

RPM Packaging

How software delivery works and why RPM packaging is more current than ever

Fabio Alessandro Locati

26 October 2016

Intro

RPM Processes

Intro

About me

- RPM user since 2001
- IT Consultant since 2004
- RPM creator since 2013

RPM components

- SPEC file
- Sources files (at least 1)
- Patches (eventually)

RPM Processes

RPM build process

- Fetch of the SPEC file
- Fetch of sources/patches
- Creation of the .src.rpm file
- Creation of binaries .rpm files

The SPEC file structure

- Preamble
- Body

The SPEC file structure - the preamble

- **Name**
- **Version**
- **Release**
- Summary
- License
- URL
- **SourceX**
- PatchX
- BuildArch
- BuildRequires
- Requires

Body sections

- **%description:** Long description
- **%prep:** Commands to prepare the environment for the build
- **%build:** Commands to build the software
- **%install:** Commands to install the software
- **%check:** Commands to test the software
- **%files:** List of files and folders to be included in the package
- **%changelog:** Change history

Helpers

- GO: `gofed`
- Perl: `cpan2rpm`
- Python: `pyp2rpm`
- Ruby: `gem2rpm`

Additional resources

- Slides: <https://static.fale.io/slides/20161026-en-rpm.pdf>
- Demo code: <https://github.com/Fale/demo-specs>
- Official website: <http://rpm.org>
- Fedora guide: https://fedoraproject.org/wiki/How_to_create_an_RPM_package
- RPM Guide: <http://rpm-guide.readthedocs.io>