

In this file I will briefly describe how I worked in order to create my package.

1. Source file (index.php, rss\_php.php, feeds.config) were downloaded from [https://github.com/info6005/6005\\_cw1\\_2012](https://github.com/info6005/6005_cw1_2012).
2. I created a directory and added the above files in it.
3. In that directory i typed `dh_make -e pp6g11@soton.ac.uk -c gpl3 -s --native --indep` in order to create the required directory hierarchy.
4. I then modified the following files under the debian folder:
  - I. Control
  - II. Copyright
  - III. Rules
  - IV. Changelog
  - V. Postinst
  - VI. Postrm
5. The rules file was the most difficult part of the packaging. I downloaded a template file from <http://www.fifi.org/doc/debhelper/examples/rules.indep> and modified it. In essence, in that file I defined the directories that need to be created and the location of all the source files. This was done with the aid of **debhelper**.
6. In addition, I added two more script. **Postinst** is the first one. In this script I added all the configurations that are need by the apache http server in order to add a web application. The second script is the **postrm**. This script has all the actions that are needed in order to disable a web site.
7. I added a launcher which invokes a browser (`xdg-open`) and navigates to <http://localhost> were the app is located. Initially I had a problem with lintian which showed a warning considering this launcher. As a result I decided to copy the launcher into `/usr/share/applications` via the *postinst* script and not during the package installation.
8. During the installation of the package via the software center application a dialog box is displayed to the user, which asks him whether he wants to set the rss application as the default web application of the system. In order to accomplish that, I used the **debconf** tool. I therefore added two more files. A **template** file (cw1-6005-pp6g11.templates) which embraces the question that is displayed to the user and a **configuration** file (cw1-6005-pp6g11.config) which communicates with the debconf tool.
9. The package comes with a man page. The user therefore, can type **man pp6g11** and see the man pages of the package. To do that I added the pp6g11.1.gz file, which contains a file in nroff format.
10. Finally, in the same directory I typed `dpkg-buildpackage -rfakeroot` so as to create the package.

Finally I want to point out, that I used the github as source control tool. Adiitional tools that I utilized were:

TortoiseGit (<http://code.google.com/p/tortoisegit/>)

Git extensions (<http://code.google.com/p/gitextensions/>)

#### EXTENSTIONS:

- I. Tags
- II. Man pages
- III. Web Front End
- IV. Examples
- V. Debconf