



Pandora FMS Command Line Interface (CLI)



om:

<https://pandorafms.com/manual/!780/>

ermanent link:

https://pandorafms.com/manual/!780/en/documentation/pandorafms/technical_reference/03_anexo_cli

025/03/04 21:22



Pandora FMS Command Line Interface (CLI)

Pandora FMS CLI

The Pandora FMS CLI (Command-Line Interface) is used for making calls in command line on the file `pandora_manage`.

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.

```
pandora_manage <pandora_server.conf path> <option> <option parameters> [
optional parameters ]
```

With backslash `\` as a line connector:

```
pandora_manage \  
  <pandora_server.conf path> \  
  <option> \  
  <option parameters> \  
  [ optional parameters ]
```

Syntax

- All parameters, mandatory or optional, can be delimited with single quotes. This is especially useful when a parameter has one or more spaces.
- If you need to omit a parameter you can use two single quotes together. In case the parameter is strictly necessary a message like `[ERROR] Error:` will be displayed followed by a description of the parameter. This is also useful to indicate empty optional parameters.
- Some parameters are case sensitive, others are case insensitive. For example, **when creating an agent** the name of the PFMS server must be strictly the same, however the name of the primary group to which the agent will belong can be in upper and/or lower case and if there is a match the group will be assigned.
- You can use the alias of an agent, instead of its name itself, by means of the `use_alias` parameter at the end of the command line.

Help

General help:

```
pandora_manage --h | more
```

To obtain help about one specific option just write option without parameters:

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

Commands are grouped into the following categories:

- [Agents](#).
- [Modules](#)
- [Alerts](#)
- [Users](#)
- [Events](#)
- [Policies](#)
- [NetFlow®](#)
- [Tools](#)
- [Graphs](#)
- [Clusters](#)
- [Visual Consoles](#)

Agents

create_agent

Required parameters ([syntax](#)):

- `< agent name >` Returns error if the name exists (see `< alias as name >`).
- `< operating system >` Must match an existing OS (case insensitive) otherwise it will be registered without any OS.
- `< agent group name >`
- `< server name >` Be case sensitive, otherwise the value of the field will be left null *without any warning*. You must verify that the agent has been assigned to a server.

Optional parameters:

- `< IP address or URL >`
- `< description >` If you omit the description Created by `< server name >` will be added (regardless of whether the specified server exists or not).
- `< interval >` In seconds, default value: 300 (see next point).
- `< alias as name >` The default value is 1 and with this value the agent alias will be the same as the agent name. If you want the name to be randomly generated, use 0.

Description:

An agent will be created with the specified name, operating system, group and server. Optionally, it can be given an address (IP address or name), a description and an interval in seconds.

If you are going to use a random agent name using 0 as the last parameter, the agent period must be specified, the recommended value being 300 seconds.

Example (see [call](#) and [syntax](#)):

An agent will be created with a random name and the alias My agent and with the recommended period of 300 seconds.

```
pandora_manage /etc/pandora/pandora_server.conf --create_agent \  
  'My agent' AIX Databases Central-Server \  
  192.168.12.123 'Agent description' '' 0
```

update_agent

Required parameters ([syntax](#)):

- < name_of_agent >
- < field_to_change >
- < new_value >

Optional parameters:

- use_alias

Description: An existing agent's < field_to_modify > will be updated. The available fields are the following:

- agent_name
- address Even if [Unique IP](#) token is enabled, do not repeated IP address warning will be received (ISSUE 12992).
- description
- group_name
- interval
- os_name
- disabled (0 to activate, 1 to inactivate, any other value used will have unpredictable results, ISSUE 13096).
- parent_name If the parent agent is changed and the Cascade protection modules token is activated in the child agent, it must be reconfigured in order to execute the cascade effect (ISSUE 13097). The parent agent must also have at least one module registered.
- cascade_protection (0 to activate, 1 to inactivate, any other value used will have unpredictable results, ISSUE 13098).
- icon_path to change the icon that represents the agent in the GIS maps, the possible values are:

- circle, cross, marker, square_marker, star and triangle.
- update_gis_data (0 to activate it, 1 to inactivate it).
- custom_id

To identify the agent by its alias instead of its name, use `use_alias` right at the end.

Example (see [call](#) and [syntax](#)):

```
pandora_manage /etc/pandora/pandora_server.conf --update_agent \  
'My Agent' agent_name 'Agent 2'
```

When changing an [agent's IP address](#) note that this command allows repeating IP addresses.

```
pandora_manage /etc/pandora/pandora_server.conf --update_agent \  
'My Agent' address 192.168.7.47
```

```
pandora_manage /etc/pandora/pandora_server.conf --update_agent \  
'My Agent' description 'New note' \  
use_alias
```

For the `disabled` option only 0 and 1 should be used, any other value will produce unpredictable search and/or display results in the Web Console.

```
pandora_manage /etc/pandora/pandora_server.conf --update_agent \  
'My Agent' disabled 1 \  
use_alias
```

For the `cascade_protection` option only 0 and 1 should be used, any other value will produce unpredictable search and/or display results in the Web Console.

```
pandora_manage /etc/pandora/pandora_server.conf --update_agent \  
'My Agent' cascade_protection 1 \  
use_alias
```

```
pandora_manage /etc/pandora/pandora_server.conf --update_agent \  
'My Agent' icon_path cross \  
use_alias
```

agent_update_custom_fields

Required parameters ([syntax](#)):

- < id_agent > agent identifier.
- < field_type > field type which can be 0 if it is a text type or 1 if it is a combo box (select) type composed of several options. See section on answers.

- < field_to_change > name of the field to be updated.
- < valor_nuevo > the value of the field to be updated.

Description: updates the **custom fields** of an agent. This command is incompatible with custom fields of type link and type password (ISSUE 13532).

The answers of which they are composed are:

- ERROR:
 - If the agent does not exist.
 - If the field does not exist.
 - In the event that the new_value of a combo does not match the ones you have preset.
 - If you were unable to update the field.
- INFO:
 - Updating the X field with the agent id Y.
 - Correctly updated.

Examples (see **call** and **syntax**):

TEXT TYPE:

```
pandora_manage /etc/pandora/pandora_server.conf \  
--agent_update_custom_fields \  
1 0 'Serial number' 'THX-1138'
```

COMBO TYPE: you must have configured a custom field named Names with at least one item named John.

```
pandora_manage /etc/pandora/pandora_server.conf \  
--agent_update_custom_fields \  
1 1 'Names' 'John'
```

delete_agent

Required parameters (**syntax**):

- < agent_name >

Optional parameter:

- use_alias

Description:

The agent will be deleted by passing its name or alias as a parameter.

Example (see **call** and **syntax**):

```
pandora_manage /etc/pandora/pandora_server.conf \  
  --delete_agent 'pandora.internals' \  
  use_alias
```

disable_group

Required parameter:

- < group_name >

Description:

Disable all agents from an entire group. The **agents of the group** considered as parameter will be disabled with the execution of this option.

If you pass `All` as group, all agents from all groups will be disabled.

Example (see [call](#) and [syntax](#)):

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

enable_group

Required parameter:

- < group_name >

Description:

The agents of the group considered as parameter will be disabled with the execution of this option. If you pass `All` as group all agents from all groups will be enabled.

Example (see [call](#) and [syntax](#)):

```
pandora_manage /etc/pandora/pandora_server.conf --enable_group FIREWALLS
```

create_group

Required parameters:

- < group_name >

Optional parameters:

- < parent_group_name >
- < icon > options available:
 - applications
 - application_osx
 - application_osx_terminal
 - bricks
 - chart_organisation
 - clock
 - computer
 - database
 - database_gear
 - docker
 - drive_network
 - email
 - eye
 - firewall
 - heart
 - house
 - images
 - lightning
 - lock
 - network
 - plugin
 - printer
 - server_database
 - transmit
 - vmware
 - without_group
 - world
- < 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, choose one from list above ISSUE 9408) and description. The parent group by default is All and the default icon is empty string (without icon).

If you need to add a description:

- If you will not use parent group set All as fill parameter.
- If no icon is to be used, place an empty string (two single quotes together) as the filler parameter.

Example (see [call](#) and [syntax](#)):

```
pandora_manage /etc/pandora/pandora_server.conf --create_group \
```

```
'New group name' \  
Web computer 'New description.'
```

delete_group

Required parameter:

- < group_name >

Description:

Delete an agent group. The special group All cannot be deleted.

Every agent belongs to indicated group will be deleted too.

Example (see [call](#) and [syntax](#)):

```
pandora_manage /etc/pandora/pandora_server.conf --delete_group 'Group name'
```

update_group

Required parameters:

- < group_id >

Optional parameters:

- < group_name_to_modify >
- < parent_group_name >
- < icon > options available:
 - applications
 - application_osx
 - application_osx_terminal
 - bricks
 - chart_organisation
 - clock
 - computer
 - database
 - database_gear
 - docker
 - drive_network
 - email
 - eye
 - firewall
 - heart
 - house

- images
- lightning
- lock
- network
- plugin
- printer
- server_database
- transmit
- vmware
- without_group
- world

- < description >

Description: A group is modified by its numerical identifier. Parameters that can be modified:

- Name group.
- Name group parent
- Icon (only name, without extension, choose one from list above).
- Description.

- You can use the “ [get_agent_group_id](#) ” function to find the id of the group of any agent.
- At Web console go to menu Management → Profiles → Manage agent groups and take number id.

You must type exactly same existing parameters for only modify description field.

Example (see [call](#) and [syntax](#)):

```
pandora_manage /etc/pandora/pandora_server.conf --update_group 2 \  
  'Group name' 'Web' 'transmit' 'Description'
```

stop_downtime

Required parameter:

- < downtime_name >

Description:

Stop a [planned downtime](#). If the downtime is finished, a message will be showed. Returning messages:

- [ERROR] Planned_downtime cannot be stopped.
- [INFO] Planned_downtime is already stopped.
- [INFO] Stopping planned downtime.

Example (see [call](#) and [syntax](#)):

```
pandora_manage /etc/pandora/pandora_server.conf --stop_downtime 'Downtime name'
```

get_agent_status

Required parameter:

- < agent_name >

Optional parameter:

To identify the agent by its alias instead of its name, use `use_alias` right at the end.

Description: Get the status of a given agent by its name or alias.

Example (see [call](#) and [syntax](#)):

```
pandora_manage /etc/pandora/pandora_server.conf --get_agent_status 'agent name' use_alias
```

get_agent_group

Required parameter:

- < agent_name >

Optional parameter:

To identify the agent by its alias instead of its name, use `use_alias` right at the end.

Description: Get the group of a given agent by its name or alias. This agent group name will be returned with URL encoding (Percent-encoding).

Example (see [call](#) and [syntax](#)):

```
pandora_manage /etc/pandora/pandora_server.conf --get_agent_group 'agent_name' use_alias
```

get_agent_group_id

Required parameter:

- < agent_name >

Optional parameter:

To identify the agent by its alias instead of its name, use `use_agent_alias` right at the end.

Description: Get the identification group of a given agent by its name or alias.

Example (see [call](#) and [syntax](#)):

```
pandora_manage /etc/pandora/pandora_server.conf --get_agent_group_id  
'pandora.internals'
```

```
pandora_manage /etc/pandora/pandora_server.conf --get_agent_group_id 'pandorafms  
agent' use_alias
```

get_agent_modules

Required parameter:

- < agent_name >

Optional parameter:

To identify the agent by its alias instead of its name, use `use_alias` right at the end.

Description: Get the modules (identifications and names, separated by a comma) of a given agent by its name or alias.

Example (see [call](#) and [syntax](#)):

```
pandora_manage /etc/pandora/pandora_server.conf --get_agent_modules 'agent name'  
use_alias
```

get_agents

Required parameter: None.

Optional parameters: Allow filtering the list of agents in a serialized format where the order matters, use two single quotes to skip parameter(s). They are numbered here for better control and visualization:

1. < group_name >

2. < os_name >
3. < status > Possible values: critical, warning, unknown, normal.
4. < number_max_of_modules > Only one numeric value should be used (ISSUE 13968).
5. < key_string > Allows filtering by agent name (or agent alias if you add at the end use_agent_alias).
6. < policy_name >
7. use_agent_alias to search for an agent by alias instead of name.

Description: Get all agents (without parameters) or agents according to a filter.

Examples (see [call](#) and [syntax](#)):

- Get all agents in the Network group:

```
pandora_manage /etc/pandora/pandora_server.conf --get_agents 'Network' '' '' ''
'' ''
```

- Get all the agents that have Linux as operating system:

```
pandora_manage /etc/pandora/pandora_server.conf --get_agents '' 'Linux' '' '' ''
''
```

- Obtain all agents whose status is normal:

```
pandora_manage /etc/pandora/pandora_server.conf --get_agents '' '' 'normal' ''
'' ''
```

- Obtain all agents with zero modules (no modules at all):

```
pandora_manage /etc/pandora/pandora_server.conf --get_agents '' '' '' '0' '' ''
```

- Get all agents with zero modules in the Servers group:

```
pandora_manage /etc/pandora/pandora_server.conf --get_agents 'Servers' '' '' '0'
'' ''
```

- Get all agents that contain the word test in their alias:

```
pandora_manage /etc/pandora/pandora_server.conf --get_agents '' '' '' '' 'test'
'' use_agent_alias
```

- Get all the agents that are in the monitoring policy called Basic Linux Monitoring:

```
pandora_manage /etc/pandora/pandora_server.conf --get_agents '' '' '' '' ''
'Basic Linux Monitoring'
```

get_agents_id_name_by_alias

Required parameter: < agent_alias >

Optional parameter: strict

Description: Lists the identifiers and aliases of agents matching the specified alias, case insensitive. Using the optional `strict` parameter limits the search to an exact match of all letters.

Example (see [call](#) and [syntax](#)):

```
pandora_manage /etc/pandora/pandora_server.conf --get_agents_id_name_by_alias  
'name' strict
```

delete_conf_file

Required parameter: < agent_name >

Optional parameter: `use_alias`

Description: The configuration file of an agent will be deleted. When accessing the Web Console via the Remote configuration option you will receive an error message:

Error: the conf file of agent is not readable..

All connection settings and agent configuration will also be deleted.

Example (see [call](#) and [syntax](#)):

```
pandora_manage /etc/pandora/pandora_server.conf --delete_conf_file 'agent alias'  
use_alias
```

clean_conf_file

Required parameter: < agent_name >

Parámetro opcional: `use_alias`

Description: The configuration file of an agent will be cleaned (all the contents of the file will be deleted - including connection values and configuration parameters ISSUE 13663).

Example (see [call](#) and [syntax](#)):

```
pandora_manage /etc/pandora/pandora_server.conf --clean_conf_file 'agent alias'  
use_alias
```

get_bad_conf_files

Parameters: None.

Description:

- The misconfigured configuration files will be listed by searching the main *tokens*:
 - `server_ip`
 - `server_path`
 - `temporary`
 - `logfile`
- In case there are corrupt files that cannot be opened:

```
[WARN] Can't open file < path_file >.
```

- In case of not finding the file in the path:

```
[WARN] File not exist < path_file >.
```

- In case the configuration files are correct (according to the *tokens* of the first point):

```
[INFO] No bad files found
```

Example (see [call](#) and [syntax](#)):

```
pandora_manage /etc/pandora/pandora_server.conf --get_bad_conf_files
```

migration_agent_queue

Only for [Command Center](#) (Metaconsole).

Required parameters:

- `< agent_id_to_migrate >`
- `< node_name_source >`
- `< node_name_target >`

Optional parameter:

- `< only_db >` by default `0`, it will migrate database and historical data, if set to `1` it will only migrate database without historical data.

Description: An agent is added to the migration queue to move from node to node.

Example (see [call](#) and [syntax](#)):

```
pandora_manage /etc/pandora/pandora_server.conf --migration_agent_queue 1 node_1  
node_2 0
```

For it to be carried out correctly, the API configuration parameters will need to be filled in `pandora_server.conf` ([configuration file elements](#)): `console_api_url` and

console_api_pass .

migration_agent

Only for [Command Center](#) (Metaconsole).

Required parameter: < agent_id_to_check >

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

Example (see [call](#) and [syntax](#)):

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

For it to be carried out correctly, the API configuration parameters will need to be filled in `pandora_server.conf` ([configuration file elements](#)): `console_api_url` and `console_api_pass` .

create_downtime

Required parameters ([syntax](#)):

- < planned_stop_name >
- < description >
- < date_from >
- < date_to >
- < group_id >
- < Monday (0|1) >
- < Tuesday (0|1) >
- < Wednesday (0|1) >
- < Thursday (0|1) >
- < Friday (0|1) >
- < Saturday (0|1) >
- < Sunday (0|1) >
- < time_from_period HH:MM:SS>
- < time_to_period HH:MM:SS>
- < day_number_from_in_period >
- < day_number_to_date_period >
- < planned_stop_type (quiet|disable_agents|disable_agents_alerts) >
- < execution_type periodically > ('once' option is disabled).
- < type_of_periodicity (weekly|monthly) >
- < user_id >

Description: A weekly or monthly scheduled stop will be created with the submitted data. The date

format must be MM/DD/YYYYYY (Month/Day/Year) for this call to work properly.

Example (see [call](#) and [syntax](#)):

```
pandora_manage /etc/pandora/pandora_server.conf --create_downtime \  
Testing Testing 05/07/2024 06/07/2024 \  
0 1 1 1 1 11 1 17:07:00 18:08:00 1 31 \  
quiet periodically weekly admin
```

For this to be done correctly, the API configuration parameters must be filled in at `pandora_server.conf` ([configuration file elements](#)): `console_api_pass`, `console_pass`, `console_user` and `console_api_url`.

add_item_downtime

Required parameters ([syntax](#)):

- < planned_stop_id >
- < agent_id1,agent_id2,agent_id3,...,agent_idN >
- < module_name1,module_name2,module_name3,...,module_nameN >

Description:

Records will be added to a scheduled stop with the submitted data. Write the list of agent identifiers and the list of module names without leaving spaces between the commas that separate the components of both lists. If a module name contains spaces, enclose the module name in two single quotes. If two or more agents are passed, it is assumed by default that you want to add common modules, otherwise it will display a warning message.

Note that the command allows the addition of repeated items.

Example (see [call](#) and [syntax](#)):

```
pandora_manage /etc/pandora/pandora_server.conf --add_item_downtime \  
1 1 Memory_Used, 'CPU Load'
```

For it to be carried out correctly, the API configuration parameters will need to be filled in `pandora_server.conf` ([configuration file elements](#)): `console_api_pass`, `console_pass`, `console_user` and `console_api_url`.

get_all_planned_downtimes

Required parameters:

- < name_to_search >

Optional parameters (Issue 9414):

- < id_group >
- < type_downtime > quiet, disable_agents, disable_agents_alerts.
- < type_execution > once, periodically.
- < type_periodicity > weekly, monthly.

Description:

All **planned shutdowns** that match the submitted data will be listed.

Example (see **call** and **syntax**):

To obtain the complete list of planned stops:

```
pandora_manage /etc/pandora/pandora_server.conf --get_all_planned_downtimes ''
```

To list a very specific planned stop:

```
pandora_manage /etc/pandora/pandora_server.conf --get_all_planned_downtimes  
"Stop 1" "8" "disable_agents" "periodically" "monthly"
```

get_planned_downtimes_items

Required parameters:

- < name_to_search >

Optional parameters (Issue 14343):

- < id_group >
- < type_downtime > quiet, disable_agents, disable_agents_alerts.
- < type_execution > once, periodically.
- < type_periodicity > weekly, monthly.

Description:

All items matching the name of the requested **planned downtimes** will be listed.

Example (see **call** and **syntax**):

To obtain all the items of all the planned stops:

```
pandora_manage /etc/pandora/pandora_server.conf --get_planned_downtimes_items ''
```

To obtain the items of the planned stops containing the word new in upper and/or lower case:

```
pandora_manage /etc/pandora/pandora_server.conf --get_planned_downtimes_items  
'new'
```

To obtain the items of the weekly planned stops:

```
pandora_manage /etc/pandora/pandora_server.conf --get_planned_downtimes_items ''  
'' '' '' 'weekly'
```

set_planned_downtimes_deleted

Required parameters ([syntax](#)):

- < planned_stop_name >

Description:

Deletes a planned stop by its corresponding name. It may respond with one of these messages:

- This planned downtime is deleted.
- Problems with this planned downtime.
- The scheduled downtime is still being executed.

Example (see [call](#) and [syntax](#)):

```
pandora_manage /etc/pandora/pandora_server.conf --set_planned_downtimes_deleted  
\ 'Name Scheduled Downtime'
```

set_disabled_and_standby

Only for [Command Center](#) (Metaconsole) and nodes.

Required parameters:

- < agent_id >
- *For Command Center (Metaconsole) the node ID, for nodes put any value, it is indifferent.*

Optional parameters:

- < value > By default 1 to enable the agent, 0 to disable.

Description: Disables an agent and also, if it has remote configuration, puts it in standby mode.

Example (see [call](#) and [syntax](#)):

To enable an agent from the node:

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

To disable an agent from node:

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

To enable an agent from Command Center (Metaconsole):

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

To disable an agent from Command Center (Metaconsole):

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

For it to be carried out correctly, the API configuration parameters will need to be filled in `pandora_server.conf` ([configuration file elements](#)): `console_api_url` and `console_api_pass`.

reset_agent_counts

Required parameters ([syntax](#)):

< agent_numeric_identifier >

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 (see [call](#) and [syntax](#)):

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

For it to be carried out correctly, the API configuration parameters will need to be filled in `pandora_server.conf` ([configuration file elements](#)): `console_api_pass`, `console_pass`, `console_user` and `console_api_url`.

insert_gis_data

Required parameters ([syntax](#)):

- < agent_numeric_identifier >
- < latitude >
- < longitude >
- < altitude >

Optional parameters:

- None.

Description: Updates the **GIS data** of an agent. This feature must be enabled in the PFMS general configuration.

Example (see [call](#) and [syntax](#)):

```
pandora_manage /etc/pandora/pandora_server.conf --insert_gis_data 1 2 3 4
```

For it to be carried out correctly, the API configuration parameters will need to be filled in `pandora_server.conf` ([configuration file elements](#)): `console_api_pass` , `console_pass` , `console_user` and `console_api_url` .

get_gis_agent

Required parameters ([syntax](#)):

- < agent_numeric_identifier >

Optional parameters:

- None.

Description: Gets an **agent's GIS data**. This feature must be enabled in the general PFMS configuration.

Examples (see [call](#) and [syntax](#)):

```
pandora_manage /etc/pandora/pandora_server.conf --get_gis_agent 1
```

For it to be carried out correctly, the API configuration parameters will need to be filled in `pandora_server.conf` ([configuration file elements](#)): `console_api_pass` , `console_pass` , `console_user` and `console_api_url` .

agent_set_os

Required parameters ([syntax](#)):

- < agent_id_numeric >
- < operating_system_id_numeric >
- < operating_system_version >

Optional parameters:

- None.

Description: Add or change **operating system (and obsolescence)** to an agent.

Examples (see **call** and **syntax**):

```
pandora_manage /etc/pandora/pandora_server.conf --agent_set_os 7 1 'Ubuntu
16.04'
```

locate_agent

Required parameter: < agent_name >

Optional parameter: use_alias to search by agent alias.

Description: Searches for an agent in the nodes of a Command Center (Metaconsole) and returns the node identifier.

Example (see **call** and **syntax**):

```
pandora_manage /etc/pandora/pandora_server.conf --locate_agent NAS use_alias
```

Modules

create_data_module

Required parameters:

1. < module_name >
2. < module_type >
3. < agent_name >
4. < description >
5. < module_group >
6. < min >
7. < max >

Although fields 4 to 7 are required, these parameters can be omitted by using quotation marks "". In the case of setting the minimum and maximum values accepted by the module, any value outside this range will be discarded.

Optional parameters:

1. < post_process >
2. < interval >
3. < warning_min >
4. < warning_max >
5. < critical_min >
6. < critical_max >
7. < history_data >
8. < definition_file >
9. < warning_str >
10. < critical_str >
11. < unknown_events >
12. < ff_threshold >
13. < each_ff >
14. < ff_threshold_normal >
15. < ff_threshold_warning >
16. < ff_threshold_critical >
17. < ff_timeout >
18. < warning_inverse >
19. < critical_inverse >
20. < critical_instructions >
21. < warning_instructions >
22. < unknown_instructions >
23. use_agent_alias: For the agent search use the agent alias instead of the agent name.
24. ignore_unknown: Discards the unknown state calculation, so that the module will never change to that state and will always keep the last known state.
25. < number_of_intervals_in_warning > Allows *state scaling* by specifying the maximum number of consecutive intervals in which the module remains in warning state. If this value is exceeded, the module will *escalate* to critical status.

Description: A data type module will be created in an agent with the module name, module type and agent name (see option use_agent_alias) where it will be created. Optionally it will be possible to give it a description, the module group, et cetera.

The default values are 0 for minimum and maximum, historical_data and post_process; 300 for interval.

Example (see [call](#) and [syntax](#)):

Basic example:

```
pandora_manage /etc/pandora/pandora_server.conf --create_data_module "My new
module" "generic_data" "pandora.internals" "My description" "Miscellaneous" "0"
"100"
```

Simple example for state scaling:

```
pandora_manage /etc/pandora/pandora_server.conf --create_data_module
data_module_test generic_data
```



```
3f9c41953a072afa229aa0a7fe3a6203a1ecf86b40b8a13d8a7d9916f0210bb1 "" "" "" "" ""
"" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" 2001
```

Complex example, the module definition file will contain something like this:

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

```
pandora_manage /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/definition_file 'warning text' 'critical text'
```

If a different name or type is entered between the parameters and the file definition, the file definition will take precedence.

create_web_module

Required parameters:

1. < module_name >
2. < module_type > web_data, web_proc, web_content_data, web_content_string.
3. < agent_name >
4. < description >

Optional parameters:

1. < module_group >
2. < min >
3. < max >
4. < post_process >
5. < interval >
6. < warning_min >
7. < warning_max >
8. < critical_min >
9. < critical_max >
10. < history_data >
11. < retries >
12. < requests >
13. < agent_browser_id >
14. < auth_server >
15. < auth_realm >
16. < definition_file >
17. < proxy_url >
18. < proxy_auth_login >
19. < proxy_auth_password >
20. < warning_str >

21. < critical_str >
22. < enable_unknown_events >
23. < ff_threshold >
24. < each_ff >
25. < ff_threshold_normal >
26. < ff_threshold_warning >
27. < ff_threshold_critical >
28. < ff_timeout >
29. < warning_inverse >
30. < critical_inverse >
31. < critical_instructions >
32. < warning_instructions >
33. < unknown_instructions >
34. < use_agent_alias >: For the agent search use the agent alias instead of the agent name.
35. < ignore_unknown >: Discards the unknown state calculation, so that the module will never change to that state and will always keep the last known state.
36. < number_of_intervals_in_warning > Allows state scaling by specifying the maximum number of consecutive intervals in which the module remains in warning state. If this value is exceeded, the module will escalate to critical status.

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 (see parameter `use_agent_alias`). 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. In the case of setting the minimum and maximum values accepted by the module, any value outside this range will be discarded.

Example (see [call](#) and [syntax](#)):

Basic example:

```
pandora_manage /etc/pandora/pandora_server.conf --create_web_module "My new
web_content_string" "web_content_string" "pandora.internals" "My description"
"Miscellaneous" "0" "100"
```

Simple example for state scaling:

```
pandora_manage /etc/pandora/pandora_server.conf --create_web_module
web_module_test web_data
3f9c41953a072afa229aa0a7fe3a6203a1ecf86b40b8a13d8a7d9916f0210bb1 "" "" "" "" ""
"" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" ""
"" "" "" "" 2001
```

As a complex example, the module definition file will contain something like this:

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

```
pandora_manage /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/file_definition
'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

Required parameters:

1. < module_name >
2. < module_type > accepts the following values: remote_icmp_proc, remote_icmp, remote_tcp, remote_tcp_proc, remote_tcp_string, remote_tcp_inc.
3. < agent_name >
4. < module_address >
5. < module_port > only for TCP: numerical value between 1 and 65535.
6. < description >
7. < module_group >
8. < min >
9. < max >
10. < post_process >
11. < interval >
12. < warning_min >
13. < warning_max >

Optional parameters:

1. < critical_min >
2. < critical_max >
3. < history_data >
4. < ff_threshold >
5. < warning_str >
6. < critical_str >
7. < enable_unknown_events >
8. < each_ff >
9. < ff_threshold_normal >
10. < ff_treshold_warning >
11. < ff_threshold_critical >
12. < timeout >
13. < retries >
14. < critical_instructions >
15. < warning_instructions >
16. < unknown_instructions >
17. < warning_inverse >
18. < critical_inverse >
19. use_agent_alias to search for the agent by its alias instead of its name.
20. ignore_unknown Discards the unknown state calculation, so that the module will never change to that state and will always keep the last known state.
21. < number_of_intervals_in_warning > Allows state scaling by specifying the maximum number of consecutive intervals in which the module remains in warning state. If this value is exceeded, the

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

Description: A network component will be created. Optionally it can be given a port, a description, minimum and maximum values, a post processing value, an interval in seconds, minimum and maximum warning values, minimum and maximum critical values, and a historical data value.

Example (see [call](#) and [syntax](#)):

```
pandora_manage /etc/pandora/pandora_server.conf --create_network_component  
"example_name" 2 7
```

For it to be carried out correctly, the API configuration parameters will need to be filled in `pandora_server.conf` ([configuration file elements](#)): `console_api_url` and `console_api_pass`.

create_snmp_module

Required parameters:

1. < module_name >
2. < module_type > the following options are valid: `remote_snmp`, `remote_snmp_inc`, `remote_snmp_proc`.
3. < agent_name >
4. < module_address >
5. < module_port > numerical value between 1 and 65535.
6. < version >
7. < community >
8. < oid >

9. < description >
10. < module_group >
11. < min >
12. < max >
13. < post_process >
14. < interval >

Optional parameters:

1. < warning_min >
2. < warning_max >
3. < critical_min >
4. < critical_max >
5. < history_data >
6. < snmp3_priv_method >
7. < snmp3_priv_pass >
8. < snmp3_sec_level >
9. < snmp3_auth_method >
10. < snmp3_auth_user >
11. < snmp3_auth_pass >
12. < ff_threshold >
13. < warning_str >
14. < critical_str >
15. < unknown_events >
16. < each_ff >
17. < ff_threshold_normal >
18. < ff_threshold_warning >
19. < ff_threshold_critical >
20. < timeout >
21. < retries >
22. use_alias to search for the agent by its alias instead of its name.
23. ignore_unknown Discards the unknown state calculation, so that the module will never change to that state and will always keep the last known state.
24. < critical_instructions >
25. < warning_instructions >
26. < unknown_instructions >
27. < warning_inverse >
28. < critical_inverse >
29. < number_of_intervals_in_warning > Allows state scaling by specifying the maximum number of consecutive intervals in which the module remains in warning state. If this value is exceeded, the module will escalate to critical status.

Description: An SNMP type module will be created in an agent with the module name (not null), module type, agent name where it will be created, module address, associated port, SNMP version, community name, OID, description, module group, minimum and maximum values, post processing value, check interval (in seconds), and other optional values.

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

Example (see [call](#) and [syntax](#)):

get_module_group

Optional parameters:

- < module_group_name >

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

Response: CSV format with headers.

- < id_module_group >
- < group_name >

Example (see [call](#) and [syntax](#)):

```
pandora_manage /etc/pandora/pandora_server.conf --get_module_group "general"
```

create_module_group

Required parameter:

- < group_name >

Description: A module group with the specified name will be created.

Example (see [call](#) and [syntax](#)):

```
pandora_manage /etc/pandora/pandora_server.conf --create_module_group "My module group"
```

module_group_synch

Required parameter: < server_name_1|server_name_2| ... |server_name_n >

Optional parameter: < return_type > JSON and CSV, CSV format by default.

Description: The Command Center module groups will be synchronized with the nodes that have been specified in the first parameter, separated with |.

Example (see [call](#) and [syntax](#)):

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

For it to be carried out correctly, the API configuration parameters will need to be filled in `pandora_server.conf` ([configuration file elements](#)): `console_api_url` and `console_api_pass`.

create_synthetic

Parameters:

- `< module_name >`
- `< synthetic_type >` arithmetic or average.
- `< agent_name >`
- `< opts >`
- `use_alias`

Description: A synthetic module will be created in the agent and with the indicated module name. The module type can be arithmetic or average. The operators can be: `+ - * / x`.

Example (see [call](#) and [syntax](#)):

```
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 >
```

Possible parameters to use in `< opts >`:

- `< opts > = < agent_source1 >,< operator >,< source_module1 > < agent_source2 >,< operator >,< source_module2 >`
- `< opts > = < agent_source1 >,< operator >,< source_module1 > < operator >,< fixed_value >`

delete_module

Required parameters:

- `< module_name >`
- `< agent_name >`

Optional parameter:

- `use_alias`

Description: A module will be removed from an agent by passing the name of both as a parameter (o con el alias del agente usando `use_alias`). In case this module is in a local agent, it will also be deleted from the configuration file. From version 771 onwards, the *descendants* of the deleted module are deleted recursively.

Examples (see [call](#) and [syntax](#)):

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

data_module

Required parameters:

- < pfms_server_name >
- < agent_name >
- < module_name >
- < module_type >
- < new_data >

Optional parameters:

- < date_time > format YYYY-MM-DD HH:mm.
- use_alias if this option is used, a valid date and time must be specified.

Description: Inserts a value to a module.

Responses:

- [ERROR] No module found with this type.: The module type differs from the one registered for the requested module.
- [INFO] Inserting data to module: Successful response.

Example (see [call](#) and [syntax](#)):

```
pandora_manage /etc/pandora/pandora_server.conf --data_module "pandorafms" "pandorafms agent" "CPU Load" generic_data 77 "2024-07-19 13:38" use_alias
```

get_module_data

Required parameters:

- < agent_name >
- < module_name >
- < interval > value in seconds.

Optional parameters:

- < separator > in quotation marks, if necessary; by default "|".
- use_alias if this option is used, a separator must be specified.

Description: The data of a module will be returned for the last X seconds (requested interval) using

the separator between *timestamp* and *value*.

Example (see [call](#) and [syntax](#)):

```
pandora_manage /etc/pandora/pandora_server.conf --get_module_data "pandorafms
agent" "CPU Load" 7200 ", " use_alias
```

get_module_id

Required parameters:

- < agent_numeric_identifier > if the identifier is not registered it will return an error message *indicating that the requested module does not exist*.
- < module_name >

Description: The numerical identifier of a specific module of an agent will be returned.

Example (see [call](#) and [syntax](#)):

```
pandora_manage /etc/pandora/pandora_server.conf --get_module_id 2 "CPU Load" &&
echo ""
```

set_module_custom_id

Required parameters:

- < module_numeric_identifier >
- < custom_identifier_value >

Description: Inserts the value of the Custom ID field of a specific module. If left null "" its content will be removed.

Example (see [call](#) and [syntax](#)):

```
pandora_manage /etc/pandora/pandora_server.conf --set_module_custom_id 21 "My id
value" && echo ""
```

get_module_custom_id

Required parameter:

- < module_id > (see [get_module_id](#)).

Description: The value of the Custom ID field (including entities and without line return) of a specific module of an agent will be returned.

Example (see [call](#) and [syntax](#)):

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

delete_data

Description: All the data associated to the specified object (module, agent or agent group) will be deleted from the data history.

Parameter for deleting module data: -m < module_name > < agent_name >

Parameter for deleting agent data: -a < agent_name > (optional use_alias)

Parameter for deleting group data: -g < group_agent_name >

Example (see [call](#) and [syntax](#)):

```
pandora_manage /etc/pandora/pandora_server.conf --delete_data -a "agent_alias"  
use_alias
```

update_module

Required parameters:

- < module_name >
- < agent_name >
- < field_to_change >
- < new_value >

Optional parameter:

- use_alias.

Description: A given field of an existing module will be updated. The module type will be detected to allow updating the specific fields of each type.

Software Agents with remote configuration enabled may overwrite the changes made.

The possible fields to be modified are:

Common to any module:

- module_name
- description
- module_group

- min
- max
- post_process
- history_data (should only be used 1 or 0).
- interval
- warning_min
- warning_max (must be greater than warning_min).
- critical_min
- critical_max (must be greater than critical_min).
- warning_str (not to be confused with warning_instructions).
- critical_str (not to be confused with critical_instructions).
- ff_threshold
- each_ff (integer numeric value used in plugins).
- 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

Example (see [call](#) and [syntax](#)):

```
pandora_manage /etc/pandora/pandora_server.conf --update_module "CPU load"  
"pandorafms agent" "description" "new description"
```

With the `agent_name` parameter you can move a module from one agent to another agent identified only by its name:

```
pandora_manage /etc/pandora/pandora_server.conf --update_module  
"last_events_24h" "pandora.internals" "agent_name"  
4be5603649674274b9b7ba841118a876c2365f70667eb0da37badd498b6a4aa8
```

add_tag_to_module

Required parameters:

- < nombre_de_agente >
- < nombre_de_módulo >
- < nombre_de_etiqueta >

Description: Adds a tag (Management → Profiles → Module tags menu) to specified module.

Example (see [call](#) and [syntax](#)):

```
pandora_manage /etc/pandora/pandora_server.conf --add_tag_to_module  
"pandora.internals" "console_log_size" "Dmz"
```

For it to be done correctly, the API configuration parameters in `pandora_server.conf` ([elements of the configuration file](#)) should be filled in: `console_api_url` and `console_api_pass`.

get_agents_module_current_data

Required parameter:

- < module_name >

Description: Gets the agent identifier, its name and the current data (separated by commas) of all modules with the exact requested name (case insensitive).

Example (see [call](#) and [syntax](#)):

```
pandora_manage /etc/pandora/pandora_server.conf --get_agents_module_current_data  
"cpu load"
```

create_network_module_from_component

Required parameters:

- < agent_name >
- < remote_component_name >

Optional parameter:

- use_alias

Description: Creates a module in a given agent from a **remote component**. The agent name or its alias must be supplied with the use_alias option.

Example (see [call](#) and [syntax](#)):

```
pandora_manage /etc/pandora/pandora_server.conf --  
create_network_module_from_component "pandorafms agent" "IIS PutRequestsPersec"  
use_alias
```

create_data_module_from_local_component

Required parameters:

- < agent_name >
- < local_component_name >

Optional parameters:

- use_alias

Description: Creates a module in a given agent from a **local component**. The agent name or its alias must be supplied with the use_alias option.

Example (see [call](#) and [syntax](#)):

```
pandora_manage /etc/pandora/pandora_server.conf --  
create_data_module_from_local_component "pandorafms agent" "Event 1000"  
use_alias
```

create_local_component

Parameters:

- < component_name >
- < data >
- < 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: To create a new local component.

Example (see [call](#) and [syntax](#)):

```
pandora_manage /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: These module templates are a grouping containing network testing modules. These templates can be applied directly to agents, avoiding having to add modules one by one.

Example (see [call](#) and [syntax](#)):

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

Alerts

create_template_module

Required parameters:

- < template_name >
- < module_name >
- < agent_name >

Optional parameters:

- use_alias

Description: By means of an [template](#) an alert will be created to an agent's module.

Example (see [call](#) and [syntax](#)):

```
pandora_manage /etc/pandora/pandora_server.conf --create_template_module "Manual alert" "CPU Load" "pandorafms agent" use_alias
```

delete_template_module

Required parameters:

- < template_name >
- < module_name >
- < agent_name >

Optional parameters:

- use_alias

Description: An alert to an agent module that has been created by means of an [alert template](#) will be removed. It will return response only if the three mandatory parameters exist and the operation is successful, otherwise no message will be displayed.

Example (see [call](#) and [syntax](#)):

```
pandora_manage /etc/pandora/pandora_server.conf --delete_template_module "Manual alert" "CPU Load" "pandorafms agent" use_alias
```

If the previous successful execution will display:

```
[INFO] Delete template 'Manual alert' from module 'CPU Load' from agent 'pandorafms agent'
```

create_template_action

Required parameters:

- < action_name >
- < template_name >
- < module_name >
- < agent_name >

Optional parameters:

- < minimum_number_of_alerts >
- < maximum_number_of_alerts >
- use_alias

Description: An action will be added to an alert by passing as parameters the name of the action and the name of the template, module and agent that form the alert. The **scaling values** minimum number of alerts and maximum number of alerts (both by default 0) can also be passed optionally.

Example (see [call](#) and [syntax](#)):

```
pandora_manage /etc/pandora/pandora_server.conf --create_template_action "Mail to Admin" "Manual alert" "CPU Load" "pandorafms agent" 3 4 use_alias
```

delete_template_action

Required parameters:

- < action_name >
- < template_name >
- < module_name >
- < agent_name >

Optional parameters:

- use_alias

Description: An alert action will be removed from an agent module that uses an **template**.

Example (see [call](#) and [syntax](#)):

```
pandora_manage /etc/pandora/pandora_server.conf --delete_template_action "Mail
```

```
to Admin" "Manual alert" "CPU Load" "pandorafms agent" use_alias
```

disable_alerts

Required parameters: None.

Optional parameters: None.

Description: Disables all alerts. If you already had alerts disabled before executing this command and then execute its counterpart [enable_alerts](#), each and every alert will be active again.

Example (see [call](#) and [syntax](#)):

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

enable_alerts

Required parameters: None.

Optional parameters: None.

Description: Enables all alerts. If you already had alerts enabled before executing this command and then execute its counterpart [disable_alerts](#), each and every alert will be inactive again.

Example (see [call](#) and [syntax](#)):

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

create_alert_template

Parameters:

- < template_name >
- < condition_type_serialized >
- < time_from >
- < time_to >
- < description >
- < group_name >
- < field_1 >
- < field_2 >
- < field_3 >
- < priority >
- < default_action >
- < days >
- < time_threshold >
- < min_alerts >

- < max_alerts >
- < alert_recovery >
- < field_2_recovery >
- < field_3_recovery >
- < condition_type_separator >

Description: An alert template will be created.

Field < condition_type_serialized >: These are the serial template type options with the default ; separator. You can change the separator with the < condition_type_separator > parameter to avoid conflicts in some options if there is a possibility that they contain the default character.

In the following examples the default separator ; is used and the matches_value field is a binary value to set whether the alert will be triggered when the value matches the conditions or not.

Regular expression:

- Syntax: <type>;<matches_value>;<value>
- Example: regex;1;stopped|error (Alert when value matches regular expression 'stopped|error')

Maximum and minimum:

- Syntax: <type>;<matches_value>;<min_value>;<max_value>
- Example: max_min;0;30;50 (Alert when the value is out of range 30-50)

Maximum:

- Syntax: <type>;<max_value>
- Example: max;70 (Alert when the value is greater than 70)

Minimum:

- Syntax: <type>;<min_value> :
- Example: min;30 (Alert when the value is less than 30)

Equal to:

- Syntax: <type>;<value>
- Example: equal;0 (Alert when the value is equal to 0)

Different from:

- Syntax: <type>;<value>
- Example: not_equal;100 (Alert when the value is different from 100)

Warning status:

- Syntax: <type>
- Example: warning (Alert when status changes to warning)

Critical condition:

- Syntax: <type>
- Example: critical (Alert when status changes to critical)

Status unknown:

- Syntax: <type>
- Example: unknown (Alert when status changes to unknown)

Status other than normal:

- Syntax: <type>
- Example: not_normal (Alert when the status is different from normal - warning, critical, et cetera -)

Change status:

- Syntax: <type>;<matches_value>
- Example: on_change;1 (Alert when the value changes)

Always:

- Syntax: <type>
- Example: always (Always alert)

Field <days>:

- There are seven binary characters that specify the days of the week when the alert will be activated.
e.g.: 0000011 to activate the alert only on Saturdays and Sundays.

Example (see [call](#) and [syntax](#)):

```
pandora_manage /etc/pandora/pandora_server.conf --create_alert_template \  
"Template name" \  
"max_min@1@3@5" \  
"09:00 18:00" \  
"It sends an email when the value is in the interval 3-5, between 9 AM and 6  
PM, and only on Mondays. The separator is forced to @" \  
"Unknown" \  
"johndoe@example.com" \  
"subject" \  
"message" \  
"3" \  
"Mail to XXX" \  
"1000000" \  
"38600" \  
"1" \  
"2" \  
"
```

```
"0" \  
"@"
```

delete_alert_template

Required parameter: < template_name >

Optional parameters: None.

Description: An [alert template](#) will be deleted if it exists.

Example (see [call](#) and [syntax](#)):

```
pandora_manage /etc/pandora/pandora_server.conf --delete_alert_template  
"Critical condition" "" "" "" "" "" ""
```

update_alert_template

Parameters:

- < template_name >
- < field_to_update >
- < new_value >

Description: A given field of an existing alert template will be updated.

For < field_to_update > the possible fields are:

- 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)

- field_1
- field_2
- field_3
- recovery_field_2
- recovery_field_3
- priority (0-4)
- default_action
- group_name

Example (see [call](#) and [syntax](#)):

```
pandora_manage /etc/pandora/pandora_server.conf --update_alert_template  
"Template name" "priority" "4"
```

get_alert_actions

Required parameters: None.

Optional parameters:

- < action_name >
- < separator >
- < format >

Description: Returns all alert actions. Optionally you can filter the result by keyword. Optionally in the second parameter you can specify the separator of the results and in the third parameter the format of the result.

Example (see [call](#) and [syntax](#)):

```
pandora_manage /etc/pandora/pandora_server.jim --get_alert_actions '%28' ';' csv
```

For it to be carried out correctly, the API configuration parameters will need to be filled in `pandora_server.conf` ([configuration file elements](#)): `console_api_url` and `console_api_pass`.

get_alert_actions_meta

Only for [Command Center](#) (Metaconsole).

Parameters:

- < server_name >
- < action_name >
- < separator >

- < return_type >

Description: Returns all the alert actions of the Command Center (Metaconsole). Optionally you can filter the result by node name or by action name. 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 (see [call](#) and [syntax](#)):

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

For it to be carried out correctly, the API configuration parameters will need to be filled in `pandora_server.conf` ([configuration file elements](#)): `console_api_url` and `console_api_pass`.

create_alert_command

Parameters:

- < command_name >
- < command >
- < id_group >
- < description >
- < internal >
- < fields_descriptions >
- < fields_values >

Description: A command will be created with the specified name and command data.

Optionally, it can be specified:

- Group name. If no group is provided, it will be assigned to the All group.
- Command Description.
- Internal (1-0).
- Description of fields in the following format:
["description_1", "description_2", "description_3", "description_4", "description_5"].
- Value of the fields in the following format:
["value_1", "value_2", "value_3", "value_4", "value_5"].

Example (see [call](#) and [syntax](#)):

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

get_alert_commands

Required parameters: None.

Optional parameters:

- < command_name >
- < command_it_self >
- < group_name >
- 1 to list commands for internal use in PFMS, 0 to list other commands.

Description: It allows to see all the **alerts commands**. Optionally you can filter the result by keyword in the following fields: command name, code that executes the alert command, group name, description or if it is for internal use.

Example (see **call** and **syntax**):

```
pandora_manage /etc/pandora/pandora_server.conf --get_alert_commands "log"
"echo" "" "0"
```

It is not possible to filter by All agent group since it is an **system group** that does not exist in the database.

validate_alert

Required parameters:

- < template_alert_name >
- < id_agent >
- < id_module >

Optional parameters:

- use_alias if this option is used, the agent's alias must be entered instead of its identifier.

Description: Validates an alert given an alert template name, an agent ID and a module ID.

Example (see **call** and **syntax**):

Assuming that an alert is set in the agent that is installed by default with PFMS.

```
pandora_manage /etc/pandora/pandora_server.conf --validate_alert "Critical
condition" pandora.internals 99 use_alias
```

validate_all_alerts

Parameters: None.

Description: All alerts will be validated.

Example (see [call](#) and [syntax](#)):

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

create_special_day

Parameters:

- < special_day >
- < same_day >
- < description >
- < group_name >

Description: Create a special day.

Example (see [call](#) and [syntax](#)):

```
pandora_manage /etc/pandora/pandora_server.conf --create_special_day  
"2024-05-03" "sunday" "description" "All"
```

delete_special_day

Parameters: < special_day >

Description: Delete specified special day.

Example (see [call](#) and [syntax](#)):

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

set_event_storm_protection

Required parameters:

- < valor > 1 to activate, 0 to deactivate.

Optional parameters: None.

Description: Enables or disables the token **Event storm protection**. If enabled, no events or alerts will be generated while still receiving data (agent checks). Only 1 or 0 should be used, any other value may produce unexpected results.

Example (see [call](#) and [syntax](#)):

```
pandora_manage /etc/pandora/pandora_server.conf --set_event_storm_protection 1
```

update_special_day

Parameters:

- < special_day >
- < field_to_change >
- < new_value >

Description: Modify a specific field of a special day. The fields that can be updated are: same_day, description y group_name.

Example (see [call](#) and [syntax](#)):

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

Users

create_user

Required parameters:

- < id_user >
- < password >
- < is_admin >

Optional parameters:

- < comment >

Description: A user will be created with the user identifier (which will be also its user name) and the received password. In addition, a binary value (< is_admin >; 0 false or 1 true) will be received specifying if the user will be **superadmin** or not. Optionally, a comment about the created user can be added.

Example (see [call](#) and [syntax](#)):

```
pandora_manage /etc/pandora/pandora_server.conf --create_user "user7500" "user7500" "0" "This user has the password 'user 7500'"
```

This function also works in the Command Center (Metaconsole).

delete_user

Required parameter:

- < id_user >

Description: An attempt will be made to delete a user by passing their name as a parameter. You can verify if a user exists or not using the command [update_user](#).

Example (see [call](#) and [syntax](#)):

```
pandora_manage /etc/pandora/pandora_server.conf --delete_user user02
```

This function also works in the Command Center (Metaconsole).

update_user

Required parameters:

- < id_user >
- < field_name_to_update >
- < new_value >

The fields available for < field_name_to_update > are:

1. email
2. phone
3. is_admin: ¿Is [superadmin](#)? (0 false,1 true).
4. comments
5. fullname
6. password
7. language, posible values:
 - default, takes the language selected in the [general configuration](#).
 - ca, Catalan.
 - en_GB, English UK.
 - es, Spanish.
 - fr, French.
 - ru, Russian.
 - ja, Japanese.
 - zh_CN, Chinese (simplified).

Description: An existing user is updated with the given field.

Example (see [call](#) and [syntax](#)):

```
pandora_manage /etc/pandora/pandora_server.conf --update_user "user 7500"  
password "XYZ"
```

This function also works in the Command Center (Metaconsole).

enable_user

Required parameter:

- < user_id >

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

Example (see [call](#) and [syntax](#)):

```
pandora_manage /etc/pandora/pandora_server.conf --enable_user "user7500"
```

This function also works in Command Center (Metaconsole).

disable_user

Required parameter:

- < user_id >

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

Example (see [call](#) and [syntax](#)):

```
pandora_manage /etc/pandora/pandora_server.conf --disable_user "user7500"
```

This function also works in Command Center (Metaconsole).

create_profile

Required parameters:

- < profile_name >
- < rights_list > 1 to grant permission to the new [profile](#) or 0 to deny it, in this order:

1. AR View agents: agent_view.
2. AW Edit agents: agent_edit.
3. AD Disable agents: agent_disable.
4. LW Edit alerts: alert_edit.
5. LM Manage alerts: alert_management.
6. UM Manage users: user_management.
7. DM Manage database: db_management.
8. ER View events: event_view.
9. EW Edit events: event_edit.
10. EM Manage events: event_management.
11. RR View reports: report_view.
12. RW Edit reports: report_edit.
13. RM Manage reports: report_management.
14. MR View network maps: map_view.
15. MW Edit network maps: map_edit.
16. MM Manage network maps: map_management.
17. VR View visual console: vconsole_view.
18. VW Edit visual console: vconsole_edit.
19. VM Manage visual console: vconsole_management.
20. PM Pandora FMS management: pandora_management.
21. 0.
22. 0.
23. 0.

The following permissions must be added by the Web Console:

1. View Network Config Management (NR View NCM data).
2. Operate Network Config Management (NW Operate NCM -must include NR-).
3. Manage Network Config Management (NM Manage NCM -must include NW and NR).
4. Groups Write (GW Manage groups).

Description: A new profile will be created by passing the profile name and permissions as parameters.

Example (see [call](#) and [syntax](#)):

To create a profile with only one permission PM:

```
pandora_manage /etc/pandora/pandora_server.conf \
--create_profile "New Profile" \
0 0 0 0 0 0 0 0 0 0 \
0 0 0 0 0 0 0 0 1 1 1 1
```

This function also works in Command Center (Metaconsole).

update_profile

Required parameters:

- < profile_name >
- < rights_list > 1 to grant permission to the new **profile** or 0 to deny it, in this order:
 1. AR View agents: agent_view.
 2. AW Edit agents: agent_edit.
 3. AD Disable agents: agent_disable.
 4. LW Edit alerts: alert_edit.
 5. LM Manage alerts: alert_management.
 6. UM Manage users: user_management.
 7. DM Manage database: db_management.
 8. ER View events: event_view.
 9. EW Edit events: event_edit.
 10. EM Manage events: event_management.
 11. RR View reports: report_view.
 12. RW Edit reports: report_edit.
 13. RM Manage reports: report_management.
 14. MR View network maps: map_view.
 15. MW Edit network maps: map_edit.
 16. MM Manage network maps: map_management.
 17. VR View visual console: vconsole_view.
 18. VW Edit visual console: vconsole_edit.
 19. VM Manage visual console: vconsole_management.
 20. PM Pandora FMS management: pandora_management.
 21. 0.
 22. 0.
 23. 0.

The following permissions must be added by the Web Console:

1. View Network Config Management (NR View NCM data).
2. Operate Network Config Management (NW Operate NCM -must include NR-).
3. Manage Network Config Management (NM Manage NCM -must include NW and NR).
4. Groups Write (GW Manage groups).

Description: A existent profile will be updated by passing the profile name and permissions as parameters.

Example (see **call** and **syntax**):

To modificate a profile with only one permission PM:

```
pandora_manage /etc/pandora/pandora_server.conf \
--create_profile "New Profile" \
0 0 0 0 0 0 0 0 0 0 \
0 0 0 0 0 0 0 0 0 1 1 1 1
```

This function also works in Command Center (Metaconsole).

add_profile

Required parameters:

- < id_user >
- < profile_name >
- < group_name >

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.

- This command allows you to specify a null or non-existent user.
- This command allows repeating profile/group for a user without issuing a warning message.
- You must verify through the user listing in the PFMS Web Console if the profile was successfully added.

Related command: [add_profile_to_user](#).

Example (see [call](#) and [syntax](#)):

```
pandora_manage /etc/pandora/pandora_server.conf --add_profile "User 07" "Group coordinator" "All"
```

This function also works in Command Center (Metaconsole).

add_tag_to_user_profile

Required parameters:

- < id_user >
- < tag_name >
- < group_name >
- < profile_name >

Description: To the [profile](#) and group assigned to a user add [a tag](#).

Example (see [call](#) and [syntax](#)):

```
pandora_manage /etc/pandora/pandora_server.conf --add_tag_to_user_profile 'User 7500' 'dmz' 'Servers' 'Chief Operator'
```

For it to be carried out correctly, the API configuration parameters will need to be filled in `pandora_server.conf` ([configuration file elements](#)): `console_api_url` and

console_api_pass .

delete_profile

Required parameters:

- < id_user >
- < profile_name >
- < group_name >

Description: An user profile will be deleted giving it as parameter the user identification, profile name and group name.

- If the user identifier includes spaces or any other extended characters you must *escape* those characters. A space is represented as ` ` and `&` as `&`, and so on.
- You must verify through the user listing in the PFMS Web Console if the profile was successfully deleted.

Example (see [call](#) and [syntax](#)):

```
pandora_manage /etc/pandora/pandora_server.conf --delete_profile "john&#x20;doe"  
"Chief Operator" "Applications"
```

This function also works in Command Center (Metaconsole).

add_profile_to_user

Required parameters:

- < id_user >
- < profile_name >

Optional parameter:

- < group_name >

Description: Add a existing profile in group to a user. If the group is not provided, the grupo will be `All`.

- This command allows repeating profile/group for a user without issuing a warning message. You must verify through the user listing in the PFMS Web Console if the profile was successfully added.

Related command: [add_profile](#).

Example (see [call](#) and [syntax](#)):

```
pandora_manage /etc/pandora/pandora_server.conf --add_profile_to_user 'user01'  
'Chief Operator' 'Network'
```

This function also works in Command Center (Metaconsole).

disable_eacl

Parameters: None.

Description: The [EACL system](#) will be disabled in the PFMS general configuration.

Example (see [call](#) and [syntax](#)):

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

enable_eacl

Parameters: None.

Description: The [EACL system](#) will be activated in the general PFMS configuration.

Example (see [call](#) and [syntax](#)):

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

disable_double_auth

Required parameter:

- < id_user >

Description: The [double authentication](#) will be disabled for the specified user.

Example (see [call](#) and [syntax](#)):

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

create_tag

Required parameter:

- < tag_name >
- < tag_description > single or double quotation marks can be used for a null description.

Optional aparameters:

- < tag_url > using any string will open a link in the Web Console.
- < tag_email >

Description: Creates a new [tag](#)].

Example (see [call](#) and [syntax](#)):

```
pandora_manage /etc/pandora/pandora_server.conf --create_tag "New tag" "My
description"
"index.php?sec=custom_report&sec2=godmode/reporting/reporting_builder"
```

```
pandora_manage /etc/pandora/pandora_server.conf --create_tag "New tag" "My
description" "https://example.com" "johndoe@example.com"
```

For it to be carried out correctly, the API configuration parameters will need to be filled in `pandora_server.conf` ([configuration file elements](#)): `console_api_url` and `console_api_pass` .

Events

create_event

Required parameters (3):

- < event_name > (allows repeating existing names).
- < event_type > One of these typified values:
 - unknown
 - alert_fired
 - alert_recovered
 - alert_ceased
 - alert_manual_validation
 - recon_host_detected
 - system
 - error
 - new_agent

 - going_up_warning
 - going_up_critical
 - going_down_warning

- going_unknown
 - going_down_normal
 - going_down_critical
 - going_up_normal
 - configuration_change
- Group name (except the group ALL).

Optional parameters:

1. Agent name.
2. Agent's Module name.
3. Event state (0 new event to be validated, 1 if is validated and 2 in process). In order to add the rest of the optional fields, one of these values must be specified.
4. Severity: 0 (Maintenance), 1 (Informational), 2 (Normal), 3 (Warning), 4 (Critical), 5 (Minor) y 6 (Major). In order to add the rest of the optional fields, one of these values must be specified.
5. Name of the alert template in case the specified module is associated to an alert.
6. User name or ID user (`id_user`, other than the `owner_user` of the event). If the user name or ID user includes space(s) or any other extended characters, these characters should be *escaped*. A space is represented as ` `; and `&` as `&`; , etcétera.
7. Comment. If the user name or identifier of the user making the comment includes spaces or any other extended characters, those characters should be *escaped*. A space is represented as ` `; and `&` as `&`; , et cetera (by Web Console will show the user name or user identifier *without escaping*, this is normal). It is recommended to use the user identifier as this is immutable and will always allow the user to be correctly identified.
8. Source, text field to indicate the origin of the event, keywords can be used.
9. Additional identifier (Extra ID), in the event view it is used for additional grouping.
10. (Version 6.0 SP5 OUM 625 or earlier, issue 525) Tags: Format should be `< tag > < url > , < tag > < url >` You can add multiple tags separated by commas. *It is important that there are no spaces between the comma and the next label.*
11. Custom data: Custom data should be entered as a JSON document. For example: `{"Location": "Office", "Priority": 42}`. If the JSON is poorly structured an unhandled exception will be raised, it is recommended to use the `json_pp` command to check the JSON before creating the event.
12. Force creation of agent (0 false, 1 true): If the agent name parameter refers to an agent that does not exist, it will be created (use null, two quotes together, in the module name and alert template name parameters). The alias of the new agent will be exactly the same name provided.
13. Critical instructions: Add the instructions to be performed in the case of a critical condition.
14. Warning instructions: Add the instructions to be performed in the case of a warning condition.
15. Unknown instructions: Add the instructions to perform in case of going to unknown state.
16. Use agent aliases, use textual `use_alias`. *Two single quotes must be specified together, with no space between them) if you use the following parameter below.*
17. Command Center (Metaconsole): Server identifier.
18. Personalized event identifier.
19. Default instructions.

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

Example (see [call](#) and [syntax](#)):

```
pandora_manage /etc/pandora/pandora_server.conf --create_event \
```

```
'Event CLI' \  
system \  
Firewalls \  
'My agent' 'My module' 0 4 Template004 \  

```

validate_event

Required parameters:

- < agent_name >

Optional parameters:

- < module_name >
- < date_time_min >
- < date_time_max >
- < owner_name >
- < id_criticality > One of these values: 0 (maintenance), 1 (informational), 2 (normal), 3 (warning), 4 (critical), 5 (minor).
- < alert_template_name > (without functionality)
- use_alias if you do not know the name of the agent, you can search by his or her alias.

Description: All the events will be validated according to a series of filters. The configurable filters are the name of the agent, the name of the module, a minimum date and time and a maximum date and time, the name of the user who owns the event, the criticality identifier and the name of the associated alert template.

You can combine the parameters in many ways, leaving blank with empty quotation marks "" or ' ' the ones you do not want to use (*including the agent name*) and filling in the others.

Example (see [call](#) and [syntax](#)):

```
pandora_manage /etc/pandora/pandora_server.conf --validate_event 'My agent' 'My  
module' '2024-06-02 22:02'
```

validate_event_id

Required parameter: < id_event >

Optional parameters: None.

Description: Allows to validate an event by means of its numeric integer identifier.

Example (see [call](#) and [syntax](#)):

```
pandora_manage /etc/pandora/pandora_server.conf 1234
```

get_event_info

Required parameter:

- < id_event >

Optional parameter:

- < separator > by default |.

Description: The information of an **event** given its identifier will be displayed.

Example (see [call](#) and [syntax](#)):

```
pandora_manage /etc/pandora/pandora_server.conf --get_event_info 341
```

add_event_comment

Required parameters:

- < id_event >
- < id_user >
- < comment >

Description: Add a comment to an event with a specified registered user.

Example (see [call](#) and [syntax](#)):

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

event_in_progress

Required parameter: < id_event >

Description: Set a event to "In progress" state by its numeric id.

Example (see [call](#) and [syntax](#)):

```
pandora_manage /etc/pandora/pandora_server.conf --event_in_progress 123
```

For it to be carried out correctly, the API configuration parameters will need to be filled in `pandora_server.conf` ([configuration file elements](#)): `console_api_url` and `console_api_pass`.

update_event_custom_id

Required parameters:

- < id_event >
- < id_custom_event >

Description: It adds a custom identifier to an event.

Example (see [call](#) and [syntax](#)):

```
pandora_manage /etc/pandora/pandora_server.conf --update_event_custom_id "232"  
"CLI test"
```

For it to be carried out correctly, the API configuration parameters will need to be filled in `pandora_server.conf` ([configuration file elements](#)): `console_api_url` and `console_api_pass`.

Policies

create_policy

Required parameteres:

- < new_monitoring_policy_name >
- < agent_group_name >

Optional parameter:

- < description >

Description: A [policy](#) with the specified name and belonging to the specified group will be created.

Example (see [call](#) and [syntax](#)):

```
pandora_manage /etc/pandora/pandora_server.conf --create_policy "New policy"  
"Workstations" "All desktop computers."
```

apply_policy

Required parameters:

- `< numeric_policy_id >` If the optional parameters are omitted, the policy will be applied to all agents belonging to it.

Optional parameters:

1. `< numeric_agent_id >`
2. `< use_numeric_id >` The default 0, to use the agent name in the optional parameter one you must specify 1 in this parameter.
3. `< server_id >` If you are working in a centralized environment this parameter must be specified, indicating the server to which the monitoring policy will be executed and applied.

Description: A `policy` will be applied in a forced way. Within the process of applying a policy is: the creation of the policy modules in all its associated agents, the creation of the policy alerts in the created modules, and making changes in the configuration file of the local agents that may have the policy to add the created modules and the collections associated to the policy.

Example (see `call` and `syntax`):

```
pandora_manage /etc/pandora/pandora_server.conf --apply_policy 6 1 0 1
```

For it to be carried out correctly, the API configuration parameters will need to be filled in `pandora_server.conf` (`configuration file elements`): `console_api_url` and `console_api_pass`.

apply_all_policies

Parameters: None.

Description: Adds to the application queue all the `policies`. The server is the one who observes the queue and applies the monitoring policies.

Example (see `call` and `syntax`):

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

add_agent_to_policy

Required parameters:

- `< agent_name >`
- `< policy_name >`

Optional parameter:

- `use_alias`

Description: An existing agent is added to an existing **policy** by specifying its name or, if the optional parameter is used, its alias.

Example (see **call** and **syntax**):

```
pandora_manage /etc/pandora/pandora_server.conf --add_agent_to_policy  
"pandorafms agent" "Basic Linux Monitoring" use_alias
```

add_collection_to_policy

Required parameters:

- < policy_name >
- < collection_name >

Optional parameters: None.

Description: An existing **collection** will be added to an existing **policy**.

Example (see **call** and **syntax**):

```
pandora_manage /etc/pandora/pandora_server.conf --add_collection_to_policy  
"Basic Remote Checks" "Apache Enterprise Plugin"
```

recreate_collection

Required parameter: < collection_id >

Optional parameters: None.

Description: Re-create the files of a **collection**.

Example (see **call** and **syntax**):

```
pandora_manage /etc/pandora/pandora_server.conf --recreate_collection 1
```

create_policy_data_module_from_local_component

Parameters:

- < policy_name >
- < component_name >

Description: A module taken from an existing local component will be added to an existing policy.

Example (see [call](#) and [syntax](#)):

```
pandora_manage /etc/pandora/pandora_server.conf \  
--create_policy_data_module_from_local_component \  
"Monitoring policy name" \  
"Local component name"
```

delete_not_policy_modules

Parameters: None.

Description: All modules that do not belong to any [policy](#) will be removed from both the database and the agent configuration file (if any).

Example (see [call](#) and [syntax](#)):

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

remove_agent_from_policy

Required parameters:

- < policy_id >
- < agent_id >

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

Example (see [call](#) and [syntax](#)):

```
pandora_manage /etc/pandora/pandora_server.conf --remove_agent_from_policy '1'  
'1'
```

For it to be carried out correctly, the API configuration parameters will need to be filled in `pandora_server.conf` ([configuration file elements](#)): `console_api_url` and `console_api_pass`.

disable_policy_alerts

Required parameter: < policy_name >

Optional parameter: None.

Description: All alerts of a [policy](#) passed by parameter will be marked as disabled.

Example (see [call](#) and [syntax](#)):

```
pandora_manage /etc/pandora/pandora_server.conf --disable_policy_alerts "Basic  
HP-UX Local Monitoring"
```

create_policy_data_module

Required parameters:

- < policy_name >
- < module_name >
- < tipo_modulo >

Optional parameters:

- < 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 >
- ignore_unknown Discards the unknown state calculation, so that the module will never change to that state and will always keep the last known state.

Description: A data module will be created in a policy. The default values [are the same as for the --create_data_module](#) option.

Example (see [call](#) and [syntax](#)):

```
pandora_manage /etc/pandora/pandora_server.conf --create_policy_data_module  
'Policy name' 'Module name' generic_proc 'Module description' 'group module' 0
```

```
100 0 300 30 60 61 100 0 'module_begin\nmodule_name modname\nmodule_end'  
'Warning string' 'Critical string'
```

create_policy_web_module

Required parameters:

- < policy_name >
- < module_name >
- < module_type > web_data, web_proc, web_content_data, web_content_string.

Optional parameters:

- < 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 >
- ignore_unknown Discards the unknown state calculation, so that the module will never change to that state and will always keep the last known state.

Description: A web module will be created in a policy. The default values are the same as for the `--create_web_module` option.

Example (see `call` and `syntax`):

```
pandora_manage /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 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 10 1 1 'critical_instructions' 'warning_instructions'
'unknown_instructions'
```

create_policy_network_module

Required parameters:

- < policy_name >
- < module_name >
- < module_type >

Optional parameters:

- < module_port > numerical value between 1 and 65535.
- < 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 >
- ignore_unknown Discards the unknown state calculation, so that the module will never change to that state and will always keep the last known state.

Description: A network module will be created in a policy. The default values are the same as for

the `--create_network_module` option.

Example (see [call](#) and [syntax](#)):

```
pandora_manage /etc/pandora/pandora_server.conf --create_policy_network_module  
'Policy name' 'Module name' remote_icmp_proc 22 'Module description' 'Name  
group' 0 100 0 300 30 60 61 100 0 0 'Warning string' 'Critical string'
```

create_policy_snmp_module

Required parameters:

- < policy_name >
- < module_name >
- < module_type >
- < module_port >
- < version >

Optional parameters:

- < 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 >

- `ignore_unknown` Discards the unknown state calculation, so that the module will never change to that state and will always keep the last known state.

Description: An SNMP module will be created in a policy. The default values are the same as for the `--create_snmp_module` option.

Example (see `call` and `syntax`):

```
pandora_manage /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

Required parameters:

- `< policy_name >`
- `< module_name >`
- `< module_kind >`
- `< module_port >` numerical value between 1 and 65535.
- `< plugin_name >`
- `< user >`
- `< password >`
- `< parameters >`

Optional 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 >`
- `ignore_unknown` Discards the unknown state calculation, so that the module will never change to that state and will always keep the last known state.

Description: A module of type *plugin* will be created in a policy. The default values are the same as for the `--create_plugin_module` option.

Example (see [call](#) and [syntax](#)):

```
pandora_manage /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 'Warning  
string' 'Critical string'
```

validate_policy_alerts

Required parameter: < policy_name >

Optional parameters: None.

Description: Validates all alerts for a given [policy](#).

Example (see [call](#) and [syntax](#)):

```
pandora_manage /etc/pandora/pandora_server.conf --validate_policy_alerts "Basic  
Linux Monitoring"
```

get_policy_modules

Required parameter: < policy_name >

Optional parameters: None.

Description: Gets the list of modules (numeric identifier and name) of a given [policy](#).

Example (see [call](#) and [syntax](#)):

```
pandora_manage /etc/pandora/pandora_server.conf --get_policy_modules "Basic  
Linux Monitoring"
```

get_policies

Required parameters: None.

Optional parameters:

- < agent_name >
- `use_alias` to search by agent alias.

Description:

- Without parameters: Obtains all policies (numeric identifier and name).
- Agent name: Gets the policies of an agent (list agent name, numeric policy identifier and policy name).

Example (see [call](#) and [syntax](#)):

```
pandora_manage /etc/pandora/pandora_server.conf --get_policies "pandorafms agent" use_alias
```

NetFlow

create_netflow_filter

Required parameters:

- < new_filter_name >
- < agent_group_name >
- < filter_as_such >
- < aggregated_by > One of the following values: none, dstip, dstport, proto, srcip, srcport.
- < formato_de_salida > kilobytes, kilobytespersecond, megabytes, megabytespersecond.

Optional parameters: None.

Description: Create a new filter to monitor via NetFlow®.

Example (see [call](#) and [syntax](#)):

```
pandora_manage /etc/pandora/pandora_server.conf --create_netflow_filter "New filter" "Network" "host 192.168.50.3 OR host 192.168.50.4 or HOST 192.168.50.6" dstport kilobytes
```

Tools

exec_from_file

Required parameters (see case study):

- < command_to_execute >
- < parameters_of_the_command_to_execute >

Description: With this option it is possible to execute any CLI PFMS command with macros from a

CSV file. The number of macros will correspond to the number of columns in the CSV file. Each macro will be called `__FIELD1__`, `__FIELD2__`, `__FIELD3__`, ... , `__FIELDn__`.

Commas in CSV columns are not yet supported.

Case study:

Register users with the command `create_user` from CSV file `/tmp/users_csv`:

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

Sentence to be executed (line connectors are included for didactic purposes):

```
pandora_manage /etc/pandora/pandora_server.conf --exec_from_file \  
/tmp/users_csv \  
create_user \  
__FIELD1__ \  
__FIELD2__ \  
__FIELD3__ \  
'Created by exec_from_file CLI PFMS command'
```

create_snmp_trap

Required parameters:

- `< snmp_trap_name >` As will be seen in the [SNMP traps console](#).
- `< OID >`
- `< description >`
- `< severity >` Numeric value, one of the following values: 0 (Maintenance), 1 (Info) , 2 (Normal), 3 (Warning), 4 (Critical), 5 (Minor), 6 (Major).

Description: Creates a new SNMP trap definition. A name must be entered, if an OID is repeated, the trap will not be created.

Example (see [call](#) and [syntax](#)):

```
pandora_manage /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"
```

start_snmptrapd

Parameters: None.

Starts the `snmptrapd` process in the [SNMP traps server](#). In case it is already started, it terminates this service (kill command) and executes it again, deleting all the traps located in the exchange log located by default in:

```
/var/log/pandora/pandora_snmptrapd.log
```

Example (see [call](#) and [syntax](#)):

```
pandora_manage /etc/pandora/pandora_server.conf --start_snmptrapd
```

kill

Required parameter:

- < PFMS_server_name >

Optional parameters: None.

Description: The [multiprocess](#) option should be enabled so that PFMS servers are executed in individual processes and can be detected by this command. Once this configuration is done, using this command without parameters will list the servers running and with their corresponding names to be terminated, one by one.

Example (see [call](#) and her [syntax](#)):

```
pandora_manage /etc/pandora/pandora_server.conf --kill dataserver
```

Graphs

create_custom_graph

Required parameters:

- < new_graph_name >
- < description >
- < id_user >
- < id_group >
- < width >
- < height >
- < events >
- < graph_type > By default 0 (Area) or one of the follow values:

1. Stacked area.
2. Line.
3. Stacked line.
4. Bullet chart.
5. Gauge.
6. Horizontal bars.
7. Vertical bars.
8. Pie chart.

- < period >
- < modules >
- < modules_separator >

Description: A chart will be created with the specified elements. Two or more modules must be distinguished with < modules_separator >, which must always be specified, even for a single module.

Although all parameters are mandatory, some can be left empty using single quotes. In such cases the default values taken by the parameters are as follows:

- Width: 550.
- Height: 210.
- Period: 86400 (in seconds).
- Events: 0.
- Graph type: 0.
- ID group: 0.

Example (see [call](#) and [syntax](#)):

```
pandora_manage /etc/pandora/pandora_server.conf --create_custom_graph "New
graph" "Graph created by CLI PFMS" "admin" "0" "" "" "0" "2" "" "1;2;3" ";"
```

edit_custom_graph

Required parameters:

- < graphic_id_to_edit >
- < graphic_name_to_edit >
- < description_to_edit >
- < id_user_to_edit >
- < group_id_to_edit >
- < width_to_edit >
- < height_to_edit >
- < events_to_edit >
- < graph_type_to_edit > By default 0 (Area) or one of the follow values:

1. Stacked area.
 2. Line.
 3. Stacked line.
 4. Bullet chart.
 5. Gauge.
 6. Horizontal bars.
 7. Vertical bars.
 8. Pie chart.
- < period_to_edit >

Description: A chart will be edited with the specified elements. *All parameters are mandatory, but some can be left empty with single quotes.* If this happens, the values will be the ones the chart already had before editing.

Example (see [call](#) and [syntax](#)):

```
pandora_manage /etc/pandora/pandora_server.conf --edit_custom_graph "1" "" "edit
graph by CLI" "" "" "" "" "" "" "" "" 3600
```

add_modules_to_graph

Required parameters:

1. < id_graph >
2. < id_modules > up to a maximum of 10 modules, delimited with < separator >.
3. < separator >

Description: The modules specified in the selected chart will be added. Although more than 10 modules can be added, only the first 10 will be displayed.

Example (see [call](#) and [syntax](#)):

```
pandora_manage /etc/pandora/pandora_server.conf --add_modules_to_graph 1 '1,2,3'
','
```

```
pandora_manage /etc/pandora/pandora_server.conf --add_modules_to_graph 1
'4;5;6;7' ';' ;'
```

delete_modules_to_graph

Required parameters:

- < id_graph >
- < id_modules > numeric module identifiers must be delimited with < separator >
- < separator >

Description: The specified modules will be removed from the selected chart.

Example (see [call](#) and [syntax](#)):

```
pandora_manage.pl /etc/pandora/pandora_server.conf --delete_modules_to_graph "1" "2,4,6" ",,"
```

Clusters

new_cluster

Required parameters:

- < cluster_name >
- < cluster_type > solamente AA o AP.
- < description >
- < group_id >

Description: A [cluster](#) will be created with the specified parameters and its associated elements.

Example (see [call](#) and [syntax](#)):

```
pandora_manage /etc/pandora/pandora_server.conf --new_cluster \  
"New cluster" "AA" "My description" "2"
```

For it to be carried out correctly, the API configuration parameters will need to be filled in `pandora_server.conf` ([configuration file elements](#)): `console_api_url` and `console_api_pass`.

add_cluster_agent

Required parameter:

- < json_data_base64 > in JSON format, then encoded in base 64: pairs of values with the cluster identifier and the agent identifier. Example:

```
[{"id":5,"id_agent":2},{"id":5,"id_agent":3}]
```

Description: A specific agent will be added to the specified [cluster](#). It is absolutely required that the JSON data be base 64 encoded.

Continuation of the previous example (see [call](#) and [syntax](#)):

```
pandora_manage /etc/pandora/pandora_server.conf --add_cluster_agent \  
W3siaWQiOi0jUslmlkX2FnZW50IjoyfSwKeyJpZCI6NSwiaWRfYWdlbnQiOi0jN9XQ==
```

For it to be carried out correctly, the API configuration parameters will need to be filled in `pandora_server.conf` ([configuration file elements](#)): `console_api_url` and `console_api_pass`.

add_cluster_item (active / active)

Parameters:

JSON data with this fields:

- `name`: (string value).
- `id_cluster`: (numeric value).
- `type`: AA (means "Active/Active").
- `critical_limit`: (numeric value).
- `warning_limit`: (numeric value).

Example JSON:

```
[
  {
    "name": "Swap_Used",
    "id_cluster": 5,
    "type": "AA",
    "critical_limit": 80,
    "warning_limit": 60
  },
  {
    "name": "TCP_Connections",
    "id_cluster": 5,
    "type": "AA",
    "critical_limit": 80,
    "warning_limit": 60
  }
]
```

Description: A specific module will be added as an item to the specified [cluster](#).

It is absolutely necessary that JSON data is encoded on base 64.

Continuation of previous example (see [call](#) and [syntax](#)):

```
perl pandora_manage.pl /etc/pandora/pandora_server.conf --add_cluster_agent
WwogIHsKICAgICJuYW1lIjogIlN3YXBfVXNlZCIsCiAgICAiaWRfY2x1c3RlciI6IDUsCiAgICAidHlw
ZSI6ICJBQSI6ICAgICAiY3JpdGljYWxfbGltaxQ0iA4MCwKICAgICJ3YXJuaW5nX2xpbWl0IjogNjAK
ICB9LAogIHsKICAgICJuYW1lIjogIlRDUf9Db25uZWNoaW9ucyIsCiAgICAiaWRfY2x1c3RlciI6IDUs
```



```
CiAgICAidHlwZSI6ICJBQSIscIAGICAIY3JpdGljYWxibGltZXQ0iA4MCwKICAgICJ3YXJuaW5nX2xp
bWl0IjogNjAKICB9Cl0=
```

For it to be carried out correctly, the API configuration parameters will need to be filled in `pandora_server.conf` ([configuration file elements](#)): `console_api_url` and `console_api_pass`.

add_cluster_item (active / passive)

Parameters:

JSON data with this fields:

- `name`: (string value).
- `id_cluster`: (numeric value).
- `type`: AP (means "Active/Passive").
- `critical_limit`: (numeric value).
- `warning_limit`: (numeric value).

Example JSON:

```
[
  {
    "name": "DiskUsed_/proc/kcore",
    "id_cluster": 5,
    "type": "AP",
    "is_critical": 1
  },
  {
    "name": "DiskUsed_/proc/sched_debug",
    "id_cluster": 5,
    "type": "AP",
    "is_critical": 1
  }
]
```

Description: A specific module will be added as an item to the specified [cluster](#).

It is absolutely necessary that JSON data is encoded on base 64.

```
pandora_manage /etc/pandora/pandora_server.conf --add_cluster_item \
WwogIHsKICAgICJuYWllIjogIkrRc2tVc2VkXy9wcm9jL2tjb3JlIiwKICAgICJpZF9jbHVzdGVyIjog
NSwKICAgICJ0eXBliIjogIkrFQIiwKICAgICJpc19jcmI0aWNhbCI6IDEKICB9LAogIHsKICAgICJuYWll
IjogIkrRc2tVc2VkXy9wcm9jL3NjaGVkX2RlYnVnIiwKICAgICJpZF9jbHVzdGVyIjogNSwKICAgICJ0
eXBliIjogIkrFQIiwKICAgICJpc19jcmI0aWNhbCI6IDEKICB9Cl0=
```

For it to be carried out correctly, the API configuration parameters will need to be filled in `pandora_server.conf` ([configuration file elements](#)): `console_api_url` and `console_api_pass`.

get_cluster_status

Required parameter: `< id_cluster >`

Description: Getting [cluster](#) status.

Example (see [call](#) and [syntax](#)):

```
pandora_manage /etc/pandora/pandora_server.conf --get_cluster_status 7
```

For it to be carried out correctly, the API configuration parameters will need to be filled in `pandora_server.conf` ([configuration file elements](#)): `console_api_url` and `console_api_pass`.

delete_cluster

Required parameter: `< cluster_id >`

Description: A [cluster](#) will be deleted.

Example (see [call](#) and [syntax](#)):

```
pandora_manage /etc/pandora/pandora_server.conf --delete_cluster 1
```

For it to be carried out correctly, the API configuration parameters will need to be filled in `pandora_server.conf` ([configuration file elements](#)): `console_api_url` and `console_api_pass`.

delete_cluster_agent

Required parameters:

- `< agent_id >`
- `< cluster_id >`

Description: An agent added to a [cluster](#) will be disassociated

Example (see [call](#) and [syntax](#)):

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

For it to be carried out correctly, the API configuration parameters will need to be filled in `pandora_server.conf` ([configuration file elements](#)): `console_api_url` and `console_api_pass`.

delete_cluster_item

Required parameter: `< item_id >`

Description: A [cluster](#) item will be deleted

Example (see [call](#) and [syntax](#)):

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

For it to be carried out correctly, the API configuration parameters will need to be filled in `pandora_server.conf` ([configuration file elements](#)): `console_api_url` and `console_api_pass`.

Visual Consoles

create_visual_console

Required parameters:

1. `< name >`
2. `< background_image >` one file stored in `.../pandora_console/images/console/background/`.
3. `< width >`
4. `< height >`
5. `< group_name >`
6. `< mode >` `static_objects` or `auto_creation`: In the first way the elements will be created without taking into account the position where to place the elements (the position described in the element JSON itself will be used), [see case study 1](#). If the mode is `auto_creation` a coordinate tapestry must be set (field `< position_to_locate_elements >`) and the algorithm will automatically set the position and size of each of the elements, [see case study 2](#).

Optional parameters:

- `< position_to_locate_elements >`
- `< background_color >` in hexadecimal format.
- `< elements >` in JSON format. It is absolutely necessary that the parameters in JSON are enclosed in

single quotes in order to include the double quotes of the data.

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

Due to database restrictions, each element to be added in the JSON (last parameter) must contain the following elements:

```
"id_layout_linked_weight": 0, "element_group": 0, "show_on_top": 0
```

Case study 1

Example with `auto_creation` and line connectors (see [call](#) and [syntax](#)):

```
pandora_manage /etc/pandora/pandora_server.conf --create_visual_console \
'Visual Console test 1' \
'map-africa.png' \
'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","id_layout_linked_weight": 0, "element_group":
0,"show_on_top":0},{ "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","id_layout_linked_weight": 0,
"element_group": 0,"show_on_top":0}]'
```

The second JSON is illustrated:

```
[
{
  "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",
"id_layout_linked_weight": 0,
"element_group": 0,
"show_on_top": 0
},
{
  "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",
  "id_layout_linked_weight": 0,
  "element_group": 0,
  "show_on_top": 0
}
]
```

Case study 2

Example with `static_objects` and line connectors (see [call](#) and [syntax](#)):

```
pandora_manage /etc/pandora/pandora_server.conf --create_visual_console \
```

```
'Visual Console test 2' \
'mapa-asia.png' \
'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", "id_layout_linked_weight": 0,
"element_group": 0, "show_on_top":0}, {"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", "id_layout_linked_weight": 0,
"element_group": 0, "show_on_top":0}]'
```

The last JSON is illustrated:

```
[
  {
    "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",
    "id_layout_linked_weight": 0,
    "element_group": 0,
    "show_on_top": 0
  },
  {
    "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",
    "id_layout_linked_weight": 0,
    "element_group": 0,
    "show_on_top": 0
  }
]

```

Case study 3

Visual console with all available elements:

```

pandora_manage /etc/pandora/pandora_server.conf --create_visual_console \
"Visual Console with all elements" \
"map-southamerica.png" \
1024 \
768 \
12 \
"static_objects" \
"" \
"#000000" \
' [{"id":69,"image":"network","pos_y":8,"pos_x":7,"width":70,"height":70,"label":
"", "type":0, "period":3600, "id_agente_modulo":0, "id_agent":0, "id_layout_linked":0
, "parent_item":0, "enable_link":1, "id_metaconsole":0, "id_group":0, "id_custom_grap
h":0, "border_width":0, "type_graph": "area", "label_position": "down", "border_color"
: "", "fill_color": "", "id_layout_linked_weight":0, "element_group":0, "show_on_top":
0}, {"id":70,"image":"percent","pos_y":9,"pos_x":123,"width":100,"height":0,"labe
l": "", "type":15, "period":3600, "id_agente_modulo":0, "id_agent":0, "id_layout_linke
d":0, "parent_item":0, "enable_link":1, "id_metaconsole":0, "id_group":0, "id_custom_
graph":0, "border_width":0, "type_graph": "area", "label_position": "down", "border_co
lor": "#000000", "fill_color": "#bcbcbc", "id_layout_linked_weight":0, "element_group

```

```
":0,"show_on_top":0},{ "id":71,"image":"white","pos_y":9,"pos_x":239,"width":300,"height":180,"label":"","type":1,"period":3600,"id_agente_modulo":0,"id_agent":0,"id_layout_linked":0,"parent_item":0,"enable_link":1,"id_metaconsole":0,"id_group":0,"id_custom_graph":0,"border_width":0,"type_graph":"area","label_position":"down","border_color":"","fill_color":"","id_layout_linked_weight":0,"element_group":0,"show_on_top":0},{ "id":72,"image":"","pos_y":0,"pos_x":614,"width":394,"height":175,"label":"","type":23,"period":3600,"id_agente_modulo":0,"id_agent":0,"id_layout_linked":0,"parent_item":0,"enable_link":1,"id_metaconsole":0,"id_group":0,"id_custom_graph":0,"border_width":0,"type_graph":"area","label_position":"down","border_color":"#000000","fill_color":"","id_layout_linked_weight":0,"element_group":0,"show_on_top":0},{ "id":73,"image":"","pos_y":197,"pos_x":14,"width":300,"height":300,"label":"","type":17,"period":3600,"id_agente_modulo":0,"id_agent":0,"id_layout_linked":0,"parent_item":0,"enable_link":1,"id_metaconsole":0,"id_group":0,"id_custom_graph":0,"border_width":0,"type_graph":"area","label_position":"down","border_color":"#ffffff","fill_color":"","id_layout_linked_weight":0,"element_group":0,"show_on_top":0},{ "id":74,"image":"white","pos_y":198,"pos_x":354,"width":300,"height":180,"label":"","type":18,"period":3600,"id_agente_modulo":0,"id_agent":0,"id_layout_linked":0,"parent_item":0,"enable_link":1,"id_metaconsole":0,"id_group":0,"id_custom_graph":0,"border_width":0,"type_graph":"horizontal","label_position":"down","border_color":"#000000","fill_color":"","id_layout_linked_weight":0,"element_group":0,"show_on_top":0},{ "id":75,"image":"","pos_y":398,"pos_x":481,"width":500,"height":70,"label":"","type":14,"period":86400,"id_agente_modulo":0,"id_agent":0,"id_layout_linked":0,"parent_item":0,"enable_link":1,"id_metaconsole":0,"id_group":0,"id_custom_graph":0,"border_width":0,"type_graph":"area","label_position":"down","border_color":"#000000","fill_color":"","id_layout_linked_weight":0,"element_group":0,"show_on_top":0},{ "id":76,"image":"string","pos_y":510,"pos_x":9,"width":0,"height":0,"label":"(_value)","type":2,"period":300,"id_agente_modulo":0,"id_agent":0,"id_layout_linked":0,"parent_item":0,"enable_link":1,"id_metaconsole":0,"id_group":0,"id_custom_graph":0,"border_width":0,"type_graph":"area","label_position":"down","border_color":"","fill_color":"","id_layout_linked_weight":0,"element_group":0,"show_on_top":0},{ "id":77,"image":"","pos_y":511,"pos_x":76,"width":50,"height":50,"label":"","type":4,"period":3600,"id_agente_modulo":0,"id_agent":0,"id_layout_linked":0,"parent_item":0,"enable_link":1,"id_metaconsole":0,"id_group":0,"id_custom_graph":0,"border_width":0,"type_graph":"area","label_position":"down","border_color":"","fill_color":"","id_layout_linked_weight":0,"element_group":0,"show_on_top":0},{ "id":78,"image":"http://192.168.7.117/pandora_console/images/console/icons/appliance.png","pos_y":510,"pos_x":162,"width":70,"height":70,"label":"","type":5,"period":3600,"id_agente_modulo":0,"id_agent":0,"id_layout_linked":0,"parent_item":0,"enable_link":1,"id_metaconsole":0,"id_group":0,"id_custom_graph":0,"border_width":0,"type_graph":"area","label_position":"down","border_color":"","fill_color":"","id_layout_linked_weight":0,"element_group":0,"show_on_top":0},{ "id":79,"image":"","pos_y":513,"pos_x":253,"width":100,"height":50,"label":"","type":19,"period":3600,"id_agente_modulo":0,"id_agent":0,"id_layout_linked":0,"parent_item":0,"enable_link":0,"id_metaconsole":0,"id_group":0,"id_custom_graph":0,"border_width":0,"type_graph":"area","label_position":"down","border_color":"#000000","fill_color":"#000000","id_layout_linked_weight":0,"element_group":0,"show_on_top":0},{ "id":80,"image":"appliance","pos_y":515,"pos_x":343,"width":70,"height":70,"label":"","type":11,"period":3600,"id_agente_modulo":0,"id_agent":0,"id_layout_linked":0,"parent_item":0,"enable_link":1,"id_metaconsole":0,"id_group":0,"id_custom_graph":0,"border_width":0,"type_graph":"area","label_position":"down","border_color":"","fill_color":"","id_layout_linked_weight":0,"element_group":0,"show_on_top":0},
```



```
{
  "id":81,"image":"","pos_y":495,"pos_x":445,"width":100,"height":100,"label":"","
  "type":12,"period":3600,"id_agente_modulo":0,"id_agent":0,"id_layout_linked":0,"
  parent_item":0,"enable_link":1,"id_metaconsole":0,"id_group":0,"id_custom_graph"
  :0,"border_width":1,"type_graph":"area","label_position":"down","border_color":"
  #000000","fill_color":"#ffffff","id_layout_linked_weight":0,"element_group":0,"s
  how_on_top":0},{
  "id":82,"image":"","pos_y":158,"pos_x":91,"width":42,"height":43
  ,"label":"","type":13,"period":3600,"id_agente_modulo":0,"id_agent":0,"id_layout
  _linked":0,"parent_item":0,"enable_link":1,"id_metaconsole":0,"id_group":0,"id_c
  ustom_graph":0,"border_width":1,"type_graph":"area","label_position":"down","bor
  der_color":"#000000","fill_color":"","id_layout_linked_weight":0,"element_group"
  :0,"show_on_top":0},{
  "id":83,"image":"","pos_y":488,"pos_x":562,"width":169,"hei
  ght":183,"label":{"\"default_color\": \"#000000\", \"color_ranges\": []},"type":20
  ,"period":3600,"id_agente_modulo":0,"id_agent":0,"id_layout_linked":0,"parent_it
  em":0,"enable_link":1,"id_metaconsole":0,"id_group":0,"id_custom_graph":0,"borde
  r_width":0,"type_graph":"area","label_position":"down","border_color":"#000000",
  "fill_color":"","id_layout_linked_weight":0,"element_group":0,"show_on_top":0},{
  "id":84,"image":"","pos_y":533,"pos_x":755,"width":909,"height":560,"label":{"\"
  labelStart\":null,\"labelEnd\":null,\"linkedStart\":null,\"linkedEnd\":null},"t
  ype":21,"period":3600,"id_agente_modulo":0,"id_agent":0,"id_layout_linked":0,"pa
  rent_item":0,"enable_link":1,"id_metaconsole":0,"id_group":0,"id_custom_graph":0
  ,"border_width":5,"type_graph":"area","label_position":"down","border_color":"#0
  00000","fill_color":"","id_layout_linked_weight":0,"element_group":0,"show_on_to
  p":0},{
  "id":85,"image":"","pos_y":0,"pos_x":0,"width":300,"height":150,"label":"
  ","type":22,"period":3600,"id_agente_modulo":0,"id_agent":0,"id_layout_linked":0
  ,"parent_item":0,"enable_link":1,"id_metaconsole":0,"id_group":0,"id_custom_grap
  h":0,"border_width":0,"type_graph":"area","label_position":"down","border_color"
  :"#000000","fill_color":"","id_layout_linked_weight":0,"element_group":0,"show_o
  n_top":0},{
  "id":86,"image":"","pos_y":583,"pos_x":864,"width":140,"height":56,"l
  abel":"","type":10,"period":3600,"id_agente_modulo":0,"id_agent":0,"id_layout_li
  nked":0,"parent_item":0,"enable_link":1,"id_metaconsole":0,"id_group":0,"id_cust
  om_graph":0,"border_width":0,"type_graph":"area","label_position":"down","border
  _color":"","fill_color":"","id_layout_linked_weight":0,"element_group":0,"show_o
  n_top":0},{
  "id":90,"image":"","pos_y":271,"pos_x":791,"width":116,"height":44,"l
  abel":"","type":22,"period":3600,"id_agente_modulo":129,"id_agent":3,"id_layout_
  linked":0,"parent_item":0,"enable_link":1,"id_metaconsole":0,"id_group":0,"id_cu
  stom_graph":0,"border_width":0,"type_graph":"area","label_position":"down","bord
  er_color":"#000000","fill_color":"","id_layout_linked_weight":0,"element_group":
  0,"show_on_top":0}]'
```

edit_visual_console

Required parameters (see description for details):

- < visual_console_console_number_id_to_change >
- < name >
- < background_image >
- < width >
- < height >
- < group_name >
- < mode >
- < position_to_locate_elements >

- < background_color >
- < elements >

Optional parameters: None.

Description: Works in conjunction with the `export_json_visual_console` command and allows you to edit an existing visual console and its elements. This command displays on screen the complete data of a registered visual console which can be copied and pasted to be modified.

After having called `edit_visual_console` and the numeric identifier of the visual console to be modified, the modified text can be pasted.

Example (see `call` and `syntax`):

- `Creación de una consola visual` sin elementos.

```
pandora_manage /etc/pandora/pandora_server.conf --create_visual_console \
  "My Visual Console" \
  "None.png" \
  1024 \
  768 \
  9 \
  "static_objects" \
  "" \
  "#ffffff" \
  '[]'
```

It will return something like this (note the numeric identifier, in this case 14):

```
[*] DB Host is 127.0.0.1
[INFO] Creating visual console 'My Visual Console'

[INFO] The visual console id is '14'

[root@pandorafms ~]#
```

- Export the code with the command `export_json_visual_console`, it is important that the last parameter is set to one to obtain the numerical identifiers of each of the elements (although in this case no item is registered).

```
pandora_manage /etc/pandora/pandora_server.conf --export_json_visual_console
"14" "" "1"
```

It will return something like this (the complete line must be copied below the second informational message [INFO]):

```
[*] DB Host is 127.0.0.1
[INFO] Exporting visual console elements with ID '14'

[INFO] JSON file now contents:
```

```
"My Visual Console" "None.png" 1024 768 9 "static_objects" "" "#ffffff" '[]'
[root@pandorafms ~]#
```

- Invoke the command `edit_visual_console`, a space, the console identifier, `visual`, another space and paste the code obtained in the previous step. Modify the name (or other element or elements).

```
pandora_manage /etc/pandora/pandora_server.conf --edit_visual_console \
"14" \
"My MODIFIED Visual Console" \
"None.png" \
1024 \
768 \
9 \
"static_objects" \
"" \
"#ffffff" \
'[]'
```

It will return something similar to the following:

```
[*] DB Host is 127.0.0.1
[INFO] The visual console with id 14 is updated
[root@pandorafms ~]#
```

It can be confirmed by the PFMS Web Console or by invoking the command `export_json_visual_console` again.

delete_visual_console

Required parameter:

- `< visual_console_id >`

Description: The visual console specified by means of its numerical identifier will be eliminated, in case of specifying a non-existent visual console, an error message will be displayed.

Example (see [call](#) and [syntax](#)):

```
pandora_manage /etc/pandora/pandora_server.conf --delete_visual_console "1"
```

delete_visual_console_objects

Required parameters:

- `< id_item >`

- < filter >
- < filter_value >

Description: The elements associated to a visual console will be eliminated following a filter. The available filters are:

- `image` to delete all items that have an exact image name.
- `id_agent` to delete all items that have a particular **numeric agent identifier**.
- `id_agente_modulo` to delete all items that have a particular **numeric module identifier**.
- `id_group` to delete all items belonging to a specific **numerical agent group identifier**. By default, and unless they have been modified, all the items belong to the All group identified with the value 0, specifying this value will delete most of the elements.
- `type_graph` to delete all items that have a particular type of graphic. By default, and unless modified, all items are set to area, specifying this value will delete most items.
- `type` to remove all items by their element type:
 - 0 Static image.
 - 1 Module Graph.
 - 2 Simple value.
 - 4 Label.
 - 5 Icon.
 - 10 Service.
 - 11 Group.
 - 12 Box.
 - 13 Line.
 - 14 Event history graph.
 - 15 Percentile item.
 - 17 Serialized pie graph.
 - 18 Bars graph.
 - 19 Clock.
 - 20 Color cloud.
 - 21 Network link.
 - 22 Odometer.
 - 23 Basic Chart.

Example (see [call](#) and [syntax](#)):

To delete all items containing the image network in the visual console 1:

```
pandora_manage /etc/pandora/pandora_server.conf --delete_visual_console_objects \
"1" image "network"
```

To delete all items showing agent 3 in the visual console 1:

```
pandora_manage /etc/pandora/pandora_server.conf --delete_visual_console_objects \
"1" id_agent "3"
```

To delete all items representing module 2 in the visual console 1:

```
pandora_manage /etc/pandora/pandora_server.conf --delete_visual_console_objects \
"1" id_agente_modulo "2"
```

To delete all items belonging to group 4 in visual console 1:

```
pandora_manage /etc/pandora/pandora_server.conf --delete_visual_console_objects \
"1" id_group "4"
```

To delete all items using the horizontal chart type in the visual console 1:

```
pandora_manage /etc/pandora/pandora_server.conf --delete_visual_console_objects \
"1" type_graph "horizontal"
```

To delete all items of type Odometer (22) in the visual console 1:

```
pandora_manage /etc/pandora/pandora_server.conf --delete_visual_console_objects \
"1" type "22"
```

duplicate_visual_console

Required parameters:

- < id_visual_console_to_copy >
- < number_of_copies > the suffix `_n` will be added from 1 to n.

Optional parameter:

- < prefix > will be added as a prefix and in addition the suffix `_n` from 0 to n-1.

Description: The indicated visual console will be duplicated the number of times needed.

Example (see [call](#) and [syntax](#)):

```
pandora_manage /etc/pandora/pandora_server.conf --duplicate_visual_console "1" \
"7" "copy of visual console #1"
```

export_json_visual_console

Required parameter:

- < visual_console_id >

Optional parameters:

- < directory_to_save >
- < with_element_identifier > 0 or 1, false and true, respectively. For the [edit_visual_console](#) command, the numeric identifier of each element is required.

Description: Exports, including data in JSON format, a visual console and its elements. You can specify the path where to save the file and if you want to include the identifiers of the associated elements to facilitate further editing.

Example (see [call](#) and [syntax](#)):

```
pandora_manage /etc/pandora/pandora_server.conf --export_json_visual_console "1" "/tmp" "1"
```

If no visual console is registered, it is normal to return this result:

```
[INFO] JSON file now contents:  
"" "" "static_objects" "" "" '[]'
```

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