



Referencia para el desarrollo de Pandora FMS



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

Permanent link:

https://pandorafms.com/manual/!current/es/documentation/pandorafms/technical_reference/01_development_and_extension
2024/06/10 14:36



Referencia para el desarrollo de Pandora FMS

Arquitectura general del código de Pandora FMS

Para una explicación aparte y detallada acerca de la estructura de la base de datos de Pandora FMS vaya al artículo [Ingeniería de Pandora FMS](#).

Cómo hacer los enlaces compatibles

Para todos los *links*, es necesario usar la función `ui_get_full_url`. Previamente al llamado de la función es necesario incluir `functions_ui.php`.

- Cuando se necesita la URL para refrescar, por ejemplo:

```
$url_refresh = ui_get_full_url();
```

- Cuando se necesita una URL para un *path* relativo, por ejemplo:

```
$url = ui_get_full_url("/relative/path/file_script.php");
```

- En JavaScript:

```
<?php
...
$url = ui_get_full_url("/relative/path/file_script.php");
...
?>

<script type="text/javascript">
...
jQuery.post ('<?php $url; ?>',
    {
        ...
    });
...
</script>
```

- En casos especiales no es necesario usar esta función, por ejemplo los enlaces directos hacia el `index.php`:

```
echo '<form method="post"
action="index.php?param=111?param=222?param=333?param=444?param=555?param=666"> '
;
```

Puntos de entrada de ejecución de Pandora Console

Pandora Console solo dispone de unos cuantos puntos de entrada de ejecución de la aplicación web, a diferencia de otras aplicaciones web, como por ejemplo:

- Wordpress solamente tiene un punto de entrada para el *frontend* y uno de *backend*, para la administración.
- Desarrollo webs de PyMEs que normalmente cada fichero PHP es un posible punto de entrada.

Instalación

Este punto de entrada es para realizar una instalación de Pandora Console y la base de datos. Una vez finalizada la instalación Pandora Console solicita el borrado del fichero por seguridad.

```
install.php
```

Ejecución Normal

Toda interacción de los usuarios en la consola la realizan por este punto de entrada.

```
index.php
```

Peticiones AJAX

En todas las peticiones Ajax se chequean los permisos de usuario: Para hacerlas coherentes y fáciles de mantener se realizan a través de este fichero pasando por método GET o POST el parámetro page la dirección relativa del *script* a ejecutar.

```
ajax.php
```

Consola móvil

Para los terminales móviles que disponen de una pantalla sensiblemente mas pequeña que el monitor de un ordenador, Pandora FMS dispone de una versión reducida de la consola para estos dispositivos, reducida tanto en aspecto visual como simplificada en su funcionalidad.

```
mobile/index.php
```

API

Pandora FMS dispone de una **API de tipo REST** con la que terceras aplicaciones pueden interactuar a través de un canal simple por el puerto 80 y con el protocolo HTTP/HTTPS.

El fortalecimiento de este *script* es por medio de tres parámetros, validando los tres:

- Que la dirección IP del cliente que lo ejecuta esté en la lista de direcciones IP válidas configuradas de Pandora Console o que coincida con la expresión regular también guardada en el mismo lugar.
- Que le hayan pasado por parámetro la contraseña API configurada por Pandora Console.
- Que se le pase por parámetro el nombre de usuario y su contraseña, que sean válidas dichas credenciales y el usuario tenga los derechos suficientes para realizar la acción.

```
include/api.php
```

Casos especiales

Dentro de Pandora Console hay varios casos especiales para puntos de entrada, normalmente por evitar inicio de sesión o procesamientos generales que se hacen en el punto de entrada principal (`index.php` del raíz).

Extensión Tareas Cron

Esta extensión por medio de un comando `wget` en el cron puede ejecutar las tareas de cron que se le han asignado sin pedir inicio de sesión de forma interactiva. Por supuesto el conjunto de tareas está delimitado y conocido impidiendo la ejecución de código malicioso sin inicio de sesión.

```
enterprise/extensions/cron/cron.php
```

Consola visual vista externa

Este genera una página con la vista de una consola visual a pantalla completa sin ser necesario un inicio de sesión, aunque para autenticación si es necesario un *hash* generado para cada vista que se publique de esta manera.

```
operation/visual_console/public_console.php
```

Consola Networkmap Popup detalle

Un formulario emergente que muestra la vista en detalle de un agente que tiene un ítem en el Consola Networkmap. Para autenticarse usa la *session* del usuario autenticado en el Pandora Console.

```
enterprise/operation/agentes/networkmap_enterprise.popup.php
```

Gráfica de módulos Popup

Un formulario emergente que muestra una gráfica de un modulo al detalle y que permite también configurar varios parámetros de visualización. Para la autenticación utiliza los datos el usuario que ha iniciado sesión en el Pandora Console.

```
operation/agentes/stat_win.php
```

Gráficas estáticas

Las gráficas estáticas son ficheros de imagen que se generan por medio de *scripts* PHP, los datos a mostrar, si son extensos se guardan serializados en un fichero especial que procesa el *script*, este fichero tiene un tiempo de vida para evitar accesos maliciosos y ataques por DOS. Para la ejecución de este *script* no es necesario estar autenticado en Pandora FMS.

```
include/graphs/fgraph.php
```

Informes

Informe en CSV

El script genera un fichero de texto que contiene los datos CSV, para la autenticación usa los datos del usuario conectado.

```
enterprise/operation/reporting/reporting_viewer_csv.php
```

Informe en PDF

El *script* genera un fichero de texto que contiene los datos PDF, para la autenticación usa los datos del usuario conectado.

```
enterprise/operation/reporting/reporting_viewer_pdf.php
```

Eventos

Eventos sonoros Popup

Ventana de Popup que está revisando periódicamente si existe algún evento para mostrarlo de forma sonora y visual. La autenticación es por medio de los datos del usuario que ha iniciado sesión en Pandora Console.

```
operation/events/sound_events.php
```

Eventos en CSV

El *script* genera un fichero de texto que contiene los datos CSV, para la autenticación usa los datos del usuario conectado.

```
operation/events/export_csv.php
```

Eventos en RSS

El *script* genera un fichero de texto que contiene los datos CSV, para la autenticación usa los datos del usuario conectado con un hash que va como parámetro.

```
operation/events/events_rss.php
```

Funciones básicas para obtener estado de agente, módulo y grupos

Criterio para Estados y codificación en la base de datos

Descripción de estados del agente:

- Crítico (color rojo): 1 o más módulos en estado crítico.
- Advertencia (color amarillo): 1 o más módulos en estado advertencia y ninguno en estado crítico.
- Desconocido (color gris): 1 o más módulos en estado desconocido y ninguno en estado crítico y advertencia.
- Normal (color verde): todos los módulos están en estado normal.

Código interno del estado en la BBDD:

- Crítica: 1
- Advertencia: 2
- Desconocida: 3
- Normal: 0

Agentes

Funciones de Estado

Estas funciones devuelven el número de módulos y alertas disparadas de un agente aplicando un filtro de forma opcional.

Todas las funciones tienen la opción `filter` que fue añadido para hacer la función más flexible. El contenido del filtro se añade al final de la consulta SQL en todas las funciones. Con el filtro puede

añadir cláusulas SQL específicas para crear filtros usando las tablas: tagente_estado, tagente y tagente_modulo.

```
agents_monitor_critical($id_agent, $filter='')
```

Devuelve el número de módulos en estado crítico para el agente.

```
agents_monitor_warning ($id_agent, $filter = '')
```

Devuelve el número de módulos en estado advertencia para el agente.

```
agents_monitor_unknown ($id_agent, $filter = '')
```

Devuelve el número de módulos en estado desconocido para el agente.

```
agents_monitor_ok ($id_agent, $filter = '')
```

Devuelve el número de módulos en estado normal para el agente.

```
agents_get_alerts_fired ($id_agent, $filter = '')
```

Devuelve el número de alertas disparadas para el agente.

Funciones auxiliares

Estas funciones realizan tareas relacionadas con los agentes para algunas vistas:

```
agents_tree_view_alert_img ($alert_fired)
```

Devuelve la ruta de la imagen de alerta usada en la vista Tree View.

```
agetns_tree_view_status_img ($critical, $warning, $unknown)
```

Devuelve la ruta de la imagen para el estado del agente usada en al vista Tree View.

Grupos

Estas funciones devuelven las estadísticas de agentes y módulos para los grupos de agentes definidos en Pandora FMS.

Las funciones del servidor y la consola deben usar las mismas consultas SQL para asegurar que el resultado es calculado de la misma forma.

Funciones del servidor

```
pandora_group_statistics
```

Esta función calcula las estadísticas de grupo si el parámetro Use realtime statistics está desactivado.

Funciones de la consola

Grupos

Las funciones de la consola calculan las estadísticas en base a una matriz de grupos de agentes.

Estas funciones no usan los agentes o módulos deshabilitados:

```
groups_agent_unknown ($group_array)
```

Devuelve en número de agentes en estado desconocido para los grupos dados.

```
groups_agent_ok ($group_array)
```

Devuelve en número de agentes en estado normal para los grupos dados.

```
groups_agent_critical ($group_array)
```

Devuelve en número de agentes en estado crítico para los grupos dados.

```
groups_agent_warning ($group_array)
```

Devuelve en número de agentes en estado advertencia para los grupos dados.

Módulos

Estas funciones calculan las estadísticas para los módulos. No usan los módulos o agentes deshabilitados:

```
groups_monitor_not_init ($group_array)
```

Devuelve el número de módulos con estado no inicializado para los grupos dados.

```
groups_monitor_ok ($group_array)
```

Devuelve el número de módulos con estado normal para los grupos dados.

```
groups_monitor_critical ($group_array)
```

Devuelve el número de módulos con estado crítico para los grupos dados.

```
groups_monitor_warning ($group_array)
```

Devuelve el número de módulos con estado advertencia para los grupos dados.

```
groups_monitor_unknown ($group_array)
```

Devuelve el número de módulos con estado desconocido para los grupos dados.

```
groups_monitor_alerts ($group_array)
```

Devuelve el número de módulos con alertas para los grupos dados.

```
groups_monitor_fired_alerts ($group_array)
```

Devuelve el número de módulos con alertas disparadas para los grupos dados.

Módulos

Estas funciones devuelve las estadísticas en base al nombre del módulo. No tienen en cuenta los agentes y módulos deshabilitados.

```
modules_agents_unknown ($module_name)
```

Devuelve el número de agentes en estado desconocido que tienen un módulo con el nombre dado.

```
modules_agents_ok ($module_name)
```

Devuelve el número de agentes en estado normal que tienen un módulo con el nombre dado.

```
modules_agents_critical ($module_name)
```

Devuelve el número de agentes en estado crítico que tienen un módulo con el nombre dado.

```
modules_agents_warning ($module_name)
```

Devuelve el número de agentes en estado advertencia que tienen un módulo con el nombre dado.

Estas funciones devuelven las estadísticas usando como filtro el grupo de módulo. No tiene en cuenta agentes o módulos deshabilitados:

```
modules_group_agent_unknown ($module_group)
```

Devuelve el número de agentes con estado desconocido que tienen módulos que pertenecen al grupo de módulos dado.

```
modules_group_agent_ok ($module_group)
```

Devuelve el número de agentes con estado normal que tienen módulos que pertenecen al grupo de módulos dado.

```
modules_group_agent_critical ($module_group)
```

Devuelve el número de agentes con estado crítico que tienen módulos que pertenecen al grupo de módulos dado.

```
modules_group_agent_warning ($module_group)
```

Devuelve el número de agentes con estado advertencia que tienen módulos que pertenecen al grupo de módulos dado.

Políticas

Estas funciones devuelven el número de agentes para cada estado y política dada. No usan los agentes y módulos deshabilitados para calcular el resultado.

```
policies_agents_critical ($id_policy)
```

Devuelve el número de agentes con estado crítico que pertenecen a una política dada.

```
policies_agents_ok ($id_policy)
```

Devuelve el número de agentes con estado normal que pertenecen a una política dada.

```
policies_agents_unknown ($id_policy)
```

Devuelve el número de agentes con estado desconocido que pertenecen a una política dada.

```
policies_agents_warning ($id_policy)
```

Devuelve el número de agentes con estado advertencia que pertenecen a una política dada.

OS

Estas funciones calculan las estadísticas para los agentes en base los Sistemas Operativos a los que pertenecen. No usan agentes o módulos deshabilitados.

```
os_agents_critical ($id_os)
```

Devuelve el número de agentes en estado crítico que pertenecen al Sistema Operativo dado.

```
os_agents_ok($id_os)
```

Devuelve el número de agentes en estado normal que pertenecen al Sistema Operativo dado.

```
os_agents_warning ($id_os)
```

Devuelve el número de agentes en estado advertencia que pertenecen al Sistema Operativo dado.

```
os_agents_unknown ($id_os)
```

Devuelve el número de agentes en estado desconocido que pertenecen al Sistema Operativo dado.

Desarrollo y ampliación

La mayoría de las ampliaciones han sido descritas como anexos independientes, especializados para creación de *plugin* de servidor, *plugins* de agente Unix, y extensiones de la Consola web. En este capítulo se describe como colaborar en Pandora FMS y como compilar el agente de Windows desde las fuentes.

Colaborar con el proyecto Pandora FMS

Este proyecto lo mantienen desarrolladores voluntarios con sus colaboraciones. Nuevos desarrolladores, redactores de documentación, o personas que quieran ayudar será siempre bienvenidas. Una buena manera de empezar es suscribiéndose a nuestras listas de correo y/o [al foro](#).

Bugs / Fallos

Informar acerca de posibles errores nos ayuda a mejorar Pandora FMS. Por favor, antes de enviar un informe de error, [revise nuestra base de datos de bugs](#).

Listas de correo

Las listas de correo son una excelente manera de mantenerse al día por correo electrónico. Tenemos una lista de correo pública para usuarios y anuncios (con poco tráfico) y una lista de correo de desarrollo para discusiones técnicas y notificaciones (a veces diarias) del desarrollo mediante el sistema automático de notificación de nuestro GIT (Sistema de control de versión de código).

Compilación del agente de Windows desde el código

Obtener la última versión del código

Para obtener la última versión del código es imprescindible hacerlo descargando las fuentes del repositorio de código [GitHub](#), donde tenemos publicado el repositorio oficial de Pandora FMS.

API de Pandora FMS

Existe una API externa de Pandora FMS para poder enlazar terceras aplicaciones con Pandora FMS, tanto para obtener información de Pandora FMS como para introducir información dentro de Pandora FMS. Toda esta documentación está en el [apéndice de API Externa de Pandora FMS](#).

Formato de los ficheros de datos XML

Conocer el formato de los XML de datos de Pandora FMS puede ayudar a mejorar los *plugins* de agente, crear agentes personalizados o simplemente enviar ficheros XML personalizados al servidor de datos de Pandora FMS.

Como cualquier documento XML, el fichero de datos debe comenzar con una declaración XML:

```
<?xml version='1.0' encoding='UTF-8'?>
```

A continuación viene el elemento `agent_data` que define al agente que envía los datos. Soporta los siguientes atributos:

- *description*: Descripción del agente.
- *group*: Nombre del grupo al que pertenece el agente (debe existir en la base de datos de Pandora FMS). Si se deja vacío y no hay un grupo configurado por defecto en el servidor, el agente no se creará.
- *os_name*: Nombre del sistema operativo en el que corre el agente (debe existir en la base de datos de Pandora FMS).
- *os_version*: Cadena libre que describe la versión del sistema operativo.
- *interval*: Intervalo del agente (en segundos).
- *version*: Cadena con la versión del agente.
- *timestamp*: Marca de tiempo que indica cuándo se generó el XML (YYYY/MM/DD HH:MM:SS).
- *agent_name*: Nombre del agente.
- *timezone_offset*: Desplazamiento que se suma a la marca de tiempo (en horas). Útil si se trabaja con UTC.
- *address*: Dirección IP del agente (o FQN).
- *parent_agent_name*: Nombre del padre del agente.
- *agent_alias*: Alias del agente.
- *agent_mode*: Modo de trabajo del agente (0: Normal mode, 1: Learning mode, 2: Autodisable mode) .
- *secondary_groups*: Grupos secundarios añadidos al agente.
- *custom_id*: ID personalizado del agente.
- *url_address*: URL de acceso al agente.

A continuación se necesita un elemento `module` por cada módulo, y se pueden anidar los siguientes elementos para definir el módulo:

- *name*: Nombre del módulo.
- *description*: Descripción del módulo.
- *tags*: Etiquetas asociadas al módulo.
- *type*: Tipo del módulo (debe existir en la base de datos de Pandora FMS).
- *data*: Dato del módulo.
- *max*: Valor máximo del módulo.
- *min*: Valor mínimo del módulo.
- *post_process*: Valor de pos-procesado.
- *module_interval*: Intervalo del módulo (intervalo en segundos / intervalo del agente).
- *min_critical*: Valor mínimo para el estado crítico.
- *max_critical*: Valor máximo para el estado crítico.
- *min_warning*: Valor mínimo para el estado de alerta.
- *max_warning*: Valor máximo para el estado de alerta.
- *disabled*: Deshabilita (0) o habilita el módulo. Los módulos deshabilitados no se procesan.
- *min_ff_event*: Umbral de FF (ver).
- *status*: Estado del módulo (NORMAL, WARNING or CRITICAL). Los límites de los estados crítico y de alerta se ignoran si se especifica el estado.
- *datalist*: Envía el dato del módulo en formato datalist (una entrada en base de datos por cada uno de los valores recibidos) [0/1].
- *unit*: Unidad del módulo. Soporta la macro `_Timeticks_` para transformar un dato en formato timeticks a dd/hh/mm/ss.
- *timestamp*: Establece un timestamp en el dato recibido del módulo.

- `module_group`: Grupo de módulos al que será añadido el módulo.
- `custom_id`: Custom ID del módulo.
- `str_warning`: Umbral de warning para módulos string.
- `str_critical`: Umbral de crítico para módulos string.
- `critical_instructions`: Critical instructions del módulo.
- `warning_instructions`: Warning instructions del módulo.
- `unknown_instructions`: Unknown instructions del módulo.
- `critical_inverse`: Activa el Inverse interval en el umbral crítico del módulo. [0/1].
- `warning_inverse`: Activa el Inverse interval en el umbral warning del módulo. [0/1].
- `quiet`: Activa el modo Quiet del módulo [0/1].
- `module_ff_interval`: Especifica un valor de FF Interval del módulo.
- `alert_template`: Asocia una plantilla de alerta al módulo.
- `crontab`: Especifica un crontab en el módulo.
- `min_ff_event_normal`: Valor de FF threshold en el cambio de estado a NORMAL.
- `min_ff_event_warning`: Valor de FF threshold en el cambio de estado a WARNING.
- `min_ff_event_critical`: Valor de FF threshold en el cambio de estado a CRITICAL.
- `ff_timeout`: Valor de FlipFlop timeout.
- `each_ff`: Habilita la opción "Change each status".
- `module_parent`: Nombre del módulo en el mismo agente que será el padre de este módulo.
- `ff_type`: Activa el Keep counters del FF threshold. [0/1].
- `min_warning_forced`: Fuerza `min_warning` al nuevo valor indicado aunque el módulo exista, tiene prioridad sobre `min_warning`.
- `max_warning_forced`: Fuerza `max_warning` al nuevo valor indicado aunque el módulo exista, tiene prioridad sobre `max_warning`.
- `min_critical_forced`: Fuerza `min_critical` al nuevo valor indicado aunque el módulo exista, tiene prioridad sobre `min_critical`.
- `max_critical_forced`: Fuerza `max_critical` al nuevo valor indicado aunque el módulo exista, tiene prioridad sobre `max_critical`.
- `str_warning_forced`: Fuerza `str_warning` al nuevo valor indicado aunque el módulo exista, tiene prioridad sobre `str_warning`.
- `str_critical_forced`: Fuerza `str_critical` al nuevo valor indicado aunque el módulo exista, tiene prioridad sobre `str_critical`.

Estos *tokens* solo funcionarán para plugins del dataserver.

Cualquier otro elemento se salvará como información extendida del módulo en la base de datos de Pandora FMS. Un módulo deberá tener al menos un elemento `name`, `type` y `data`, por ejemplo:

```
<module>
  <name>CPU</name>
  <description>CPU usage percentage</description>
  <type>generic_data</type>
  <data>21</data>
</module>
```

Puede haber cualquier número de elementos en un fichero de datos XML antes de cerrar la etiqueta `agent_data`.

Hay un caso especial de XML multiitem, basado en una lista de datos. Esto se aplica únicamente a

datos de tipo cadena. El XML sería algo parecido a esto:

```
<module>
  <type>async_string</type>
  <datalist>
    <data><value><![CDATA[xxxxx]]></value></data>
    <data><value><![CDATA[yyyyy]]></value></data>
    <data><value><![CDATA[zzzzz]]></value></data>
  </datalist>
</module>
```

Se puede especificar una marca de tiempo para cada valor:

```
<module>
  <type>async_string</type>
  <datalist>
    <data>
      <value><![CDATA[xxxxx]]></value>
      <timestamp>1970-01-01 00:00:00</timestamp>
    </data>
    <data>
      <value><![CDATA[yyyyy]]></value>
      <timestamp>1970-01-01 00:00:01</timestamp>
    </data>
    <data>
      <value><![CDATA[zzzzz]]></value>
      <timestamp>1970-01-01 00:00:02</timestamp>
    </data>
  </datalist>
</module>
```

[Volver al Índice de Documentación Pandora FMS](#)