide <input type="checkbox"/>
<b>2 field description</b>	Queue	<b>2 field values</b>	Systems,Systems;Network,Network;Databases,Databases	Hide <input type="checkbox"/>
<b>3 field description</b>	State	<b>3 field values</b>	new;New;closed successful;Closed successful	Hide <input type="checkbox"/>
<b>4 field description</b>	Type	<b>4 field values</b>	Incident,Incident;Resources,Resources;Unclassified,Unclassified	Hide <input type="checkbox"/>
<b>5 field description</b>	Priority	<b>5 field values</b>	5 very high;Very high;4 high;High;3 normal;Normal;2 low;Low;1 very k	Hide <input type="checkbox"/>
<b>6 field description</b>	Customer User	<b>6 field values</b>		Hide <input type="checkbox"/>
<b>7 field description</b>	Owner	<b>7 field values</b>		Hide <input type="checkbox"/>
<b>8 field description</b>	Title	<b>8 field values</b>		Hide <input type="checkbox"/>
<b>9 field description</b>	Subject	<b>9 field values</b>		Hide <input type="checkbox"/>
<b>10 field description</b>	Body	<b>10 field values</b>		Hide <input type="checkbox"/>

## Alert action

With the command registered in this way, alert actions in the Pandora FMS console for different types of tickets can be created, allowing them to be created and closed automatically when there is a failure and when recovering.

To do it, access the menu *"Alerts> Actions"* of Pandora FMS console and click on *"Create"*.

For example, to create a ticket action for the group *"Network"*:

**Name:** Pandora OTRS (Network)

**Group:** Network

**Command:** Pandora OTRS

**Threshold:** 0

The rest of the form will be adjusted according to what is set in the alert command, and the fields for triggering and recovering the alert can be filled out.

- **For triggering:**

**Operation:** Create

**Queue:** Network

**State:** New

**Type:** Incident

**Priority:** Very high

**Customer User:** artica

**Owner:** pandora

**Title:** Pandora FMS alert fired for agent `_agent_` on module `_module_`

**Subject:** Module `_module_` is in status `_modulestatus_`

**Body:**

Group: `_agentgroup_`

Agent: `_agent_`

Address: `_address_`

Module: `_module_`

Data: `_data_`

- **For recovery:** It is important to leave the fields that you do not wish to modify with

the same value as in the triggering, since due to the way the plugin works when passing those parameters in the call they will try to get updated.

**Operation:** Update

**Queue:** Network

**State:** Closed successful

**Type:** Incident

**Priority:** Very high

**Customer User:** artica

**Owner:** pandora

**Title:** Empty. This will make it use the same text as the one used by the triggering preceded by the text "[RECOVER]".

**Subject:** Module recovered

**Body:** Alert has been automatically recovered.

<b>Name</b>	Pandora OTRS (Network)	
<b>Group</b>	Network	
<b>Command</b>	Pandora OTRS <input type="button" value="Create Command"/>	
<b>Threshold</b>	0	
<b>Command preview</b>	<pre> /usr/share/pandora_server/util/pandora_otrs_ticket --Server '192.168.70.133' --UserLogin 'pandora' --Password 'pandora' --PandoraConf '/etc/pandora/pandora_server.conf' --StoreID '._id_module.' --TicketNumber '._modulecustomid.' --Operation 'TicketCreate' --Queue 'Network' --State 'new' --Type 'Incident' --Priority '5 very high' --CustomerUser 'artica' --Owner 'pandora' --Title 'Pandora FMS alert fired for agent_{{agent}}_on module_{{module}}.' --Subject 'Module_{{module}}_is in status_{{modulestatus}}.' --Body 'Group: {{agentgroup}} Agent: {{agent}}'                     </pre>	
<b>Operation</b>	Create	Update
<b>Queue</b>	Network	Network
<b>State</b>	New	Closed successful
<b>Type</b>	Incident	Incident
<b>Priority</b>	Very high	Very high
<b>Customer User</b>	artica	artica
<b>Owner</b>	pandora	pandora
<b>Title</b>	Pandora FMS alert fired for agent_{{agent}}_on module_{{module}}.	
<b>Subject</b>	Module_{{module}}_is in status_{{modulestatus}}.	Module recovered
<b>Body</b>	Group: {{agentgroup}} Agent: {{agent}}	Alert has been automatically recovered.



## Module alert

Once the action configuration has been set, an alert can be assigned to a module to create and close tickets automatically in OTRS.

Choose an alert template with the desired terms for alert triggering, assign it to a module and add the action you want to be executed.

For example, on the agent "SW00721" the "Critical condition" template has been added to the "Host Alive" module with the action "Pandora OTRS (Network)".

That way, when the module goes into critical status, a ticket is created in OTRS automatically:

The ticket ID is stored in the "Custom ID" field of the module.

**Custom ID**

201908301000027

When the module returns to normal status and the alert is recovered, the recovery action will be executed, and the status and title of the ticket will be updated in addition to adding a comment about it:

## CONFIGURATION AS EVENT RESPONSE

The plugin can also be registered as an event response, giving operators the option to create tickets in OTRS directly from the Pandora FMS console.

It is necessary for the plugin to be located in a directory on the computer that hosts the Pandora FMS console and is executable by it. In this case, the same route as for alerts is used, since both (Pandora FMs server and console) are on the same computer:

**`/usr/share/pandora_server/util/pandora_otrs_ticket`**

To create the event response, access the menu "*Events> Event responses*" and click on "*Create response*".

**Name:** Pandora OTRS create

**Group:** All

**Description:** Create tickets in OTRS

**Location:** Modal window

**Size:** 620 x 500

**Type:** Command

**Server to execute command:** Local console

In this case, since the answer will only be useful to create tickets, type in directly the value of the "*- Operation*" parameter in the command, instead of leaving it as a variable (as it was the case of the alert).

In addition, do not store the ticket ID in Pandora FMS, since not all events have an associated module.

The rest of the parameters can also be set in a fixed way or you can give the operator the option to fill them out manually. In this example, they are set to be filled out by the operator in each case.

**Command:** `/usr/share/pandora_server/util/pandora_otrs_ticket --Server '192.168.70.133' --UserLogin 'pandora' --Password 'pandora' --Operation 'TicketCreate' --Queue '_Queue_' --State '_State_' --Type '_Type_' --Priority '_Priority_' --CustomerUser '_CustomerUser_' --Owner '_Owner_' --Title '_Title_' --Subject '_Subject_' --Body '_Body_'`

**Parameters:** Queue,State,Type,Priority,CustomerUser,Owner,Title,Subject,Body

<b>Name</b>	<input type="text" value="Pandora OTRS create"/>	<b>Group</b>	<input type="text" value="All"/>
<b>Description</b>	<input type="text" value="Create tickets in OTRS"/>		
<b>Location</b>	<input type="text" value="Modal window"/>	<b>Size</b>	Width (px) <input type="text" value="620"/> Height (px) <input type="text" value="500"/>
<b>Parameters</b>	<input type="text" value="Queue.State.Type.Priority.CustomerUser.Owner.Title.Subject.B"/>		
<b>Command</b>	<input type="text" value="usr/share/pandora_server/util/pandora_otrs_ticket --Server '192.168.70.133' --UserLogin 'pandora' --Password 'pandora' --Operation 'TicketCreate' --Queue 'Queue' --State 'State' --Type 'Type' --Priority 'Priority' --CustomerUser 'CustomerUser' --Owner 'Owner' --Title 'Title' --Subject 'Subject' --Body 'Body'"/>		<b>Server to execute command</b>
			<input type="text" value="Local console"/>

With the response created, when an operator accesses the details of an event, it can choose it as "Custom response" and the form will appear with the fields to be filled out based on what is configured in the response.

Module 'Host Alive' is going to CRITICAL (0) ✕

General	Details	Agent fields	Comments	Responses
<b>Change owner</b>	<input type="text" value="None"/>			<input type="button" value="Update &gt;"/>
<b>Change status</b>	<input type="text" value="Validated"/>			<input type="button" value="Update &gt;"/>
<b>Comment</b>				<input type="button" value="Add comment &gt;"/>
<b>Delete event</b>				<input type="button" value="Delete event ✕"/>
<b>Custom responses</b>	<input type="text" value="Pandora OTRS create"/>			<input type="button" value="Execute &gt;"/>
<b>Description</b>	<input type="text" value="Create tickets in OTRS"/>			
<b>Parameters</b>				
<i>Queue</i>	<input type="text" value="Network"/>			
<i>State</i>	<input type="text" value="new"/>			
<i>Type</i>	<input type="text" value="Incident"/>			
<i>Priority</i>	<input type="text" value="5 very high"/>			
<i>CustomerUser</i>	<input type="text" value="artica"/>			
<i>Owner</i>	<input type="text" value="pandora"/>			
<i>Title</i>	<input type="text" value="SW00721 is down"/>			
<i>Subject</i>	<input type="text" value="Host Alive is going critical"/>			
<i>Body</i>	<input type="text" value="Module Host Alive is going"/>			

By clicking on the “Execute” button, a modal window will appear with the command executed and the result thereof. If everything works properly, it will show the ID of the created ticket:

Pandora OTRS create
✕

> Executing command: /usr/share/pandora\_server/util/pandora\_otrs\_ticket --Server '192.168.70.133' --UserLogin 'pandora' --Password 'pandora' --Operation 'TicketCreate' --Queue 'Network' --State 'new' --Type 'Incident' --Priority '5 very high' --CustomerUser 'artica' --Owner 'pandora' --Title 'SW00721 is down' --Subject 'Host Alive is going critical' --Body 'Module Host Alive is going critical for agent SW00271 in Pandora FMS'

2019083010000045

Execute again >

And in OTRS the following ticket will be created:

■ Ticket#2019083010000045 — SW00721 is down

Back | Print | Priority | People | Communication | Pending | Close | Miscellaneous |

▼ Article Overview - 1 Article(s)

NO.	☆	⇄	SENDER	VIA	SUBJECT	CREATED
1		→	Artica ST	OTRS	Host Alive is going critical	30/08/2019 13:33

▼ #1 – Host Alive is going critical – Artica ST – 30/08/2019 13:33 via OTRS by pandora pandora

Reply to note | Print | Split

Module Host Alive is going critical for agent SW00271 in Pandora FMS