

# Assuming Direct Control

2013-12-03

Eric Rasche

## Modifying the Control file

In this exercise, you will be modifying the control file to specify some additional dependencies

## Control Files

Here is an example control file:

```
Source: mypackage
Section: unknown
Priority: extra
Maintainer: Josip Rodin <joy-mg@debian.org>
Build-Depends: debhelper (>=9)
Standards-Version: 3.9.4
Homepage: <insert the upstream URL, if relevant>
```

```
Package: mypackage
Architecture: any
Depends: ${perl:Depends}
Description: <insert up to 60 chars description>
    <insert long description, indented with spaces>
.
<further description>
```

Briefly,

Parameter	Meaning/Values
Source	Your package's name.
Section	section of the distribution the source package goes into. (I often use "science")
Priority	This will pretty much always be extra for your stuff
Maintainer	That's you!
Package	Same as 'Source'
Architecture	<b>amd64, i386, all, any</b> (there are more.) Use <b>all</b> for script and architecture independent stuff
Depends	List your dependencies here
Description	There's a short description (<60 char) and a long one. Be sure to write both, as they are re helpful to end users.

You can read in depth about control files and their options [here](#). Additionally, there are a lot more complicated methods of specifying dependencies which can be helpful to users figuring out what they need to do to run your software.

## Important Files

`/src/any/usr/bin/cpt-build-tut-02.pl` Our initial perl script now has some dependencies

`/debian/control` This is one of the more important files in a package and part of what makes packages and the APT system work. Control files allow you to specify dependencies and package relationships. In this exercise we'll do precisely that. Specifically the dependencies we'll add are

- perl
- libdigest-md5-perl
- libdigest-crc-perl
- libdigest-sha-perl

## Building

```
cd files  
dpkg-buildpackage
```

## Installation

This package has an additional installation step. Because we're installing from a .deb file, instead of from a repository, we have to do the following:

```
dpkg -i <package>.deb  
apt-get -f install
```

This happens because we've specified dependencies in the package, but dpkg has no knowledge of how to handle those extra dependencies. For that we would need a higher level tool like **apt-get** or **aptitude**.