
CrossPM Documentation

Release 0.8.1

Alexander Kovalev

February 10, 2017

1	Introduction	3
2	Documentation	5
3	Installation	7
4	Usage	9
5	Examples	11
6	Indices and tables	13

Contents:

Introduction

CrossPM (Cross Package Manager) is a universal extensible package manager. It lets you download and as a next step - manage packages of different types from different repositories.

Out-of-the-box modules:

- Adapters
 - Artifactory
- Package file formats
 - zip
 - tar.gz
 - nupkg (treats like simple zip archive for now)

Modules planned to implement:

- Adapters
 - filesystem
 - git
 - smb
 - sftp/ftp
- Package file formats
 - nupkg (nupkg dependencies support)
 - 7z

We also need your feedback to let us know which repositories and package formats do you need, so we could plan its implementation.

The biggest feature of CrossPM is flexibility. It is fully customizable, i.e. repository structure, package formats, packages version templates, etc.

To handle all the power it have, you need to write configuration file (**crosspm.yaml**) and manifest file with the list of packages you need to download.

Configuration file format is YAML, as you could see from its filename, so you free to use yaml hints and tricks, as long, as main configuration parameters remains on their levels :)

Documentation

We just started to write documentation for this project, but we'll try to cover all usage topics ASAP.

Here is direct link to it: <https://crosspm.readthedocs.io>

Installation

To install CrossPM, simply:

```
pip install crosspm
```

Usage

To see commandline parameters help, run:

```
crosspm --help
```

You'll see something like this:

```
CrossPM (Cross Package Manager) version: *** The MIT License (MIT)

Usage:
  crosspm download [options]
  crosspm promote [options]
  crosspm pack <OUT> <SOURCE> [options]
  crosspm -h | --help
  crosspm --version

Options:
  <OUT>                Output file.
  <SOURCE>             Source directory path.
  -h, --help          Show this screen.
  --version           Show version.
  -l, --list          Do not load packages and its dependencies. Just show what's found.
  -v, --verbose       Increase output verbosity.
  --verbosity=LEVEL   Set output verbosity level: (critical, error, warning, info, debug)
  -c=FILE, --config=FILE Path to configuration file.
  -o OPTIONS, --options OPTIONS Extra options.
  --depslock-path=FILE Path to file with locked dependencies [./dependencies.txt.lock]
  --out-format=TYPE   Output data format. Available formats:(['shell', 'python', 'json', 'yaml'])
  --output=FILE       Output file name (required if --out_format is not stdout)
  --out-prefix=PREFIX Prefix for output variable name [default: ] (no prefix at all)
  --no-fails         Ignore fails config if possible.
```

Examples

We'll add some more examples soon. Here is one of configuration file examples for now:

```
cpm:
  dependencies: dependencies.txt
  dependencies-lock: dependencies.txt.lock
  cache:
    cmdline: cache
    env: CROSSPM_CACHE_ROOT
    default:

columns: "*package, version, branch"

values:
  quality:
    1: banned
    2: snapshot
    3: integration
    4: stable
    5: release

options:
  compiler:
    cmdline: cl
    env: CROSSPM_COMPILER
    default: vc110

  arch:
    cmdline: arch
    env: CROSSPM_ARCH
    default: x86

  osname:
    cmdline: os
    env: CROSSPM_OS
    default: win

parsers:
  common:
    columns:
      version: "{int}.{int}.{int}[.{int}][-{str}]"
    sort:
      - version
      - '*'
```

```
index: -1

artifactory:
  path: "{server}/{repo}/{package}/{branch}/{version}/{compiler|any}/{arch|any}/{osname}/{package}"
  properties: "some.org.quality = {quality}"

defaults:
  branch: master
  quality: stable

fails:
  unique:
    - package
    - version

common:
  server: https://repo.some.org/artifactory
  parser: artifactory
  type: jfrog-artifactory
  auth_type: simple
  auth:
    - username
    - password

sources:
  - repo:
      - libs-release.snapshot
      - libs-release/extlibs

  - type: jfrog-artifactory
    parser: artifactory
    server: https://repo.some.org/artifactory
    repo: project.snapshot/temp-packages
    auth_type: simple
    auth:
      - username2
      - password2

output:
  tree:
    - package: 25
    - version: 0
```

Indices and tables

- `genindex`
- `modindex`
- `search`