



Pandora FMS 開発リファレンス



URL: <https://pandorafms.com/manual/!776/>
Permanent link:
https://pandorafms.com/manual/!776/ja/documentation/pandorafms/technical_reference/01_development_and_extension
2024/06/10 14:34



Pandora FMS 開発リファレンス

[Pandora FMS ドキュメント一覧に戻る](#)

Pandora FMS の開発

Pandora FMS コードのアーキテクチャ

Pandora FMS データベースの構造に関する詳細については、[Pandora FMS の技術情報](#) を参照してください。

互換リンクの作成方法

すべてのリンクでは `ui_get_full_url` 関数を利用する必要があります。

- `ui_get_full_url` の利用方法
以前は、“`functions_ui.php`” をインクルードする必要がありました。

- リフレッシュ用の url が必要です。
例

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

- url に相対パスが必要です
例

旧手法

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

新手法

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

- javascript であっても同様に簡単です。
例

旧手法

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

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

新手法

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

- 特別なケース:
 - この関数を利用するためにindex.php への直接リンクは必要ありません。
例

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

Pandora コンソールの実行開始ポイント

Pandora コンソールのウェブアプリケーション実行開始ポイントは非常に少ないです。

例えば Wordpress などの他のウェブアプリケーションは、フロントエンドやバックエンドで一つの開始ポイントになっています。またPyMEs などは通常それぞれの php ファイルが実行開始ポイントになっています。

インストール

この実行開始ポイントは、Pandora コンソールおよびデータベースのインストール処理のためのものです。インストールが終了したらPandora コンソールはセキュリティのためにこのファイルを削除するように求めます。

```
install.php
```

通常実行

ブラウザからユーザが利用するコンソールでは、すべての操作がこの実行開始ポイントです。

```
index.php
```

AJAX リクエスト

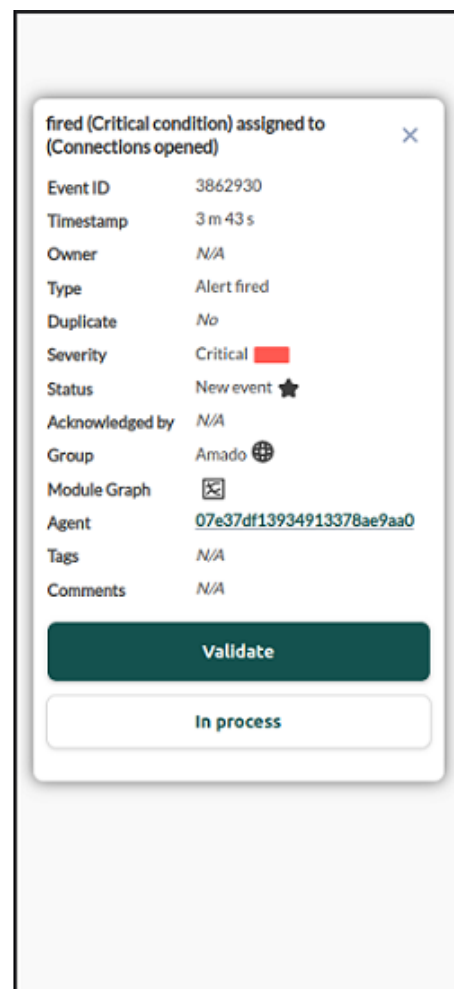
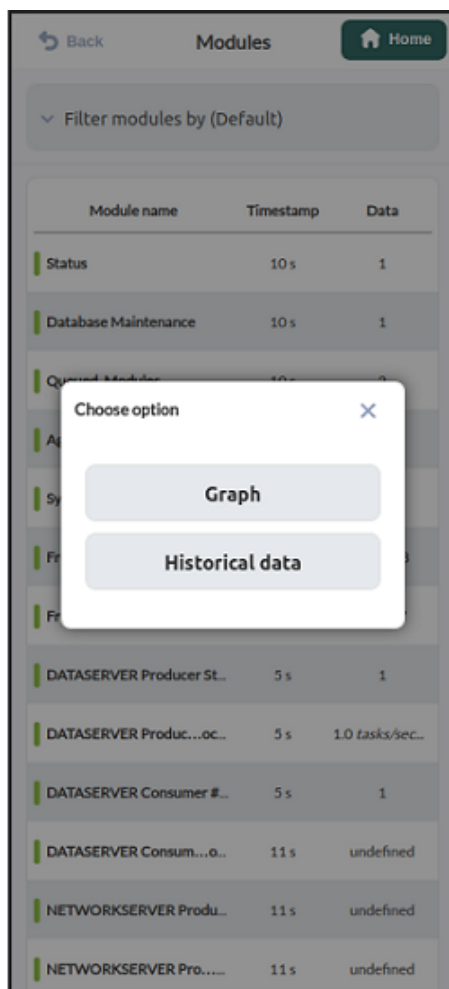
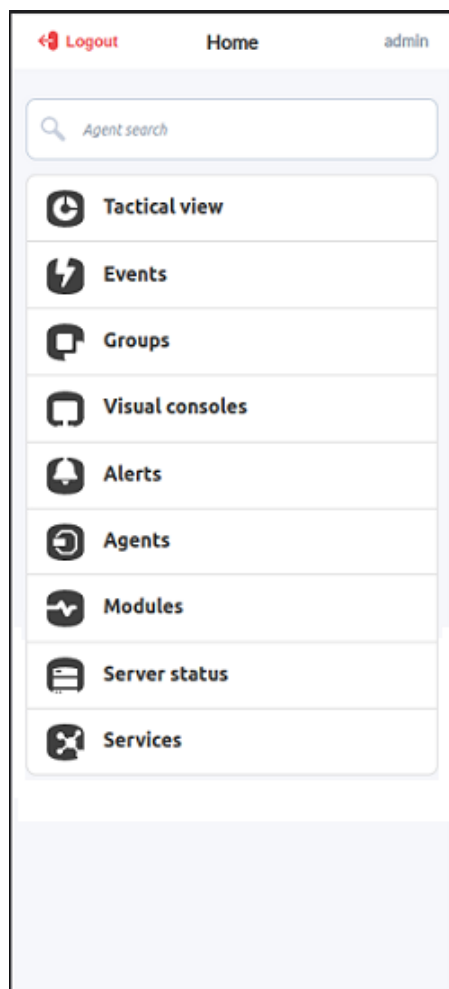
セキュリティの必要性(ユーザの権限チェック)および、構造およびメンテナンスを簡単にするために、すべての AJAX リクエストはこのファイルにアクセスします。実行処理では、このファイルに GET や POST で実際に実行するスクリプトの相対ディレクトリを “ ページ ” パラメータで渡します。

```
ajax.php
```

モバイルコンソール

コンピュータのモニタよりもはるかに小さい画面のモバイル端末の場合、Pandora FMS はこれらのデバイス用にコンソールの縮小バージョンを提供し、視覚的にも機能的にも簡素化されています。

```
mobile/index.php
```



API

Pandora FMS 3.1 から、80番ポートの HTTP プロトコルを使って Pandora FMS を他のアプリケーションから操作できる ASI を備えています。

このスクリプトのセキュリティは次の 3 つから成ります。

- クライアント IP は、指定したリストに書かれているか、リストの正規表現にマッチしなければいけません。IP リストは、Pandora FMS の設定画面から設定します。
- API パスワードパラメータを渡さないといけません。このパスワードはまた Pandora FMS 設定画面で設定します。
- パラメータとして、パスワードとユーザを指定しなければいけません。このユーザは、API で操作を実行する権限を持っていないければなりません。

```
include/api.php
```

特別ケース

Pandora コンソールでは、実行開始ポイントについていくつかの特別ケースがあります。それは、メインの開始ポイント(index.php)からのログインを回避するような場合です。

Cron タスクの実行

これは cron から wget コマンドを呼び出すことにより実行されます。また、ログインすることなくタスクを実行することができます。もちろん、マルウェアでの実行を避けるために、タスクのグループで制限されています。

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

ビジュアルコンソールの外部表示

このスクリプトは、ビジュアルコンソールをログインせずに(メニューなしで)全画面表示するものです。ただしハッシュ認証が必要です。ハッシュはそれぞれのビジュアルコンソールで生成します。

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

ネットワークマップコンソールのポップアップ

ネットワークマップのアイテムからエージェント詳細をポップアップウィンドウで表示します。これは Pandora コンソールにログインしているユーザのセッション情報を認証に使用します。

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

モジュールグラフのポップアップ

モジュールグラフをポップアップウィンドウ表示するものです。そのウィンドウの中で表示するグラフを変更するパラメータ設定もできます。これは Pandora コンソールにログインしているユーザ

のセッション情報を認証に使用します。

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

静的グラフ

データを表示する画像ファイルを生成する PHP スクリプトです。大量のデータがある場合、データはスクリプトにより特定のファイルの保存されます。ファイルは、不正アクセスや DOS アタックを防ぐために、有効期限があります。このスクリプトの実行には Pandora の認証は不要です。

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

レポート

CSV レポート

CSV フォーマットでデータが書かれたテキストファイルを生成するスクリプトです。このスクリプトはログインユーザの認証を利用します。

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

PDF レポート

このスクリプトは PDF ファイルを生成します。このスクリプトはログインユーザの認証を利用しません。

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

イベント

サウンドイベントポップアップ

このポップアップウィンドウは、サウンドイベントを表示するために新たなイベント定期的に確認します。このスクリプトはログインユーザの認証を使用します。

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

CSV イベント

このスクリプトは CSV フォーマットでデータが書かれたテキストファイルを生成します。このスクリプトはログインユーザの認証を使用します。

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

RSS イベント

このスクリプトは RSS フォーマットでイベントをテキストファイルに書き出します。このスクリプトはハッシュ認証を使います。

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

エージェント、モジュール、グループステータスの基本機能

状態を表す基準と DB での記録

状態は次の通りです。

- 障害(赤): 1つまたはそれ以上のモジュールが障害状態の場合
- 警告(黄色): 1つまたはそれ以上のモジュールが警告状態で、障害状態のものが無い場合
- 不明(グレー): 1つまたはそれ以上のモジュールが不明状態で、障害状態や警告状態のものが無い場合
- 正常(緑): すべてのモジュールが正常状態の場合

DB 内部の状態を表すレコードは次の通りです。

- 障害: 1
- 警告: 2
- 不明: 3
- 正常: 0

エージェント

状態関数

これらの関数は、エージェントの状態またはアラートの発生状況でフィルタリングしたモニター数を取得するものです。

すべての関数には、より柔軟に利用できるように filter パラメータが追加されています。すべての関数におけるフィルタの内容は、sql クエリの最後に追加されます。このフィルタで、tagente_estado, tagente, tagente_modulo テーブルを使ったフィルタをするために、いくつかの特別な sql 問い合わせを追加することができます。

- agents_monitor_critical (\$id_agent, \$filter = ""): 対象のエージェントの障害モジュール数を返します。
- agents_monitor_warning (\$id_agent, \$filter = ""): 対象のエージェントの警告モジュール数を返します。
- agents_monitor_unknown (\$id_agent, \$filter = ""): 不明状態のモジュール数を返します。
- agents_monitor_ok (\$id_agent, \$filter = ""): 正常状態のモジュール数を返します。
- agents_get_alerts_fired (\$id_agent, \$filter = ""): 対象エージェントのアラート数を返します。

補助関数

これらの関数は、いくつかの場面でエージェントに関連した処理を実行します。

- `agents_tree_view_alert_img ($alert_fired)`: 発生したアラートの数に依存したツリービューのアラート画像のパスを返します。
- `agents_tree_view_status_img ($critical, $warning, $unknown)`: ツリービューの状態画像のパスを返します。

グループ

これらの関数は、Pandora に定義されたエージェントグループにもとづいて、エージェントおよびモジュールの状態を返します。

サーバとコンソールの関数は、出力結果が同じになるように同じ sql クエリを利用しなければいけないことに注意してください。

サーバ関数

- `pandora_group_statistics`: この関数は、リアルタイム更新が off のときに、グループの状態を計算します。

コンソール関数

コンソール関数は、エージェントグループの構造に基づいて状態を計算します。これらの関数は、無効化されたエージェントやモジュールは返しません。

- `groups_agent_unknown ($group_array)`: 指定したグループの不明状態のエージェント数を返します。
- `groups_agent_ok ($group_array)`: 指定したグループの正常状態のエージェント数を返します。
- `groups_agent_critical ($group_array)`: 指定したグループの障害状態のエージェント数を返します。
- `groups_agent_warning ($group_array)`: 指定したグループの警告状態のエージェント数を返します。

これらの関数はモジュールの状態を計算します。無効化されたモジュールやエージェントには使えません。

- `groups_monitor_not_init ($group_array)`: 指定したグループの未初期化状態のエージェント数を返します。
- `groups_monitor_ok ($group_array)`: 指定したグループの正常状態のモジュール数を返します。
- `groups_monitor_critical ($group_array)`: 指定したグループの障害状態のモジュール数を返します。
- `groups_monitor_warning ($group_array)`: 指定したグループの警告状態のモジュール数を返します。
- `groups_monitor_unknown ($group_array)`: 指定したグループの不明状態のモジュール数を返します。
- `groups_monitor_alerts ($group_array)`: 指定したグループのアラートがついているモジュール数を返します。

ます。

- `groups_monitor_fired_alerts` (`$group_array`): 指定したグループのアラートが発生しているモジュール数を返します。

モジュール

これらの関数は、モジュール名をもとにした情報を返します。無効化したエージェントやモジュールには使えません。

- `modules_agents_unknown` (`$module_name`): 指定したモジュール名が不明状態のエージェント数を返します。
- `modules_agents_ok` (`$module_name`): 指定したモジュール名が正常状態のエージェント数を返します。
- `modules_agents_critical` (`$module_name`): 指定したモジュール名が障害状態のエージェント数を返します。
- `modules_agents_warning` (`$module_name`): 指定したモジュール名が警告状態のエージェント数を返します。

これらの関数は、モジュールグループみ基づいた状態を返します。無効化したエージェントやモジュールには使えません。

- `modules_group_agent_unknown` (`$module_group`): 指定したモジュールグループに所属しているモジュールが不明状態のエージェント数を返します。
- `modules_group_agent_ok` (`$module_group`): 指定したモジュールグループに所属しているモジュールが正常状態のエージェント数を返します。
- `modules_group_agent_critical` (`$module_group`): 指定したモジュールグループに所属しているモジュールが障害状態のエージェント数を返します。
- `modules_group_agent_warning` (`$module_group`): 指定したモジュールグループに所属しているモジュールが警告状態のエージェント数を返します。

ポリシー

これらの関数は、指定したポリシーのそれぞれの状態にあるエージェント数を返します。無効化したエージェントやモジュールには使えません。

- `policies_agents_critical` (`$id_policy`): 指定したポリシーに属する障害状態のエージェント数を返します。
- `policies_agents_ok` (`$id_policy`): 指定したポリシーに属する正常状態のエージェント数を返します。
- `policies_agents_unknown` (`$id_policy`): 指定したポリシーに属する不明状態のエージェント数を返します。
- `policies_agents_warning` (`$id_policy`): 指定したポリシーに属する警告状態のエージェント数を返します。

OS

これらの関数は、OSに基づいたエージェントの状態を返します。無効化したエージェントやモジュール

ルには使えません。

- `os_agents_critical ($id_os)`: 指定した OS で障害状態のエージェント数を返します。
- `os_agents_ok($id_os)`: 指定した OS で正常状態のエージェント数を返します。
- `os_agents_warning ($id_os)`: 指定した OS で障害状態のエージェント数を返します。
- `os_agents_unknown ($id_os)`: 指定した OS で不明状態のエージェント数を返します。

開発

ほとんどの拡張は、サーバプラグイン、Unix エージェントプラグイン、コンソール拡張といったそれぞれの章で説明しています。この章では、Pandora FMS との連携および Windows エージェントのソースからのコンパイル方法について説明します。今後も、特定の章に属さない開発に関わる内容については、この章に記載していきます。

Pandora FMS プロジェクトへの協力

このプロジェクトは、プロジェクトを支えるボランティアの開発者によって成り立っています。新たな開発者、ドキュメント作成者や、その他協力したいと考えている方は、常に歓迎です。まず最初は、メーリングリストやフォーラムに参加いただくのが良いです。

バグ / エラー

不具合報告は、Pandora FMS の拡張の助けになります。なお、不具合報告を送信する前に、我々の不具合に関するデータベースを確認し、同様の報告がすでにあがっていないかを確認してください。不具合報告は、我々のプロジェクトページ <http://sourceforge.net/projects/pandora/> から Sourceforge のバグトラッキングシステムを使って送信することができます。

メーリングリスト

メーリングリストは、簡単に最新情報をメールで受け取るには良い手段です。ユーザ向けのお知らせを送信する public メーリングリスト (流量は少ないです) と、技術的な議論を行う develop メーリングリストを用意しています。また SVN システムから自動的に通知される変更情報を受け取るためのメーリングリスト (ほぼ毎日) があります。

Windows エージェントのソースからのコンパイル

最新ソースの入手

我々のリポジトリから最新のソースを入手するには Subversion クライアントが必要です。そして、

次のように実行します。

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

Windows

ソースからビルドするには、MinGW tools を含む最新の Dev-Cpp IDE 版が必要です。 [こちらからダウンロードできます](#)。

PandoraAgent.dev を Dev-Cpp で開き、プロジェクトを作成します。デフォルトのインストールですべてコンパイル可能です。

ソースからのビルドで問題を見つけた場合は、メール (ramon.novoa@artica.es) もしくは [SourceForge 上のプロジェクトページ](#) にてお知らせください。

Linux でのクロスコンパイル

Linux で Pandora FMS Windows エージェントをクロスコンパイルするには、次のようにします。

Linux への MinGW のインストール

Ubuntu/Debian の場合:

```
sudo aptitude install mingw32
```

SUSE または RPM 管理の環境の場合 (手動での Zypper の利用) は、以下の RUL より:

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

エージェントに必要な追加ライブラリのインストール

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

例えば、openssl パッケージをインストールするには次のようにします。

<http://sourceforge.net/projects/devpaks/files/> へ行き、ファイルをダウンロードします。

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

openssl-0.9.8e-1cm.DevPak を展開します。

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

ライブラリおよびインクルードファイルを MinGW のあるクロスコンパイル環境にコピーします。

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

依存関係やライブラリの問題は、ご自身で解決する必要があります。 これらを簡単にできるように Pandora FMS の公式ダウンロードサイトに必要なライブラリとインクルードファイルを tar で固めたファイルを *mingw_pandorawin32_libraries_9Oct2009.tar.gz* という名前で置いてあります。

コンパイルおよびリンク

コンパイラおよび、インクルードファイル、ライブラリのインストール後、Pandora FMS エージェントのソースディレクトリへ行き、以下を実行します。

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

これで、実行可能な .exe ファイルが生成されます。

外部 API

Pandora FMS には、他のアプリケーションとの連携をとり Pandora FMS から情報を取得したり Pandora FMS に情報を提供するための外部 API があります。これに関するドキュメントは、補足資料の Pandora FMS 外部 API にあります。

Pandora FMS XML データファイルフォーマット

Pandora FMS の XML データファイルのフォーマットを理解することで、エージェントプラグインの拡張や、独自エージェントの作成、Pandora FMS データサーバへの独自 XML ファイルの送信などができるようになります。

すべての XML ドキュメントと同じように、データファイルは次のような XML の宣言で始まります。

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

次に、エージェントが送るデータを定義する `agent_data` 要素が来ます。次に示す属性に対応しています。

- *description*: エージェントの説明
- *group*: エージェントが属するグループ名 (Pandora FMS のデータベースに存在する必要があります)

- *os_name*: エージェントで動作している OS 名 (Pandora FMS のデータベースに存在する必要があります)
- *os_version*: OS のバージョンを示す任意の文字列
- *interval*: エージェントの実行間隔 (秒で)
- *version*: エージェントのバージョン文字列
- *timestamp*: XML ファイルが生成された日時 (YYYY/MM/DD HH:MM:SS)
- *agent_name*: エージェントの名前
- *timezone_offset*: 日時に追加する時間の差分(時間単位)UTC にて動作している場合に便利です。
- *address*: エージェントの IP アドレス(または FQDN)
- *parent_agent_name*: エージェントの親の名前
- *agent_alias*: エージェントの別名
- *agent_mode*: エージェントの動作モード(0: 通常モード, 1: 学習モード, 2: 自動無効化モード)
- *secondary_groups*: エージェントに追加するセカンダリグループ
- *custom_id*: カスタムエージェント ID
- *url_address*: エージェントアクセス URL

XML ヘッダの例を見てみましょう。

```
<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'>
```

そして、モジュールごとに、*module* という要素が必要です。モジュールの定義のために、次の要素を入れることができます。

- *name*: モジュールの名前。
- *description*: モジュールの説明。
- *tags*: モジュールに関連付けられたタグ。
- *type*: モジュールのタイプ (Pandora FMS のデータベースに存在する必要があります)
- *data*: モジュールのデータ。
- *max*: モジュールの最大値。
- *min*: モジュールの最小値。
- *post_process*: データ保存倍率。
- *module_interval*: モジュールの実行間隔。(秒間の実行間隔 / エージェントの実行間隔)
- *min_critical*: 障害状態となる最小値。
- *max_critical*: 障害状態となる最大値。
- *min_warning*: 警告状態となる最小値。
- *max_warning*: 警告状態となる最大値。
- *disabled*: モジュールの無効化(0)、有効化(1)。無効化されているモジュールは処理されません。
- *min_ff_event*: 連続抑制回数
- *status*: モジュールの状態(正常、警告、障害) *status* が設定されている場合は、警告および障害のしきい値は無視されます。
- *datalist*: *datalist* フォーマットでモジュールデータを送信します。(受信した値ごとに 1つのデータベースエントリ) [0/1]
- *unit*: モジュールの単位 *timeticks* 形式のデータを dd/hh/mm/ss に変換する *_Timeticks_* マクロをサポートします。
- *timestamp*: モジュールから受信したデータにタイムスタンプを設定します。
- *module_group*: モジュールが追加されるモジュールグループ。
- *custom_id*: モジュールカスタム ID
- *str_warning*: 文字列モジュールの警告閾値。
- *str_critical*: 文字列モジュールの障害閾値。

- critical_instructions: モジュール障害時手順。
- warning_instructions: モジュール警告時手順。
- unknown_instructions: モジュール不明時手順。
- critical_inverse: モジュールの障害閾値を反転。 [0/1]
- warning_inverse: モジュールの警告閾値を反転。 [0/1]
- quiet: モジュールの静観モードを有効化。 [0/1]
- module_ff_interval: モジュールの連続障害検知抑制回数の指定。
- alert_template: モジュールに関連付けるアラートテンプレート。
- crontab: モジュールの crontab 指定。
- min_ff_event_normal: 状態が正常に変わる際の連続障害検知抑制回数。
- min_ff_event_warning: 状態が警告に変わる際の連続障害検知抑制回数。
- min_ff_event_critical: 状態が障害に変わる際の連続障害検知抑制回数。
- ff_timeout: 連続障害検知抑制のタイムアウト値。
- each_ff: 個々の状態ごとの連続障害検知抑制の有効化。
- module_parent: 同一エージェントでこのモジュールの親となるモジュールの名前。
- ff_type: 連続障害検知抑制のカウント維持の有効化。 [0/1]

Pandora FMS バージョン 749 から、しきい値を強制するための新たなトークンが追加されました。

- min_warning_forced: モジュールに設定が存在する場合でも、警告の最小値を新しい値にします。min_warning よりも優先されます。
- max_warning_forced: モジュールに設定が存在する場合でも、警告の最大値を新しい値にします。max_warning よりも優先されます。
- min_critical_forced: モジュールに設定が存在する場合でも、障害の最小値を新しい値にします。min_critical よりも優先されます。
- max_critical_forced: モジュールに設定が存在する場合でも、障害の最大値を新しい値にします。max_critical よりも優先されます。
- str_warning_forced: モジュールに設定が存在する場合でも、文字列の警告条件を新しい値にします。str_warning よりも優先されます。
- str_critical_forced: モジュールに設定が存在する場合でも、文字列の障害条件を新しい値にします。str_critical よりも優先されます。

これらのトークンは、データサーバプラグイン のみで動作します。

その他の要素は、モジュールの拡張情報として Pandora FMS のデータベースに保存されます。

Full list of monitors

F.	P.	Type	Module name	Description	Status	Warn	Data	Graph	Last contact
			CPU	CPU usage percentage.		N/A - N/A	8		5 seconds

No simple alerts found

モジュールは、少なくとも name, type, および data の要素を持たなければいけません。

例:

```
<module>
  <name>CPU</name>
```

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

XML データファイルは任意の数のモジュール要素を持つことができます。最後に、agent_data タグをクローズするのを忘れないでください。

アイテムリストによる、複数の XML データを含む特別なケースがあります。これは、文字列タイプの場合にのみ有効です。XML は次のようになります。

```
<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>
```

タイムスタンプは、それぞれの値ごとに定義できます。

```
<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>
```

さらに、使用量の単位と閾値の定義を含んだ例を見てみましょう。

```
<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>
```