

Let's package a Go project into Fedora

Jan Chaloupka <jchaloup@redhat.com>
Jakub Cajka <jcajka@redhat.com>

Gofed





2 Theory

8 Break

10 Practice

Spec file

Guidelines

Analysis

Builds

Review request

29 Break

31 Advanced

Packaging

Updates

37 Contributing



- Yes, Golang is in Fedora ;)
- GC(aka Go compiler, golang) since version 1.1 in Fedora 17
- gcc-go "ever since"
- Fedora defaulting to GC
- you can have both on same system thanks to Fedora-alternatives
- "dnf install golang"



- F24 go1.6.4
- F25 go1.7.4
- F26(rawhide) go1.8rc2
- x86_64, armv7, aarch64, ppc64le, s390x, ppc64
- one version per release
- security fixes, no-more features removed



- GOROOT is /usr/lib/golang
- GOPATH is /usr/share/gocode
- all std tools are there(fix, fmt, doc,...)
- there are also other tools available in repos
- vim-go, gofed, md2man, glide, godep



- golang subpackages
 - golang, -bin, -src
 - -test, -misc
 - -doc



- shared linking
- lib -build-mode=shared
- bin -linkshared
- dynamically link stdlib
- dnf install golang-shared
- looking forward for F27



2 Theory

8 Break

10 Practice

Spec file

Guidelines

Analysis

Builds

Review request

29 Break

31 Advanced

Packaging

Updates

37 Contributing



- Check your Internet connection

SSID: vutbrno USER:devconfcz PWD:pinkhatbrno

- Install dev environment

```
# dnf install fedora-packager  
# dnf install golang go-srpm-macros
```

- Install gofed rpms

```
# dnf copr enable jchaloup/gofed  
# dnf install gofed  
# usermod -a -G gofed USERNAME  
$ gofed --help
```

- Think of GitHub API limits, i.e. 80 requests per hour
- <https://admin.fedoraproject.org/accounts> (Fedora account creation, sign CLA under "My account")
- BZ account <https://bugzilla.redhat.com/>, use same email as in FAS (optional, but needed to become package maintainer)
- <http://piratepad.net/goinddevconf2017>
- If you take your own project it is a high chance you will end with a review request (do you have a sponsor?)



2 Theory

8 Break

10 Practice

Spec file

Guidelines

Analysis

Builds

Review request

29 Break

31 Advanced

Packaging

Updates

37 Contributing



Command (Spec file generator)

```
gofed repo2spec --detect PROJECTURL [--commit COMMIT]  
[--format] [--force] [--with-build] [--with-extra]
```

- PROJECTURL: url of a project, e.g. `github.com/kr/pretty`
- COMMIT: specific commit/revision in a VCS (if not set, the latest commit gets detected)
- `--format`: pretty printing
- `--force`: skip distro pkg check (e.g. regenerate a spec file)
- `--with-build`: generate build sections as well

Example

```
gofed repo2spec --detect github.com/kr/pretty --format  
--commit cfb55aafdaf3ec08f0db22699ab822c50091b1c4
```



Command (Spec file generator)

```
gofed repo2spec --detect PROJECTURL [--commit COMMIT]  
[--format] [--force] [--with-build] [--with-extra]
```

- Response:
 - Identification of a project
 - Checks
 - Data collection
 - Spec file composition
 - Dependency analysis
- More at <https://github.com/gofed/gofed#spec-file-generator>



2 Theory

8 Break

10 Practice

Spec file

Guidelines

Analysis

Builds

Review request

29 Break

31 Advanced

Packaging

Updates

37 Contributing



- Fedora packaging guidelines: <https://fedoraproject.org/wiki/Packaging:Guidelines>
- Golang packaging guidelines draft: <http://fedoraproject.org/wiki/PackagingDrafts/Go>
- go-srpm-macros rpm (Fedora only)
- unit-test subpackage (CI support), architectures coverage (as much as possible)
- devel subpackage (minimal subset of source codes)
- more docs at
<https://github.com/gofed/gofed/tree/master/docs>



- bundled vs. unbundled vs. combination
<https://fedorahosted.org/fpc/ticket/575>
- backward compatibility
- be a good citizen (dependency sharing and incompatibilities)
- migration of projects (e.g. code.google.com/p to github.com), naming convention then?



- common project locations:
 - [google].golang.org: mostly stored on github
 - github.com: majority of projects is here
 - code.google.com: read-only, mostly migrated to github
 - bitbucket.org: very few
 - gopkg.in: mostly stored on github
 - new prefixes showing up
- how to find project's real provider? mostly github.com, check docs and try to go through links



2 Theory

8 Break

10 Practice

Spec file

Guidelines

Analysis

Builds

Review request

29 Break

31 Advanced

Packaging

Updates

37 Contributing



- what dependencies my project requires?
- what tests my projects can run?
- what packages my project provides?
- what binaries can I build from my project ?



Command (Fetch source code)

gofed fetch --spec

- run after a spec file is generated
- reads macros defined in the spec to construct tarball URL

```
%global provider      github
%global provider_tld  com
%global project       spf13
%global repo          viper
# https://github.com/spf13/viper
%global provider_prefix %{provider}. %{provider_tld} / %{project} / %{repo}
%global import_path   %{provider_prefix}
%global commit        16990631d4aa7e38f73dbbf37fa13e67c648531
```



Command (Code analysis I)

gofed inspect -p | -t | -m [--spec]

- **-p:** list provided packages
- **-t:** list tests
- **-m:** list main packages
- **--spec:** list items in spec file format
- **--all-occurrences:** include dependencies of main packages as well

Example

```
$ cd pretty-cfb55aafdaf3ec08f0db22699ab822c50091b1c4  
$ gofed inspect -p
```



Command (Code analysis II)

```
gofed ggi [--classes [-pkgdb]] [--spec] [ --all-occurrences ]
```

- `--classes`: decompose all dependencies into classees
- `--pkgdb`: check if a dependency is already in Fedora
- `--spec`: list all items in spec file format
- `--all-occurrences`: include dependencies of main packages as well

Example

```
$ cd pretty-cfb55aafdaf3ec08f0db22699ab822c50091b1c4  
$ gofed ggi --classes
```



- various ways how to capture project dependencies
- Glide.lock, Godeps.json, GLOCK.yaml, vendor.json, etc.
- both imported package and its commit/revision are important
- working with tools like glide, godeps, etc. is out of scope
- subset of glide.lock:
 - name: github.com/cloudfoundry-incubator/candiedyaml
version: 99c3df83b51532e3615f851d8c2dbb638f5313bf
 - name: github.com/coreos/etcd
version: 8b320e7c550067b1dfb37bd1682e8067023e0751
subpackages:
 - client
 - pkg/pathutil
 - pkg/tlsutil
 - pkg/transport



Command (Snapshot analysis)

```
gofed check-deps --glidefile | --glogfile | --godeps | --vendorfile
```

- For supported formats
- Pulls affected repositories locally to get dates of commits/revisions
- Checks if a dependency is in a distribution (outdated, uptodate or newer)
- Need to scan distribution first,
<https://github.com/gofed/data>

Example

```
$ cd flannel-dbb8b859827b2280ac11e10022eec1fd2c9a58ab  
$ gofed check-deps --glidefile=glide.lock
```



2 Theory

8 Break

10 Practice

Spec file

Guidelines

Analysis

Builds

Review request

29 Break

31 Advanced

Packaging

Updates

37 Contributing



- successful local rpmbuild is a good sign:

```
# dnf builddep *.spec  
$ rpmbuild -ba *.spec
```

- successful scratch build(if you are already Fedora packager) even better:

```
$ koji build --scratch rawhide SRPM.rpm.srpm
```

- alternatively you can try Copr builds (FAS account needed)



- log in using FAS username to <http://copr.fedoraproject.org/>
- “New Project”
- fill in the name
- check the fedora-25-x86_64 chroot
- “Create”
- “Your project” → “Builds” → “New build” → “Upload SRPM”



2 Theory

8 Break

10 Practice

Spec file

Guidelines

Analysis

Builds

Review request

29 Break

31 Advanced

Packaging

Updates

37 Contributing



Command (Review request)

```
gofed review-request [–skip-rpmlint-errors] [–skip-koji]
```

- `--skip-rpmlint-errors`: sometimes rpmlint report false errors
- `--skip-koji`: if you want to create a review without a scratch build
- runs rpmbuild and rpmlint
- runs scratch build
- fill in review request template
- just copy and use:

[https://bugzilla.redhat.com/enter_bug.cgi?
product=Fedora&format=fedora-review](https://bugzilla.redhat.com/enter_bug.cgi?product=Fedora&format=fedora-review)



2 Theory

8 Break

10 Practice

Spec file

Guidelines

Analysis

Builds

Review request

29 Break

31 Advanced

Packaging

Updates

37 Contributing



- Coffee, tea, or whatever will make you feel better





2 Theory

8 Break

10 Practice

Spec file

Guidelines

Analysis

Builds

Review request

29 Break

31 Advanced

Packaging

Updates

37 Contributing



- <https://github.com/akrennmair/gopcap>
- from pcap.go:

```
package pcap
/*
#cgo linux LDFLAGS: -lpcap
#cgo freebsd LDFLAGS: -lpcap
...
#include <pcap.h>
```

- which rpm provides the **pcap.h**?

- \$ whereis pcap.h
pcap: /usr/include/pcap.h /usr/include/pcap ...
\$ rpm -qf /usr/include/pcap.h
libpcap-devel-1.7.4-2.fc24.x86_64
- Or just google it?



- To be or not to be? That is the question.
<https://fedorahosted.org/fpc/ticket/575>
- De-bundling is a good practice
 - security
 - incompatibility detections
 - but it takes more packaging time
- Building with bundled dependency needs proper justification:
 - backward incompatibility
 - very rapidly changing dependency (e.g. docker, kubernetes)
- Only for devel subpackages so far
 - use **Provides: bundles(golang(...))**
- De-bundle as much as you can, build the rest from bundled dependencies



2 Theory

8 Break

10 Practice

Spec file

Guidelines

Analysis

Builds

Review request

29 Break

31 Advanced

Packaging

Updates

37 Contributing



- Not every update to newer version is backward compatible
- Updates are usually commit driven, not version driven (lack of versioning)
- Not many projects adopting vendoring (not easy to tell which commits of dependencies are the right ones)
 - Vendoring reconstruction by taking the closest commits to a given date
- Usually recursive process (updating project itself and its dependencies)
- For more info read https://github.com/gofed/gofed/blob/master/docs/UPDATE_ANALYSIS.md



Command (Bump project)

```
gofed bump [--commit COMMIT] [ --skip-checks ]
```

- --commit: update spec to a particular commit (only tarball and commit)
- --skip-checks: don't check for a list of tags and versions
- list of build/run-time dependencies needs to be updated separately
- list of provided packages and tests needs to be updated separately

Example

```
$ ls  
golang-github-kr-pretty.spec pretty-cfb55aa.tar.gz ...  
$ gofed bump --commit  
7497514e37543f8a9dfdb86a97c2e8d175a0f8a0 --skip-checks
```



2 Theory

8 Break

10 Practice

Spec file

Guidelines

Analysis

Builds

Review request

29 Break

31 Advanced

Packaging

Updates

37 Contributing



- Project page: <https://github.com/gofed/gofed>
- Lot of RFE to implemented <https://github.com/gofed/gofed/issues?utf8=%E2%9C%93&q=RFE%20>
- Distro scans, currently each two hours:
<https://github.com/gofed/data>



- Better spec file manipulation:
<https://github.com/gofed/specker>
- More golang ecosystem scans and analysis
- More automation

Thank you for your attention!