



Référence pour le développement de Pandora FMS



URL: <https://pandorafms.com/manual/!778/>
Permanent link: https://pandorafms.com/manual/!778/fr/documentation/pandorafms/technical_reference/01_development_and_extension
2024/12/03 19:30



Référence pour le développement de Pandora FMS

Architecture générale du code Pandora FMS

Pour une explication séparée et détaillée de la structure de la base de données de Pandora FMS, voir l'article [Pandora FMS Engineering](#).

Comment rendre les liens compatibles

Pour tous les liens, il faut utiliser la fonction `ui_get_full_url`. Avant d'appeler la fonction, il faut inclure `functions_ui.php`.

- Lorsque l'URL est nécessaire pour le rafraîchissement, par exemple :

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

- Lorsqu'une URL est nécessaire pour un chemin relatif, par exemple :

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

- JavaScript :

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

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

- Dans certains cas particuliers, il n'est pas nécessaire d'utiliser cette fonction, par exemple pour les liens directs vers `index.php` :

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

Points d'entrée de la console Web de PFMS

La console Web PFMS ne comporte que quelques points d'entrée pour l'exécution d'applications Web, contrairement à d'autres applications Web telles que :

- WordPress n'a qu'un seul point d'entrée pour *frontend* et un seul pour *backend* pour l'administration.
- Le développement web des PME qui normalement chaque fichier PHP est un point d'entrée possible.

Installation

Ce point d'entrée permet de procéder à l'installation de la console Web de PFMS et de la base de données. Une fois l'installation terminée, la console Web PFMS vous invite à supprimer le fichier pour des raisons de sécurité.

```
install.php
```

Exécution normale

Toutes les interactions de l'utilisateur avec la console se font par l'intermédiaire de ce point d'entrée.

```
index.php
```

Requêtes AJAX

Dans toutes les requêtes AJAX, les autorisations de l'utilisateur sont vérifiées : pour les rendre cohérentes et faciles à maintenir, elles sont effectuées par l'intermédiaire de ce fichier en passant par la méthode GET ou POST le paramètre page l'adresse relative du *script* à exécuter.

```
ajax.php
```

Console mobile

Pour les terminaux mobiles dont l'écran est nettement plus petit que celui d'un ordinateur, Pandora FMS dispose d'une version réduite de la console pour ces appareils, dont l'aspect visuel est réduit et les fonctionnalités simplifiées.

```
mobile/index.php
```

API

Pandora FMS dispose d'une [API](#) avec laquelle les applications tierces peuvent interagir par un simple canal via le port 80 et le protocole HTTP/HTTPS.

Le renforcement de ce *script* se fait au moyen de deux paramètres, en les validant tous les deux :

- L'adresse IP du client en cours d'exécution figure dans la liste des adresses IP valides configurées de la console Web PFMS ou correspond à l'expression régulière également stockée au même endroit.
- L'API jeton configurée par la console Web PFMS a été transmise en tant que paramètre.

```
include/api.php
```

Cas particuliers

La console Web PFMS comporte plusieurs cas particuliers de points d'entrée, généralement pour éviter la connexion ou le traitement général qui est effectué au point d'entrée principal (`index.php` à la racine).

Extension des tâches Cron

Cette extension, au moyen d'une commande `wget` dans le `cron`, peut exécuter les tâches `cron` qui lui sont assignées sans demander de connexion interactive. Bien entendu, l'ensemble des tâches est délimité et connu afin d'éviter l'exécution de code malveillant sans connexion.

```
entreprise/extensions/cron/cron.php
```

Vue externe de la console visuelle

Cela génère une page avec une console visuelle en plein écran sans nécessiter de connexion, bien que pour l'authentification il faille générer un *hash* pour chaque vue publiée de cette manière.

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

Console Networkmap Popup détaillée

Un formulaire contextuel qui affiche la vue détaillée d'un agent ayant un élément dans la console Networkmap. Pour s'authentifier, il utilise le *session* de l'utilisateur authentifié dans la console Web PFMS.

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

Graphique du module popup

Un formulaire pop-up qui affiche un graphique d'un module en détail et permet également de configurer divers paramètres d'affichage. Pour l'authentification, il utilise les données de l'utilisateur connecté à la console Web PFMS.

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

Graphiques statiques

Les graphiques statiques sont des fichiers images générés au moyen de *scripts* PHP, les données à afficher, si elles sont nombreuses, sont sauvegardées en série dans un fichier spécial qui traite le *script*, ce fichier a une durée de vie afin d'éviter les accès malveillants et les attaques par déni de service. Pour l'exécution de ce *script*, il n'est pas nécessaire d'être authentifié dans Pandora FMS.

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

Rapports

Rapport CSV

Le *script* génère un fichier texte contenant les données CSV. Pour l'authentification, il utilise les données de l'utilisateur connecté.

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

Rapport PDF

Le *script* génère un fichier texte contenant les données PDF. Pour l'authentification, il utilise les données de l'utilisateur connecté.

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

Événements

Événements sonores *Popup*

Fenêtre *popup* qui vérifie périodiquement l'existence d'un événement et qui s'affiche de manière sonore et visuelle. L'authentification se fait au moyen des données de l'utilisateur connecté à la console Web PFMS.

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

Événements CSV

Le *script* génère un fichier texte contenant les données CSV. Pour l'authentification, il utilise les données de l'utilisateur connecté.

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

Événements RSS

Le *script* génère un fichier texte contenant les données CSV, pour l'authentification il utilise les données de l'utilisateur connecté avec un *hash* comme paramètre.

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

Fonctions de base pour obtenir l'état de l'agent, du module et des groupes

Critères pour les états et la codification dans la base de données

Description des états de l'agent :

- Critique (couleur rouge) : 1 ou plusieurs modules dans un état critique.
- Avertissement (couleur jaune) : 1 ou plusieurs modules en état d'avertissement et aucun en état critique.
- Inconnu (gris) : 1 ou plusieurs modules dans un état inconnu et aucun dans un état critique ou d'avertissement.
- Normal (couleur verte) : Tous les modules sont en état normal.

Code interne de l'Etat dans le base de données :

- Critique : 1
- Avertissement : 2
- Inconnu : 3
- Normal : 0

Agents

Fonctions de l'État

Ces fonctions renvoient le nombre de modules et d'alertes déclenchés par un agent en appliquant éventuellement un filtre.

Toutes les fonctions disposent de l'option filter qui a été ajoutée pour rendre la fonction plus flexible. Le contenu du filtre est ajouté à la fin de la requête SQL dans toutes les fonctions. Avec le filtre, vous pouvez ajouter des clauses SQL spécifiques pour créer des filtres en utilisant les tables : tagente_estado, tagente et tagente_modulo.

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

Il renvoie le nombre de modules en état critique pour l'agent.

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

Il renvoie le nombre de modules en état d'alerte pour l'agent.

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

Il renvoie le nombre de modules dont l'état est inconnu pour l'agent.

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

Il renvoie le nombre de modules en état normal pour l'agent.

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

Il renvoie le nombre d'alertes déclenchées pour l'agent.

Fonctions auxiliaires

Ces fonctions exécutent des tâches liées à l'agent pour certaines vues :

```
agents_tree_view_alert_img ($alert_fired)
```

Il renvoie le chemin d'accès à l'image d'alerte utilisée dans la vue Tree View.

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

Il renvoie le chemin d'accès à l'image de l'état de l'agent utilisé dans l'arborescence.

Groupes

Ces fonctions renvoient les statistiques des agents et des modules pour les groupes d'agents définis dans Pandora FMS.

Les fonctions serveur et console doivent utiliser les mêmes requêtes SQL pour s'assurer que le résultat est calculé de la même manière.

Fonctions du serveur

```
pandora_group_statistics
```

Cette fonction calcule les statistiques de groupe si le paramètre Use realtime statistis est désactivé.

Fonctions de la console

Groupes

Les fonctions de la console calculent des statistiques basées sur une matrice de groupes d'agents.

Ces fonctions n'utilisent pas les agents ou modules désactivés :

```
groups_agent_unknown ($group_array)
```

Il renvoie le nombre d'agents en état inconnu pour les groupes donnés.

```
groups_agent_ok ($group_array)
```

Il renvoie le nombre d'agents en état normal pour les groupes donnés.

```
groups_agent_critical ($group_array)
```

Il renvoie le nombre d'agents en état critique pour les groupes donnés.

```
groups_agent_warning ($group_array)
```

Il renvoie le nombre d'agents en état d'alerte pour les groupes donnés.

Modules

Ces fonctions calculent des statistiques pour les modules. Elles n'utilisent pas les modules ou agents désactivés :

```
groups_monitor_not_init ($group_array)
```

Il renvoie le nombre de modules dont l'état n'est pas initialisé pour les groupes donnés.

```
groups_monitor_ok ($group_array)
```

Il renvoie le nombre de modules ayant un état normal pour les groupes donnés.

```
groups_monitor_critical ($group_array)
```

Il renvoie le nombre de modules ayant un état critique pour les groupes donnés.

```
groups_monitor_warning ($group_array)
```

Il renvoie le nombre de modules ayant un état d'avertissement pour les groupes donnés.

```
groups_monitor_unknown ($group_array)
```

Il renvoie le nombre de modules dont l'état est inconnu pour les groupes donnés.

```
groups_monitor_alerts ($group_array)
```

Il renvoie le nombre de modules avec des alertes pour les groupes donnés.

```
groups_monitor_fired_alerts ($group_array)
```

Il renvoie le nombre de modules ayant déclenché des alertes pour les groupes donnés.

Modules

Ces fonctions renvoient des statistiques basées sur le nom du module. Elles ne tiennent pas compte des agents et des modules désactivés.

```
modules_agents_unknown ($module_name)
```

Il renvoie le nombre d'agents en état inconnu qui ont un module avec le nom donné.

```
modules_agents_ok ($module_name)
```

Il renvoie le nombre d'agents en état normal qui ont un module avec le nom donné.

```
modules_agents_critical ($module_name)
```

Il renvoie le nombre d'agents en état critique qui ont un module portant le nom donné.

```
modules_agents_warning ($module_name)
```

Il renvoie le nombre d'agents en état d'alerte qui ont un module portant le nom donné.

Ces fonctions renvoient des statistiques en utilisant le groupe de modules comme filtre. Elles ne tiennent pas compte des agents ou des modules désactivés :

```
modules_group_agent_unknown ($module_group)
```

Il renvoie le nombre d'agents dont l'état est inconnu et dont les modules appartiennent au groupe de modules donné.

```
modules_group_agent_ok ($module_group)
```

Il renvoie le nombre d'agents ayant un état normal et dont les modules appartiennent au groupe de modules donné.

```
modules_group_agent_critical ($module_group)
```

Il renvoie le nombre d'agents ayant un état critique et dont les modules appartiennent au groupe de modules donné.

```
modules_group_agent_warning ($module_group)
```

Il renvoie le nombre d'agents ayant un état d'alerte et dont les modules appartiennent au groupe de modules donné.

Politiques

Ces fonctions renvoient le nombre d'agents pour chaque état et politique donnés. Elles n'utilisent pas les agents et modules désactivés pour calculer le résultat.

```
policies_agents_critical ($id_policy)
```

Il renvoie le nombre d'agents ayant un état critique et appartenant à une politique donnée.

```
policies_agents_ok ($id_policy)
```

Il renvoie le nombre d'agents ayant un état normal et appartenant à une politique donnée.

```
policies_agents_unknown ($id_policy)
```

Il renvoie le nombre d'agents dont l'état est inconnu et qui appartiennent à une politique donnée.

```
policies_agents_warning ($id_policy)
```

Il renvoie le nombre d'agents ayant un état d'avertissement et appartenant à une politique donnée.

OS

Ces fonctions calculent les statistiques des agents en fonction des systèmes d'exploitation auxquels ils appartiennent. Elles n'utilisent pas d'agents ou de modules désactivés.

```
os_agents_critical ($id_os)
```

Il renvoie le nombre d'agents en état critique qui appartiennent au système d'exploitation donné.

```
os_agents_ok($id_os)
```

Il renvoie le nombre d'agents en état normal qui appartiennent au système d'exploitation donné.

```
os_agents_warning ($id_os)
```

Il renvoie le nombre d'agents en état d'alerte qui appartiennent au système d'exploitation donné.

```
os_agents_unknown ($id_os)
```

Il renvoie le nombre d'agents dans un état inconnu qui appartiennent à l'OS donné.

Développement et extension

La plupart des extensions ont été décrites comme des annexes indépendantes, spécialisées dans la création d'extensions de serveur *plugins*, d'agent Unix® *plugins* et de console web. Ce chapitre décrit comment collaborer dans Pandora FMS et comment compiler l'agent MS Windows® à partir des sources.

Collaborer avec le projet Pandora FMS

Ce projet est maintenu par des développeurs bénévoles grâce à leurs contributions. Les nouveaux développeurs, les rédacteurs de documentation, ou les personnes qui veulent aider sont toujours les bienvenus. Une bonne façon de commencer est de s'inscrire à nos listes de diffusion et/ou au [forum](#).

Bugs / Bogues

Signaler des bogues nous aide à améliorer Pandora FMS. Avant de soumettre un rapport de bogue, veuillez [vérifier notre base de données de bogues](#).

Listes de diffusion

Les listes de diffusion sont un excellent moyen de se tenir informé par courrier électronique. Nous avons une liste de diffusion publique pour les utilisateurs et les annonces (avec un faible trafic) et une liste de diffusion de développement pour les discussions techniques et les notifications de développement (parfois quotidiennes) via le système de notification automatique de notre GIT (système de contrôle de la version du code).

Compilation de l'agent Windows à partir du code

Obtenir la dernière version du code

Pour obtenir la dernière version du code, il est essentiel de télécharger les sources à partir du dépôt de code [GitHub](#), où le dépôt officiel de Pandora FMS est publié.

Pandora FMS API

Il existe une API externe de Pandora FMS pour pouvoir lier des applications tierces à Pandora FMS, à la fois pour obtenir des informations de Pandora FMS et pour introduire des informations à l'intérieur de Pandora FMS. Toute cette documentation se trouve dans l'annexe [Pandora FMS External API](#).

Format des fichiers de données XML

Connaître le format XML des données de Pandora FMS peut aider à améliorer l'agent *plugins*, à créer des agents personnalisés ou simplement à envoyer des fichiers XML personnalisés au serveur de données de Pandora FMS.

Comme tout document XML, le fichier de données doit commencer par une déclaration XML :

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

Vient ensuite l'élément `agent_data` qui définit l'agent qui envoie les données. Il prend en charge les attributs suivants :

- *description* : La description de l'agent.
- *group* : Le nom du groupe auquel l'agent appartient (il doit exister dans la base de données de Pandora FMS). S'il est laissé vide et qu'aucun groupe par défaut n'est configuré sur le serveur, l'agent ne sera pas créé.
- *os_name* : Le nom du système d'exploitation sur lequel l'agent fonctionne (il doit exister dans la base de données de Pandora FMS).
- *os_version* : La chaîne libre décrivant la version du système d'exploitation.
- *interval* : L'intervalle d'agent (en secondes).
- *version* : La chaîne avec la version de l'agent.
- *timestamp* : L'horodatage indiquant quand le XML a été généré (YYYY/MM/DD HH:MM:SS).
- *agent_name* : Le nom de l'agent.
- *timezone_offset* : Le décalage à ajouter à l'horodatage (en heures). Utile pour travailler avec l'UTC.
- *address* : L'adresse IP de l'agent (ou FQDN).
- *parent_agent_name* : Le nom du père de l'agent.
- *agent_alias* : L'alias de l'agent.
- *agent_mode* : Le fonctionnement de l'agent (0: Normal mode, 1: Learning mode, 2: Autodisable mode).
- *secondary_groups* : Les groupes secondaires ajoutés à l'agent.
- *custom_id* : L'identifiant personnalisé de l'agent.
- *url_address* : L'URL d'accès à l'agent. Un élément module est alors nécessaire pour chaque module, et les éléments suivants peuvent être imbriqués pour définir le module :
 - *name* : Le nom du module.
 - *description* : La description du module.
 - *tags* : Les tags associés au module.
 - *type* : Le type de module (doit exister dans la base de données de Pandora FMS).
 - *data* : Les données du module.
 - *max* : La valeur maximale du module.
 - *min* : La valeur minimale du module.
 - *post_process* : La valeur de post-traitement.
 - *module_interval* : L'intervalle du module (intervalle en secondes / intervalle de l'agent).
 - *min_critical* : La valeur minimale pour l'état d'urgence.
 - *max_critical* : La valeur maximale pour l'état d'urgence.
 - *min_warning* : La valeur minimale pour l'état d'alerte.
 - *max_warning* : La valeur maximale pour l'état d'alerte.
 - *disabled* : Il désactive (0) ou active le module. Les modules désactivés ne sont pas traités.
 - *min_ff_event* : Le seuil FF.
 - *status* : L'état du module (NORMAL, WARNING ou CRITICAL). Les limites des états d'urgence et d'alerte sont ignorées si l'état est spécifié.
- *datalist* Il envoie les données du module sous forme de datalist (une entrée dans la base de données pour chacune des valeurs reçues). [0/1].
- *unit* : L'unité de module. Supporte la macro `_timeticks_` pour transformer une donnée au format *timeticks* en dd/hh/mm/ss.
- *timestamp* : Il fixe un horodatage sur les données reçues du module.
- *module_group* : Le groupe de modules auquel le module sera ajouté.

- `custom_id` : Le custom ID du module.
- `str_warning`: Le seuil *warning* pour les modules *string*.
- `str_critical` : Le seuil de *critical* pour les modules *string*.
- `critical_instructions` : L'instructions relatives aux modules critiques.
- `warning_instructions` : L'instructions d'avertissement du module.
- `unknown_instructions` : L'instructions de module inconnues.
- `critical_inverse` : Activer le Inverse interval au seuil critique du module [0/1].
- `warning_inverse` : Activer le Inverse interval au seuil d'alerte du module [0/1].
- `quiet` : Il active le mode *Quiet* du module [0/1].
- `module_ff_interval` : Il spécifie une valeur pour FF intervall du module.
- `alert_template` : Il associe un modèle d'alerte au module.
- `crontab` : Il spécifie une crontab dans le module.
- `min_ff_event_normal` : La valeur de FF threshold dans le changement d'état à NORMAL.
- `min_ff_event_warning` : La valeur de FF threshold dans le changement d'état à WARNING.
- `min_ff_event_critical` : La valeur de FF threshold dans le changement d'état à CRITICAL.
- `ff_timeout` : La valeur de FlipFlop timeout.
- `each_ff` : Activer l'option « Change each status ».
- `module_parent`: Nom du module du même agent qui sera le parent de ce module.
- `ff_type` : Activer les compteurs d'entretien de FF threshold [0/1].
- `min_warning_forced` : Forcer `min_warning` à la nouvelle valeur spécifiée même si le module existe, il est prioritaire sur `min_warning`.
- `max_warning_forced` : Forcer `max_warning` à la nouvelle valeur spécifiée même si le module existe, il est prioritaire sur `max_warning`.
- `min_critical_forced` : Forcer `min_critical` à la nouvelle valeur spécifiée même si le module existe, il est prioritaire sur `min_critical`.
- `max_critical_forced` : Forcer `max_critical` à la nouvelle valeur spécifiée, même si le module existe, il est prioritaire sur `max_critical`.
- `str_warning_forced` : Force `str_warning` à la nouvelle valeur spécifiée même si le module existe, il est prioritaire sur `str_warning`.
- `str_critical_forced` : Force `str_critical` à prendre la nouvelle valeur spécifiée, même si le module existe, il est prioritaire sur `str_critical`.

Ces *tokens* ne fonctionneront que pour *plugins* du dataserwer.

Tout autre élément sera enregistré en tant qu'information étendue du module dans la base de données de Pandora FMS. Un module doit avoir au moins un élément nom, type et données, par exemple :

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

Un fichier de données XML peut contenir un nombre quelconque d'éléments avant la fermeture de la balise `agent_data`.

Il existe un cas particulier de XML multi-éléments, basé sur une liste de données. Cela ne

s'applique qu'aux données de type chaîne de caractères. Le XML ressemblerait à quelque chose comme ceci :

```
<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>
```

Un horodatage peut être spécifié pour chaque valeur :

```
<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>
```

[Retour à l'index de la documentation de Pandora FMS](#)