



# プラグイン開発ガイド



om:

<https://pandorafms.com/manual/!779/>

Permanent link:

[https://pandorafms.com/manual/!779/ja/documentation/pandorafms/technical\\_reference/11\\_pfms\\_plugins](https://pandorafms.com/manual/!779/ja/documentation/pandorafms/technical_reference/11_pfms_plugins)

2025/01/22 19:13



# プラグイン開発ガイド

[Pandora FMS ドキュメント一覧に戻る](#)

## 概要

このプラグイン開発ガイドには、独自のプラグインを作成したいユーザ向けのドキュメントが含まれています。このドキュメントを使用して、ビジネスニーズを解決するプラグインを作成する方法を学習してください。

## プラグインとは?

プラグインとも呼ばれるコンピュータアドオンは、別のアプリケーション (この場合は Pandora FMS) の機能を拡張できるようにするアプリケーションです。

## プラグインはなぜ便利なのか?

プラグインを使用すると Pandora FMS の機能を拡張できます。これにより、新しい監視オプションを開発する際にさまざまな可能性が提供されます。

それらを使用すると、サービス、アプリケーション、またはデータベースの監視などを統合でき、タスクや処理を変更または自動化するために使用することもできます。

目的の 1 つは、同じ作業環境でできるだけ多くのシステムとサービスを監視できるようにすることであり、プラグインの利用はそのタスクに大いに役立ちます。

ほとんどのプラグインは、Pandora FMS の外部サービスから収集されたデータまたはパフォーマンス統計を表示できるようにするのに便利です。

## Pandora FMS におけるプラグインタイプ

実行タイプ別:

- エージェントプラグイン: エージェントプラグインは Pandora FMS ソフトウェアエージェントによって実行されます。通常はローカルにあり、リモートでは動作しないため、エージェントデーモンによって実行されます。エージェントプラグインは通常、実行時に XML を出力し、モジュールにはデータが含まれます。
- サーバプラグイン: サーバプラグインは [プラグインサーバ](#) によって実行され、通常は API または別の方法でリモートでデータを取得します。(ほとんどの場合) エージェントプラグインとして設定することもできますが、推奨されるオプションは、サーバプラグインとして設定することです。サーバプラグ

インは、実行の成功を示す 1 などの 1 つの値のみを返します。

## プラグインの見つけ方

Pandora FMS には、システム用に作成されたすべてのプラグインをダウンロードできるライブラリがあります。

プラグインは誰でもライブラリにアップロードできますが、受け入れて公開する前にレビュープロセスがあります。

<https://pandorafms.com/library/>

現在、ライブラリには、すべての分野、データベース、アプリケーション、クラウド、メッセージングサービスなどをカバーする多数のプラグインがあります。

ライブラリには、快適に閲覧できるさまざまなタグやメニューに加えて、ライブラリにアップロードされたプラグインを検索できる検索エンジンがあります。

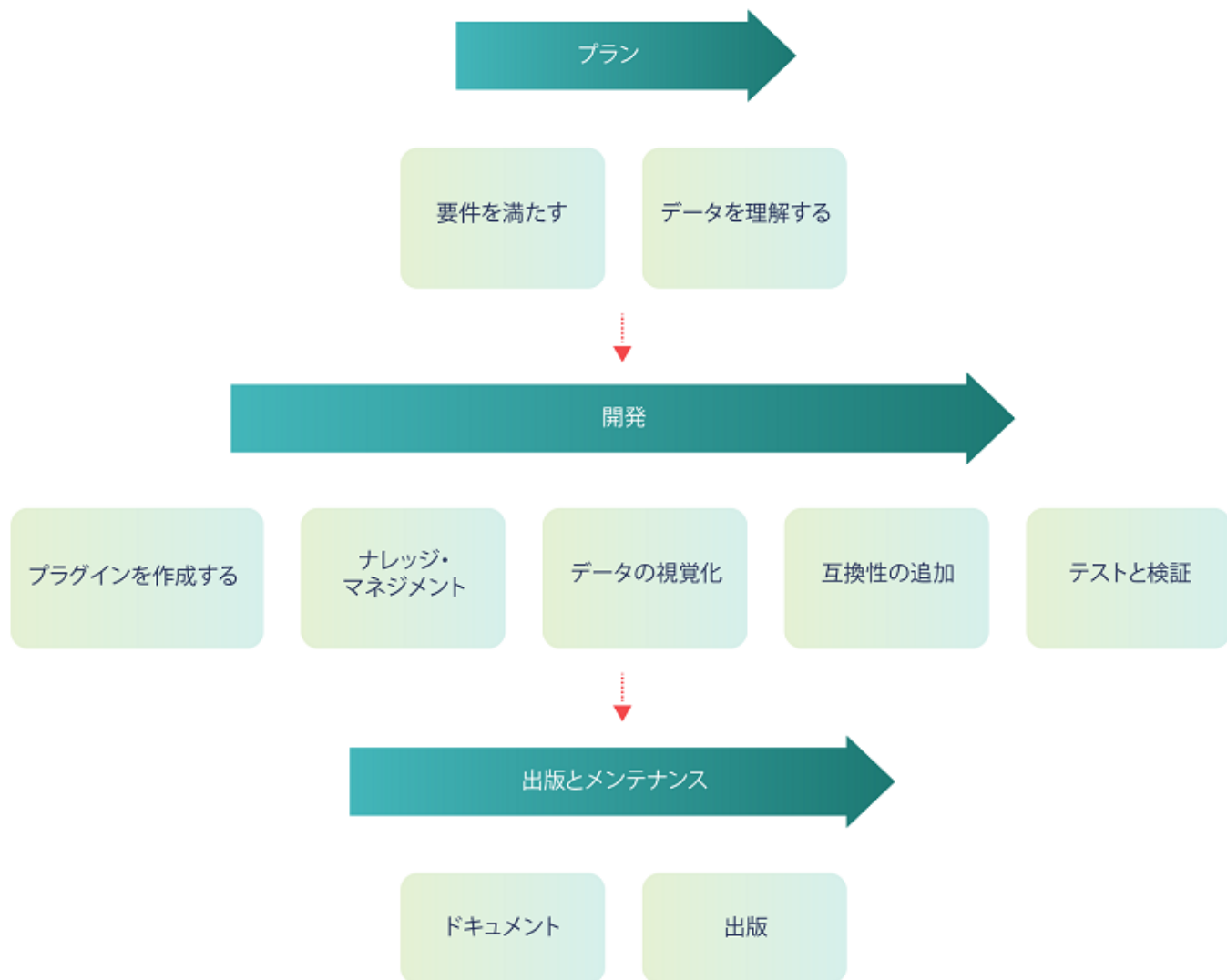
通常、各プラグインには、インストール、プラグインの設定、およびプラグインが収集するデータの詳細が記載されたドキュメントが付属しています。

## プラグインのインストール方法

プラグインは、Pandora FMS コンソールからインストールできます。タイプに応じて 2 つの異なる方法で実行でき、プラグイン設定が作成されます。ただし、エージェントプラグインはソフトウェアエージェントによって実行され、サーバプラグインはプラグインサーバによって実行されます。通常、サーバプラグインはエージェントとモジュールを作成しますが、エージェントプラグインは、それが設定されているソフトウェアエージェント内でのみモジュールを作成します。

プラグインをインストールするには Pandora FMS でカスタム実行を作成します。設定可能な特定の時間間隔ごとにプラグインが実行され、更新されたモジュールが表示されます。

## プラグインのライフサイクル



## プラグインの計画

プラグインを開発するときは、いくつかの側面を考慮して、その開発を正しく計画することが重要です。

### プラグインで解決できる問題

プラグイン計画の最初のステップで、次の質問に答えることができる必要があります。

- プラグインの目的は何でしょうか？
- あなたのプラグインが対応するユースケースは何ですか？

プラグインで達成しようとしているソリューションを明確にすることが重要です。 関心のあるサービスを監視し、Pandora FMS ですべてのデータを統合することで、時間やコストの面で有利になる場合があります。 また、プラグインにより電子メールを読み込んで特定のメッセージが到着したときアラームを鳴らすといった自動化をすることもできます。

プラグインは、ニーズを解決したり、相互作用、処理、およびデータ表示を容易にしたりできます。

## プラグインはどのようにデータを取得するのでしょうか？

プラグインの主な機能は、サービス、アプリケーション、データベースなどからデータを取得し、このデータを Pandora FMS に表示できるようにすることです。しかし、必要なサービスからデータを取得するにはどうすればよいのでしょうか？

これは間違いなく、プラグインで利用することを想定している技術やサービスに依存するため、それらについて予備調査を実施し、プラグインの実現可能性を調べ、プラグインを作成できるかどうかを調べる必要があります。可能であればテスト環境を用意します。

通常、これらのサービスには監視統計を取得する何らかの方法があり、その方法はサービスによって異なります。多くの場合、必要なデータを処理し、プラグインで表示できる API または CLI を持っています。

場合によっては、技術やサービス自体を作成したコミュニティまたは会社が、データとのやり取りをより簡単にするために作成したライブラリがあります。

## ユーザはデータを取得するためのアクセス許可が必要ですか？

サービスや技術からデータを取得する場合、データを操作するために特定のアクセス許可があるアカウントが必要になるか、データを取得する前に環境内で事前に設定を行う必要がある可能性があります。

一般的に言えば、これらの要件は、サービスまたは技術のドキュメントで示されています。

プラグインのドキュメントとして、サービスからデータを取得し詳細を残すためには何が必要でどのような要件があるかを明確にしておくことが重要です。

## データはリモートまたはローカルで収集されますか？

パフォーマンス統計は、ライブラリ、CLI または HTTP 呼び出しを使用した API のいずれかを介して、リモートで取得できます。または、データは、コマンドであったり独自の技術を使用して内部またはローカルでのみ取得できる場合があります。ケースに応じて、プラグインの実行は異なります。

データをリモートで取得できるプラグインは、データを取得するためにサービスマシン上に存在する必要がないため、プラグインサーバで実行できます。

ソフトウェアエージェントを介してのみ実行できる特定のプラグインもあります。データの取得がサービスと同じシステム上である必要がある場合です。

## データをどのように表示しますか？

Pandora コンソールでプラグイン データを表示する方法を定義する場合、このデータをどのように表示するかを指定することが重要です。

通常、これらのデータはエージェントとモジュールの構造に従って表示されるため、この構造を正しく定義して、混乱しないように、情報をできるだけ簡単に表示できるようにすることが重要です。大量のデータを処理したり、多くのデータベースや仮想マシンなどを持っている場合など、多くの「もの」を監視したりするのは、プラグインで実現するのが難しくなります。

プラグイン内に、エージェントまたはモジュールの名前をカスタマイズするオプションを含めたり、それらの名前にある種のプレフィックスを追加して、それらをより見やすく、区別しやすくすることができます。

## プラグインの要件と依存関係

プラグインで使用されている技術によっては、正しく動作させるために依存関係のインストールが必要になる場合があります。これらは、何らかの機能を実行するためにプラグインにインポートした言語のライブラリ、またはそれ自体で使用される言語のライブラリです。

プラグインが python3 で記述されている場合、プラグインを使用するマシンに python3 をインストールする必要があります。これは、そのプラグインのユーザにとって、潜在的な大きな障害になる可能性があります。

Python で作成されたプラグインでの一つのオプションとしては、プラグインで使用されるすべての依存関係をプラグインと共に requirements.txt ファイルで提供することです。このファイルをインストールすると、必要なすべての依存関係が自動的にインストールされます。

リモート MongoDB プラグインの requirements.txt ファイルの例。

```
dnspython==2.1.0  
pymongo==3.12.0
```

ファイルには、プラグインを実行するためにシステムにインストールしておく必要がある 2 つのライブラリが含まれていることがわかります (コンパイルされたプラグインでない場合)。

このファイルは手動で作成できますが、pip freeze コマンドを使用して、インストールされたすべての仮想環境の依存関係を含む requirements.txt ファイルを生成することもできます。

```
pip freeze> requirements.txt
```

依存関係をインストールする環境に requirements.txt ファイルをコピーし、次のコマンドを実行してそれらをインストールできます。

```
pip install -r requirements.txt
```

## プラグインのコンパイル

プラグインの依存関係の問題を解決する一つの方法は、すべての依存関係が既にインストールされている実行可能ファイルを作成することです。これは perl プラグインおよび python プラグインにとって理想的なソリューションです。

python で実行可能ファイルを作成する場合、pyinstaller ライブラリが利用可能です:

<https://pypi.org/project/pyinstaller/>

これは、次のコマンドでインストールできます。

```
pip install pyinstaller
```

インストールしたら、次のコマンドを使用してプラグインのバイナリファイルに作成できます。

GNU/Linux:

```
pyinstaller --onefile < plugin_name >.py
```

MS Windows:

```
python3 -m PyInstaller --onefile <nombre_plugin>.py
```

dist というフォルダーが生成され、その中にバイナリが生成されます。

あるオペレーティングシステムでコンパイルされたプラグインは別のオペレーティングシステムで動作しない場合があります。プラグインのコンパイルは Rocky Linux 7 で行うことをお勧めします。このシステムでコンパイルされたプラグインは通常、他のすべての GNU/Linux オペレーティングシステムで動作するためです。

Fedora および Ubuntu で作成したバイナリを使用すると、他のシステムで依存関係エラーが発生することを確認しています。

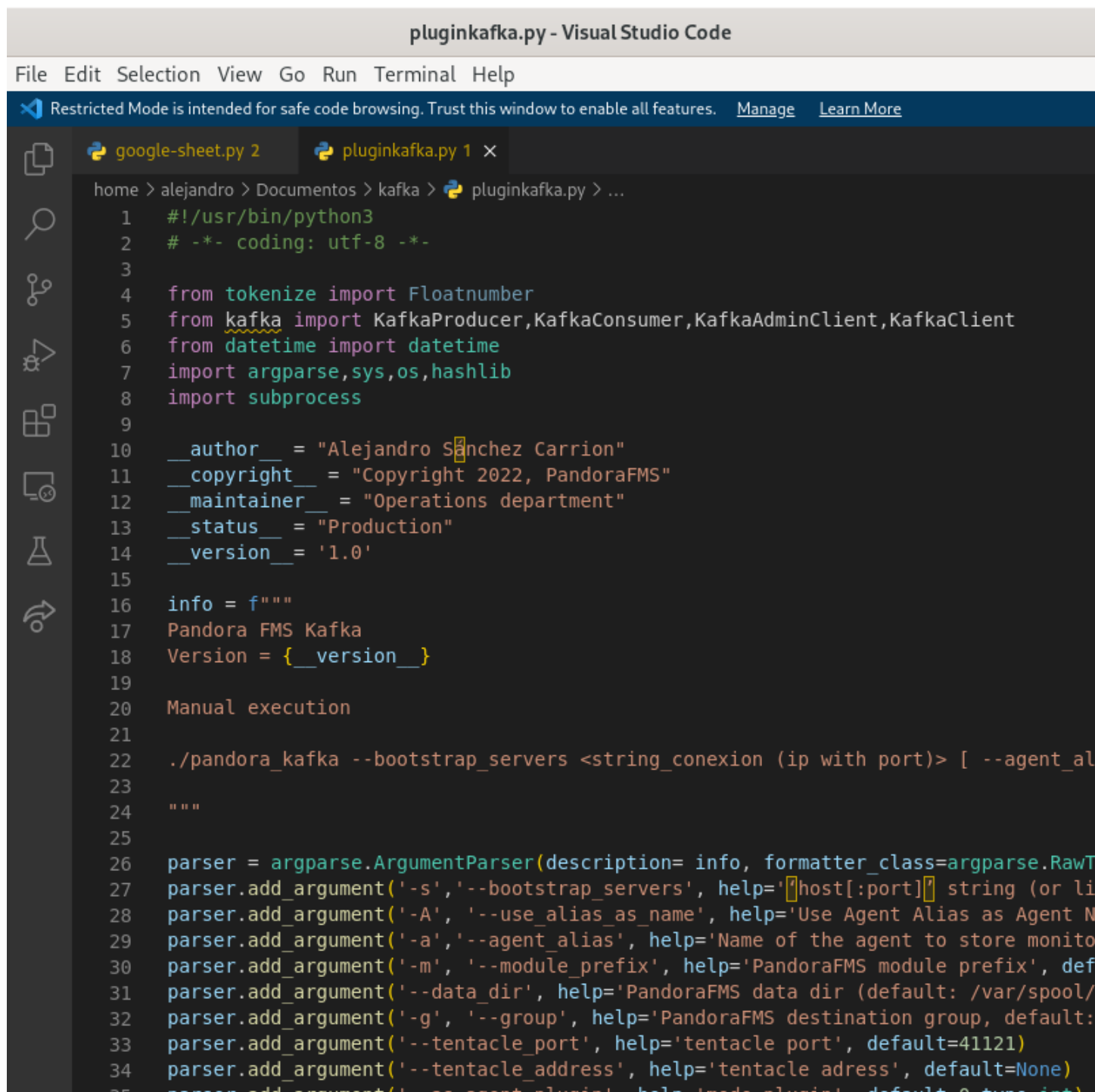
## プラグインの開発

プラグインの開発にどのツールと技術を使用するかを明確にすることが重要です。



## 開発ツール

プラグインは、データや Pandora FMS の追加機能を統合するスクリプトです。これらは小さなプログラムであるため、開発するには、フレームワークまたは Visual Studio Code などのコードエディタが必要です。



```
plugin_kafka.py - Visual Studio Code
File Edit Selection View Go Run Terminal Help
Restricted Mode is intended for safe code browsing. Trust this window to enable all features. Manage Learn More
google-sheet.py 2 plugin_kafka.py 1 x
home > alejandro > Documentos > kafka > plugin_kafka.py > ...
1  #!/usr/bin/python3
2  # -*- coding: utf-8 -*-
3
4  from tokenize import Floatnumber
5  from kafka import KafkaProducer,KafkaConsumer,KafkaAdminClient,KafkaClient
6  from datetime import datetime
7  import argparse,sys,os,hashlib
8  import subprocess
9
10  __author__ = "Alejandro Sánchez Carrion"
11  __copyright__ = "Copyright 2022, PandoraFMS"
12  __maintainer__ = "Operations department"
13  __status__ = "Production"
14  __version__ = '1.0'
15
16  info = f"""
17  Pandora FMS Kafka
18  Version = {__version__}
19
20  Manual execution
21
22  ./pandora_kafka --bootstrap_servers <string_conexion (ip with port)> [ --agent_al
23
24  """
25
26  parser = argparse.ArgumentParser(description= info, formatter_class=argparse.RawT
27  parser.add_argument('-s','--bootstrap_servers', help='[host[:port]] string (or li
28  parser.add_argument('-A', '--use_alias_as_name', help='Use Agent Alias as Agent N
29  parser.add_argument('-a','--agent_alias', help='Name of the agent to store monito
30  parser.add_argument('-m', '--module_prefix', help='PandoraFMS module prefix', def
31  parser.add_argument('--data_dir', help='PandoraFMS data dir (default: /var/spool/
32  parser.add_argument('-g', '--group', help='PandoraFMS destination group, default:
33  parser.add_argument('--tentacle_port', help='tentacle port', default=41121)
34  parser.add_argument('--tentacle_address', help='tentacle adress', default=None)
35  parser.add_argument('--as_agent_plugin', help='mode plugin', default=0, type=int)
```

次に、github や gitlab などのツールは、作成されたプラグインのコードを保存および維持するのに役立ちます。

## プログラミング言語

プラグインの開発には、プログラミング言語を使用する必要があります。Pandora FMS は、データが XML 形式で提供される限り、あらゆるタイプの言語をサポートします。Pandora FMS はそれらを理解しコンソールに表示できます。そのために、エージェント、モジュール、およびそれらの属性を含む XML でラベル構造に従います。

通常 Pandora FMS プラグインで最も一般的に使用されるスクリプト言語は、Python、Perl、BASH および Powershell であり、最初の 2 つはおそらく最も充実したものです。

Python と perl には Pandora FMS によって開発された Plugin Tools と呼ばれるプラグイン作成を容易にするツールがあります。このツールには、プラグイン作成を容易にし、自動化するように設計された多くの機能があるため、言語を選択する際の決定的な利点となる可能性があります。

## テスト環境構築に便利なツール

プラグインの開発には、通常、監視するサービスのインストール、アカウントなどのテスト環境が必要です。速度とシンプルさから環境作成に役立つ 2 つの技術としては docker と vagrant があります。もちろん、テストしたいサービスのプロジェクトでソリューションを用意する必要がありますが、いくつかのコマンドまたは小さな構成のみで実行できます。

## XML

Pandora FMS がプラグインから受信したデータを使用して表示するには、それらのデータを XML 形式に変換する必要があります。

Pandora FMS の XML データ形式を理解すると、プラグインを改善したり、プラグインを使用してカスタムエージェントを作成したり、カスタム XML ファイルを Pandora FMS データサーバに送信したりするのに役立ちます。

他の XML ドキュメントと同様に、データファイルは XML 宣言で開始する必要があります。

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

ただし、プラグインとそのデータが XML に基づいているのはエージェントプラグインのみです。エージェントプラグインのみがターミナル実行で XML を出力します。サーバプラグインは単純なデータのみを表示します。サーバプラグインにおける最も基本的なことは、正常な場合は 1 を出力し、逆に何らかのエラーが発生した場合は 0 を出力することです。

## エージェントとモジュール

プラグインが収集するデータは、エージェントとモジュールを介して Pandora FMS に表示できるため、データを表示する方法を示すことが重要です。たとえば、各エージェントは監視対象のデータベースで、モジュールとしてその情報を表すなどです。

エージェントとモジュールの xml 構造には、それらを構成するためのさまざまな属性があります。

データを送信するエージェントを定義する要素 agent\_data は、次の属性をサポートします。

- 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の名前 (Pandora FMS データベースに存在する必要があります)。空のままにし、サーバにデフォルトで設定されたグループがない場合、エージェントは作成されません。
- os\_name: エージェントが実行されているオペレーティングシステムの名前 (Pandora FMS データベースに存在する必要があります)。
- os\_version: オペレーティングシステムのバージョンを説明する任意の文字列。
- interval: エージェント実行間隔(秒単位)。
- version: エージェントバージョン文字列。
- timestamp: XML が生成されたタイムスタンプ(YYYY/MM/DD HH:MM:SS)[]
- agent\_name: エージェント名。
- timezone\_offset: タイムスタンプのオフセット(時間単位)[]UTC で動作している場合に便利です。
- address: エージェント IP アドレス(または FQDN)[]
- parent\_agent\_name: 親エージェントの名前。
- 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: エージェントのセカンダリグループ。
- custom\_id: エージェントカスタム ID[]
- url\_address: エージェントログイン URL[]

XML ヘッダ例:

```
<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'>
```

各モジュールにはモジュール要素が必要であり、次の要素をネストして定義できます。

- name: モジュール名。
- 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: モジュールタイプ(Pandora FMS データベースに存在するひつようがあります)。
- data: モジュールデータ。
- max: モジュール最大値。
- min: モジュール最小値。
- post\_process: 保存倍率。
- module\_interval: モジュール実行間隔(間隔をエージェントの実行間隔で割った値)。
- min\_critical: 障害状態の最小値。
- max\_critical: 障害状態の最大値。
- min\_warning: 警告状態の最小値。
- max\_warning: 警告状態の最大値。
- disabled: モジュールの無効化 有効化。無効化モジュールは処理されません。

- min\_ff\_event: 連続抑制回数 ([連続抑制回数](#) を参照)。
- status: モジュールの状態(NORMAL, WARNING または CRITICAL)[]ステータスが指定されている場合、障害およびアラートステータスの制限は無視されます。
- datalist: モジュールデータを datalist フォーマットで送信。(受け取った値ごとに 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 フォーマットを 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: 障害モジュール手順。
- warning\_instructions: 警告モジュール手順。
- unknown\_instructions: 不明モジュール手順。
- critical\_inverse: 障害閾値判定範囲を反転させる。 [0/1]
- warning\_inverse: 警告閾値判定範囲を反転させる。 [0/1]
- quiet: モジュールの静観モードを有効化。 [0/1]
- module\_ff\_interval: モジュールの連続抑制の間隔を定義。
- alert\_template: アラートテンプレートをモジュールに関連付けます。
- crontab: モジュール crontab を指定します。
- min\_ff\_event\_normal: 正常状態に変化する場合の連続抑制回数。
- min\_ff\_event\_warning: 警告状態に変化する場合の連続抑制回数。
- min\_ff\_event\_critical: 障害状態に変化する場合の連続抑制回数。
- ff\_timeout: 連続抑制回数のタイムアウト値。
- each\_ff: 連続抑制で状態ごとの回数定義を有効化。
- module\_parent: このモジュールの親になる同じエージェント内のモジュール名。
- ff\_type: 連続抑制でのカウンタ値の維持の有効化。 [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しいトークンが利用可能になりました。

- min\_warning\_forced: モジュールが存在する場合でも[]min\_warning を新しい指定値に強制し、min\_warning よりも優先されます。
- max\_warning\_forced: モジュールが存在する場合でも[]max\_warning を新しい指定値に強制し、max\_warning よりも優先されます。
- min\_critical\_forced: モジュールが存在する場合でも[]min\_critical を新しい指定値に強制し、min\_critical よりも優先されます。
- max\_critical\_forced: モジュールが存在する場合でも[]max\_critical を新しい指定値に強制し、max\_critical よりも優先されます。
- str\_warning\_forced: モジュールが存在する場合でも[]str\_warning を新しい指定値に強制し、str\_warning よりも優先されます。
- str\_critical\_forced: モジュールが存在する場合でも[]str\_critical を新しい指定値に強制し、str\_critical よりも優先されます。
- モジュールには、少なくとも 1 つの名前、型、およびデータ要素が必要です。

例:

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

```
<data>21</data>
</module>
```

XML データ ファイルには、任意の数の要素を含めることができます。

agent\_data タグを閉じることを忘れないでください!

リストモジュールもあり、1つの値だけではなく、複数の値を含むことができます。これらは文字列に役立ちます。次の例に示すように、このタイプのモジュールでは datalist タグを使用する必要があります。

```
<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>
```

各値にタイムスタンプを指定できます。

```
<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>
```

しきい値と単位の指定を含むその他の例:

```
<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>
```

## プラグインツール

Pandora FMS には、プラグインの開発を容易にする perl および python プログラミング言語で利用できる `plugintools` というツールがあります。これらは、プラグインの作成に必要なほとんどを処理を簡単に実現する機能が組み込まれています。

- エージェント作成。
- モジュール作成。
- `Tentacle` でのファイル送信。
- 設定ファイルの読み込み。

### Python 版 `plugintools`

<https://pypi.org/project/pandoraPlugintools/>

### Perl 版 `plugintools`

[https://github.com/pandorafms/pandorafms/blob/develop/pandora\\_server/lib/PandoraFMS/PluginTools.pm](https://github.com/pandorafms/pandorafms/blob/develop/pandora_server/lib/PandoraFMS/PluginTools.pm)

## パラメータと設定

プラグインを使用する際に便利なものとして、パラメータと設定ファイルがあります。これらを使って、サービス IP や対象となる認証情報のような常に変動するデータを定義したり、モジュールやエージェントの名前をカスタマイズしたりすることができます。プラグインの設定にパラメータを使用することは、設定ファイルを使用することに比べて利点と欠点があります。パラメータを使うとプ

プラグインの使用に多くの設定ファイルが必要なくなりますが、設定ファイルを使用した場合はプラグインの設定を確実にすばやく実行できます。各プラグインに最適なオプションを検討することが重要です。

## プラグインヘルプ

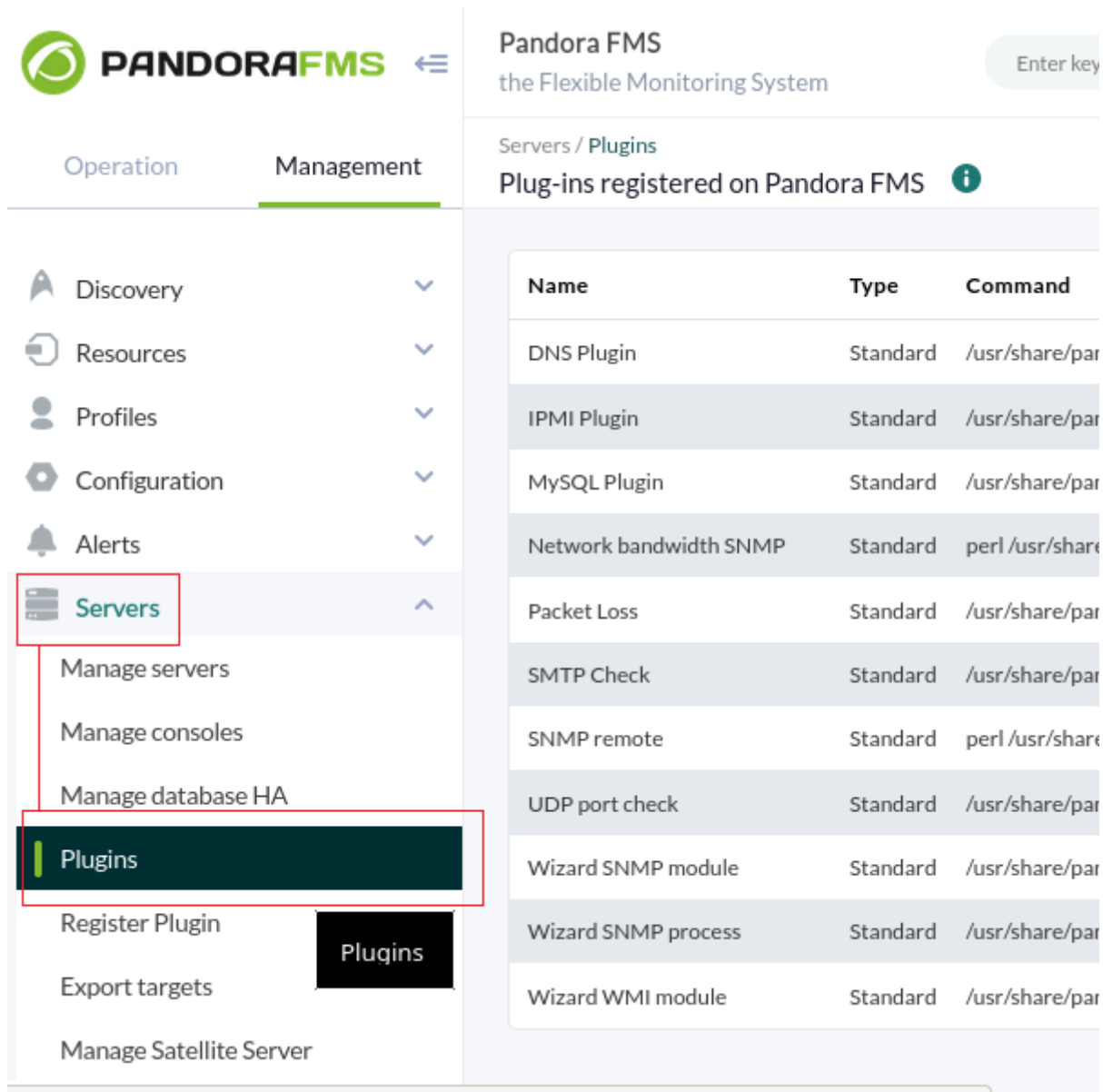
通常、プラグインで非常に便利なもう 1 つのオプションは、パラメータをヘルプに含めて、すべてのプラグインオプションとパラメータ、およびその設定方法を表示することです。これは、ソフトウェアに含まれるプラグインに関する小さなデジタルドキュメントのようなものです。

## Pandora FMS プラグイン設定

プラグインの設定は、これがエージェントプラグインかサーバプラグインかによって異なる場合があります。

### サーバプラグイン

サーバプラグインを使用して Pandora FMS から監視するには、サーバメニューの “プラグイン” に移動します。



The screenshot shows the Pandora FMS web interface. The top left features the Pandora FMS logo and navigation tabs for 'Operation' and 'Management'. The 'Management' tab is active, and the 'Servers' menu item is highlighted with a red box. A dropdown menu is open under 'Servers', with the 'Plugins' option also highlighted by a red box. A black box with the text 'Plugins' is positioned over the 'Register Plugin' option. The main content area displays a table titled 'Plug-ins registered on Pandora FMS' with columns for Name, Type, and Command.

Name	Type	Command
DNS Plugin	Standard	/usr/share/par
IPMI Plugin	Standard	/usr/share/par
MySQL Plugin	Standard	/usr/share/par
Network bandwidth SNMP	Standard	perl /usr/share
Packet Loss	Standard	/usr/share/par
SMTP Check	Standard	/usr/share/par
SNMP remote	Standard	perl /usr/share
UDP port check	Standard	/usr/share/par
Wizard SNMP module	Standard	/usr/share/par
Wizard SNMP process	Standard	/usr/share/par
Wizard WMI module	Standard	/usr/share/par

“追加(add)”をクリックします。好みの名前と説明を付ける必要があります。

コマンドとして、プラグインパスに実行コマンドを入力する必要があります。

サーバプラグインを利用するための推奨パスは次のとおりです。

```
/usr/share/pandora_server/util/plugin/
```

プラグインパスには、コマンドを入力する必要があります。



**Command****Plugin command****Plugin parameters**

プラグインパラメータでは、マクロ `_field_` に続いてそれらを入力します。

–SERVER

**Description** (\_field1\_)**Default value** (\_field1\_)**Hide value****Command****Plugin command****Plugin parameters**

–SUBJECT

**Description** (\_field4\_)**Default value** (\_field4\_)**Hide value**

–BODY

Description (_field5_)	<input type="text" value="body"/>	Default value (_field5_)	
------------------------	-----------------------------------	--------------------------	--

Resources / Manage agents / Agent plugins

Agent setup view ( nodo-1-pandorafms agent )

^ Add

Advanced i

























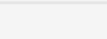



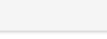



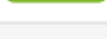


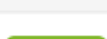

New plugin

```
module_plugin /etc/pandora/plugins/MyMonitor.pl /etc/pandora/plugins/MyMonitor.conf
```

Plugins	Policy	Update	Enable
pandora_df_used			

## Pandora FMS でのプラグインの表示

プラグインデータの表示は通常、エージェント表示から行われます。通常、データ構造はエージェントとモジュールにて定義されるため、プラグインがエージェントとモジュールを作成するのは普通のことです。したがって、これらのデータはエージェント自体の表示メニューに表示できます。

	Sybase ASE Backup server status	Status of the backup server		N/A - N/A	0	 	1 minutos 07 segundos
	Sybase ASE status	The status of the Adaptive Server		N/A - N/A	1	 	1 minutos 07 segundos
	Sybase CPU Busy	Time that the Adaptive Server was busy during the previous p...		N/A - N/A	3.1 %	 	1 minutos 07 segundos
	Sybase Locks Remaining	Number of locks available for use		N/A - N/A	10,000	 	1 minutos 07 segundos
	Sybase Memory Use	Sybase memory use in MB		N/A - N/A	0.2 MB	 	1 minutos 07 segundos
	Sybase Mirror Devices	Number of devices that are configured to be mirrored but are...		N/A - N/A	0	 	1 minutos 07 segundos
	Sybase Suspect Databases	Number of databases marked as suspect or unusable by the Ada...		N/A - N/A	0	 	1 minutos 07 segundos
	Sybase_active_locks	Number of active locks		N/A - N/A	0	 	1 minutos 07 segundos
	Sybase_total_locks	Total number of locks		N/A - N/A	10,000	 	1 minutos 07 segundos
	Sybase_users_connected	Number of users connected		N/A - N/A	1	 	1 minutos 07 segundos

仮想サーバマシンごとにエージェントを作成する Xenserver プラグインなど、多くのプラグインでは複数のエージェントが作成される場合があります。この場合、エージェントのプレフィックスに例えば xen- を定義できるため、エージェントを表示した際に名前にその構造が含まれていることがわかり、それらのエージェントがその特定のプラグインに由来するものであることがわかります。

## PSPZ2

pspz2 ファイルは、プラグインとモジュールの定義を含む plugin\_definition.ini ファイルの 2 つのファイルを zip 圧縮したファイルです。

## PLUGIN REGISTRATION

### PLUGIN REGISTRATION



This extension makes registering server plugins an easier task. Here you can upload a server plugin in .pspz zipped format.

Please refer to the official documentation on how to obtain and use Server Plugins.

You can get more plugins in our Public Resource Library

Seleccionar archivo Ninguno a...hivo selec.

Upload >

このファイル形式は、プラグインのインストールを簡素化するため、プラグインをパッケージ化するとき非常に便利です。この形式を使用するとPandora FMS コンソールの“プラグイン登録(plugins registration)”画面からプラグインをよりすばやくインストールできます。

プラグインの plugins\_definition.ini ファイルの例:

```
[plugin_definition]
name = Pandora Azure Storage
description = "Take data from azure storage"
timeout = 20
filename = pandora_azure
execution_command =
execution_postcommand =
parameters = -c _field1_ -agent_name _field2_ -g _field3_
plugin_type = 0
total_modules_provided = 0
total_macros_provided = 8
[macro_1]
hide = 0
description = "Conf path (obligatory)"
help = "Path conf"
value =
[macro_2]
hide = 0
description = "Agent name (optional)"
help = "Name of the agent"
value = _agentname_
[macro_3]
hide = 0
description = "group (optional)"
help = "Pandora FMS Target Group "
```

```
value =
```

filename: ファイル .pspz に含まれるスクリプトと同じ名前にする必要があります。以前は < script\_file > という名前でした。この例では、ssh\_pandoraplugin.sh という名前のシェルスクリプト (.sh 形式) です。

plugin\_type: 0 は Pandora FMS 標準のプラグイン、1 は Nagios プラグインです。

total\_modules\_provided: .ini ファイルの次のセクションで定義されているモジュールの数を指定します。少なくとも 1 つ設定します (少なくとも例で使用)。

execution\_command: 使用する場合は、script の前に置かなければなりません。これは、java - jar のようなインタプリタかもしれません。プラグインは Pandora FMS プラグインサーバから次のようなコードで呼び出され実行されます: java -jar < plugin\_path/ plugin\_filename

execution\_postcommand: 使用すると、< plugin\_filename > の後にプラグインに送信される追加パラメータを定義できます。ユーザからは見えません。

total\_macros\_provided: プラグインが持つ動的マクロの数を定義します。

macro\_value: そのモジュールの値を動的マクロで定義し、存在しない場合はデフォルト値をとります。

また、動的マクロごとにセクションを作成しなければいけません。例:

```
[macro_<N>]
hide = 0
description = <your_description>
help = <text_help>
value = <your_value>
```

## 互換性

プラグインはできるだけ互換性を持たせることが重要です。ある種のプラグインは、MS Windows® のような特定の OS でしか利用できないサービス特有のものであったりします。しかし、避けられない制限の中で、できるだけ多くのシナリオをカバーできる互換性をプラグインに持たせることが重要です。

OSによっては、他の OS には存在しないパッケージがあり、コンパイルしたプラグインの実行でエラーが発生することがあるため、必要なプラグインの依存関係がすべてインストールされている Rocky Linux 7 の環境でプラグインをコンパイルすることをお勧めします。

## エラー制御

適切なエラー制御を行うと、プラグインの潜在的な将来起こりうるエラーを検出し、迅速に原因を見つけることができるため、長期にわたってプラグインを維持するのに役立ちます。

また、エラーを正確に検出するのも役立ちます。エラー制御がないと、特定のケースではエラー出力があまりにも一般的になり、問題が何であるかを理解するのが難しくなる可能性があるためです。

### プラグインのテストと検証

プラグインを開発したら、環境でテストを実行してその動作を確認すると良いです。複数のテスト環境でテストできると良いです。

良いテストプロセスは、起こりうるエラーを解決し、長期にわたってプラグインの動作をより強固にするのに役立ちます。よく言われる「転ばぬ先の杖」です。

### プラグインの公開

プラグインを公開する際には、ある点を考慮することが重要です。

### プラグインのドキュメント

プラグインの中には、最初は理解しにくいものがあります。いくつかのプラグインにはいくつかのオプションがあり、それらを理解するのは少し難しいかもしれません。

良いドキュメントは、それを理解するのに役立ちますし、将来的にはプラグイン制作者自身が、時間が経つにつれて忘れてしまったプラグインのある事柄を思い出すのに役立つことさえあります。

Pandora FMS には、ドキュメントのメンテナンスを容易にするためのオンラインクイックガイドのシステムがあり、以下のリンクからアクセスできます。


<https://pandorafms.com/guides/public/shelves/en-quick-guides>

クイックガイド形式のドキュメントに代わるものとしてPDF 形式のものもあります。

### ライブラリへのプラグインのアップロード

Pandora FMS ライブラリにプラグインをアップロードする方法は非常に簡単です。ライブラリの“my plugins” セクションにある“upload new module” セクションにアクセスするだけです。

# Upload new module

 Click [here](#) to go to help.

## Name

## Description

\*Click the Add Media button to upload files.

 Add Media

Visual Text

Paragraph 

## Categories

\* You can select more than one option by pressing ctrl + click

Application monitoring  
Databases category  
  msSQLServer  
  Oracle category  
Document Management  
Email and Groupware

## Tags

\* You can put several tags by separating them with commas.


タイトル、プラグインの説明、zip 形式で圧縮されたプラグイン、ドキュメント、カテゴリ、タグなど、エントリーのすべてのセクションを設定したら、submit をクリックしてプラグインを公開することができます。

ライブラリにアップロードされたすべてのプラグインは、最終的なリリース前にレビューが行われます。レビュー時間がかかります。


プラグインが公開されると「latest uploads」に表示されるようになります。





### Recent uploads

Kafka monitoring 


Synology plugin

Pandora openweathermap 


Pandora Azure Event Hubs 


Pandora Azure PostgreSQL 

iLert integration

Plugin Cisco Inventory configuration 

Plugin IMAP

PandoraFMS IBM i Enterprise plugin (formerly known as AS400) 

Pandora SAP HANA Database 

## プラグインのレビューとメンテナンスに役立つグッドプラクティス

いくつかのグッドプラクティスは、タスクに費やす時間を減らし、プラグインの作成プロセスやプラグインのメンテナンスのさまざまな段階でよくあるエラーを防止 回避し、それらのレビューを容易にするのに役立ちます。これらは以下の通りです。

- システム上でこれまで行ったことのあるプラグインの設定を行う必要がある場合に備えて、正しく動作する設定を十分に文書化して残しておくこと。
- プラグインの使い方をイメージしやすいように、実際に実行した例をドキュメント化すること。
- プラグインのコードにコメントを書くことで、より深く、あるいはより速く理解することができます。
- コードの読みやすさを優先します。複雑であればあるほど、それに対処するための時間とリソースが必要になります。
- コードのテストをします。プラグインの動作をテストして、すべてが問題ないことを確認することが重要です。エラーを時間内に発見して修正することで、将来的な問題を回避することができます。

[Pandora FMS ドキュメント一覧に戻る](#)