

「日医標準レセプトソフト」

ORCA Project

日レセプラグイン リファレンス

2013 年 5 月 20 日

公益社団法人日本医師会

1	パッケージ	2
2	パッケージ構成	2
3	link ファイル	3
4	スクリプトファイル(pre*, post*)	3
5	パッケージストア	4
6	パッケージデータベース	4
7	パッケージリストによるパッケージデータベースの更新	5
8	パッケージのバージョンとインストール時の制約	6
9	署名検証	6
10	パッケージ管理コマンド jma-plugin	7
11	パッケージ管理コマンドの共通オプション	7
12	設定ファイル	8
13	install コマンド	8
14	uninstall コマンド	8
15	enable コマンド	9
16	disable コマンド	9
17	update コマンド	9
18	upgrade コマンド	10
19	list コマンド	10
20	search コマンド	11
21	info コマンド	11
21	info コマンド	11

1 パッケージ

- ・パッケージにはバイナリモジュール、設定ファイル、データファイル、スクリプトなどを格納する
- ・パッケージは名前とバージョンにより特定される
 - 名前は英数字、アンダースコアから構成される文字列
 - バージョンは英数字、アンダースコア、ピリオドから構成される文字列
- ・パッケージのファイル名は[名前]-[バージョン]. jpp
- ・パッケージのファイル形式は tgz とする
- ・パッケージには GPG による署名を行う
- ・パッケージには以下の操作が可能である
 - インストール
 - アンインストール
 - リンク
 - インストールしたファイルの symlink を作成する
 - モジュールの有効/無効の制御のため
 - アンリンク
 - リンクで作成した symlink を削除する
 - モジュールの有効/無効の制御のため

2 パッケージ構成

- ・パッケージを展開すると以下のような構成になっている

```
[名前]-[バージョン]/
+-- meta/
| +-- link      ... リンクリスト
| +-- preinst   ... インストール前スクリプト
| +-- postinst  ... インストール後スクリプト
| +-- prelink   ... リンク前スクリプト
| +-- postlink  ... リンク後スクリプト
| +-- preunlink ... アンリンク前スクリプト
| +-- postunlink ... アンリンク後スクリプト
| +-- prerm     ... 削除前スクリプト
| +-- postrm    ... 削除後スクリプト
+-- その他、任意のディレクトリ、ファイル
...
```

- ・ meta 以下にパッケージメタファイルを配置する
- ・ link 以外のスクリプトファイルはなくてもよい

- ・インストールすると全てのファイルがパッケージストアに配置される

3 link ファイル

- ・link ファイルはリンク操作時のインストールファイルとリンク先の対のリストである
- ・link ファイルはインストールファイルとリンク先の配列の配列の YAML 形式である
 - インストールファイルはパッケージディレクトリからの相対パスである
 - リンク先はリンクプレフィックスからの相対パスで指定する
 - 実際のリンク先は リンクプレフィックス/相対パス となる
 - リンク先はファイル、またはディレクトリが指定できる
 - リンクプレフィックスは設定ファイルに記載する
- ・link ファイルの例を以下に示す

```
---  
- [data1/1.txt , plugin/1.txt]  
- [data1/2.txt , plugin/2.txt]  
- [data2/data3/3.txt , plugin/]  
- [data2/data3/4.txt , plugin/]
```

4 スクリプトファイル(pre*, post*)

- ・スクリプトファイルは実行可能であればどのような種類(shell、ruby、バイナリ)でも構わない
- ・スクリプトにはパッケージ名、バージョンが引数として渡される
- ・スクリプト実行時のカレントディレクトリはスクリプトのあるディレクトリ (File.dirname (SCRIPT_PATH)) となる
 - よってパッケージ内のファイルを相対パスで参照可能
- ・preinst はインストール実行前に実行される
 - インストール実行前なので展開用のテンポラリなディレクトリで実行される
- ・postinst はインストール実行後に実行される
- ・prelink はリンク実行前に実行される
- ・postlink はリンク実行後に実行される
- ・preunlink はアンリンク実行前に実行される
- ・postunlink はアンリンク実行後に実行される
- ・prerm はパッケージ削除実行前に実行される
- ・postrm はパッケージ削除実行後に実行される
 - パッケージ削除後なのでテンポラリなディレクトリで実行される

5 パッケージストア

- ・インストールされたパッケージはパッケージストアに指定されたディレクトリに配置される
- ・パッケージストアにはパッケージ以外に、バージョンファイル、パッケージデータベースファイル、キャッシュディレクトリが配置される
 - バージョンファイルにはパッケージシステムのバージョンを記述する
 - パッケージデータベースファイルはパッケージ情報を格納する sqlite3 のファイルである
 - キャッシュディレクトリはダウンロードしたパッケージを置くディレクトリである
- ・パッケージストアの配置例を以下に示す

パッケージストア/	
+-- VERSION	... バージョンファイル
+-- package.db	... データベースファイル
+-- cache/	... キャッシュディレクトリ
+-- foo-1.0.jpp	... パッケージファイル1
+-- bar-1.0.jpp	... パッケージファイル2
...	
+-- foo-1.0/	... パッケージ1
+-- bar-1.0/	... パッケージ2
...	

6 パッケージデータベース

- ・現在利用可能なパッケージをパッケージデータベースに格納する
- ・パッケージデータベースには sqlite3 を利用する
 - データベースの文字コードは UTF-8 である
- ・パッケージデータベースに格納する内容は以下である
 - system テーブル
 - パッケージシステムのバージョンを格納する

```
CREATE TABLE system(version TEXT NOT NULL);
```

- control テーブル
- パッケージ情報を格納する

```
CREATE TABLE control (  
  id          INTEGER PRIMARY KEY, # ID  
  name        TEXT NOT NULL,      # パッケージ名  
  version     TEXT NOT NULL,      # パッケージバージョン  
  description TEXT,               # 説明  
  vendor      TEXT NOT NULL,      # ベンダー名  
  date        TEXT NOT NULL,      # 日付  
  url         TEXT NOT NULL,      # パッケージの取得先  
  install     TEXT NOT NULL,      # インストールフラグ、  
                                     "TRUE" or "FALSE"  
  link        TEXT NOT NULL       # リンクフラグ、  
                                     "TRUE" or "FALSE"  
);
```

7 パッケージリストによるパッケージデータベースの更新

- ・パッケージデータベースは複数のパッケージリストを参照して更新される
- ・パッケージリストは HTTP サーバ、またはファイルシステムより取得する
 - パッケージリストは GPG による署名を行う
- ・パッケージリストはパッケージの各種メタ情報が YAML 形式で記載される
 - 以下にパッケージリストの例を示す

```
---  
- :name: foo # パッケージ名 foo  
  :version: "1.1" # バージョン  
  :vendor: test # ベンダー名  
  :date: "20130102" # 更新日時  
  :url: http://localhost:3000/foo-1.1.jpp # パッケージ URL  
- :name: bar # パッケージ名 bar  
  :version: "1.0" # ... 同上  
  :vendor: test  
  :date: "20130102"  
  :url: http://localhost:3000/bar-1.0.jpp
```

8 パッケージのバージョンとインストール時の制約

- ・ある名前のパッケージについて複数のバージョンが存在したときにインストールできるのは常に一つのバージョンのみである
 - 複数のバージョンを同時にインストールできない
 - ・データベースのパッケージ情報ではインストールしていない古いバージョンは自動的に削除される
 - ある名前のパッケージについてデータベースには以下の二つのパターンしか格納されない
 - (a) 最新パッケージのみ
 - (b) インストールされている古いパッケージと最新パッケージ
- * アップグレード可能な状態

※ 依存関係は現段階では考慮しない

- ・依存関係が必要なパッケージは少ないのではないかという予想
- ・仮に依存関係のような問題が起きたとしても、最新パッケージをすぐ出すことで対応する
 - ユーザは常にパッケージを更新するだけでよい

9 署名検証

- ・パッケージとパッケージリストは GPG により署名される
- ・署名検証はパッケージ、パッケージリスト取得時に署名ファイルを取得して行う
 - 署名検証は以下のコマンドで行う

```
$ gpg --verify <署名ファイル(XXX.asc)>
```

- 署名検証を行うために事前に公開鍵を設定しておく必要がある
 - 公開鍵の設定は以下のコマンドで行う

```
$ gpg --import publickey.asc
```

- ・パッケージ、パッケージリストの署名は以下のコマンドで行う

```
$ gpg -abs <署名対象ファイル(パッケージファイル、パッケージリスト)>
```

- 署名対象ファイル名 + “.asc” というファイル名で署名ファイルが作成される
- 署名ファイルは署名対象のファイルと同じディレクトリに配置する

10 パッケージ管理コマンド jma-plugin

- ・ jma-plugin コマンドによりパッケージの管理操作を行う
- ・ jma-plugin の基本コマンドは以下である
 - install ... パッケージをインストールする
 - uninstall ... パッケージをアンインストールする
 - enable ... リンクを作成しパッケージを有効化する
 - disable ... リンクを削除しパッケージを無効化する
 - update ... パッケージリストを取得しパッケージデータベースを更新する
 - upgrade ... アップグレード可能なパッケージを更新する
 - list ... パッケージデータベースのパッケージ情報をリスト表示する
 - search ... パッケージデータベースからパッケージ名が指定された文字列に部分一致するパッケージ情報を表示する
 - info ... 指定されたパッケージのデータベース情報、インストールファイル一覧、リンクファイル一覧を表示する
- ・ install, uninstall, enable, disable, update, upgrade コマンド実行時は排他ロックがかかる

11 パッケージ管理コマンドの共通オプション

- ・ 管理コマンドの共通オプションを以下に示す
 - -c, --config=VAL ... 設定ファイルを指定する
 - -r, --root=VAL ... パッケージストアパスを指定する、設定ファイルより優先
 - -l, --list=VAL ... パッケージリストの URL、またはファイルパスを指定する、設定ファイルより優先
 - -p, --linkprefix=VAL ... リンクプレフィックスを指定する、設定ファイルより優先
 - -L, --log=VAL ... ログファイルを指定する、デフォルトは STDERR
 - -n, --no-verify ... 署名検証を行わない、設定ファイルより優先
 - -d, --debug ... ログレベルを DEBUG にする
- ・ パッケージストアパス、パッケージリストは必須である

1.2 設定ファイル

- ・設定ファイルはYAML形式でパッケージストアパス、パッケージリストのURL、ファイルパスのリスト、署名検証フラグ、リンクプレフィックスを設定する
 - 以下に設定ファイルの例を示す

```
---
:root: /tmp/test-jma-plugin          # パッケージストアパス
:list:                               # パッケージリストのリスト
- http://localhost:3000/list1.yml    # パッケージリスト1
- http://localhost:3000/list2.yml    # パッケージリスト2
- http://localhost:3000/list3.yml    # パッケージリスト3
:linkprefix: /tmp/link              # リンクプレフィックス
:verify: true                       # 署名検証フラグ
```

1.3 install コマンド

- ・コマンドの引数は以下である

```
% jma-plugin [オプション] install <パッケージ名>-<バージョン> [<パッケージ名>-<バージョン>]...
```

- ・引数のパッケージを左から一つずつインストールする
 - 自動的に同名の古いパッケージはアンインストールされる
 - 古いパッケージがリンクされていれば、アップグレードしたパッケージもリンクされる
- ・preinst スクリプトがあれば実行する
 - preinst スクリプトの実行にエラーがあればインストールを中断する
- ・postinst スクリプトがあれば実行する
 - postinst スクリプトの実行にエラーがあればインストールしたファイルを削除しインストールしなかったことにする

1.4 uninstall コマンド

- ・コマンドの引数は以下である

```
% jma-plugin [オプション] uninstall <パッケージ名>-<バージョン> [<パッケージ名>-<バージョン>]...
```

- ・引数のパッケージを左から一つずつアンインストールする
 - パッケージがリンクされていた場合、自動でアンリンクする
- ・preuninst スクリプトがあれば実行する
 - preuninst スクリプトの実行にエラーがあればアンインストールを中断する

- ・ postrm スクリプトがあれば実行する
 - postrm スクリプトの実行にエラーがあったとしてもアンインストールされたことにする

1 5 enable コマンド

- ・ コマンドの引数は以下である

```
% jma-plugin [オプション] enable <パッケージ名>-<バージョン> [<パッケージ名>-<バージョン>]...
```

- ・ 引数のパッケージを左から一つずつリンクを作成し有効化する
 - リンク先のディレクトリは作成しない
 - リンク作業中にエラーがあった場合、リンク作業を中止し、それまでに作成されたすべてのリンクを削除する
- ・ prelink スクリプトがあれば実行する
 - prelink スクリプトの実行にエラーがあればリンク作成を中断する
- ・ postlink スクリプトがあれば実行する
 - postlink スクリプトの実行にエラーがあった場合、リンクを全て削除する

1 6 disable コマンド

- ・ コマンドの引数は以下である

```
% jma-plugin [オプション] disable <パッケージ名>-<バージョン> [<パッケージ名>-<バージョン>]...
```

- ・ 引数のパッケージを左から一つずつリンクを削除し無効化する
 - アンリンク作業中にエラーがあったとしてもアンリンクを続行する
- ・ preunlink スクリプトがあれば実行する
 - prelink スクリプトの実行にエラーがあればリンクの削除を中断する
- ・ postunlink スクリプトがあれば実行する
 - postlink スクリプトの実行にエラーがあった場合でも削除したリンクの復帰はしない

1 7 update コマンド

- ・ コマンドの引数は以下である

```
% jma-plugin [オプション] update
```

- ・ オプションで指定されたパッケージリストを取得しパッケージデータベースを更新する
 - 同名のパッケージでバージョンが低く、インストールされていないパッケージ情報はデータベースより自動削除される

- ・パッケージリスト取得時に GPG の署名検証を行う
 - 事前にコマンドを実行するユーザのストアに GPG 公開鍵 を設定しておく必要がある
 - オプションにより署名検証をスキップできる

18 upgrade コマンド

- ・コマンドの引数は以下である

```
% jma-plugin [オプション] upgrade
```

- ・データベースよりアップグレード可能なパッケージを検索し、アップグレード可能なパッケージをインストールする
 - 自動的に古いパッケージはアンインストールされる
 - 古いパッケージがリンクされていれば、アップグレードしたパッケージもリンクされる

19 list コマンド

- ・コマンドの引数は以下である

```
% jma-plugin [オプション] list [install|enable|upgrade]
```

- ・データベースよりパッケージ情報を取り出しリスト表示する
 - install を指定するとインストールされているパッケージのみ表示する
 - enable を指定すると有効化されているパッケージのみ表示する
 - upgrade を指定するとアップグレード可能なパッケージのみ表示する
 - その他の場合、データベース内のすべてのパッケージを表示する
- ・出力はYAML形式である
- ・出力例を以下に示す

```
$ ./bin/jma-plugin -r test/store/ -l test/data/list1.yml list
---
- :description:
  :link: "FALSE"
  :vendor: test
  :version: "1.0"
  :name: bar
  :install: "TRUE"
  :date: "20090102"
  :url: http://localhost:3000/bar-1.0.jpg
  :id: 3
```

```
- :description:  
  :link: "TRUE"  
  :vendor: test  
  :version: "1.2"  
  :name: foo  
  :install: "TRUE"  
  :date: "20090102"  
  :url: http://localhost:3000/foo-1.2.jp  
  :id: 4
```

2 0 search コマンド

- ・ コマンドの引数は以下である

```
% jma-plugin [オプション] search <部分パッケージ名>
```

- ・ データベースより指定された部分パッケージ名に部分一致するパッケージの情報をリスト表示する
- ・ 出力形式は list コマンドと同様の YAML 形式である

2 1 info コマンド

- ・ コマンドの引数は以下である

```
% jma-plugin [オプション] info <パッケージ名>-<バージョン>
```

- ・ 指定されたパッケージのパッケージ情報、インストールファイルリスト、リンクファイルリストを表示する
 - インストールファイルリストはインストール済みの場合のみ表示される
 - リンクファイルリストはリンク済みの場合のみ表示される
- ・ 出力は YAML 形式である

- ・ 出力例を以下に示す

```
$ ./bin/jma-plugin -r test/store/ -l test/data/list1.yml info foo-1.2
---
:file:
- /home/oruser/orca-plugin/test/store/foo-1.2/data1/1.txt
- /home/oruser/orca-plugin/test/store/foo-1.2/data1/2.txt
- /home/oruser/orca-plugin/test/store/foo-1.2/data2/data3/3.txt
- /home/oruser/orca-plugin/test/store/foo-1.2/data2/data3/4.txt
- /home/oruser/orca-plugin/test/store/foo-1.2/meta/link
- /home/oruser/orca-plugin/test/store/foo-1.2/meta/postinst
- /home/oruser/orca-plugin/test/store/foo-1.2/meta/postrm
- /home/oruser/orca-plugin/test/store/foo-1.2/meta/preinst
- /home/oruser/orca-plugin/test/store/foo-1.2/meta/prerm
:control:
  :vendor: test
  :date: "20090102"
  :link: "TRUE"
  :version: "1.2"
  :name: foo
  :install: "TRUE"
  :url: http://localhost:3000/foo-1.2.jpp
  :description:
  :id: 4
:link:
- /tmp/jma_plugin/foo-1.2.txt
```