

configure.ac 由一些宏组成(如果已经有源代码,你可以运行autoscan来产生一个configure.scan文件,在此基础上修改成configure.ac将更加方便.

最基本的组成可以是:

```
AC_INIT([FULL-PACKAGE-NAME], [VERSION], [BUG-REPORT-ADDRESS])
# Checks for programs.
# Checks for libraries.
# Checks for header files.
# Checks for typedefs, structures, and compiler characteristics.
# Checks for library functions.
# Output files.
AC_CONFIG_FILES([FILES])
AC_OUTPUT
```

## AC\_INIT(PACKAGE, VERSION, BUG-REPORT-ADDRESS)

autoconf 强制性初始化。告诉autoconf包名称, 版本, 一个bug报告email。

例如: AC\_INIT([hello], [1.0], [bug-report@address])

并且这些名称将出现在config.h, 你可以在程序直接引用这些宏。

```
/* Define to the address where bug reports for this package should be sent. */ #define PACKAGE_BUGREPORT "BUG-REPORT-ADDRESS" /*
Define to the full name of this package. */ #define PACKAGE_NAME "FULL-PACKAGE-NAME" /* Define to the full name and version of this
package. */ #define PACKAGE_STRING "FULL-PACKAGE-NAME VERSION" /* Define to the one symbol short name of this package. */ #define
PACKAGE_TARNAME "full-package-name" /* Define to the version of this package. */ #define PACKAGE_VERSION "VERSION"
```

## AC\_PREREQ(VERSION)

需要的最低autoconf版本, 如: AC\_PREREQ([2.65])

## AC\_CONFIG\_SRCDIR(FILE)

一个安全的检查。FILE将是一个发布的源文件。这让configure脚本确保自己运行在正确的目录中。如:

AC\_CONFIG\_SRCDIR([src/main.c])

## AC\_CONFIG\_AUX\_DIR(DIRECTORY)

配置辅助脚本位置, 如install-sh和depcomp等。如: AC\_CONFIG\_AUX\_DIR([build-aux])

## AC\_PROG\_CC, AC\_PROG\_CXX, AC\_PROG\_F77, ...

编译器检测。

## AC\_PROG\_SED, AC\_PROG\_YACC, AC\_PROG\_LEX, ...

发现好的实现并且设置变量SED, YACC, \$LEX等。

## AC\_CHECK\_PROGS(VAR, PROGS, [VAL-IF-NOT-FOUND])

将VAR定义为第一个发现的程序, 如果没有发现就设置为VAL-IF-NOT-FOUND

```
if      "$STAR"      then
fi
```

如果发现tar,gtar, 就设置到变量\$STAR中, 如果没有发现就是设置为‘:

## AC\_MSG\_ERROR(ERROR-DESCRIPTION, [EXIT-STATUS])

打印错误信息ERROR-DESCRIPTION并退出, EXIT-STATUS设置退出状态。

## AC\_MSG\_WARN(ERROR-DESCRIPTION)

打印但是不退出。

## AC\_DEFINE(VARIABLE, VALUE, DESCRIPTION)

输出到config.h中。

```
/* DESCRIPTION */ #define VARIABLE VALUE
```

## AC\_SUBST(VARIABLE, [VALUE])

定义变量\$VARIABLE=VALUE,进入Makefile。

如: AC\_SUBST([FOO],[foo])

等价于

```
FOO=foo
```

```
AC_SUBST([FOO])
```

或者

```
AC_SUBST([FOO])
```

```
FOO=foo
```

## AC\_CHECK\_LIB(LIBRARY, FUNCT, [ACT-IF-FOUND], [ACT-IF-NOT])

检查库是否存在并且包括函数FUNCT。如果发现库就执行ACT-IF-FOUND, 否则执行ACT-IF-NOT。

例如:

```
AC_CHECK_LIB([efence], [malloc], [EFENCELIB=-leference])
```

```
AC_SUBST([EFENCELIB])
```

- 1
- 2

稍后可以使用

(EFENCELIB)在链接规则中。如果没有使用ACT-IF-FOUND项, 并且库被发现存在。ACcHECKLIB将会做下面的操作, LIBS="-lLIBRARY

LIBS"和#dedefine HAVE\_LIB{LIBRARY}。Autmake会在链接的时候使用\$LIBS。

## AC\_CHECK\_HEADERS(HEADERS...)

检查头HEADERS并且给每一个发现的头文件定义#define HAVE\_HEADER\_H

例如:

```
        .h          .h
        .h
```

检查sys/param.h unistd.h如果存在定义#define HAVE\_SYS\_PARAM\_H,HAVE\_UNISTD\_H。

注意: 一般情况下不能转换的符号都用‘\_’代替。

你可以在你的头文件这么使用

```
#if HAVE_UNISTD_H
```

```
#include <unistd.h>
```

```
#endif
```

## AC\_CHECK\_HEADER(HEADER, [ACT-IF-FOUND], [ACT-IF-NOT])

检查一个头文件。

## AC\_CONFIG\_HEADERS(HEADERS...)

创建头文件的HEADER.in, HEADERS包含使用AC\_DEFINE的定义。

例如:

```
AC_CONFIG_HEADERS([config.h])
```

将从config.h.in创建config.h。

## AC\_CONFIG\_FILES([Makefile sub/Makefile script.sh:script.in])

Automake创建FILE.in从每一个有FILE.am的FILE。他也可以出来非Makefile。

```
script.in
```

```
#!/bin/sh
```

```
'@SED@'
```

```
'@TAR@'
```

```
$1 "$d"
```

```
for do
```

```
"$SED" 's/#.*//' "$f"
```

```
"$d/$f"
```

```
done
```

```
"$TAR" "$d.tar" "$d"
```

.in是文件模板。@SED@、@TAR@将被AC\_SUBST([SED]),AC\_SUBST([TAR])替换。这样你就可以将配置中的变量传送到其他文

件中去。  
Makefile.in也使用@XYZ@作为替换部分，但是Automake已经做了将所有的XYZ=@XYZ@的定义，你只需要简单使用\$(XYZ)就可以了。

AM\_INIT\_AUTOMAKE([OPTIONS...])

检查automake尝试Makefile时的必要的工具。  
例如：AM\_INIT\_AUTOMAKE([foreign -Wall -Werror])  
一些可选的选项：

| 选项          | 解释   |
|-------------|--|
| -Wall       | 打开全部警告                                     |
| -Werror     | 将警告当错误处理                                   |
| -foreign    | 放宽一些GNU标准需求                                |
| -1.11.1     | 需要automake的最低版本                            |
| -dist-bzip2 | 在使用make dist和make distcheck期间同时创建tar.bz2存档 |
| -tar-ustar  | 使用ustar格式创建tar存档                           |

AC\_CONFIG\_FILES([Makefile sub/Makefile])

写入Makefile.am和sub/Makefile.am文件

AM\_CONDITIONAL(NAME, CONDITION)

定义一个条件名NAME,CONDITION将由shell解释执行，如果成功NAME将被打开。  
举例：configure.ac

```
AC_CHECK_HEADER [bar.h] [use_bar=yes]
AM_CONDITIONAL [WANT_BAR] [test "$use_bar" = yes]
```

仅当当前系统中存在bar.h文件时打开WANT\_BAR。