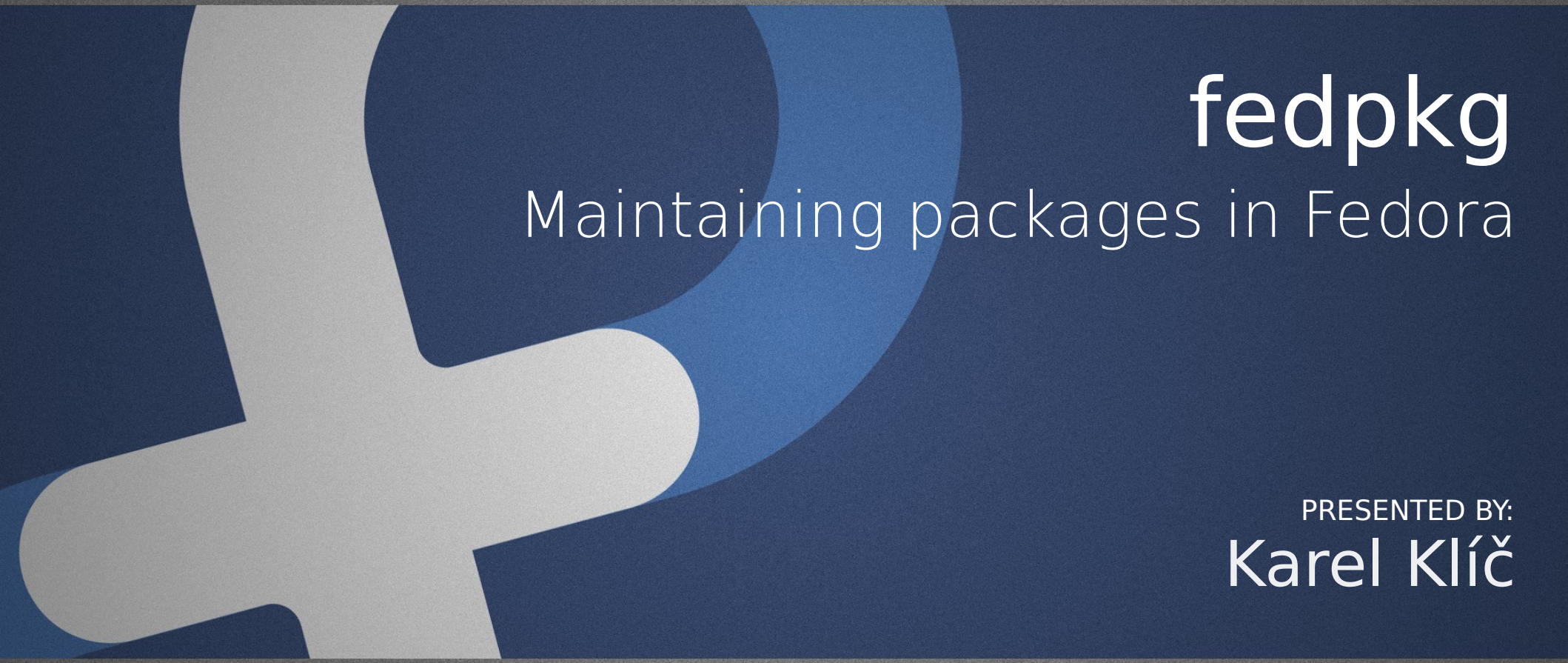


fedpkg

Maintaining packages in Fedora

PRESENTED BY:
Karel Klíč

Creative Commons Attribution 3.0

Today's Topics

- The fedpkg tool
- Exploring packages
- Updating a package
- Backporting changes to older Fedora releases
- Teamwork on a single package


The fedpkg tool

What is it?

- Front-end to the Fedora infrastructure for package maintainers

fedpkg						
Git	lookaside cache	Koji	Bodhi	rpmbuild	mock	rpmlint

- Makes packaging easier, nothing more
- Command line interface, ~2500 lines of Python code (small, easy to read, easy to hack!)

```
$ git clone git://git.fedorahosted.org/git/fedora-packager
```

Getting a package

When you need to change something:

```
$ fedpkg clone emacs
```

...it's equivalent to...

```
$ git clone ssh://kklic@pkgs.fedoraproject.org/emacs
```

If you want a read-only access:

```
$ fedpkg clone emacs --anonymous
```

...or...

```
$ git clone git://pkgs.fedoraproject.org/emacs
```

Authentication

- How does fedpkg know it's you?

  It reads `~/.fedora.cert`, generated by the Fedora Account System for your account

- Used for access to git, lookaside cache, koji, bodhi

`https://admin.fedoraproject.org/accounts/`

```
$ fedora-packager-setup
```

(do not run it on multiple computers for a single account)

It's just a git repository!

```
$ git branch -r
$ gitk
```

Sending a patch to other package maintainer:

```
$ git format-patch origin/master
```

Web interface to all repositories (needs work):

http://pkgs.fedoraproject.org/gitweb/

Updating packages

Preparing an update (1)

```
$ fedpkg pull
$ fedpkg prep
```

- Downloads emacs-23.2.tar.bz2 from the lookaside cache: http://pkgs.fedoraproject.org/repo/pkgs/emacs (like Koji)

```
$ fedpkg diff
$ gitk
```

Preparing an update (2)

New upstream version:

```
$ fedpkg new-sources emacs-23.2.tar.bz2
```

- Uploads the file to the lookaside cache using https://pkgs.fedoraproject.org/repo/pkgs/upload.cgi

Adding and removing patches and helper files:

```
$ git add file1 file2
$ git rm file1 file2
```

Test builds

```
$ fedpkg local
```

- Same as `$ rpmbuild -ba emacs.spec`

```
$ fedpkg lint
```

- Same as `$ rpmlint emacs-*.rpm`

Full build with your changes without committing:

```
$ fedpkg srpm
$ fedpkg scratch-build --srpm *.src.rpm
```

Committing

Step 1: commit locally

```
$ fedpkg commit -m “Fixed a crash.”
```

- later additions and fixes: `$ git commit --amend`

Step 2: push to the Fedora server

```
$ fedpkg push
```

Both steps quickly, using a spec file change log:

```
$ fedpkg clog
$ fedpkg commit --file clog --push
```

Final build

```
$ fedpkg build
```

- All changes must be committed and pushed to the server beforehand.

Submitting a new update to bodhi:

```
$ fedpkg update
```

- Opens a text editor
- Pre-fills the bug numbers from the latest change log entry in the spec file, so use the right format #[0-9]+ (e.g. #620300) there.


Backporting
changes

The newest is fixed first!

- Rawhide → Fedora 14 → Fedora 13 → Fedora 12
- Do not update the other way around:
  - Some work is lost if the time pressure is high
  - You are updating old stabler branches sooner than new unstable branches, using them for testing

```
$ fedpkg switch-branch
$ fedpkg switch-branch f14
```

Merging

If the package is same in all branches:

```
$ git merge master
```

Otherwise you might want to backport only selected commits from the master (rawhide) branch:

```
$ git cherry-pick --no-commit master^1
```

Read gitrevisions(1) to know how to select commits.


Teamwork

Using branches

- It takes time to do some changes properly (fine-tuning a patch, build system changes)
- You can cooperate, and ask for help
- Several people reworking their patches in the same repository ⇒ disaster

Solution? Create a new local branch...

```
$ git checkout -b bz660320
```

...do the changes and commit...

Sharing the branch (1)

Share your (un)finished work:

```
$ git push origin HEAD:f14/karel-bz660320
```

- `origin` = remote repository
- `HEAD` = current local branch = `bz660320`
- `f14/karel-bz660320` = remote branch
- Other maintainers get an email about this

Sharing the branch (2)

On the other side:

```
$ git pull origin f14/karel-bz660320:k320
```

- `origin` = remote repository
- `f14/karel-bz660320` = remote branch
- `k320` = new local branch

Use fedpkg or git to switch to the new branch:

```
$ fedpkg switch-branch k320
$ git checkout k320
```

Managing branches

Merging the branch to Fedora 14 official branch:

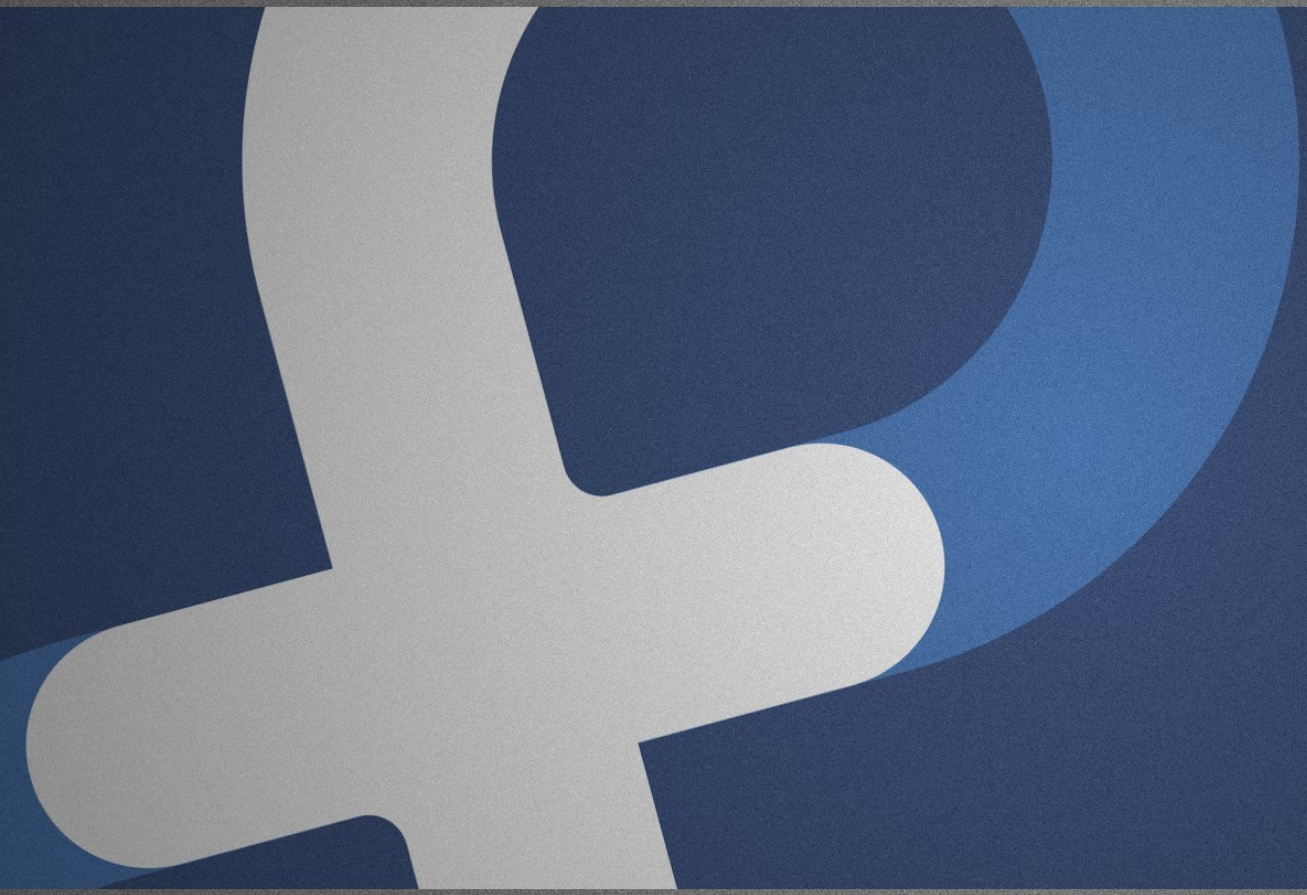
```
$ fedpkg switch-branch f14
$ git merge k320
```

Deleting the local branch:

```
$ git branch -d k320
```

Deleting the remote branch:

```
$ git branch -dr origin/f14/karel-bz660320
```



Questions?

CONTACT:
kklic@redhat.com

Creative Commons Attribution 3.0