<http://www.laruence.com/2009/11/18/1154.html>

## 概览

autoconf 根据一个宏文件生成configure文件，automake根据另一个宏文件生Makekfile.in文件，在使用configure根据Makefile.in生存Makefile文件。

## 实例

hello.cpp文件

第一步：生成autoconf的输入文件configure.ac

autoscan:生成configure.ac模版文件configure.scan，修改为：

|  |
| --- |
| # -\*- Autoconf -\*-  # Process this file with autoconf to produce a configure script.  #AC\_PREREQ([2.69])  AC\_INIT(hello.cpp)  AM\_INIT\_AUTOMAKE(hello,1.0)  #AC\_CONFIG\_SRCDIR([hello.cpp])  #AC\_CONFIG\_HEADERS([config.h])  # Checks for programs.  AC\_PROG\_CXX  # Checks for libraries.  # Checks for header files.  # Checks for typedefs, structures, and compiler characteristics.  # Checks for library functions.  AC\_OUTPUT(Makefile) |

mv configure.scan configure.ac

aclocal：生成m4文件

autoheader:生成config.h.in

autoconf：根据confiure.ac和m4文件生成configure

第二步：automake生成Makefile.in

vim Makefile.am:

|  |
| --- |
| AUTOMAKE\_OPTIONS=foreign  bin\_PROGRAMS=hello  hello\_SOURCES=hello.cpp |

automake --add-missing:生成Makefile.in文件。

第三步：./configure使用Makefile.in生成Makefile文件

第四步：make；./hello

## 命令解释

autoscan：该命令扫描工作目录，生成configure.scan，需要将其重命名为configure.ac（作为autoconf输入文件，autoconf展开其中的m4宏），然后修改。

|  |
| --- |
| # -\*- Autoconf -\*-  # Process this file with autoconf to produce a configure script.  AC\_PREREQ([2.69]) #表示本文autoconf版本  AC\_INIT([FULL-PACKAGE-NAME], [VERSION], [BUG-REPORT-ADDRESS])#定义软件的名称和版本信息，改为AC\_INIT(hello.cpp)  AC\_CONFIG\_SRCDIR([hello.cpp])#通过侦测该文件是否存在，来确定源码目录是否有效，可以选择源码目录的任何一个文件  AC\_CONFIG\_HEADERS([config.h])#生成config.h，如果程序中需要使用则添加#include<config.h>  AM\_INIT\_AUTOMAKE(hello,1.0) #指定使用automake生成Makefile.in  AC\_CONFIG\_FILES([Makefile src/Makefile data/Makefile])#指定在这些目录下生成Makefile，而不只是根目录下  # Checks for programs.  AC\_PROG\_CXX  # Checks for libraries.  # Checks for header files.  # Checks for typedefs, structures, and compiler characteristics.  # Checks for library functions.  AC\_OUTPUT(Makefile)#指定生成文件名 |

## debug版本

./configure CFLAGS=’-g -O0’

make