Resurrecting "cruft"

A tool for cleaning up the system and finding packaging bugs

Marcin Owsiany

June 7, 2007

Outline

1. Introduction
2. How it works
3. History and current status

What is `cruft`

A tool for finding things:

- which shouldn't exist, but do exist, or
- which should exist, but — don't

What is cruft

A tool for finding things:

- which shouldn’t exist, but do exist, or
- which should exist, but — don’t

Example

```
porridge@kielonek:~$ sudo cruft
```

What is cruft

A tool for finding things:

- which shouldn't exist, but do exist, or
- which should exist, but — don't

Example

```
porridge@kielonek:~$ sudo cruft

cruft report: Tue Jun  5 18:12:03 BST 2007

---- unexplained: / ----
        /razor-agent.log
        /etc/passwd.tmp
        /var/log/bootstrap.log

end.
porridge@kielonek:~$
```

Outline

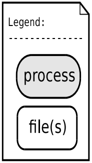
1. Introduction

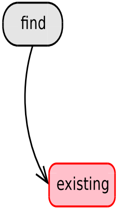
2. How it works

3. History and current status

Running find on all filesystems, except:

- explicitly ignored,
- `proc`, `nfs`, etc

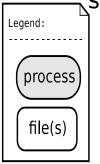


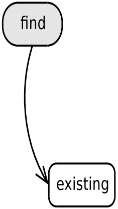


Running find on all filesystems, except:

- explicitly ignored,
- `proc`, `nfs`, etc

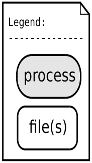
Result put into `/var/spool/cruft/file_*` (one file per filesystem scanned)

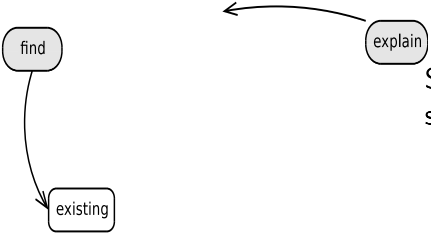




Second stage — “explain” scripts:

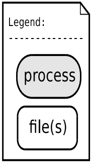
- provided by cruft, package maintainers, or sysadmin,

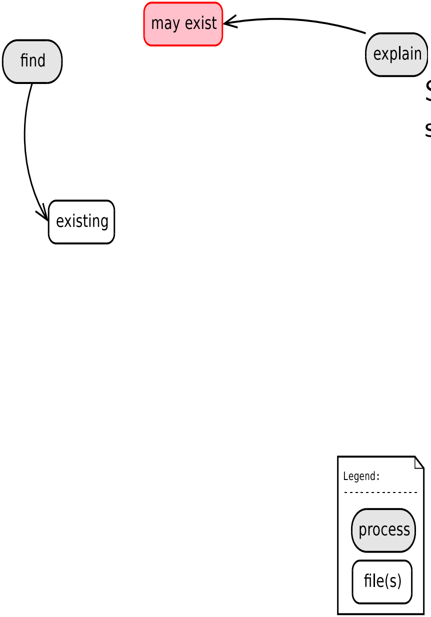




Second stage — "explain" scripts:

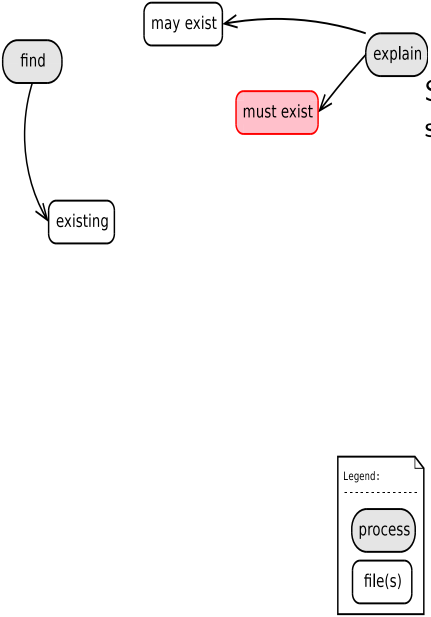
- provided by cruft, package maintainers, or sysadmin,
- each can produce 3 lists of files:





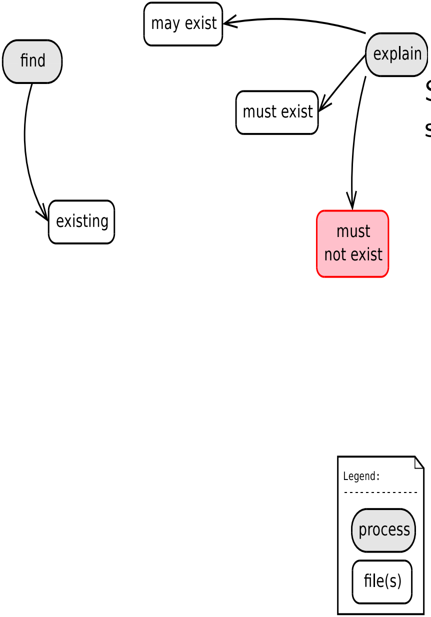
Second stage — “explain” scripts:

- provided by cruft, package maintainers, or sysadmin,
- each can produce 3 lists of files:
    - which *may* exist (logs, spools, caches),



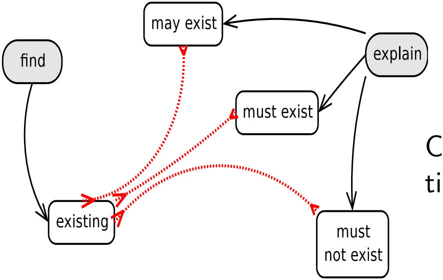
Second stage — “explain” scripts:

- provided by cruft, package maintainers, or sysadmin,
- each can produce 3 lists of files:
  - which *may* exist (logs, spools, caches),
  - which *must* exist (dpkg-installed, diversions. . .),

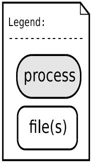

find
existing
may exist
explain
must exist
must
not exist
Legend:
process
file(s)

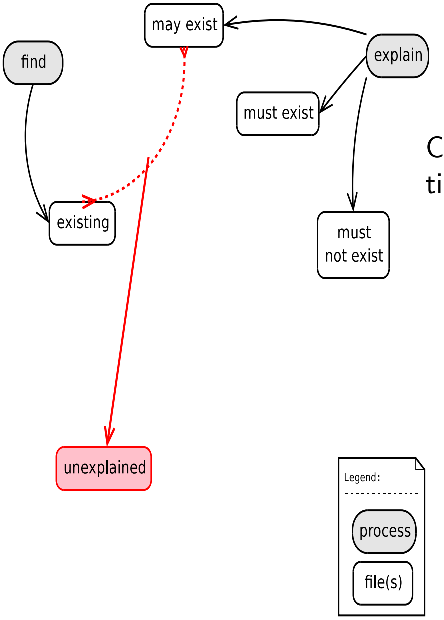
Second stage — “explain” scripts:

- provided by cruft, package maintainers, or sysadmin,
- each can produce 3 lists of files:
    - which *may* exist (logs, spools, caches),
    - which *must* exist (dpkg-installed, diversions. . .),
    - which *must not* exist (localepurge)



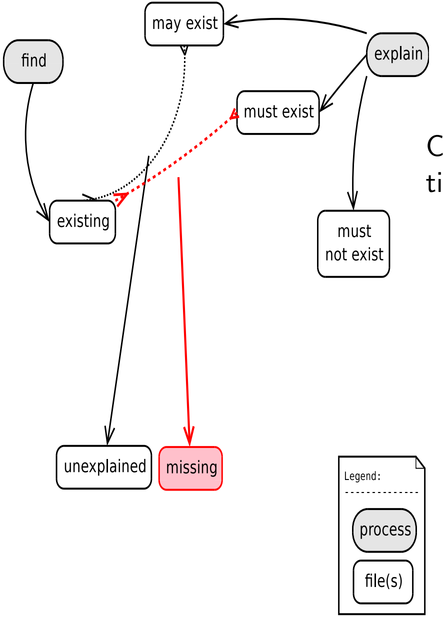
Comparing system to explanations:





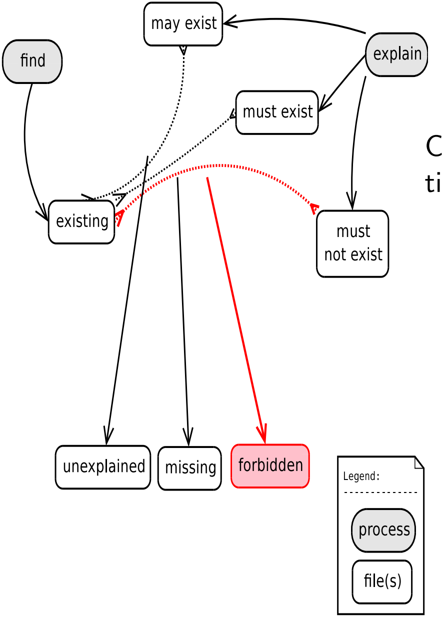
Comparing system to explanations:

- file exists, but may not → unexplained



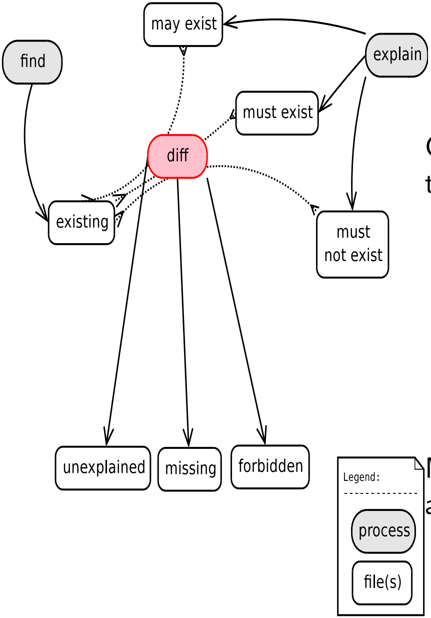
Comparing system to explanations:

- file exists, but may not → unexplained
- file must exist, but does not → missing



Comparing system to explanations:

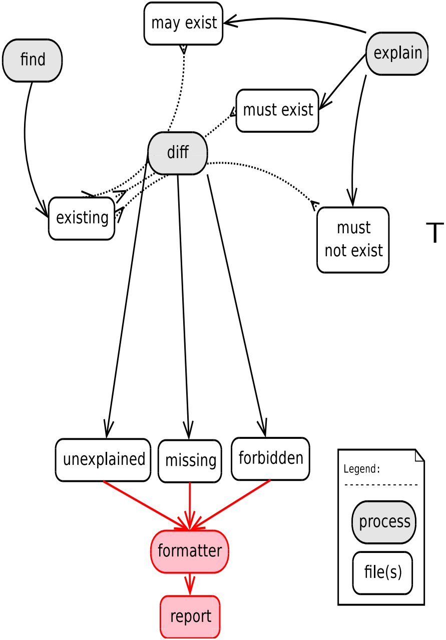
- file exists, but may not → unexplained
- file must exist, but does not → missing
- file must not exist, but does → forbidden



Comparing system to explanations:

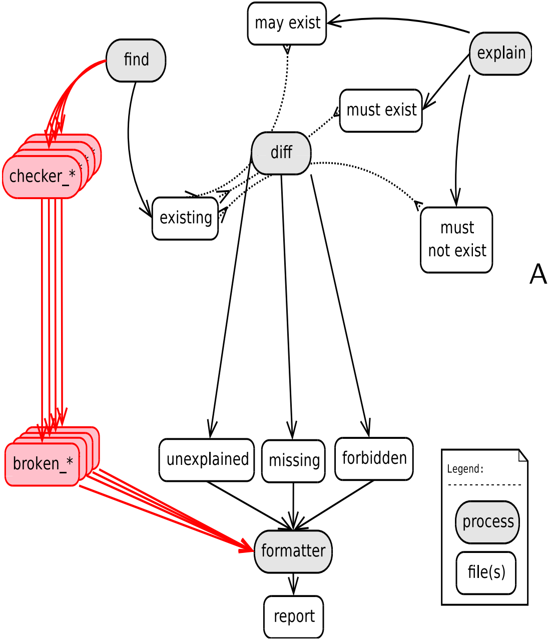
- file exists, but may not → unexplained
- file must exist, but does not → missing
- file must not exist, but does → forbidden

Modified mergesort — all three at once



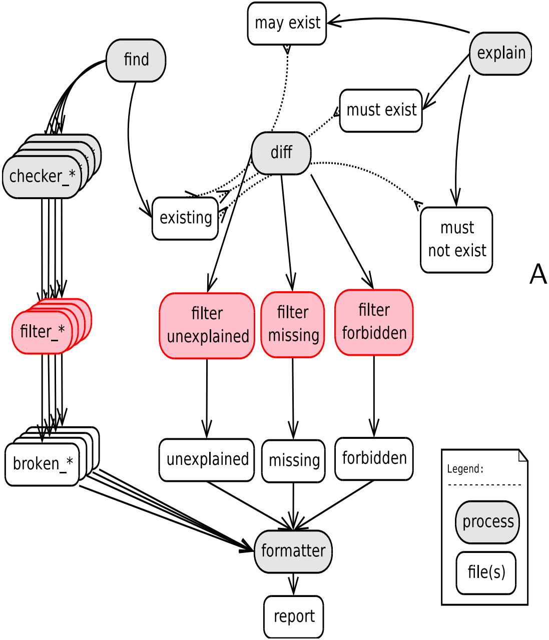
The result is:

- merged,
- formatted,
- output → e-mail/pager/MC VFS



Additional functionality:

- file “integrity” checking,



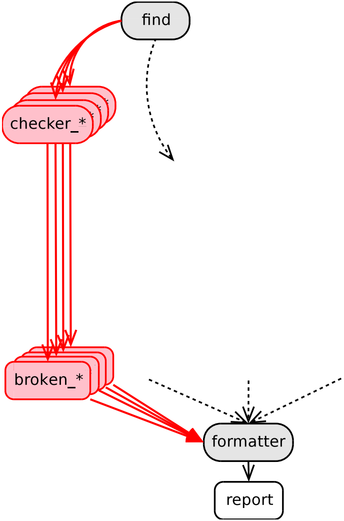
Additional functionality:

- file "integrity" checking,
- filtering

General file "integrity" checking

Mechanism:

- as `find` runs, it additionally produces lists of files of given type (e.g. all symlinks),
- a checker script for the given type is then run on resulting list,
- output from the checker subsequently added to the report



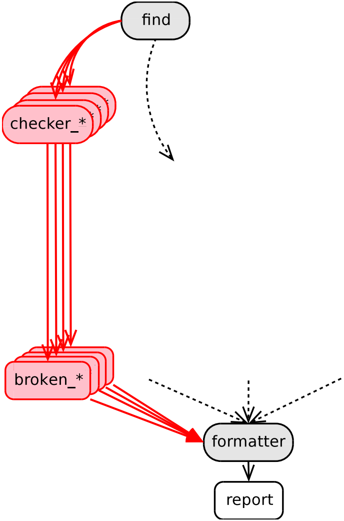
General file “integrity” checking

Mechanism:

- as `find` runs, it additionally produces lists of files of given type (e.g. all symlinks),
- a checker script for the given type is then run on resulting list,
- output from the checker subsequently added to the report

Purpose:

- currently: only checking for broken symlinks,
- potentially: doing any filetype-specific attributes check (sockets, HURD translators. . .)



Filtering

How:

- performed by a small pattern-matching program packaged with cruft,
- patterns provided in files by packagers, sysadmin or cruft

Example

```
/var/cache/apt/archives/*
/usr/share/doc/HTML/**
/var/lib/doc-base/info/*.list
```

Filtering

How:

- performed by a small pattern-matching program packaged with cruft,
- patterns provided in files by packagers, sysadmin or cruft

Example

```
/var/cache/apt/archives/*
/usr/share/doc/HTML/**
/var/lib/doc-base/info/*.list
```

Why:

- lets sysadmin ignore problematic cruft which they accept or over which they have no control (a result of a bug in package),
- helps with conflicting rules or buggy packages — (re)moving other pkgs' files, broken symlinks. . .
- will let cruft use dpkg "extrafiles" database

History

- written in early 1998 by Anthony Towns,

History

- written in early 1998 by Anthony Towns,
- shortly afterwards — policy proposal for dpkg "extrafiles";

History

- written in early 1998 by Anthony Towns,
- shortly afterwards — policy proposal for dpkg “extrafiles”; dpkg not changed, proposal abandoned,
- bit-rotting 1999-2005,

History

- written in early 1998 by Anthony Towns,
- shortly afterwards — policy proposal for dpkg "extrafiles"; dpkg not changed, proposal abandoned,
- bit-rotting 1999-2005,
- picked up by me in 2005; since then:
  - imported to alioth SVN,
  - almost all bugs fixed,
  - a few features added,
  - some code rewritten,
  - most code cleaned up
- current status can be summed up as "does not report false-positives on a fresh base installation"

Conclusions

- condition improved a lot over the last couple of years,
- still far from complete:
  - some refactoring and cleanup still necessary,
  - the huge job of covering the distribution with explain scripts and/or filter files still left

Conclusions

- condition improved a lot over the last couple of years,
- still far from complete:
  - some refactoring and cleanup still necessary,
  - the huge job of covering the distribution with explain scripts and/or filter files still left
- possibility to use as a package testing tool:
  - declare what files can be created during package installation and usage,
  - cruft checks no other filesystem changes occured,
  - could help detecting bugs such as:
    - mistakes setting up alternatives and diversions,
    - things like `/razor-agent.log` appearing when using programs

Conffiles and Configuration files (again).
http://lists.debian.org/debian-policy/1998/04/msg00032.html

PROPOSAL: Extrafiles (was Re: Conffiles...).
http://lists.debian.org/debian-policy/1998/04/msg00089.html

Cruft update.
http://lists.debian.org/debian-devel/2001/09/msg00647.html

The package “cruft” needs help.
http://lists.debian.org/debian-devel/2004/02/msg01915.html

Cruftfs wishlist bug.
http://bugs.debian.org/407475

Cruft project on Alioth.
http://alioth.debian.org/projects/cruft/