# Configuración de los Agentes Software

# Configuración de los Agentes Software

## Qué es un Agente Software

Son pequeñas piezas de software que se instalan en los sistemas operativos y permanecen ejecutándose en ellos para extraer información de monitorización y enviarla regularmente al Pandora FMS Data server, que los procesa y almacena en la base de datos.

## Introducción a la configuración del Agente Software

El funcionamiento del Agente Software (parámetros de funcionamiento y módulos) está determinado por su fichero de configuración.

- En MS Windows®:

```
%ProgramFiles%\pandora_agent\pandora_agent.conf
```

- En sistemas GNU/Linux®:

```
/etc/pandora/pandora_agent.conf
```

## Parámetros generales del Agente

La mayoría de los parámetros son comunes para sistemas MS Windows® y GNU/Linux®. Después de haber modificado algún parámetro general deberá reiniciar el Agente Software.

La codificación del archivo de configuración del agente es
UTF-8 tanto en sistemas GNU/Linux® como MS Windows®.

La primera vez que se reciben datos del Agente Software se guarda toda la información en la base de datos. Para sucesivos envíos de información (y dependiendo si está habilitado el modo aprendizaje) sólo se actualizarán los siguientes campos del XML: `version`, `timestamp` (fecha) y `os_version` (versión de sistema operativo), así como los siguientes parámetros del archivo de configuración `gis_exec`, `latitude`, `longitude`, `altitude`, `parent_agent_name`, `timezone_offset`, `address`, `custom_field` .

### server_ip

Dirección IP o nombre del servidor de Pandora FMS al que se enviarán los datos.

### server_path

Ruta del servidor donde este recibe los ficheros de datos enviados por los agentes. Por defecto `/var/spool/pandora/data_in`.

### temporal

Ruta donde el Agente Software almacena los ficheros de datos antes de que sean enviados al servidor y eliminados localmente.

### description

Envía la descripción del Agente Software en el XML, y Pandora FMS importa esta descripción cuando crea el Agente Lógico.

### group

Si existe un grupo con el nombre indicado en este parámetro, el Agente se creará dentro de este grupo a no ser que el servidor fuerce la creación de todos los agentes en un grupo determinado.

### temporal_min_size

Si el espacio libre (en megabytes) de la partición en la que se encuentra el directorio temporal es menor que este valor (por defecto un megabyte), se detiene la generación de paquetes de datos.

### temporal_max_size

Tamaño máximo (en megabytes) permitido para el *buffer* XML, valor por defecto `1024`.

### temporal_max_files

Número máximo de archivos permitidos para el *buffer* XML, valor por defecto `1024`.

### logfile

Ruta del *log* del Agente de Pandora FMS.

### interval

Tiempo, en unidades de segundos, de muestreo del agente. Cada vez que se complete este intervalo el Agente recogerá información y la enviará al servidor de Pandora FMS.

### disable_logfile

Solo para MS Windows®: Inhabilita la escritura en `pandora_agent.log`.

### debug

Si se encuentra activo ( `1` ), los ficheros de datos del Agente son almacenados y renombrados en el directorio temporal y no son eliminados tras enviarse al servidor, pudiendo abrir los archivos XML y analizar su contenido.

### agent_name

Permite establecer un nombre personalizado. Si no se encuentra habilitado, el nombre del agente será el *hostname* de la máquina.

### agent_name_cmd

Define el nombre del Agente utilizando un comando externo. Si `agent_name_cmd` está definido, `agent_name` se ignora. El comando deberá devolver el nombre del agente por STDOUT. Si devuelve mas de una línea solo se utilizará la primera.

### agent_alias_cmd

Define el alias del Agente utilizando un comando externo. Si `agent_alias_cmd` está definido, `agent_alias` se ignora. El comando deberá devolver el nombre del agente por STDOUT. Si devuelve mas de una línea solo se utilizará la primera.

### address

Dirección IP o nombre de dominio asociado al Agente Software. Si se configura en `auto`, se obtendrá la dirección IP de la máquina y se añadirá al agente como dirección principal.

### encoding

Instala el tipo de codificación del sistema local, como por ejemplo `ISO-8859-15`, o `UTF-8`.

### server_port

Puerto en el que el servidor Tentacle de Pandora FMS escucha para recibir los archivos de datos, `41121` por defecto.

### transfer_mode

Modo de transferencia de los archivos de datos al servidor de Pandora FMS. Valor por defecto `tentacle`.

### transfer_timeout

*Timeout* para la transferencia de ficheros; si se supera el número de segundos indicado sin completar la transferencia, esta será cancelada.

### server_pwd

Contraseña del servidor para la autenticación: Específica para el FTP de Windows® y para el modo de transferencia Tentacle, aunque la contraseña en este último es opcional.

### server_ssl

Específica para el modo de transferencia Tentacle. Permite habilitar ( `yes` ) o deshabilitar ( `no` ) el cifrado de las conexiones mediante SSL.

### server_opts

Utilizado para configuraciones avanzadas de Tentacle para poder usar un proxy HTTP para enviar

los datos al servidor. Este proxy HTTP debe tener habilitado el método CONNECT. Para poder usar la salida a través de un *proxy*, use la siguiente opción (por ejemplo):

```
server_opts -y user:pass@proxy.inet:8080
```

Esta opción fuerza al cliente de Tentacle a enviar los datos a través de un proxy situado en `proxy.inet` y que usa el puerto `8080`, usando el usuario `user` y el contraseña `pass` para autenticarse en dicho *proxy*.

### delayed_startup

*Deshabilitado por defecto.* Tiempo de espera (segundos o minutos) hasta que el agente empieza a funcionar una vez iniciado. Para todos los Agentes Software excepto en MS Windows®.

### startup_delay

*Deshabilitado por defecto.* Tiempo de espera, en segundos, hasta que el agente empieza a funcionar una vez iniciado. Solamente para MS Windows®.

### pandora_nice

Sólo está disponible para agentes Unix/Linux. Este parámetro permite especificar la prioridad que el proceso del Agente de Pandora FMS tendrá en el sistema.

### autotime

Si está habilitado ( `1` ) envía un *timestamp* de ejecución especial (AUTO) que hace que el servidor utilice la fecha y hora local del servidor para establecer la hora de los datos, ignorando la hora enviada por el Agente. Esto es necesario en aquellos agentes que por la razón que sea, tienen una hora incorrecta o muy diferente de la del servidor.

### cron_mode

Con este parámetro es posible hacer que el Agente use el crontab de Linux® para ejecutarse en un intervalo determinado en vez de usar el propio sistema interno del Agente para ejecutarse cada cierto tiempo. Desactivado por defecto ( `0` ).

### remote_config

Habilita ( 1 ) o deshabilita ( 0 ) la configuración remota de los agentes. Su funcionamiento solo se permite con el modo de transferencia Tentacle.

### xml_buffer

Si se habilita ( 1 ), el Agente Software guardará en su directorio temporal los ficheros XML que no haya podido enviar al servidor en caso de un problema de conectividad. Serán enviados cuando se restablezcan las comunicaciones.

### timezone_offset

El Agente Software bien puede instalar su *timezone offset* con el servidor. Esto le permite al servidor hacer un desplazamiento de la hora recogida por el Agente, de forma que concuerde con la hora local del servidor.

```
# Timezone offset: Difference with the server timezone
timezone_offset 3
```

Se calcula restándole la zona horaria del agente a la zona horaria del servidor.

### parent_agent_name

Indica el *padre* del Agente Software. Debe ser el nombre de un Agente existente en Pandora FMS.

### agent_threads

Sólo disponible para agentes Unix/Linux: Número de hilos que lanzará el Agente para ejecutar módulos en paralelo. Por defecto, los Módulos se ejecutan uno tras otro sin lanzar ningún hilo adicional. Ejemplo:

```
# Number of threads to execute modules in parallel
agent_threads 4
```

### include

```
include <file>
```

Permite incluir un fichero ( < file > ) de configuración adicional. Este archivo puede incluir Módulos y colecciones adicionales a las del archivo principal. El fichero lo podrán subir aquellos usuarios que tengan permisos de escritura sobre Agentes (AW).

### broker_agent

```
broker_agent <broker_name>
```

Habilita la funcionalidad de Agente Broker. Para activarlo únicamente es necesario quitar de los comentarios el parámetro e indicar el nombre (< broker_name >) que se asignará al Agente Broker.

### pandora_user

```
pandora_user <user>
```

Este parámetro es opcional y permitirá ejecutar el Agente con el usuario del sistema ( <user> ) que especifique. Este usuario deberá contar con los permisos para poder ejecutar el Agente y sus recursos asociados.

### custom_id

Identificador personalizado del Agente para aplicaciones externas.

### url_address

URL personalizada para abrirla desde el Agente en la Consola.

### custom_fieldX_name

Nombre de un campo personalizado de Agentes que ya exista en el sistema. Si no existe, se ignorará. Ejemplo:

```
custom_field1_name Model
```

### custom_fieldX_value

Valor para el campo personalizado custom_fieldX_name definido en el parámetro anterior.

Ejemplo:

```
custom_field1_value C1700
```

### module_macro

Agente Software para Unix/Linux.

```
module_macro<_macro_name_> <value>
```

Define una macro de ejecución local que se puede utilizar en la definición de un Módulo. Estas macros se utilizan en el sistema de la Metaconsola y en el sistema de componentes de Módulos locales para “abstraer” la dificultad de usar un Módulo editando directamente el código, presentando a un usuario menos avanzado, una interfaz local que permita “rellenar” valores. Estos valores se emplean por debajo, usando un sistema de macros, relativamente similar al sistema de macros de los *plugins* locales.

Las macros de ejecución locales comienzan por `_fieldx_`.

### group_password

```
group_password <password>
```

*Password* para el grupo del Agente. Si el grupo no está protegido por contraseña debe dejar esta línea como comentario.

### ehorus_conf

```
ehorus_conf <path>
```

Ruta absoluta (*path*) a un fichero de configuración válido de un agente de eHorus. El Agente creará un campo personalizado llamado `eHorusID` que contiene la clave de identificación del agente de eHorus.

### transfer_mode_user

```
transfer_mode_user <user>
```

Usuario (*user*) de los ficheros copiados en el modo de transferencia local. En las carpetas de la Consola este usuario debe tener permisos de lectura y escritura para que funcione correctamente

la configuración remota. Por defecto es apache.

### secondary_groups

```
secondary_groups <group name1>, <group name2>, ... <group nameN>
```

Nombre de los grupos secundarios (*group name*) asignados al Agente. Se pueden especificar varios grupos secundarios separados por comas. Si alguno de los grupos no existe en el servidor al que se envía la información, no se asignará ese grupo, pero no se verá afectada la creación del Agente.

### standby

```
standby <1|0>
```

Si un Agente tiene modo de espera habilitado ( `standby 1` ), el Agente no realizará ningún chequeo ni enviará ni generará ningún XML. Esta directiva de configuración tiene sentido en instalaciones Enterprise donde hay configuración remota. Gracias a ello, se puede silenciar un Agente a voluntad con solo deshabilitarlo.

El modo debug sobreescribe esta funcionalidad y el Agente se ejecuta normalmente.

### module_absoluteinterval

```
module_absoluteinterval <interval>[s,m,h,d]
```

```
module_absoluteinterval once
```

Especifica el intervalo de ejecución del módulo, pero a diferencia de module_interval:

1. Recuerda la fecha de la última ejecución cuando se reinicia el agente. El módulo no se ejecutará hasta que haya pasado el intervalo especificado.
2. Permite especificar el intervalo en segundos, minutos, horas o días (e.g., `30s`, `5m`, `1h`, `7d`).
3. Es posible configurar módulos que se ejecuten una única vez especificando `once` como el valor del intervalo.

## Servidor Secundario

Se puede definir un servidor secundario al que se le enviarán los datos en dos posibles situaciones dependiendo de la configuración:

- on_error: Envía datos al servidor secundario solo si no puede enviarlas al primario.
- always: Siempre envía datos al servidor secundario, independientemente si puede contactar o no con

el servidor principal.

## Servidor UDP

Tenga presente que UDP es por naturaleza inseguro (pero eficiente para enviar mensajes sin comprometer una respuesta cierta).

El Agente Software de Pandora FMS puede configurarse para la escucha de comandos remotos. Este servidor escucha en un puerto UDP especificado por el usuario, y permite recibir órdenes desde un sistema remoto, habitualmente desde la Consola de Pandora FMS, mediante la ejecución de alertas en el servidor.

Para configurar el servidor remoto UDP, existen las siguientes opciones en su fichero de configuración `pandora_agent.conf>`

- udp_server: Para activar el servidor UDP establecer el valor a 1. Por defecto está desactivado.
- udp_server_port: Número de puerto de escucha.
- udp_server_auth_address: Direcciones IP autorizadas para enviar órdenes. Para especificar varias direcciones, hay que separarlas por comas. Si se configura con 0.0.0.0, acepta órdenes de cualquier dirección.

Aunque puede establecerse a *0.0.0.0* para que acepte desde todos los orígenes, dicha práctica no es recomendada. Si tiene varios Servidores PFMS y/o utiliza IPv6 puede colocar diferentes direcciones IP separadas por comas. Por ejemplo si tiene en IPv6:`2001:0db8:0000:130F:0000:0000:087C:140B` y su abreviatura es `2001:0db8:0:130F::87C:140B`utilice ambas separadas por comas.

- process_<name>_start <command>: Comando que arrancará un proceso definido por el usuario.
- process_<name>_stop <command>: Comando que parará el proceso.
- service_<name> 1: Permite que el servicio <name> sea parado o arrancado remotamente desde el servidor UDP.

Existe un *script* en el servidor, en `/util/udp_client.pl` que es el usado por Pandora FMS Server como comando de una alerta, para arrancar procesos, o servicios. Tiene esta sintaxis:

```
./udp_client.pl <address> <port> <command>
```

## Definición de los Módulos

Los Módulos de ejecución local se definen en el fichero de configuración `pandora_agent.conf`. La sintaxis general es la siguiente:

```
module_begin
module_name <module_name>
module_type generic_data
module_exec <local_command>
module_end
```

### Elementos comunes de todos los Módulos

Los campos del Módulo (salvo el dato del Módulo, la descripción y la información extendida) sólo se actualizan en la creación del Módulo, nunca se actualizarán una vez que el Módulo ya existe.

#### module_begin

Etiqueta de inicio de un Módulo. Obligatorio.

#### module_name

```
module_name <name>
```

Nombre (*name*) del Módulo. Dicho nombre debe ser único y singular en el Agente. Obligatorio.

#### module_type

```
module_type <type>
```

Tipo (*type*) de datos que devolverá el Módulo. Obligatorio. Los tipos disponibles son:

- Numérico ( `generic_data` ): Datos numéricos sencillos, con coma flotante o enteros.
- Incremental ( `generic_data_inc` ): Dato numérico igual a la diferencia entre el valor actual y el valor anterior dividida por el número de segundos transcurridos. Cuando esta diferencia es negativa, se reinicia el valor, esto significa que cuando la diferencia vuelva a ser positiva de nuevo se tomará el valor anterior siempre que el incremento vuelva a dar un valor positivo.
- Absoluto incremental ( `generic_data_inc_abs` ): Dato numérico igual a la diferencia entre el valor actual y el valor anterior, sin realizar la división entre el número de segundos transcurridos, para medir incremento total en lugar de incremento por segundo. Cuando esta diferencia es negativa, se reinicia el valor, esto significa que cuando la diferencia de nuevo vuelva a ser positiva, se empleará el último valor desde el que el actual incremento obtenido da positivo.
- Alfanumérico ( `generic_data_string` ): Recoge cadenas de texto alfanuméricas.
- Booleanos ( `generic_proc` ): Para valores que solo pueden ser correcto o afirmativo (1) o incorrecto o negativo (0). Útil para comprobar si un equipo está vivo, o un proceso o servicio está corriendo. Un valor negativo (0) trae preasignado el estado crítico, mientras que cualquier valor superior se considerará correcto.

- Alfanumérico asíncrono ( `async_string` ): Para cadenas de texto de tipo asíncrono. La monitorización asíncrona depende de eventos o cambios que pueden ocurrir o no, por lo que este tipo de Módulos nunca están en estado desconocido.
- Booleano asíncrono ( `async_proc` ): Para valores booleanos de tipo asíncrono.
- Numérico asíncrono ( `async_data` ): Para valores numéricos de tipo asíncrono.

**module_min**

```
module_min <value>
```

Valor (*value*) mínimo que el Módulo debe devolver para que sea aceptado. En caso contrario será descartado por el servidor.

**module_max**

```
module_max <value>
```

Valor (*value*) máximo que el Módulo debe devolver para que sea aceptado. En caso contrario será descartado por el servidor.

**module_min_warning**

```
module_min_warning <value>
```

Valor (*value*) mínimo del umbral de advertencia `warning`.

**module_max_warning**

```
module_max_warning <value>
```

Valor (*value*) máximo del umbral de advertencia `warning`.

**module_min_critical**

```
module_min_critical <value>
```

Valor mínimo del umbral crítico `critical`.

**module_max_critical**

```
module_max_critical <value>
```

Valor máximo del umbral crítico `critical`.

#### module_disabled

```
module_disabled <0|1>
```

Indica si el Módulo esta habilitado ( `0` ) o deshabilitado ( `1` ).

#### module_min_ff_event

```
module_min_ff_event <value>
```

Valor de la protección flip flop para falsos positivos. Será necesario que se produzcan el número de cambios de estado indicados en este valor para que el módulo modifique visualmente su estado en la Consola web.

#### module_each_ff

```
module_each_ff <0|1>
```

Si está habilitado ( `1` ), en vez de usar `module_min_ff_event` se utilizarán los umbrales flip flop por estado:

- `module_min_ff_event_normal`.
- `module_min_ff_event_warning`.
- `module_min_ff_event_critical`.

#### module_min_ff_event_normal

```
module_min_ff_event_normal <value>
```

Valor de la protección flip flop para paso a estado `normal`.

#### module_min_ff_event_warning

```
module_min_ff_event_warning <value>
```

Valor de la protección flip flop para paso a estado `warning`.

**module_min_ff_event_critical**

```
module_min_ff_event_critical <value>
```

Valor de la protección flip flop para paso a estado `critical`.

**module_ff_timeout**

```
module_ff_timeout <seconds>
```

Reinicia el contador de flip flop threshold después del número de segundos dado. Esto implica que el número de cambios de estado determinado en `module_min_ff_event` deberá ocurrir en un intervalo de `module_ff_timeout` segundos antes de que el estado cambie en la consola a nivel visual.

**module_ff_type**

Versión NG 734 o superior.

```
module_ff_type <value>
```

Se trata de una opción avanzada del Flip Flop para el control de estado de un Módulo. Mediante `Keep counters` establece unos valores de contador para pasar de un estado a otro dependiendo, en lugar del valor, del estado del módulo con el valor recibido.

Indica si `Keep counters` esta habilitado ( `1` ) o deshabilitado ( `0` ).

**module_ff_event**

```
module_ff_event X
```

Esta directiva es el umbral flip flop de ejecución del módulo (en segundos)

**module_description**

```
module_description <text>
```

Texto libre con información sobre el Módulo.

**module_interval**

```
module_interval <factor>
```

Intervalo individual de Módulo. Este valor es un factor multiplicador del intervalo del agente, no un tiempo libre.

Para que el `module_interval` funcione en Agentes Broker, debe tener el mismo intervalo que el del Agente del cual proviene. En caso contrario, puede fallar su funcionamiento. A partir de la versión 776 el campo intervalo de los Agentes broker en la Consola web ha sido retirado.

**module_timeout**

```
module_timeout <secs>
```

En segundos, tiempo máximo permitido para la ejecución del Módulo. Si se supera este tiempo antes de haber finalizado su ejecución, será interrumpida.

**module_postprocess**

```
module_postprocess <factor>
```

Valor numérico por el que se multiplicará el dato devuelto por el Módulo. Útil para realizar conversiones de unidades.

**module_save**

```
module_save <var name>
```

Almacena el valor devuelto por el Módulo en una variable con el nombre indicado en este parámetro ( `<var name>` ). Este valor podrá ser utilizado posteriormente en otros Módulos.

Ejemplo en Unix/Linux:

```
module_begin
module_name echo_1
module_type generic_data
module_exec echo 41121
module_save ECHO_1
```

```
module_end
```

Almacenará el valor “41121” en la variable “ECHO_1”.

```
module_begin
module_name echo_2
module_type generic_data
module_exec echo $ECHO_1
module_end
```

Este segundo Módulo mostrará el contenido de la variable “$ECHO_1”, siendo “41121”.

En Agentes Software en Windows® la sintaxis del Módulo debe formarse encerrando la variable entre símbolos de porcentaje `%var%` en lugar de `$var`. Siguiendo el ejemplo dado:

```
module_begin
module_name echo_2
module_type generic_data
module_exec echo %ECHO_1%
module_end
```

**module_crontab**

Se pueden programar los Módulos para que se ejecuten en determinadas fechas según el siguiente formato:

`module_crontab <minuto> <hora> <día> <mes> <día de la semana>`

Siendo:

- Minuto 0-59.
- Hora 0-23 .
- Día del mes 1-31
- Mes 1-12 .
- Día de la semana 0-6 (0 es Domingo) .

**module_condition**

module_condition <operación> <comando>

Permite definir acciones que serán ejecutadas por el Agente en función del valor devuelto por el Módulo. Solo disponible para valores numéricos. La sintaxis es la siguiente:

- > [valor]: Ejecuta el comando cuando el valor del Módulo es mayor que el valor dado.
- < [valor]: Ejecuta el comando cuando el valor del Módulo es menor que el valor dado.
- = [valor]: Ejecuta el comando cuando el valor del Módulo es igual al valor dado.
- != [valor]: Ejecuta el comando cuando el valor del Módulo es distinto al valor dado.

- =~ [expresion|regular]: Ejecuta el comando cuando el valor del módulo concuerda con la expresión regular dada.
- (valor, valor): Ejecuta el comando cuando el valor del módulo está comprendido entre los valores dados.

Se pueden especificar múltiples condiciones para un mismo módulo.

En el sistema operativo MS Windows® es recomendable anteponer cmd.exe /c al comando para asegurar que se ejecuta de forma adecuada.

**module_precondition**

Funciona de la misma manera que `module_condition`.

**module_unit**

```
module_unit <string>
```

Unidades expresada en una cadena de texto para mostrar junto al valor obtenido por el Módulo. Ejemplo: `module_unit %` .

**module_group**

```
module_group <value>
```

Permite indicar el grupo de Módulos al que será asignado el Módulo. Ejemplo: `module_group Networking` .

**module_custom_id**

```
module_custom_id <value>
```

Esta directiva es un identificador personalizado del Módulo. Ejemplo: `module_custom_id host101` .

**module_str_warning**

```
module_str_warning <value>
```

Permite indicar una expresión regular para definir el umbral de advertencia `warning` en Módulos de tipo alfanumérico.

**module_str_critical**

```
module_str_critical <value>
```

Permite indicar una expresión regular para definir el umbral crítico `critical` en Módulos de tipo alfanumérico.

**module_warning_instructions**

```
module_warning_instructions <value>
```

Informa de instrucciones que se mostrarán en el evento generado por el Módulo al pasar a estado de advertencia `warning`.

**module_critical_instructions**

```
module_critical_instructions <value>
```

Informa de instrucciones que se mostrarán en el evento generado por el módulo al pasar a estado `critical`.

**module_unknown_instructions**

```
module_unknown_instructions <value>
```

Informa de instrucciones que se mostrarán en el evento generado por el módulo al pasar a estado desconocido unknown.

**module_tags**

```
module_tags <value>
```

Etiquetas que se deseen asignar al Módulo, separadas por comas.

**module_warning_inverse**

```
module_warning_inverse <value>
```

Permite activar ( 1 ) el intervalo inverso para el umbral de advertencia `warning`.

**module_critical_inverse**

```
module_critical_inverse <value>
```

Permite activar ( 1 ) el intervalo inverso para el umbral crítico `critical`.

**module_native_encoding**

Para Win32 únicamente.

```
module_native_encoding <value>
```

Este *token* de configuración solo afecta a los Módulos que se ejecutan mediante una directiva de comandos, es decir, hay un `module_exec` presente.

MS Windows® maneja tres codificaciones para sus procesos: la codificación de la línea de comandos (OEM), la codificación del sistema (ANSI) y UTF-16. Estas codificaciones coinciden en los caracteres básicos, pero difieren en aquellos menos comunes, como podrían ser las tildes. Con este *token*, el agente de Pandora FMS convierte la salida del comando a la codificación especificada en el *encoding* del archivo de configuración.

`module_native_encoding` tiene cuatro valores válidos:

- `module_native_encoding OEM`: Para la codificación de la línea de comandos.
- `module_native_encoding ANSI`: Para la codificación del sistema.
- `module_native_encoding UTFLE`: Para UTF-16 little-endian.
- `module_native_encoding UTFBE`: Para UTF-16 big-endian.

Si no aparece `module_native_encoding`, no se realizará ninguna recodificación.

**module_quiet**

```
module_quiet <value>
```

Si se encuentra habilitado ( 1 ) el Módulo estará en modo silencioso: no generará eventos ni disparará alertas.

**module_ff_interval**

```
module_ff_interval <value>
```

Permite indicar un umbral Flip Flop en el Módulo.

#### module_macro

```
module_macro<macro> <value>
```

Solo aplicable en componentes locales desde la Consola. No tiene utilidad en el fichero de configuración.

#### module_alert_template

```
module_alert_template <template_name>
```

Esta macro asigna al Módulo creado la plantilla de alerta correspondiente al nombre introducido como parámetro (ver Plantillas de alerta).

#### intensive_interval

Intervalo de monitorización intensiva. Los módulos que utilicen `module_intensive_monitorig` podrán notificar si su estado es incorrecto en este intervalo.

#### module_intensive_condition

Condición para monitorización intensiva. Cuando un módulo de monitorización intensiva alcance el valor configurado en este parámetro, notificará en el intervalo intensivo definido.

#### module_end

Etiqueta de final de Módulo. Es obligatorio.

### Directivas específicas para obtener información

En cada Módulo *solo se puede utilizar uno de estos tipos*.

#### module_exec

```

```

```
module_exec <command>
```

Se debe especificar la ejecución deseada para obtener la información en una única línea.

En GNU/Linux, el comando se ejecutará mediante el intérprete de comandos por defecto. El intérprete por defecto viene determinado por el enlace simbólico de `/bin/sh`. Normalmente el enlace apunta a bash, pero en sistemas como Ubuntu no es así. Una solución que funcionará en la mayoría de ocasiones:

```
module_exec bash -c "<command>"
```

Si la ejecución del comando devuelve un código de error (return code) diferente de 0, se interpretará que el comando da error y se descartará el dato obtenido.

Para un Agente Software sobre MS Windows® existen más directivas para obtener datos, éstas se describen a continuación.

#### module_exec_powershell

Solamente para MS Windows®.

```
module_exec_powershell < commands >
```

Permite realizar chequeos nativos con PowerShell.

#### module_service

```
module_service <service>
```

Comprueba si un determinado servicio se está ejecutando en la máquina.

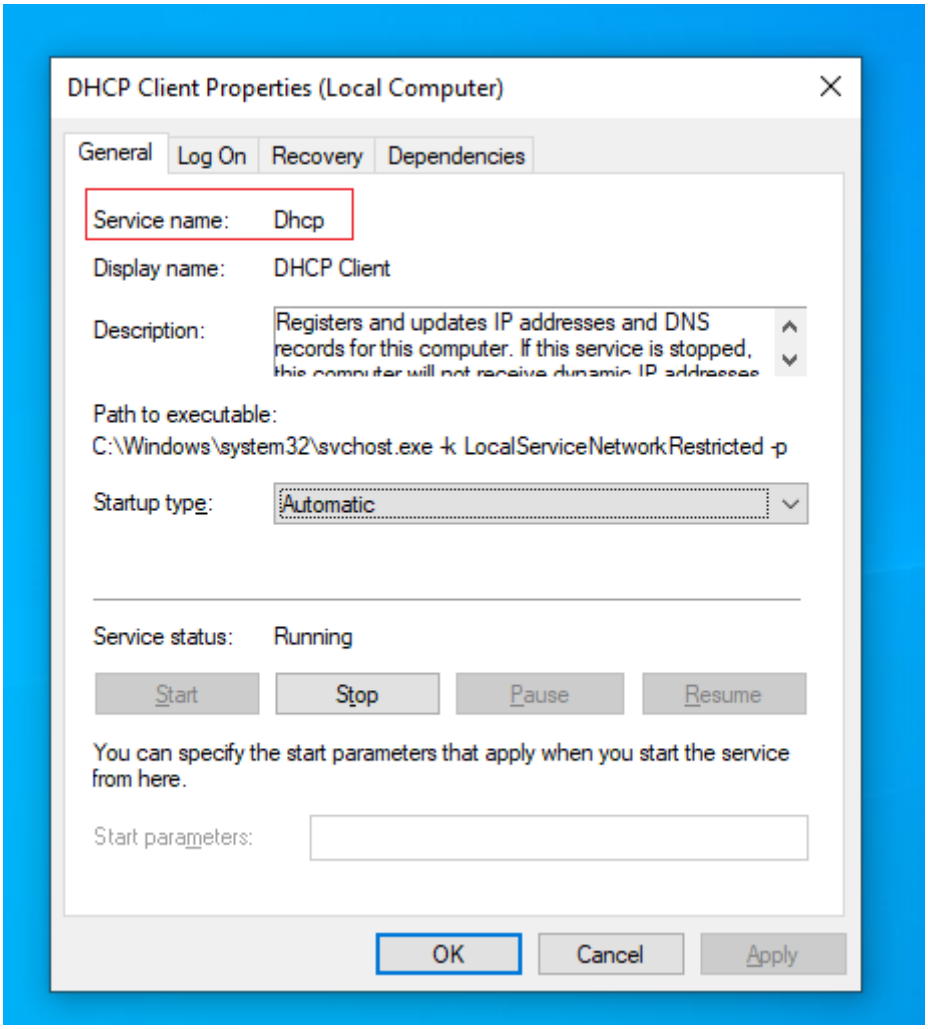
##### En MS Windows

Si el nombre del servicio contiene espacios en blanco debe utilizar entrecomillado “ ”.

```
module_begin
module_name Service_Dhcp
module_type generic_proc
module_service Dhcp
module_description Service DHCP Client
```

```
module_end
```

El servicio se identifica con el nombre corto del servicio ( `Service name` ), tal como aparece en el gestor de servicios de Windows.




*Modo asíncrono*

Para ello basta con agregar la directiva:

```
module_async yes
```

Esta funcionalidad no está soportada en los Agentes Broker.

En las versiones de Windows Home Edition® esta funcionalidad asíncrona no está soportada y, solamente en estas versiones, el Agente de Pandora FMS realiza una consulta periódica para saber si el servicio está corriendo o no. Esto puede consumir bastantes recursos así que se recomienda usar la versión síncrona si se monitoriza un número elevado de servicios.

*Watchdog de servicios*

Existe un modo de vigilancia o *watchdog* para los servicios, de tal forma que el agente puede iniciarlos de nuevo si estos se detienen, se especifica como:

```
module_watchdog yes
```

#### En Unix

En Unix funciona igual que en MS Windows®, solo que para Unix proceso y servicio es el mismo concepto.

El modo *watchdog* y la detección asíncrona no son posibles en el Agente de Unix.

Para `module_service` debe colocar la ruta completa tal cual aparece el servicio con el comando `ps aux` . Por ejemplo, para buscar el servicio SSH:

```
 ps aux | grep ssh
```

Se debe configurar:

```
module_service /usr/sbin/sshd -D
```

### module_proc

```
module_proc <process>
```

Comprueba si un determinado nombre de proceso está operando en esta máquina.

En MS Windows®

Es innecesario el entrecomillado para el nombre del proceso. Tenga en cuenta que el nombre del proceso debe tener la extensión `.exe`. El Módulo devolverá el número de procesos que se estén ejecutando con este nombre.

Modo asíncrono

De una forma similar a los servicios, monitorizar procesos puede ser crítico en algunos casos. Ahora el Agente Software para Windows® soporta comprobaciones asíncronas para el parámetro

`module_proc`. En este caso el Agente notifica inmediatamente cuando el proceso cambia de estado, sin esperar a que se cumpla el intervalo de ejecución del Agente. De esta forma, puede conocer de la caída de procesos críticos casi al instante de que ocurran. Esta funcionalidad no está soportada en los Agentes Broker.

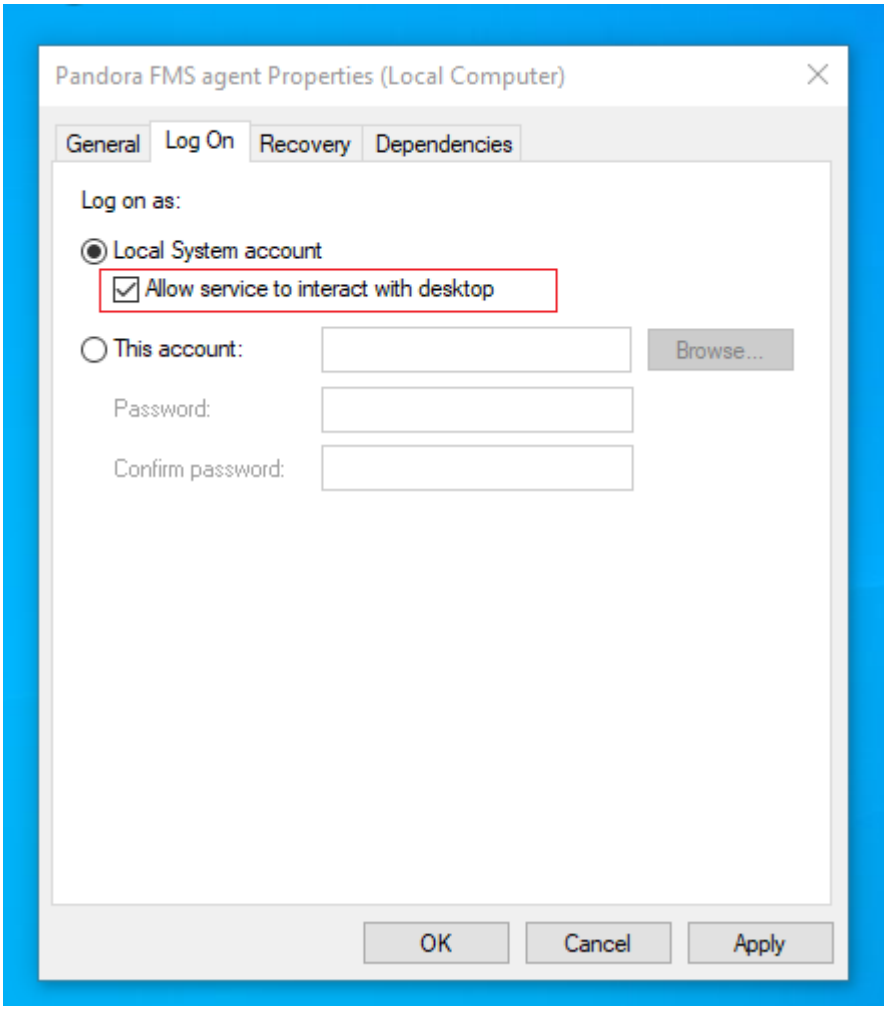
*Watchdog* de procesos

Es importante destacar que el modo *Watchdog* solo funciona cuando el tipo de módulo es *asíncrono*.

Dado que ejecutar un proceso puede requerir algunos parámetros, hay algunas opciones adicionales de configuración para este tipo de módulos. Ejemplo de configuración de un `module_proc` con watchdog:

```
module_begin
module_name Notepad
module_type generic_proc
module_proc notepad.exe
module_description Notepad
module_async yes
module_watchdog yes
module_start_command c:\windows\notepad.exe
module_startdelay 3000
module_retrydelay 2000
module_retries 5
module_end
```

Para versiones anteriores a Windows Vista® el *token* `module_user_session` puede configurarse de manera general habilitando en las propiedades del servicio de Pandora FMS la casilla “Acceso interactivo con el escritorio” (Allow service to interact with desktop):




Pandora FMS, como servicio, se ejecuta bajo la cuenta SYSTEM y que el proceso ejecutado lo hará bajo ese usuario y con ese entorno, de forma que si usted necesita ejecutar algún proceso determinado que requiera ser usado con un usuario específico, deberá encapsular en un *script* (`.bat` o similar) los procesos previos para inicializar el entorno, variables de entorno, etc), y ejecutar ese *script* como acción del *Watchdog*.

En Unix

En Unix funciona exactamente igual que en module_service. Tampoco soporta modo asíncrono ni *watchdog*.

**module_cpuproc**

Solamente para Unix.

```
module_cpuproc <process>
```

Devuelve el uso de CPU específico de un proceso.

**module_memproc**

Solamente para Unix.

```
module_memproc <process>
```

Devuelve el consumo de memoria específico de un proceso.

**module_freedisk**

```
module_freedisk <disk_letter:>|<vol>
```

Comprueba el espacio libre en la unidad.

**module_freepercentdisk**

```
module_freepercentdisk <disk_letter:>|<vol>
```

Este Módulo devuelve el porcentaje de disco libre en una unidad lógica.

**module_occupiedpercentdisk**

Solamente para Unix.

```
module_occupiedpercentdisk <vol>
```

Este Módulo devuelve el porcentaje de disco ocupado.

**module_cpuusage**

```
module_cpuusage [<cpu id>|all]
```

Devuelve el uso de CPU en un número de CPU. Si sólo existe una CPU no establezca ningún valor o utilice el valor `all`. Para Windows® y Unix.

**module_freememory**

Funciona tanto en Unix como en Windows®. Devuelve la memoria libre en todo el sistema.

```
module_begin
module_name FreeMemory
module_type generic_data
module_freememory
module_description Non-used memory on system
module_end
```

**module_freepercentmemory**

Funciona tanto en Unix como en MS Windows®. Este Módulo devuelve el porcentaje de memoria libre en un sistema:

```
module_begin
module_name freepercentmemory
module_type generic_data
module_freepercentmemory
module_end
```

**module_tcpcheck**

Solo MS Windows®. Este Módulo inicia conexión con la dirección IP y puerto especificados. Devuelve `1` si tuvo éxito y `0` en caso contrario. Se debe especificar un tiempo de expiración con `module_timeout`. Ejemplo:

```
module_begin
module_name tcpcheck
module_type generic_proc
module_tcpcheck www.pandorafms.com
module_port 80
module_timeout 5
module_end
```

**module_regexp**

Solamente para MS Windows®

Este Módulo monitoriza un *log* buscando coincidencias usando expresiones regulares, descartando las líneas ya existentes al iniciar la monitorización. Los datos devueltos por el Módulo dependen del tipo de Módulo:

- generic_data_string, async_string: Devuelve todas las líneas que coincidan con la expresión regular.
- generic_data: Devuelve el número de líneas que coincidan con la expresión regular.
- generic_proc: Devuelve 1 si existe alguna coincidencia, 0 de otra forma.
- module_noseekeof: Por defecto inactivo `0`. Con este *token* de configuración activo `1`, en cada ejecución, independientemente de las modificaciones en el fichero del *log*, el módulo reinicia su comprobación sin buscar el final del archivo (*flag*`EOF`). De esta manera siempre sacará en el XML todas aquellas líneas que coincidan con el patrón de búsqueda.

**module_wmiquery**

Solo Windows®. Los Módulos WMI permiten ejecutar localmente cualquier consulta o *query* WMI sin utilizar una herramienta externa. Se configura por medio de dos parámetros:

- module_wmiquery: WQL *query* empleada. Se pueden obtener varias líneas como resultado, que serán insertados como varios datos.
- module_wmicolumn: Nombre de la columna que se va a usar como fuente de datos.

**module_perfcounter**

Solamente para MS Windows®.

Obtiene los datos del contador de rendimiento (performance counter) a través de la interfaz de PDH. El fichero `pdh.dll` debe de estar instalado en el sistema. PDH.DLL pertenece a una biblioteca de MS Windows®, si no está disponible debe instalar la herramienta de análisis de rendimiento de MS Windows® que suele venir por defecto.

**module_inventory**

Actualmente esta funcionalidad ha sido sustituida por inventario desde plugins de Agente tanto en sistemas Windows® como Linux/Unix®.

**module_logevent**

Solamente para MS Windows®.

Permite obtener información del *log* de eventos de MS Windows® basándose en los patrones indicados, permitiendo filtrar en función de la fuente y el tipo de evento.

El formato general de este Módulo es el siguiente:

```
module_begin
module_name MyEvent
module_type async_string
module_logevent
module_source <logName>
module_eventtype <event_type/level>
module_eventcode <event_id>
module_application <source>
module_pattern <text substring to match>
module_description
module_end
```

Para evitar mostrar información repetida, solo se tienen en cuenta aquellos eventos que hayan tenido lugar desde la última vez que se ejecutó el Agente.

`module_logevent` acepta los siguientes parámetros, todos ellos exigen la introducción correcta de mayúsculas y minúsculas:

- module_source: Origen del evento (System, Application, Security). Campo obligatorio.
- module_eventtype: Tipo de evento (error, information...). Campo opcional.
- module_pattern: Patrón a buscar (subcadena). Campo opcional.
- module_eventcode: ID numerico del evento. Campo opcional.
- module_application: Aplicación que origina el evento registrado en el *log*. distinguir bien de `module_source` que indica el nombre de la fuente o fichero *log* de donde se buscan los eventos.

**module_logchannel**

Solamente para MS Windows®.

Tipo de Módulo que permite obtener información de los canales de *logs* de MS Windows®. Funciona de igual manera que `module_logevent`

Para obtener el nombre del canal del evento será necesario hacer clic derecho en el mismo, seleccionar Propiedades y copiar el parámetro Full name, necesario para `module_source`.

**module_plugin**

Para la ejecución de *plugins* de Agente. Es un caso especial ya que no requiere de ninguna otra etiqueta tipo `module_begin` o `module_end` y tampoco indicar el tipo de Módulo.

Sintaxis con sus respectivos parámetros:

```
module_plugin plugin_filename parameter_1 parameter_2 (...) parameter_X
```

Sin embargo es posible usarlo también entre las etiquetas habituales de Módulos para añadir opciones adicionales como condiciones o intervalo:

```
module_begin
module_plugin plugin_filename parameter_1 parameter_2 (...) parameter_X
module_interval 2
module_condition (0, 1) script.sh
module_end
```

Los parámetros a utilizar serán diferentes para cada *plugin*, por lo que será necesario remitirse a su documentación particular. Para describir el funcionamiento de uno de los *plugins* que vienen por defecto con el Agente, el *plugin* `grep_log` sirve como ejemplo para buscar coincidencias en un fichero:

```
module_plugin grep_log /var/log/syslog Syslog ssh
```

En este ejemplo, el nombre del plugin se llama, `grep_log` y buscará la expresión regular `ssh` en el fichero `/var/log/syslog` y lo guardará en un módulo llamado `Syslog`.

#### module_ping

Solamente para MS Windows®.

```
module_ping <host>
```

Este módulo realiza un ping al anfitrión o *host* especificado y devuelve 1 si está en línea.

Parámetros de configuración:

- module_ping_count x: Número de paquetes ECHO_REQUEST a enviar (1 por defecto).
- module_ping_timeout x: Tiempo de expiración en milisegundos de espera para cada respuesta (1000 por defecto).
- module_advanced_options: Opciones avanzadas para `ping.exe`.

#### module_snmpget

Solamente para MS Windows®.

```
module_snmpget
```

Este Módulo ejecuta una consulta `SNMP get` y devuelve el valor solicitado. Los parámetros de

configuración se deben especificar en líneas subsiguientes de esta manera:

- module_snmpversion [1_2c_3]: Versión de SNMP (1 por defecto).
- module_snmp_community <community>: Comunidad SNMP (*public* por defecto).
- module_snmp_agent <host>: Agente SNMP objetivo.
- module_snmp_oid <oid>: OID objetivo.
- module_advanced_options: Opciones avanzadas para `snmpget.exe`.

**module_wait_timeout**

Solamente para MS Windows®.

Tiempo de expiración que se utiliza cuando se comprueba la salida de módulos module_exec y module_plugin.

```
module_wait_timeout X
```

Valor por defecto 500 milisegundos.

Cuando se recopilan muchos datos (más de dos millones de bytes) este valor puede ser disminuido a 10 milisegundos para que los llenados y procesados de *buffer* (bloques de dieciséis mil bytes) sean atendidos rápidamente, aumentando así el desempeño del PFMS server. Se debe usar a discreción, para otros casos evítese modificar este valor predeterminado.

**module_advanced_options**

Solamente para MS Windows®.

```
module_advanced_options < parameter >
```

Para module_ping y module_snmpgetpermite utilizar parámetros adicionales.

# Configuración automática de agentes

## Introducción

En el proceso de autoconfiguración de Agentes puede establecer una serie de reglas para que se configuren de manera automática y funciona de la siguiente manera:

1. Prepare las configuraciones automáticas en su Consola Pandora FMS o Metaconsola Pandora FMS .

2. Instale los Agentes reportando hacia su Pandora FMS (si tiene una Metaconsola con el sistema de autoaprovisionamiento configurado, establezca como servidor la propia Metaconsola).
3. Pandora FMS Server recibirá un XML ( `.data` ) con los datos del Agente por primera vez.
4. Se evaluarán las reglas para determinar la configuración automática a aplicar.
5. El Agente recogerá la nueva configuración y reportará en el siguiente ciclo con la configuración actualizada.

## Creación/edición de autoconfiguración

Consola

Acceda a la administración de las configuraciones automáticas a través de Configuration → Manage agent autoconfiguration.

Metaconsola

Vaya a Centralised management → Agent management → icono de configuración automática de agentes.

Al acceder a la página de administración puede crear nuevas configuraciones automáticas presionando el botón Add new configuration definition. Deberá elegir un nombre y una descripción para su configuración automática.

Una vez creada la nueva configuración automática, puede mostrar los formularios de configuración pulsando en la sección que necesite: Rules, Agent autoconfiguration o Extra actions.

### Reglas

Para definir los Agentes sobre los que se aplicará la configuración automática, en primer lugar puede agregar reglas para identificarlos.

Despliegue el apartado de reglas dentro de su configuración automática, y seleccione Add new rule. Podrá elegir en el selector de reglas una serie de opciones, para identificar los agentes que se vayan a configurar.

- Server name: Coincidencia en nombre de servidor.
- Group name: Coincidencia en nombre de grupo.
- OS: Coincidencia en nombre de sistema operativo mediante expresiones regulares.
- Custom field: Coincidencia por clave/valor en base a un campo personalizado reportado por el Agente. Indique el nombre del campo personalizado y el valor que debe tener.
- IP range: Coincidencia por rango de direcciones IP (red), utilice la notación IP/máscara.
- Script output (> 0): Pensado para ejecutar un *script* cuyo resultado de la ejecución se evalúa como válida cuando la salida estándar sea mayor que 0.
- Llamada al *script* de reglas: Admite las siguientes macros en el campo 'argumentos' (puede elegir entre operadores `AND` y `OR` para modificar la lógica de las reglas):
    - `_agent_` : Se sustituirá por el nombre del Agente.

- `_agentalias_` : Se sustituirá por el alias del Agente.
- `_address_` : Se sustituirá por la dirección IP principal reportada por el Agente.
- `_agentgroup_` : Se sustituirá por el nombre del grupo reportado por el Agente.
- `_agentos_` : Se sustituirá por el sistema operativo del Agente.

Si no agrega ninguna regla, la configuración automática no se aplicará. Si necesita una única configuración para todos los agentes, puede utilizar la expresión regular siguiente para que coincida con cualquier *alias*: `.*`

#### Configuraciones

- Grupo del Agente: Puede mantenerlo sin cambios o forzarlo a ser uno específico.
- Grupos secundarios: Los grupos seleccionados aquí se agregarán como grupos secundarios al Agente.
- Políticas: Puede seleccionar políticas para que se apliquen automáticamente cuando el Agente alcance el servidor.
- Bloque de configuración: Agrega la configuración extra en bruto al fichero de configuración del Agente.

Si intenta acceder a la administración de configuraciones automáticas desde un nodo que pertenece a una Metaconsola, con la administración centralizada activa, la vista será únicamente de lectura.

#### Acciones extra

Desde esta sección puede asociar otras acciones a la autoconfiguración, como por ejemplo:

1. Lanzar un evento personalizado (Launch custom event).
2. Ejecutar una acción de alerta (Launch alert action).
3. Ejecutar un *script* (Launch script).

El sistema admite las siguientes macros:

- `_agent_` Se sustituirá por el nombre del Agente.
- `_agentalias_` Se sustituirá por el alias del Agente.
- `_address_` Se sustituirá por la dirección IP principal reportada por el agente.
- `_agentgroup_` Se sustituirá por el nombre del grupo reportado por el Agente.
- `_agentos_` Se sustituirá por el sistema operativo del Agente.
- `_agentid_` Se sustituye por el ID del Agente.

## Agentes Unix/Linux

## Configuración de los Agentes Unix de Pandora FMS

Las rutas y directorios fundamentales a tener en cuenta son:

- `/usr/share/pandora_agent` : Donde se instala el Agente de Pandora FMS. En los sistemas donde por políticas esto no se permita, se recomienda crear un enlace a esta ruta desde la ruta real de instalación, por ejemplo `/opt/pandora` → `/usr/share/pandora_agent`.
- `/etc/pandora/pandora_agent.conf` : Fichero principal de configuración del Agente. Los Módulos de ejecución local y *plugins* de Agente se configuran aquí.
- `/usr/local/bin/pandora_agent`: Binario ejecutable del Agente. Generalmente tiene un enlace a `/usr/bin/pandora_agent`.
- `/usr/local/bin/tentacle_client`: Binario ejecutable de Tentacle, para la transferencia de ficheros hacia el servidor. Generalmente tiene un enlace a `/usr/bin/tentacle_client`.
- `/etc/init.d/pandora_agent_daemon`: *Script* de inicio/parada/reinicio.
    - En los sistemas AIX el *daemon* es`/etc/rc.pandora_agent_daemon`.
- `/var/log/pandora/pandora_agent.log`: Fichero de texto donde se guarda la actividad del Agente de Pandora FMS, cuando el Agente se ejecuta en modo de depuración.
- `/etc/pandora/plugins`: Directorio que contiene los *plugins* de agente. Está enlazado al directorio `/usr/share/pandora_agent/plugins`.
- `/etc/pandora/collections`: Directorio que contiene las colecciones desplegadas al Agente. Está enlazado al directorio `/usr/share/pandora_agent/collections`.

## Ejecución inicial del agente Unix

Para iniciar el Agente únicamente es necesario ejecutar:

```
/etc/init.d/pandora_agent_daemon start
```

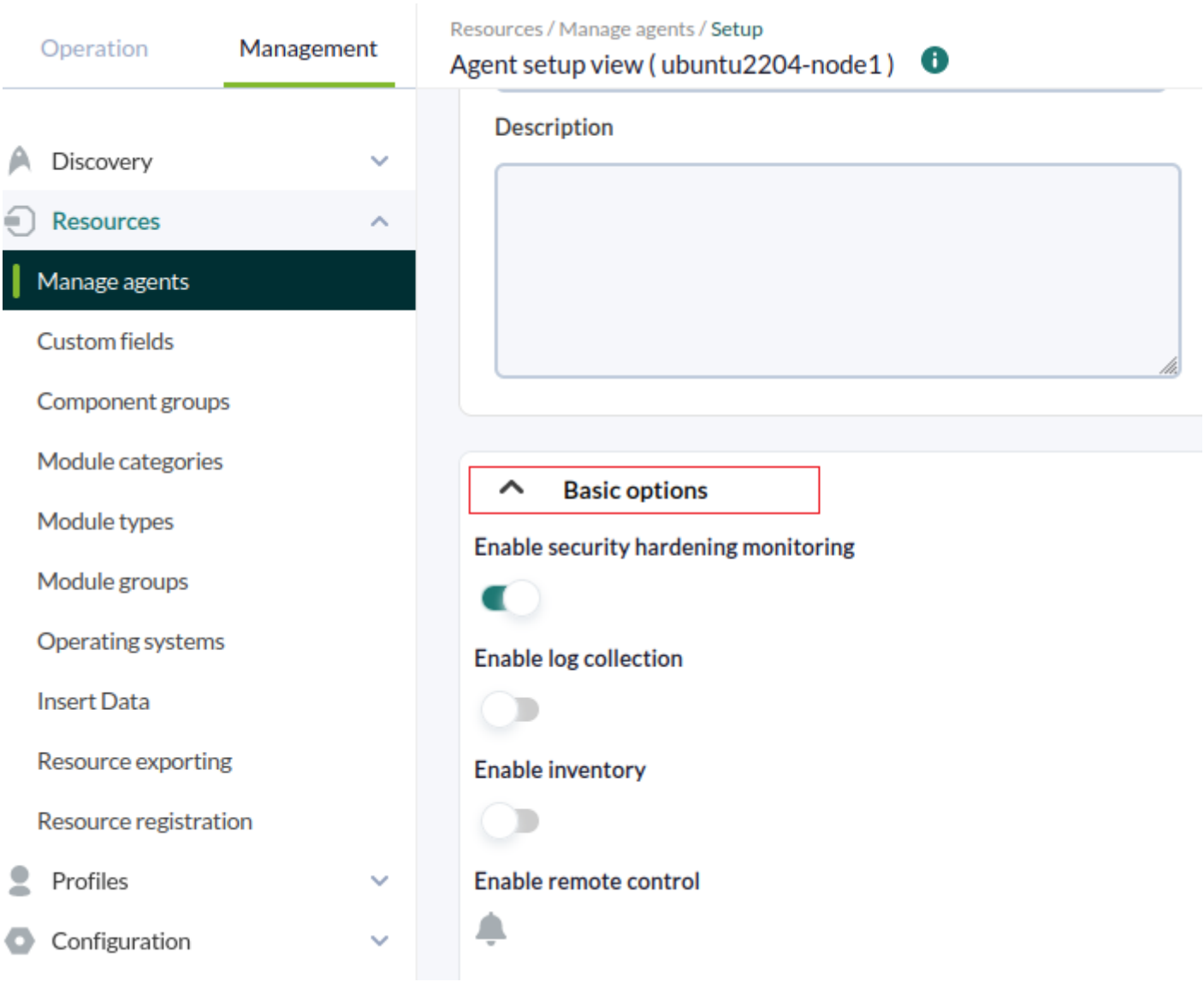
Para detener el Agente, ejecute:

```
/etc/init.d/pandora_agent_daemon stop
```

Este *script* de arranque podrá iniciar o detener el Agente de Pandora FMS, que al iniciarse quedará por defecto corriendo en el sistema como un demonio.

## Opciones básicas de agente

E Si el agente software tiene la configuración remota activada y corresponde a una versión 774 o posterior, se podrán habilitar las siguientes opciones en la sección Basic options en la configuración de agente en la Consola web.




- Enable security hardening monitoring: Habilita el *plugin* para fortalecer la seguridad en el dispositivo monitorizado. En el fichero de configuración se habilitarán las siguientes opciones:

```
#Hardening plugin for security compliance analysis. Enable to use it.
module_begin
module_plugin /usr/share/pandora_agent/plugins/pandora_hardening -t 150
module_absoluteinterval 7d
module_end
```

Donde los parámetros están configurados a un tiempo de espera de 150 segundos para su ejecución (`-t 150`) en un intervalo de 7 días (`7d`).

- Enable log collection: Esto recopilará los archivos de registro para el análisis forense y almacenará todos los registros. En el fichero de configuración se habilitarán las siguientes opciones:

```
# This is for LOG COLLECTION monitoring, different than log monitoring.
module_plugin grep_log_module /var/log/messages Syslog \.\*
```

- Enable inventory: Habilita la opción de monitorización de inventario. En el fichero de configuración se habilitarán las siguientes opciones:

```
# Plugin for inventory on the agent.
```

```
module_plugin inventory 1 cpu ram video nic hd cdrom software init_services
filesystem users route
```

### Modificar la forma en que los agentes Unix obtienen información del sistema

Existen algunos Módulos que obtienen la información de forma predefinida sin necesidad de indicar un comando con `module_exec`. Estos módulos son:

- module_procmem
- module_freedisk
- module_freepercentdisk
- module_cpuproc
- module_proc
- module_procmem
- module_cpuusage
- module_freememory
- module_freepercentmemory

Es posible modificar el funcionamiento de estos Módulos por defecto editando directamente el ejecutable del Agente (por defecto `/usr/bin/pandora_agent` ). El Agente de Pandora FMS está ubicado generalmente en `/usr/bin/pandora_agent`.

Busque la cadena `Commands to retrieve` la cual lleva el código que contiene los comandos internos. Bien puede hacer las modificaciones que necesite para adaptarlos al sistema.

```
# Commands to retrieve total memory information in kB
use constant TOTALMEMORY_CMDS => {
   linux => 'cat /proc/meminfo | grep MemTotal: | awk \'{ print $2 }\,
   solaris => 'MEM=`prtconf | grep Memory | awk \'{print $3}\'` bash -c \'echo
$(( 1024 * $MEM ))\,
   hpux => 'swapinfo -t | grep memory | awk \'{print $2}\
};

# Commands to retrieve partition information in kB
use constant PART_CMDS => {
    # total, available, mount point
    linux => 'df -P | awk \'NR> 1 {print $2, $4, $6}\,
    solaris => 'df -k | awk \'NR> 1 {print $2, $4, $6}\,
    hpux => 'df -P | awk \'NR> 1 {print $2, $4, $6}\,
    aix => 'df -kP | awk \'NR> 1 {print $2, $4, $6}\
};
```

Para cambiar cualquiera de los comandos predefinidos, simplemente edite el código para modificar el comando, pero tenga cuidado con los siguientes aspectos:

1. Verifique que las bloques `{ };` siempre terminen en punto y coma.
2. Verifique que los comandos están encerrados entre comillas simples: `' '` .
3. A su vez dentro de dichas comillas puede ser que necesite otro entrecomillado adicional con `` ` ``

(observe bien el ejemplo anterior).
4. Verifique que cualquier comilla simple que quiera usar en el comando, esté escapada previamente con el carácter \, es decir \'. Por ejemplo, este comando que normalmente sería:

```
df -P | awk 'NR> 1 {print $2, $4, $6}'
```

Debe escribirlo como:

```
df -P | awk \'NR> 1 {print $2, $4, $6}\'
```

# Agentes Windows de Pandora FMS

## Configuración del Agente para Windows de Pandora FMS

Las rutas y directorios fundamentales en las instalaciones del Agente para MS Windows® se encuentran en el propio directorio donde se haya instalado el Agente, por defecto `%ProgramFiles%`.

Los más importantes a tener en cuenta son:

`%ProgramFiles%\pandora_agent`

Donde se instala el Agente de Pandora FMS, su ejecutable y sus directorios.

`%ProgramFiles%\pandora_agent\pandora_agent.conf`

Fichero principal de configuración del Agente. Los Módulos de ejecución local y *plugins* de Agente se configuran aquí.

`%ProgramFiles%\pandora_agent\PandoraAgent.exe`

Binario ejecutable del Agente.

`%ProgramFiles%\pandora_agent\util\tentacle_client.exe`

Binario ejecutable de Tentacle, para la transferencia de ficheros hacia el servidor.

`%ProgramFiles%\pandora_agent\scripts`

*Scripts* de inicio/parada/reinicio del Agente de Pandora FMS.

`%ProgramFiles%\pandora_agent\pandora_agent.log`

Fichero de texto donde se guarda la actividad del Agente de Pandora FMS, cuando el agente se

ejecuta en modo de depuración.

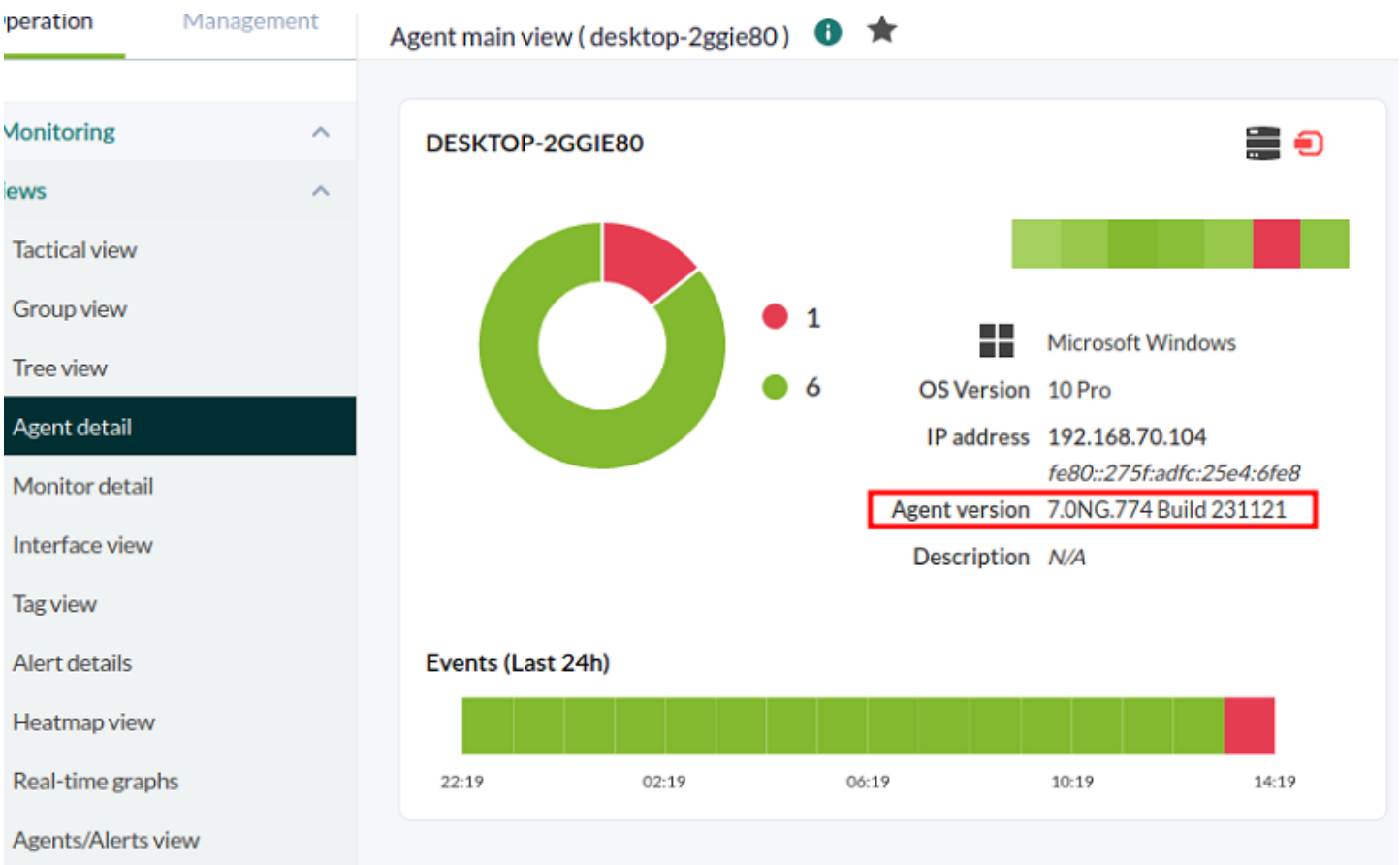
```
%ProgramFiles%\pandora_agent\util
```

Directorio que contiene los *plugins* de agente.

```
%ProgramFiles%\pandora_agent\collections
```

Directorio que contiene las colecciones del Agente.

### Opciones básicas de agente para MS Windows




Cuando el agente software tiene la configuración remota activada y la versión instalada es 774 o posterior, se podrán habilitar las siguientes opciones en la sección Basic options en la configuración de agente en la Consola web.

- Enable inventory: Habilita la opción de monitorización de inventario. En el fichero de configuración se habilitarán los siguientes parámetros:

```
module_begin
module_plugin cscript.exe //B //t:20
"%PROGRAMFILES%\Pandora_Agent\util\cpuinfo.vbs"
module_crontab * 12-15 * * 1
module_end
```

```
module_begin
module_plugin cscript.exe //B //t:20
"%PROGRAMFILES%\Pandora_Agent\util\moboinfo.vbs"
module_crontab * 12-15 * * 1
module_end

module_begin
module_plugin cscript.exe //B //t:20
"%PROGRAMFILES%\Pandora_Agent\util\diskdrives.vbs"
module_crontab * 12-15 * * 1
module_end

module_begin
module_plugin cscript.exe //B //t:20
"%PROGRAMFILES%\Pandora_Agent\util\cdromdrives.vbs"
module_crontab * 12-15 * * 1
module_end

module_begin
module_plugin cscript.exe //B //t:20
"%PROGRAMFILES%\Pandora_Agent\util\videocardinfo.vbs"
module_crontab * 12-15 * * 1
module_end

module_begin
module_plugin cscript.exe //B //t:20
"%PROGRAMFILES%\Pandora_Agent\util\ifaces.vbs"
module_crontab * 12-15 * * 1
module_end

module_begin
module_plugin cscript.exe //B //t:20
"%PROGRAMFILES%\Pandora_Agent\util\monitors.vbs"
module_crontab * 12-15 * * 1
module_end

module_begin
module_plugin cscript.exe //B //t:20
"%PROGRAMFILES%\Pandora_Agent\util\printers.vbs"
module_crontab * 12-15 * * 1
module_end
module_begin
module_plugin cscript.exe //B //t:20
"%PROGRAMFILES%\Pandora_Agent\util\raminfo.vbs"
module_crontab * 12-15 * * 1
module_end

module_begin
module_plugin cscript.exe //B //t:20
"%PROGRAMFILES%\Pandora_Agent\util\software_installed.vbs"
module_crontab * 12-15 * * 1
module_end
```

```
module_begin
module_plugin cscript.exe //B //t:20
"%PROGRAMFILES%\Pandora_Agent\util\userslogged.vbs"
module_crontab * 12-15 * * 1
module_end

module_begin
module_plugin cscript.exe //B //t:20
"%PROGRAMFILES%\Pandora_Agent\util\productkey.vbs"
module_crontab * 12-15 * * 1
module_end

module_begin
module_plugin cscript.exe //B //t:20
"%PROGRAMFILES%\Pandora_Agent\util\productID.vbs"
module_crontab * 12-15 * * 1
module_end
```

- Enable security hardening monitoring: Habilita el *plugin* para fortalecer la seguridad en el dispositivo monitorizado. En el fichero de configuración se habilitarán los siguientes parámetros:

```
#Hardening plugin for security compliance analysis. Enable to use it.
module_begin
module_plugin "%PROGRAMFILES%\Pandora_Agent\util\pandora_hardening.exe -t 150"
module_absoluteinterval 7d
module_end
```

Donde las opciones están configurados a un tiempo de espera de 150 segundos para su ejecución (`-t 150`) en un intervalo de 7 días (`7d`) como período de agente.

- Enable log collection: Esto recopilará los archivos de registro para el análisis forense y almacenará todos los registros. En el fichero de configuración se habilitarán los siguientes parámetros:

```
module_begin
module_name PandoraAgent_log
module_type generic_data_string
module_regexp C:\archivos de programa\pandora_agent\pandora_agent.log
module_description This module will return all lines from the specified logfile
module_pattern .*
module_end
```

### Opciones de seguridad de agente para MS Windows

Para el agente software PFMS versión 775 o posterior está incluido un *plugin* que viene desactivado por defecto. Para habilitarlo deben descomentar las siguientes instrucciones en el fichero de configuración:

```
# Pandora basic security check plugin for windows.
#module_begin
#module_plugin "%PROGRAMFILES%\Pandora_Agent\util\pandora_security_win.exe"
```

```
#module_end
```

Una vez se haya reiniciado el agente software se recogerán los siguientes módulos de agente:

- Antivirus instalado y funcionando, ya sea de Microsoft o terceros, y si se encuentran actualizadas sus definiciones de virus (dos módulos). En MS Windows server® esta característica no está disponible por lo que los módulos no son creados.
- Verifica si el bloqueo automático de pantalla está activo (Lock screen status), esto protege la cuenta del usuario cuando deja desatendido el equipo, sin actividad de ratón y teclado (un módulo).
- Consulta si MS Windows® se encuentra actualizado (Windows updated®) hace una semana o menos (un módulo).
- Situación del cortafuegos, activado o no (tres módulos, uno por cada perfil del *firewall*: Domain network, Private network y Public network).
- Verificación de que todas las cuentas locales cuentan con contraseña establecida (un módulo).
- Un módulo es dedicado a monitorizar si el registro de intentos fallidos de sesión se encuentra activo o no (solamente para OS instalados en inglés y en español).

## Despliegue automático de agentes software

Puede desplegar Agentes Software utilizando la central de despliegues a través del sistema Discovery, más información en este enlace.

## Auto-actualización de los Agentes Software

Utilizando las colecciones de archivos y la herramienta `pandora_update` puede proporcionar una manera de “auto-actualizar” los Agentes Software.

La herramienta `pandora_update` necesita el módulo de Perl `Digest::MD5` para funcionar. A partir de la versión 5.14 de Perl, este módulo está integrado por defecto, pero en versiones anteriores deberá instalarlo manualmente.

Funciona del siguiente modo:

1. Los Agentes reciben nuevos binarios en el directorio de entrada de las colecciones.

Ejemplo en MS Windows®:

```
%ProgramFiles%\pandora_agent\collections\fc_1\PandoraAgent.exe
```

Ejemplo en GNU/Linux®:

```
/etc/pandora/collections/fc_1/pandora_agent
```

2. El Agente ejecuta el *plugin* `pandora_update`. Este *plugin* recibe un único parámetro: el nombre corto de la colección (en este ejemplo, `fc_1`). Analizará el directorio de la colección buscando el binario del Agente (no el instalador entero), comparará el binario ubicado en la colección con el que se encuentra corriendo en ese momento y, si son diferentes, `pandora_update` detiene el Agente, reemplaza el binario y reinicia el Agente de nuevo utilizando el nuevo binario.

Para actualizar diferentes arquitecturas deberá establecer una colección diferente para cada arquitectura. Por ejemplo, si se desean actualizar agentes de Windows® de 32 y 64 bits, debe crear dos colecciones y en cada una de ellas incluir el binario `PandoraAgent.exe` correspondiente.

3. `Pandora_update` también escribe a un *log* pequeño el evento actualizado, para ser capaz de recuperar en la siguiente ejecución y avisar al usuario (mediante el uso de un Módulo `async_string` ) acerca del proceso de actualización del Agente.

Esto implica que los Módulos utilizados para completar el proceso de actualización podrán ser configurados para tener un intervalo alto.

Unix instalación estándar

```
 module_begin
 module_name Pandora_Update
 module_type async_string
 module_interval 20
 module_exec nohup /etc/pandora/plugins/pandora_update fc_1 2> /dev/null && tail
-1 nohup.out 2> /dev/null
 module_description Module to check new version of pandora agent and update
itself
 module_end
```

Unix instalación personalizada

```
 module_begin
 module_name Pandora_Update
 module_type async_string
 module_interval 20
 module_exec nohup /var/opt/PandoraFMS/etc/pandora/plugins/pandora_update fc_1
/var/opt/PandoraFMS 2> /dev/null && tail -1 nohup.out 2> /dev/null
 module_description Module to check new version of pandora agent and update
itself
 module_end
```

El comando `pandora_update` acepta como segundo
parámetro la vía del directorio de instalación de Pandora

FMS, es innecesario especificarlo si la instalación se realizó en la vía por defecto.

MS Windows®

```
 module_begin
 module_name Pandora_Update
 module_type async_string
 module_interval 20
 module_exec pandora_update.exe fc_1
 module_description Module to check new version of pandora agent and update
itself
 module_end
```

## Autocreación de Agentes y Módulos desde XML

Los Agentes pueden configurarse desde la Consola en tres modos de trabajo:

- Modo aprendizaje: Si el XML recibido del Agente Software contiene nuevos Módulos, éstos serán automáticamente creados. Este es el comportamiento por defecto.
- Modo normal: No se crearán nuevos Módulos que lleguen en el XML si no han sido declarados previamente en la consola.
- Modo autodeshabilitado: Similar al *modo aprendizaje*, en este modo, además, si todos los Módulos pasan a estado desconocido el Agente se deshabilitará automáticamente, pasando a habilitarse de nuevo si recibe nueva información.

### Datos que se actualizan de un Módulo ya existente al recibir un XML

Cuando se recibe un XML que contiene información de un Módulo ya existente, únicamente se actualiza la descripción y la información extendida, además del dato del Módulo.

Los datos GIS se actualizan siempre (si están habilitados) sin importar si el *learning mode* está desactivado.

Volver al índice de documentación de Pandora FMS