Build Moonshot from source on Debian

The Moonshot source code is available from our GIT repository and it all can be built by hand relatively easier, assuming you have all of the prerequisite packages installed. This page has instructions for getting those packages and building the software itself.

Contents

1. System Preparation

1.1. Requirements

To build all of the Moonshot components, you need various packages installed. To install all of these, see below.

1.1.1. Get the required packages on Debian 8

To get all of the requirements on Debian 8 (Jessie), you will need to add the Wheezy-backports repository, and issue a couple of apt-get install commands:

1. Add the Wheezy-backports repository.

```
$ echo "deb http://ftp.us.debian.org/debian jessie-backports main" >> /etc/apt/sources.list
```

2. Refresh local package information

```
$ apt-get update
```

3. Install a whole load of packages from the normal Wheezy repo:

```
$ apt-get install libxml-security-c-dev libxerces-c-dev libcurl4-openssl-dev liblog4cpp5-dev autotools-
dev doxygen libtool libconfuse-dev autoconf automake python libsqlite3-dev libtalloc-dev fakeroot
libkrb5-dev libevent-dev build-essential libboost-all-dev libxmltooling-dev liblog4shib-dev xmltooling-
schemas autopoint libgtk2.0-dev libgee-dev libdbus-glib-1-dev valac libgnome-keyring-dev desktop-file-
utils
```

4. Install a package from the Wheezy-backports repo

```
$ apt-get install -t jessie-backports libsaml2-dev
```

2. Checkout the Moonshot source

The Moonshot source code is all stored in a GIT repository. To fetch it, issue the following command.

```
$ git clone --recursive http://www.project-moonshot.org/git/moonshot.git
```

3. Build Moonshot

To build Moonshot, simply enter the moonshot directory of the git repo that you just checked out, and run the builder script. In the --prefix option, you can specify where the code will be installed to. Note that you will need to *build as a user who has permission to install in the target directory*, because the builder script installs during the build, instead of doing an install after the build has finished (so you can't build first as an underprivileged user and install as a superuser as you would normally do).

```
$ cd moonshot && ./builder --prefix=/opt/moonshot
```

You probably know this, but if you stuck your installation into an unusual location, you may need to do the following (as root) - where, here, /opt/moonshot is the install path you used for the prefix option above:

```
# echo /opt/moonshot/lib >/etc/ld.so.conf.d/moonshot.conf
# ldconfig
```

(TODO: document /etc/init.d/ stuff?)