ELIMINATING SCRIPTLETS FOR FUN AND PROFIT*

* This is a lie

WILL WOODS
Senior Software Engineer

STEPHEN GALLAGHER
Principal Software Engineer
WHAT ARE SCRIPTLETS?
WHAT ARE THEY FOR?
WHAT'S THE PROBLEM?
• **Insecure**
  - Everything runs as root

• **Hard to maintain**
  - Every package has its own unique scriptlets

• **CPU/IO overhead**
  - Makes installs & upgrades reeeeeeally slow

• **Under-designed**
  - Package versions? Exit codes? Progress reporting?

• **Opaque & Non-deterministic**
  - *We can't predict or reason about what scriptlets do!*
HEY WHAT'S THIS DO?

/usr/bin/install -m 600 /dev/null /var/log/tallylog
convert-to-edition.lua
file changed

```lua
@@ -64,7 +64,7 @@
 local function clear_presets()
     local path = "/usr/lib/systemd/system-preset"
     for file in posix.files(path) do
-       if file:match("^80-.*%.preset$") then
+       if file:match("^80%-.*%.preset$") then
         os.remove(path .. "/" .. file)
     end
 end
```
1. \texttt{%pretrans} of new package
2. \texttt{%pre} of new package
3. (\texttt{package install})
4. \texttt{%post} of new package
5. \texttt{%triggerin} of other packages (set off by installing new package)
6. \texttt{%triggerin} of new package (if any are true)
7. \texttt{%triggerun} of old package (if it's set off by uninstalling the old package)
8. \texttt{%triggerun} of other packages (set off by uninstalling old package)
9. \texttt{%preun} of old package
10. (\texttt{removal of old package})
11. \texttt{%postun} of old package
12. \texttt{%triggerpostun} of old package (if it's set off by uninstalling the old package)
13. \texttt{%triggerpostun} of other packages (if they're set off by uninstalling the old package)
14. \texttt{%posttrans} of new package
DO WE NEED THEM?
SCRIPTLET REFORM
Any sufficiently advanced technology is indistinguishable from magic.

-- Arthur C. Clarke

Sometimes magic is just someone spending more time on something than anyone else might reasonably expect.

-- R. J. Teller
SIX SCRIPTLET TASKS

1. Create users and groups
2. Enable / disable / start services
3. Create empty / placeholder files
4. Generate system-specific data
5. System configuration: SELinux, firewall, etc.
6. Update registries (caches, catalogs)
%osbuild_groupadd -g 11 miracle
%osbuild_useradd -g miracle -G wheel -u 100 -d /var/beefy -s /sbin/nologin beefy

description
Hello, DevConf!

pre
%osbuild_pre

install
%osbuild_install

files
%osbuild_files

Fedora Packaging Committee ticket: https://pagure.io/packaging-committee/issue/850
osbuild framework: https://github.com/fabrix/osbuild
SYSTEMCTL

USE SYSTEMD PRESETS!
EMPTY/DEFAULT FILES

USE TMPFILES.D SNIPPETS!
SYSTEM-SPECIFIC DATA

KEYS, CERTIFICATES, MACHINE-ID, ETC.

USE A ONE-SHOT "INITIAL SETUP" SERVICE!

https://fedoraproject.org/wiki/Packaging:Initial_Service_Setup
SYSTEM CONFIG

FIREWALL, SELINUX, KERNEL MODULES, ETC.

DON'T.

JUST.... DON'T.

(If you really think you need this, come talk to us or email Fedora's devel list so we can set you straight help find a solution)
CACHES & CATALOGS

LET'S START USING FILE TRIGGERS INSTEAD!

- Fedora 28 handles ldconfig using filetriggers
- Fedora 30 removes all scriptlets that use ldconfig!
- Coming soon: install-info, gtk-update-icon-cache, ...
- TODO: alternatives (/usr/lib/alternatives.d!)
QUESTIONS?
THANKS.