

Autotools: An Overview

Danny Robson

2014-09-24

Autotools: An Overview

Danny Robson

Overview

Autotools

autoconf

automake

pkg-config

Final Words

Autotools: An Overview

Danny Robson

2014-09-24

└ About Me

- Performance systems programming
 - C, C++
 - Linux
 - GNU
- autotools
 - Self-taught. Slowly.
 - caveat emptor
- Ask lots of questions



About Me

Autotools: An Overview

Danny Robson

Overview

Autotools

autoconf

automake

pkg-config

Final Words

■ Performance systems programming

- C, C++
- Linux
- GNU
- autotools
 - Self-taught. Slowly.
 - caveat emptor
- Ask lots of questions



└ About Me

- Performance systems programming
 - C, C++
 - Linux
 - GNU
- autotools
 - Self-taught. Slowly.
 - caveat emptor
- Ask lots of questions



About Me

Autotools: An Overview

Danny Robson

Overview

Autotools

autoconf

automake

pkg-config

Final Words

- Performance systems programming
 - C, C++
 - Linux
 - GNU
- autotools
 - Self-taught. Slowly.
 - caveat emptor
- Ask lots of questions



└ About Me

- Performance systems programming
 - C, C++
 - Linux
 - GNU
- autotools
 - Self-taught. Slowly.
 - caveat emptor
- Ask lots of questions



Autotools: An Overview

Danny Robson

Overview

Autotools

autoconf

automake

pkg-config

Final Words

About Me

- Performance systems programming
 - C, C++
 - Linux
 - GNU
- autotools
 - Self-taught. Slowly.
 - caveat emptor
- Ask lots of questions



└─What You're In For

- 1 Overview
- 2 Autotools
- 3 autoconf
- 4 automake
- 5 pkg-config
- 6 Final Words

What You're In For

Autotools: An
Overview

Danny Robson

Overview

Autotools

autoconf

automake

pkg-config

Final Words

1 Overview

2 Autotools

3 autoconf

4 automake

5 pkg-config

6 Final Words

Section 1

Overview

Autotools: An
Overview

Danny Robson

Overview

Autotools

autoconf

automake

pkg-config

Final Words

Section 1

Overview

- Automate for consistency, tedium:
 - compilation
 - packaging
 - testing

Build automation

Autotools: An Overview

Danny Robson

Overview

Autotools

autoconf

automake

pkg-config

Final Words

- Automate for consistency, tedium:
 - compilation
 - packaging
 - testing

- manual dependencies
- replicate existing functionality

- **make**
- bjam, SCons, Waf
- autotools
- CMake, gyp, premake
- Many IDEs: Visual Studio, Eclipse, Netbeans, ...

Prominent Systems

Autotools: An
Overview

Danny Robson

Overview

Autotools

autoconf

automake

pkg-config

Final Words

■ **make**

■ bjam, SCons, Waf

■ autotools

■ CMake, gyp, premake

■ Many IDEs: Visual Studio, Eclipse, Netbeans, ...

- build your own systems
- less well known and supported
- many similar flaws
 - speed
 - complexity
 - systems supported

- make
- **bjam, SCons, Waf**
- autotools
- CMake, gyp, premake
- Many IDEs: Visual Studio, Eclipse, Netbeans, ...

Prominent Systems

Autotools: An
Overview

Danny Robson

Overview

Autotools

autoconf

automake

pkg-config

Final Words

- make
- **bjam, SCons, Waf**
- autotools
- CMake, gyp, premake
- Many IDEs: Visual Studio, Eclipse, Netbeans, ...

- make
- bjam, SCons, Waf
- **autotools**
- CMake, gyp, premake
- Many IDEs: Visual Studio, Eclipse, Netbeans, ...

Prominent Systems

Autotools: An
Overview

Danny Robson

Overview

Autotools

autoconf

automake

pkg-config

Final Words

- make
- bjam, SCons, Waf
- **autotools**
- CMake, gyp, premake
- Many IDEs: Visual Studio, Eclipse, Netbeans, ...

- Multiple target backends
- Newer languages
- Faster backends: ninja
- IDE backends, useful for Windows

- make
- bjam, SCons, Waf
- autotools
- **CMake, gyp, premake**
- Many IDEs: Visual Studio, Eclipse, Netbeans, ...

Prominent Systems

Autotools: An
Overview

Danny Robson

Overview

Autotools

autoconf

automake

pkg-config

Final Words

- make
- bjam, SCons, Waf
- autotools
- **CMake, gyp, premake**
- Many IDEs: Visual Studio, Eclipse, Netbeans, ...

- make
- bjam, SCons, Waf
- autotools
- CMake, gyp, premake
- Many IDEs: Visual Studio, Eclipse, Netbeans, ...

Prominent Systems

Autotools: An
Overview

Danny Robson

Overview

Autotools

autoconf

automake

pkg-config

Final Words

- make
- bjam, SCons, Waf
- autotools
- CMake, gyp, premake
- Many IDEs: Visual Studio, Eclipse, Netbeans, ...

- More platforms than you have access to
- Will just work everywhere, solved problems already

- Features for downstream
 - VPATH, DESTDIR, standard targets, config.site, cross compilation, ...
- Portability
- Just works: `configure && make && make install`

Why autotools

Autotools: An Overview

Danny Robson

Overview

Autotools

autoconf

automake

pkg-config

Final Words

- Features for downstream
 - VPATH, DESTDIR, standard targets, config.site, cross compilation, ...
- Portability
- Just works: `configure && make && make install`

- More platforms than you have access to
- Will just work everywhere, solved problems already

- Features for downstream
 - VPATH, DESTDIR, standard targets, config.site, cross compilation, ...
- Portability
 - Just works: `configure && make && make install`

Why autotools

Autotools: An Overview

Danny Robson

Overview

Autotools

autoconf

automake

pkg-config

Final Words

- Features for downstream
 - VPATH, DESTDIR, standard targets, config.site, cross compilation, ...
- Portability
 - Just works: `configure && make && make install`

- More platforms than you have access to
- Will just work everywhere, solved problems already

- Features for downstream
 - VPATH, DESTDIR, standard targets, config.site, cross compilation, ...
- Portability
- Just works: `configure && make && make install`

Why autotools

Autotools: An Overview

Danny Robson

Overview

Autotools

autoconf

automake

pkg-config

Final Words

- Features for downstream
 - VPATH, DESTDIR, standard targets, config.site, cross compilation, ...
- Portability
- Just works: `configure && make && make install`

- Windows support is possible, but painful.
- IDE integration is error prone, manual syncing is common
- 70s tools
- 90s framework
- What Year Is It

- Targets UNIX systems
- Error prone in IDE heavy development
- Archaic toolsets

Why not autotools

Autotools: An Overview

Danny Robson

Overview

Autotools

autoconf

automake

pkg-config

Final Words

- Targets UNIX systems
- Error prone in IDE heavy development
- Archaic toolsets

- Windows support is possible, but painful.
- IDE integration is error prone, manual syncing is common
- 70s tools
- 90s framework
- What Year Is It

- Targets UNIX systems
- Error prone in IDE heavy development
- Archaic toolsets

Why not autotools

Autotools: An Overview

Danny Robson

Overview

Autotools

autoconf

automake

pkg-config

Final Words

- Targets UNIX systems
- Error prone in IDE heavy development
- Archaic toolsets

- Windows support is possible, but painful.
- IDE integration is error prone, manual syncing is common
- 70s tools
- 90s framework
- What Year Is It

- Targets UNIX systems
- Error prone in IDE heavy development
- Archaic toolsets

Why not autotools

Autotools: An Overview

Danny Robson

Overview

Autotools

autoconf

automake

pkg-config

Final Words

- Targets UNIX systems
- Error prone in IDE heavy development
- Archaic toolsets



Autotools: An Overview

Danny Robson

Overview

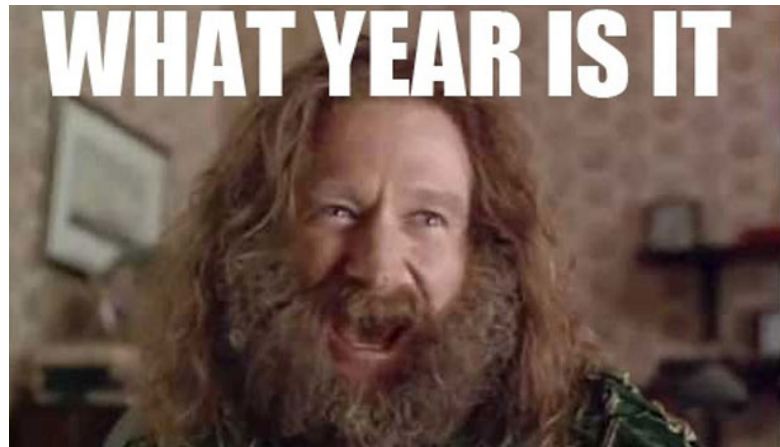
Autotools

autoconf

automake

pkg-config

Final Words



Section 2

Autotools

- UNIX focused
- Extreme portability
- Probe, don't ask
- Convention

- Probe avoids out of date tables, allows hacked systems
- GNITS, foreign. README, NEWS, COPYING, ... targets.

Philosophy

Autotools: An
Overview

Danny Robson

Overview

Autotools

autoconf

automake

pkg-config

Final Words

- UNIX focused
- Extreme portability
- Probe, don't ask
- Convention

- Java via gcj
- Scripts more suited to own tools
- Erlang...

Primarily geared towards systems languages.

- C, C++, Objective-C, Fortran
- lex, yacc
- Erlang
- Lisp, Python, Java

Language Support

Autotools: An
Overview

Danny Robson

Overview

Autotools

autoconf

automake

pkg-config

Final Words

Primarily geared towards systems languages.

- C, C++, Objective-C, Fortran
- lex, yacc
- Erlang
- Lisp, Python, Java

- Java via gcj
- Scripts more suited to own tools
- Erlang...

Primarily geared towards systems languages.

- C, C++, Objective-C, Fortran
- lex, yacc
- Erlang
- Lisp, Python, Java

Language Support

Autotools: An Overview

Danny Robson

Overview

Autotools

autoconf

automake

pkg-config

Final Words

Primarily geared towards systems languages.

- C, C++, Objective-C, Fortran
- lex, yacc
- Erlang
- Lisp, Python, Java

- Java via gcj
- Scripts more suited to own tools
- Erlang...

Primarily geared towards systems languages.

- C, C++, Objective-C, Fortran
- lex, yacc
- Erlang
- Lisp, Python, Java

Language Support

Autotools: An Overview

Danny Robson

Overview

Autotools

autoconf

automake

pkg-config

Final Words

Primarily geared towards systems languages.

- C, C++, Objective-C, Fortran
- lex, yacc
- Erlang
- Lisp, Python, Java

- Java via gcj
- Scripts more suited to own tools
- Erlang...

Primarily geared towards systems languages.

- C, C++, Objective-C, Fortran
- lex, yacc
- Erlang
- Lisp, Python, Java

Language Support

Autotools: An
Overview

Danny Robson

Overview

Autotools

autoconf

automake

pkg-config

Final Words

Primarily geared towards systems languages.

- C, C++, Objective-C, Fortran
- lex, yacc
- Erlang
- Lisp, Python, Java

Possible to avoid shell scripting, but useful

Useful, but not necessary, to know a little of:

- Makefile
- sh
- m4
- Google

Languages Needed

Autotools: An
Overview

Danny Robson

Overview

Autotools

autoconf

automake

pkg-config

Final Words

Useful, but not necessary, to know a little of:

- Makefile
- sh
- m4
- Google

Useful, but not necessary, to know a little of:

- Makefile
- sh
- m4
- Google

Possible to avoid shell scripting, but useful

Languages Needed

Autotools: An
Overview

Danny Robson

Overview

Autotools

autoconf

automake

pkg-config

Final Words

Useful, but not necessary, to know a little of:

- Makefile
- sh
- m4
- Google

Useful, but not necessary, to know a little of:

- Makefile
- sh
- m4
- Google

Languages Needed

Possible to avoid shell scripting, but useful

Autotools: An
Overview

Danny Robson

Overview

Autotools

autoconf

automake

pkg-config

Final Words

Useful, but not necessary, to know a little of:

- Makefile
- sh
- m4
- Google

```
Makefile
hello.o: hello.c
        gcc -c hello.c -o hello.o

hello: hello.o
        gcc hello.o -o hello
```

naive

1. naive
2. dependency
3. generalised rules
4. parameterised compiler
5. parameterised flags.
6. what about extensions, install, test, clean, output/input paths, clang/icc/solaris, ...

A Simple Makefile

Autotools: An
Overview

Danny Robson

Overview

Autotools

autoconf

automake

pkg-config

Final Words

Makefile

```
hello.o: hello.c
        gcc -c hello.c -o hello.o

hello: hello.o
        gcc hello.o -o hello
```

```
Makefile
hello.o: hello.c hello.h
        gcc -c hello.c -o hello.o

hello: hello.o
        gcc hello.o -o hello
```

1. naive
2. dependency
3. generalised rules
4. parameterised compiler
5. parameterised flags.
6. what about extensions, install, test, clean, output/input paths, clang/icc/solaris, ...

A Simple Makefile

Autotools: An
Overview

Danny Robson

Overview

Autotools

autoconf

automake

pkg-config

Final Words

Makefile

```
hello.o: hello.c hello.h
        gcc -c hello.c -o hello.o

hello: hello.o
        gcc hello.o -o hello
```

1. naive
2. dependency
3. generalised rules
4. parameterised compiler
5. parameterised flags.
6. what about extensions, install, test, clean, output/input paths, clang/icc/solaris, ...

```
Makefile
```

```
hello.o: hello.c hello.h
```

```
gcc -c $< -o $@
```

```
hello: hello.o
```

```
gcc $^ -o $@
```

A Simple Makefile

Autotools: An
Overview

Danny Robson

Overview

Autotools

autoconf

automake

pkg-config

Final Words

Makefile

```
hello.o: hello.c hello.h
        gcc -c $< -o $@
```

```
hello: hello.o
        gcc $^ -o $@
```

```
Makefile
hello.o: hello.c hello.h
$(CC) -c $< -o $@

hello: hello.o
$(CC) $^ -o $@
```

1. naive
2. dependency
3. generalised rules
4. parameterised compiler
5. parameterised flags.
6. what about extensions, install, test, clean, output/input paths, clang/icc/solaris, ...

A Simple Makefile

Autotools: An
Overview

Danny Robson

Overview

Autotools

autoconf

automake

pkg-config

Final Words

Makefile

```
hello.o: hello.c hello.h
$(CC) -c $< -o $@

hello: hello.o
$(CC) $^ -o $@
```



```
Makefile
hello.o: hello.c hello.h
$(CC) $(CFLAGS) -c $< -o $@

hello: hello.o
$(CC) $(LDFLAGS) $^ -o $@
```

1. naive
2. dependency
3. generalised rules
4. parameterised compiler
5. parameterised flags.
6. what about extensions, install, test, clean, output/input paths, clang/icc/solaris, ...

A Simple Makefile

Autotools: An
Overview

Danny Robson

Overview

Autotools

autoconf

automake

pkg-config

Final Words

Makefile

```
hello.o: hello.c hello.h
$(CC) $(CFLAGS) -c $< -o $@

hello: hello.o
$(CC) $(LDFLAGS) $^ -o $@
```

1. naive
2. dependency
3. generalised rules
4. parameterised compiler
5. parameterised flags.
6. what about extensions, install, test, clean, output/input paths, clang/icc/solaris, ...

```
Makefile
hello.o: hello.c hello.h
$(CC) $(CFLAGS) -c $< -o $@

hello: hello.o
$(CC) $(LDFLAGS) $^ -o $@
```

A Naive Makefile

Autotools: An
Overview

Danny Robson

Overview

Autotools

autoconf

automake

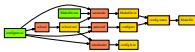
pkg-config

Final Words

Makefile

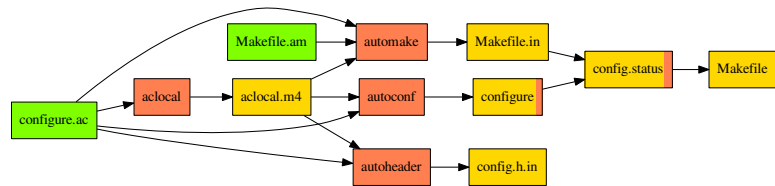
```
hello.o: hello.c hello.h
$(CC) $(CFLAGS) -c $< -o $@

hello: hello.o
$(CC) $(LDFLAGS) $^ -o $@
```



1. User provides `configure.ac`, `Makefile.am`
2. `aclocal` collates required `m4` macros
3. `autoconf` generates `configure` script
4. `autoheader` generates template `config.h`
5. `automake` generates template Makefiles
6. `configure` executes macros and writes `config.status`
7. `config.status` fills in the various templates

How it fits together



1. autoreconf
2. configure
3. Makefile



Flow of execution

Autotools: An Overview

Danny Robson

Overview

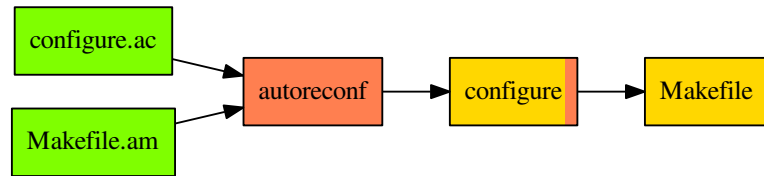
Autotools

autoconf

automake

pkg-config

Final Words





Initialisation

Autotools: An
Overview

Danny Robson

Overview

Autotools

autoconf

automake

pkg-config

Final Words

Files

- `configure.ac`
 - `autoscan`
 - `autoprotect`
- `Makefile.am`
- `m4/`

Commands

```
autoreconf -i
```

Section 3

autoconf

Autotools: An
Overview

Danny Robson

Overview

Autotools

autoconf

automake

pkg-config

Final Words

Section 3

autoconf

Substitution values for config.status Variables for automake Quoting is a common pitfall

- Discovers system characteristics
 - autoconf-archive
 - pkg-config
 - gnulib
- Compute substitution values, variables
- Curious mixture of M4 and sh
 - Vanilla shell more-or-less supported
 - Easy to start, hard to master

- Discovers system characteristics
 - autoconf-archive
 - pkg-config
 - gnulib
- Compute substitution values, variables
- Curious mixture of M4 and sh
 - Vanilla shell more-or-less supported
 - Easy to start, hard to master

- Discovers system characteristics
 - autoconf-archive
 - pkg-config
 - gnuilib
- Compute substitution values, variables
- Curious mixture of M4 and sh

Substitution values for config.status Variables for automake Quoting is a common pitfall

autoconf

Autotools: An Overview

Danny Robson

Overview

Autotools

autoconf

automake

pkg-config

Final Words

- Discovers system characteristics
 - autoconf-archive
 - pkg-config
 - gnuilib
- Compute substitution values, variables
- Curious mixture of M4 and sh
 - Vanilla shell more-or-less supported
 - Easy to start, hard to master

- Discovers system characteristics
 - autoconf-archive
 - pkg-config
 - gnulib
- Compute substitution values, variables
- Curious mixture of M4 and sh

Substitution values for config.status Variables for automake Quoting is a common pitfall

autoconf

Autotools: An Overview

Danny Robson

Overview

Autotools

autoconf

automake

pkg-config

Final Words

- Discovers system characteristics
 - autoconf-archive
 - pkg-config
 - gnulib
- Compute substitution values, variables
- Curious mixture of M4 and sh
 - Vanilla shell more-or-less supported
 - Easy to start, hard to master

- Discovers system characteristics
 - autoconf-archive
 - pkg-config
 - gnuilib
- Compute substitution values, variables
- Curious mixture of M4 and sh

Substitution values for config.status Variables for automake Quoting is a common pitfall

autoconf

Autotools: An Overview

Danny Robson

Overview

Autotools

autoconf

automake

pkg-config

Final Words

- Discovers system characteristics
 - autoconf-archive
 - pkg-config
 - gnuilib
- Compute substitution values, variables
- Curious mixture of M4 and sh
 - Vanilla shell more-or-less supported
 - Easy to start, hard to master

- Discovers system characteristics
 - autoconf-archive
 - pkg-config
 - gnuilib
- Compute substitution values, variables
- Curious mixture of M4 and sh

Substitution values for config.status Variables for automake Quoting is a common pitfall

autoconf

Autotools: An Overview

Danny Robson

Overview

Autotools

autoconf

automake

pkg-config

Final Words

- Discovers system characteristics
 - autoconf-archive
 - pkg-config
 - gnuilib
- Compute substitution values, variables
- Curious mixture of M4 and sh
 - Vanilla shell more-or-less supported
 - Easy to start, hard to master

- Discovers system characteristics
 - autoconf-archive
 - pkg-config
 - gnuilib
- Compute substitution values, variables
- Curious mixture of M4 and sh
 - Vanilla shell more-or-less supported
 - Easy to start, hard to master

Substitution values for config.status Variables for automake Quoting is a common pitfall

autoconf

Autotools: An Overview

Danny Robson

Overview

Autotools

autoconf

automake

pkg-config

Final Words

- Discovers system characteristics
 - autoconf-archive
 - pkg-config
 - gnuilib
- Compute substitution values, variables
- Curious mixture of M4 and sh
 - Vanilla shell more-or-less supported
 - Easy to start, hard to master

- Discovers system characteristics
 - autoconf-archive
 - pkg-config
 - gnuilib
- Compute substitution values, variables
- Curious mixture of M4 and sh
 - Vanilla shell more-or-less supported
 - Easy to start, hard to master

Substitution values for config.status Variables for automake Quoting is a common pitfall

- Discovers system characteristics
 - autoconf-archive
 - pkg-config
 - gnuilib
- Compute substitution values, variables
- Curious mixture of M4 and sh
 - Vanilla shell more-or-less supported
 - Easy to start, hard to master

- Discovers system characteristics
 - autoconf-archive
 - pkg-config
 - gnuilib
- Compute substitution values, variables
- Curious mixture of M4 and sh
 - Vanilla shell more-or-less supported
 - Easy to start, hard to master

Substitution values for config.status Variables for automake Quoting is a common pitfall

- Discovers system characteristics
 - autoconf-archive
 - pkg-config
 - gnuilib
- Compute substitution values, variables
- Curious mixture of M4 and sh
 - Vanilla shell more-or-less supported
 - Easy to start, hard to master

```
configure.ac
AC_INIT([foo], [1.2.3], [example@example.com])
AC_PROG_CC
AC_CONFIG_FILES([Makefile])
AC_OUTPUT
```

One of two important input files. And M4sh script calling autoconf macros squarebracket is quote

- AC_INIT first macro, initialises everything
- AC_PROG_FOO locate prog and init vars
- AC_CONFIG_FILES paths we want to parameterise. Anything ending in .in
- AC_OUTPUT interpolate the files

A simple autoconf input

configure.ac

```
AC_INIT([foo], [1.2.3], [example@example.com])
AC_PROG_CC
AC_CONFIG_FILES([Makefile])
AC_OUTPUT
```

```
Makefile.in
hello.o: hello.c hello.h
    @CC@ @CFLAGS@ -c $< -o $@

hello: hello.o
    @CC@ @LDFLAGS@ $^ -o $@
```

Example: Makefile

Autotools: An
Overview

Danny Robson

Overview

Autotools

autoconf

automake

pkg-config

Final Words

Makefile.in

```
hello.o: hello.c hello.h
    @CC@ @CFLAGS@ -c $< -o $@

hello: hello.o
    @CC@ @LDFLAGS@ $^ -o $@
```



```
Doxyfile.in
PROJECT_NAME = @PACKAGE_NAME@
PROJECT_NUMBER = @PACKAGE_VERSION@
OUTPUT_DIRECTORY = @top_builddir@/doc
```

Example: Template Doxyfile

Autotools: An
Overview

Danny Robson

Overview

Autotools

autoconf

automake

pkg-config

Final Words

Doxyfile.in

```
PROJECT_NAME = @PACKAGE_NAME@
PROJECT_NUMBER = @PACKAGE_VERSION@
OUTPUT_DIRECTORY = @top_builddir@/doc
```

```
configure.ac
AC_SEARCH_LIBS(
  [glBegin],
  [OpenGL opengl32 GL],
  [],
  [AC_MSG_ERROR(["Can't find OpenGL"])]
)
```

Example: Library discovery

Autotools: An
Overview

Danny Robson

Overview

Autotools

autoconf

automake

pkg-config

Final Words

configure.ac

```
AC_SEARCH_LIBS(
  [glBegin],
  [OpenGL opengl32 GL],
  [],
  [AC_MSG_ERROR(["Can't find OpenGL"])]
)
```

```
configure.ac
AC_ARG_ENABLE(
  [debugging],
  [AS_HELP_STRING(
    [--enable-debugging],
    [enables developer debugging support]
  )
]
)
AS_IF([test "x${enable_debugging}" = "xyes"], [
  AX_APPEND_COMPILE_FLAGS([-O0])
], [
  AX_APPEND_COMPILE_FLAGS([-O2])
])
```

Example: conditional enable

configure.ac

```
AC_ARG_ENABLE(
  [debugging],
  [AS_HELP_STRING(
    [--enable-debugging],
    [enables developer debugging support]
  )
]
)

AS_IF([test "x${enable_debugging}" = "xyes"], [
  AX_APPEND_COMPILE_FLAGS([-O0])
], [
  AX_APPEND_COMPILE_FLAGS([-O2])
])
```

Section 4

automake

Autotools: An
Overview

Danny Robson

Overview

Autotools

autoconf

automake

pkg-config

Final Words

Section 4

automake

└─ automake

└─ automake

- Generates Makefile.in from Makefile.am
- Compiler derived dependency tracking
- make-syntax with magic variables

automake

Autotools: An
Overview

Danny Robson

Overview

Autotools

autoconf

automake

pkg-config

Final Words

- Fine to use make syntax everywhere, own variables, own rules
- Must avoid clashes with some key variables
- Dependency tracking uses the compiler. Various iterations before, this is most robust.

- Generates Makefile.in from Makefile.am
- Compiler derived dependency tracking
- make-syntax with magic variables

2014-09-24

Autotools: An Overview

└─ automake

└─ automake



automake

Autotools: An Overview

Danny Robson

Overview

Autotools

autoconf

automake

pkg-config

Final Words



Unsure if target/source is correct nomenclature

```
Makefile.am
bin_PROGRAMS = hello

hello_SOURCES = hello.cpp hello.hpp
hello_LDADD   = $(BOOST_LDFLAGS)
hello_CXXFLAGS = $(BOOST_CPPFLAGS)

Atoms
target where_PRIMARY
source target_WHAT
```

A simple automake input

Makefile.am

```
bin_PROGRAMS = hello

hello_SOURCES = hello.cpp hello.hpp
hello_LDADD   = $(BOOST_LDFLAGS)
hello_CXXFLAGS = $(BOOST_CPPFLAGS)
```

Atoms

```
target where_PRIMARY
source target_WHAT
```

- custom locations via foodir

Atom
bin_PROGRAMS where_PRIMARY
where
location bin, lib, include, share, man, info, ... noinst Not installed. Used during build only. check Not installed. Only created for tests.

Targets

Autotools: An
Overview

Danny Robson

Overview

Autotools

autoconf

automake

pkg-config

Final Words

Atom

```
bin_PROGRAMS
where_PRIMARY
```

where

```
location bin, lib, include, share, man, info, ...
noinst Not installed. Used during build only.
check Not installed. Only created for tests.
```


Atom	
	bin_PROGRAMS where_PRIMARY
PRIMARY	
PROGRAMS	Executable binaries
LIBRARIES	Libraries
HEADERS	Installable headers
SCRIPTS	Installable scripts
DATA	Architecture independent data

Targets

Atom

```
bin_PROGRAMS
where_PRIMARY
```

PRIMARY

```
PROGRAMS Executable binaries
LIBRARIES Libraries
HEADERS Installable headers
SCRIPTS Installable scripts
DATA Architecture independant data
```

Atom
hello_SOURCES target_WHAT
WHAT
SOURCES All source files; headers too.
CFLAGS Compiler flags
LDFLAGS Linker flags
LIBADD Libraries; extra libs
LDADD Programs; extra libs

Source

Autotools: An Overview

Danny Robson

Overview

Autotools

autoconf

automake

pkg-config

Final Words

Atom

```
hello_SOURCES  
target_WHAT
```

WHAT

SOURCES All source files; headers too.

CFLAGS Compiler flags

LDFLAGS Linker flags

LIBADD Libraries; extra libs

LDADD Programs; extra libs

```

Makefile.am
SUFFIXES = .cpp .cpp.rl
.cpp.rl.cpp:
$(RAGEL) $(RAGELFLAGS) -C $< -o $(builddir)/$@

bin_PROGRAMS = hello

hello_SOURCES = hello.cpp hello.hpp parse.cpp
hello_LDADD = $(BOOST_LDFLAGS)
hello_CXXFLAGS = $(BOOST_CPPFLAGS)

```

Ignoring EXTRA_DIST and CLEANFILES

Example: Vanilla Make

Autotools: An
Overview

Danny Robson

Overview

Autotools

autoconf

automake

pkg-config

Final Words

Makefile.am

```
SUFFIXES = .cpp .cpp.rl
```

```
.cpp.rl.cpp:
```

```
$(RAGEL) $(RAGELFLAGS) -C $< -o $(builddir)/$@
```

```
bin_PROGRAMS = hello
```

```
hello_SOURCES = hello.cpp hello.hpp parse.cpp
```

```
hello_LDADD = $(BOOST_LDFLAGS)
```

```
hello_CXXFLAGS = $(BOOST_CPPFLAGS)
```

AC_HELP_STRING

```
configure.ac
AC_ENABLE([network], [--enable-network])
AM_CONDITIONAL(
  [ENABLE_NETWORK],
  [test "x${enable_network}" = "xyes"])
```

```
Makefile.am
if ENABLE_NETWORK
  foo_SOURCES += network.c
  foo_LDADD += $(NET_LIBS)
endif
```

Example: Conditionals

Autotools: An
Overview

Danny Robson

Overview

Autotools

autoconf

automake

pkg-config

Final Words

configure.ac

```
AC_ENABLE([network], [--enable-network])
AM_CONDITIONAL(
  [ENABLE_NETWORK],
  [test "x${enable_network}" = "xyes"])
```

Makefile.am

```
if ENABLE_NETWORK
  foo_SOURCES += network.c
  foo_LDADD += $(NET_LIBS)
endif
```

Section 5

pkg-config

- A better query for installed libraries
 - Compiler parameters
 - Linker parameters
 - Versioning
- Command line and autotools wrappers

Autotools: An Overview

Danny Robson

Overview

Autotools

autoconf

automake

pkg-config

Final Words

- A better query for installed libraries
 - Compiler parameters
 - Linker parameters
 - Versioning
- Command line and autotools wrappers

```
libfoo.pc
prefix=@prefix@
exec_prefix=@exec_prefix@
libdir=@libdir@
includedir=@includedir@

Name: libfoo
Description: A simple test library
URL: http://example.com/
Version: @VERSION@
Cflags: -I${includedir}
Libs: -L${libdir} -lfoo
```

pkg-config: producers

Autotools: An
Overview

Danny Robson

Overview

Autotools

autoconf

automake

pkg-config

Final Words

libfoo.pc

```
prefix=@prefix@
exec_prefix=@exec_prefix@
libdir=@libdir@
includedir=@includedir@
```

```
Name: libfoo
Description: A simple test library
URL: http://example.com/
Version: @VERSION@
Cflags: -I${includedir}
Libs: -L${libdir} -lfoo
```

```
configure.ac
PKG_CHECK_MODULES([FOO], [libfoo >= 0.2.0])

Makefile.am
hello_CFLAGS += FOO_CFLAGS
hello_LDADD  += FOO_LIBS
```

pkg-config: consumers

Autotools: An
Overview

Danny Robson

Overview

Autotools

autoconf

automake

pkg-config

Final Words

configure.ac

```
PKG_CHECK_MODULES([FOO], [libfoo >= 0.2.0])
```

Makefile.am

```
hello_CFLAGS += FOO_CFLAGS
hello_LDADD  += FOO_LIBS
```


Section 6

Final Words

Autotools: An
Overview

Danny Robson

Overview

Autotools

autoconf

automake

pkg-config

Final Words

Section 6

Final Words

I'm not going to pretend that autoconf is good [...] but it's much better than the alternatives once you start needing to compile on many different platforms, cross compile, or integrate with Linux distro packaging systems. – Richard WM Jones

Autotools: An Overview

Danny Robson

Overview

Autotools

autoconf

automake

pkg-config

Final Words

I'm not going to pretend that autoconf is good [...] but it's much better than the alternatives once you start needing to compile on many different platforms, cross compile, or integrate with Linux distro packaging systems. – Richard WM Jones

- gnuilib
- autoheader
- autoconf-archive
- gettext
- libtool
- ...

More Features

Autotools: An
Overview

Danny Robson

Overview

Autotools

autoconf

automake

pkg-config

Final Words

- gnuilib
- autoheader
- autoconf-archive
- gettext
- libtool
- ...

- macro and variable indexes

- **Goat-book**
- GNU autoconf manual
- GNU automake manual
- GNU make manual
- man pkg-config
- Autotools Mythbuster
- Alexandre Duret-Lutz's tutorial

Further Reading

Autotools: An Overview

Danny Robson

Overview

Autotools

autoconf

automake

pkg-config

Final Words

■ **Goat-book**

- GNU autoconf manual
- GNU automake manual
- GNU make manual
- man pkg-config
- Autotools Mythbuster
- Alexandre Duret-Lutz's tutorial

- macro and variable indexes

- ~~Goat-book~~
- GNU autoconf manual
- GNU automake manual
- GNU make manual
- ~~man pkg-config~~
- Autotools Mythbuster
- Alexandre Duret-Lutz's tutorial

Further Reading

Autotools: An
Overview

Danny Robson

Overview

Autotools

autoconf

automake

pkg-config

Final Words

- ~~Goat-book~~
- GNU autoconf manual
- GNU automake manual
- GNU make manual
- ~~man pkg-config~~
- Autotools Mythbuster
- Alexandre Duret-Lutz's tutorial

- macro and variable indexes

- ~~Goat-book~~
- GNU autoconf manual
- GNU automake manual
- GNU make manual
- man pkg-config
- Autotools Mythbuster
- Alexandre Duret-Lutz's tutorial

Further Reading

Autotools: An
Overview

Danny Robson

Overview

Autotools

autoconf

automake

pkg-config

Final Words

- ~~Goat-book~~
- GNU autoconf manual
- GNU automake manual
- GNU make manual
- man pkg-config
- Autotools Mythbuster
- Alexandre Duret-Lutz's tutorial

Section 7

Secret Level

- vanilla shell embedded

```
configure.ac
AC_CANONICAL_HOST

case $host_os in
mingw32)
  AC_CHECK_HEADERS([winsock2.h ws2tcpip.h])
  AC_SUBST([NET_LIBS], [-lws2_32])
  AX_APPEND_COMPLE_FLAGS([
    [-DBOOST_THREAD_USE_LIB]
  ])
  ;;
esac
```

Example: host specific configuration

configure.ac

```
AC_CANONICAL_HOST
```

```
case $host_os in
  mingw32)
    AC_CHECK_HEADERS([winsock2.h ws2tcpip.h])
    AC_SUBST([NET_LIBS], [-lws2_32])
    AX_APPEND_COMPLE_FLAGS([
      [-DBOOST_THREAD_USE_LIB]
    ])
  ;;
esac
```



```
Makefile.am
noinst_LIBRARIES = libclcruft.a
libclcruft_SOURCES = cl/base.cpp # etc..

bin_PROGRAMS = waif cl-info

waif_SOURCES = waif.cpp
waif_LDADD = \
  ${top_builddir}/libclcruft.a \
  ${top_builddir}/util/libutil.a

cl_info_SOURCES = tools/cl-info.cpp
cl_info_SOURCES = ${top_builddir}/libclcruft.a
```

Multi-source

Makefile.am

```
noinst_LIBRARIES = libclcruft.a
libclcruft_SOURCES = cl/base.cpp # etc..

bin_PROGRAMS = waif cl-info

waif_SOURCES = waif.cpp
waif_LDADD = \
  ${top_builddir}/libclcruft.a \
  ${top_builddir}/util/libutil.a

cl_info_SOURCES = tools/cl-info.cpp
cl_info_SOURCES = ${top_builddir}/libclcruft.a
```