GNU autotools pour la publication et le déploiement de logiciels

François Gannaz

14 octobre 2004

Qu'est-ce que les Autotools ?

- Un ensemble d'exécutables autour de Autoconf/Automake.

Pour quoi faire ?

- Contrôler et automatiser le processus de construction à partir des sources.

Les Autotools sont officiellement appelés « The GNU build system ».

L'ancien processus classique : *make*

```
user@work> tar xzf program-0.1.tgz
user@work> cd program-1.0
user@work> make
```

L'ancien processus classique : *make*

```
user@work> tar xzf program-0.1.tgz
user@work> cd program-1.0
user@work> make


gcc -I. -g -O2 -c split.c
[...]
Compilation done. Type 'make install' to install.
user@work>
```

L'ancien processus classique : *make*

```
user@work> tar xzf program-0.1.tgz
user@work> cd program-1.0
user@work> make


gcc -I. -g -O2 -c split.c
split.cpp: Dans function << int main(int, char**) >>:
split.cpp:941: error: erreur d'analyse syntaxique before
  string constant
user@work>
```

L'ancien processus classique : *make*

```
user@work> tar xzf program-0.1.tgz
user@work> cd program-1.0
user@work> make


gcc -I. -g -O2 -c split.c
split.cpp: Dans function << int main(int, char**) >>:
split.cpp:941: error: erreur d'analyse syntaxique before
  string constant
user@work>


user@work> less INSTALL || less README
 [..]
user@work> vim Makefile
 [..]
```

Le nouveau processus classique : *configure*

```
user@work> tar xzf program-0.1.tgz
user@work> cd program-1.0
user@work> ./configure
```

Le nouveau processus classique : *configure*

```
user@work> tar xzf program-0.1.tgz
user@work> cd program-1.0
user@work> ./configure

checking build system type... i686-pc-linux-gnu
checking for gcc... gcc
[...]
config.status: creating Makefile
config.status: executing depfiles commands
user@work>
```

Le nouveau processus classique : *configure*

```
user@work> tar xzf program-0.1.tgz
user@work> cd program-1.0
user@work> ./configure


checking build system type... i686-pc-linux-gnu
[...]
Checking for GTK version ...
Error: The GUI requires GTK devel packages
  (which were not found).
Check "configure.log" if you do not understand
  why it failed.
user@work>
```

Rôle de *configure*

Quels fichiers crée *configure* ?

- Des *Makefile* ad hoc dans chaque répertoire du projet

Rôle de *configure*

Quels fichiers crée *configure* ?

- Des *Makefile* ad hoc dans chaque répertoire du projet
- Un fichier de macros C/C++ *config.h*

Rôle de *configure*

Quels fichiers crée *configure* ?

- Des *Makefile* ad hoc dans chaque répertoire du projet
- Un fichier de macros C/C++ *config.h*
- Un logfile, *configure.log*

Rôle de *configure*

Quels fichiers crée *configure* ?

- Des *Makefile* ad hoc dans chaque répertoire du projet
- Un fichier de macros C/C++ *config.h*
- Un logfile, *configure.log*

Qu'est-ce que *configure* ?

Rôle de *configure*

Quels fichiers crée *configure* ?

- Des *Makefile* ad hoc dans chaque répertoire du projet
- Un fichier de macros C/C++ *config.h*
- Un logfile, *configure.log*

Qu'est-ce que *configure* ?

- Un script shell.
  Portable sur tout Bourne shell (Unixes, cygwin...)

Exemple élaboré

```
Config files successfully generated by ./configure !

  Install prefix: /usr/local
  Data directory: /usr/local/share/mplayer
  Config direct.: /usr/local/etc/mplayer

  Byte order: little-endian
  Optimizing for: athlon-xp mmx mmx2 3dnow 3dnowex sse mtrr

  Languages:
    Messages/GUI: en
    Manual pages:  en

  Enabled optional drivers:
    Input: edl matroska cdda mpdvdkit2 vcd dvb
   Codecs: qtx xvid libavcodec real xanim dshow/dmo win32 faad2(internal) libmpeg2 liba52 mp3lib libvorbis
    Audio output: alsa oss nas sdl mpegpes(dvb)
   Video output: xvidix cvidix sdl vesa jpeg png mpegpes(dvb) fbdev svga aa opengl dga xv x11 xover tga
  Disabled optional drivers:
    Input: ftp network tv-v4l2 tv-v4l tv-bsdbt848 tv live.com dvdread smb
    Codecs: opendivx x264 libdv libdts libtheora toolame libmad liblzo gif
    Audio output: sgi sun jack esd arts dxr2 dsound win32 macosx
   Video output: winvidix bl zr zr2 dxr3 dxr2 directx gif89a caca ggi xmga mga xvmc directfb tdfx_vid tdfxfb 3dfx

'config.h' and 'config.mak' contain your configuration options.
Note: If you alter theses files (for instance CFLAGS) MPlayer may no longer
      compile *** DO NOT REPORT BUGS if you tweak these files ***

'make' will now compile MPlayer and 'make install' will install it.
Note: On non-Linux systems you might need to use 'gmake' instead of 'make'.

Check configure.log if you wonder why an autodetection failed (check whether
the development headers/packages are installed).
```

Historique

Les solutions en 1990

- Cygnus *configure*
- GCC *configure*
- Imake (X11)
- metaconfig (Perl)

Historique

Les solutions en 1990

- Cygnus *configure*
- GCC *configure*
- Imake (X11)
- metaconfig (Perl)

L'émergence d'Autoconf

juin 1991 Lars Appel maintient certains outils GNU :
Makefile + *make -D...*
(environ 20 paramètres suivant le système)

Historique

Les solutions en 1990

- Cygnus *configure*
- GCC *configure*
- Imake (X11)
- metaconfig (Perl)

L'émergence d'Autoconf

juin 1991 Lars Appel maintient certains outils GNU : script *configure* donnant un *Makefile*.

Historique

Les solutions en 1990

- Cygnus *configure*
- GCC *configure*
- Imake (X11)
- metaconfig (Perl)

L'émergence d'Autoconf

juin 1991 Lars Appel maintient certains outils GNU : script *configure* donnant un *Makefile*. Ajoute un fichier de *template* à son script.

Historique

Les solutions en 1990

- Cygnus *configure*
- GCC *configure*
- Imake (X11)
- metaconfig (Perl)

L'émergence d'Autoconf

juin 1991 Lars Appel maintient certains outils GNU : script *configure* donnant un *Makefile*. Ajoute un fichier de *template* à son script.

oct 1991 Création de Autoconf : générateur de *configure*. Le programme est basé sur le langage m4.

Historique

Les solutions en 1990

- Cygnus *configure*
- GCC *configure*
- Imake (X11)
- metaconfig (Perl)

L'émergence d'Autoconf

juin 1991 Lars Appel maintient certains outils GNU : script *configure* donnant un *Makefile*. Ajoute un fichier de *template* à son script.

oct 1991 Création de Autoconf : générateur de *configure*. Le programme est basé sur le langage m4.

juill. 1992 Autoconf 1.0

Les fondamentaux d'Autoconf/Automake

Buts d'Autoconf/Automake

L'objectif initial d'Autoconf était la portabilité sur UNIX.

Automake a été développé séparément pour le calcul des dépendances dans les sources.

Buts d'Autoconf/Automake

L'objectif initial d'**Autoconf** était la portabilité sur UNIX.

Automake a été développé séparément pour le calcul des dépendances dans les sources.

Objectifs des Autotools

- Génération automatique d'un *Makefile*.

Buts d'Autoconf/Automake

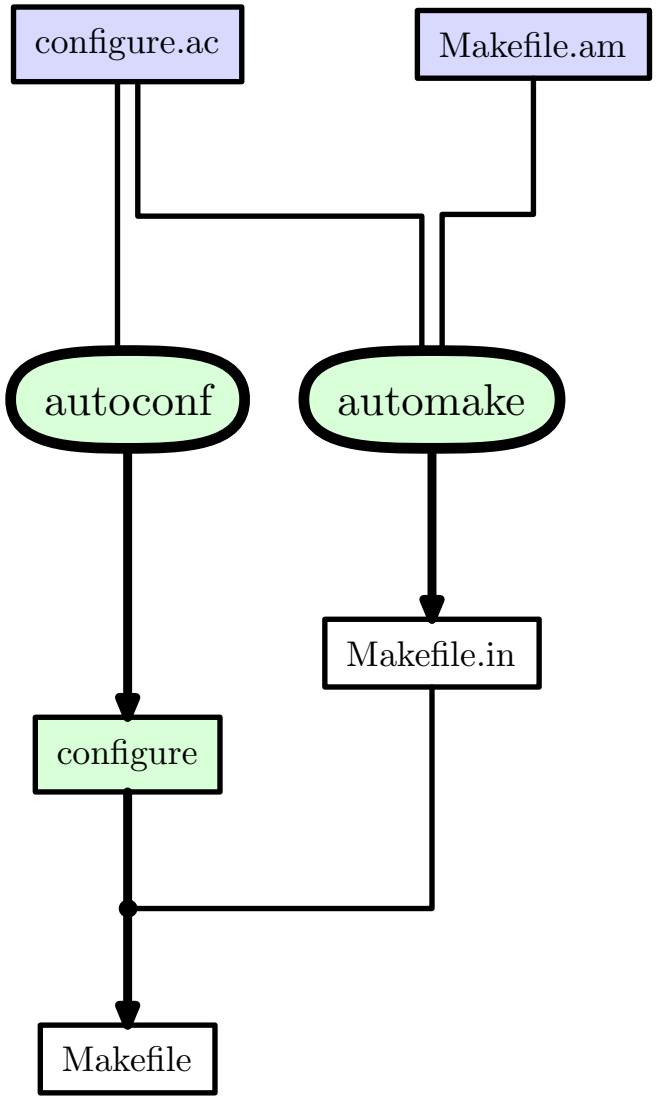
L'objectif initial d'**Autoconf** était la portabilité sur UNIX.

Automake a été développé séparément pour le calcul des dépendances dans les sources.

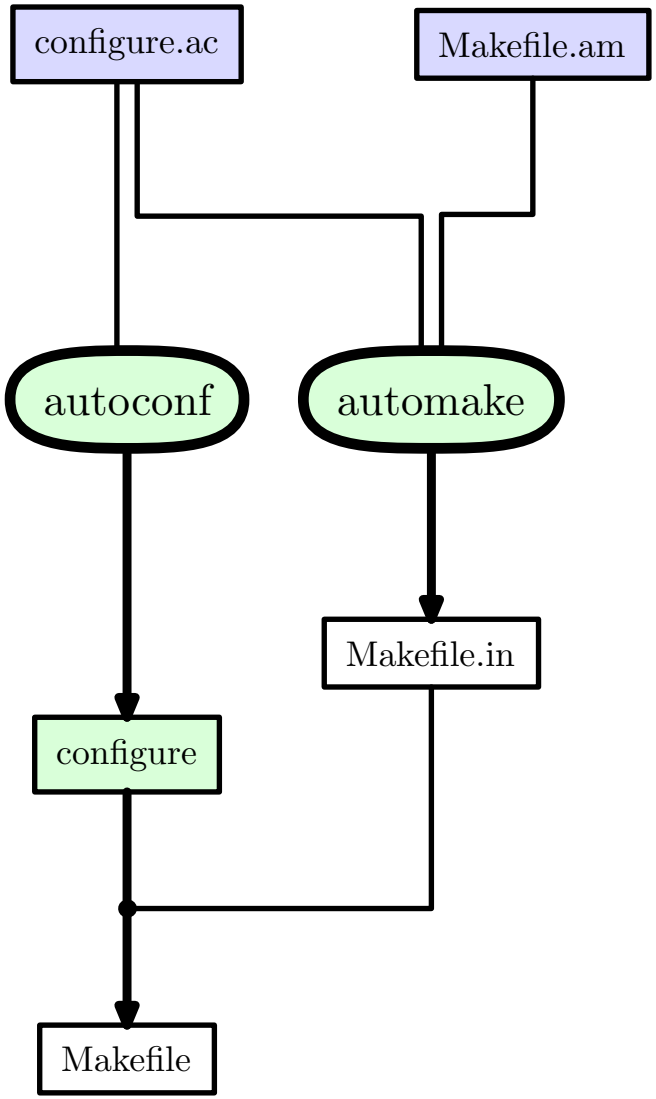
Objectifs des Autotools

- Génération automatique d'un *Makefile*.
- Pour cette tâche, génération d'un script *configure* portable.

Schema simplifié



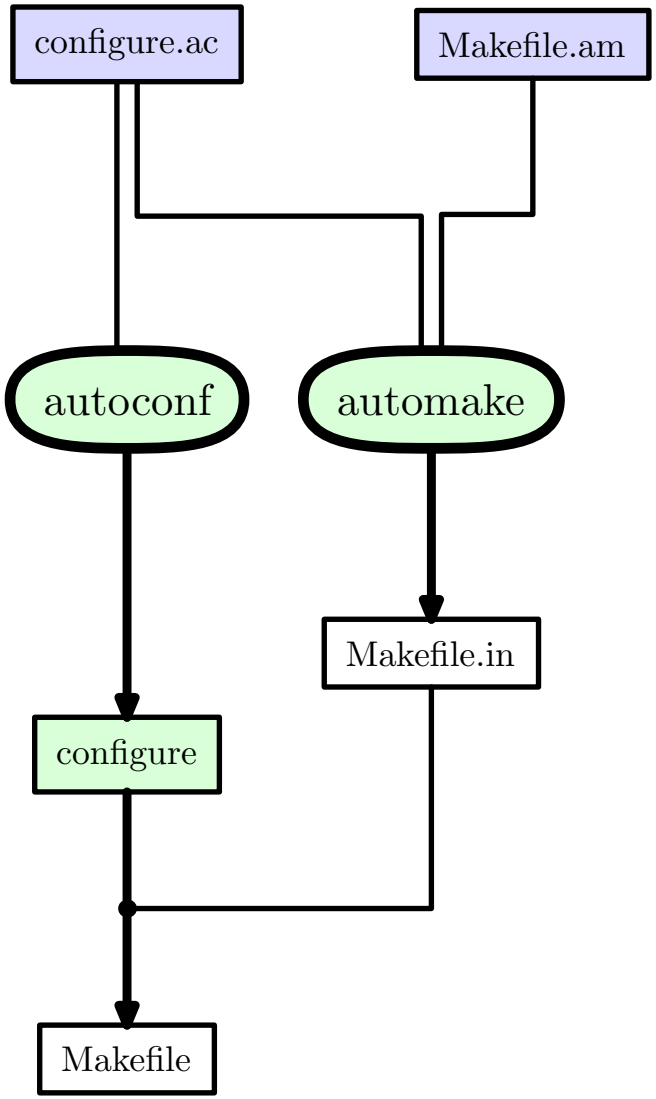
Schema simplifié



Deux fichiers principaux contrôlent le processus.

Jusqu'à la version 2.50, le fichier de configuration d'Autoconf s'appelait *configure.in*.
Aujourd'hui l'appellation privilégiée est *configure.ac*.

Schema simplifié

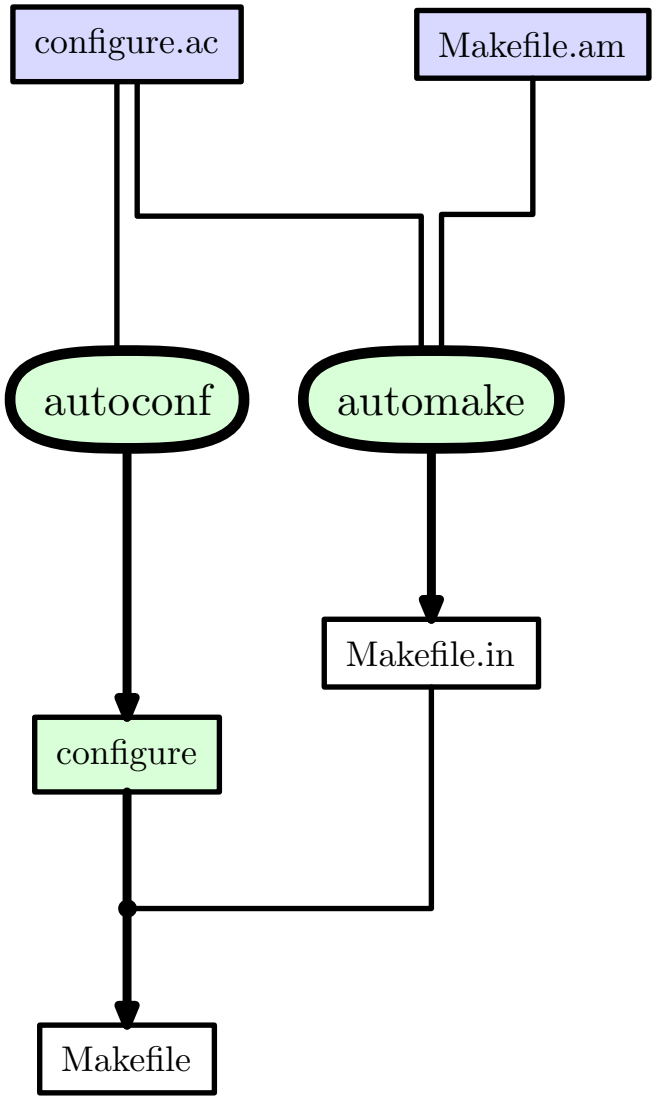


Deux fichiers principaux contrôlent le processus.

Jusqu'à la version 2.50, le fichier de configuration d'Autoconf s'appelait *configure.in*.
Aujourd'hui l'appellation privilégiée est *configure.ac*.

Les autotools proposent aussi **Libtool** pour le développement de bibliothèques partagées.

Schema simplifié



Deux fichiers principaux contrôlent le processus.

Jusqu'à la version 2.50, le fichier de configuration d'Autoconf s'appelait *configure.in*.
Aujourd'hui l'appellation privilégiée est *configure.ac*.

Les autotools proposent aussi **Libtool** pour le développement de bibliothèques partagées.

Automake ne génère pas de *Makefile* mais un *template*.

Programme C : « Bonjour, monde ! »

hello.c

```
#include <stdio.h>
main() {
   printf("Bonjour, monde !\n");
}
```

Programme C : « Bonjour, monde ! »

hello.c

```
#include <stdio.h>
main() {
   printf("Bonjour, monde !\n");
}
```

The GNU build system

Makefile.am

```
bin_PROGRAMS = hello
hello_SOURCES = hello.c
```

configure.ac

```
AC_INIT(Hello world,1.0)
AM_INIT_AUTOMAKE
AC_PROG_CC
AC_PROG_INSTALL
AC_CONFIG_FILES(Makefile)
AC_OUTPUT
```

Programme C : « Bonjour, monde ! »

hello.c

```
#include <stdio.h>
main() {
   printf("Bonjour, monde !\n");
}
```

The GNU build system

Makefile.am

```
bin_PROGRAMS = hello
hello_SOURCES = hello.c
```

configure.ac

```
AC_INIT(Hello world,1.0)
AM_INIT_AUTOMAKE
AC_PROG_CC
AC_PROG_INSTALL
AC_CONFIG_FILES(Makefile)
AC_OUTPUT
```

Programme C : « Bonjour, monde ! »

hello.c

```
#include <stdio.h>
main() {
   printf("Bonjour, monde !\n");
}
```

The GNU build system

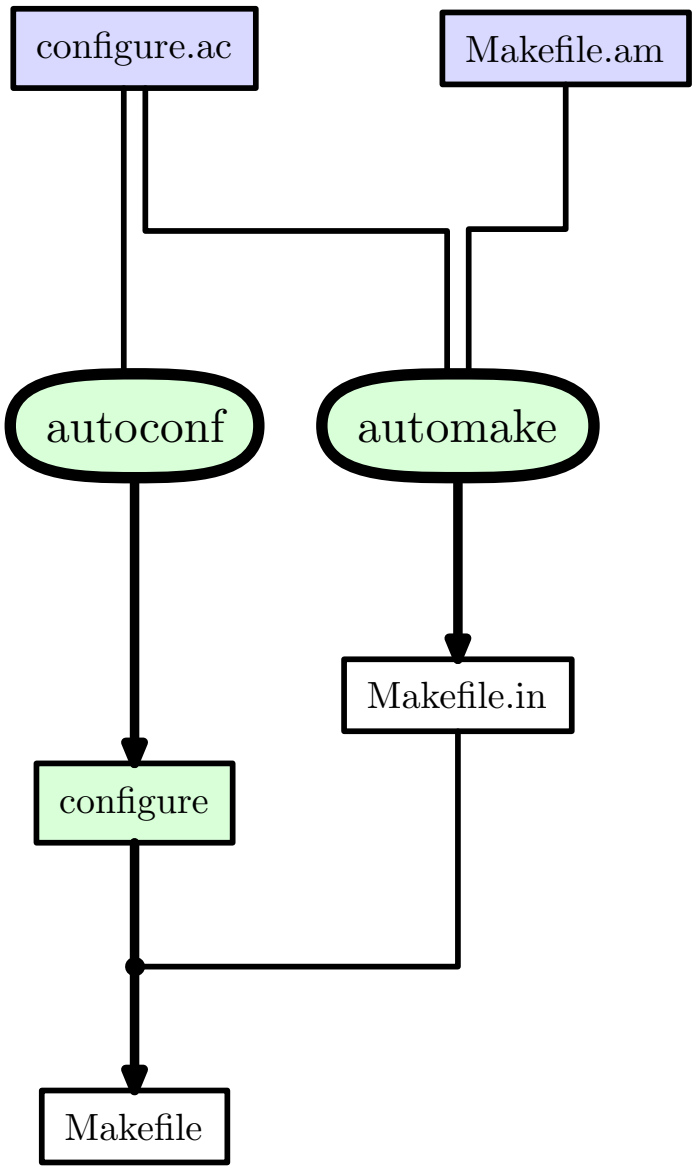
Makefile.am

```
bin_PROGRAMS = hello
hello_SOURCES = hello.c
```

configure.ac

```
AC_INIT(Hello world,1.0)
AM_INIT_AUTOMAKE
AC_PROG_CC
AC_PROG_INSTALL
AC_CONFIG_FILES(Makefile)
AC_OUTPUT
```

Schema simplifié



« Bonjour, monde ! » et les Autotools

```
user@work> aclocal
user@work> autoconf
user@work>
```

« Bonjour, monde ! » et les Autotools

```
user@work> aclocal
user@work> autoconf
user@work> automake
automake: configure.ac: required file ‘./install-sh’ not found
automake: configure.ac: required file ‘./mkinstalldirs’ not found
automake: configure.ac: required file ‘./missing’ not found
automake: configure.ac: required file ‘./config.guess’ not found
automake: configure.ac: required file ‘./config.sub’ not found
automake: Makefile.am: required file ‘./INSTALL’ not found
automake: Makefile.am: required file ‘./NEWS’ not found
automake: Makefile.am: required file ‘./README’ not found
automake: Makefile.am: required file ‘./COPYING’ not found
automake: Makefile.am: required file ‘./AUTHORS’ not found
automake: Makefile.am: required file ‘./ChangeLog’ not found
user@work>
```

« Bonjour, monde ! » et les Autotools

```
user@work> aclocal
user@work> autoconf
user@work> automake -a
automake: configure.ac: installing ‘./install-sh’
automake: configure.ac: installing ‘./mkinstalldirs’
automake: configure.ac: installing ‘./missing’
automake: configure.ac: installing ‘./config.guess’
automake: configure.ac: installing ‘./config.sub’
automake: Makefile.am: installing ‘./INSTALL’
automake: Makefile.am: required file ‘./NEWS’ not found
automake: Makefile.am: required file ‘./README’ not found
automake: Makefile.am: installing ‘./COPYING’
automake: Makefile.am: required file ‘./AUTHORS’ not found
automake: Makefile.am: required file ‘./ChangeLog’ not found
user@work>
```

« Bonjour, monde ! » et les Autotools

```
user@work> aclocal
user@work> autoconf
user@work> automake -a --foreign
automake: configure.ac: installing ‘./install-sh’
automake: configure.ac: installing ‘./mkinstalldirs’
automake: configure.ac: installing ‘./missing’
automake: configure.ac: installing ‘./config.guess’
automake: configure.ac: installing ‘./config.sub’
user@work>
```

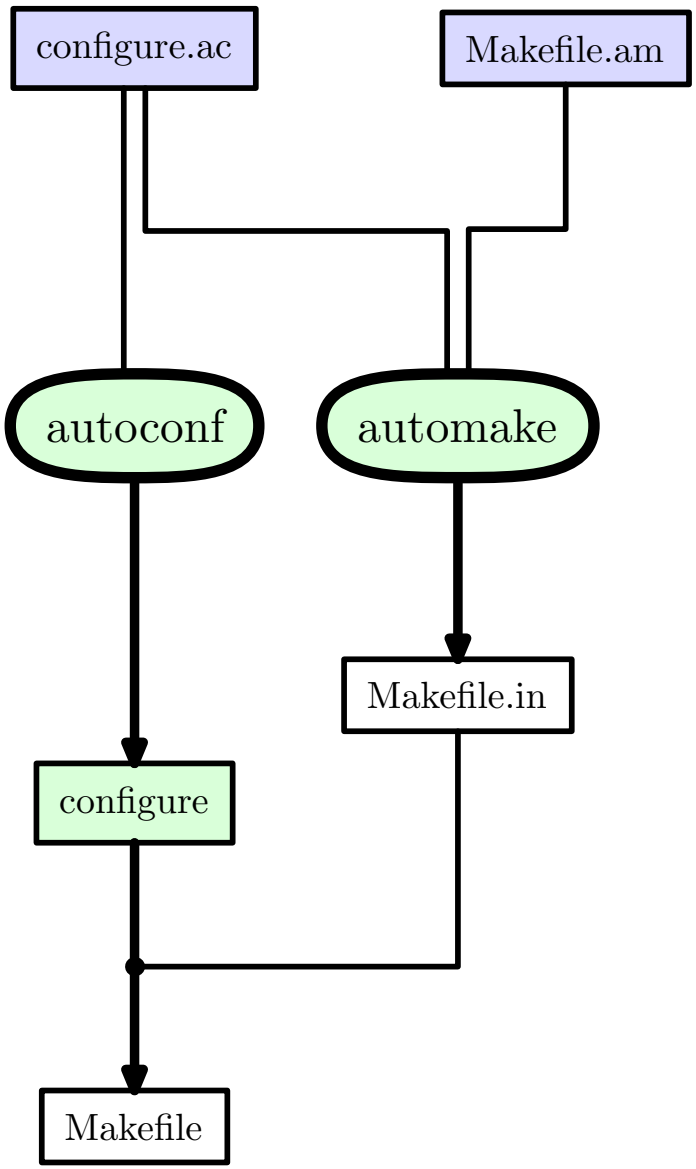
« Bonjour, monde ! » et les Autotools

```
user@work> aclocal
user@work> autoconf
user@work> automake -a --foreign
automake: configure.ac: installing `./install-sh'
automake: configure.ac: installing `./mkinstalldirs'
automake: configure.ac: installing `./missing'
automake: configure.ac: installing `./config.guess'
automake: configure.ac: installing `./config.sub'
user@work> ./configure
checking for a BSD-compatible install... /usr/bin/install -c
checking whether build environment is sane... yes
checking whether make sets $(MAKE)... yes
checking for working aclocal-1.4... found
checking for working autoconf... found
 [...]
user@work>
```

« Bonjour, monde ! » et les Autotools

```
user@work> aclocal
user@work> autoconf
user@work> automake -a --foreign
automake: configure.ac: installing ‘./install-sh’
automake: configure.ac: installing ‘./mkinstalldirs’
automake: configure.ac: installing ‘./missing’
automake: configure.ac: installing ‘./config.guess’
automake: configure.ac: installing ‘./config.sub’
user@work> ./configure
checking for a BSD-compatible install... /usr/bin/install -c
checking whether build environment is sane... yes
checking whether make sets $(MAKE)... yes
checking for working aclocal-1.4... found
checking for working autoconf... found
 [...]
user@work> make && make install
 [...]
```

Schema simplifié



« Bonjour, monde ! » expliqué

Makefile.am

```
bin_PROGRAMS = hello
hello_SOURCES = hello.c
```

bin_PROGRAMS donne la liste des exécutables du projet.

*_SOURCES liste les fichiers sources nécessaires à la construction.

« Bonjour, monde ! » expliqué

Makefile.am

```
bin_PROGRAMS = hello
hello_SOURCES = hello.c
```

bin_PROGRAMS donne la liste des exécutables du projet.

*_SOURCES liste les fichiers sources nécessaires à la construction.

« Bonjour, monde ! » expliqué

Makefile.am

```
bin_PROGRAMS = hello
hello_SOURCES = hello.c
```

bin_PROGRAMS donne la liste des exécutables du projet.

*_SOURCES liste les fichiers sources nécessaires à la construction.

« Bonjour, monde ! » expliqué

Makefile.am

```
bin_PROGRAMS = hello
hello_SOURCES = hello.c
```

bin_PROGRAMS donne la liste des exécutables du projet.

*_SOURCES liste les fichiers sources nécessaires à la construction.

Automake calcule alors les dépendances pour créer un template de *Makefile*.

« Bonjour, monde ! »: explications

configure.ac

```
AC_INIT(Hello world,1.0)
AM_INIT_AUTOMAKE
AC_PROG_CC
AC_PROG_INSTALL
AC_CONFIG_FILES(Makefile)
AC_OUTPUT
```

AC_INIT initialise Autoconf avec le nom et la version du projet.

AM_INIT_AUTOMAKE permet la collaboration Autoconf/Automake.

AC_PROG_CC recherche et identifie le compilateur C.

AC_PROG_INSTALL cherche l'utilitaire *install*.

AC_CONFIG_FILES donne l'ordre à *configure* de générer *Makefile* à partir de *Makefile.in*.

AC_OUTPUT indique la fin des directives Autoconf.

« Bonjour, monde ! »: explications

configure.ac

```
AC_INIT(Hello world,1.0)
AM_INIT_AUTOMAKE
AC_PROG_CC
AC_PROG_INSTALL
AC_CONFIG_FILES(Makefile)
AC_OUTPUT
```

AC_INIT initialise Autoconf avec le nom et la version du projet.
AM_INIT_AUTOMAKE permet la collaboration Autoconf/Automake.
AC_PROG_CC recherche et identifie le compilateur C.
AC_PROG_INSTALL cherche l'utilitaire *install*.
AC_CONFIG_FILES donne l'ordre à *configure* de générer *Makefile* à partir de *Makefile.in*.
AC_OUTPUT indique la fin des directives Autoconf.

« Bonjour, monde ! »: explications

configure.ac

```
AC_INIT(Hello world,1.0)
AM_INIT_AUTOMAKE
AC_PROG_CC
AC_PROG_INSTALL
AC_CONFIG_FILES(Makefile)
AC_OUTPUT
```

AC_INIT initialise Autoconf avec le nom et la version du projet.
AM_INIT_AUTOMAKE permet la collaboration Autoconf/Automake.
AC_PROG_CC recherche et identifie le compilateur C.
AC_PROG_INSTALL cherche l'utilitaire *install*.
AC_CONFIG_FILES donne l'ordre à *configure* de générer *Makefile* à partir de *Makefile.in*.
AC_OUTPUT indique la fin des directives Autoconf.

« Bonjour, monde ! »: explications

configure.ac

```
AC_INIT(Hello world,1.0)
AM_INIT_AUTOMAKE
AC_PROG_CC
AC_PROG_INSTALL
AC_CONFIG_FILES(Makefile)
AC_OUTPUT
```

AC_INIT initialise Autoconf avec le nom et la version du projet.
AM_INIT_AUTOMAKE permet la collaboration Autoconf/Automake.
AC_PROG_CC recherche et identifie le compilateur C.
AC_PROG_INSTALL cherche l'utilitaire *install*.
AC_CONFIG_FILES donne l'ordre à *configure* de générer *Makefile* à partir de *Makefile.in*.
AC_OUTPUT indique la fin des directives Autoconf.

« Bonjour, monde ! »: explications

configure.ac

```
AC_INIT(Hello world,1.0)
AM_INIT_AUTOMAKE
AC_PROG_CC
AC_PROG_INSTALL
AC_CONFIG_FILES(Makefile)
AC_OUTPUT
```

AC_INIT initialise Autoconf avec le nom et la version du projet.
AM_INIT_AUTOMAKE permet la collaboration Autoconf/Automake.
AC_PROG_CC recherche et identifie le compilateur C.
AC_PROG_INSTALL cherche l'utilitaire *install*.
AC_CONFIG_FILES donne l'ordre à *configure* de générer *Makefile* à partir de *Makefile.in*.
AC_OUTPUT indique la fin des directives Autoconf.

« Bonjour, monde ! »: explications

configure.ac

```
AC_INIT(Hello world,1.0)
AM_INIT_AUTOMAKE
AC_PROG_CC
AC_PROG_INSTALL
AC_CONFIG_FILES(Makefile)
AC_OUTPUT
```

AC_INIT initialise Autoconf avec le nom et la version du projet.
AM_INIT_AUTOMAKE permet la collaboration Autoconf/Automake.
AC_PROG_CC recherche et identifie le compilateur C.
AC_PROG_INSTALL cherche l'utilitaire *install*.
AC_CONFIG_FILES donne l'ordre à *configure* de générer *Makefile* à partir de *Makefile.in*.
AC_OUTPUT indique la fin des directives Autoconf.

« Bonjour, monde ! »: résumé

Fichiers sources (*hello.c*)

Makefile.am

```
bin_PROGRAMS = hello
hello_SOURCES = hello.c
```

configure.ac

```
AC_INIT(Hello world,1.0)
AM_INIT_AUTOMAKE
AC_PROG_CC
AC_PROG_INSTALL
AC_CONFIG_FILES(Makefile)
AC_OUTPUT
```

« Bonjour, monde ! »: résumé

Fichiers sources (*hello.c*)

Makefile.am

```
bin_PROGRAMS = hello
hello_SOURCES = hello.c
```

configure.ac

```
AC_INIT(Hello world,1.0)
AM_INIT_AUTOMAKE
AC_PROG_CC
AC_PROG_INSTALL
AC_CONFIG_FILES(Makefile)
AC_OUTPUT
```

Construction :

« Bonjour, monde ! »: résumé

Fichiers sources (*hello.c*)

Makefile.am

```
bin_PROGRAMS = hello
hello_SOURCES = hello.c
```

configure.ac

```
AC_INIT(Hello world,1.0)
AM_INIT_AUTOMAKE
AC_PROG_CC
AC_PROG_INSTALL
AC_CONFIG_FILES(Makefile)
AC_OUTPUT
```

Construction :

- Première phase : *aclocal* — *autoconf* — *automake*

« Bonjour, monde ! »: résumé

Fichiers sources (*hello.c*)

Makefile.am

```
bin_PROGRAMS = hello
hello_SOURCES = hello.c
```

configure.ac

```
AC_INIT(Hello world,1.0)
AM_INIT_AUTOMAKE
AC_PROG_CC
AC_PROG_INSTALL
AC_CONFIG_FILES(Makefile)
AC_OUTPUT
```

Construction :

- Première phase : *aclocal* — *autoconf* — *automake*
- Seconde phase : *configure* — *make*

Autoconf/Automake **en pratique**

The GNU build system

vskip*-0.2cm

Le fichier principal : *configure.ac*

Contient du code *Bourne shell* et du code m4 :

- `AC_XXX` directives Autoconf
- `AM_XXX` directives Automake
- `AH_XXX`, `AT_XXX` ...

The GNU build system

vskip*-0.2cm

Le fichier principal : *configure.ac*

Contient du code *Bourne shell* et du code m4 :

- `AC_XXX` directives Autoconf
- `AM_XXX` directives Automake
- `AH_XXX`, `AT_XXX` ...

Structure de *configure.ac* *(extraite du manuel Autoconf)*

`AC_INIT`(package, version, bug@report)
information on the package
checks for programs
checks for libraries
checks for header files
checks for types and structures
checks for compiler characteristics
checks for library functions
checks for system services
`AC_CONFIG_FILES`([file...])
`AC_OUTPUT`

The GNU build system

vskip*-0.2cm

Le fichier principal : *configure.ac*

Contient du code *Bourne shell* et du code m4 :

- `AC_XXX` directives Autoconf
- `AM_XXX` directives Automake
- `AH_XXX`, `AT_XXX` ...

Structure de *configure.ac* *(extraite du manuel Autoconf)*

`AC_INIT`(package, version, bug@report) → AC_INIT(Hello world,1.0)
information on the package → AM_INIT_AUTOMAKE
checks for programs → AC_PROG_CC
checks for libraries → AC_PROG_INSTALL
checks for header files → AC_CONFIG_FILES(Makefile)
checks for types and structures → AC_OUTPUT
checks for compiler characteristics
checks for library functions
checks for system services
`AC_CONFIG_FILES`([file...])
`AC_OUTPUT`

The GNU build system

Les pré-*templates* de *Makefile* : *Makefile.am*

- Directives Automake :
  AUTOMAKE_OPTIONS = dist-zip foreign
  bin_PROGRAMS = ...
  hello_SOURCES = ...
  hello_LDADD = ...
  hello_CPPFLAGS = ...
- Texte inclus verbatim :
  cvs-local:
  for i in * ; do ...
- Variables évaluées par *configure* :
  libalsa09_la_LDFLAGS = @PLUGIN_LDFLAGS@

The GNU build system

Les pré-*templates* de *Makefile* : *Makefile.am*

- Directives Automake :
  AUTOMAKE_OPTIONS = dist-zip foreign
  bin_PROGRAMS = ...
  hello_SOURCES = ...
  hello_LDADD = ...
  hello_CPPFLAGS = ...
- Texte inclus verbatim :
  cvs-local:
  for i in *; do ...
- Variables évaluées par *configure* :
  libalsa09_la_LDFLAGS = @PLUGIN_LDFLAGS@

The GNU build system

Les pré-*templates* de *Makefile* : *Makefile.am*

- Directives Automake :
  AUTOMAKE_OPTIONS = dist-zip foreign
  bin_PROGRAMS = ...
  hello_SOURCES = ...
  hello_LDADD = ...
  hello_CPPFLAGS = ...
- Texte inclus verbatim :
  cvs-local:
  for i in *; do ...
- Variables évaluées par *configure* :
  libalsa09_la_LDFLAGS = @PLUGIN_LDFLAGS@

Organisation du projet

Séparation en sous-répertoires

Au minimum, utiliser doc/ et src/.

Organisation du projet

Séparation en sous-répertoires

Au minimum, utiliser doc/ et src/.

La prise en charge par les Autotools est simple.

- *configure.ac* : (unique)
  `AC_CONFIG_FILES([Makefile doc/Makefile src/Makefile])`
- *Makefile.am* dans le répertoire racine :
  `SUBDIRS = doc src`
- Un *Makefile.am* par sous-répertoire cible.
  doc/Makefile.am src/Makefile.am

Organisation du projet

Séparation en sous-répertoires

Au minimum, utiliser doc/ et src/.

La prise en charge par les Autotools est simple.

- *configure.ac* : (unique)
  `AC_CONFIG_FILES([Makefile doc/Makefile src/Makefile])`
- *Makefile.am* dans le répertoire racine :
  `SUBDIRS = doc src`
- Un *Makefile.am* par sous-répertoire cible.
  doc/Makefile.am src/Makefile.am

Une option utile :
`AC_CONFIG_AUX_DIR(admin)` utilise admin/ au lieu de ./ pour la plupart des fichiers Autotools.

La distribution

Paquet source

La création d'un paquet à partir des sources est immédiate :

```
make dist
```

produit un *tarball* appelé *package-version.tar.gz*.

On peut rajouter en ligne de commande ou dans *Makefile.am*

```
AUTOMAKE_OPTIONS = dist-bzip2 dist-zip
```

Ce qui définira ces cibles pour *make*.

La distribution

Paquet source

La création d'un paquet à partir des sources est immédiate :

```
make dist
```

produit un *tarball* appelé *package-version.tar.gz*.

On peut rajouter en ligne de commande ou dans *Makefile.am*

```
AUTOMAKE_OPTIONS = dist-bzip2 dist-zip
```

Ce qui définira ces cibles pour *make*.

Paquet debian

Grâce à la normalisation des Autotools, le processus est court.

- deb-make
- Éditer les fichiers modèles du répertoire debian (description, mainteneur...)
  Vérifier les dépendances (surtout concernant les sources)
- dpkg-buildpackage -rfakeroot

Outils annexes

Création de projet

Deux outils créent un squelette de nouveau projet :

- *autoproject* (interactif)
- *acmkdir*

Outils annexes

Création de projet

Deux outils créent un squelette de nouveau projet :

- *autoproject* (interactif)
- *acmkdir*

Entretien du projet

- *autoscan* parcourt les répertoires récursivement et génère *configure.scan*.
  - Si le projet est autoconfisqué, les recommandations de *configure.scan* sont souvent à intégrer dans *configure.ac* après vérification de *autoscan.log*.
  - Sinon, le fichier *configure.scan* peut être renommé en *configure.ac* après quelques modifications.

Outils annexes

Création de projet

Deux outils créent un squelette de nouveau projet :

- *autoproject* (interactif)
- *acmkdir*

Entretien du projet

- *autoscan* parcourt les répertoires récursivement et génère *configure.scan*.
- *autoreconf* lance les différents programmes (*aclocal*, *autoheader*...) pour obtenir le script *configure*.

Outils annexes

Création de projet

Deux outils créent un squelette de nouveau projet :

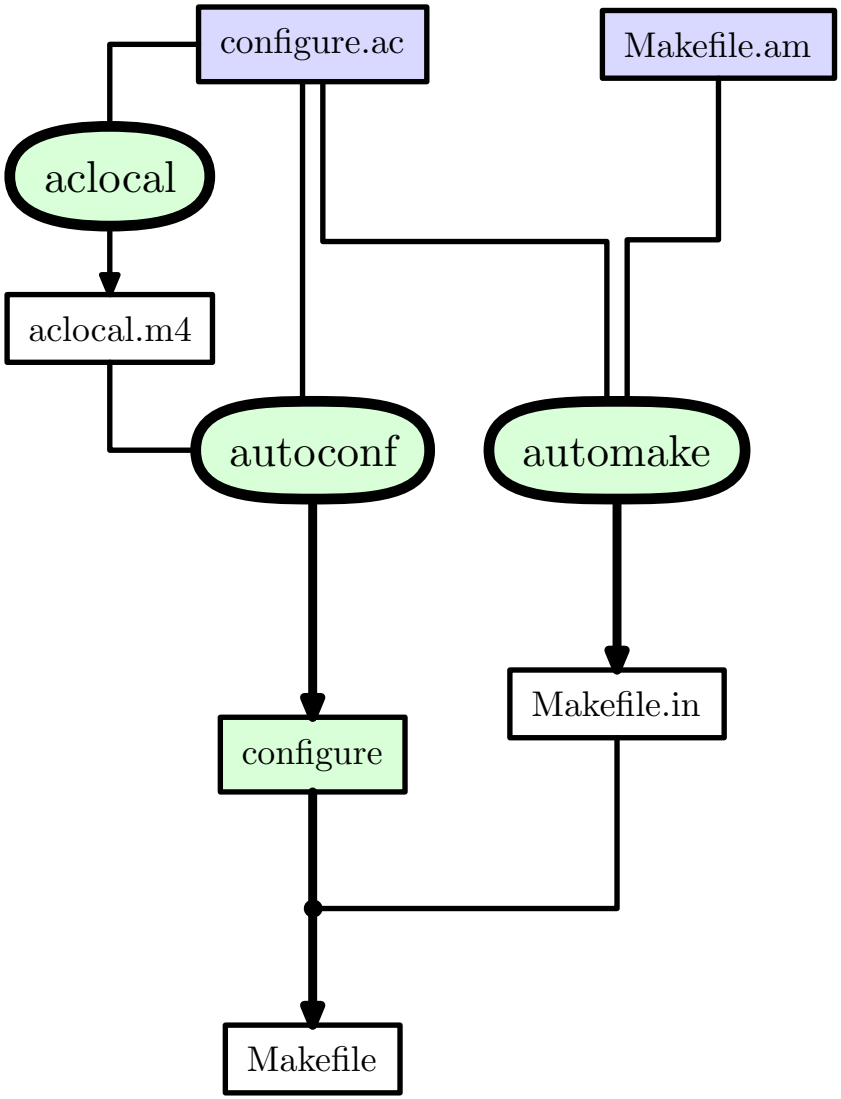
- *autoproject* (interactif)
- *acmkdir*

Entretien du projet

- *autoscan* parcourt les répertoires récursivement et génère *configure.scan*.
- *autoreconf* lance les différents programmes (*aclocal*, *autoheader*...) pour obtenir le script *configure*.
- L'automatic remaking est désormais activé par défaut par autoconf. Si un des fichiers d'Autoconf/Automake est modifié, *make* relancera les commandes adéquates.

Fonctionnement avancé

Diagramme détaillé



Pourquoi des macros locales ?

Un problème fréquent :

Autoconf n'a pas de macro pour tester la librairie souhaitée. . .

Pourquoi des macros locales ?

Un problème fréquent :

Autoconf n'a pas de macro pour tester la librairie souhaitée. . .

La solution :

Rajouter une macro à Autoconf.

acinclude.m4 contient les macros personnelles.

Pourquoi des macros locales ?

Un problème fréquent :

Autoconf n'a pas de macro pour tester la librairie souhaitée. . .

La solution :

Rajouter une macro à Autoconf.

acinclude.m4 contient les macros personnelles.

La méthode :

- Utiliser l'Autoconf macro archive.
  `http://www.gnu.org/software/ac-archive/`
- Programmer sa propre macro en m4.

Pourquoi des macros locales ?

Un problème fréquent :

Autoconf n'a pas de macro pour tester la librairie souhaitée. . .

La solution :

Rajouter une macro à Autoconf.

acinclude.m4 contient les macros personnelles.

La méthode :

- Utiliser l'Autoconf macro archive.
  http://www.gnu.org/software/ac-archive/
- Programmer sa propre macro en m4.

Exemple de macro locale

acinclude.m4

```
dnl MYLIB_PATH([ACTION-IF-FOUND [, ACTION-IF-NOT-FOUND]])
dnl
AC_DEFUN(MYLIB_PATH,
[
  AC_ARG_WITH(mylib-prefix,
   [  --with-mylib-prefix=PFX    Prefix where mylib is installed],
   mylib_prefix="$withval",mylib_prefix="")

  AC_TRY_RUN([...programme à tester...],, no_mylib=yes,)

  if test "x$no_mylib" = "!" ; then
     AC_MSG_RESULT(no)
     echo "La librairie mylib ne semble pas installée"
  fi
])
```

Exemple de macro locale

configure.ac

```
AC_INIT(myprog,0.1,[report@bugs.org])
 [...]
MYLIB_PATH()
 [...]
AC_OUTPUT
```

Exemple de macro locale

configure.ac

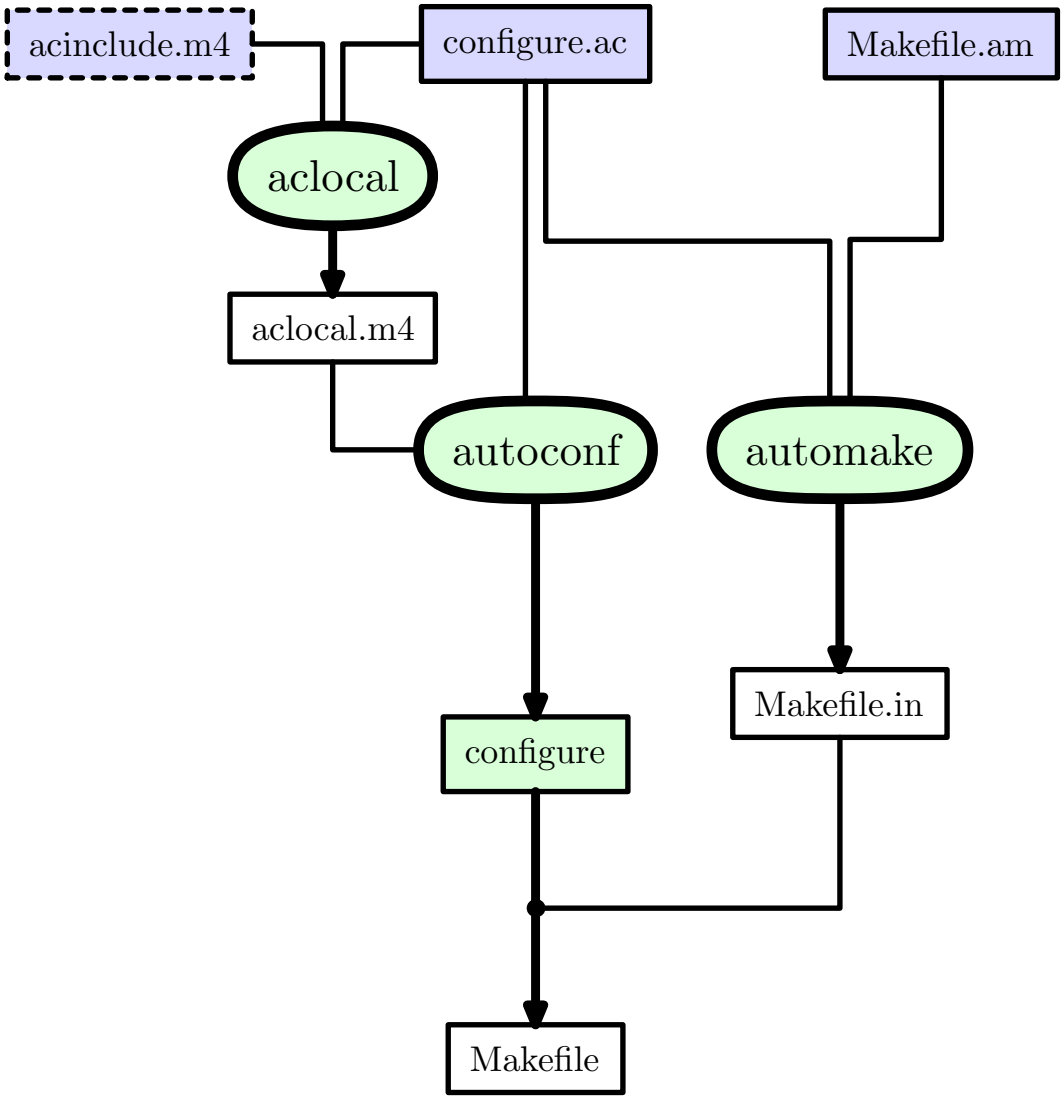
```
AC_INIT(myprog,0.1,[report@bugs.org])
 [...]
MYLIB_PATH()
 [...]
AC_OUTPUT
```

On lance alors *aclocal* qui :

- lit *acinclude.m4*
- génère *aclocal.m4*

Autoconf prendra en compte le *aclocal.m4* ainsi créé.

Le schema (toujours partiel) des autotools



Pour un code souple : config.h

config.h

C'est un fichier de macros pour le préprocesseur C/C++,

Il n'est pas construit par défaut :

- Ajouter `AM_CONFIG_HEADER(config.h)` à *configure.ac*.

Pour un code souple : config.h

config.h

C'est un fichier de macros pour le préprocesseur C/C++,

Il n'est pas construit par défaut :

- Ajouter `AM_CONFIG_HEADER(config.h)` à *configure.ac*.
- Lancer *autoheader* avant Automake.

Pour un code souple : config.h

config.h

C'est un fichier de macros pour le préprocesseur C/C++,

Il n'est pas construit par défaut :

- Ajouter `AM_CONFIG_HEADER(config.h)` à *configure.ac*.
- Lancer *autoheader* avant Automake.

Exemple de *config.h*

```
/* config.h.  Generated by configure.  */
/* config.h.in.  Generated from configure.in by autoheader.  */

/* Define to 1 if you have the <stdint.h> header file. */
#define HAVE_STDINT_H 1

/* Name of package */
#define PACKAGE "project"

/* Version number of package */
#define VERSION "0.1.0"
```

Pour un code souple : config.h

config.h

C'est un fichier de macros pour le préprocesseur C/C++,

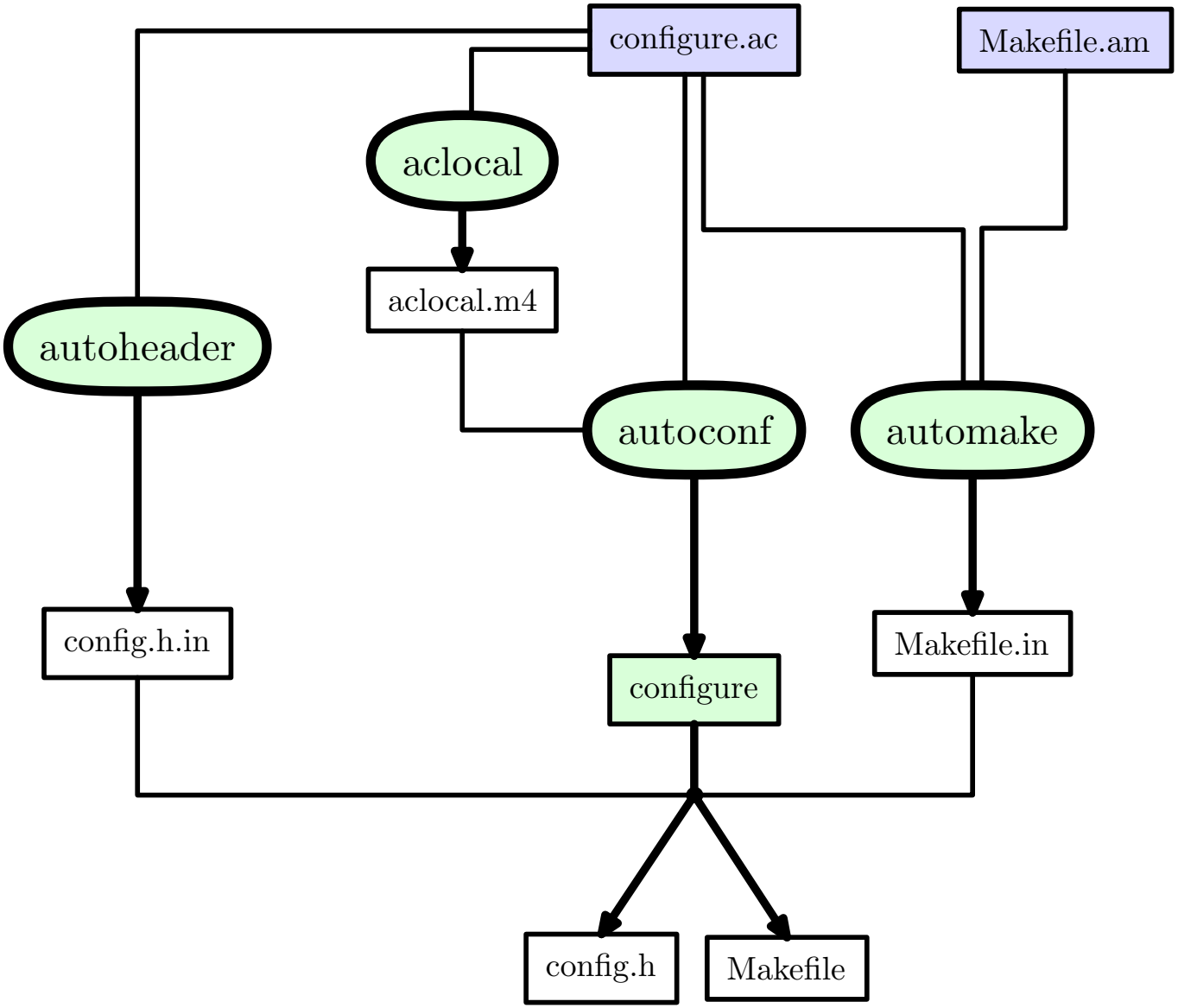
Il n'est pas construit par défaut :

- Ajouter `AM_CONFIG_HEADER(config.h)` à *configure.ac*.
- Lancer *autoheader* avant Automake.

Intérêt

Adapter le code compilé aux tests définis par *configure.ac* et effectués dans *configure*.

Un schema assez complet



Le script configure

- Généré par Autoconf.
- Destiné à être diffusé.
  - Script en Bourne shell.
  - Ne fait pas appel aux Autotools à l'exécution.
  - Ne dépend que des outils de base : sed, egrep ...
- Configurable en ligne de commande.

Le script configure

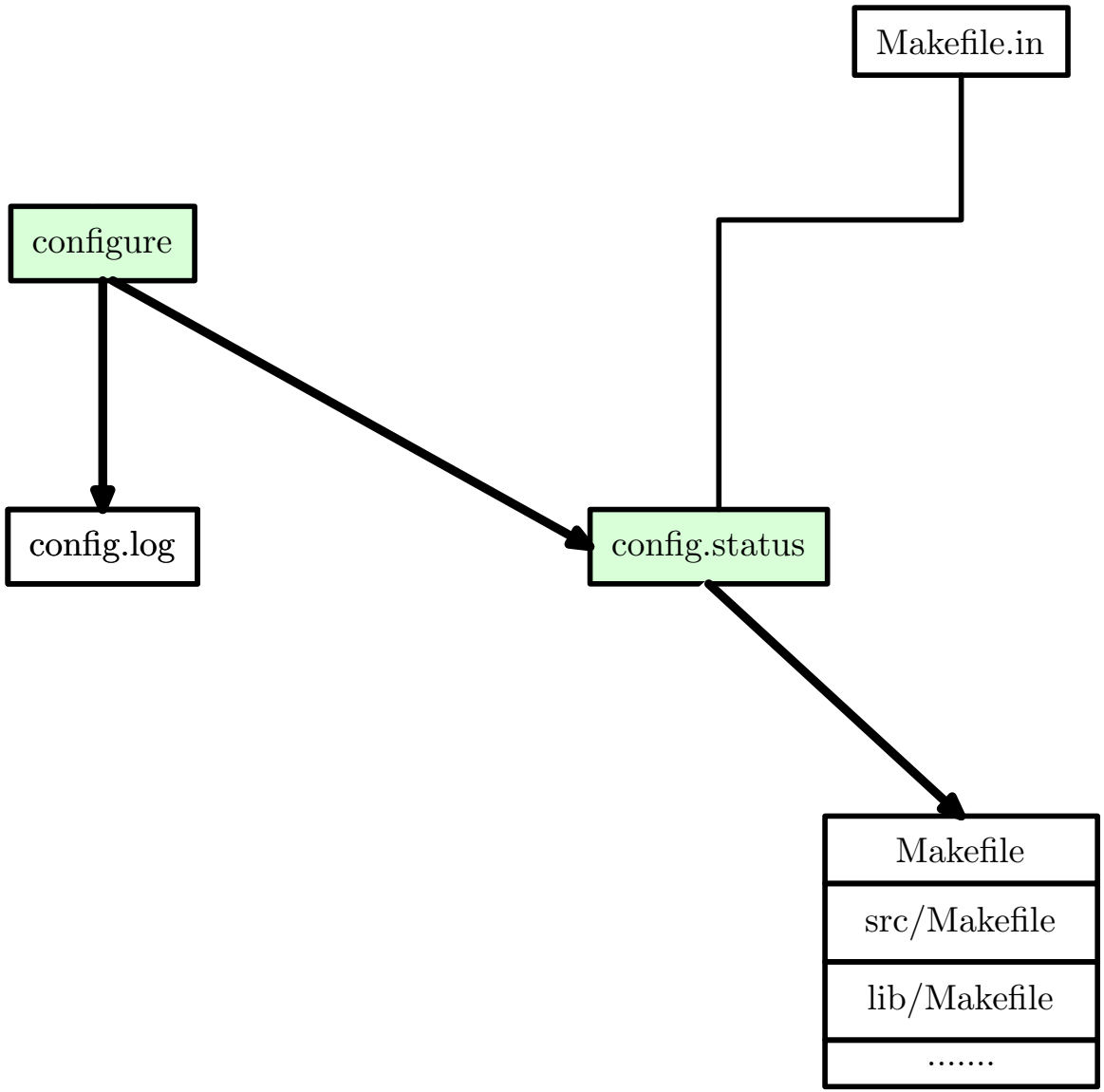
- Généré par Autoconf.
- Destiné à être diffusé.
  - Script en Bourne shell.
  - Ne fait pas appel aux Autotools à l'exécution.
  - Ne dépend que des outils de base : sed, egrep ...
- Configurable en ligne de commande.

Fonctionnement

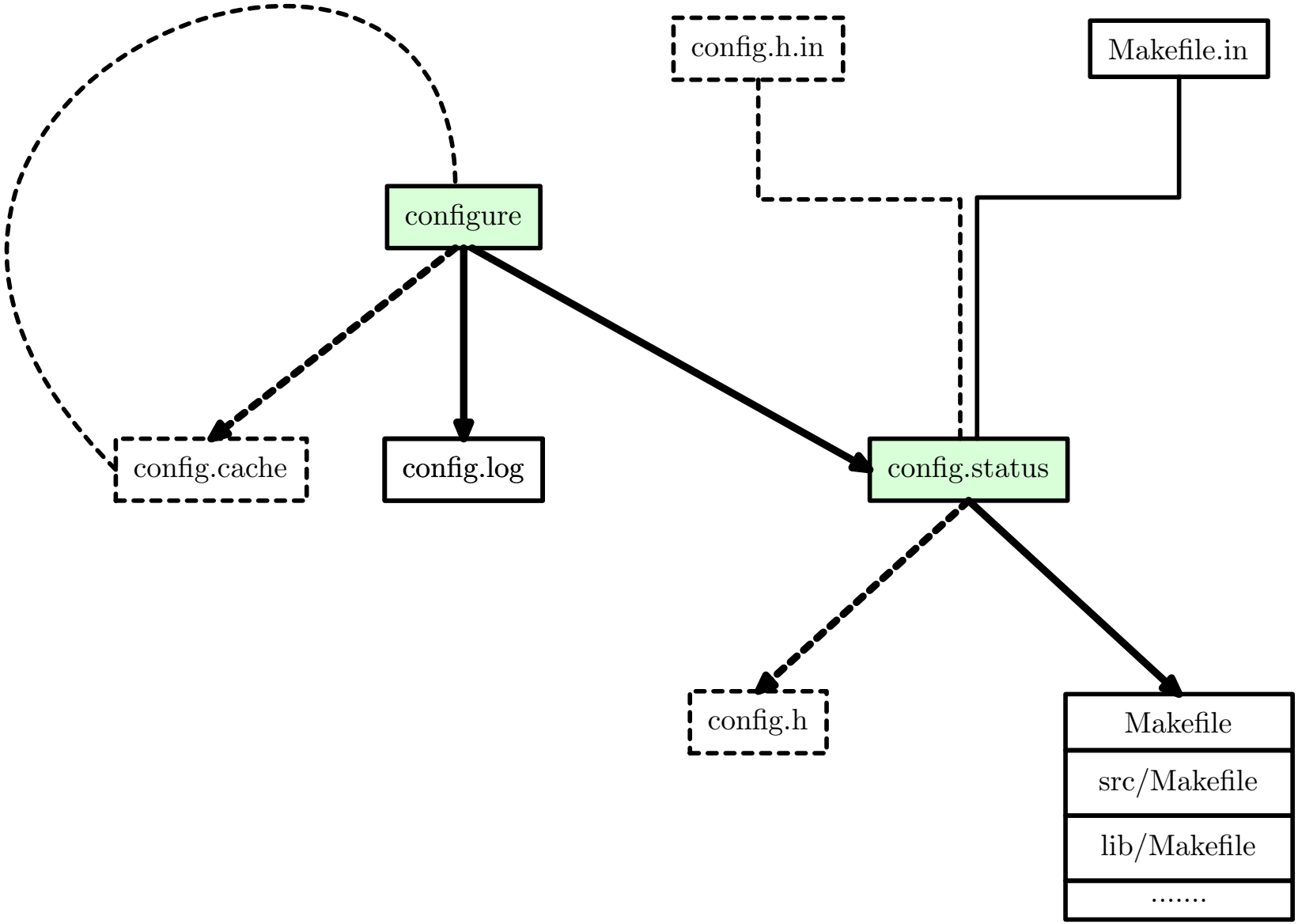
configure crée un fichier *config.status* à partir des paramètres reçus puis l'exécute.

config.status remplace les champs @VARIABLE@ dans les fichiers *xxx.in* désignés pour créer des fichiers *xxx*.

Fonctionnement de configure



Fonctionnement de configure



Conclusion

The GNU build system

Make seul est insatisfaisant dès que le projet devient complexe.
Autoconf/Automake en est le complément et le prolongement.

The GNU build system

Make seul est insatisfaisant dès que le projet devient complexe.
Autoconf/Automake en est le complément et le prolongement.

Le système de développement GNU est :

- Portable sur les UNIX, Linux, Cygwin.
- Normalisé pour le développeur, l'utilisateur final et les outils de distribution (en particulier, la création de paquets pour une distribution).
- Extensible par la programmation de macros m4.
- Configurable par l'utilisateur.
- Simplifié par la rédaction de *Makefile.am* au lieu de *Makefile*. Les dépendances sont calculées automatiquement.

Reproches au GNU build system

Pour le développeur :

- Le processus est multi-couches et complexe.
- Maîtriser Autoconf nécessite de savoir programmer en m4.
- Problèmes de compatibilité d'une version à l'autre...

Reproches au GNU build system

Pour le développeur :

- Le processus est multi-couches et complexe.
- Maîtriser Autoconf nécessite de savoir programmer en m4.
- Problèmes de compatibilité d'une version à l'autre...

Plus généralement :

- Le *Makefile* produit est humainement illisible.
- La portabilité est parfois difficile sous Windows (Cygwin).
- Les limitations inhérentes à *make* demeurent.

Autres approches

Tentatives sans lendemain :

Imake : Make en plus compliqué ! (X11)

Metaconfig : Principe similaire à Autoconf. (Perl)

Autres approches

Tentatives sans lendemain :

Imake : Make en plus compliqué ! (X11)

Metaconfig : Principe similaire à Autoconf. (Perl)

Tentatives en cours :

Cons : Gère l'intégralité du processus de construction, sans *make*. Basé sur Perl.

SCons : Même principe que Cons. Basé sur Python.

Références bibliographiques

Ouvrages :

- L'autobook : **Autoconf, Automake, Libtool**.
  http://sources.redhat.com/autobook/
- **Learning the GNU development tools**.
  http://autotoolset.sourceforge.net/tutorial.html
- **Les manuels autoconf et automake**.

Tutoriaux :

- **Autotut: using GNU auto{conf,make,header}**
  http://seul.org/docs/autotut/
- **Using Automake and Autoconf with C++**
  http://www.openismus.com/documents/linux/automake/

Références bibliographiques

Ouvrages :

- L'autobook : **Autoconf, Automake, Libtool**.
  http://sources.redhat.com/autobook/
- **Learning the GNU development tools**.
  http://autotoolset.sourceforge.net/tutorial.html
- Les manuels autoconf et automake.

Tutoriaux :

- Autotut: using GNU auto{conf,make,header}
  http://seul.org/docs/autotut/
- Using Automake and Autoconf with C++
  http://www.openismus.com/documents/linux/automake/

Références bibliographiques

Ouvrages :

- L'autobook : **Autoconf, Automake, Libtool**.
  http://sources.redhat.com/autobook/
- **Learning the GNU development tools**.
  http://autotoolset.sourceforge.net/tutorial.html
- **Les manuels autoconf et automake**.

Tutoriaux :

- **Autotut: using GNU auto{conf,make,header}**
  http://seul.org/docs/autotut/
- **Using Automake and Autoconf with C++**
  http://www.openismus.com/documents/linux/automake/

Références bibliographiques

Ouvrages :

- L'autobook : **Autoconf, Automake, Libtool**.
  http://sources.redhat.com/autobook/
- **Learning the GNU development tools**.
  http://autotoolset.sourceforge.net/tutorial.html
- **Les manuels autoconf et automake**.

Tutoriaux :

- **Autotut: using GNU auto{conf,make,header}**
  http://seul.org/docs/autotut/
- **Using Automake and Autoconf with C++**
  http://www.openismus.com/documents/linux/automake/

Références bibliographiques

Ouvrages :

- L'autobook : **Autoconf, Automake, Libtool**.
  http://sources.redhat.com/autobook/
- **Learning the GNU development tools**.
  http://autotoolset.sourceforge.net/tutorial.html
- **Les manuels autoconf et automake**.

Tutoriaux :

- **Autotut: using GNU auto{conf,make,header}**
  http://seul.org/docs/autotut/
- **Using Automake and Autoconf with C++**
  http://www.openismus.com/documents/linux/automake/

L'épigraphe au manuel Autoconf

A physicist, an engineer, and a computer scientist were discussing the nature of God.
"Surely a Physicist," said the physicist, "because early in the Creation, God made Light ; and you know, Maxwell's equations, the dual nature of electromagnetic waves, the relativistic consequences..."
"An Engineer!," said the engineer, "because before making Light, God split the Chaos into Land and Water ; it takes a hell of an engineer to handle that big amount of mud, and orderly separation of solids from liquids..."
The computer scientist shouted: "And the Chaos, where do you think it was coming from, hmm ?"

– Anonymous

Autoheader: le générateur de config.h

Autoheader crée un *template* de fichier de définitions qui sera ensuite adapté par *configure*.

Par défaut :

En entrée : *configure.ac*

En sortie : *config.h.in*

Autoheader: le générateur de config.h

Autoheader crée un *template* de fichier de définitions qui sera ensuite adapté par *configure*.

Par défaut :

En entrée : *configure.ac*

En sortie : *config.h.in*

Exemple simple

Avec la commande suivante dans *configure.ac* :

```
AH_TEMPLATE(HAVE_MYLIB,[Defined if we can use mylib.])
```

configure produira dans *config.h* :

```
/* Defined if we can use mylib. */
/* #undef HAVE_MYLIB */
```

Autoheader: le générateur de config.h

Autoheader crée un *template* de fichier de définitions qui sera ensuite adapté par *configure*.

Par défaut :

En entrée : *configure.ac*

En sortie : *config.h.in*

Exemple simple

Avec la commande suivante dans *configure.ac* :

```
AH_TEMPLATE(HAVE_MYLIB,[Defined if we can use mylib.])
AC_DEFINE(HAVE_MYLIB)
```

configure produira dans *config.h* :

```
/* Defined if we can use mylib. */
#define HAVE_MYLIB 1
```

Exemple de projet avec *config.h*

configure.ac

```
AC_INIT(Hello world,1.1)
AM_CONFIG_HEADER(config.h)
AM_INIT_AUTOMAKE
AC_PROG_CC
AC_PROG_CXX
AC_PATH_PROG(PERL_PATH,perl)
AC_DEFINE_UNQUOTED(PERL_PATH,"$PERL_PATH",
        [Chemin où trouver l'exécutable de Perl])
AC_PROG_INSTALL
AC_OUTPUT
```

Exemple de projet avec *config.h*

config.h

```
/* config.h.  Generated by configure.  */
/* config.h.in.  Generated from configure.ac by autoheader.  */

/* Name of package */
#define PACKAGE "hello"

/* Chemin où trouver l'exécutable de Perl */
#define PERL_PATH "/usr/bin/perl"

/* Version number of package */
#define VERSION "1.1"
```

Exemple de projet avec *config.h*

hello.cc

```
#ifdef HAVE_CONFIG_H
#  include "config.h"
#endif

#include <iostream>
using namespace std;

main() {
  cout << "-=< " << PACKAGE;
  cout << " v" << VERSION << " >=-";
#ifdef PERL_PATH
  cout << " dit bonjour à " << PERL_PATH;
#endif
  cout << "\n";;
}
```