



Contents

1. [DSO Linking Changes for wheezy](#)
 1. [Not resolving symbols in indirect dependent shared libraries](#)
 2. [Unresolved symbols in shared libraries](#)
 3. [Only link with needed libraries](#)
 4. [Updated toolchain packages](#)
 5. [Changes required for packages](#)
 6. [Further information](#)

DSO Linking Changes for wheezy

Currently the BFD based linker uses indirect dependent shared libraries to resolve symbols when linking shared libraries or binaries. This situation has a hidden violation of encapsulation. The gold linker doesn't use the indirect shared libraries to resolve symbols by default. Change the default of the BFD linker to behave like gold.

Currently the linker adds DT_NEEDED entries for every library that is mentioned on the command line, even if no symbols are needed from a library. This leads to dependencies on library packages which are not needed. Reducing the number of dependencies for binary packages makes library transitions easier, reduces the amount of binNMUs needed for a library transition, and eases migration of packages to testing. Change the default to only link libraries as needed.

Some packages will fail to build. Binaries, which are using symbols from an indirectly linked shared library will fail to link and need to be fixed to link with the shared library directly, and some binaries like plugins have to be linked with all libraries as given on the command line.

Not resolving symbols in indirect dependent shared libraries

Some packages fail to build with `ld.gold` or `ld.bfd --no-add-needed/--no-copy-dt-needed-entries` due to undefined reference. This is in most cases the problem that your executable links against a library A which links against library B, but your executable needs symbols in library B. This is problematic in the situation where library A removes its dependency to library B. The next time the executable gets rebuild it will break and cannot be linked.

There are safety/correctness issues too, outlined at [\[1\]](#). Namely, this makes it possible to create binaries that use symbols from both `libfoo` and `libbar` when `libfoo` itself links to `libbar` (because `ld` will no longer skip the direct linkage based on the presence of indirect linkage), and not have those binaries break if `libfoo` ever stops linking to `libbar`. The current situation has a hidden violation of encapsulation.

This doesn't affect the created executable as it will still have entries in `DT_NEEDED` for both library A and library B in the case of a successful build. It isn't known which will be the preferred linker in the future or if the behavior will be changed in GNU `ld` or GNU `gold`.

All packages can be fixed by adding the needed library which provides the symbols to the linker call. In some cases the `pkg-config` file of library A could be broken as it requires library B for the build. This could be the case if the executable doesn't use library B directly in its source code, but library A inserts the symbols of library B using `defines` or other similar mechanisms inside their header files to executable, but hasn't the correct `Requires` statement inside the `.pc` file of library B.

A common problem seems to be calls to `XOpenDisplay` without directly linking to `-lX11`, but to a library like `Xtst`. Similar is to use `glib2.0`, but only linking to `gtk2.0`: using `libm`, but not linking to it; `libqt*`, but only linking to `kdecore`. Another interesting problem is that some packages link their C++ object files using `gcc` instead of `g++`, but the `g++` symbols get resolved by another C++ library they link to.

These issues are tracked in <http://bugs.debian.org/cgi-bin/pkgreport.cgi?tag=no-add-needed;>

[users=peter.fritzsche@gmx.de](mailto:peter.fritzsche@gmx.de) When submitting new reports, please tag these:

- User:  peter.fritzsche@gmx.de
- Usertags: no-add-needed

Unresolved symbols in shared libraries

There are unresolved symbols in shared libraries, which have unresolved symbols. In most cases the library should be linked against the library providing the missing symbols, such that other packages can link against it without problems (i.e. without the necessity linking to the missing library itself, even if it doesn't use any of it's symbols), because the shared object has still some unresolved symbols which the program which links against it must resolve. This isnt a good idea because when you introduce new dependencies the package previously linked against the old version will break because it doesnt know about the new dependency.

dpkg-shlibdeps warns about these in the form:

```
dpkg-shlibdeps: warning: symbol <symbol> used by debian/<foo>/usr/lib/<libfoo> found in none of the libraries.
```

Note that sometimes symbols are unresolved by intent (e.g. libcgic, bug #558872).

These issues are tracked in  <http://bugs.debian.org/cgi-bin/pkgreport.cgi?tag=unresolved-symbols-so;users=peter.fritzsche@gmx.de> When submitting new reports, please tag these:

- User:  peter.fritzsche@gmx.de
- Usertags: unresolved-symbols-so

Only link with needed libraries

dpkg-shlibdeps already warns if a library is not needed to resolve symbols in a binary. For the majority of packages it is safe to link with `--as-needed`, reducing the dependencies of a package. `--as-needed` is a positional parameter, and can be given multiple times on the command line. To add an explicit dependency on a library use `--no-as-needed -lfoo --as-needed`.

OpenSuse does build the packages by default with `--as-needed`. A list of packages which are not built with `--as-needed` is found below (fixme: the reasons why `--as-needed` was undone are mostly unknown, maybe even just a quick "fix" by the package maintainer). Feel free to add pointers to more information for these packages. The default of passing `--as-needed` can be undone by passing `--no-as-needed` to the linker (`-Wl,--no-as-needed`). Don't confuse this with `--no-add-needed`.

Introducing `--as-needed` will show some sloppy build files and configure scripts, which fail to build with `--as-needed`.

- In configure scripts libraries should be added to LIBS, not LDFLAGS.
- Object files, including .la files from a package build must appear before any library on the command line.

Removing libraries due to `--as-needed` might also produce binaries with different behaviour at run time only. Though only in corner cases using ELF specialities like overriding symbols from other libraries (for example Xt programs no longer pulling in Xaw might silently lose the ability to speak to editres).

- MozillaSunbird
- MozillaThunderbird
- OpenOffice.org
- WindowMaker-applets
- capisuite

- cpufrequtils
- cups-drivers
- dirac
- dvbtune
- ebttables
- encfs
- firescope
- firmwarekit
- freeradius-server
- gfxboot
- giggle
- gpsdrive
- heartbeat
- hwinfo
- i4l-base
- ibmasm
- input-utils
- ivtv
- jack-rack
- kexec-tools
- ksh
- ladspa
- libapparmor1
- libcppunit-1_12-0
- libjingle
- libmicro
- libopensync-plugin-python-module
- libpst
- libxspf
- loki_setup

- m17n-lib
- mISDNuser
- mbuffer
- meterbridge
- multipath-tools
- mysql-workbench
- nspluginwrapper
- openssh
- ozerocdoff
- pacemaker
- pam_ccreds
- pax-utils
- pcsc-gempc
- perf
- pinfo
- policycoreutils
- pommed
- prelink
- pulseaudio
- python-sip
- python3
- qterm
- qtpcr
- samba
- sax2
- seamonkey
- servicelog
- simias
- skim
- stardict-tools

- statserial
- subversion
- syslog-ng
- turnpike
- unison
- virtualbox-ose
- wavemon
- wvdial
- wvstreams
- xalan-c
- xawtv
- xfce4-panel-plugins
- xgalaga
- xiterm
- xstereograph
- yast2-core
- BASE/binutils (testsuite)
- BASE/device-mapper
- BASE/gcc (gcj plugins)
- BASE/gpm
- BASE/perl
- BASE/python
- GNOME/dconf
- GNOME/gir-repository
- GNOME/gnome-libs
- GNOME/gtkam
- GNOME/kompozer
- GNOME/mail-notification
- KDE/celestia
- KDE/decibel

These issues are tracked in <http://bugs.debian.org/cgi-bin/pkgreport.cgi?tag=ld-as-needed;users=debian-gcc@lists.debian.org> When submitting new reports, please tag these:

- User: debian-gcc@lists.debian.org
- Usertags: ld-as-needed

Updated toolchain packages

Updated toolchain packages are now available in unstable.

Changes required for packages

The preferred way to not indirectly resolve symbols is to add the missing libraries to the link step to resolve the explicitly referenced symbols.

For the --as-needed default change, a workaround/fix would be to link with `-Wl,--no-as-needed -lfoo -Wl,-as-needed`. Note that the --as-needed and --no-as-needed are positional parameters and the default behaviour should be restored after using this workaround.

Further information

- Builds failures (originally with ld.gold) are tracked in Debian:
 - <http://bugs.debian.org/cgi-bin/pkgreport.cgi?tag=no-add-needed;users=peter.fritzsche@gmx.de>
 - More info: <http://wiki.debian.org/ga.debian.org/FTBFS#A2009-11-02Packagesfailingbecausebinutils-go>
- The Fedora Wiki offers information on a similar change:
 - [1] <https://fedoraproject.org/wiki/UnderstandingDSOLinkChange>

- <http://fedoraproject.org/wiki/Features/ChangeImplicitDSOLinking>