

Distributed Bug Tracking for Debian with SD

Christine Spang
christine@debian.org

Overview

- distributed bugtracking
- distributed bugtracking for Debian
- intro to SD
- SD debugs integration: present and future
- helping!
- questions (welcome along the way, of course)

Distributed bug tracking

most bug trackers are centralized

Google Code



GitHub

The background consists of several overlapping, semi-transparent shapes in various shades of brown and dark grey. The shapes are fluid and organic, creating a layered, abstract composition. The word "Trac" is centered in the middle of these shapes.

Trac



RT

Bugzilla



Debbugs

*one canonical place where all the bugs
live*

[About Debian](#)[News](#)[Getting Debian](#)[Support](#)[Developers' Corner](#)[Site map](#)[Search](#)

Debian bug tracking system

Debian has a bug tracking system (BTS) in which we file details of bugs reported by users and developers. Each bug is given a number, and is kept on file until it is marked as having been dealt with.

How to report a bug in Debian

A separate page has instructions and tips on [how to report a bug](#) if you encounter problems in the Debian distribution.

Bug tracking system documentation

- [Advanced information on how to use the system](#)
- [Information on manipulating bugs by email](#)
- [Mailservers' reference card](#)
- [Ways of accessing the bug report logs](#)
- [Requesting bug reports by email](#)

Viewing bug reports on the WWW

Find a bug by **number**:



as mbox



show all headers



show boring messages

Select bug reports on the WWW

More selections can be added after the first search. If a later selection is on the same search field, the results are ORed. If it is on a different field, the results are ANDed.



it's the future

network is ubiquitous

disks never fail

*(and even if they did, the server would
have perfect backups)*

server administrators are not evil

life is good.

... except, it's not

*network is never there
when you need it*

hardware is fallible

we have to think about who to trust

bugtracking is a network service too

*all the same issues about the cloud
apply*

*we can address some of these problems
with a distributed approach*

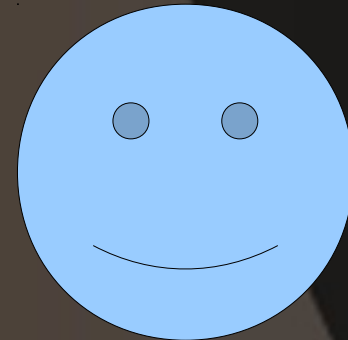
Distributed bugtracking for Debian

some issues don't really apply

we run our own servers



+ DSA =



The **Debian bug tracking system**, or **Debbugs**, is the bug tracking system used by the Debian project. It was mainly written by Ian Jackson, former Debian project leader, from January 1998 to December 1998.

but we have other unique issues

The **Debian bug tracking system**, or **Debbugs**, is the bug tracking system used by the Debian project. It was mainly written by Ian Jackson, former Debian project leader, from January 1998 to December 1998.

we want to integrate with upstream

sometimes we ARE upstream

*goal: pull together bugs from
different sources*

*goal: the bug tracker should be simple
and get out of your way*

SD (Simple Defects)

*a distributed bugtracker built on
Prophet, a peer-to-peer property
database*

<http://syncwith.us/>

*bug data is kept in a local database,
can be synced back and forth between
multiple sources (including ones that
aren't SD)*

<sd demo>

I hate all bugtrackers.

but I almost like SD

two main cases for Debian

1. Debian-maintainer only

- simple interface, command-line or local web
- grab all the bugs you care about easily, keep them in the same SD database
 - `SD_REPO=debiandsd sd clone --from "debugs:http://localhost:7912/|maint=christine@debian.org"`

2. *Debian+upstream*

- separate database per package containing all the bugs in Debian AND upstream for that piece of software
- could be extended to include “sidestream” sources
 - other distros
 - tricky: linking together associated bugs
- no limit to how many different places you can pull bugs from

SD/debbugs integration: current state

very alpha read-only support

*supports downloading individual bugs
and all queries to get_bugs soap
method*

<http://wiki.debian.org/DebbugsSoapInterface>

incomplete history

*depends on a patched debbugs:
new soap call
'get_machine_readable_bug_log'*

<sd/debbugs/soap demo>

***SD/debbugs integration: where we
want to be***

full history available locally

full read/write support

*(implementation: read/write soap?
email?)*

pony: pull together bugs from every source related to a piece of software, work on locally, link, push back as necessary

implementation hurdles

making debugs record better log data

commands missing machine-readable

comments in buglogs:

- forwarded*
- merge*
- owner*
- severity*
- probably others*

graceful fallback for older bugs

major SD feature blocker: partial push

*a standardized data format / API
so we don't have to write a bridge
for every bug tracker out there*

How you can help

- SD polish and blockers
- Documenting how to write foreign replica interfaces for SD
- Debbugs cleanup and modularisation
- Completing debbugs action recording in bug logs
- Writing tests for the new SOAP method
- Deep thinking about merging, standardized API for bug trackers, etc.

Getting the code

```
git clone git://github.com/bestpractical/sd.git  
branch 'debbugs'
```

```
bzr branch http://spang.cc/data/debbugs
```

```
aptitude install debbugs-local/experimental
```

Get in touch

Christine Spang

email: christine@debian.org
irc.debian.org: christine