



# エージェントプラグイン開発



m:

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

Permanent link:

[https://pandorafms.com/manual/!current/ja/documentation/pandorafms/technical\\_reference/06\\_anexo\\_agent\\_plugins](https://pandorafms.com/manual/!current/ja/documentation/pandorafms/technical_reference/06_anexo_agent_plugins)

2024/03/18 21:07



# エージェントプラグイン開発

[Pandora FMS ドキュメント一覧に戻る](#)

## エージェントプラグイン開発

### エージェントプラグインの基本機能

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で実行され、いくつかの特別な機能があります。

- それぞれのプラグインの実行では、一つまたは複数のモジュールの値を返します。出力は後述する XML フォーマットである必要があります。
- ローカルマシンのリソースおよび、リモートで他のマシンのリソースの両方にアクセスすることができます。
- Pandora ソフトウェアエージェントがインストールされている OS でサポートしている、任意のプログラミング言語を利用することができます。
- プラグインを実行するために必要な依存プログラムは、Pandora ソフトウェアエージェントを実行するのと同じマシンにインストールされている必要があります。

エージェントプラグインは、“自動検出タスク”のようなことを実行でき、プラグインは一度の実行で複数のモジュールを返すことができ、また、実行ごとに異なる数を返すことができます。

UNIX および Linux では、プラグインの終了ステータスは 0 でなければなりません。そうでないと、プラグインの出力は無視されます。

### エージェントプラグイン開発の例

簡単なプラグインの例を説明します。このエージェントプラグインは、ファイルシステムの利用率を返します。プログラムコードは次の通りです。

```
#!/usr/bin/perl

use strict;

sub usage() {
    print "\npandora_df.pl v1r1\n\n";
    print "usage: ./pandora_df\n";
    print "usage: ./pandora_df tmpfs /dev/sda1\n\n";
}
```

```
# Retrieve information from all filesystems
my $all_filesystems = 0;

# Check command line parameters
if ($#ARGV < 0) {
    $all_filesystems = 1;
}

if ($ARGV[0] eq "-h") {
    usage();
    exit(0);
}

# Parse command line parameters
my %filesystems;
foreach my $fs (@ARGV) {
    $filesystems{$fs} = '-1%';
}

# Retrieve filesystem information
# -P use the POSIX output format for portability
my @df = `df -P`;
shift (@df);

# No filesystems? Something went wrong.
if ($#df < 0) {
    exit 1;
}

# Parse filesystem usage
foreach my $row (@df) {
    my @columns = split (' ', $row);
    exit 1 if ($#columns < 4);
    $filesystems{$columns[0]} = $columns[4] if (defined
($filesystems{$columns[0]}) || $all_filesystems == 1);
}

while (my ($filesystem, $use) = each (%filesystems)) {

    # Remove the trailing %
    chop ($use);

    # Print module output
    print "<module>\n";
    print "<name><![CDATA[" . $filesystem . "]]></name>\n";
    print "<type><![CDATA[generic_data]]></type>\n";
    print "<data><![CDATA[" . $use . "]]></data>\n";
    print "<description>% of usage in this volume</description>\n";
    print "</module>\n";
}

exit 0;
```

コードの重要部分は、利用方法を表示する関数です。

```
sub usage() {
    print "\npandora_df.pl v1r1\n\n";
    print "usage: ./pandora_df\n";
    print "usage: ./pandora_df tmpfs /dev/sda1\n\n";
}
```

この関数では、プラグインのバージョンと利用方法を説明します。これはとても重要で、パラメータ無しや `-h`, `-help` といったオプションをつけてプラグインを実行したときに表示すべきです。この例では `-h` オプションをつけて実行した場合、次の処理でそれを判断します。

```
if ($ARGV[0] eq "-h") {
    usage();
    exit(0);
}
```

プラグインから返される値としては、ファイルシステムから一回データが収集されることがわかると思います。データを含めた XML の書式を生成し標準出力に出力します。この処理は次の部分で行われています。

```
while (my ($filesystem, $use) = each (%filesystems)) {

    # Remove the trailing %
    chop ($use);

    # Print module output
    print "<module>\n";
    print "<name><![CDATA[" . $filesystem . "]]></name>\n";
    print "<type><![CDATA[generic_data]]></type>\n";
    print "<data><![CDATA[" . $use . "]]></data>\n";
    print "<description>% of usage in this volume</description>\n";
    print "</module>\n";
}
```

このプラグインの出力結果は、次のようになります。

```
<module>
<name><![CDATA[tmpfs]]></name>
<type><![CDATA[generic_data]]></type>
<data><![CDATA[0]]></data>
<description>% of usage in this volume</description>
</module>
<module>
<name><![CDATA[/dev/mapper/VolGroup-lv_home]]></name>
<type><![CDATA[generic_data]]></type>
<data><![CDATA[26]]></data>
<description>% of usage in this volume</description>
</module>
```

```
<module>
<name><![CDATA[/dev/sda9]]></name>
<type><![CDATA[generic_data]]></type>
<data><![CDATA[34]]></data>
<description>% of usage in this volume</description>
</module>
```

このプラグインから返されるモジュール数は、設定したファイルシステムの数に依存し、実行ごとに変更することができます。

この XML がソフトウェアエージェントが生成する全体の XML に追加され、Pandora サーバへ送信され、データサーバにて処理されます。

## トラブルシューティング

Pandora FMS がエージェントプラグインを認識できないと、期待した値が取得できなかったりエージェントが動作しません。その場合の確認ポイントを以下に示します。

### pandora\_agent.conf のドキュメント確認

ソフトウェアエージェントでは、プラグインの正しいパスがこのファイル内に書かれている必要があります。

例:

```
module_plugin /etc/pandora/plugins/MyMonitor.pl
/etc/pandora/plugins/MyMonitor.conf 2> /etc/pandora/plugins/MyMonitor.err
```

MyMonitor.pl はエージェントプラグインで、MyMonitor.conf は引数として渡される設定ファイルです。また、MyMonitor.errはプラグインを実行した場合に出力されるエラーを書き込むファイルです。

### pandora\_agent\_daemon の再起動

Pandora FMS の (Enterprise 版ではなく) OSS 版 を利用している場合は、ソフトウェアエージェントは 5分ごとにプラグインを実行します。次の実行まで待てない場合は、コマンドラインからソフトウェアエージェントを再起動します。

例:

```
/etc/init.d/pandora_agent_daemon restart
```

### プラグインのパーミッションの確認

プラグインおよび利用するファイルは、リード、ライト、実行の権限を正しく設定する必要があります。Unix であれば、次のようにします。

```
chmod 755 <プラグインファイル>
```

## 出力の確認

エラーを見つける簡単な方法は、コマンドラインでプラグインを手動実行することです。注意して出力結果を確認してください。例えば、次のように実行します。

```
popeye:/etc/pandora/plugins # ./pandora_df
<module>
<name><![CDATA[/dev/sda2]]></name>
<type><![CDATA[generic_data]]></type>
<data><![CDATA[19]]></data>
<description>% of usage in this volume</description>
</module>
<module>
<name><![CDATA[udev]]></name>
<type><![CDATA[generic_data]]></type>
<data><![CDATA[1]]></data>
<description>% of usage in this volume</description>
</module>
```

## XMLの確認

プラグインが出力する XML は、正しい XML の書式である必要があります。XML はまた、正しく整形されている必要があります。それを確認するためには、コマンドラインから次の手順を実行します。

1. プラグインの出力を xml ファイルに落とします。: `./Plugin.pl > Plugin.xml`
2. xml ファイルの書式をチェックします。: `xmllint Plugin.xml`

## デバッグモード

`pandora_agent.conf` ファイル内で、`debug` の値を 0 から 1 に変更することで、デバッグモードを有効にすることができます。これを行うと、ソフトウェアエージェントがプラグインを実行した結果が、エージェントのその他の情報とあわせて xml ファイルに保存されます。

ファイル名は、エージェント名と `.data` を含むもので、`/tmp` ディレクトリに保存されます (pandora エージェントのログは、`/var/log/pandora/pandora_agent.log` を確認してください)。内容を確認し、期待したプラグインの実行結果が含まれているかどうかを見ます。

デバッグモードを有効にすると、エージェントは 1 回の実行

で終了します。

## フォーラム

それでもまだうまくいかない場合は、我々の [フォーラム](#) (日本語フォーラムは [こちら](#)) で聞いてみてください。

[Pandora FMS ドキュメント一覧に戻る](#)