# Autoconf Framework

22nd Feburary 2006

# Outline

# Sources of Non-portability

- Functions like strtod, strtof may not exist
- Function like strchr and index may be used interchangeably
  In Linux strchr is provided in sting.h while index is provided in strings.h
- Function are defined in different headers sting.h, strings.h, memory.h
- Prototypes can be different e.g setpgrp(void) or setpgrp(int, int)

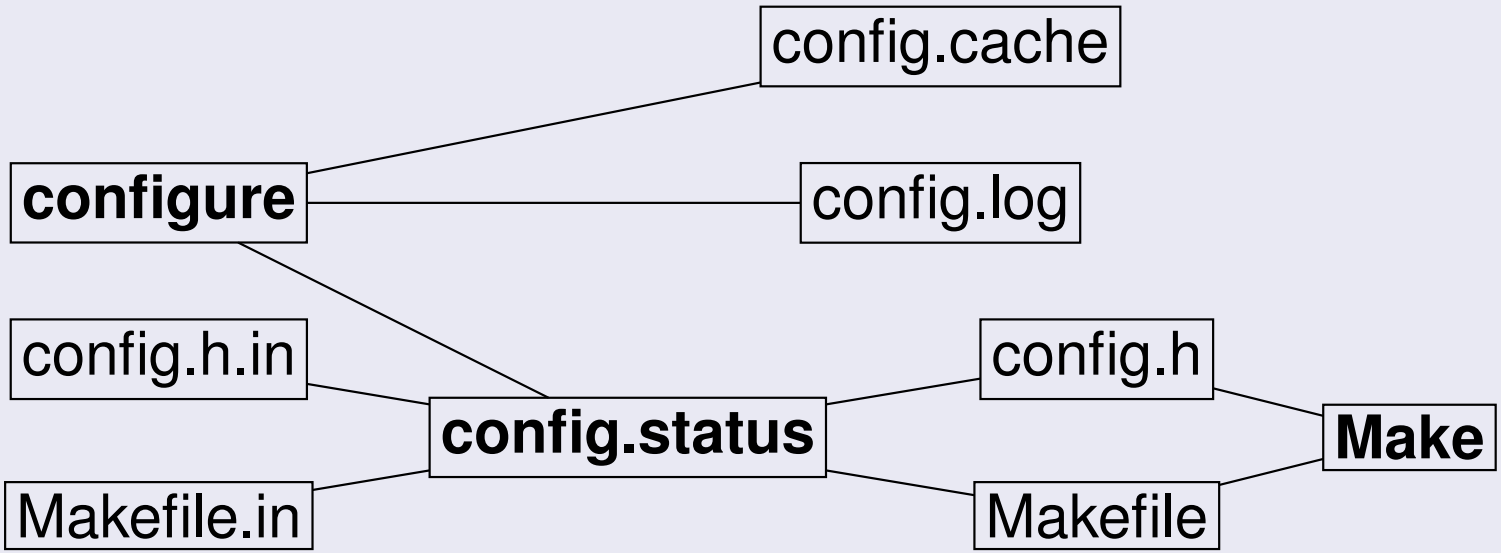
## Solution

- Slice the code with lot of #if/#else
- Create macros
- Create replacement function

**Later 2 are the ideal methods**

# Configure Block

## Configure block



## Default Targets for Make

- **make all** - Build programs, libraries, documentation etc
- **make install** - Installs
- **make install-strip** - strip the debugging symbols and install
- **make uninstall** - Uninstall
- **make clean** - Cleans all
- **make distclean** - Cleans up the distribution
- **make check** - Checks, runs tests
- **make installcheck** - Checks the installation
- **make dist** - Makes the distribution
- **make distcheck** - Makes distribution and Checks it

## Make targets are important

If we support a third party Makefile, as it should properly take care of the actions against given Target

# Autoconf distribution

## What gets distributed

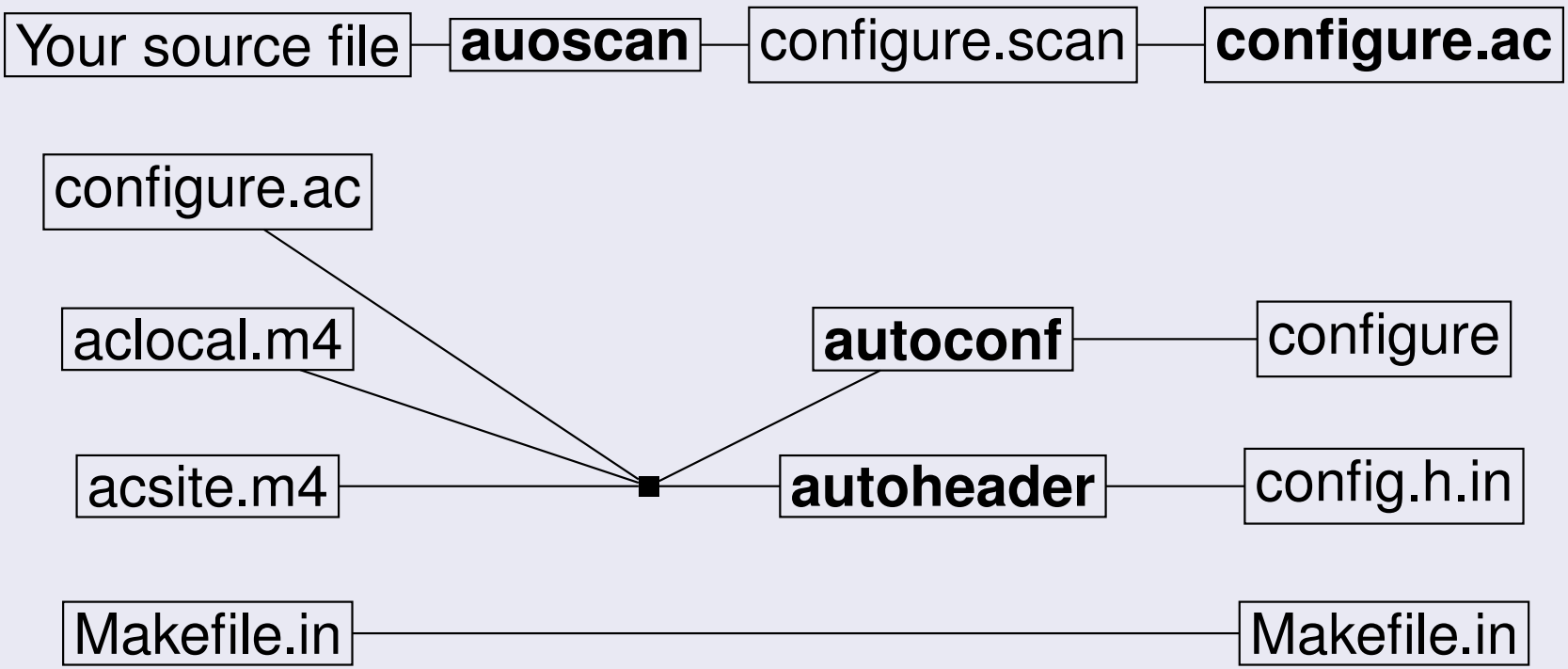
- All the m4 file
- configure script
- All the third party tools output
  - Makefile.in the Output from Makefile.am from automake
  - All the script needed by automake
  - Config.h.in the Output from autoheader
  - C Source from libtoolize in case of plug-in support
  - C Source and Macros from autopoint, gettextize etc. . .
- All the files marked by the Makefile.am though SOURCES and EXTRA flags

## ./configure script is independent of other framework

This is still a open issue, whether to distribute ./configure script as it is auto-generated.

# Autoconf block

## Autoconf Block



# Writing configure.ac

## configure.ac Structure

- Autoconf Requirements e.g **AC_PREREQ(version)** - Minimum version of Autoconf required
- **AC_INIT(package, version, bug-report-address)**
- Initialize Third Party software e.g **AM_INIT_AUTOMAKE**, **AM_CONFIG_HEADER**
- Check for Programs e.g **AC_PROG_CC**
- Check for Libraries e.g **AC_CHECK_LIB(m, cos)**
- Check for Header files e.g **AC_CHECK_HEADERS(stdlib.h varargs.h)**
- Check for types e.g **AC_TYPE_OFF_T**
- Check for Structure e.g **AC_CHECK_ST_BLKSIZE**
- Check for compiler Characteristics e.g **AC_C_INLINE**
- Check for Library Functions .eg **AC_CHECK_FUNCTIONS([bzero memset, break])**
- Check for system services e.g **AC_CHECK_TOOL(...)**
- **AC_CONFIG_FILES([files...])**
- **AC_OUTPUT**

# autoconf Files

## autoconf Output

- During configuration the configure script creates a file **confdef.h**, which can be used in the end e.g

  *cp confdef.h mydef.h*

  This method should not be used as the temp file name can change
- **AC_OUTPUT** will produce **config.status** script, which can be run - Used generally to speed up autoconf process
- Setting of C Preprocessor symbol
- Results are saved to Cache file
- Standard output e.g **AC_MSG_CHECKING**
- Files specified in **AC_CONFIG_FILES** - Input files (.in) to Output files with @value@ replaces e.g

  *Makefile.in( prefix = @prefix@) –> Makefile ( prefix = /usr/local)*

## Setting of preprocessor symbol

Use **AC_DEFINE(variable,value,[description])**

## Setting of Output variable

Use **AC_SUBST(variable,value)**

## Saving to Cache

1. Use **AC_CACHE_SAVE**
2. Use **AC_CACHE_LOAD** <– Called automatically from AC_INIT
3. Use **AC_CACHE_VAL(cache-id,command to set it)**
4. Use **AC_CACHE_CHECK(message,cache-id,command-to-set-it** - wrapper over AC_CACHE_VAL with message

# Using Shell Scripts

- Autoconf uses [ ] for its own purpose, so shell script should avoid using them
- For testing use the command **test**

  *Instead of: if [ "$abc" = "yes]; then . . . ; fi*

  *Use : if test "$abc" = "yes" ; then . . . ; fi*

- In certain cases [ ] becomes unavoidable e.g grep
  1. Use [[ ]]
  2. Use **changequote({,})** and back **changequote([,])**
  3. Use Quadrigraphs : @<:@ -> [ , @>:@ -> ]

**Use portable scripts defaulting to sh**

Don't use Bash, ksh special features

# Example

```
AC_INIT(ttt,0.0.1,admin@sendmail.com)
AC_PROG_CC
AC_PROG(C)
AC_CHECK_LIB(m,pow)
AC_STRUCT_ST_BLKSIZE
cp confdefs.h my_config.h
```

**Run autoconf and ./configure**

```
#define PACKAGE_NAME ""
#define PACKAGE_STRING ""
#define HAVE_LIBM 1
#define STDC_HEADERS 1
#define HAVE_SYS_TYPES_H 1
#define HAVE_SYS_STAT_H 1
#define HAVE_STDLIB_H 1
#define HAVE_STRING_H 1
#define HAVE_MEMORY_H 1
#define HAVE_STRINGS_H 1
#define HAVE_INTTYPES_H 1
#define HAVE_STDINT_H 1
#define HAVE_UNISTD_H 1
#define HAVE_STRUCT_STAT_ST_BLKSIZE 1
#define HAVE_ST_BLKSIZE 1
```

# Adding Messages

- **AC_MSG_CHECKING(feature Description)**
- **AC_MSG_RESULT(result Description)**
- **AC_MSG_NOTICE(message)**
- **AC_MSG_ERROR(error-description,[exit-status])**
- **AC_MSG_FAILURE(error-description,[exit-situs])**
- **AC_MSG_WARN(problem-description)**
- **echo** shell script

## Adding your own Arguments

- AC_ARG_WITH(package, help-string, [action-if-given], [action-if-not-given])
- AC_HELP_STING(lhs,rhs)

```
AC_ARG_WITH([xyz], AC_HELP_STRING([--with-xyz],
                                  [use xyz (default is no)]),
                  [AC_MSG_NOTICE([--- $with_xyz]);
                        AC_DEFINE_UNQUOTED(XYZ,[$with_xyz]);
                  [ac_cv_use_abc=$with_xyz]],[ac_cv_use_abc=no])
# if passed with --with-xyz=abarakadabra
    Adds: #define XYZ abarakdabra
```

# Check Functions

1. **AC_CHECK_LIB**(library,function,[action-if-found], [action-if-not-found], [include=other-libs])
2. **AC_CHECK_FUNC**(function,[action-if-found], [action-if-not-found])
3. **AC_CHECK_FUNCS**([functions . . . ],[action-if-found], [action-if-not-found])
4. **AC_CHECK_HEADERS**([header-file . . . ],[action-if-found], [action-if-not-found], [include=default-includes])
5. **AC_CHECK_HEADER**(header-file,[action-if-found], [action-if-not-found], [include=default-includes])
6. **AC_CHECK_HEADERS**([header-file . . . ],[action-if-found], [action-if-not-found], [include=default-includes])
7. **AC_CHECK_DECL**(symbol,[action-if-found], [action-if-not-found], [include=default-includes])
8. **AC_CHECK_DECLS**([symbol . . . ],[action-if-found], [action-if-not-found], [include=default-includes])
9. **AC_CHECK_MEMBER**(aggregate.member,[action-if-found], [action-if-not-found], [include=default-includes])
10. **AC_CHECK_MEMBERS**([members . . . ],[action-if-found], [action-if-not-found], [include=default-includes])
11. **AC_CHECK_TYPE**(type,[action-if-found], [action-if-not-found], [include=default-includes])
12. **AC_CHECK_TYPES**([type . . . ],[action-if-found], [action-if-not-found], [include=default-includes])
13. **AC_CHECK_SIZEOF**([type . . . ],[unused], [include=default-includes])

# Others . . .

- **AC_LANG**(language)
- **AC_PROG_CC** <– Check for C
- **AC_PROG_CPP** <– Check for preprocessor
- **AC_PROG_CXX** <– Check for C++
- **AC_CANONICAL_BUILD**
- **AC_CANONICAL_HOST**
- **AC_CANONICAL_TARGET**
- **AC_DEFUN**(macro-name, body)

## Uses autom4te over default m4

1. m4sugar - add on macros for extending autoconf framework
2. m4sh - add on macros for extending autoconf shell usage

# Extending Autoconf

## Avoid Name-space pollution

- Avoid names starting with AC, AM, AS, m4
- Avoid putting the macros to aclocal.m4 use acinclude.m4 or own names
- When aclocal is run it creates aclocal.m4, which includes other macros from third party e.g automake, your own macros

## Using Automake/Autoheader

When automake is to be used, export the macros using aclocal.
aclocal reads configure.ac file, and tries to find external macros
If AM_INIT_AUTOMAKE is found, it exports its macros into aclocal.m4
So default exceution order in such a case is

***aclocal*** *then* ***autoconf*** *[ then* ***autoheader*** *] then* ***automake***

# Autoconf framework

## Input / Output Graph