



サーバプラグイン開発



<https://pandorafms.com/manual/!775/>

Permanent link:

https://pandorafms.com/manual/!775/ja/documentation/pandorafms/technical_reference/05_anexo_server_plugins_development

2022/03/18 21:03

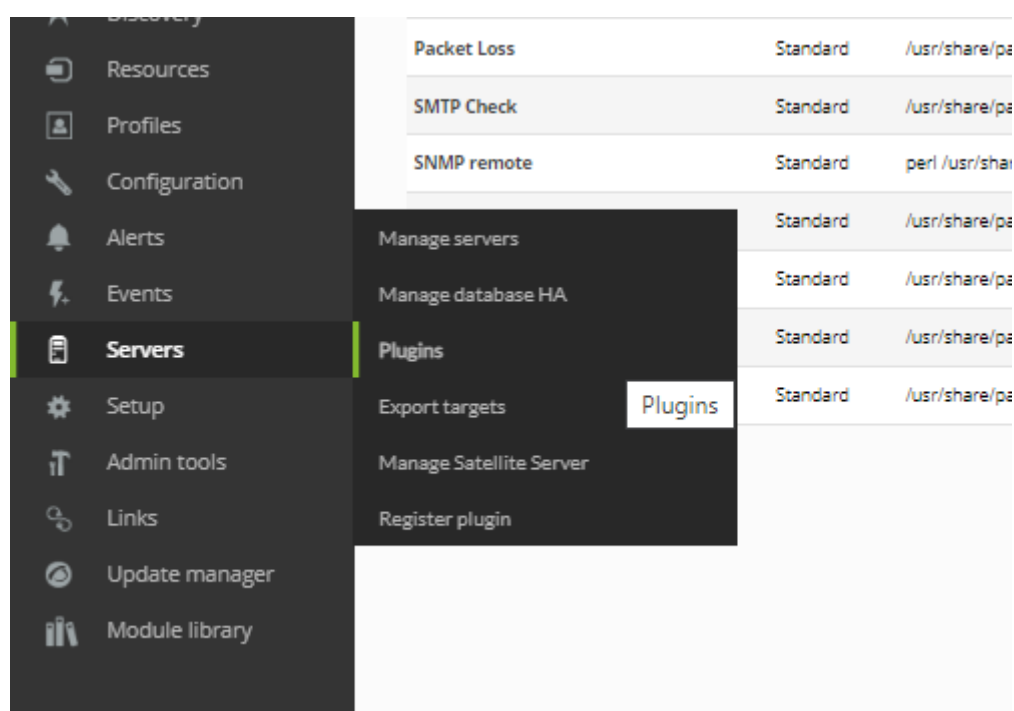


サーバプラグイン開発

[Pandora FMS ドキュメント一覧に戻る](#)

サーバプラグイン開発

サーバプラグインの基本機能



サーバプラグインは、Pandora FMS プラグインサーバにより実行されるため、次に示す特別な機能を備えている必要があります。

- プラグインの実行では単一の値を返す必要があります。サーバプラグインは、プラグインモジュールによって実行されるためです。
- リモートで監視対象のリソースにアクセスできる必要があります。
- Pandora サーバのインストール先の OS がサポートす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。
- プラグインの実行に必要な依存ソフトウェアは、Pandora サーバを実行するのと同じのマシンにインストールされている必要があります。

リモートサーバプラグインを使った監視に関する詳細は、[こちら](#)を確認してください。そこには、簡単な例と、モジュールとエージェントがどのように機能するかが記載されています。この記事では、サーバプラグインの作成について詳しく示します。

サーバプラグイン開発

次に、Pandora FMS のサーバプラグインの例を説明します。

以下のプラグインは、インタフェースの入出力トラフィックの合計を返します。データは SNMP で取得します。

プラグインのコードは次の通りです。

```
#!/usr/bin/perl -w

use strict;
use warnings;

sub get_param($) {
    my $param = shift;
    my $value = undef;

    $param = "-" . $param;

    for(my $i=0; $i< $#ARGV; $i++) {

        if ($ARGV[$i] eq $param) {
            $value = $ARGV[$i+1];
            last;
        }

    }
    return $value;
}

sub usage () {
    print "iface_bandwidth.pl version v1r1\n";
    print "\nusage: $0 -ip <device_ip> -community <community> -ifname
<iface_name>\n";
    print "\nIMPORTANT: This plugin uses SNMP v1\n\n";
}

#Global variables
my $ip = get_param("ip");
my $community = get_param("community");
my $ifname = get_param("ifname");

if (!defined($ip) ||
    !defined($community) ||
    !defined($ifname) ) {
    usage();
    exit;
}

#Browse interface name
```

```
my $res = `snmpwalk -c $community -v1 $ip .1.3.6.1.2.1.2.2.1.2 -On`;

my $suffix = undef;

my @iface_list = split(/\n/, $res);

foreach my $line (@iface_list) {

    #Parse snmpwalk line
    if ($line =~ m/^(([\d|\.]*) = STRING: (.*)$/) {
        my $aux = $1;

        #Chec if this is the interface requested
        if ($2 eq $ifname) {

            my @suffix_array = split(/\./, $aux);

            #Get last number of OID
            $suffix = $suffix_array[$#suffix_array];
        }
    }
}

#Check if iface name was found
if (defined($suffix)) {
    #Get octets stats
    my $inoctets = `snmpget $ip -c $community -v1
.1.3.6.1.2.1.2.2.1.10.$suffix -OUevqt`;
    my $outoctets = `snmpget $ip -c $community -v1
.1.3.6.1.2.1.2.2.1.16.$suffix -OUevqt`;

    print $inoctets+$outoctets;
}
```

このコードの重要な説明は、利用方法の関数にあります。

```
sub usage () {
    print "iface_bandwidth.pl version v1r1\n";
    print "\nusage: $0 -ip <device_ip> -community <community> -ifname
<iface_name>\n";
    print "\nIMPORTANT: This plugin uses SNMP v1\n\n";
}
```

この関数では、バージョンおよびプラグインの利用方法を説明しています。これはとても重要で、パラメータを指定せずにプラグインを実行するかまたは `-h` や `-help` を指定してプラグインを実行した場合に常に表示される必要があります。

プラグインが返す値に関しては、最後から 2 行目に標準出力に表示する以下のようなコマンドがあります。

```
print $inocets+$outocets;
```

見ての通り、プラグインが返す値は単一のデータで、これを Pandora プラグインサーバが関連モジュールにデータとして追加します。

このプラグインを実行するには Pandora サーバを実行するマシンに *snmpwalk* および *snmpget* コマンドをインストールする必要があります。

プラグインの手動登録

PLUGIN REGISTRATION

General

Name

Plugin type

Standard

Max.

timeout

15 seconds



Description

- 名前(Name)

プラグインの名前です。

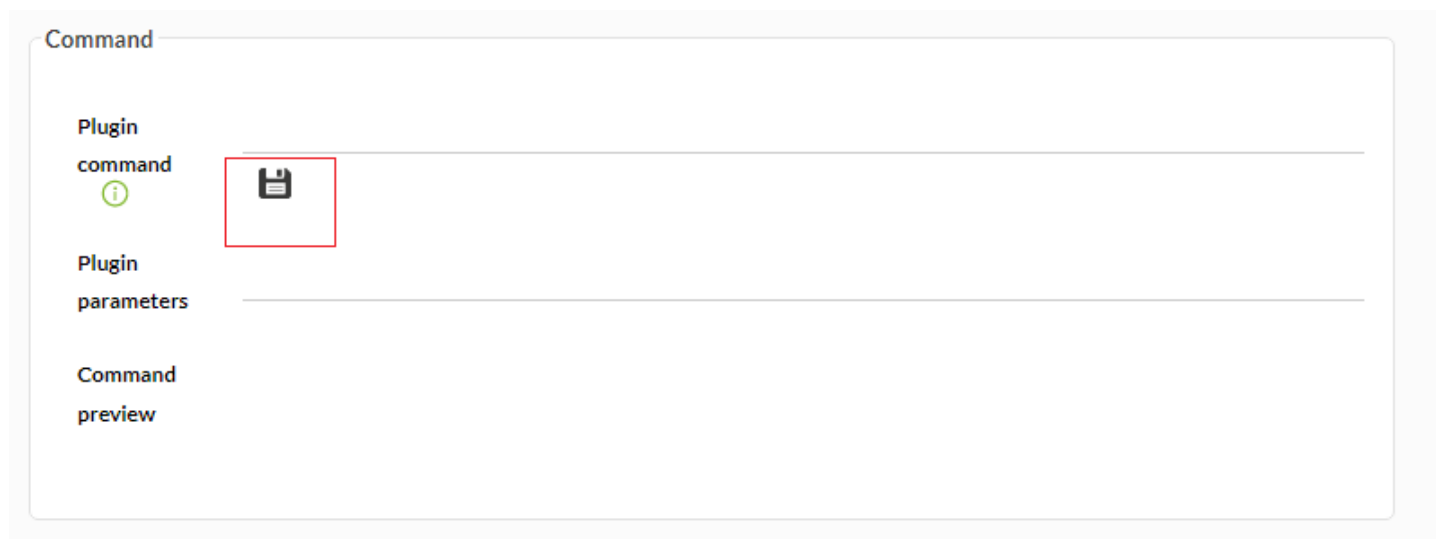
- プラグインタイプ(Plugin type)

プラグインには、標準プラグインと Nagios の2種類があります。標準プラグインは、アクションを実行してパラメータを受け取るスクリプトです。Nagios プラグインは、その名前が示すように Nagios で利用できる形式のプラグインです。主な違いは、テストが成功したかどうかを Nagios プラグインはエラーレベルで示すことです。

もし Nagios プラグインを使いたい場合で(OK/NG ではなく)データを取得したい場合は、Nagios プラグインを “標準” モードで利用します。

- 最大タイムアウト(Max. timeout)

プラグインの実行時間制限です。この時間内に応答がない場合は、モジュールを不明として処理し、その値は更新されません。プラグインを使用してモニタリングを実装する場合、これは非常に重要な要素です。そのため、プラグインの実行にかかる時間がこの数値より大きいと値を取得できません。この値は、プラグインとして使用されるスクリプトや実行ファイルが値を返すまでにかかる時間よりも常に大きくなければなりません。何も設定しない場合は、plugin_timeout の設定値が使われます。



The image shows a configuration interface for a plugin. It is titled "Command" and contains three main sections: "Plugin command", "Plugin parameters", and "Command preview". In the "Plugin command" section, there is a save icon (a floppy disk) highlighted with a red rectangular box. There is also an information icon (a lowercase 'i' in a circle) to the left of the "Plugin command" label.

- プラグインコマンド(Plug-in command)


プラグインコマンドのパスです。デフォルトインストールでのプラグインディレクトリは、/usr/share/pandora_server/util/plugin/ です。ただし、任意の場所を指定することができます。ここでは、フィールドに /usr/share/pandora_server/util/plugin/udp_nmap_plugin.sh を指定しています。

サーバは、このスクリプトを実行します。そのため、読み取りおよび実行権限がある必要があります。

- プラグインパラメータ(Plug-in parameters)

コマンドのパラメータ文字列で、コマンドの引数としてしているものです。パラメータフィールドには `{}_field1_{}_field2_ ... _fieldN_` といったマクロが使えます。

Macro parameters

Description	Default value
<input type="text" value="(_field1)"/>	<input type="text" value="(_field1)"/>
Hide value 	<input type="checkbox"/>
Help <input type="text" value="(_field1)"/>	<div style="border: 1px solid #ccc; height: 100px; width: 100%;"></div>

[Add macro !\[\]\(bf453bf87847ae2129e7d24917b16f87_img.jpg\)](#)

[Create !\[\]\(47a3b08df1ba32720cad69432dbd8375_img.jpg\)](#)

Pandora FMS v7.0NG.764 - OUM 764 - MR 56

Page generated on 2022-09-13 00:22:54















- パラメータマクロ(Parameters macros)

プラグインパラメータフィールドで使うマクロを追加することができます。このマクロは、モジュール設定の通常のテキストフィールドとして表示されます。

プラグインマクロ

Pandora FMS サーバにデフォルトでインストールされている DNSプラグイン について考えてみます。

Plug-ins registered on Pandora FMS

Name	Type	Command	Op.
DNS Plugin	Standard	<code>/usr/share/pandora_server/util/plugin/dns_plugin.sh</code>	 
IPMI Plugin	Standard	<code>/usr/share/pandora_server/util/plugin/ipmi-plugin.pl</code>	 
MySQL Plugin	Standard	<code>/usr/share/pandora_server/util/plugin/mysql_plugin.sh</code>	  
Network bandwidth SNMP	Standard	<code>perl /usr/share/pandora_server/util/plugin/pandora_snmp_bandwidth.pl</code>	 
Packet Loss	Standard	<code>/usr/share/pandora_server/util/plugin/packet_loss.sh</code>	  
SMTP Check	Standard	<code>/usr/share/pandora_server/util/plugin/SMTP_check.pl</code>	 

このプラグインを使用するとWebドメインとそれに対応する IP を DNS サーバでチェックできます。

- `-i` : 既知のドメインの IP アドレス。
- `-d` : 対応する Web ドメイン。
- `-s` : チェックする DNS サーバ。

これらの3つのパラメータで、新しいプラグインの手動登録に進み、“New DNS Plugin” という名前で、説明に前の概要をコピーします。さらに監視のためにモジュールが `false(0)` または `true(1)` を取得する必要があることを示します。このタイプのデータはブール値とも呼ばれPandora FMS では `generic_proc` と呼ばれます。セクション マクロパラメータ(Macro parameters) に、3つのマクロフィールド `field1`, `field2` および `field3` を追加します。

`_field1_` では、パラメーター `-i` に関する説明を追加します。同様に、`-d` および `-s` の説明を追加します。デフォルト(default) の値を空のままにし、それぞれのヘルプ(Help)に必要なテキストを入力します。

テキストボックスに プラグインコマンド を入力します。

```
/usr/share/pandora_server/util/plugin/dns_plugin.sh
```

プラグインパラメータ(Plugin parameters) ボックスに入力します。

```
-i _field1_ -d _field2_ -s _field3_
```

このプラグインをモジュールで利用する際には、それぞれのフィールドはモジュールのパラメータに置き換えられ DNS Plugin が実行されます。

```
/usr/share/pandora_server/util/plugin/dns_plugin.sh -i _field1_ -d _field2_ -s  
_fiel
```

動作

マクロは `_field1_`, `_field2_`, (...), `_fieldN_` のように機能しますが、モジュールとそれらのモジュールを含むエージェント両方からの特別な値を利用します。

`_fieldN_` のデフォルト値が空白のままになっている前の章の例に戻ります。 `_field2_` のデフォルト値を編集し、マクロ `_module_` を追加します。

モジュールまたはコンポーネントがそのサーバプラグインを使用している場合、更新できないロックアイコンが表示されます。

`_module_` マクロは、プラグインが使用するモジュールを返し、そのモジュールが作成されるときにユーザまたはポリシーに追加されます。 これを確認するには “DNS verify” と呼ばれる新しいエージェントを作成し、プラグインサーバモジュールの新規作成(Create a new plugin server module) オプションを使用して新しいモジュールを作成します。



新しいモジュールの編集フォームで、プラグインで “New DNS Plugin” をリストから選択すると、`_module_` マクロが次のように表示されます。

Type ?

Warning threshold
 Min.
 Max.
 Inverse interval

Critical threshold
 Min.
 Max.
 Inverse interval

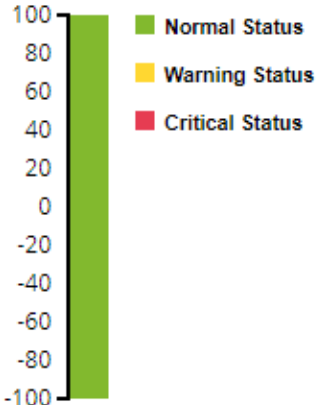
Historical data

Plugin ? This plugin is used to check if a specific domain return a specific IP address.

IP

Domain

Server DNS



収集するデータのタイプは“Generic boolean”である必要があることを忘れないでください。また、新しいモジュールの名前、指定した DNS サーバで IP アドレスを確認する Web ドメインを追加します。保存して機能することを確認します。

この方法の良い点は、モジュールとしてチェックする必要のある Web ドメインがあり、名前によってさまざまなコンポーネント(ダッシュボード、レポートなど)で簡単に識別できることです。

マクロ一覧

- `_agent_`: エージェントの別名。別名が割り当てられていない場合は、エージェント名が使用されます。
- `_agentalias_`: エージェントの別名。
- `_agentdescription_`: エージェントの説明。
- `_agentstatus_`: エージェントの現在の状態。
- `_agentgroup_`: エージェントグループ名。
- `_agentname_`: エージェント名。(`_agent_` も参照してください)
- `_address_`: エージェントのアドレス。
- `_module_`: モジュール名。
- `_modulegroup_`: モジュールグループ名。
- `_moduledescription_`: モジュールの説明。
- `_modulestatus_`: モジュールの状態。
- `_moduletags_`: モジュールタグに関連付けられた URL
- `_id_module_`: モジュールの ID
- `_id_agente_`: エージェントの ID Pandora FMS コンソールへアクセスする URL を生成するのに便利です。
- `_id_group_`: エージェントグループの ID

- `_interval_`: モジュールの実行間隔。
- `_target_ip_`: モジュールのターゲット IP
- `_target_port_`: モジュールのターゲットポート番号。
- `_policy_`: モジュールが所属するポリシー名。(適用される場合)
- `_plugin_parameters_`: モジュールのプラグインパラメータ。
- `_email_tag_`: モジュールタグに関連付けられた Email
- `_phone_tag_`: モジュールタグに関連付けられた電話番号。
- `_name_tag_`: モジュールに関連付けられたタグ名。

PSPZ パッケージ

Pandora サーバプラグイン Zip ファイル (.pspz)

Pandora FMS 3.0では、プラグインおよび新しいプラグインを(プラグインに依存した、モジュールのライブラリのように)使用するモジュールを登録する新しい方法があります。基本的には、以下で説明する.pspzフォーマットのファイルをアップロードする管理コンソール拡張機能です。システムでファイルを読み込み、展開後、バイナリファイルとスクリプトをシステムにインストールし、Pandora FMSのモジュールライブラリへのプラグインの登録と.pspzファイル中で定義されたすべてのモジュール生成を行います。

この節では、.pspzファイルの作成方法について説明します。

パッケージファイル

.pspz ファイルは2つのファイルからなるzipファイルです。

`plugin_definition.ini`: プラグインとモジュールの仕様を含むファイル。ファイル名(大文字と小文字は区別されます)は固定です。

`<script_file>`: プラグインスクリプトもしくはバイナリファイルそのもの。ファイル名は自由につけることができます。 [ここ](#) に .pspzファイルのサンプル(ファイル名は.zipに変更されています)があります。

`plugin_definition.ini` の構造

ヘッダー/定義

これがオプションセクションを持つ標準的なINIファイルです。最初のセクションは最も重要で、セクション名は `plugin_definition` 固定です。以下に例を示します。

```
[plugin_definition]
name = Remote SSH exec
filename = ssh_pandoraplugin.sh
```

```
description = This plugin execute remotely any command provided
timeout = 20
ip_opt = -h
execution_command =
execution_postcommand =
user_opt = -u
port_opt =
pass_opt =
plugin_type = 0
total_modules_provided = 1
```

filename: 先に<script_file>で参照されていた.pspzファイルに含まれるスクリプト名と同じ値となります。このサンプルでは“ssh_pandoraplugin.sh”というシェルスクリプトです。

*_opt: プラグインに登録するオプションを指定します。Pandora FMSコンソールで“手動で”プラグインに登録するフォームに表示されるものと同様のものです。

plugin_type: 標準的なPandora FMSプラグインの場合は0。Nagiosタイププラグインの場合は1を指定します。

total_modules_provided: 以下で定義するモジュールの数を指定します。必要最小限の値を指定してください(XXX)

execution_command: 値が設定された場合、スクリプトの前に値が追加されます。“java -jar”といったインタプリタを指定する際に設定されます。したがって、プラグインはPandoraFMSプラグインサーバから“java -jar <plugin_path>/<plugin_filename>”と実行されます。

execution_postcommand: 値が設定された場合、追加パラメータがplugin_filenameの後にプラグインに渡されます。追加パラメータはユーザには見えません。

モジュール定義/ネットワークコンポーネント

前のセクションの total_modules_provided で定義したものと同じ数のモジュールを定義します。

もし4個のモジュールがある場合、セクション名は module1, module2, module3 および module4 でなければいけません。

以下にモジュール定義の例を示します。

```
[module1]
name = Load Average 1Min
description = Get load average from command uptime
id_group = 12
type = 1
max = 0
```

```
min = 0
module_interval = 300
id_module_group = 4
id_modulo = 4
plugin_user = root
plugin_pass =
plugin_parameter = "uptime | awk '{ print $10 }' | tr -d ','"
max_timeout = 20
history_data = 1
min_warning = 2
min_critical = 5
str_warning = "peligro"
min_critical = "alerta"
min_ff_event = 0
tcp_port = 0
critical_inverse = 0
warning_inverse = 0
critical_instructions = "Call head of department"
warning_instructions = "Calling the server manager to reduce the load"
unknown_instructions = "Verify that Pandora FMS agent is running"
```

注意すべき点がいくつかあります。

- どのフィールドも " 忘れない " こと。すべてのフィールドが必須となっています。設定値がない場合は、上記例の `plugin_pass` フィールドのように空白にしておくこと
- 上記例の `plugin_parameter` フィールドのように、特殊文字や空白を含む値を定義する場合はダブルクォートで囲むこと。INIファイル中に ' " / - _ () [documentation_ja](#) と書いた文字を含む場合はダブルクォートを使用すること。データ中に " を含まないようにしましょう。もし必要になった場合は / " とエスケープしましょう
- フィールドの意味や目的がよくわからない場合は、Pandora FMS データベースの `tnetwork_component` を参照してください。ほとんど同じフィールドがあります。ネットワークコンポーネントを作成する時は、利用するプラグインとデータベースに保存される全ての値を確認しながら作成してみてください。
- `id_module` の値は常に4(これがプラグインモジュールであることを意味します)です。
- `type` はモジュールの種類を指定します。 `ttipo_modulo` テーブルで `generic_data (1)`, `generic_proc (2)`, `generic_data_string (3)`, `generic_data_inc (4)` が定義されています。
- `id_group` はグループ定義を含む `tgrupo` テーブルのPK(プライマリキー)を設定します。1は " 全グループ " を意味し、特別なグループのようにふるまいます。
- `id_module_group` の値は `tmodule_group` テーブルを参照しており "XXXX"1" を設定することで、一般モジュールグループを指定できます。

バージョン2

Pandora FMS v5.1 SP1 から、サーバプラグインはマクロを利用します。

これらのプラグインは、`.pspz2` という拡張子で区別されま
す。

また `plugin_definition.ini` が変更になっています。次のフィールドが追加されました。

plugin_definition セクション:

- *total_macros_provided* プラグインが持つ動的マクロの数を定義します。

module<N> セクション

- *macro_<N>_value* 動的マクロを使ったモジュールの値を定義します。存在しない場合はデフォルト値が利用されます。

そして、それぞれの動的マクロ用の次のような新たなセクションが作られています。

```
[macro_<N>]
hide = 0
description = descripción
help = texto de ayuda
value = valor
```

この新たな構造をバージョン2 とします。

execution_postcommand の部分に指定したマクロの置換 (*_fieldx_*) を明示的に呼び出す必要があります。以下の例を参照してください。

以前のバージョンとの互換性もあります。バージョンパラメータが定義されてなければ、バージョン1 と想定します。

バージョン2 のプラグイン定義例

```
[plugin_definition]
name = PacketLoss
filename = packet_loss.sh
description = "Measure packet loss in the network in %"
timeout = 20
ip_opt =
execution_command =
execution_postcommand =
parameters = _field1_ _field2_
user_opt =
port_opt =
pass_opt =
plugin_type = 0
total_modules_provided = 1
total_macros_provided = 2

[macro_1]
hide = 0
description = Timeout
```

```
help = Timeout in seconds
value = 5

[macro_2]
hide = 0
description = Target IP
help = IP address
value = 127.0.0.1

[module1]
name = Packet loss
description = "Measure target packet loss in % "
id_group = 15
type = 4
max = 0
min = 0
module_interval = 300
id_module_group = 2
id_modulo = 1
max_timeout = 20
history_data = 1
min_warning = 30
min_critical = 40
min_ff_event = 0
tcp_port = 0
macro_1_value = 5
macro_2_value = localhost
unit = %
```

古い PSPZ のアップグレード

Pandora FMS バージョン 4 までのサーバプラグインの動的パラメータが無い、パラメータが静的な以前の PSPZ は、新しいバージョンの pandora では動作しません。

それらを移行するには、適切な権限でアップグレードプロセスを実行した後、次の手順を実行します。

```
/usr/share/pandora_server/util/pandora_migrate_plugins.pl <dbname> <dbhost>
<dbuser> <dbpass>
```

メジャーおよびマイナーバージョンのアップデートに関する詳細は、[こちら](#)を参照してください。