How-To Build
a Simple Slack Package

Autore: Paolo Ottolino

Revisione: Paolo Ottolino

Approvato: Paolo Ottolino

Version

Version	Date	Modification	Author
1.0	18/01/11	Stesura	Paolo Ottolino

Reference Documents

Ref.	Title	Author
1		
2		
3		
4		
5		
6		

Indice

1 Introduction

1.1 Objectives

The aim of this document is about “quick and dirty” package build for Slackware. It shows just a row procedure and it is not an all inclusive howto.

Special attention should be made for:

- permissions
- dependencies
- file location

1.2 Document Structure

This document contains a unique chapter:

- **Chapter 2**: How-To Build a Simple Slackware Package

1.3 Acronyms

TAR	Tape ARchive
TGZ	Tar GNU Zip
TXZ	Tar eXtreme Zip

2 Build a Simple Slackware Package

2.1 Clean Work Area

The first step is to arrange a clean work area, encompassing:

- work (mkdir /usr/local/src/work)
- scripts: cripts to make it easy to upgrade them if needed (mkdir /usr/local/src/work/scritps)
- builds: where to try to install the program to (mkdir /usr/local/src/work/builds)

The next obvious step is to:

- Get the program (that is, download from Internet)
- Unarchive it
- Move to “work” area

2.2 Building

Then, it needs to switch to usual building process: configure-make-make_install. So, the third step is about building the proper Makefile and using it:

- **Configure**: *./configure --prefix=/usr --sysconfdir=/etc –localstatedir=/var* (it works on 90% of the times)
- **Make**: just make the package as usual (it could take time of CPU running)

2.3 Installing

Now, the binary are ready, those are to be correctly positioned, in order to build the package:

- **Make Install**: move the file to the proper location (that is the builds directory). Usually, to accomplish this it is sufficient to add a further parameter in the command line. One of the following 3 should work:
  - *make install DESTDIR=//usr/local/src/work/builds*
  - *make install ROOT=//usr/local/src/work/builds*
  - *make install prefix=/usr/local/src//work/builds*
- **Check**: the three options above, normally work for most programs, however some will not. Perform a check no binary have copied outside desired folder. That is, do make a search inside /usr forlder. Use one of the following:
  - installwatch: use this tool (there are 2 versions)
  - find: issue the following command find /usr -cnewer /usr/local/src/work/builds/usr/bin/<file>
- **Move**: if anything was find, move to “build” area
- **Install**: create the folder install under /usr/localsrc/work/builds
- **Slack-Desc**: write this little file, in install directory, containing a description of the program and application and the name of the person who performed package builder

- **DoInst.sh**: bash file to be created, inside the install directory, in the case there are some symbolic link to treat

2.4 Packaging

The final phase. In order to finally create the package, follow this steps:

- **Permission**: check the builds directory and any file inside is owned by root
- **makepkg**: final step, package creation

- cd /usr/loca/src/work/builds
- read carefully the next paragraph about filename
- issue the command: *makepkg <app>-<ver>-<arch>-<build><3lettername>.txz*
- **Reset**: during makepkg running, the question about to reset all directory permissions to 755 and ownership to root.root will be issued. Say yes

2.5 Filename

The package filename format is the following:

<application>-<version>-<arch>-<build><3letter-name>.tOz

That is:

- <application>: program or application name
- <version>: program or application version
- <build>: compilation attempts (usually number greater than 1 implies bug solutions)
- <3letter-name>: shot for the compiler (it implies not official package)

A brief note about archiving and compression (tOz). "O" could assume 2 values: 'g' or 'x', leading to:

- **tgz**: older version, encompassing tar archive in standard gzip format (Lempel Ziv algorithm). In order to obtain that directly during archiving use: tar -xcvf
- **txz**: newer version (adopted on May 8, 2009), encompassing tar archive in modern xz (lzma) format (Lempel Ziv Markov). In order to obtain that directly during archiving use: tar -Jcvf

2.6 Dependencies

Be sure that if a package requires libraries that are not part of a standard Slackware:

- usually the best option: install you either create a new package with just those libraries
- only that program uses: include them in the package
- optional library: make sure you put that and any other information in a readme file and put it with your package where ever you put it for download.

2.6.1 Legal Issues

Also where ever you put your package you need to put the source remember the GPL. Also check your program make sure its legal to do what you are doing with it