

PANDORAFMS



Pandora FMS Command Line Interface (CLI)

17-09-2021





Pandora FMS Command Line Interface (CLI)

[Go back to Pandora FMS documentation index](#)



We are working on the translation of the Pandora FMS documentation. Sorry for any inconvenience.

Pandora FMS CLI

The Pandora FMS CLI (Command-Line Interface) is used for making calls in command line on the file `/usr/share/pandora_server/util/pandora_manage.pl`. This method is specially useful to integrate applications of thirds parts with Pandora FMS through automated tasks. Basically, it consists on one call with the parameters formatted to do and action such as the creation an elimination of one agent, one module or one user, among other things.

The CLI is a file in Perl, so one call to CLI is as easy as this:

```
perl pandora_manage.pl <pandora_server.conf path> <option> <option parameters>
```



If you are an **enterprise** customer, you will need to use the binary executable instead of the `.pl`, since there are some enterprise features that may not work properly with the `.pl`. Just run `pandora_manage` to get started.

Only Enterprise:

```
pandora_manage <pandora_server.conf path> <option> <option parameters>
```

Pandora FMS CLI has the following options:



- **Agents**

- **-create_agent:** Create an agent
- **-update_agent:** Update an agent field
- **-delete_agent:** Delete an agent
- **-disable_group:** Disable all agents from one group
- **-enable_group:** Enable all agents from one group
- **-create_group:** Create a group
- **-delete_group:** Delete a group
- **-update_group:** Update a group
- **-stop_downtime:** Stop a planned downtime
- **-get_agent_status:** Get the status of a given agent
- **-get_agent_group:** Get the group name of a given agent
- **-get_agent_modules:** Get the module list of a given agent
- **-get_agents:** Get list of agents with optative filter parameters
- **-delete_conf_file:** Delete a local conf of a given agent
- **-clean_conf_file:** Clean a local conf of a given agent deleting all modules, policies and collections data
- **-get_bad_conf_files:** Get the files bad configured (without essential tokens)
- **-migration_agent_queue:** migrate an agent from one node to another with the metaconsole (for metaconsole only)
- **-migration_agent:** see if there is an agent in the migration queue (only for metaconsole)
- **-set_disabled_and_standby:** Disable an agent and put it in standby mode (Enterprise feature).
- **-reset_agent_counts:** Actualiza conteos de módulo y conteos de alertas.

- **Modules**

- **-create_data_module:** Add one data module to one agent
- **-create_web_module:** Add one web module to one agent
- **-create_network_module:** Add one network module to one agent
- **-create_network_component:** Create a new network component
- **-create_snmp_module:** Add one SNMP module to one agent
- **-create_plugin_module:** Add one module kind plugin to one agent
- **-get_module_group:** Displays the filtered groups of available modules
- **-create_module_group:** Create a group (module type)
- **-module_group_synch:** Synchronize group of modules from the meta console
- **-delete_module:** Delete one module from one agent
- **-data_module:** Insert data to one module
- **-get_module_data:** Show data from one module in the last X seconds (interval) in CSV format
- **-get_module_id:** Show id from one module.
- **-get_module_custom_id:** Shows the Custom ID of a module.
- **-set_module_custom_id:** Set the Custom ID of a module. If it is left blank it will remove the content that would have existed previously leaving the field empty.



- **-delete_data** Delete the historic data from a module, from the modules of one agent or from the modules of the agents of one group
- **-update_module:** Update one module field
- **Alerts**
 - **-create_template_module:** Add an alert template to an agent.
 - **-delete_template_module:** Delete an alert template from an agent.
 - **-create_template_action:** Create an action to one agent
 - **-delete_template_action:** Delete an action from an agent
 - **-disable_alerts:** Disable alerts in all groups.
 - **-enable_alerts:** Enable alerts in all groups.
 - **-create_alert_template:** Create an alert template
 - **-delete_alert_template:** Delete an alert template
 - **-update_alert_template:** Update field of an alert template
 - **-get_alert_actions:** Allows to visualize the actions of alerts
 - **-get_alert_actions_meta:** Allows to visualize the actions of node alerts from the meta console (only for meta console).
 - **-create_alert_command:** Creates an alert command
 - **-get_alert_commands:** Allows to visualize alert commands
 - **-validate_all_alerts:** Validates all the alerts
- **Users**
 - **-create_user:** Create one user.
 - **-delete_user:** Delete one user.
 - **-update_user:** Update field of a user
 - **-enable_user:** Enable a given user
 - **-disable_user:** Disable a given user
 - **-meta_synch_user:** Synchronize users with nodes (Metaconsole only)
 - **-create_profile:** Create a new profile.
 - **-update_profile:** Update a profile.
 - **-add_profile:** Add a profile to an user.
 - **-delete_profile:** Delete a profile from an user.
 - **-add_profile_to_user:** Add a profile to a user in a group
 - **-disable_eacl:** Disable the ACL Enterprise system.
 - **-enable_eacl:** Enable the ACL Enterprise system.
- **Events**
 - **-create_event:** Create an event.
 - **-validate_event:** Validate an event.
 - **-validate_event_id:** Validate an event given a event id.
 - **-get_event_info:** Display info about a event given a event id.
- **Incidents**
 - **-create_incident** Create a incident
- **Policies**
 - **-create_policy:** Create a policy
 - **-apply_policy** Force the application of a policy.
 - **-apply_all_policies:** Add to the application queue all the policies
 - **-add_agent_to_policy:** Add an agent to a policy



- **-add_collection_to_policy:** Add a collection to a policy
 - **-create_policy_data_module_from_local_component:** Create a data module from a local component to a policy
 - **-delete_not_policy_modules** Delete all the modules not associated to policies from the conf file
 - **-remove_agent_from_policy:** Eliminar un agente de una política
 - **-disable_policy_alerts:** Disable all the alerts from a policy
 - **-create_policy_data_module:** Create a policy data module
 - **-create_policy_web_module:** Create a policy web module
 - **-create_policy_network_module:** Create a policy network module
 - **-create_policy_snmp_module:** Create a policy SNMP module
 - **-create_policy_plugin_module:** Create a policy plugin module
 - **-validate_policy_alerts:** Validate all the alerts of a given policy
 - **-get_policy_modules:** Get the module list of a policy
 - **-get_policies:** Get all the policies (without parameters) or the policies of a given agent (agent name as parameter)
- **Tools**
 - **-exec_from_file:** Execute any CLI option using macros from a CSV file
 - **Consolas visuales**
 - **-create_visual_console:** Create a visual console in pandora following the indicated scheme
 - **-edit_visual_console:** Edit a visual console in pandora following the indicated scheme
 - **-delete_visual_console:** Eliminate a visual console and all its associated elements
 - **-delete_visual_console_objects:** Remove elements associated with a visual console following a filter
 - **-duplicate_visual_console:** Duplicate a visual console as many times as indicated
 - **-export_json_visual_console:** Exports a visual console and its elements to JSON format

Agents

Create_agent

Parameters: <agent_name> <operative_system> <group_name> <server_name> [<address> <description> <interval> <alias as name>]

Description: An agent with the name, the operative system, the group and the server specified will be created. Optionally, it will be possible to give it an address (IP or name), a description and an interval in seconds (by default 300).

The parameter <alias as name> is set to 1 by default and, with that value, the alias will be



the same as the agent name. If you want the name to be generated randomly this parameter must be set to 0.

Example:

```
perl pandora_manage.pl /etc/pandora/pandora_server.conf --
create_agent 'My agent' Windows Databases Central-Server
192.168.12.123 'Agent description' 600 0
```

Update_agent

(>=5.0)

Parameters: <agent_name> <field> <new_value> <use_agent_alias>

Description: A given field of an existent agent will be updated. The possible fields are the following: agent_name, address, description, group_name, interval, os_name, disabled, parent_name, cascade_protection, icon_path, update_gis_data, custom_id.

Example: perl pandora_manage.pl /etc/pandora/pandora_server.conf -update_agent 'Agent name' group_name 'Network'

Delete_agent

Parameters: <agent_name> <use_agent_alias>

Description: The agent processed will be deleted with name as parameter

Example:

```
perl pandora_manage.pl /etc/pandora/pandora_server.conf --
delete_agent 'Mi agente'
```

Disable_group

Parameters: <group_name>

Description: the agents of the group considered as parameter will be disabled with the execution of this option. If we pass 'All' as group, all agents from all groups will be disabled.

Example:



```
perl pandora_manage.pl /etc/pandora/pandora_server.conf --  
disable_group Firewalls
```

Enable_group

Parameters: <group_name>

Description: The agents of the group considered as parameter will be disabled with the execution of this option. If we pass 'All' as group all agents from all groups will be enabled.

Example:

```
perl pandora_manage.pl /etc/pandora/pandora_server.conf --  
enable_group All
```

Create_group

Parameters: <group_name> [<parent_group_name> <icon> <description>]

Description: A new group will be created if it doesn't exist and optionally, can be assigned a parent group, a icon (the icon name without extension) and description. The parent group by default is 'All' and the default icon is empty string (without icon)

Example:

```
perl pandora_manage.pl /etc/pandora/pandora_server.conf --  
create_group 'New group' 'Parent group' 'computer'
```

Delete_group

Parameters: <group_name>

Description: Delete a group introducing the group name

Example:

```
perl pandora_manage.pl /etc/pandora/pandora_server.conf -delete_group 'New group'
```

Update_group

Parameters: <group_id> [<group_name> <parent_group_name> <icon>]



<description>]

Description: A group is modified if there id (You can use the “Get_agent group_id” function to find the id of the group). The parameters that can be modified with the name, the father pointed to the icon and description.

Example: perl pandora_manage.pl /etc/pandora/pandora_server.conf -update_group 18 'Test' 'Web' 'SpiderMan' 'Description'

Stop_downtime

(>=5.0)

Parameters: <downtime_name>

Description: Stop a planned downtime. If the downtime is finished, a message will be showed

Example:

```
perl pandora_manage.pl /etc/pandora/pandora_server.conf --
show_downtime 'Downtime name'
```

Get_agent_status

(>=5.0)

Parameters: <agent_name> <use_agent_alias>

Description: Get the status of a given agent

Example:

```
perl pandora_manage.pl /etc/pandora/pandora_server.conf --
get_agent_status 'Agent name'
```

Get_agent_group

(>=5.0)

Parameters: <agent_name> <use_agent_alias>

Description: Get the group name of a given agent



Example:

```
perl pandora_manage.pl /etc/pandora/pandora_server.conf --  
get_agent_group 'Agent name'
```

Get_agent_group_id

(>=5.1)

Parameters: <agent_name> <use_agent_alias>

Description: Get the group id of a given agent

Example:

```
perl pandora_manage.pl /etc/pandora/pandora_server.conf --  
get_agent_group_id 'Agent name'
```

Get_agent_modules

(>=5.0)

Parameters: <agent_name> <use_agent_alias>

Description: Get the module list (id and name) of a given agent

Example:

```
perl pandora_manage.pl /etc/pandora/pandora_server.conf --  
get_agent_modules 'Agent name'
```

Get_agents

(>=5.0)

Parameters: [<group_name> <os_name> <status> <max_modules> <filter_substring>
<policy_name> <use_agent_alias>]

Description: Get list of agents with optative filter parameters

Possible values for the parameter <status> critical, warning, unknown, normal

Example:



```
perl pandora_manage.pl /etc/pandora/pandora_server.conf --  
get_agents 'Network' 'Linux' 'critical' 'Policy name'
```

Delete_conf_file

(>=5.0)

Parameters: <agent_name> <use_agent_alias>

Description: The conf file of one agent will be deleted

Example:

```
perl pandora_manage.pl /etc/pandora/pandora_server.conf --  
delete_conf_file 'Agent name'
```

Clean_conf_file

(>=5.0)

Parameters: [<agent_name> <use_agent_alias>]

Description: The conf file of one or all agents (without parameters) will be cleaned (All modules, policies, file collections and comments will be deleted).

Example:

```
perl pandora_manage.pl /etc/pandora/pandora_server.conf --  
clean_conf_file 'Agent name'
```

Get_bad_conf_files

(>=5.0)

Parameters: No

Description: A list of the bad configured conf files will be showed (Files without main tokens: server_ip,server_path,temporal,logfile)

Example:

```
perl pandora_manage.pl /etc/pandora/pandora_server.conf --
```



get_bad_conf_files

migration_agent_queue

(>=7.21) only for metaconsole

Parameters: [id of the agent to migrate][name of the source node][name of the destination node][optional parameter by default this 0 will migrate the data of both the history and the db itself if it is at 1 it will only migrate those of the db itself]

Description: Add an agent to the migration queue to move from node to another node this is for the metaconsole only. To do it correctly, you will need to fill in the API configuration parameters in pandora_server.conf.

Example:

```
perl pandora_manage.pl /etc/pandora/pandora_server.conf --
migration_agent_queue 1 nodo1 nodo2 0
```

migration_agent

(>=7.21) only for metaconsole **Parametrizes:** [id of the agent to check]

Description: it will return true or false depending on whether the agent entered exists in the agent migration table

Example:

```
perl pandora_manage.pl /etc/pandora/pandora_server.conf --
migration_agent 1
```

create_downtime

(>=5.1)

Parametrizes: <downtime_name> <description> <date_from> <date_to> <id_group>
<monday> <tuesday><wednesday> <thursday> <friday> <saturday> <sunday>
<periodically_time_from><periodically_time_to> <periodically_day_from>
<periodically_day_to><type_downtime> <type_execution> <type_periodicity>

Description: Created Planned downtime. You must have configured data API in pandora_server.conf.



Example:

```
perl pandora_manage.pl /etc/pandora/pandora_server.conf --
create_downtime Testing Testing 05/07/2015 06/07/2015 0 1 1 1 1 1 1
1 17:06:00 19:06:00 1 31 quiet periodically weekly
```

add_item_planned_downtime

(>=5.1)

Parametrizes: <id_downtime> <id_agente1,id_agente2,id_agente3...id_agenteN>
<name_module1,name_module2,name_module3...name_moduleN>

Description: Add items of planed downtime. You must have configured data API in pandora_server.conf.

Example:

```
perl pandora_manage.pl /etc/pandora/pandora_server.conf --
add_item_downtime 1 1 'Status','AvailableMemory'
```

get_all_planned_downtimes

(>=5.1)

Parametrizes: <name> [<id_group> <type_downtime> <type_execution>
<type_periodicity>]

Description: List all matches of planned downtime.

Example:

```
perl pandora_manage.pl /etc/pandora/pandora_server.conf --
get_all_planned_downtimes testing 0
```

planned_downtimes_items

(>=5.1)

Parametrizes: <name> [<id_group> <type_downtime> <type_execution>
<type_periodicity>]

Description: List all matches of planned downtime items.



Example:

```
perl pandora_manage.pl /etc/pandora/pandora_server.conf --
get_planned_downtimes_items testing 0
```

set_planned_downtimes_deleted

(>=5.1)

Parametrizes: <name>

Description: Delete a planned downtime.

Example:

```
perl pandora_manage.pl /etc/pandora/pandora_server.conf --
set_planned_downtimes_deleted test
```

set_disabled_and_standby

This feature is only available for Enterprise.

Parameters: <id_agent> (<id_node> <value>)

Description: It disables an agent and also, if it has remote configuration, it puts it in standby mode. In order for it to work properly in the server it has to have configured the connection with the API.

The second parameter, the console id, is only used in the meta. Any value can be set for the node. The third is optional and the default is 1.

Example to enable an agent from the node:

```
perl pandora_manage.pl /etc/pandora/pandora_server.conf --
set_disabled_and_standby 2 0 0
```

Example to disable an agent from the node:

```
perl pandora_manage.pl /etc/pandora/pandora_server.conf --
set_disabled_and_standby 2 0 1
```

Example to enable an agent from the metaconsole:



```
perl pandora_manage.pl /etc/pandora/pandora_server.conf --  
set_disabled_and_standby 2 1 0
```

Example to disable an agent from the metaconsole:

```
perl pandora_manage.pl /etc/pandora/pandora_server.conf --  
set_disabled_and_standby 2 1 1
```

reset_agent_counts

Parameters: <id_agent>

Description: Synchronizes module counts and alerts for a given agent. If you want to synchronize with all agents, the first parameter must be "All".

Example:

```
perl pandora_manage.pl /etc/pandora/pandora_server.conf --  
reset_agent_counts "All"
```

Modules

Create_data_module

Parameters: <module_name> <module_kind> <agent_name> [<description>
<module_group> <min> <max> <post_process> <interval> <warning_min>
<warning_max> <critical_min> <critical_max> <history_data> <def_file> <warning_str>
<critical_str> <enable_unknown_events> <ff_threshold> <each_ff>
<ff_threshold_normal> <ff_threshold_warning> <ff_threshold_critical> <ff_timeout>
<critical_instructions> <warning_instructions> <unknown_instructions>]
<use_agent_alias>

Description: A module kind data will be created in an agent with the module name, kind of module and name of the agent where it will be created. Optionally it will be possible to give a description, the module group, min and max values, a post_process value, an interval in seconds, min and max warning values, min and max critical values, a history data value and one module definition file.

The module definition file will contain some like this:

```
module_begin  
module_name My module
```



```
module_type generic_data
module_exec cat /proc/meminfo | grep MemFree | awk '{ print $2 }'
module_end
```

The default values are 0 for the minimum and maximum, history_data and post_process and 300 for the interval.

Notes:

The next parameters are only for the Pandora version 5 and next versions:

- <enable_unknown_events>

The next parameters are only for the Pandora version 5.1 and next versions:

- <ff_threshold>
- <each_ff>
- <ff_threshold_normal>
- <ff_threshold_warning>
- <ff_threshold_critical>
- <ff_timeout>

Example:

```
perl pandora_manage.pl /etc/pandora/pandora_server.conf --
create_data_module 'My module' generic_data 'My agent' 'module
description' 'General' 1 3 0 300 0 0 0 0 1
/home/user/filedefinition 'string for warning' 'string for
critical'
```

If you introduce a different name or kind between the parameters and the file definition, the fixed on the file will have priority.

Create_web_module

(>=7.0)

Parameters: <module_name> <module_kind> <agent_name> [<description>
<module_group> <min> <max>[<post_process> <interval> <warning_min>
<warning_max> <critical_min> <critical_max> <history_data> <retries> <requests>
<agent_browser_id> <auth_server> <auth_realm> <fichero_definicion> <proxy_url>
<proxy_auth_login> <proxy_auth_password> <warning_str> <critical_str>
<enable_unknown_events> <ff_threshold> <each_ff> <ff_threshold_normal>
<ff_threshold_warning> <ff_threshold_critical> <ff_timeout> <ff_timeout>
<warning_inverse> <critical_inverse> <critical_instructions> <warning_instructions>



<unknown_instructions>] <use_agent_alias>

Description: A web data module will be created in an agent with the module name, kind of module and name of the agent where it will be created. Optionally it will be possible to give a description, the module group, min and max values, a post_process value, an interval in seconds, min and max warning values, min and max critical values, a history data value and one module definition file among others.

The module definition file will contain some like this:

```
task_begin
get http://pandorafms.com
task_end
```

Example:

```
perl pandora_manage.pl /etc/pandora/pandora_server.conf --
create_web_module 'module_name' web_data 'pandorafms'
'module_description' 'General' 0 100 0 300 0 0 0 0 1 0 1 'Pandora
FMS' auto public /home/user/ficherodefinicion 'http://proxy.url'
'proxy_login' 'proxy_password' 1 10 10 10 10 10 10 10 1 1
'critical_instructions' 'warning_instructions'
'unknown_instructions'
```

Create_network_module

Parameters: <module_name> <module_kind> <agent_name> <module_address>
[<module_port> <description> <module_group> <min> <max> <post_process>
<interval> <warning_min> <warning_max> <critical_min> <critical_max>
<history_data> <ff_threshold> <warning_str> <critical_str> <enable_unknown_events>
<each_ff> <ff_threshold_normal> <ff_treshhold_warning> <ff_threshold_critical>
<critical_instructions> <warning_instructions> <unknown_instructions>]
<use_agent_alias>

Description:

A network module will be created in an agent with the module name, kind of module, name of the agent where it will be created and the module address specified. Optionally, it will be possible to give it a port, a description, values min and max, a post_process value, an interval in seconds, a warning min and max values, critical min and max values and a history data value.

The default values are 0 for the min and max, history_data and post_process an another 300 for the interval.



the port is optional, so the modules kind ICMP don't need it. In the rest of kinds, it is necessary to specify one module.

Notes:

The next parameters are only for the Pandora version 5 and next versions:

- <enable_unknown_events>

The next parameters are only for the Pandora version 5.1 and next versions:

- <each_ff>
- <ff_threshold_normal>
- <ff_threshold_warning>
- <ff_threshold_critical>

Example:

```
perl pandora_manage.pl /etc/pandora/pandora_server.conf --
create_network_module 'My module' remote_tcp 'My agent'
192.168.12.123 8080 'Module description' 'General' 1 3 0 300 0 0 0
0 1 'string for warning' 'string for critical'
```

Create_network_component

Parameters: <network_component_name> <network_component_group>
<network_component_type> [<description> <module_interval> <max_value>
<min_value> <snmp_community> <id_module_group> <max_timeout> <history_data>
<min_warning> <max_warning> <str_warning> <min_critical> <max_critical>
<str_critical> <min_ff_event> <post_process> <disabled_types_event> <each_ff>
<min_ff_event_normal> <min_ff_event_warning> <min_ff_event_critical>]

Description:

A network component will be created. Optionally, it will be possible to give it a port, a description, values min and max, a post_process value, an interval in seconds, a warning min and max values, critical min and max values and a history data value.

The default values are 0 for the min and max, history_data and post_process an another 300 for the interval.

the port is optional, so the modules kind ICMP don't need it. In the rest of kinds, it is necessary to specify one module.

Notes:



The next parameters are only for the Pandora version 5 and next versions:

- <enable_unknown_events>

The next parameters are only for the Pandora version 5.1 and next versions:

- <each_ff>
- <ff_threshold_normal>
- <ff_threshold_warning>
- <ff_threshold_critical>

Example:

```
perl /tmp/pandorafms/pandora_server/util/pandora_manage.pl
/etc/pandora/pandora_server.conf --create_network_component
"example_name" 2 7
```

Create_snmp_module

Parameters: <module_name> <module_kind> <agent_name> <module_address> <module_port> <version> [<community> <oid> <description> <module_group> <min> <max> <post_process> <interval> <warning_min> <warning_max> <critical_min> <critical_max> <history_data> <snmp3_priv_method> <snmp3_priv_pass> <snmp3_sec_level> <snmp3_auth_method> <snmp3_auth_user> <snmp3_auth_pass> <ff_threshold> <warning_str> <critical_str> <enable_unknown_events> <each_ff> <ff_threshold_normal> <ff_treshold_warning> <ff_threshold_critical> <critical_instructions> <warning_instructions> <unknown_instructions>] <use_agent_alias>

Description:A module kind snmp will be created in an agent with the module name, module kind, name of the agent where it will be created, the module address, the associated port and the SNMP version especificed. Optionally it will be given a community, am OID, a description, the module group, min and max values, a post_process value, an interval in seconds, min and max values, critical min and max values, an history data value, and the snmp3 values like methods, passwords, etc.

The default values are 0 for the min and max, history_data and post_process and 300 for the interval.

Notes:

The next parameters are only for the Pandora version 5 and next versions:

- <enable_unknown_events>

The next parameters are only for the Pandora version 5.1 and next versions:



- <each_ff>
- <ff_threshold_normal>
- <ff_threshold_warning>
- <ff_threshold_critical>

Example:

```
perl pandora_manage.pl /etc/pandora/pandora_server.conf --
create_snmp_module 'My module' remote_snmp_inc 'My agent'
192.168.12.123 8080 1 mycommunity myoid 'Module description'
```

Create_plugin_module

Parameters: <module_name> <module_kind> <agent_name> <module_address> <module_port> <plugin_name> <user> <password> <parameters> [<description> <module_group> <min> <max> <post_process> <interval> <warning_min> <warning_max> <critical_min> <critical_max> <history_data> <ff_threshold> <warning_str> <critical_str> <enable_unknown_events> <each_ff> <ff_threshold_normal> <ff_treshold_warning> <ff_threshold_critical> <critical_instructions> <warning_instructions> <unknown_instructions>] <use_agent_alias>

Description: A module kind plugin will be created in an agent with the module name, module kind, name of the agent where it will be created, the module address, the associated port and the corresponding plugin name. Optionally it will be possible to give it a description, the module group, min and max values, a post_process value, an interval in seconds, values warning min and max, critical values min and max and a history data value.

The values by default are 0 for min and max, history_data and post_process and 300 for the interval.

Notes:

The next parameters are only for the Pandora version 5 and next versions:

- <enable_unknown_events>

The next parameters are only for the Pandora version 5.1 and next versions:

- <each_ff>
- <ff_threshold_normal>
- <ff_threshold_warning>
- <ff_threshold_critical>

Example:



```
perl pandora_manage.pl /etc/pandora/pandora_server.conf --
create_plugin_module 'My module' generic_data 'My agent'
192.168.12.123 8080 myplugin myuser mypass 'param1 param2 param3'
'Module description' 'General' 1 3 0 300 0 0 0 0 1 'string for
warning' 'string for critical'
```

Get_module_group

Parameters: [<nombre del grupo de módulo>]

Description: Displays the available module groups. We can filter by adding the name of the group or part of it.

Example:

```
perl pandora_manage.pl /etc/pandora/pandora_server.conf --
get_module_group "grupo de módulos"
```

Create_module_group

Parameters: <module_group_name>

Description: A module group will be created with the module_group_name

Example:

```
perl pandora_manage.pl /etc/pandora/pandora_server.conf --
create_module_group 'My module group'
```

Module_group_synch

Parameters: <server_name_1|server_name_2|...server_name_n> [<return_type>]

Description: The groups of modules of the goal will be synchronized with the nodes that have been specified in the first parameter, separated with "|". Optionally you can indicate in the second parameter the format in which the data will be returned (csv or json). By default they will come in csv format.

Example:

```
perl pandora_manage.pl /etc/pandora/pandora_server.conf --
module_group_synch "server_name1|server_name2|server_name3" "json"
```



Create_synthetic

Parameters: <module_name> <synthetic_type> <agent_name> <opts>
<use_agent_alias>

Where <synthetic_type> can be "arithmetic" or "average".

Examples

```
pandora_manage /etc/pandora/pandora_server.conf --create_synthetic
example_module arithmetic example_agent <opts>
pandora_manage /etc/pandora/pandora_server.conf --create_synthetic
example_module average example_agent <opts>
```

Let's explain all the possible parameters that can be used on the <opts>

- <opts> = <source_agent1>,<operator>,<source_module1>
<source_agent2>,<operator>,<source_module2>
- <opts> = <source_agent1>,<operator>,<source_module1>
<operator>,<fixed_value>

Examples

```
pandora_manage /etc/pandora/pandora_server.conf --create_synthetic
example_module arithmetic example_agent my_agent,+,my_module
my_agent2,-,my_module2
pandora_manage /etc/pandora/pandora_server.conf --create_synthetic
example_module arithmetic example_agent my_agent,+,my_module *,10
pandora_manage /etc/pandora/pandora_server.conf --create_synthetic
example_module average example_agent my_agent,x,my_module
my_agent,x,my_module2
```

On the first example, we will create in the agent *example_agent* an arithmetic type of module named *example_module*. Its content will be the result of the operation: *my_module - my_module2*. On the second example, we will create in the agent *example_agent* an arithmetic type of module named *example_module*. Its content will be the result of the operation: *my_module * 10*. On the third example, we will create in the agent *example_agent* an average type of module named *example_module*. Its content will be the average value between *my_module* and *my_module2*.

The first operator that is used, on the previous case between *my_agent* and *my_module* (+), is ignored and doesn't affect to the result of the operations, but it is needed due to the internal structure of the tool. It's very important to keep into account that the arithmetic operation will be done with the next (in this case, the second) operator, on the current example, between *my_agent2* and *my_module2*, or right before the value *10* (in



this case, operators “-” or “*”). The possible operators are:

- + sum
- - subtraction
- * multiplication
- / division
- x average. This operator only can be used with the *average* type.

Note:

Available in Pandora 5.1 SP4 and above

Example:

```
pandora_manage /etc/pandora/pandora_server.conf --create_synthetic memory_sum arithmetic nodo51sp3 production1,+ ,AvailableMemory production2,+ ,AvailableMemory
```

```
pandora_manage /etc/pandora/pandora_server.conf --create_synthetic traffic_difference arithmetic nodo51sp3 nodo51sp3,+ ,InputTraffic nodo51sp3,- ,OutputTraffic
```

Delete_module

Parameters: <module_name> <agent_name> <use_agent_alias>

Description: An agent module will be eliminated considering both as parameters

Example:

```
perl pandora_manage.pl /etc/pandora/pandora_server.conf --delete_module 'My module' 'My agent'
```

Data_module

Parameters: <server_name> <agent_name> <module_name> <module_type> <module_new_data> [<datehour>] <use_agent_alias>

Description: It'll be send data to an agent module giving it as parameter the server name, the agent, the module name, the type of module amd the new data to be inserted. Optionally, it'll be possible to send the date-hour that will be as that of the data sending with 24 hours format: 'YYY-MM-DD HH:mm'. In the case of not sending this parameter, the current data will be shown.

Example:



```
perl pandora_manage.pl /etc/pandora/pandora_server.conf --  
data_module Servidor-General 'My agent' 'My modulo' 'generic_data'  
1 '2010-05-31 15:53'
```

Get_module_data

(>=5.0)

Parameters: <agent_name> <module_name> <interval> [<csv_separator>]
<use_agent_alias>

Description: Will be returned the data of a module as 'timestamp data' in CSV format of the last X seconds (interval) using as default separator ','

Example:

```
perl pandora_manage.pl /etc/pandora/pandora_server.conf --  
get_module_data 'agent name' 'module name' 86400 ':'
```

Get_module_id

Parameters: <agent_id> <module_name>

Description: The id of a module of an agent is returned.

Example:

```
perl pandora_manage.pl /etc/pandora/pandora_server.conf --  
get_module_id 4 'host alive'
```

Get_module_custom_id

Parameters: <agentmodule_id>

Description: The value of the field *Custom ID* of a specific module of an agent will be returned.

Example:

```
perl pandora_manage.pl /etc/pandora/pandora_server.conf --  
get_module_custom_id 4
```




Set_module_custom_id

Parameters: <agentmodule_id> [<custom_id>]

Description: The value of the field *Custom ID* of a specific module of an agent will be inserted. If it is left blank it will eliminate the content that would have previously left the field empty.

Example:

```
perl pandora_manage.pl /etc/pandora/pandora_server.conf --
set_module_custom_id 4 '21'
```

Delete_data

Parameters: <module_name> <agent_name> | -a <agent_name> | -g <group_name> <use_agent_alias>

Description: All data associated to a module will be deleted from the historical data in case of having as parameter -m and the name of this one and its agent name; from the agent modules if as parameter the option '-a' is given, and the agent or modules name of all agents from a group, if as parameter the option '-g' and the group name is given.

Example:

```
perl pandora_manage.pl /etc/pandora/pandora_server.conf -delete_data -a 'My agent'
```

In this example all historic data will be deleted from all modules that belongs to the 'My agent' agent.

Update_module

Parameters: <module_name> <agent_name> <field_to_update> <new_value> <use_agent_alias>

Description: A given field of an existent data module will be updated. The module type will be detected to allow update the specific fields for each type.

The possible fields are the following:

- **Common to any module:** module_name, agent_name, description, module_group, min, max, post_process, history_data, interval, warning_min, warning_max, critical_min, critical_max, warning_str, critical_str, ff_threshold, each_ff, ff_threshold_normal, ff_threshold_warning, ff_threshold_critical, critical_instructions,



warning_instructions, unknown_instructions

- **For the data modules:** ff_timeout
- **For the network modules:** module_address, module_port
- **For the SNMP modules:** module_address, module_port, version, community, oid, snmp3_priv_method, snmp3_priv_pass, snmp3_sec_level, snmp3_auth_method, snmp3_auth_user, snmp3_priv_pass
- **For the plugin modules:** module_address, module_port, plugin_name, user, password, parameters

Example:

```
perl pandora_manage.pl /etc/pandora/pandora_server.conf --
update_module 'Module name' 'Agent name' description 'New
description'
```

Get_agents_module_current_data

(>=5.0)

Parameters: <module_name>

Description: Get the agent and current data of all the modules with a given name.

Example:

```
perl pandora_manage.pl /etc/pandora/pandora_server.conf --
get_agents_module_current_data 'Module name'
```

Create_network_module_from_component

(>=5.0)

Parameters: <agent_name> <component_name> <use_agent_alias>

Description: Create a new network module in the specified agent from a network component.

Example:

```
perl pandora_manage.pl /etc/pandora/pandora_server.conf --
create_network_module_from_component 'Agent name'
```



Create_data_module_from_local_component

(>=5.1)

Parameters: <agent_name> <component_name> <use_agent_alias>

Description: Create a new data module in the specified agent from a local component.

Example:

```
perl pandora_manage.pl /etc/pandora/pandora_server.conf --
create_data_module_from_local_component 'Agent name' 'My local
component'
```

Create_local_component

(>=5.1)

Parameters: <component_name> [<description> <id_os> <os_version> <id_network_component_group> <type> <min> <max> <module_interval> <id_module_group> <history_data> <min_warning> <max_warning> <str_warning> <min_critical> <max_critical> <str_critical> <min_ff_event> <post_process> <unit> <wizard_level> <critical_instructions> <warning_instructions> <unknown_instructions> <critical_inverse> <warning_inverse> <id_category> <disabled_types_event> <tags> <min_ff_event_normal> <min_ff_event_warning> <min_ff_event_critical> <each_ff> <ff_timeout>]

Description: Create a new local component.

Example:

```
perl pandora_manage.pl /etc/pandora/pandora_server.conf --
create_local_component 'New local component'
'module_begin\nmodule_name name\nmodule_type
generic_data\nmodule_exec exec\nmodule_end'
```

Apply_module_template

Parameters: [<id_template> <id_agent>]

Description: Tis module templates are an agrupation that contenaís network modules. This templates can be applied directly to agents, without having to include each module onw by one.



Example:

```
perl pandora_manage.pl /etc/pandora/pandora_server.conf --  
apply_module_template 1 1
```

Alerts

Create_template_module

Parameters: <template_name> <module_name> <agent_name> <use_agent_alias>

Description: A template will be assigned to an agent module giving it the template name, the module and the agent as parameters.

Example:

```
perl pandora_manage.pl /etc/pandora/pandora_server.conf --  
create_template_module template001 'My module' 'My agent'
```

Delete_template_module

Parameters: <template_name> <module_name> <agent_name> <use_agent_alias>

Description: it'll be unassigned a module template of one agent giving it the template name, the module and the agent as parameters.

Example:

```
perl pandora_manage.pl /etc/pandora/pandora_server.conf --  
delete_template_module plantilla001 'Mi modulo' 'Mi agente'
```

Create_template_action

Parameters: <action_name> <template_name> <module_name> <agent_name>
[<fires_min> <fires_max>] <use_agent_alias>

Description: It'll be added an action to an alert giving as parameter the name of the action and that of the template, module and agent that composes the alert. It'll be also possible giving it in an optional way the values of scaling fires_min and fires_max (by default 0).

Example:



```
perl pandora_manage.pl /etc/pandora/pandora_server.conf --  
create_template_action action012 template001 'My module' 'My agent'  
0 4
```

Delete_template_action

Parameters: <action_name> <template_name> <module_name> <agent_name>
<use_agent_alias>

Description: It'll be added an action to an alert giving as parameters the names of the action, template, module and agent that composes the alert.

Example:

```
perl pandora_manage.pl /etc/pandora/pandora_server.conf --  
delete_template_action action012 template001 'My module' 'My agent'
```

Disable_alerts

Parameters: No

Description: All alerts will be disabled with the execution of this option. If when it's executed we have any alert disabled and we activate all again, this one will be also enabled.

Example:

```
perl pandora_manage.pl /etc/pandora/pandora_server.conf --  
disable_alerts
```

Enable_alerts

Parameters: No

Description: All the alerts will be activated with the execution of this option. If when it's executed we had any alert enabled and we disabled all again, this one will be also disabled.

Example:

```
perl pandora_manage.pl /etc/pandora/pandora_server.conf --  
enable_alerts
```



Create_alert_template

Parameters: <template_name> <condition_type_serialized> <time_from> <time_to> [
<description> <group_name> <field1> <field2> <field3> <priority> <default_action>
<days> <time_threshold> <min_alerts> <max_alerts> <alert_recovery>
<field2_recovery> <field3_recovery> <condition_type_separator>]

Description: An alert template will be created.

The field <condition_type_serialized> is the type options of the template serialized with the separator ';' by default. It's possible to change the separator with the parameter <condition_type_separator> to avoid conflicts with some options if it could contain the default character.

The possibilities are the following:

NOTE: In this examples is used the default separator ';' and the field matches_value is a binary value to set if the alert will be fired when the value match or when the value not match with the conditions.

- **Regular expression:**
 - **Syntax:** <type>;<matches_value>;<value>
 - **Example:** regex;1;stopped|error (Alert when value match regexp 'stopped|error')
- **Max and min:**
 - **Syntax:** <type>;<matches_value>;<min_value>;<max_value>
 - **Example:** max_min;0;30;50 (Alert when value is out of interval 30-50)
- **Max.:**
 - **Syntax:** <type>;<max_value>
 - **Example:** max;70 (Alert when value is above 70)
- **Min.:**
 - **Syntax:** <type>;<min_value>
 - **Example:** min;30 (Alert when value is below 30)
- **Equal to:**
 - **Syntax:** <type>;<value>
 - **Example:** equal;0 (Alert when value is equal to 0)
- **Not equal to:**
 - **Syntax:** <type>;<value>
 - **Example:** not_equal;100 (Alert when value is not equal to 100)
- **Warning status:**
 - **Syntax:** <type>
 - **Example:** warning (Alert when status turns into warning)
- **Critical status:**
 - **Syntax:** <type>
 - **Example:** critical (Alert when status turns into critical)
- **Not normal:**



- **Syntax:** <type>
- **Example:** not_normal (Alert when status differs from normal)
- **Unknown status:**
 - **Syntax:** <type>
 - **Example:** unknown (Alert when status turns into unknown)
- **On Change:**
 - **Syntax:** <type>;<matches_value>
 - **Example:** onchange;1 (Alert when value changes)
- **Always:**
 - **Syntax:** <type>
 - **Example:** always (Alert all times)

The field <days> is seven binary characters that specify the days of the week when the alert will be activated. i.e.: 0000011 to activate the alert only Saturday and Sunday.

Example:

```
perl pandora_manage.pl /etc/pandora/pandora_server.conf.2011-10-25
--create_alert_template 'template name' "max_min@1@3@5" 09:00 18:00
"Email will be sent when the value is in the interval 3-5,
between 9AM and 6PM, and only the Mondays. Separator is forced to
@" "Unknown" "mail@mail.com" "subject" "message" 3 "Mail to XXX"
1000000 38600 1 2 0 // // @
```

Delete_alert_template

(>=5.0)

Parameters: <template_name>

Description: An alert template will be deleted if exists.

Example:

```
perl pandora_manage.pl /etc/pandora/pandora_server.conf --
delete_alert_template 'Template name'
```

Update_alert_template

(>=5.0)

Parameters: <template_name> <field_to_update> <new_value>

Description: A given field of an existent alert template will be updated. The possible



fields are the following: name, description, type, matches_value, value, min_value, max_value, time_threshold(0-1), time_from, time_to, monday(0-1), tuesday(0-1), wednesday(0-1), thursday(0-1), friday(0-1), saturday(0-1), sunday(0-1), min_alerts, max_alerts, recovery_notify(0-1), field1, field2, field3, recovery_field2, recovery_field3, priority(0-4), default_action, group_name.

Example:

```
perl pandora_manage.pl /etc/pandora/pandora_server.conf --
update_alert_template 'Template name' priority 4
```

Get_alert_actions

(>=5.0)

Parameters: [<action_name> <separator> <return_type>]

Description: Returns all action alerts. Optionally you can filter the result by name. Optionally in the second parameter you can specify the separator of the results and in the third parameter the format of the returned list (csv, json, string).

Example:

```
perl pandora_manage.pl /etc/pandora/pandora_server.conf --
get_alert_actions 'Action 1' ';' 'json'
```

Get_alert_actions_meta

(>=5.0)

This feature is in Metaconsole.

Parameters: [<server_name> <action_name> <separator> <return_type>]

Description: returns all alert actions of the meta console. Optionally you can filter the result by the name of the node or by the name of the action. Optionally in the second parameter you can specify the results separator and in the third parameter the format of the returned list (csv, json, string).

Example:



```
perl pandora_manage.pl /etc/pandora/pandora_server.conf --  
get_alert_actions_meta // 'Action 1' ';' 'json'
```

Create_alert_command

(>=5.0)

Parameters: <name> <command> [<group|name> <description> <internal>
<field_description> <field_value>]

Description: A command will be created with the specified name and command data. Optionally you can specify:

- Group name. If no group is provided, it will be assigned in the group 'All'.
- Command Description.
- Internal (1-0).
- Field description. This will be formatted as follows:
["descr1","descr2","descr3","descr4","descr5"]'.
- Field value: This will be formatted as follows: ["val1","val2","val3","val4","val5"].

Example:

```
perl pandora_manage.pl /etc/pandora/pandora_server.conf --  
create_alert_command 'PRUEBA' 'command' 'All' 'Desc' '1'  
'["des1","des2","des3","","des5","","","","",""]'  
'["val1","val2","val3","val4","","","","",""]'
```

Get_alert_commands

(>=5.0)

Parameters: [<name|<command> <group name> <description> <internal>]

Description: Allows you to view all alerts. Optionally you can filter the result by: name, command, group, description or if it is internal (1-0). In the case of the name, command and description it is not necessary to enter the exact data.

Example:

```
perl pandora_manage.pl /etc/pandora/pandora_server.conf --  
get_alert_commands "command_name"
```



Validate_all_alerts

(>=5.0)

Parameters: None

Description: Validate all the alerts.

Example:

```
perl pandora_manage.pl /etc/pandora/pandora_server.conf --  
validate_all_alerts
```

Create_special_day

(>=5.1)

Parameters: <special_day> <same_day> <description> <group_name>

Description: Create a special day. The possible same_days are monday, tuesday, wednesday, thursday, friday, saturday and sunday.

Example:

```
perl pandora_manage.pl /etc/pandora/pandora_server.conf --  
create_special_day 2014-05-03 sunday Desc All
```

Delete_special_day

(>=5.1)

Parameters: <special_day>

Description: Delete specified special day.

Example:

```
perl pandora_manage.pl /etc/pandora/pandora_server.conf --  
delete_special_day 2014-05-03
```

Update_special_day



(>=5.1)

Parameters: <special_day> <field_to_change> <new_value>

Description: Update specific field of a special day with new value. The possible fields are same_day, description and group_name. When same_day is set, possible new_values are monday, tuesday, wednesday, thursday, friday, saturday and sunday.

Example:

```
perl pandora_manage.pl /etc/pandora/pandora_server.conf --
update_special_day 2014-05-03 same_day monday
```

Users

Create_user

Parameters: <user_name> <password> <es_admin> [<comments>]

Description: It'll be created an user with the name and password that are received as parameters. It will be received also a binary value that specify if the user will be or will be not the administrator. Optionally, it could be also sent comments about the created user.

Example:

```
perl pandora_manage.pl /etc/pandora/pandora_server.conf --
create_user user002 'renardo' 0 'This user has renardo as password'
```

Delete_user

Parameters: <user_name>

Description: An user will be eliminated giving its name as parameter.

Example:

```
perl pandora_manage.pl /etc/pandora/pandora_server.conf --
delete_user user002
```

Update_user

(>=5.0)



Parameters: <id_user> <field_to_update> <new_value>

Description: A given field of an existent user will be updated. The possible fields are the following: email, phone, is_admin (0-1), language, id_skin, flash_chart (0-1), comments, fullname, password.

Example:

```
perl pandora_manage.pl /etc/pandora/pandora_server.conf --
update_user 'User Id' password 'New password'
```

Enable_user

(>=5.0)

Parameters: <user_id>

Description: An existent user will be enabled. If it's already enabled, will showed only a message

Example:

```
perl pandora_manage.pl /etc/pandora/pandora_server.conf --
enable_user 'User id'
```

Disable_user

(>=5.0)

Parameters: <user_id>

Description: An existent user will be disabled. If it's already disabled, will showed only a message

Example:

```
perl pandora_manage.pl /etc/pandora/pandora_server.conf --
disable_user 'User id'
```

Meta_synch_user

(>=5.0)



This feature is in Metaconsole.

Parameters: <user_name_1,user_name_2,..user_name_n> <server_name>
[<profile_mode> <group_name> <profile_1,profile_2,..profile2> <create_groups>]

Description: Synchronizes one or more users from the Metaconsole to the node. The first parameter will specify the users we want to synchronize separated by commas. The second parameter will specify the name of the node to which we want to synchronize. Optionally you can specify:

- If it is wanted to copy the original profile or profiles of the user or users (0), or another profile and new group that we will indicate in the following parameters(1). By default it will be 0
- Name of the group. The 'All' group will be specified if you want the user to be valid in all groups..
- The profiles you want to assign separated by comma.
- In case the group does not exist in the indicated node, the one of the meta console (1) will be copied, otherwise (0) it will be left empty. By default it will be 0

Example:

```
perl pandora_manage.pl /etc/pandora/pandora_server.conf --  
meta_synch_user "usuario1,usuario2" "nodo1" "1" "All" "Chief  
operator,Group coordinator" "0"
```

Create_profile

Parameters: <profile_name> <incident_view> <incident_edit>
<incident_management> <agent_view> <agent_edit> <agent_disable> <alert_edit>
<alert_management> <user_management> <db_management> <event_view>
<event_edit> <event_management> <report_view> <report_edit>
<report_management> <map_view> <map_edit> <map_management>
<vconsole_view> <vconsole_edit> <vconsole_management> <pandora_management>

Description: A new profile will be created, receiving the profile name and permissions as parameters. All fields are mandatory.

Example:

```
perl pandora_manage.pl /etc/pandora/pandora_server.conf --  
create_profile 'newProfile' 1 0 1 1 0 0 0 1 1 1 1 1 1 0 0 1 0 0 0 1  
1 1 1
```



Update_profile

Parameters: <profile_name> <incident_view> <incident_edit>
<incident_management> <agent_view> <agent_edit> <agent_disable> <alert_edit>
<alert_management> <user_management> <db_management> <event_view>
<event_edit> <event_management> <report_view> <report_edit>
<report_management> <map_view> <map_edit> <map_management>
<vconsole_view> <vconsole_edit> <vconsole_management> <pandora_management>

Description: A profile will be updated, receiving the profile name and permissions as parameters. All fields are mandatory.

Example:

```
perl pandora_manage.pl /etc/pandora/pandora_server.conf --  
update_profile 'existingProfile' 1 0 1 1 0 0 0 1 1 1 1 1 1 0 0 1 0  
0 0 1 1 1 1
```

add_profile

Parameters: <user_name> <profile_name> <group>

Description: A profile will be added to an user giving it as parameter the names of user, profile and group on which they will have the privileges of this profile. You should specify the group 'All' if you want that the profile has validity on all groups.

Example:

```
perl pandora_manage.pl /etc/pandora/pandora_server.conf --add_profile usuario002 'Group  
coordinator' All
```

Delete_profile

Parameters: <user_name> <profile_name> <group>

Description: An user profile will be deleted giving it as parameter the names of user, profile and group on which the profiles has the privileges. If the profile to delete is associated to the "ALL group", we should specify as group "All".

Example:

```
perl pandora_manage.pl /etc/pandora/pandora_server.conf --  
delete_profile usuario002 'Chief Operator' Applications
```



Add_profile_to_user

(>=5.0)

Parameters: <id_user> <profile_name> [<group_name>]

Description: A profile in a group to a user will be assigned. If the group is not provided, the grupo will be 'All'.

Example:

```
perl pandora_manage.pl /etc/pandora/pandora_server.conf --
update_user 'User Id' 'Chief Operator' 'Network'
```

Disable_aecl

Parameters: No

Description: The Enterprise mode ACL system will be disabled in the configuration with the execution of this option.

Example:

```
perl pandora_manage.pl /etc/pandora/pandora_server.conf --
disable_eacl
```

Enable_aecl

Parameters: No

Description: The Enterprise mode ACL system will be enabled in the configuration with the execution of this option.

Example:

```
perl pandora_manage.pl /etc/pandora/pandora_server.conf --
enable_eacl
```

Disable_double_auth

Parameters: <id_user>



Description: The double authentication will be disabled for the specified user.

Example:

```
perl pandora_manage.pl /etc/pandora/pandora_server.conf --
disable_double_auth 'admin'
```

Events

Create_event

Parameters: <event_name> <event_type> <group_name> [<agent_name> <module_name> <event_state> <severity> <template_name> <user_name> <comment> <source> <id_extra> <tags> <custom_data> <force_create_agent> <critical_instructions> <warning_instructions> <unknown_instructions>] <use_agent_alias>

Description: An event will be created with these data: the name and kind of the event, name of the module, agent and group associated. Optionally it could be sent:

- agent name
- module name
- event state (0 if it isn't validated and 1 if it is)
- severity (from 1 to 4)
- severity: 0 (Maintenance), 1 (Informational), 2 (Normal), 3 (Warning), 4 (Critical).

From version 5.0 there are 5 (Minor) y 6 (Major) too.

- template name in the case that is would be associated to one alert.
- user name
- comment
- source
- Extra id
- tags: Format should be <tag> <url>,<tag> <url>

You can add multiple tags separated by commas

- custom data: Custom data should be entered as a JSON document. For example: '{ "Location": "Office", "Priority": 42 }'
- Force creation of agent (bool): If the agent name parameter refers to an agent that does not exist, it will be created.
- Critical instructions: Add the instructions to be performed in the case of a critical condition
- Warning instructions: Add the instructions to be performed in the case of a warning condition



- Unknown instructions: Add the instructions to perform in case of going to unknown state

Nota: Event type could be: unknown, alert_fired, alert_recovered, alert_ceased, alert_manual_validation, recon_host_detected, system, error, new_agent, going_up_warning, going_up_criticalgoing_down_warning, going_down_normal, going_down_critical, going_up_normal, configuration_change.

Example:

```
perl pandora_manage.pl /etc/pandora/pandora_server.conf --
create_event ' CLI Event' system Firewalls 'My agent' 'My module' 0
4 Template004
```

Validate_event

Parameters: <agent_name> <module_name> <datehour_min> <datehour_max> <name_user> <criticity> <template_name> <use_agent_alias>

Description: All events will be validated considering a group of filters. The configurable filters are: the agent name, the module name, date-hour minimum and date-hour maximum, the user name, the severity and the name of the associated template.

It's possible to combine the parameters in several ways, leaving blank with empty inverted commas () *the ones that you don't want to use and filling in the rest.* **Example:** `perl pandora_manage.pl /etc/pandora/pandora_server.conf -validate_event 'My agent' 'My module' '2010-06-02 22:02'`

In this example will be validated all the events associated to the module 'Mi module' of the agent 'My agent' which data would be previous to 2 june 2010 not considering the rest of the filters. It would be also possible to filter the events between two dates filling both of them or the ones that have a data higher to an specific one, filling in only the date-hour minimum.

Validate_event_id

(>=5.0)

Parameters: <id_event>

Description: A event will be validated.

Example:



```
perl pandora_manage.pl /etc/pandora/pandora_server.conf --  
validate_event_id 1894
```

In this example, will be validated the event whose identifier is 1894.

Get_event_info

(>=5.0)

Parameters: <id_event>[<separator>]

Description: Display info about a event given a id.

Example:

```
perl pandora_manage.pl /etc/pandora/pandora_server.conf --  
get_event_info 1894
```

In this example, will be displayed info about the event whose identifier is 1894. The fields will be separated by |

Add_event_comment

(>=5.1)

Parameters: <id_event> <user_name> <comment>

Description: Add an event comment.

Example:

```
perl pandora_manage.pl /etc/pandora/pandora_server.conf --  
add_event_comment 1234 admin TEXT
```

In this example, comment 'TEXT' by 'admin' will be added to the event whose identifier is 1234.

Incidents

Create_incident



(>=5.0)

Parameters: <title> <description> <origin> <status> <priority> <group> [<owner>]

Description: An incident will be created passing the title, the description, the origin, the status, the priority, the group and optionally the owner to it.

The priority will be a number according to the following correspondence:

0: Informative; 1: Low; 2: Medium; 3: Important; 4: Very important; 5: Maintenance

the status will be a number according to the following correspondence:

0: Active incident; 1: Active incident with comments; 2: Rejected incident ; 3: Expired incident; 13: Closed incident

Example:

```
perl pandora_manage.pl /etc/pandora/pandora_server.conf --
create_incident 'Incident' 'Incident Description' 'Other data
source' 3 2 'id_owner_user'
```

Policies

Create_policy

Parameters: <policy_name> <group> <description>

Description: A policy will be created with policy name, group and description. The description is not mandatory.

Example:

```
perl pandora_manage.pl /etc/pandora/pandora_server.conf --
create_policy 'My policy' 'My group' 'Description'
```

Apply_policy

Parameters: <policy_name>

Description:A policy passed as parameter will be forcibly applied. Within the process of applying a policy is: the creation of the modules of the policy in all its associated agents, the creation of the alerts of the policy in the created modules, and the accomplishment of changes in the configuration file of the local agents that the policy may have to add the



created modules and the collections associated to the policy.

As this command works through the API, it needs the following parameters:

- `id =<id_policy>` (required)
- `id2=<id_agent>` (optional) Id or name of the agent as indicated in the other parameter. If it is empty, the policy will be applied to all the agents of this.
- `other =<serialized parameters>`
 - `<name_agent (Boolean)>` Indicates if the agent will be sent by Id (0) or by name (1).
 - `<server_id>` (required if using Metaconsole) Id of the server in which the policy will be applied.

Example:

```
perl pandora_manage.pl /etc/pandora/pandora_server.conf --
apply_policy 'policy_id' 'agent_id' 'other1|other2'
```

Apply_all_policies

(>=5.0)

Parameters: None

Description: Add to the application queue all the policies. The server is who watch the queue and apply the policies

Example:

```
perl pandora_manage.pl /etc/pandora/pandora_server.conf --
apply_all_policies
```

Add_agent_to_policy

(>=5.0)

Parameters: <agent_name> <policy_name> <use_agent_alias>

Description: An existent group will be added to an existent policy

Example:

```
perl pandora_manage.pl /etc/pandora/pandora_server.conf --
add_agent_to_policy 'Agent name' 'Policy name'
```



Add_collection_to_policy

Parameters: <policy_name> <collection_name>

Description: An existent collection will be added to an existent policy

Example:

```
perl pandora_manage.pl /etc/pandora/pandora_server.conf --
add_collection_to_policy 'Policy name' 'Collection name'
```

Create_policy_data_module_from_local_component

Parameters: <policy_name> <component_name>

Description: A module will be created from a local component to a policy

Example:

```
perl pandora_manage.pl /etc/pandora/pandora_server.conf --
create_policy_data_module_from_local_component 'Policy name'
'Component name'
```

Delete_not_policy_modules

Parameters: Not

Description: All modules that doesn't belong to any policy will be deleted both from the database and the agent configuration file (if there is one).

Example:

```
perl pandora_manage.pl /etc/pandora/pandora_server.conf --
delete_not_policy_modules
```

Remove_agent_from_policy

Parameters: <id_policy> <id_agent>

Description: Removes an agent from a policy. It is necessary to specify the id of the policy and the id of the agent to be deleted.



Example:

```
perl pandora_manage.pl /etc/pandora/pandora_server.conf --
remove_agent_from_policy '11' '2'
```

Disable_policy_alerts

Parameters: <policy_name>

Description: All the alerts from a policy passed by parameter will be flagged as disabled

Example:

```
perl pandora_manage.pl /etc/pandora/pandora_server.conf --
disable_policy_alerts 'My policy'
```

Create_policy_data_module

(>=5.0)

Parameters: <policy_name> <module_name> <module_type> [<description>
<module_group> <min> <max> <post_process> <interval> <warning_min>
<warning_max> <critical_min> <critical_max> <history_data> <data_configuration>
<warning_str> <critical_str> <enable_unknown_events> <ff_threshold> <each_ff>
<ff_threshold_normal> <ff_threshold_warning> <ff_threshold_critical> <ff_timeout>
<critical_instructions> <warning_instructions> <unknown_instructions>

Description: A policy data module will be created. The default values are the same of -create_data_module option

Notes:

The next parameters are only for the Pandora version 5.1 and next versions:

- <ff_threshold>
- <each_ff>
- <ff_threshold_normal>
- <ff_threshold_warning>
- <ff_threshold_critical>
- <ff_timeout>

Example:

```
perl pandora_manage.pl /etc/pandora/pandora_server.conf --
```



```
create_policy_data_module 'policy name' 'module name' generic_proc
'module description' 'group name' 0 100 0 300 30 60 61 100 0
"module_begin\nmodule_name modname\nmodule_end" 'string for
warning' 'string for critical'
```

Create_policy_web_module

(>=7.0)

Parameters: <policy_name> <module_name> <module_type> [<description>
<module_group> <min> <max> <post_process> <interval> <warning_min>
<warning_max> <critical_min> <critical_max> <history_data> <retries> <requests>
<agent_browser_id> <auth_server> <auth_realm> <configuration_data> <proxy_url>
<proxy_auth_login> <proxy_auth_password> <warning_str> <critical_str>
<enable_unknown_events> <ff_threshold> <each_ff> <ff_threshold_normal>
<ff_threshold_warning> <ff_threshold_critical> <ff_timeout> <warning_inverse>
<critical_inverse> <critical_instructions> <warning_instructions>
<unknown_instructions>]

Description: A policy web module will be created. The default values are the same of
-create_web_module option

Example:

```
perl pandora_manage.pl /etc/pandora/pandora_server.conf --
create_policy_web_module 'policy_name' 'module_name' web_data
'module_description' 'General' 0 100 0 300 0 0 0 1 0 1 'Pandora
FMS' auto public "module_begin\nmodule_name modname\nmodule_end"
'http://proxy.url' 'proxy_login' 'proxy_password' 1 10 10 10 10
10 10 10 1 1 'critical_instructions' 'warning_instructions'
'unknown_instructions'
```

Create_policy_network_module

(>=5.0)

Parameters: <policy_name> <module_name> <module_type> [<module_port>
<description> <module_group> <min> <max> <post_process> <interval>
<warning_min> <warning_max> <critical_min> <critical_max> <history_data>
<ff_threshold> <warning_str> <critical_str> <enable_unknown_events> <each_ff>
<ff_threshold_normal> <ff_threshold_warning> <ff_threshold_critical>
<critical_instructions> <warning_instructions> <unknown_instructions>]

Description: A policy network module will be created. The default values are the same of



-create_network_module option

Notes:

The next parameters are only for the Pandora version 5.1 and next versions:

- <each_ff>
- <ff_threshold_normal>
- <ff_threshold_warning>
- <ff_threshold_critical>

Example:

```
perl pandora_manage.pl /etc/pandora/pandora_server.conf --
create_policy_network_module 'policy name' 'module name'
remote_icmp_proc 22 'module description' 'group name' 0 100 0 300
30 60 61 100 0 0 'string for warning' 'string for critical'
```

Create_policy_snmp_module

(>=5.0)

Parameters: <policy_name> <module_name> <module_type> <module_port>
<version> [<community> <oid> <description> <module_group> <min> <max>
<post_process> <interval> <warning_min> <warning_max> <critical_min>
<critical_max> <history_data> <snmp3_priv_method> <snmp3_priv_pass>
<snmp3_sec_level> <snmp3_auth_method> <snmp3_auth_user> <snmp3_priv_pass>
<ff_threshold> <warning_str> <critical_str> <enable_unknown_events> <each_ff>
<ff_threshold_normal> <ff_threshold_warning> <ff_threshold_critical>
<critical_instructions> <warning_instructions> <unknown_instructions>]

Description: A policy SNMP module will be created. The default values are the same of -create_snmp_module option

Notes:

The next parameters are only for the Pandora version 5.1 and next versions:

- <each_ff>
- <ff_threshold_normal>
- <ff_threshold_warning>
- <ff_threshold_critical>

Example:

```
perl pandora_manage.pl /etc/pandora/pandora_server.conf --
```




```
create_policy_snmp_module 'policy name' 'module name'  
remote_snmp_inc 8080 1 mycommunity myoid 'Module description'
```

Create_policy_plugin_module

(>=5.0)

Parameters: <policy_name> <module_name> <module_kind> <module_port>
<plugin_name> <user> <password> <parameters> [<description> <module_group>
<min> <max> <post_process> <interval> <warning_min> <warning_max>
<critical_min> <critical_max> <history_data> <warning_str> <critical_str>
<enable_unknown_events> <each_ff> <ff_threshold_normal> <ff_threshold_warning>
<ff_threshold_critical> <critical_instructions> <warning_instructions>
<unknown_instructions>]

Description: A policy plugin module will be created. The default values are the same of
-create_plugin_module option

Notes:

The next parameters are only for the Pandora version 5.1 and next versions:

- <each_ff>
- <ff_threshold_normal>
- <ff_threshold_warning>
- <ff_threshold_critical>

Example:

```
perl pandora_manage.pl /etc/pandora/pandora_server.conf --  
create_policy_plugin_module 'policy name' 'module name'  
generic_data 22 myplugin myuser mypass 'param1 param2 param3'  
'Module description' 'General' 1 3 0 300 0 0 0 0 1 'string for  
warning' 'string for critical'
```

Validate_policy_alerts

(>=5.0)

Parameters: <policy_name>

Description: Validate all the alerts of a given policy

Example:



```
perl pandora_manage.pl /etc/pandora/pandora_server.conf --  
validate_policy_alerts 'Policy name'
```

Get_policy_modules

(>=5.0)

Parameters: <policy_name>

Description: Get the module list (id and name) of a given policy

Example:

```
perl pandora_manage.pl /etc/pandora/pandora_server.conf --  
get_policy_modules 'Policy name'
```

Get_policies

(>=5.0)

Parameters: [<agent_name> <use_agent_alias>]

Description: Get all the policies (without parameters) or the policies of a given agent (agent name as parameter)

Example:

```
perl pandora_manage.pl /etc/pandora/pandora_server.conf --  
get_policies 'Agent name'
```

Netflow

Create_netflow_filter

(>=5.0)

Parameters: <filter_name> <group_name> <filter> <aggregate_by> <output_format>

Description: Create a new netflow filter.

The possible values of aggregate_by parameter are: dstip,dstport,none,proto,srcip,srcport
The possible values of ouput_format parameter are:



kilobytes,kilobytespersecond,megabytes,megabytespersecond

Example:

To create a netflow filter we execute the following option:

```
perl pandora_manage.pl /etc/pandora/pandora_server.conf --
create_netflow_filter "Filter name" Network "host 192.168.50.3 OR
host 192.168.50.4 or HOST 192.168.50.6" dstport kilobytes
```

Tools

Exec_from_file

(>=5.0)

Parameters: <file_path> <option_to_execute> <option_params>

Description: With this option is possible to execute any CLI option with macros from a CSV file. The number of macros will be the number of columns in the CSV file. Each macro will be named FIELD1 , FIELD2 , FIELD3 etc.

Example:

We are going to create users from a CSV file.

We need a CSV file like that:

```
User 1,Password 1,0
User 2,Password 2,0
User 3,Password 3,0
User Admin,Password Admin,1
```

The name of the file will be '/tmp/users_csv'

We are going to execute the option -create_user with the following options: <user_name> <user_password> <is_admin> <comments>

To do this, we execute the following option:

```
perl pandora_manage.pl /etc/pandora/pandora_server.conf --
exec_from_file /tmp/users_csv create_user __FIELD1__ __FIELD2__
__FIELD3__ 'User created with exec_from_file option from CLI'
```

NOTE: Commas into the CSV columns are not yet supported



create_snmp_trap

(>=5.0)

Parameters: <file_path> <name> <oid> <desc> <severity>

Name: As seen in the snmp trap console.

OID: SNMP trap main OID.

Severity: Numeric value, which have following values: Severity 0 (Maintenance), 1(Info) , 2 (Normal), 3 (Warning), 4 (Critical), 5 (Minor) and 6 (Major).

Sample:

```
perl pandora_manage.pl /etc/pandora/pandora_server.conf --
create_snmp_trap Cisco_FAN_Crash 1.3.3.3.2.12.3.3.4.1 "Something
happen with the FAN inside the CISCO device, probably a failure" 3
```

Graphs

create_custom_graph

Parameters: <name> <description> <user> <id_group> <width> <height> <events> <graph_type> <period> <modules> <separator>

Description: You can create a graph with these elements. All parameters are required, but they can be empty by singles quotes. Their default values are:

width: 550, height: 210, period: 86400 (seconds), events: 0, graph_type: 0, id_group: 0

Example:

```
perl pandora_manage.pl /etc/pandora/pandora_server.conf --
create_custom_graph 'My graph' 'Created by CLI' 'admin' 0 '' '' 0 2
'' '1;2;5;30' ''
```

edit_custom_graph

Parameters: <id_graph> <name> <description> <user> <id_group> <width> <height> <events> <graph_type> <period>

Description: You can edit a graph with these values. All parameters are required, but



they can be empty by singles quotes. Fields not specified keep their values.

Example:

```
perl pandora_manage.pl /etc/pandora/pandora_server.conf --
edit_custom_graph 12 '' 'edit graph by CLI' '' '' '' '' '' '' 25200
```

add_modules_to_graph

Parameters: <id_graph> <modules> <separator>

Description: These modules will be added to the graph. All parameters are required.

Example:

```
perl pandora_manage.pl /etc/pandora/pandora_server.conf --
add_modules_to_graph 12 '25,26' ','
```

delete_modules_to_graph

Parameters: <id_graph> <modules> <separator>

Description: These modules will be removed to the graph. All parameters are required.

Example:

```
perl pandora_manage.pl /etc/pandora/pandora_server.conf -delete_modules_to_graph 12
'1,25,26' ','
```

Clusters

new_cluster

Parameters: <cluster_name> <cluster_type> <description> <group_id>

Description: A cluster with the specified parameters and their associated elements will be created.

Example:

```
perl pandora_manage.pl /etc/pandora/pandora_server.conf --
new_cluster cluster1 AA description 12
```




```
perl pandora_manage.pl /etc/pandora/pandora_server.conf --
add_cluster_agent
W3sibmFtZSI6IkRpc2tVc2VkXy9wcm9jL2tjb3JlIiwiaWRfY2x1c3RlciI6NSwidHl
wZSI6IkF0IiwiaXNfY3JpdGljYWwi0jF9LAp7Im5hbWUiOiJEaXNrVXNlZF8vcHJvYy
9zY2hlZF9kZWJ1ZyIsImlkX2NsdXN0ZXIiOi0jUsInR5cGUiOiJBUCIsImIzX2NyaXRpY
2FsIjoxfV0=
```

delete_cluster

Parameters: <cluster_id>

Description: A cluster will be deleted.

Example:

```
perl pandora_manage.pl /etc/pandora/pandora_server.conf --
delete_cluster_item 1
```

delete_cluster_agent

Parameters: <agent_id> <cluster_id>

Description: An agent added to a cluster will be disassociated

Example:

```
perl pandora_manage.pl /etc/pandora/pandora_server.conf --
delete_cluster_agent 1 1
```

delete_cluster_item

Parameters: <item_id>

Description: A cluster item will be deleted

Example:

```
perl pandora_manage.pl /etc/pandora/pandora_server.conf --
delete_cluster_item 1
```



Visual Consoles

create_visual_console

Parameters: <name> <background> <width> <height> <group> <mode>
[<position_to_locate_elements><background_color> <elements>]]

Description: A visual console will be created with the specified parameters and their associated elements. It is absolutely necessary that the JSONs go in single quotes because if they don't, there could be errors in the shell from which the script is launched.

"Interesting data:" The mode can adopt the values 'static_objects' and 'auto_creation', in the first way the elements will be created without taking into account the position where to place the elements, but the position described in the elements json itself will be used. However, if the mode is 'auto_creation' a coordinate tapestry ('position_to_locate_elements' field with the format described in the example below) must be set and the algorithm will automatically establish the position and size of each of the elements.

Example (auto_creation):

```
perl pandora_manage.pl /etc/pandora/pandora_server.conf -create_visual_console 'test console' 'mapamundo.jpg' 1024 768 12 'auto_creation'
'{"pos1x":0,"pos1y":60,"pos2x":800,"pos2y":600}' '#FFF' '[{"image": "_engine","height": 0,"width": 0,"label": "", "type": 0,"period": 300,"id_agent": 1,"id_agente_modulo": 0,"id_group": 0,"id_layout_linked": 0,"parent_item": 0,"enable_link": 1,"id_metaconsole": 0,"id_custom_graph": 0,"border_width": 0,"border_color": "", "fill_color": "", "type_graph": "area","label_position": "down"}, {"image": "_printer","height": 0,"width": 0,"label": "", "type": 0,"period": 300,"id_agent": 2,"id_agente_modulo": 0,"id_group": 0,"id_layout_linked": 0,"parent_item": 0,"enable_link": 1,"id_metaconsole": 0,"id_custom_graph": 0,"border_width": 0,"border_color": "", "fill_color": "", "type_graph": "area","label_position": "down"}]
```

Example (static_objects):

```
perl pandora_manage.pl /etc/pandora/pandora_server.conf --
create_visual_console 'test console' 'mapamundo.jpg' 1024 768 12
'static_objects' "" '#FFF' '[{"image":
"_engine","pos_x":100,"pos_y":100,"height": 0,"width": 0,"label":
"", "type": 0,"period": 300,"id_agent": 1,"id_agente_modulo":
0,"id_group": 0,"id_layout_linked": 0,"parent_item":
0,"enable_link": 1,"id_metaconsole": 0,"id_custom_graph":
0,"border_width": 0,"border_color": "", "fill_color":
"", "type_graph": "area","label_position": "down"}, {"image":
"_printer","pos_x":400,"pos_y":100,"height": 0,"width": 0,"label":
```




```
"","type": 0,"period": 300,"id_agent": 2,"id_agente_modulo":  
0,"id_group": 0,"id_layout_linked": 0,"parent_item":  
0,"enable_link": 1,"id_metaconsole": 0,"id_custom_graph":  
0,"border_width": 0,"border_color": "", "fill_color":  
"","type_graph": "area","label_position": "down"]}]'
```

edit_visual_console

Parameters: <id> [<name> <background> <width> <height> <group> <mode>
<position_to_locate_elems> <background_color> <elements>]

Description: A visual console with the specified parameters and its associated elements will be edited. It is absolutely necessary that the JSONs go between single quotes since if it didn't occur it could cause errors in the shell from where the script is launched.

Example (auto_creation):

```
perl pandora_manage.pl /etc/pandora/pandora_server.conf --  
edit_visual_console 1 'test console edited' 'auto_creation'  
{"pos1x":0,"pos1y":60,"pos2x":800,"pos2y":600}  
'[{"id":103,"image": "_engine","height": 180,"width": 200,"label":  
"","type": 0,"period": 300,"id_agent": 3,"id_agente_modulo":  
0,"id_group": 0,"id_layout_linked": 0,"parent_item":  
0,"enable_link": 1,"id_metaconsole": 0,"id_custom_graph":  
0,"border_width": 0,"border_color": "", "fill_color":  
"","type_graph": "area","label_position":  
"down"}, {"pos_x":400,"pos_y":100,"image": "_printer","height":  
0,"width": 0,"label": "", "type": 0,"period": 300,"id_agent":  
4,"id_agente_modulo": 0,"id_group": 0,"id_layout_linked":  
0,"parent_item": 0,"enable_link": 1,"id_metaconsole":  
0,"id_custom_graph": 0,"border_width": 0,"border_color":  
"","fill_color": "", "type_graph": "area","label_position":  
"down"}]'
```

Example (static_objects):

```
perl pandora_manage.pl /etc/pandora/pandora_server.conf --  
edit_visual_console 1 'test console edited' 'static_objects'  
'[{"id":103,"image": "_engine","height": 180,"width": 200,"label":  
"","type": 0,"period": 300,"id_agent": 3,"id_agente_modulo":  
0,"id_group": 0,"id_layout_linked": 0,"parent_item":  
0,"enable_link": 1,"id_metaconsole": 0,"id_custom_graph":  
0,"border_width": 0,"border_color": "", "fill_color":  
"","type_graph": "area","label_position":
```



```
"down"}, {"pos_x":400,"pos_y":100,"image": "_printer","height":  
0,"width": 0,"label": "", "type": 0,"period": 300,"id_agent":  
4,"id_agente_modulo": 0,"id_group": 0,"id_layout_linked":  
0,"parent_item": 0,"enable_link": 1,"id_metaconsole":  
0,"id_custom_graph": 0,"border_width": 0,"border_color":  
"", "fill_color": "", "type_graph": "area", "label_position":  
"down"}]'
```

delete_visual_console

Parameters: <id>

Description: The specified visual console and its associated elements will be removed.

Example:

```
perl pandora_manage.pl /etc/pandora/pandora_server.conf --  
delete_visual_console 1
```

delete_visual_console_objects

Parameters: <id> <mode> <id_mode>

Description: Items associated with a visual console will be removed by following a filter. The available modes are “type”, “image”, “id_agent”, “id_agente_modulo”, “id_group” and “type_graph” where its identifier must be specified, which may be the type ID, the image name, the agent ID, the module ID, the group ID or the type of graphic. (area, line...).

Example:

```
perl pandora_manage.pl /etc/pandora/pandora_server.conf --  
delete_visual_console_objects 1 'id_agent' 1
```

duplicate_visual_console

Parameters: <id> <times> [<prefix>]

Description: The indicated visual console will be duplicated as many times as desired.

Example:

```
perl pandora_manage.pl /etc/pandora/pandora_server.conf --
```



```
duplicate_visual_console 1 10 'test console 2'
```

export_json_visual_console

Parameters: <id> [<path> <with_element_id>]

Description: Exports to json a visual console and its elements being able to specify the path where to save the file and if you want to include the id of the associated elements to facilitate a later edition.

Interesting data: The field “with_element_id” can return 1 and 0, which mean with or without ID respectively.

Example:

```
perl pandora_manage.pl /etc/pandora/pandora_server.conf --  
export_json_visual_console 1 '/tmp' 1
```

Others

meta_get_agent_group

Parameters: <host> <api_pass> <user> <pass> <agent_name>

Description: Return the group name of the agent.

Example:

```
perl pandora_manage.pl /etc/pandora/pandora_server.conf --  
meta_get_agent_group http://127.0.0.1/pandora_console 1234 admin  
pandora NAS
```

meta_get_agent_group_id

Parameters: <host> <api_pass> <user> <pass> <agent_name>

Description: Return the group id of the agent.

Example:

```
perl pandora_manage.pl /etc/pandora/pandora_server.conf --  
meta_get_agent_group_id http://127.0.0.1/pandora_console 1234 admin
```



```
pandora NAS
```

meta_delete_agent

Parameters: <host> <api_pass> <user> <pass> <agent_name>

Description: Delete an agent on metaconsole and node.

Example:

```
perl pandora_manage.pl /etc/pandora/pandora_server.conf --  
meta_delete_agent http://127.0.0.1/pandora_console 1234 admin  
pandora NAS
```

locate_agent

Parameters: <agent_name> <use_agent_alias>

Description: Return the id of the node that manages the agent.

Example:

```
perl pandora_manage.pl /etc/pandora/pandora_server.conf --  
locate_agent NAS
```

meta_set_create_group

Parameters: <url_host> <api_pass> <user> <pass> <group_name> (<icon_name> | "<icon_name>|<id_group_parent>")

Description: Create group and return the id of the new group.

Example:

```
perl pandora_manage.pl /etc/pandora/pandora_server.conf --  
meta_set_create_group http://127.0.0.1/pandora_console 1234 admin  
pandora salamanca "world|14"
```

```
perl pandora_manage.pl /etc/pandora/pandora_server.conf --  
meta_set_create_group http://127.0.0.1/pandora_console 1234 admin  
pandora salamanca world
```



Help

To obtain general help with the Pandora FMS CLI you only need to write:

```
perl pandora_manage.pl --h
```

To obtain help of one specific option, it would be enough with putting this option without parameters (this for the options that use parameters).

```
perl pandora_manage.pl /etc/pandora/pandora_server.conf --  
create_user
```

[Go back to Pandora FMS documentation index](#)



From:

<https://pandorafms.com/manual/> - **Pandora FMS Documentation**

Permanent link:

https://pandorafms.com/manual/en/documentation/08_technical_reference/03_anexo_cli

Last update: **2021/09/16 09:17**