

## Robotics Systems Commons - Bug #1430

### Fix debian packaging to conform more strongly to debian policy

02/20/2013 08:37 AM - A. Tuleu

<b>Status:</b>	New	<b>Start date:</b>	02/20/2013
<b>Priority:</b>	Normal	<b>Due date:</b>	
<b>Assignee:</b>		<b>% Done:</b>	0%
<b>Category:</b>	Packaging	<b>Estimated time:</b>	0.00 hour
<b>Target version:</b>			
<b>Description</b>			
<p>I know this is an hassle, but it gives me lot of maintainance nightmare at higher levels because the debian policy is not strongly followed in the debian packaging of librsc.</p> <p>Please fix the following issues :</p> <ul style="list-style-type: none"><li>- have a runtime and a dev package.</li><li>- Make sure runtime package have the same full name than the library (with SONAME). Currently it should be librsc0.9</li><li>- No header in the runtime package, but in the dev package</li><li>- No librsc.so in the runtime, but in the dev package a symlink to librsc.so</li><li>- fixed pkg-config file for this setup (no runtime path, and point simply to -lrsc)</li><li>- provide a shlibs file for better debian integration. (see section 8.6). Basically it will be used by dpkg to maintain its database of symbols / shared lib dependency</li></ul> <p>This come from the debian manual policy, and could causes problem, especially when using pkg-config to symlink with rsc (which is the case for liboncilla compilation from webots).</p> <p>Please refer to <a href="http://www.debian.org/doc/debian-policy/ch-sharedlibs.html">http://www.debian.org/doc/debian-policy/ch-sharedlibs.html</a></p>			

#### History

##### #1 - 02/20/2013 08:41 AM - A. Tuleu

sorry I made a typo. in the dev package librsc.so should be a pointer to librsc.so.0.9

##### #2 - 02/21/2013 03:41 PM - Anonymous

- Status changed from New to Feedback

CPack offers support for some of these aspects, so we might be able to stick to CPack and solve these issues (which I would be in favor of).

For the split-up into dev and runtime package: I did this once for another package with cpack and this was actually quite easy and worked fine. See

[Component Install With CPack](#)

Since this might have larger impact on existing installations, I suggest to start with an rsc branch testing a split into packages before doing this in master.

##### #3 - 02/27/2013 01:25 PM - Anonymous

- Category set to Packaging

- Status changed from Feedback to New

I played around a bit in a separate branch [wip-debian-packaging](#) and an [CI job for it](#).

It turns out, that the combination of cpack debian generator and cpack components is not powerful enough for our needs. For example:

- The debian **package names** of the package components (bin and dev in the test-case) are created automatically as {CPACK\_DEBIAN\_PACKAGE\_NAME}-{COMPONENT}, so that there is always the -dev or -bin suffix in the package name. So having a package with the same name as the library, as requested by lintian, is not possible this way (there is *maybe* a [very ugly work-around](#), which involves calling cpack manually, but I didn't try this one).
- Several debian package options are considered for all components, e. g. the shlibs file is inserted in all components, leading to other lintian issues, complaining about an shlibs file in a package without libs (i.e. the dev package)
- etc. (see lintian plugin in the [CI job](#))