



Autotools Basics

Navigating the auto-jungle

Kristian Lyngstol

Redpill Linpro AS

January 21, 2010



Outline

- 1 Introduction
- 2 autogen.sh and auto-tools evocation
- 3 Getting started
- 4 Demonstration
- 5 Portable code
- 6 Finding the macro



What you will learn

You will hopefully be able to:

- Get started on new/migrated projects
- Work with the build system of existing projects
- Debug broken build systems

You will not learn:

- Every single m4-macro available for auto-*
- Every detail of make
- Using autotools to build libraries (ie: libtool)



What is Auto-tools and Make?

Make

- Generic, simple and available
- Tracks source-file changes

Auto-tools

- Collection of various tools: autoconf, autoheader, automake, aclocal, etc.
- GNU-specific
- Augments make
- Generates a portable make-environment



Or....

```
./configure  
make  
make install
```



Introduction to make

- Make is similar to a deterministic language
- Implicit and explicit rules

Example

```
binary: object1.o object2.o
    gcc -Wall -o binary object1.o object2.o

install: binary
    cp whatyouwant /usr/local/bin

all: whatyouwant install
```



Some Makefile details

Example

```
.c.o:  
    gcc -Wall -g -o $@ $<  
.xml.html:  
    xsltproc --xinclude -o $@ $<
```



Some Makefile details

Example

```
.c.o:  
    gcc -Wall -g -o $@ $<  
.xml.html:  
    xsltproc --xinclude -o $@ $<
```

Example

```
help:  
    @echo Some help  
install:  
    cp src/binary /usr/local/bin/  
.PHONY: help install
```




Introduction to auto-tools

- “make” does the building using “Makefile”
- “automake” generates portable “Makefile”-templates: “Makefile.in”
- “configure” generates system-specific Makefiles from Makefile.in and config.h
- “autoconf” generates portable configure-scripts



Why?



Why?

- Because FreeBSD doesn't have epoll and Linux doesn't have kqueue



Why?

- Because FreeBSD doesn't have epoll and Linux doesn't have kqueue
- Because build systems are beyond boring



Introduction to auto-tools - timeline

- Write `configure.ac` `Makefile.am`
- `$ aclocal # Basic m4-macros`
- `$ autoheader # config.h.in`
- `$ automake # Makefile.in`
- `$ autoconf # configure`
- Distribute
- `$./configure # Makefile, config.h`
- `$ make # Builds what's necessary`
- `$ make install # Builds if necessary and installs`



Introduction to auto-tools - timeline

- [Write configure.ac Makefile.am](#)
- \$ `aclocal # Basic m4-macros`
- \$ `autoheader # config.h.in`
- \$ `automake # Makefile.in`
- \$ `autoconf # configure`
- Distribute
- \$ `./configure # Makefile, config.h`
- \$ `make # Builds what's necessary`
- \$ `make install # Builds if necessary and installs`



Introduction to auto-tools - timeline

- Write `configure.ac` `Makefile.am`
- `$ aclocal # Basic m4-macros`
- `$ autoheader # config.h.in`
- `$ automake # Makefile.in`
- `$ autoconf # configure`
- Distribute
- `$./configure # Makefile, config.h`
- `$ make # Builds what's necessary`
- `$ make install # Builds if necessary and installs`



Introduction to auto-tools - timeline

- Write `configure.ac` `Makefile.am`
- `$ aclocal # Basic m4-macros`
- `$ autoheader # config.h.in`
- `$ automake # Makefile.in`
- `$ autoconf # configure`
- Distribute
- `$./configure # Makefile, config.h`
- `$ make # Builds what's necessary`
- `$ make install # Builds if necessary and installs`



Introduction to auto-tools - timeline

- Write `configure.ac` `Makefile.am`
- `$ aclocal # Basic m4-macros`
- `$ autoheader # config.h.in`
- `$ automake # Makefile.in`
- `$ autoconf # configure`
- Distribute
- `$./configure # Makefile, config.h`
- `$ make # Builds what's necessary`
- `$ make install # Builds if necessary and installs`



Introduction to auto-tools - timeline

- Write `configure.ac` `Makefile.am`
- `$ aclocal # Basic m4-macros`
- `$ autoheader # config.h.in`
- `$ automake # Makefile.in`
- `$ autoconf # configure`
- Distribute
- `$./configure # Makefile, config.h`
- `$ make # Builds what's necessary`
- `$ make install # Builds if necessary and installs`



Introduction to auto-tools - timeline

- Write `configure.ac` `Makefile.am`
- `$ aclocal # Basic m4-macros`
- `$ autoheader # config.h.in`
- `$ automake # Makefile.in`
- `$ autoconf # configure`
- **Distribute**
- `$./configure # Makefile, config.h`
- `$ make # Builds what's necessary`
- `$ make install # Builds if necessary and installs`



Introduction to auto-tools - timeline

- Write `configure.ac` `Makefile.am`
- `$ aclocal # Basic m4-macros`
- `$ autoheader # config.h.in`
- `$ automake # Makefile.in`
- `$ autoconf # configure`
- Distribute
- `$./configure # Makefile, config.h`
- `$ make # Builds what's necessary`
- `$ make install # Builds if necessary and installs`



Introduction to auto-tools - timeline

- Write `configure.ac` `Makefile.am`
- `$ aclocal # Basic m4-macros`
- `$ autoheader # config.h.in`
- `$ automake # Makefile.in`
- `$ autoconf # configure`
- Distribute
- `$./configure # Makefile, config.h`
- `$ make # Builds what's necessary`
- `$ make install # Builds if necessary and installs`



Introduction to auto-tools - timeline

- Write `configure.ac` `Makefile.am`
- `$ aclocal # Basic m4-macros`
- `$ autoheader # config.h.in`
- `$ automake # Makefile.in`
- `$ autoconf # configure`
- Distribute
- `$./configure # Makefile, config.h`
- `$ make # Builds what's necessary`
- `$ make install # Builds if necessary and installs`



Auto-tool related files

- `/Makefile.am` `/*/Makefile.am` maps source files to binaries.
- `/configure.ac` - Dependency checks and meta-data.
- `/autogen.sh` - Contains auto-tools commands

Generated files: `*in`, `Makefile`, `configure`, `config.h`



Auto-tool strictness

- “-foreign”: Basic sanity
- “-gnu” (default): Adds GNU-compliance: NEWS, README, AUTHORS, ChangeLog and COPYING required.
- “-gnits”: GNU nitpicking: THANKS required, -help and -version sanity checks.

“automake -add-missing” grabs generic COPYING and INSTALL files.



Auto-tool strictness

- “-foreign”: Basic sanity
- “-gnu” (default): Adds GNU-compliance: NEWS, README, AUTHORS, ChangeLog and COPYING required.
- “-gnits”: GNU nitpicking: THANKS required, -help and -version sanity checks.

“automake -add-missing” grabs generic COPYING and INSTALL files.

Recommendation

Use -foreign for most non-GNU projects.



A simple autogen.sh

Example

```
#!/bin/sh -x
aclocal
autoheader
automake --add-missing --copy --foreign
autoconf
```

Example

```
./autogen.sh && ./configure && make
```



A simple autogen.sh

Example

```
#!/bin/sh -x
aclocal
autoheader
automake --add-missing --copy --foreign
autoconf
```

Example

```
./autogen.sh && ./configure && make
```

Cargo-cult it! (Like everyone else)



Initial setup - existing project

Step 1: Copy or write a autogen.sh

Step 2: Run autoscan and move configure.scan to configure.ac

Step 4: Make a basic Makefile.am

Step 5: Edit configure.ac

Step 6: Test it



Initial setup - existing project

Step 1: Copy or write a autogen.sh

Step 2: Run autoscan and move configure.scan to configure.ac

Step 4: Make a basic Makefile.am

Step 5: Edit configure.ac

Step 6: Test it

Alternative

Spend 7 hours creating your own configure.ac instead of using autoscan



Initial setup - existing project

Step 1: Copy or write a autogen.sh

Step 2: Run autoscan and move configure.scan to configure.ac

Step 4: Make a basic Makefile.am

Step 5: Edit configure.ac

Step 6: Test it

Alternative

Spend 7 hours creating your own configure.ac instead of using autoscan

Recommendation

*Run ./autogen.sh whenever you modify part of the build system.
In theory, the Makefile will update itself. It mostly works.*



Two simple Makefile.am files

Top level:

Example

```
SUBDIRS = src
```

src/Makefile.am

Example

```
AM_CFLAGS = -Wall  
bin_PROGRAMS = prads  
prads_SOURCES = servicefp/servicefp.c servicefp/mac.c servicefp/...  
prads_LDADD = -lpcrc -lpcap
```



configure.ac

Example

```
AC_PREREQ([2.63])
#AC_INIT([FULL-PACKAGE-NAME], [VERSION], [BUG-REPORT-ADDRESS])
AC_INIT([prads], [0.1-foo], [bugs@example.com])
AC_CONFIG_SRCDIR([src/prads.c])
AC_CONFIG_HEADERS([config.h])

AM_INIT_AUTOMAKE(prads, 0.1-foo, [-Wall -Werror foreign])
```




configure.ac continued

Example

```
AC_PROG_CC
AC_PROG_INSTALL
# FIXME: Replace 'main' with a function in '-lpcap':
AC_CHECK_LIB([pcap], [main])
# FIXME: Replace 'main' with a function in '-lpcrc':
AC_CHECK_LIB([pcrc], [main])
```



configure.ac continued

Example

```
AC_CHECK_HEADERS([fcntl.h limits.h netinet/in.h stddef.h stdli
(...))
AC_TYPE_UINT8_T
AC_CHECK_TYPES([ptrdiff_t])

# Checks for library functions.
AC_FUNC_FORK
AC_FUNC_MALLOC
AC_CHECK_FUNCS([alarm dup2 endgrent endpwent ftruncate getcwd
```



configure.ac continued

Example

```
AC_CONFIG_FILES([Makefile
                 src/Makefile])

AC_OUTPUT
```



Demonstration: Basic Makefile

Tabs and spaces are NOT the same! (Hooray!)



Demonstration: Autotoolizing prads



Demonstration: Advanced projects

- Compiz
- Varnish



Writing portable code

Different systems may have different features, different variations of the same feature, different location for header files, and so on.

Adding a check for what you use in `configure.ac` and then evaluating the variable from `config.h` is the key.



Writing portable code

Different systems may have different features, different variations of the same feature, different location for header files, and so on.

Adding a check for what you use in `configure.ac` and then evaluating the variable from `config.h` is the key.

Recommendation

Read `config.h`



config.h

Example

```
/* Define to 1 if you have the 'epoll_ctl' function. */  
#define HAVE_EPOLL_CTL 1
```

Example

```
#include "config.h"  
(...)  
#if defined(HAVE_EPOLL_CTL)  
    sm->sess.ev.data.ptr = NULL;  
#endif
```



Finding the macro for autoconf

Checking for a custom program, configure.ac:

Example

```
AC_CHECK_PROGS(XSLTPROC, [xsltproc], "no")
if test "x$XSLTPROC" = "xno"; then
AC_MSG_WARN([xsltproc not found - not building documentation])
fi
AM_CONDITIONAL(HAVE_XSLTPROC,[test "x$XSLTPROC" != "xno"])
```



Finding the macro for autoconf - continued

doc/Makefile.am

Example

```
if HAVE_XSLTPROC
all: ${CHANGELOGS}
endif
(...)
.xml.html:
if HAVE_XSLTPROC
    ${XSLTPROC} --xinclude -o $@ $<
else
    @echo "You need xsltproc installed to make dist"
endif
```



Questions from the audience?

Any questions?



Thank you

Thank you for listening