



# Référence pour le développement de Pandora FMS



URL: <https://pandorafms.com/manual/!776/>  
Permanent link: [https://pandorafms.com/manual/!776/fr/documentation/pandorafms/technical\\_reference/01\\_development\\_and\\_extension](https://pandorafms.com/manual/!776/fr/documentation/pandorafms/technical_reference/01_development_and_extension)  
2024/06/10 14:34





# Référence pour le développement de Pandora FMS

## Développement Pandora FMS

### Architecture générale du code de Pandora FMS

Pour une explication distincte et détaillée de la structure de la base de données Pandora FMS, consultez l'article [Ingénierie Pandora FMS](#).

### Comment faire des liens compatibles

Pour tous les liens, vous devez utiliser la fonction `ui_get_full_url`.

- Comment utiliser `ui_get_full_url`  
Avant d'appeler la fonction, vous devez inclure « `fonctions_ui.php` ».
  - Lorsque vous avez besoin de l'URL pour rafraîchir :  
Par exemple

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

- Vous avez besoin d'une URL pour un chemin relatif  
Par exemple

#### *Méthode ancienne*

```
$url = $config['homeurl'] . "/relative/path/file_script.php";
```

#### *Nouvelle méthode*

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

- Et en javascript ? C'est aussi simple que ça.  
Par exemple

#### *Méthode ancienne*

```
<?php
...
$url = $config['homeurl'] . "/relative/path/file_script.php";
...
?>
<script type="text/javascript">
...
```

```
jQuery.post ('<?php $url; ?>',  
{  
...  
});  
...  
</script>
```

### Nouvelle méthode

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

- Cas particuliers :
  - Dans les liens directs vers index.php, il n'est pas nécessaire d'utiliser cette fonction.  
Par exemple

```
echo
```

## Points d'entrée d'exécution de la console Pandora FMS

La Console Pandora FMS ne dispose que de quelques points d'entrée d'exécution de l'application Web.

Contrairement à d'autres applications Web telles que Wordpress qui a à peine un point d'entrée pour le frontend et un backend pour l'administration. Ou le développement de sites Web de PME qui, normalement, chaque fichier php est un point d'entrée possible.

### Installation

Ce point d'entrée est pour effectuer une installation de la Console Pandora FMS et la base de données. Une fois l'installation terminée, Pandora FMS Console demande la suppression du fichier par sécurité.

```
install.php
```

## Exécution normale

Toute interaction des utilisateurs dans la console se fait par ce point d'entrée.

```
index.php
```

## Pétitions AJAX

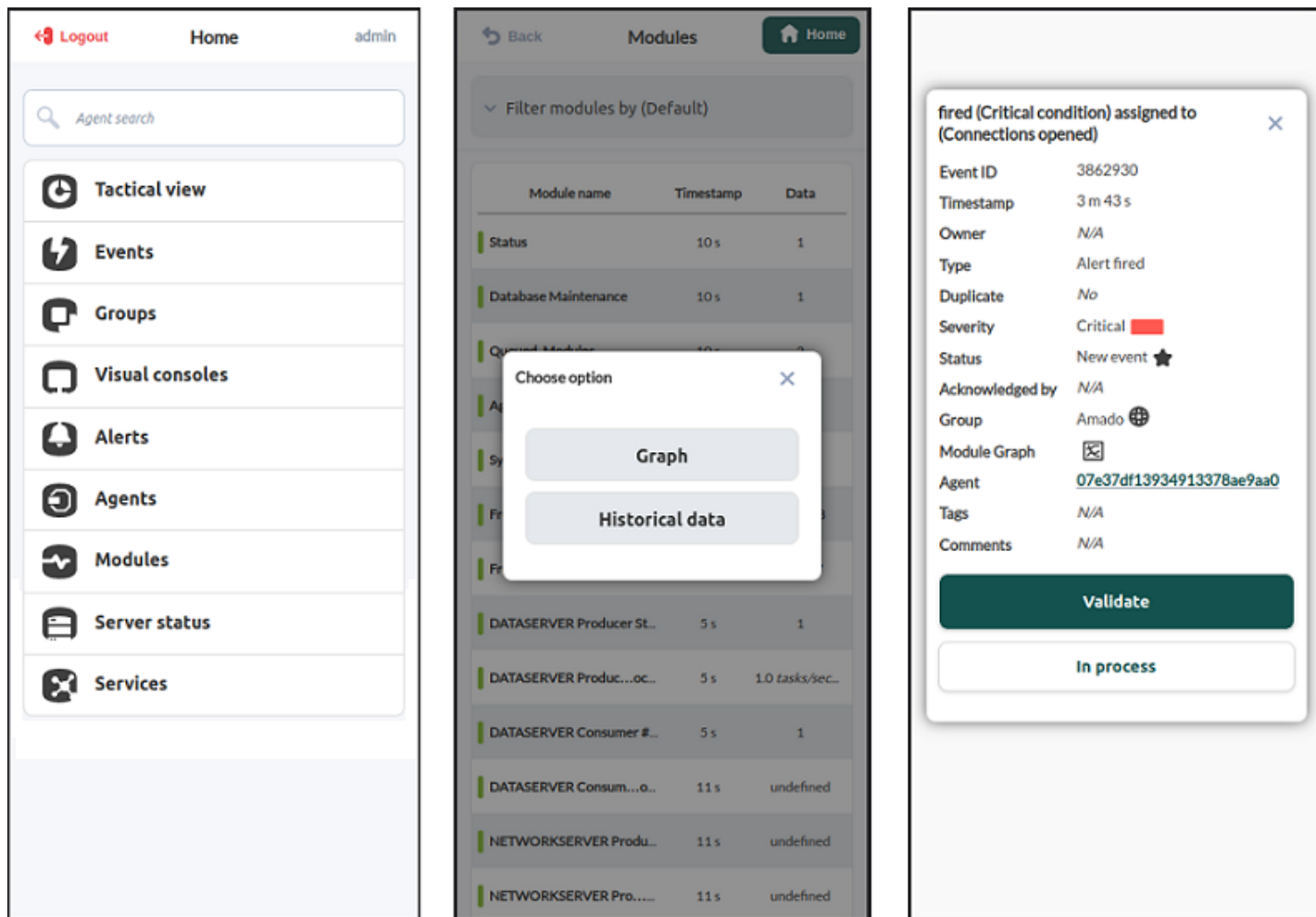
Toutes les requêtes Ajax pour les sécuriser (car les autorisations utilisateur sont vérifiées) et les rendre cohérentes et faciles à maintenir sont effectuées à travers ce fichier en passant par GET ou POST le paramètre page l'adresse relative du script à exécuter.

```
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

Depuis la version Pandora FMS 3.1, il dispose d'une API de type REST avec laquelle des applications tierces peuvent interagir via un simple canal sur le port 80 et avec le protocole HTTP.

La sécurisation de ce script se fait au moyen de trois paramètres, en validant les trois :

- Que l'adresse IP du client qui l'exécute soit dans la liste des adresses IP valides réglées dans la configuration de la Console Pandora FMS ou corresponde avec l'expression régulière également enregistrée au même endroit.
- Vous avez passé par paramètre l'API password qui est également configurée dans le setup de la Console Pandora FMS.
- Faites passer par paramètre l'utilisateur et le mot de passe de l'utilisateur qui va effectuer l'action de l'API.

```
include/api.php
```

## Cas particuliers

Dans la Console Pandora FMS, il existe plusieurs cas spéciaux pour les points d'entrée, généralement en évitant les connexions ou les traitements généraux effectués au point d'entrée principal (index.php de la racine).

## Extension Tâches Cron

Cette extension à l'aide d'une commande wget dans le cron peut exécuter les tâches cron qui lui ont été assignées sans demander de connexion de manière interactive. Bien sûr, l'ensemble des tâches est délimité et connu empêchant l'exécution de code malveillant sans connexion.

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

## Console visuelle vue externe

Il génère une page avec la vue d'une console visuelle en plein écran sans avoir besoin d'une connexion, bien que pour l'authentification si un hachage généré pour chaque vue publiée de cette manière soit nécessaire.

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

## Console Networkmap Popup détail

Une fenêtre contextuelle qui affiche la vue détaillée d'un agent qui a un élément dans la console Networkmap. Permet d'authentifier la session de l'utilisateur qui s'est connecté à la Console Pandora FMS.

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

## Graphique de modules Popup

Une fenêtre contextuelle qui affiche un graphique d'un module en détail et qui permet également de configurer plusieurs paramètres d'affichage. Elle utilise les données de l'utilisateur qui s'est connecté à la Console Pandora FMS pour l'authentification.

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

## Graphiques statiques

Les graphiques statiques sont des fichiers image qui sont générés à l'aide de scripts PHP, les données à afficher, si elles sont étendues sont sérialisées dans un fichier spécial qui traite le script, ce fichier a une durée de vie pour éviter les accès malveillants et les attaques par DEUX. Pour exécuter ce script, il n'est pas nécessaire d'être authentifié dans Pandora FMS.

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

## Rapports

### Rapport sur CSV

Le script génère un fichier texte contenant les données CSV, pour l'authentification utilise les

données de l'utilisateur connecté.

```
entreprise/operation/reporting/reporting_viewer_csv.php
```

Rapport en PDF

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

```
entreprise/operation/reporting/reporting_viewer_pdf.php
```

Événements

Événements sonores Popup

Fenêtre contextuelle qui vérifie périodiquement s'il existe un événement pour le montrer de manière sonore et visuelle. L'authentification se fait à l'aide des données de l'utilisateur connecté.

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

Événements en 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 en RSS

Le script génère un fichier texte contenant les données CSV, pour l'authentification il utilise les données de l'utilisateur connecté dans un hash en tant que paramètre.

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

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

### Critère pour les États et 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 (couleur grise) : 1 ou plusieurs modules dans un état inconnu et aucun dans un état critique et d'avertissement.
- Normal (couleur verte) : Tous les modules sont à l'état nominal.



Code interne de l'état dans la base de données :

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

## Agents

### Fonctions d'État

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

Toutes les fonctions ont 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 : tagent\_état, tagent et tagent\_module.

- agents\_monitor\_critical (\$id\_agent, \$filter = \_ckgedit\_QUOTckgedit>) : Il renvoie le nombre de modules dans un état critique pour l'agent. \*agents\_monitor\_warning (\$id\_agent, \$filter = \_ckgedit\_QUOTckgedit>) : Il renvoie le nombre de modules en état d'avertissement pour l'agent.
- agents\_monitor\_unknown (\$id\_agent, \$filter = \_ckgedit\_QUOTckgedit>) : Il renvoie le nombre de modules dans un état inconnu pour l'agent. \*agents\_monitor\_ok (\$id\_agent, \$filter = \_ckgedit\_QUOTckgedit>) : 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 aux agents pour certaines vues :

- agents\_tree\_view\_alert\_img (\$alert\_fired) : Il retourne le chemin de l'image d'alerte utilisée dans la vue Tree View.
- agents\_tree\_view\_status\_img (\$critical, \$warning, \$unknown) : Il retourne le chemin de l'image pour l'état de l'agent utilisé dans la vue Tree View.

## Groupes

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

Soyez prudent ! 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 statistics` est désactivé.

### Fonctions de la console

Les fonctions de la console calculent les statistiques sur la base d'un tableau de groupes d'agents. Ces fonctions n'utilisent pas les agents ou modules désactivés.

- `groups_agent_unknown` (`$group_array`) : Il renvoie le nombre d'agents dans un état inconnu pour les groupes donnés.
- `groups_agent_ok` (`$group_array`) : Il renvoie le nombre d'agents dans un état normal pour les groupes donnés.
- `groups_agent_critical` (`$group_array`) : Il renvoie le nombre d'agents dans un état critique pour les groupes donnés.
- `groups_agent_warning` (`$group_array`) : Il renvoie le nombre d'agents dans un état d'avertissement pour les groupes donnés.

Ces fonctions calculent les statistiques pour les modules. Ces fonctions n'utilisent pas les agents ou modules 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 en état normal pour les groupes donnés.
- `groups_monitor_critical` (`$group_array`) : Il renvoie le nombre de modules en état critique pour les groupes donnés.
- `groups_monitor_warning` (`$group_array`) : Il renvoie le nombre de modules en état d'avertissement pour les groupes donnés.
- `groups_monitor_unknown` (`$group_array`) : Il renvoie le nombre de modules en état inconnu pour les groupes donnés.
- `groups_monitor_alerts` (`$group_array`) : Il renvoie le nombre avec des alertes pour les groupes donnés.
- `groups_monitor_fired_alerts` (`$group_array`) : Il renvoie le nombre de modules avec des alertes déclenchées pour les groupes donnés.

### Modules

Ces fonctions renvoient les statistiques basées sur le nom du module. Il ne prend pas en compte les agents et les 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` (`$module_name`) : Il renvoie le nombre d'agents en état critique qui ont un module avec le nom donné.
- `modules_agents_warning` (`$module_name`) : Il renvoie le nombre d'agents en état d'avertissement qui ont un module avec le nom donné.

Ces fonctions renvoient les statistiques en utilisant le groupe de modules comme filtre. Il ne prend pas en compte les agents et les modules désactivés.

- `modules_group_agent_unknown` (`$module_group`) : Il renvoie le nombre d'agents à l'état inconnu ayant des modules appartenant au groupe de modules donné.
- `modules_group_agent_ok` (`$module_group`) : Il renvoie le nombre d'agents à l'état normal ayant des modules appartenant au groupe de modules donné.
- `modules_group_agent_critical` (`$module_group`) : Il renvoie le nombre d'agents à l'état critique ayant des modules appartenant au groupe de modules donné.
- `modules_group_agent_warning` (`$module_group`) : Il renvoie le nombre d'agents à l'état d'avertissement ayant des modules appartenant au groupe de modules donné.

## Politiques

Ces fonctions renvoient le nombre d'agents pour chaque état et politique donnée. Ils n'utilisent pas d'agents et de modules désactivés pour calculer le résultat.

- `policies_agents_critical` (`$id_policy`) : Il renvoie le nombre d'agents d'état critiques appartenant à une stratégie donnée.
- `policies_agents_ok` (`$id_policy`) : Il renvoie le nombre d'agents d'état normales appartenant à une politique donnée.
- `policies_agents_unknown` (`$id_policy`) : Il renvoie le nombre d'agents d'état inconnu appartenant à une politique donnée.
- `policies_agents_warning` (`$id_policy`) : Il renvoie le nombre d'agents en état d'avertissement appartenant à une politique donnée.

## OS

Ces fonctions calculent les statistiques pour les agents sur la base des systèmes d'exploitation auxquels ils appartiennent. Ces fonctions n'utilisent pas les agents ou modules désactivés.

- `os_agents_critical` (`$id_os`) : Il renvoie le nombre d'agents dans un état critique appartenant au système d'exploitation donné.
- `os_agents_ok` (`$id_os`) : Il renvoie le nombre d'agents dans un état normal appartenant au système d'exploitation donné.
- `os_agents_warning` (`$id_os`) : Il renvoie le nombre d'agents dans un état d'avertissement appartenant au système d'exploitation donné.
- `os_agents_unknown` (`$id_os`) : Il renvoie le nombre d'agents dans un état inconnu appartenant au système d'exploitation donné.

## Développement et élargissement

La plupart des extensions ont été décrites comme des annexes distinctes, spécialisées dans la création de plugins de serveur, de plugins d'agent Unix et d'extensions de la console. Ce chapitre décrit comment collaborer avec Pandora FMS et comment compiler l'agent Windows à partir des sources. À l'avenir, tout autre sujet lié au développement qui n'a pas d'appendice spécifique fera l'objet de ce chapitre.

### Collaborer avec le projet Pandora FMS

Ce projet est maintenu par des développeurs bénévoles avec leurs collaborations. Les nouveaux développeurs, les rédacteurs de documents ou les personnes qui souhaitent aider seront toujours les bienvenus. Une bonne façon de commencer est de vous abonner à nos listes de diffusion et/ou au forum.

### Bugs / Défauts

Signaler des erreurs potentielles vous aide à améliorer Pandora FMS. Avant de soumettre un rapport d'erreur, veuillez consulter notre base de données de bogues et, en cas de détection d'un bogue non signalé, l'envoyer à l'aide de l'excellent outil Sourceforge pour le suivi et le signalement des bogues sur le SITE WEB du projet : <http://sourceforge.net/projects/pandora/>

### Listes de diffusion

Les listes de diffusion sont un excellent moyen de se tenir au courant 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 (parfois quotidiennes) de développement en utilisant le système de notification automatique de notre SVN (Code Version Control System).

## 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 le faire en téléchargeant les sources du dépôt de code SubVersion de Sourceforge, où nous avons publié le référentiel officiel Pandora FMS. Pour ce faire, vous devez installer un client Subversion. Lorsque vous l'avez prêt, exécutez la commande suivante à partir de la ligne de commande :

```
svn co https://svn.sourceforge.net/svnroot/pandora pandora
```

## Compilation à partir de Windows

Pour compiler/liaison à partir de sources, vous devrez obtenir la dernière version de l'environnement de développement Dev-CPP (IDE), qui à son tour inclut le compilateur MinGW. [Téléchargez-le à partir de ce lien](#).

Ouvrez le fichier *PandoraAgent.dev* avec Dev-CPP et générez le projet. Après avoir ajusté différents paramètres (principalement des chemins), vous devriez être en mesure de compiler l'agent Windows.

Si vous rencontrez des problèmes de compilation à partir de sources, contactez directement le responsable du développement de l'équipe Windows ([ramon.novoa@artica.es](mailto:ramon.novoa@artica.es)) ou via notre site Web [Sourceforge](#).

## Compilation croisée à partir de Linux

Pour réaliser la compilation croisée (cela signifie créer l'exécutable Windows natif à partir d'un environnement Linux), suivez ces étapes :

### Installez le compilateur WinGW pour Linux

Pour les environnements Ubuntu/Debian :

```
sudo aptitude install mingw32
```

Pour les environnements SUSE ou RPM pris en charge, vous pouvez le faire avec zypper ou manuellement avec cette URL :

```
http://download.opensuse.org/repositories/CrossToolchain:/mingw/openSUSE_11.1/
```

### Installation des bibliothèques et includes extras dont l'agent a besoin

- win32api
- odbc++
- curl
- openssl
- zlib
- Boost C++ libraries (<http://sourceforge.net/projects/boost/files/>)

Par exemple, pour installer le paquet de bibliothèques OpenSSL, accédez à

<http://sourceforge.net/projects/devpaks/files> et téléchargez le fichier :

```
openssl-0.9.8e-1cm.DevPak
```

Décompressez le fichier *openssl-0.9.8e-1cm.DevPak* :

```
tar jxvf openssl-0.9.8e-1cm.DevPak
```

Copiez les bibliothèques et includes dans votre environnement de compilation croisée avec MinGW :

```
cp lib/*.a /usr/i586-mingw32msvc/lib/  
cp -r include/* /usr/i586-mingw32msvc/include/
```

Téléchargez et installez les paquets 'autogen' et 'autoconf'

```
sudo apt-get install autogen autoconf
```

Accédez au répertoire où se trouvent les sources de l'agent Windows et exécutez :

```
./autogen.sh
```

Il existe une alternative, plus rapide, plus confortable et plus facile, mais vous devrez résoudre vous-même les problèmes potentiels de dépendance et de version. Nous avons fait un tarball avec toutes les bibliothèques et includes nécessaires et les avons téléchargées sur le SITE officiel sous le nom de *mingw\_pandorawin32\_libraries\_9Oct2009.tar.gz*

### Compilation et lien

Après avoir installé le compilateur, accédez au répertoire où se trouvent les sources de l'agent Windows et exécutez

```
./configure --host=i586-mingw32msvc && make
```

Cela devrait compiler et lier l'exécutable de l'agent Pandora FMS pour Windows.

### API de Pandora FMS

Il existe une API externe de Pandora FMS permettant de relier des applications tierces à Pandora FMS, à la fois pour obtenir des informations de Pandora FMS et pour saisir des informations au sein de Pandora FMS. Toute cette documentation se trouve [dans l'appendice API externe de Pandora FMS](#) (anglais)

## Format des fichiers de données XML

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

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

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

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

- *description* : Description de l'agent.
- *group* : Nom du groupe auquel appartient l'agent (il doit exister dans la base de données Pandora FMS). S'il est laissé vide et qu'il n'y a pas de groupe configuré par défaut sur le serveur, l'agent ne sera pas créé.
- *os\_name* : Nom du système d'exploitation auquel appartient l'agent (il doit exister dans la base de données Pandora FMS).
- *os\_version* : Chaîne libre décrivant la version du système d'exploitation.
- *interval* : Intervalle de l'agent (en secondes).
- *version* : 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* : Nom de l'agent.
- *timezone\_offset* : Déplacement qui s'ajoute à l'horodatage (en heures). Utile si vous travaillez avec UTC.
- *address* : Adresse IP de l'agent (ou FQN).
- *parent\_agent\_name* : Nom du père de l'agent.
- *agent\_alias* : Alias de l'agent.
- *agent\_mode* : Mode de travail de l'agent (0 : Normal mode, 1 : Learning mode, 2 : Autodisable mode)
- *secondary\_groups* : Groupes secondaires ajoutés à l'agent.
- *custom\_id* : Identifiant personnalisé de l'agent.
- *url\_address* : URL d'accès à l'agent.

Prenons l'exemple de l'en-tête XML :

```
<agent_data description= group= os_name='linux' os_version='Ubuntu 10.10'  
interval='30' version='3.2(Build 101227)' timestamp='2011/04/20 12:24:03'  
agent_name='foo' timezone_offset='0' parent_agent_name='too'  
address='192.168.1.51' custom_id='BS4884' url_address='http://mylocalhost:8080'>
```

Ensuite, vous avez besoin d'un élément module pour chaque module, et les éléments suivants peuvent être imbriqués pour définir le module :

- *name* : Nom du module.
- *description* : Description du module.
- *tags* : Étiquettes associées au module.

- type : Type de module (il doit exister dans la base de données Pandora FMS).
- data : Nom du module.
- max : Valeur maximale du module.
- min : Valeur minimale du module.
- post\_process : Valeur de post-traitement.
- module\_interval : Intervalle du module (intervalle en secondes / intervalle de l'agent).
- min\_critical : Valeur minimale pour l'état critique.
- max\_critical : Valeur maximale pour l'état critique.
- min\_warning : Valeur minimale pour l'état d'alerte.
- max\_warning : 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 : Seuil FF (voir).
- status : État du module (NORMAL, WARNING ou CRITICAL). Les limites des états critiques et d'alerte sont ignorées si l'état est spécifié.
- datalist : Il envoie les données du module au format datalist (une entrée de base de données pour chacune des valeurs reçues) [0/1].
- unit : Unité du module. Il prend en charge la macro `_Timeticks_` pour transformer une donnée au format `timeticks` en `jj/hh/mm/ss`.
- timestamp : Il définit un horodatage sur les données reçues du module.
- module\_group : Groupe de modules auquel le module sera ajouté.
- custom\_id : ID personnalisé du module.
- str\_warning : Seuil d'avertissement pour les modules string.
- str\_critical : Seuil critique pour les modules string.
- critical\_instructions : Critical instructions du module.
- warning\_instructions : Instructions warning du module.
- unknown\_instructions : Instructions unknown du module.
- critical\_inverse : Il active l'intervalle inverse sur le seuil critique du module. [0/1].
- warning\_inverse : Il active l'intervalle inverse sur le seuil d'avertissement du module. [0/1].
- quiet : Il active le mode Quiet du module [0/1].
- module\_ff\_interval : Il spécifie une valeur de l'Intervalle FF du module.
- alert\_template : Il associe un modèle d'alerte au module.
- crontab : Il spécifie un crontab dans le module.
- min\_ff\_event\_normal : Valeur du seuil FF au changement d'état à NORMAL.
- min\_ff\_event\_warning : Valeur du seuil FF au changement d'état à WARNING.
- min\_ff\_event\_critical : Valeur du seuil FF au changement d'état à CRITICAL.
- ff\_timeout : Valeur de FlipFlop timeout.
- each\_ff : Il active l'option « Change each status ».
- module\_parent : Nom du module dans le même agent qui sera le père de ce module.
- ff\_type : Il active le Keep counters du FF threshold. [0/1].

À partir de la version 749 de Pandora FMS, de nouveaux jetons sont ajoutés pour forcer les seuils :

- min\_warning\_forced : Il force `min_warning` à la nouvelle valeur indiquée même si le module existe. Il a priorité sur `min_warning`.
- max\_warning\_forced : Il force `max_warning` à la nouvelle valeur indiquée même si le module existe. Il a priorité sur `max_warning`.
- min\_critical\_forced : Il force `min_critical` à la nouvelle valeur indiquée même si le module existe. Il a priorité sur `min_critical`.
- max\_critical\_forced : Il force `max_critical` à la nouvelle valeur indiquée même si le module existe. Il a priorité sur `max_critical`.
- str\_warning\_forced : Il force `str_warning` à la nouvelle valeur indiquée même si le module existe. Il a priorité sur `str_warning`.



- `str_critical_forced` : Il force `str_critical` à la nouvelle valeur indiquée même si le module existe. Il a priorité sur `str_critical`.

Ces jetons ne fonctionneront que pour les plugins du serveur de données.

Tout autre élément sera enregistré en tant qu'information étendue du module dans la base de données Pandora FMS :

#### Full list of monitors

F.	P.	Type	Module name	Description	Status	Warn	Data	Graph	Last contact
			CPU	CPU usage percentage.	<span style="background-color: green; color: white;"> </span>	N/A - N/A	8		5 seconds

No simple alerts found

Un module doit avoir au moins un élément name, type et data.

Par exemple :

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

Il peut y avoir n'importe quel nombre d'éléments dans un fichier de données XML. Enfin, n'oubliez pas de fermer l'étiquette `agent_data` !

Il existe un cas particulier de XML multi item, basé sur une liste de données. Cela s'applique uniquement aux données de type chaîne. Le XML ressemble à 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>
```

Vous pouvez spécifier un horodatage 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>
```

Voyons quelques autres exemples qui incluent l'utilisation de seuils et d'unités:

```
<module>
  <name><![CDATA[Cache mem free]]></name>
  <description><![CDATA[Free cache memory in MB]]></description>
  <tags>tag</tags>
  <type>generic_data</type>
  <module_interval>1</module_interval>
  <min_critical>100</min_critical>
  <max_critical>499</max_critical>
  <min_warning>500</min_warning>
  <max_warning>600</max_warning>
  <unit><![CDATA[MB]]></unit>
  <data><![CDATA[3866]]></data>
</module>

<module>
  <name><![CDATA[Load Average]]></name>
  <description><![CDATA[Average process in CPU (Last minute)
]]></description>
  <tags>tag</tags>
  <type>generic_data</type>
  <module_interval>1</module_interval>
  <data><![CDATA[1.89]]></data>
</module>
```

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