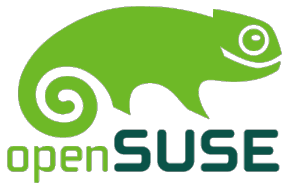


Small introduction to Autotools



Tomáš Chvátal
openSUSE Team

2013/10/06

Introduction

Who the hell is Tomáš Chvátal

- Gentoo developer since fall 2008 and Council member since 2010
- SUSE Employee since 2011 (QA, openSUSE)
- Libreoffice contributor since 2011 (not related with SUSE :P)

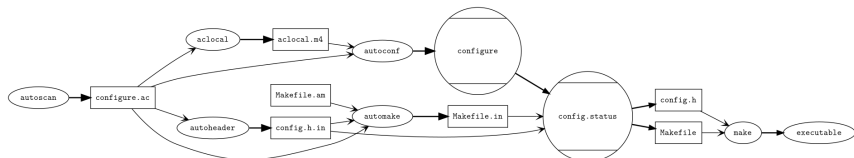
Available build systems

What are our options when building software

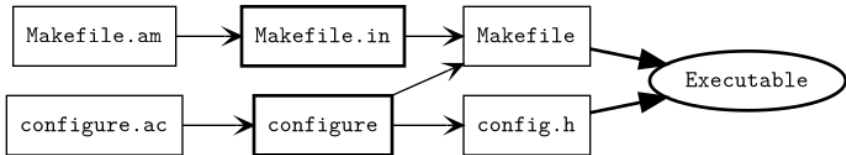
- Autotools, cmake
- GNU Makefile
- Waf, scon, jam
- Own awesome build system (bam, ...)

Autotools process

Complete autotools process



Simplified autotools process



How does deploying look

Developer	Consumer
<pre>\$ cd git-repo/ \$ sh autogen.sh \$./configure \$ make distcheck "upload tarball"</pre>	<pre>\$ wget "uploaded tarball" \$ unpack "uploaded tarball" \$./configure --enable-this --disable-that \$ make -jVALUE \$ make install</pre>

Writting autotools for dummies

Autoconf aka configure.ac

What is autogen.sh

- <http://tinyurl.com/q596zjt>
- The script usually contains some safeguards and then executes following commands:
 1. libtoolize
 2. aclocal
 3. autoheader
 4. automake
 5. autoconf
- Shorter version is just to run "autoreconf -vi"

Defining version

```
# Process this file with autoconf to create configure.
AC_PREREQ([2.65])

AC_INIT([package], [m4_esyscmd([./version.sh])])
AC_CONFIG_MACRO_DIR([m4])
AC_CONFIG_HEADERS([config.h])
AM_INIT_AUTOMAKE([1.11 foreign no-dist-gz dist-xz])
AM_SILENT_RULES([yes])
AC_LANG([C])

PC_VERSION=m4_esyscmd([./version.sh -p])
```

Version.sh script

<https://github.com/lu-zero/nanomsg/blob/autotools/version.sh>

How does it work:

- Version gets detected from .version file or git tag (+sha if it is dirty)
- So version is grabbed automatically from main header file where you set it based on rules we will talk about later in this presentation

Finding used apps/libs

```
AC_PROG_CC_C99
AC_PROG_CPP
AC_PROG_INSTALL
AC_PROG_LN_S
AC_PROG_MAKE_SET
PKG_PROG_PKG_CONFIG([0.20])

PKG_CHECK_MODULES([CURL],[
    libcurl
])
PKG_CHECK_MODULES([XDG],[
    libxdg-basedir
])
```

Hand-searching for libraries

```
AC_CHECK_HEADERS([buffio.h tidy.h], [], [  
    AC_MSG_ERROR([Unable to find htmltidy headers])  
])  
AC_SEARCH_LIBS([tidyParseBuffer], [tidy], [], [  
    AC_MSG_ERROR([Unable to find htmltidy library])  
])
```


Automagicness in autoconf

```
AC_ARG_WITH([foo],
    AS_HELP_STRING([--without-foo],
        [Ignore presence of foo and disable it]))

AS_IF([test "x$with_foo" != "xno"],
    [CHECK_FOR_FOO([have_foo=yes], [have_foo=no])],
    [have_foo=no])

AS_IF([test "x$have_foo" = "xyes"],
    [do_whatever_needed],
    [AS_IF([test "x$with_foo" = "xyes"],
        [AC_MSG_ERROR([foo requested but not found])])])
])
```

Werror handling

```
AC_ARG_ENABLE([werror],
    [AS_HELP_STRING([--disable-werror],
        [Treat all warnings as errors])],
    [enable_werror="$enableval"],
    [enable_werror=yes]
)
AS_IF([test x"$enable_werror" != "xno"], [
    CFLAGS="$CFLAGS -Werror"
])
CFLAGS="$CFLAGS -Wall -Wextra -pedantic"
```

Generate results

```
AC_CONFIG_FILES([
    Makefile
    mylibrary.pc
])
AC_OUTPUT
```

```
AC_MSG_NOTICE([
```

```
Build configuration:
```

```
    werror:           ${enable_werror}
```

```
])
```

M4 macros

M4 macros sample

```
AC_DEFUN([libo_CHECK_EXTENSION],[
AC_ARG_ENABLE(ext-$4,
    AS_HELP_STRING([--enable-ext-$4],
        [Enable the $1 extension])
)
AC_MSG_CHECKING([for building the $1 extension])
$2_EXTENSION_PACK=
if test "x$enable_ext_$3" = "xyes"; then
    BUILD_TYPE="$BUILD_TYPE $2"
    AC_MSG_RESULT([yes])
else
    AC_MSG_RESULT([no])
fi
AC_SUBST($2_EXTENSION_PACK)
])
```

Getting more macros

`http://www.gnu.org/software/autoconf-archive/`

Automake aka Makefile.am

Non-recursive Makefiles

- Easier to see the changes in one file in top `$(srcdir)`
- Faster with paralelization, because of the depgraph calculations
- Easier to write the stack as you always start from the top

Basic declarations

ACLOCAL_AMFLAGS = -I m4

AM_LDFLAGS = -no-undefined

pkgconfdir = \$(libdir)/pkgconfig

pkgconf_DATA = libexample.pc

EXTRA_DIST = \
 autogen.sh \
 libexample.pc.in \
 data/extrafile.blob \
 \$(MY_VARIABLE) \
 .version version.sh

Adding sources

```
bin_PROGRAMS = binary
binary_SOURCES = \
    src/file.c \
    src/example.c \
    src/whatever.c
binary_CFLAGS = \
    $(EXTERNALPACKAGE_CFLAGS) \
    -I$(srcdir)/where/ever/
binary_LDADD = \
    $(EXTERNALPACKAGE_LIBS) \
    libexample.la
```

Adding headers

```
libexample_includedir = $(includedir)/libexample
libexample_include_HEADERS = \
    libexample/example.h \
    config.h

noinst_HEADERS = \
    libexample/example_internal1.h \
    include/example_internal2.h
```

Creating tests

```
check_PROGRAMS = test
TESTS = test
test_SOURCES = test/test.c
test_CFLAGS = \
    $(EXTERNALPKG_CFLAGS)
test_LDADD = \
    $(EXTERNALPKG_LIBS) \
    lib.a
test_LDFLAGS = -no-install
```

Creating custom rules

```
distclean-local:
```

```
rm -rf *.cache *~
```

```
dist-hook:
```

```
git log --date=short \
--pretty="format:@%cd \
%an <%ae> [%H]%n%n%s%n%n%e%b" \
| sed -e "s|^\[^\@]\|t\1|" \
-e "s|^@\||" >$(distdir)/ChangeLog
```

```
.version: version.sh .git
```

```
./$< > $@
```

Libtool

Libtool versioning

- Start with version information of '0:0:0' for each libtool library
- If the library source code has changed at all since the last update, then increment revision ('c:r:a' becomes 'c:r+1:a')
- If any interfaces have been added, removed, or changed since the last update, increment current, and set revision to 0
- If any interfaces have been added since the last public release, then increment age
- If any interfaces have been removed or changed since the last public release, then set age to 0

configure.ac changes

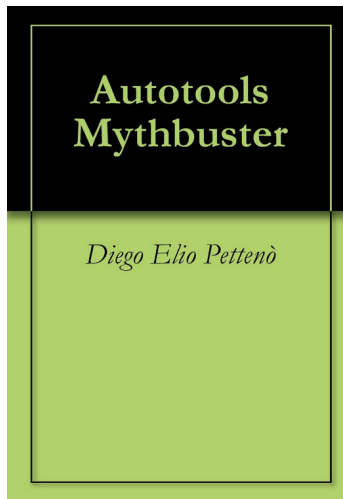
```
LT_VERSION=m4_esyscmd([./version.sh -v])  
LT_INIT([disable-static pic-only])  
AC_PROG_LIBTOOL
```


Makefile.am changes

```
lib_LTLIBRARIES = libexample.la
libexample_la_SOURCES = \
    src/something.c \
    src/somethingelse.c \
    src/whatever.c
libexample_la_CFLAGS = \
    $(MYEXTERNALPACKAGE_CFLAGS)
libexample_la_LDFLAGS = \
    $(MYEXTERNALPACKAGE_LIBS) \
    -version-info $(LT_VERSION) \
    -export-symbols-regex '^foo_'
```

Reading

Reading



Endnote

Thanks

Thank you for your attention.