
rpg Documentation

Release 0.0.1

See AUTHORS in RPG source distribution.

March 16, 2016

1	RPG Command Reference	3
2	Writing plugins	5
3	Base class	7
4	Command module	9
5	Plugin class	11
6	Spec module	13
7	Indices and tables	17
	Python Module Index	19

Warning: RPG is in alpha stage and the docs could change at any time without any notice

Contents:

RPG Command Reference

1.1 Synopsis

```
rpg [options]
```

1.2 Description

RPG is tool, that guides people through the creation of a RPM package. *RPG* makes packaging much easier due to the automatic analysis of packaged files. Beginners can get familiar with packaging process or the advanced users can use our tool for a quick creation of a package.

1.3 Options

-h, --help Show help message and exit.

--plugin-dir <dir> [<dir> ...] Include plugin directory.

--disable-plugin <plugin-name> [<plugin-name> ...] Exclude specific plugin.

--disable-dnf Disable loading DNF sack.

Writing plugins

Plugins are the main part of RPG. Their purpose is to append `Spec` class with tags and scripts. Whole process of creating spec file from source code in RPG app is following:

- plugins are triggered in project directory - they set some tags/scripts in `Spec` instance
- guessed values from `Spec` are auto filled in GUI forms and user can modify them
- next spec command (`%prep`, `%build`, `%install`, `%files`) is executed and new directory is created
- repeat steps for new directory

Plugin is a class that is derived from `rpg.plugin.Plugin` and overrides at least one of following methods: `download`, `extraction`, `extracted`, `patched`, `compiled`, `installed`, `package_built`. Each of them takes (`spec`, `current_dir`, `sack`) parameters. `spec` is instance of `Spec` class, `sack` is initialized from DNF and `current_dir` is `pathlib.Path` instance where are project files of future RPM package located. `current_dir` is different in each phase.

Table 2.1: Phases

Phase	Description	Base call
down-load	downloads archive from url. This is because url adress may be github repository - then plugin only add archive/master.zip to the url and downloads it	–
extrac-tion	method that extract files from archive. This exists because there are many types of archives like tar, zip, ...	–
ex-tracted	raw files are extracted from chosen archive or copied files from project working directory	extracted source analysis
patched	after application of patches on source files	patched source analysis
com-piled	after execution of <code>%build</code> script (e.g. calling <code>make</code>)	compiled source analysis
in-stalled	directory containing files after <code>make install</code>	installed source analysis
pack-age_built	path to final rpm package	–
mock_recover	after building project in mock enviroment, build errors (if there are any) are passed as parameter to this method. This should fix build by parsing the errors and finding the solution, i.e. append missing required files to <code>build_required_files</code>	–

Inside plugin can be helper methods that should not be named as any of the phase. It should follow conventions as any private Python method (e.g. `_helper_method`).

For plugin examples take a look at [core plugins folder](#).

Base class

Command module

Plugin class

```
class plugin.Plugin
    Class from which are plugins derived
    def extraction(self, source, dest): pass
    def download(self, source, dest): pass
    def extracted(self, project_dir, spec, sack): pass
    def patched(self, project_dir, spec, sack): pass
    def compiled(self, project_dir, spec, sack): pass
    def installed(self, project_dir, spec, sack): pass
    def package_build(self, package_path, spec, sack): pass
    def mock_recover(self, log, spec): pass
```

Spec module

class spec.Spec
 SPEC properties holder

Example

```
>>> from rpg.spec import Spec
>>> spec = Spec()
>>> spec.Name = "Example"
>>> spec.Version = "0.6.11"
>>> spec.Release = "1%{?snapshot}%{?dist}"
>>> spec.License = "GPLv2"
>>> spec.Summary = "Example ..."
>>> spec.description = ("Example ...")
>>> spec.URL = "https://github.com/example_repo"
```

BuildArch = None
 initial value: ""

BuildRequires = None
 initial value: set()

BuildRoot = None
 initial value: ""

class Changelog (*date, author, email, *message*)

Spec.Conflicts = None
 initial value: ""

Spec.Group = None
 initial value: ""

Spec.License = None
 initial value: ""

Spec.Name = None
 initial value: ""

Spec.Obsoletes = None
 initial value: ""

Spec.Packager = None
 initial value: ""

Spec.Patch = None
 initial value: ""

Spec.Provides = None
initial value: set()

Spec.Release = None
initial value: ""

Spec.Requires = None
initial value: set()

Spec.Source = None
initial value: ""

Spec.Summary = None
initial value: ""

Spec.URL = None
initial value: ""

Spec.Vendor = None
initial value: ""

Spec.Version = None
initial value: ""

Spec.build = None
initial value: Command()

Spec.changelog = None
initial value: []

Spec.changelogs = []

Spec.check = None
initial value: Command()

Spec.clean = None
initial value: Command()

Spec.description = None
initial value: ""

Spec.files = None
initial value: []

Spec.install = None
initial value: Command()

Spec.package = None
initial value: ""

Spec.post = None
initial value: Command()

Spec.posttrans = None
initial value: Command()

Spec.postun = None
initial value: Command()

Spec.pre = None
initial value: Command()

Spec.prep = None
initial value: Command()

Spec.**pretrans** = None
initial value: Command()

Spec.**preun** = None
initial value: Command()

Indices and tables

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

p

plugin, [11](#)

s

spec, [13](#)

B

build (spec.Spec attribute), 14
BuildArch (spec.Spec attribute), 13
BuildRequires (spec.Spec attribute), 13
BuildRoot (spec.Spec attribute), 13

C

changelog (spec.Spec attribute), 14
changelogs (spec.Spec attribute), 14
check (spec.Spec attribute), 14
clean (spec.Spec attribute), 14
Conflicts (spec.Spec attribute), 13

D

description (spec.Spec attribute), 14

F

files (spec.Spec attribute), 14

G

Group (spec.Spec attribute), 13

I

install (spec.Spec attribute), 14

L

License (spec.Spec attribute), 13

N

Name (spec.Spec attribute), 13

O

Obsoletes (spec.Spec attribute), 13

P

package (spec.Spec attribute), 14
Packager (spec.Spec attribute), 13
Patch (spec.Spec attribute), 13
Plugin (class in plugin), 11

plugin (module), 11
post (spec.Spec attribute), 14
posttrans (spec.Spec attribute), 14
postun (spec.Spec attribute), 14
pre (spec.Spec attribute), 14
prep (spec.Spec attribute), 14
pretrans (spec.Spec attribute), 14
preun (spec.Spec attribute), 15
Provides (spec.Spec attribute), 13

R

Release (spec.Spec attribute), 14
Requires (spec.Spec attribute), 14

S

Source (spec.Spec attribute), 14
Spec (class in spec), 13
spec (module), 13
Spec.Changelog (class in spec), 13
Summary (spec.Spec attribute), 14

U

URL (spec.Spec attribute), 14

V

Vendor (spec.Spec attribute), 14
Version (spec.Spec attribute), 14