

# LibreOffice and IDEs

## ▼ Bjoern Michaelsen

- ▼ LibreOffice Ubuntu Package Maintainer

bjoern.michaelsen@canonical.com

- ▼ Deputy Member, Board of Directors, the Document Foundation

bjoern.michaelsen@documentfoundation.org



All text and image content in this document is licensed under the [Creative Commons Attribution-Share Alike 3.0 License](#) (unless otherwise specified). "LibreOffice" and "The Document Foundation" are registered trademarks. Their respective logos and icons are subject to international copyright laws. The use of these therefore is subject to the [trademark policy](#).

# OLD



# OLD

## Build Environment Effort: Goals

- ▼ Create a compelling, user friendly build experience by making building OOo less complex, stable and faster
- ▼ Concrete goals:
  - ▼ Improve the build performance
  - ▼ Make the build reliable even for non-experts
  - ▼ Better support for execution of tests in the build
  - ▼ Create runnable office in the build

# OLD

## A Way to Do It Better?



- ▼ Problems in the current system
  - ▼ manual dependencies
  - ▼ incompatible builds
  - ▼ scalability
  - ▼ TIMTOWTDI
- ▼ Consequences
  - ▼ broken dependencies
  - ▼ horizon effects
  - ▼ build segmentation
  - ▼ bad quality of makefiles



# OLD

## Requirements for a new Build System

- ▼ Scalability/Parallelization
- ▼ Lean dependencies
- ▼ No custom tooling
- ▼ Full dependencies
- ▼ Easy to use
- ▼ Code quality
- ▼ Multiple repository support

# OLD

## GNU Make Build System: NoOp Performance



Modules: framework, sfx2, svl, svtools, sw, toolkit, tools, xmloff

# Dont repeat yourself (DRY)

In software engineering, don't repeat yourself (DRY) is a principle of software development, a principle aimed at reducing repetition of information of all kinds, especially useful in multi-tier architectures. The DRY principle is stated as "Every piece of knowledge must have a single, unambiguous, authoritative representation within a system."

– *Wikipedia on DRY*

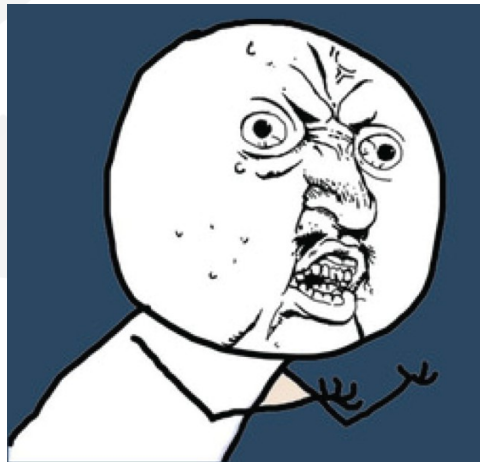
In software engineering, don't repeat yourself (DRY) is a principle of software development, a principle aimed at reducing repetition of information of all kinds, especially useful in multi-tier architectures. The DRY principle is stated as "Every piece of knowledge must have a single, unambiguous, authoritative representation within a system."

– *Wikipedia on DRY, again*

# Dont repeat yourself (DRY)

To add a library to LibreOffice now:

- ▼ Add a build description
- ▼ Register in Repository.mk



To add a library in LibreOffice 3.3:

- ▼ Add a build description
- ▼ Register in solenv, and explain among other things:
  - ▼ The rpaths of the library
  - ▼ The name of the library for others to link against it
- ▼ Possible add files to explain which symbols the library should export
- ▼ Add to deliver.lst
- ▼ Add to scp2, and explain:
  - ▼ where the library will end up in a installation

# TIMTOWTDI, read-only code and consistency

“There is more than one way to do it!”

“... but to read the code, you have to know them all!”



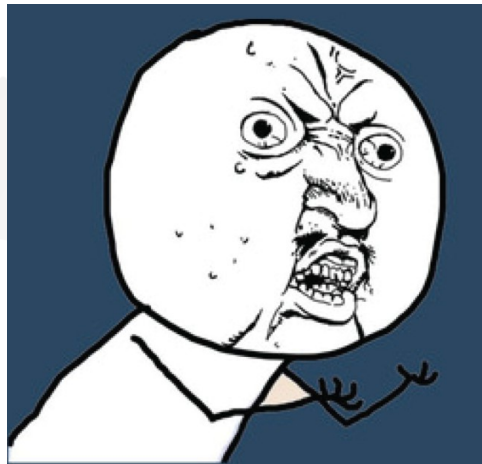
Whenever the C++ language designers had two competing ideas as to how they should solve some problem, they said, “OK, we’ll do them both”. So the language is too baroque for my taste.

— *Donald Knuth*

# TIMTOWTDI, read-only code and consistency

To export a symbol in LibreOffice now:

- ▼ Add `${LIBNAME}_DLLPUBLIC` to the declaration



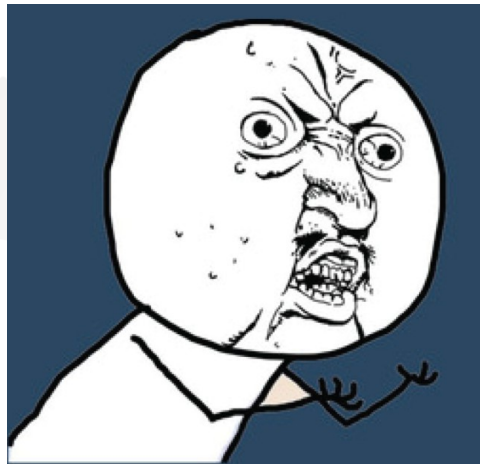
To export a symbol in LibreOffice 3.3:

- ▼ It depends.
- ▼ You might need to add `${LIBNAME}_DLLPUBLIC` to the declaration.
- ▼ Alternatively, you might need to add the platform-dependent mangled name of the symbol to an export file and hope that the little Perl script of ours unmangles and remangles the symbol name right on all platforms.
- ▼ Possibly you need to do both.
- ▼ Or something completely different.

# TIMTOWTDI, read-only code and consistency

Where is my Library in LibreOffice now?

- ▼ In instdir in exactly the place where it would end up in an installation
- ▼ (the same file is also hardlinked from workdir/)

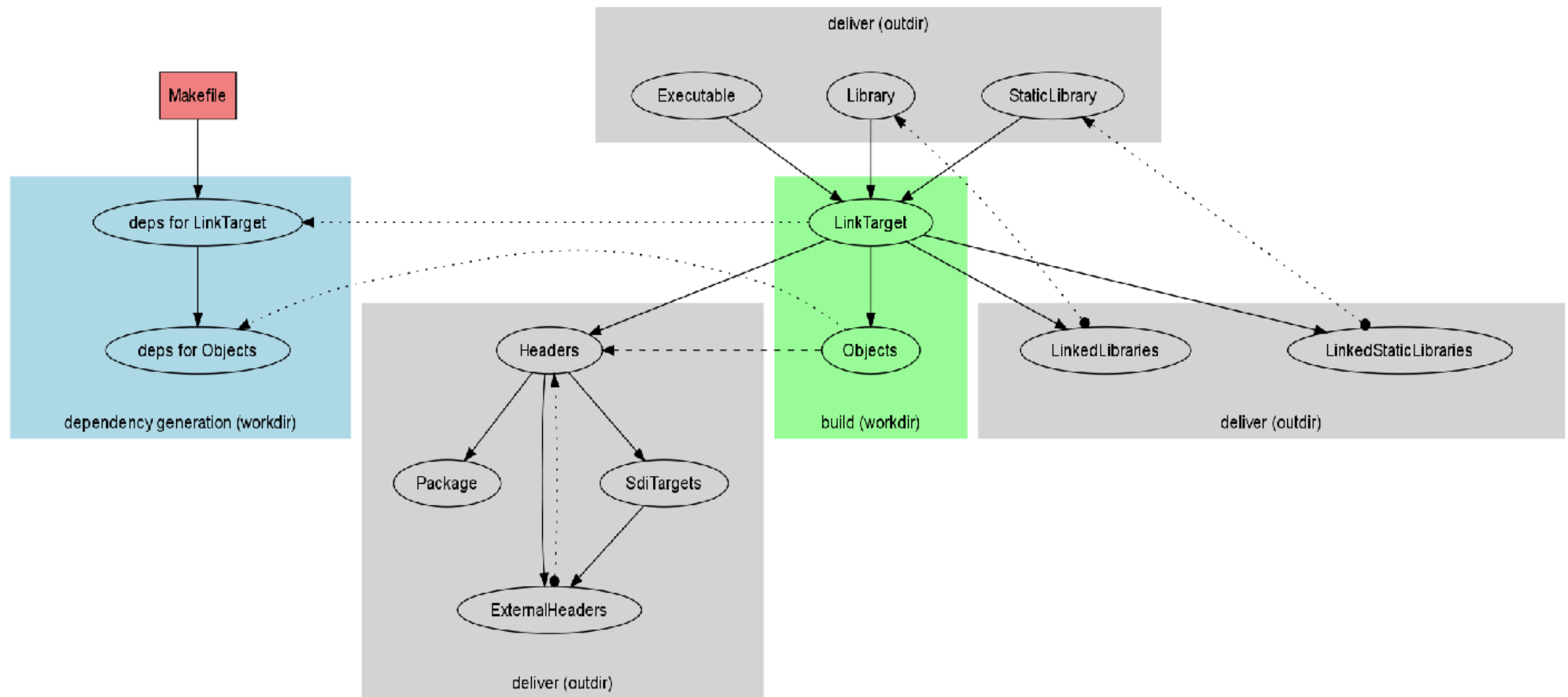


Where is my Library in LibreOffice 3.3:

- ▼ In solver/\${something}/lib
- ▼ Possibly also in \${module}/\${something}/lib
- ▼ Possibly also in workdir/
- ▼ Finally also in an installation dir
  - ▼ Which cant be incrementally updated
  - ▼ Thus it always needs to be cleared completely and then repopulated completely by scp2, which is slow and not part of the default build

# OLD

## GNU make prototype: example





# Thanks to everyone working on gbuild!

commit 2b791f1cc51eaad25bd3464f94231fe4b236fae6

Author: Bjoern Michaelson <bjoern.michaelson@canonical.com>

Date: Mon Feb 25 11:56:21 2013 +0100

remove dmake, build.pl, deliver.pl and zipdep.pl

This is writing history: LibreOffice builds in ONE non-recursive make process with full dependencies. We will now be able to really move forward without the old build system. A big 'Thank you!' goes out to everyone contributing to solenv/gbuild, especially:

- David Tardon
- Norbert Thiebaud
- Tor Lillqvist
- Michael Stahl
- Matúš Kukan
- Stephan Bergmann
- Luboš Luňák
- Caolán McNamara
- Mathias Bauer
- Jan Holesovsky
- Peter Foley
- Andras Timar
- Hans-Joachim Lankenau

and all the heroes migrating all the modules of LibreOffice to gbuild.

By explicit request this commit has to be completed with this quote:

I say we take off and nuke the entire site from orbit.  
It's the only way to be sure.

Hold on a second.

This installation has a substantial dollar value attached to it.

They can bill me.

Change-Id: I72fa17cfae00ca78cfe0eb5782c1788d2dcc

Reviewed-on: <https://gerrit.libreoffice.org/2445>

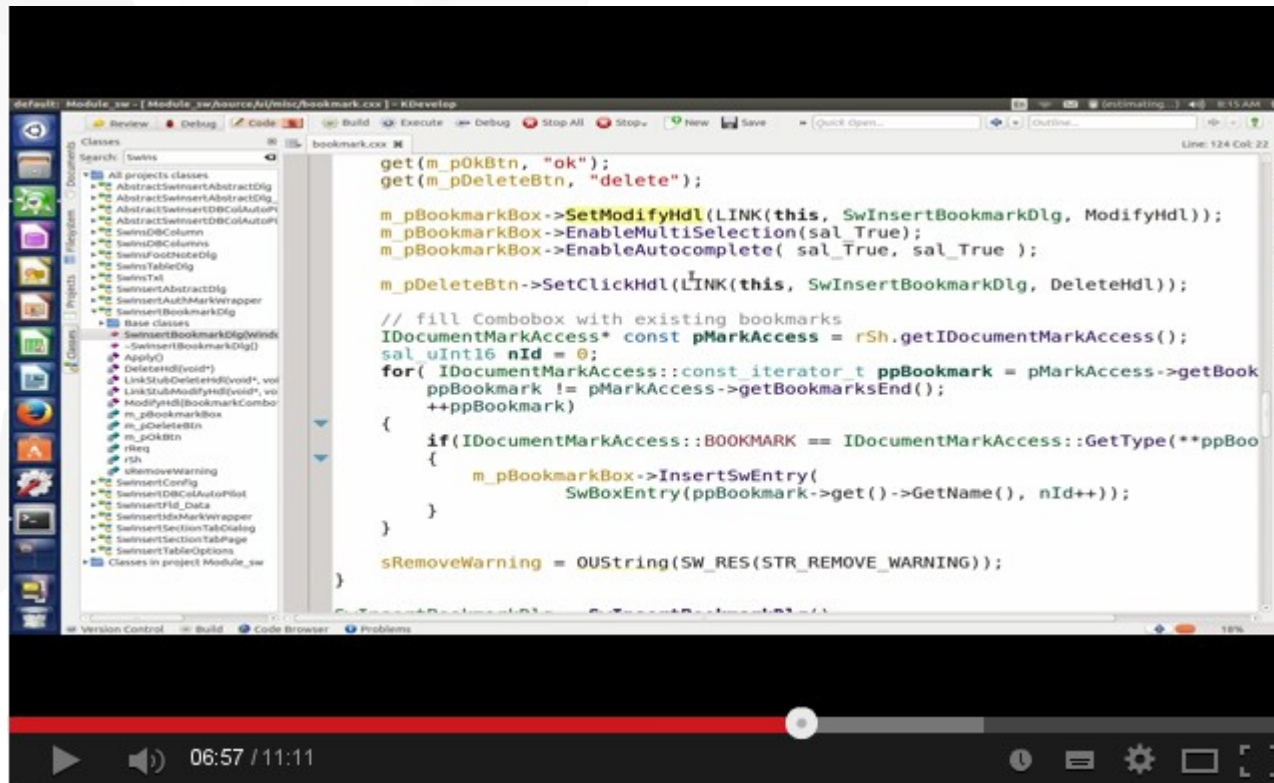
Tested-by: LibreOffice gerrit bot <gerrit@libreoffice.org>

Reviewed-by: Niko Rönkkö <ronkko@iki.fi>

Reviewed-by: Norbert Thiebaud <nthiebaud@gmail.com>

# IDE integration

make kdevelop-ide-integration



see: <https://wiki.documentfoundation.org/Development/BuildingOnLinux>

# gbuild-to-ide (first implementation)

- ▼ ran another “make -np”
- ▼ needed a full build
- ▼ make output turned out to be quite inconsistent over releases
  - ▼ result: flaky

# gbuidtojson

- ▼ a “fake platform”
- ▼ does not need a full build
- ▼ creates stable JSON output in WORKDIR
- ▼ JSON files are then used to create IDE solutions

# IDEs

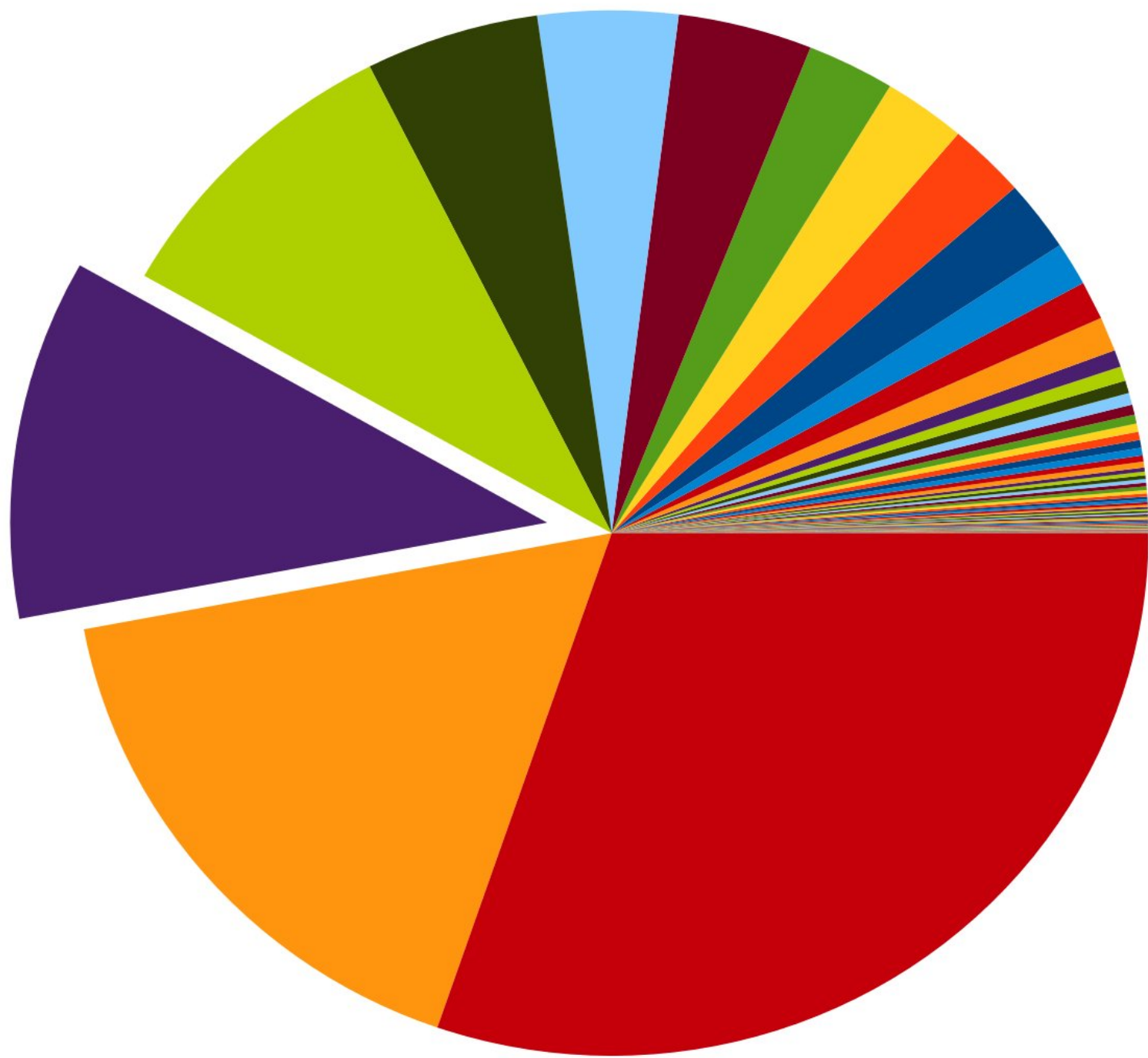
- ▼ KDevelop (me, various)
  - ▼ Building, Debugging, Code Completion
- ▼ Visual Studio (Honza Havlocek, various)
  - ▼ Building, Debugging, Code Completion
- ▼ Xcode (Jan, initial work by Tor)
  - ▼ Building Debugging, Code Completion (?)
- ▼ Vim & YouCompleteMe (Moggi)
  - ▼ Building, Code Completion
- ▼ QtCreator, Eclipse ...

# The abyss gazing back

*Wer mit Ungeheuern kämpft, mag zusehn, dass er nicht dabei zum Ungeheuer wird. Und wenn du lange in einen Abgrund blickst, blickt der Abgrund auch in dich hinein.*

*– Nietzsche, Jenseits von Gut und Boese*

*He who fights with monsters should look to it that he himself does not become a monster. And if you gaze long into an abyss, the abyss also gazes into you.*







# OLD

## Current Build System: Problems

- ▼ Parallelization
- ▼ Full dependencies
  - ▼ module borders
  - ▼ “compatible builds”
- ▼ Cargo cult programming in makefiles
- ▼ Source tree is not read-only
- ▼ Lots and lots of process instantiations
- ▼ Too many tools for the same job
- ▼ Maintenance of “homegrown” tools

No DRY because we ignored preventing TIMTOWTDI

# “native build” – challenges

- ▼ Having multiple sources of Truth sucks
- ▼ Having multiple build systems requires syncing
- ▼ Having autogenerated intermediates sucks
- ▼ Custom Targets
- ▼ Degrees of Freedom in Config

# “this is not an exit” – replacing gbuild

- ▼ Configuration Checks:
  - ▼ Cmake
- ▼ High Level Build Description:
  - ▼ Cmake, GNU make & guile
- ▼ Build Execution/Driver:
  - ▼ ninja

## Contact me

- ▼ <http://skyfromme.wordpress.com>
- ▼ Sweetshark on #libreoffice-dev/#libreoffice-qa on freenode
- ▼ Sweet5hark on twitter



All text and image content in this document is licensed under the [Creative Commons Attribution-Share Alike 3.0 License](#) (unless otherwise specified). "LibreOffice" and "The Document Foundation" are registered trademarks. Their respective logos and icons are subject to international copyright laws. The use of these therefore is subject to the [trademark policy](#).

# Agenda

- ▼ Recursive make considered harmful
- ▼ Dont repeat yourself (DRY)
- ▼ TIMTOWTDI, read-only code and consistency

# gbuild

- ▼ solenv/gbuild
- ▼ solenv/gbuild/platform
- ▼ solenv/gbuild/extensions
- ▼ \${MODULE}/\*.mk

# Recursive make considered harmful

To build LibreOffice now:

- ▼ make

To build sw in LibreOffice now

- ▼ make sw

▼ To build LibreOffice in 3.3:

- ▼ Execute a bash
- ▼ source \*Env.Set.sh
- ▼ cd instsetoo\_native
- ▼ build --all -P8 -- -P8

To build sw in LibreOffice 3.3:

- ▼ Execute a bash
- ▼ source \*Env.Set.sh
- ▼ cd sw
- ▼ build -P8 -- -P8
- ▼ deliver

# Recursive make considered harmful

Current speed hacks:

- ▼ Concat-deps
  - ▼ tiny native C program
  - ▼ creates one dependency file per library
    - ▼ less files to open for make
    - ▼ removes duplicates and cruft along the way
  - ▼ creates the initial (empty) per object dependency files
    - ▼ this speeds up the build on Windows by 10 minutes (mostly independent of hardware)
- ▼ Only relink libraries on public API change
- ▼ Instdir: drive-by populating a relocatable runnable instance



# Recursive make considered harmful

Further speed hacks (work in progress):

- ▼ Depcache
  - ▼ Based on unpatched upstream GNU make 4.0:
    - ▼ libswlo.so dependency file:
      - ▼ Reading simple include: 1.038s
      - ▼ Reading cached file: 0.257s (**>400% speedup**)
- ▼ VPATH – the undiscovered country

# TIMTOWTDI, read-only code and consistency

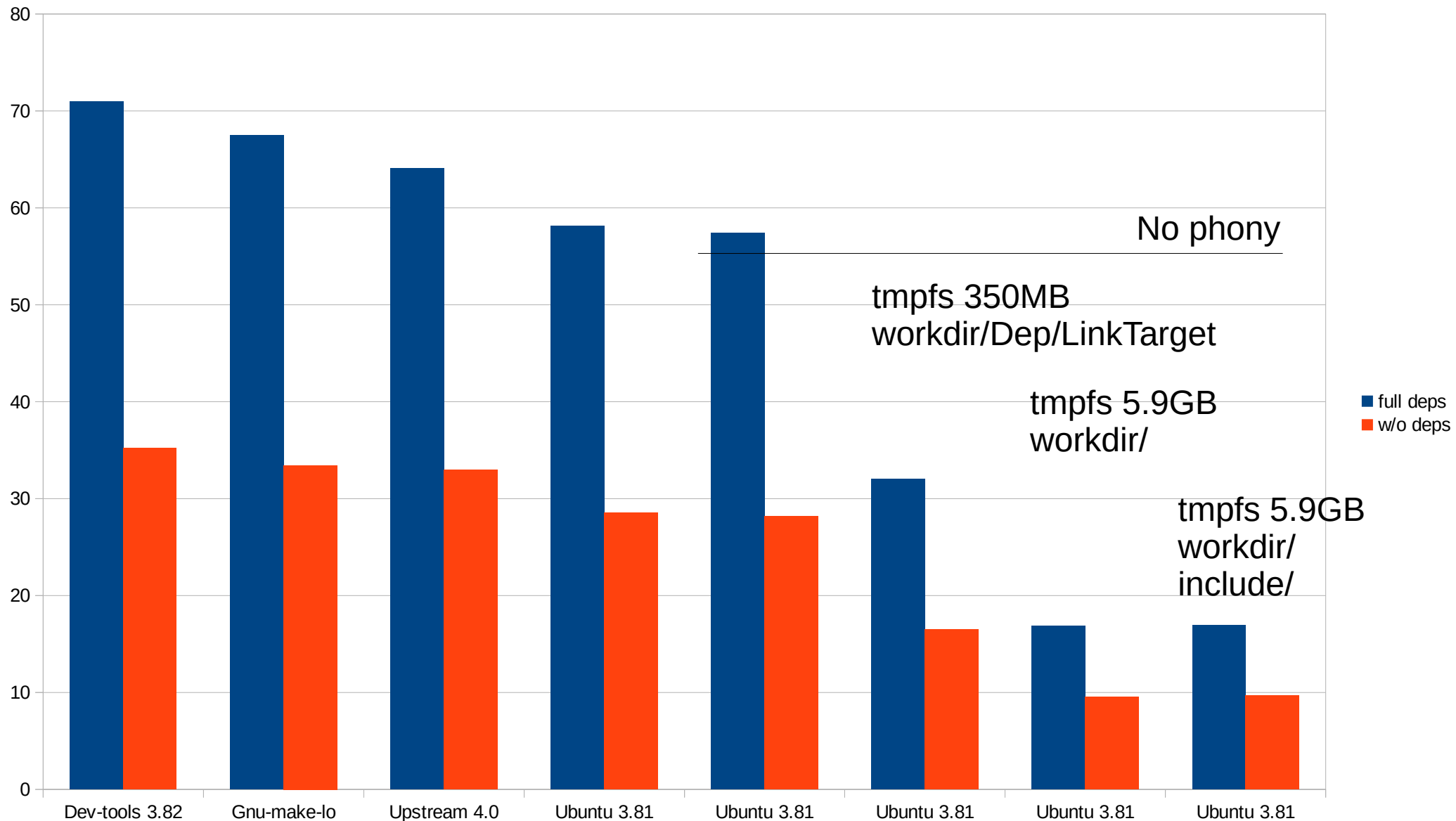
Apropos scp2, this is WIP:

- ▼ We want to kill it with fire  
(announced by Michael Stahl on the LibreOffice conference, Milan)
- ▼ Now that we have an installation populated during the build, install directly from there
- ▼ Less differences between what developers see in their builds and what ends up at the user
- ▼ And one copy less even for packagers



- ▼ Status quo:
  - ▼ We can now build and package localization independently (shipped with LibreOffice 4.2)
  - ▼ There is initial work for doing the same with the binary packages too.

# Recursive make considered harmful



# Recursive make considered harmful

“This paper explores an number of problems regarding the use of recursive make, and shows that they are all symptoms of the same problem. Symptoms that the UNIX community have long accepted as a fact of life, but which need not be endured any longer. These problems include

- ▼ recursive makes which take “forever” to work out that they need to do nothing,
- ▼ recursive makes which do too much,
- ▼ or too little,
- ▼ recursive makes which are overly sensitive to changes in the source code
- ▼ and require constant Makefile intervention to keep them working.”

– Peter Miller, 1997

<http://miller.emu.id.au/pmiller/books/rmch/>

# Recursive make considered harmful

## What we did back in a LibreOffice 3.3 release build:

- ▼ Call autogen.sh (which as you suspect is a Perl script)
  - ▼ Which calls autoconf to generate ./configure
    - ▼ Which configures a patchsystem and prepares a second inner ./configure
- ▼ Call make
  - ▼ Which unpacks the source
  - ▼ Which patches the source
  - ▼ Which call the second (inner) ./configure
    - ▼ Which generates a Perlscript called set\_soenv
    - ▼ Calls set\_soenv which generates a LinuxX86-64Env.Set.sh
  - ▼ Executes a shell
    - ▼ Which sources the LinuxX86-64Env.Set.sh
    - ▼ Which then calls a ./bootstrap script
      - ▼ Which among other things, compiles a dmake binary
    - ▼ Calls a Perl script called "build.pl"
      - ▼ Which calls a the freshly build dmake binary in directories
        - ▼ Which (hopefully) start compiling your first bits of code ...

