Compiling rudder Agent for Debian from source

Over time, quite a few ARM CPU based systems have come together at my home./ I wanted to compile a rudder agent for them so I could enjoy the niceties of CFEngine based automation on them with the wonderful lazyness that Rudder brings.

Basically, I have a bunch of Debian based ones (Cubietrucks, Raspi B+ and Raspi2) and also some NAS systems running ARCH Linux.

This article will describe what was needed to get a set of working Debian packages.

(Debian support is mostly ready to use. Other cases like Arch, AlpineLinux and FreeBSD are a different story)

The agent types

I'll describe building rudder-agent (one large bundle) and rudder-agent-thin (using system-side FusionInventory packages)

Personally, I'd recommend the 'thin' agent since the full one needs helluvalot of perl packages and compiled *forever*. It also comes with an older perl version. If you're not in a internet-restricted environment it's probably easier to just rely on the distro.

But: I'm not sure if you're not missing some of the Rudder patches for FusionInventory

Build dependencies

I found I needed to add some extra packages beyond the build-essentials

```
apt-get install build-essential autoconf devscripts lintian diffutils patch
patchutils git-core libssl-dev bison flex libtool libpcre3-dev libpam0g-dev
lsb-release
```

If you want to use the "smaller" agent, also install fusioninventory.

```
apt-get install fusioninventory-agent
```

Setup

Next, define your build directory, I worked in /usr/src, and move to that directory.

In here, clone the rudder source tree, and switch to the new version 3.0 branch.

```
git clone https://github.com/Normation/rudder-packages.git &&
cd rudder-packages &&
git checkout branches/rudder/3.0
```

Next, you'll need to run some special script which, among others, sets up the "debian" folder used for creating debian packages with dpkg-buildpackage.

Use this to setup building the full agent, or the next to build the thin agent.

```
# Prepare the sources
cd rudder-agent-thin/SOURCES
make RUDDER_VERSION_TO_PACKAGE=3.0.3
```

Now you need to tell debian what you're exactly gonna build in your package, this is done using the "dch" script.

It will launch you into an $EDITOR session.

```
# Create a Debian changelog entry corresponding to the current version you want to package
# (the version is the most important part: use e.g. "3.0.3-wheezy0")
cd ..
dch
rudder-agent-thin (2.11.0~git-1) UNRELEASED; urgency=low


  *

 -- root <root@cubie1.there>  Mon, 30 Mar 2015 23:29:33 +0000
```

This is how I changed it to build for a cubietruck running wheezy.

```
rudder-agent-thin (3.0.3-wheezy0) wheezy; urgency=low

  * build for ARMv7 on cubietruck / raspi2

 -- root <root@cubie1.there>  Mon, 30 Mar 2015 23:29:33 +0000
```

For a ARM-based CPU I was told you also need to remove the dmidecode dependency.

DMI is a management standard for PC based systems, on ARM it is not implemented.

I found the dependency in

debian/rudder-agent-thin/DEBIAN/control

it wasn't enough to remove it like that and it's fucking annoying to not know _where_ it needs to originally be removed. The resulting dpkgs still depend on dmidecode and so it's not possible to install cleanly.

Debian being debian that means it's not possible to install without inheriting a shitload of errors from dpkg.

Running the build

Once this has been configured, you can launch the build.

```
# Let's roll
dpkg-buildpackage -us -uc
[....]
ls -ltr ../*deb
```

The thin build didn't yield cfengine binaries. Maybe can also be installed from the distro, but I was thinking they're included.

The full agent has so far built on all my tiny systems.

On the Raspi 1 I had to add extra swap space (has 700MB now)

I also needed to add a 'dmidecode' script which gives an uuid when called (fusioninventory runs dmidecode -s)

After installation, the rudder init script from /etc was missing, and also the initial set of promises.

Put in place using

```
cp -r /opt/rudder/share/initial-promises/*
/var/rudder/cfengine-community/inputs
```

Actually, I found I *had* CFEngine3 installed from some earlier "I should have that around" session.

I uninstalled it.

I also had to remove the /opt/rudder/etc/disable-agent flagfile, it was in place after the installation.

Most probably all the config scripts don't complete because of the missing init script.