

*Software Packaging,
RPM Packaging,
Packaging Guidelines,
Build Systems,*

...

but why?

Software Packaging

Why does:

```
$ ./configure && make && \  
sudo make install
```

work (or not)?

#1: “Oops!”

#2: *“Lather, rinse, repeat”*

(and hope for the best)

RPM Packaging

*Package once,
deploy many times,
develop deployment through packaging,
update production systems.*

Binary compatibility,

reproducibility,

stripping debugging symbols,

update / upgrade path,

management & auditability,

...

Packaging Guidelines

Define standards, best practices.

Make packaging consistent.

Make packaging easier.

Question:

Where is the Ruby gem directory?

Answer:

```
%global gemdir %(ruby -rubygems -e 'puts Gem::dir' 2>/dev/null)
```

Usage:

```
%install  
mkdir -p ./%{gemdir}  
gem install --local --install-dir ./%{gemdir} \  
  --force --rdoc %{SOURCE0}
```

Build Systems

(i.e. Koji)

What does it do (for you)?

- *clean build-roots*
- *build for multiple architectures in one go*
- *guarantee unique NEVR(A)*
- *buildSRPMFromSCM -> collaboration*
- *'build repository' management*
- *'result / product' repository management*
- ...

Tools That Help You

- *fedpkg*
- *mock*
- *koji*