slackware® linux

An introduction to Slackware packages

Stuart Winter

Sunday 27th Februrary
FOSDEM 2005

stuart@armedslack.org

Introduction: Slackware Linux

- First released in 1993
- Slackware aims to be the most “UNIX-like” Linux distribution
- Simplicity and stability are of paramount importance
- Patrick Volkerding is the project leader & sole maintainer

Presentation agenda

- Introduce the basic structure of a Slackware package
- Package management tools
- Package naming, version numbering, build numbers
- Package guidelines & policies
- Package Build scripts
- Checklist to help produce Slackware compliant packages

Introduction: Slackware packages

Slackware Linux consists of a variety of software packages that make up the distribution.

Slackware 10.1:

- 552 main packages (*slackware/*)
- 128 extra packages (*extra/*)

Overview of a Slackware package

- gzipped tar archives (.tgz files) containing the directory structure and files
- Contains post install scripts
- Contains package description
- Typically monolithic (all-in-one) packages

Package management tools

Introducing the Slackware package management tools

Command line tools

- installpkg: install packages
- removepkg: remove installed packages
- upgradepkg: upgrade installed packages with new .tgz packages
- explodepkg: extract .tgz packages into the current directory

Introducing the Slackware package management tools

Curses based tools

Package anatomy

Package anatomy

Package naming scheme

packagename-version-architecture-buildnumber.tgz

Package names are either the name of a single program or a collection/bundle of utilities.

autoconf-2.59-noarch-1.tgz: a single application

tcpip-0.17-i486-29.tgz: a bundle of utilities

Package anatomy

Package version numbers

Single programs: version number is that of the released software.

autoconf-2.59-noarch-1.tgz

'Bundle' packages: version number usually relates to the most major piece of software contained within the package.

tcpip-0.17-i486-29.tgz

Package anatomy

Package architectures

The current official values are:

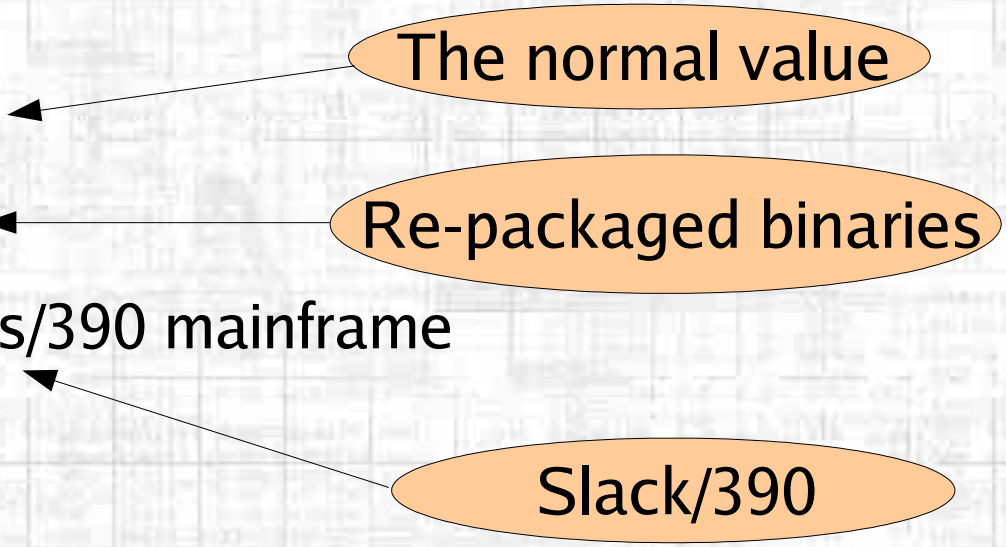
noarch - architecture independent files such as configfiles

i386 - for the i386 or newer

i486 - for the i486 or newer

i586 - for the i586 or newer

s390 - packages for the IBM s/390 mainframe

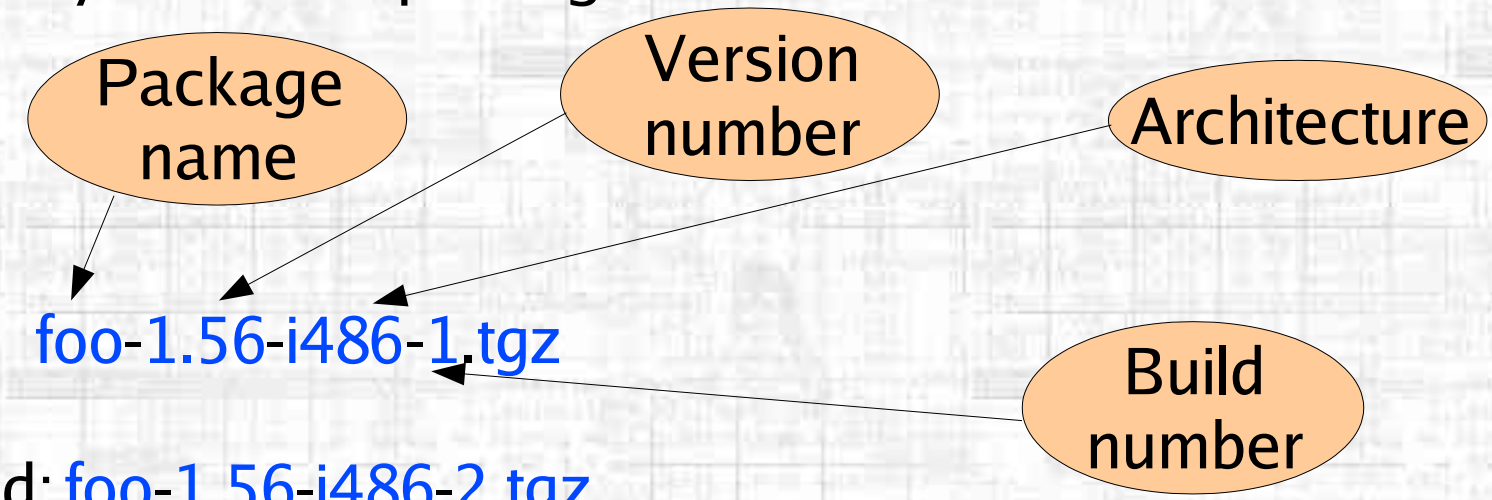


Package anatomy

Build numbers

- Supplements the major version number
- Changed only when the package maintainer makes a change

Example:



First build: foo-1.56-i486-1.tgz

Second build: foo-1.56-i486-2.tgz

Package anatomy

Exploding a package

```
drwxr-xr-x root/root     0 2005-01-26 03:23:42 ./
drwxr-xr-x root/root     0 2005-01-26 03:23:42 etc/
-r--r----- root/root   608 2005-01-26 03:23:41 etc/sudoers.new
drwxr-xr-x root/root     0 2005-01-26 03:23:42 usr/
drwxr-xr-x root/bin      0 2005-01-26 03:23:42 usr/bin/
-rws--x--x root/bin  93056 2005-01-26 03:23:41 usr/bin/sudo
drwxr-xr-x root/root     0 2005-01-26 03:23:42 usr/doc/
drwxr-xr-x root/root     0 2005-01-26 03:23:41 usr/man/
drwxr-xr-x root/root     0 2005-01-26 03:23:41 usr/man/man5/
-r--r--r-- root/root 18214 2005-01-26 03:23:41 usr/man/man5/sudoers.5.g
drwxr-xr-x root/bin      0 2005-01-26 03:23:41 usr/sbin/
-rwxr-xr-x root/bin  61352 2005-01-26 03:23:41 usr/sbin/visudo
-rw-r--r-- root/root   561 2005-01-26 03:23:42 install/doinst.sh
-rw-r--r-- root/root   869 2005-01-26 03:23:42 install/slack-desc
```

Package anatomy

Exploding a package: package description

```
sudo: sudo (give limited root privileges to certain users)
sudo:
sudo: 'sudo' is a command that allows users to execute some
sudo: commands as root.  The /etc/sudoers file
sudo: (edited with 'visudo') specifies which users have access
sudo: to sudo and which commands they can run.  'sudo'
sudo: logs all its activities to /var/log/ so the system
sudo: administrator can keep an eye on things.
sudo:
sudo:
sudo:
```

Package anatomy

Exploding a package: post installation scripts

Two types of post installation script:

install/doinst.sh:
- non-interactive

/var/log/setup/setup.<package name>:
- curses based interactive script

/var/log/setup/setup.onlyonce.<package name>:
- run once only from the OS installer
- last used in 1993/1994 era –**now deprecated**

Package anatomy

Package policies & general guidelines

- Major binary directories are chown root:bin
- gzipped info/man pages and Kernel modules
- Hard links converted to soft links
- Soft links converted to shell script code within install/doinst.sh
- Stripping of static archives, shared objects & binaries
- Deletion of perllocal.pod files
- HTML man pages removed from /usr/doc if there is /usr/man counterpart
- /usr/{doc,info,man} – not FHS (Filesystem Hierarchy) compliant

Package anatomy

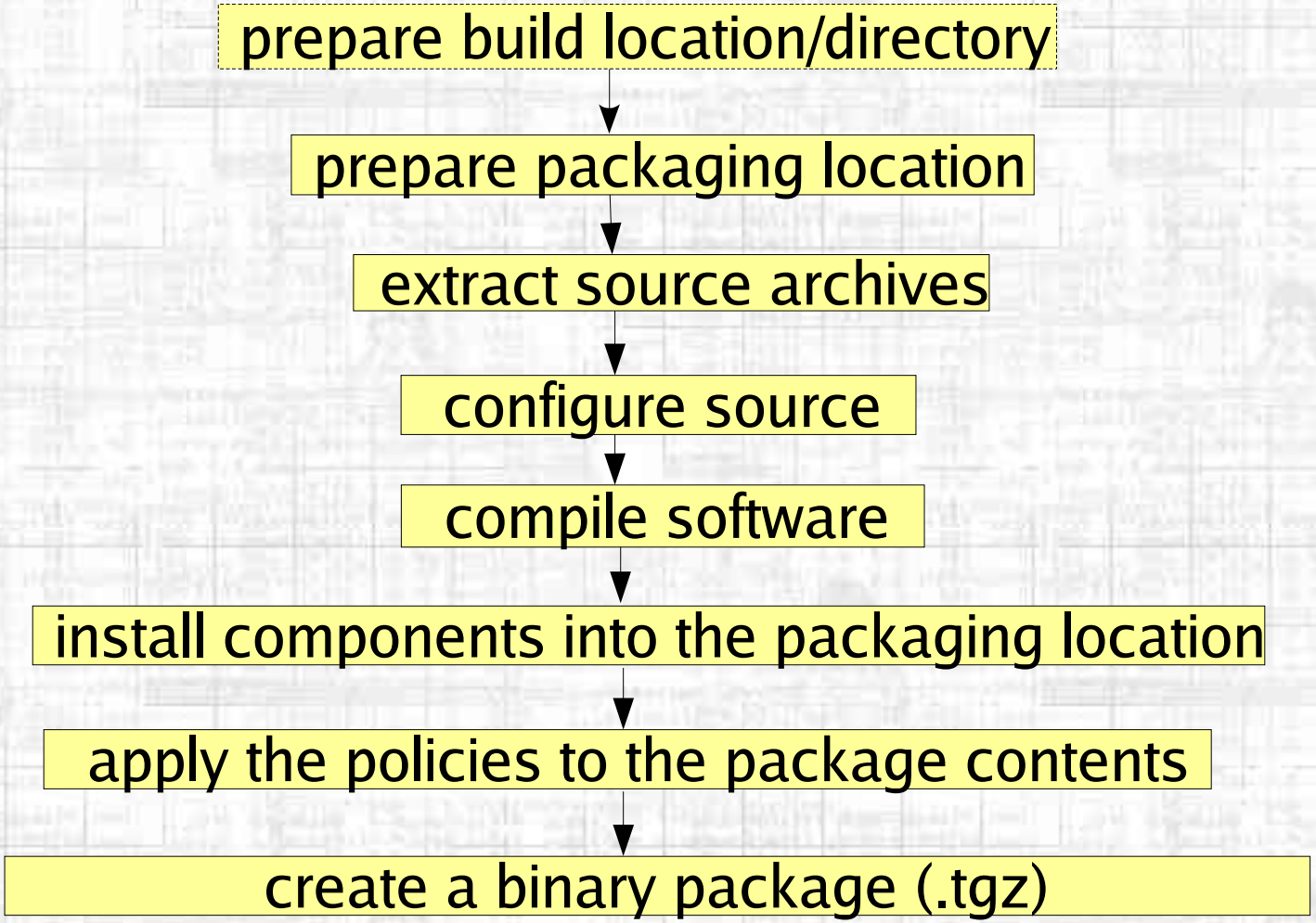
Package dependencies

- Full installation of Slackware is the recommended option
- Official Slackware package tools do not handle dependencies
- Dependencies can be noted in the slack-desc or in config files
- a/aaa_elflibs package satisfies core dependencies
- 3rd party packages can help with dependency resolution but will remain unsupported. Packages may require additional metadata.

Building packages

Building packages

Build scripts: basic process



Building packages

Introducing makepkg

- Produces .tgz packages from the contents of the $PWD
- Converts soft links to shell script code
- Checks for zero length files

Normal syntax:

```
# makepkg -l y -c n /tmp/foo-3.0-i486-1.tgz
```

Convert soft links to script code

Do not reset perms & ownerships

Building packages

Two types of build script

SlackBuild

'Clean' package building scripts:

- Compiles software
- Installs package contents into a special build location
- Creates a binary .tgz package

.build

'Dirty' package building scripts

- Compiles software
- Installs onto the root filesystem!

Building packages

Build notes

- Packages are always built on 'disposable' development machines
- Slackware packages are always built as root
- All packages are built on a full installation of Slackware

Most build scripts are available in the source directories of each package.

Building packages

SlackBuild scripts

- Reproducible package builds
- Simple 'bash' shell scripts – no external calls to macros/functions
- Understandable at a glance and (usually) well commented
- Installs package into a pseudo root directory
- Sets Slackware packaging policy

The preferred method.

Building packages

SlackBuild scripts: example

```
VERSION=2.4
ARCH=${ARCH:=i486}
BUILD=${BUILD:=1}
```

Can override at shell

setup build locations & extract source

```
./configure --prefix=/usr
make
make install DESTDIR=/tmp/package-foo
```

Perms, ownerships stripping, gzipping

install docs into package

set Slackware packaging policies

```
makepkg -l y -c n /tmp/foo-$VERSION-$ARCH-$BUILD.tgz
```

To compile the software and build the package:

```
# ./foo.SlackBuild
```

Run from source dir

Building packages

.build scripts

- Very basic shell scripts
- Do not set Slackware packaging policies
- Installs onto the **root file system**
- Does **not** produce packages!
- Used when the software's build system has no provision for installing into pseudo root location
- Gradually being phased out of Slackware

Replaced with SlackBuilds

Building packages

.build scripts: example

setup build location & extract source

```
./configure --prefix=/usr
make
make install
```

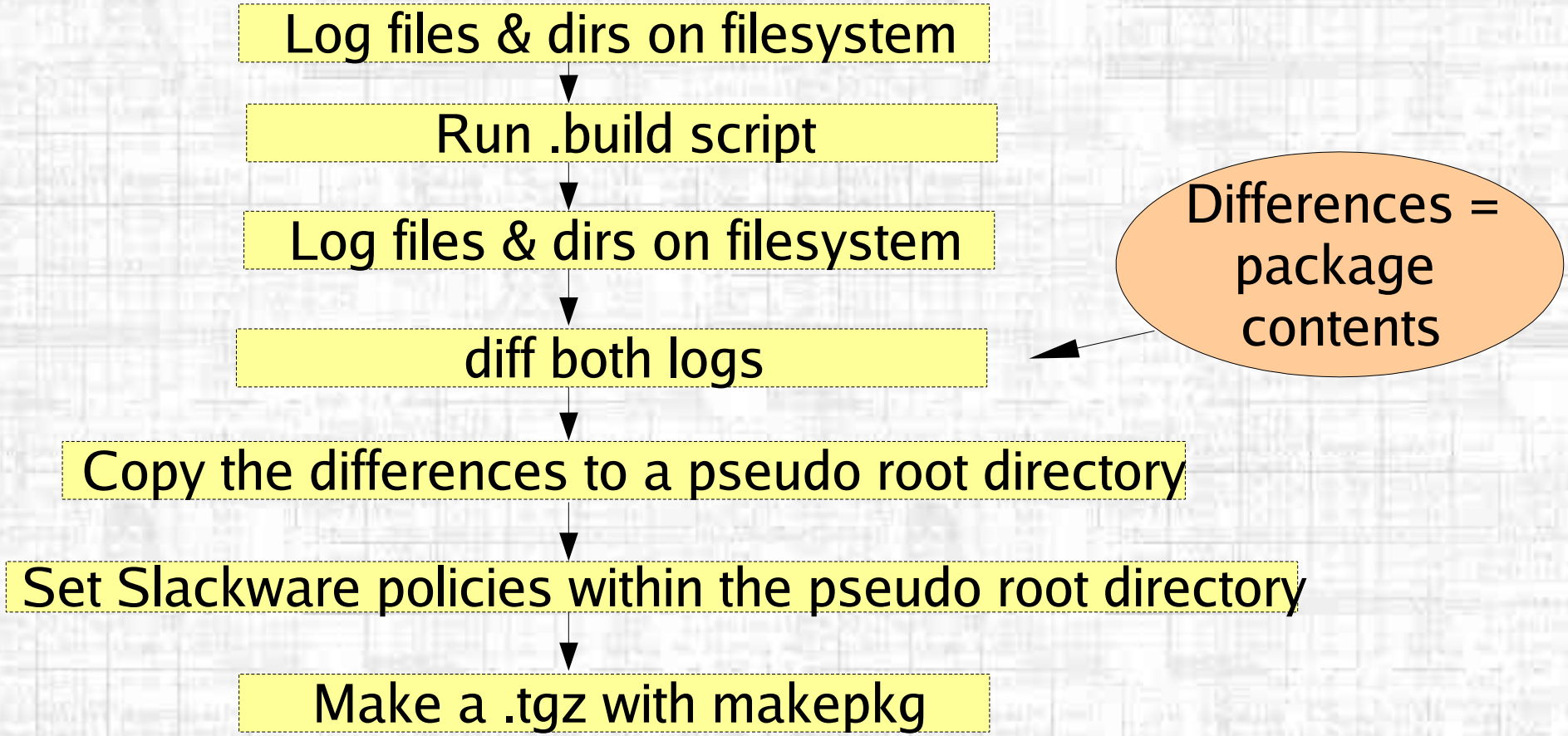
install docs onto the filesystem

To compile the software and install onto the filesystem:

```
# ./foo.build
```

Building packages

.build scripts: the process of package building



Building packages

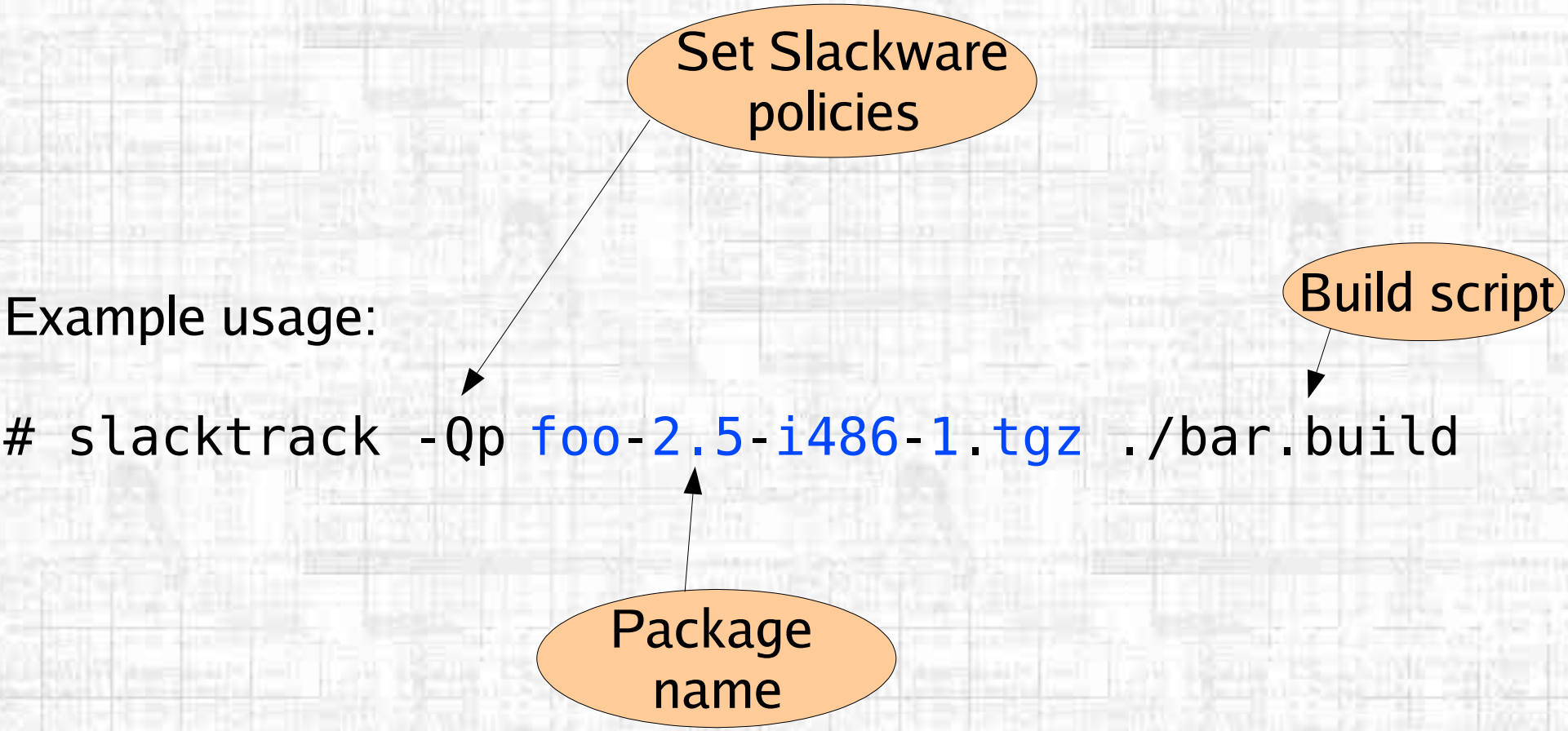
Creating packages from .build scripts: slacktrack

- Originally written to be ARMedslack's primary build tool
- Slackware specific – only produces .tgz
- Can apply all Slackware packaging policies
- Can build replica packages from Slackware's .build scripts
- Can be found in '*extra/slacktrack*' in recent Slackware releases
- Uses installwatch to track the installation
- An alternate version is available: altertrack

Tracks filesystem changes

Building packages

Creating packages from .build scripts: slacktrack



Building packages

Package building checklist

- Directory locations
- install/slack-desc package description file
- install/doinst.sh post installation script
- Binaries, shared & static libraries stripped
- Permissions & ownerships
- Man pages: gzipped with no broken soft links
- Compare your package with the official .tgz (if applicable)

Non FHS compliance

13 lines max, 70 chars wide

Relative paths; ash compatible

/ = chmod 755, root:bin bin dirs & files

Summary

- Simple 'bash' build scripts
- Two types of build script: SlackBuild and .build
- Build scripts available in source tree
- Read the 'OVERVIEW' file included with slacktrack to learn more

slackware®
linux

www.slackware.com