**configure.ac (configure.in)详解**

[](https://www.jianshu.com/u/b1d20cb66d89)

[id被吃](https://www.jianshu.com/u/b1d20cb66d89)

2017.02.07 20:29\* 字数 2130 阅读 3725评论 0喜欢 1

转载自：[configure.ac (configure.in)详解--鹏博客](https://link.jianshu.com?t=http:/www.ivpeng.com/pblog/configure-ac.html)（http://www.ivpeng.com/pblog/configure-ac.html）

用自动生成Makefile的工具来生成Makefile的过程中，需要用autoscan命令来生成configure.scan文件，然后将它改名字为configure.ac或者configure.in,然后就来更改其中的一些相关信息来完成下面需要完成的配置，那么这个configure.ac怎么来写？今天我们就将这个过程分开来和大家共同研究和探讨下。

我们在生成的过程中会用autoconf命令来处理configure.ac/configure.in文件，生成一个configure的脚本。生成后的configure文件是一个可以移植的shell脚本，运行的时候它检查编译环境，来决定哪些库是可以用的，所用到的平台有哪些个特征，那些个头文件和库是已经找到的等等， 然后收集到的这些信息，它修改编译标记，生成一个Makefile文件，同时生成一个包含已定义的预处理符号的config.h文件。configure并不需要运行autoconf，所以我们在发布应用程序之前生成这个文件，如此我们就不必有autoconf的软件包了。

说了这么多现在应该步入主题了，现在我们需要坐的就是写一个configure.in文件。configure.in文件里基本的内容就是一系列的m4宏，在运行时根据传递给它们的参数，定义的宏就会扩展为shell的脚本代码段。也可以手工书写shell代码。不过我们就不说这个了，要想完全的理解configure.in怎样写，其实是要求有一些m4的知识，还要一些 Bourneshell的知识。要这么多知识做基础好难，不过还好的的是，有省事的方法，就是可以找一个已有的configure.in文件，然后以此来修改它以适应你所要完成的任务。如果想了解可以看看autoconf手册，其中介绍了很多预先写好的宏。

下面我们就来通过一个简单的configure.ac文件，来了解一些非常常用的宏和含义：

AC\_INIT(src/hello.c)

AM\_CONFIG\_HEADER(config.h)

AM\_INIT\_AUTOMAKE(GnomeHello,0.1)

AM\_MAINTAINER\_MODE

AM\_ACLOCAL\_INCLUDE(macros)

GNOME\_INIT

AC\_PROG\_CC

AC\_ISC\_POSIX

AC\_HEADER\_STDC

AC\_ARG\_PROGRAM

AM\_PROG\_LIBTOOL

GNOME\_COMPILE\_WARNINGS

ALL\_LINGUAS=”es”

AM\_GNU\_GETTEXT

AC\_SUBST(CFLAGS)

AC\_SUBST(CPPFLAGS)

AC\_SUBST(LDFLAGS)

AC\_OUTPUT([

Makefile

macros/Makefile

src/Makefile

intl/Makefile

po/Makefile.in

pixmaps/Makefile

doc/Makefile

doc/C/Makefile

doc/es/Makefile

])

下面我们逐一对例子中的代码进行一下解释，例子中以AC开头的宏来自autoconf，以AM开头的宏来自automake。可以从autoconf或 automake中寻求帮助，这一点很有用。以GNOME开头的宏来自于Gnomemacros目录。这些宏都是用m4宏语言写的。如果将 autoconf和automake安装在/usr目录下，autoconf和automake中的标准宏一般放在/usr/share/aclocal 目录下。

**AC\_INIT**总是configure.in中的第一个宏。它扩展为许多可由其他configure脚本共享的模板文件代码。这些代码解析传到 configure中的命令行参数。这个宏的一个参数是一个文件名，这个文件应该在源代码目录中，它用于健全性检查，以保证configure脚本已正确定位源文件目录。

**AM\_CONFIG\_HEADER**指定了要创建的头文件，差不多总是config.h。创建的头文件包含由configure定义的C预处理符号。最低限度应该定义PACKAGE和VERSION符号，这样可以将应用程序名称和版本传送到代码中，而无须对它们硬编码(非公用的源文件应该包含 config.h(#include)以利用这些定义。然而，不要将config.h文件安装到系统中，因为它有可能与其他的软件包冲突)。

**AM\_INIT\_AUTOMAKE**初始化automake。传到这个宏里的参数是要编译的应用程序的名称和版本号(这些参数成为config.h中定义的PACKAGE和VERSION值)。

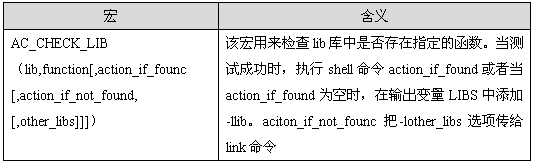
**AM\_MAINTAINER\_MODE**关闭缺省时仅供程序维护者使用的makefile目标，并修改以使configure能理解 –enable-maintainer-mode选项。–enable-maintainer-mode将maintaineronly目标重新打开。仅供维护者使用的makefile目标允许最终用户清除自动生成的文件，比如configure，这意味着要修复编译故障，必须安装有autoconf和automake软件。注意，因为autogen.sh脚本主要是给开发人员用的，autogen.sh会自动传递一个–enable- maintainer-mode选项给configure。

**AM\_ACLOCAL\_INCLUDE**指定一个附加的目录，用于搜索m4宏。在这里，它指定为macros子目录。在这个目录中应该有Gnome宏的拷贝。

**GNOME\_INIT**给configure添加一个与Gnome相关的命令行参数个数，并为Gnome程序定义一些makefile变量，这些变量中包含了必要的预处理程序和链接程序标志。这些标志是由gnome-config脚本取得的。安装gnome-libs时会安装gnome- config脚本。

**AC\_PROG\_CC**定位C编译器。

**AC\_CHECK\_LIB**如果程序中加入了多线程用到的锁的话就要加入像 AC\_CHECK\_LIB([pthread], [main])这样的检测，这个宏的含义如下：



LIBS是link的一个选项，程序中使用了读写锁，所以要测试pthread库中是否存在pthread\_rwlock\_init函数。

**AC\_PROG\_RANLIB**如果是多线程的程序的话要加入这句话，要不运行automake命令时会出错。

**AC\_ISC\_POSIX**添加一些在某些平台上实现POSIX兼容需要的标志。

**AC\_HEADER\_STDC**检查当前平台上是否有标准的ANSI头文件，如果有，则定义STDC\_HEADERS。

**AC\_ARG\_PROGRAM**添加一些选项到configure中，让用户能够修改安装程序的名称(如果在用户系统上碰巧有一个与要安装的程序名称相同的程序，这是很有用的)。

**AM\_PROG\_LIBTOOL**是由automake用来设置libtool的用途的。只在计划编译共享库或动态可加载模块时才需要设置这个值。

**GNOME\_COMPILE\_WARNINGS**给gcc命令行添加许多警告选项，但是在其他绝大多数的编译器上什么也不做。

**ALL\_LINGUAS=“es”**不是一个宏，只是一句shell代码。它包含一个由空格分隔的语言种类缩写表，对应于po子目录下的.po文件。.po文件包含翻译成其他语言的文本，所以ALL\_LINGUAS应该列出程序已经被翻译成的所有语言。

**AM\_GNU\_GETTEXT**由automake使用，但是这个宏会随gettext软件包发布。它让 automake执行一些与国际化相关的任务。

**AC\_SUBST**输出一个变量到由configure生成的文件中。具体内容将在后面说明。

**AC\_OUTPUT**列出由configure脚本创建的文件。这些文件都是由带.in后缀的同名文件生成的。例如，src/Makefile是由src/Makefile.in生成的，config.h是由config.h.in生成的。在执行AC\_OUTPUT宏时，configure脚本处理包含有两个@符号标志的变量(例如@PACKAGE@)的文件。只有用AC\_SUBST输出了变量，它才能识别这些变量(许多在上面讨论过的预先写好的宏都用AC\_SUBST定义变量)。这些特征用于将一个Makefile.in文件转换成一个Makefile文件。典型情况下，Makefile.in 是由automake从Makefile.am（了解更多Makefile.am的写法请阅读《[Makefile.am 规则和实例详解](https://link.jianshu.com?t=http:/www.ivpeng.com/pblog/makefile-am.html)》）生成的(不过，你可以只用autoconf，而不用automake，自己编写一个 Makefile.in)。

上面例子中提到的宏都是基本配置和常用的，还有一些可以用到的宏大家有兴趣的话可以学习收集下，不过基本上上面所说的已经可以完成一次配置，不过这里写的一些配置大多数是收集于网络，希望大家和鹏博客共同学习提高，有不对的地方希望大家能之出，希望大家能支持。