



*LIGO Laboratory / LIGO Scientific Collaboration*

LIGO-T1500458-v1

*LIGO*

08/20/2015

---

Adding Ubuntu OS to existing aLIGO DAQ

---

K. Thorne

Distribution of this document:

LIGO Scientific Collaboration

This is an internal working note of the LIGO Laboratory.

**California Institute of Technology**

**LIGO Project – MS 18-34**

**1200 E. California Blvd.**

**Pasadena, CA 91125**

Phone (626) 395-2129

Fax (626) 304-9834

E-mail: [info@ligo.caltech.edu](mailto:info@ligo.caltech.edu)

**Massachusetts Institute of Technology**

**LIGO Project – NW22-295**

**185 Albany St**

**Cambridge, MA 02139**

Phone (617) 253-4824

Fax (617) 253-7014

E-mail: [info@ligo.mit.edu](mailto:info@ligo.mit.edu)

**LIGO Hanford Observatory**

**P.O. Box 1970**

**Mail Stop S9-02**

**Richland WA 99352**

Phone 509-372-8106

Fax 509-372-8137

**LIGO Livingston Observatory**

**P.O. Box 940**

**Livingston, LA 70754**

Phone 225-686-3100

Fax 225-686-7189

<http://www.ligo.caltech.edu/>

## Contents

|            |  |           |
|------------|--|-----------|
| <b>1</b>   | <b><i>Introduction</i></b> .....                                   | <b>3</b>  |
| <b>2</b>   | <b><i>Scope</i></b> .....  | <b>3</b>  |
| <b>3</b>   | <b><i>References</i></b> .....                                     | <b>3</b>  |
| <b>4</b>   | <b><i>Overview</i></b> .....                                       | <b>3</b>  |
| <b>5</b>   | <b><i>Detailed Installation Steps</i></b> .....                    | <b>4</b>  |
| <b>5.1</b> | <b>Install Ubuntu 12 versions of LIGO packages</b> .....           | <b>4</b>  |
| <b>5.2</b> | <b>CDS Environment Configuration Scripts</b> .....                 | <b>5</b>  |
| <b>5.3</b> | <b>Install and configure DAQ computer with new OS</b> .....        | <b>6</b>  |
| <b>5.4</b> | <b>NFS mounts of front-end system directories</b> .....            | <b>7</b>  |
| <b>5.5</b> | <b>Create ‘controls’ user, customize BASH initialization</b> ..... | <b>7</b>  |
| <b>5.6</b> | <b>Create ldconfig files for LIGO packages</b> .....               | <b>8</b>  |
| <b>5.7</b> | <b>Create RCG checkout, DAQ build area for Ubuntu 12</b> .....     | <b>8</b>  |
| <b>5.8</b> | <b>Install init scripts for caRepeater, daqd</b> .....             | <b>9</b>  |
| <b>5.9</b> | <b>Build and install DAQ executables</b> .....                     | <b>10</b> |

## 1 Introduction

This document describes installing DAQ computers using a different OS (here Ubuntu 12) to an existing IFO DAQ system running a single OS (Gentoo 2.6.34/5).

## 2 Scope

The scope of this document is the software and script changes required to add an Ubuntu OS DAQ computer and configuration of that computer. It does not cover installing Ubuntu on the computer, nor general DAQ setup. It does not cover using Ubuntu 12 on a front-end computer.

## 3 References

[LIGO-T1000379](#) - CDS Environment Configuration Scripts

[LIGO-T1000248](#) – aLIGO CDS File System Directories

[LIGO-T1500227](#) – aLIGO DAQ hardware, software setup

## 4 Overview

The initial DAQ systems for aLIGO had a non-real-time version (Gentoo 2.6.35) of the real-time OS (Gentoo 2.6.34-cs) that is used on the front-end computers. Thus, they could mount the same NFS directories from the boot server, use the same LIGO packages and use the same build areas for executables.

But we want to use new Linux kernels for the DAQ computers to use updated packages, newer hardware, and for improved cyber- security from regular upgrades. To support an additional operating system (in our case Ubuntu 12), the following steps are required

- Install LIGO packages (EPICS, framecpp) for another OS
- Add support for another OS to the CDS environment configuration
- Install alternate OS on DAQ computer. Configure BIOS settings, host table, Ethernet ports
- Add NFS mounts from boot server
- Add ‘controls’ user and customize BASH initialization
- Create separate RCG subversion checkout for the OS and create separate build area.
- Add init scripts for caRepeater, daqd, nds
- Build DAQ executables and install in target areas

## 5 Detailed Installation Steps

### 5.1 Install Ubuntu 12 versions of LIGO packages

The DAQ code requires some LIGO-specific packages to run. It needs EPICS and framecpp. One may also want access to a command-line GDS ‘diag’. We need to install these packages so they can be used and built against

We eventually want to move to using pre-built LIGO DASWG packages distributed with ‘aptitude’ or ‘rpm’. Since Ubuntu 12 is not a fully supported OS, we are not quite there yet, so we will install them in directories on the front-end boot server.

First create an OS-specific sub-directory (‘ubuntu12’) under the main real-time applications directory (‘/opt/rtapps’). This follows the practice detailed in aLIGO CDS file system directories document [LIGO-T1000248](#).

Then get the Ubuntu 12 pre-built binary tar-balls available at the LIGO CDS DAQ download site: <https://llocds.ligo-la.caltech.edu/daq/software/binary/rtapps/ubu12/>. There you will find tar-balls for EPICS, gds, ldas-tools, libframe, ligotools, metaio. Add a ‘tarfiles’ subdirectory to ‘/opt/rtapps/ubuntu12’ and copy them there. Then expand each tarball into ‘/opt/rtapps/ubuntu12’. Lastly create soft-links from the version-specific package to a generic one (i.e. epics links to epics-3.14.12.2\_long). Note that we make soft-links for both ‘framecpp’ and ‘ldas-tools’ to the same location.

At the end, your directory listing should look like:

```
controls@l1build0 /opt/rtapps/ubuntu12 $ ls -l
total 32
lrwxrwxrwx 1 controls 1001 20 Apr 20 13:10 epics -> epics-3.14.12.2_long
drwxrwxr-x 10 controls 1001 4096 May 6 02:52 epics-3.14.12.2_long
lrwxrwxrwx 1 controls 1001 21 Aug 18 09:21 framecpp -> ldas-tools-1.19.32-p1
lrwxrwxrwx 1 controls 1001 10 Apr 20 13:10 gds -> gds-2.16.3
drwxrwxr-x 6 controls 1001 4096 May 23 2013 gds-2.16.3
lrwxrwxrwx 1 controls 1001 21 Aug 18 09:21 ldas-tools -> ldas-tools-1.19.32-p1
drwxr-xr-x 9 controls 1001 4096 Aug 15 23:02 ldas-tools-1.19.32-p1
lrwxrwxrwx 1 controls 1001 15 Apr 20 13:10 libframe -> libframe-8.17.2
drwxrwxr-x 7 controls 1001 4096 May 23 2013 libframe-8.17.2
drwxr-xr-x 10 controls 1001 4096 Jun 24 2010 ligotools
lrwxrwxrwx 1 controls 1001 12 Apr 20 13:10 metaio -> metaio-8.2.1
drwxrwxr-x 6 controls 1001 4096 May 23 2013 metaio-8.2.1
drwxrwxr-x 2 controls 1001 4096 Aug 17 15:28 tarfiles
```

The ‘1.19.32-p1’ revision of ldas-tools has the July 1, 2015 leap second patch to ‘framecpp’.

These are also available in source format at <https://llocds.ligo-la.caltech.edu/daq/software/source/>, but then you need to build and install each one.

## 5.2 CDS Environment Configuration Scripts

Described in [LIGO-T1000379](#), these scripts set up paths, libraries based on the location and computer. Initially, the real-time setup scripts only support one OS (Gentoo 2.6.34/5). We need to add support to automatically detect the OS and set paths appropriately.

The scripts ‘rtrc\_linux.sh’ and ‘stdir\_linux.sh’ in /opt/cdscfg/<site>/<ifo> need to be updated to detect the OS and set the apps folder correctly. We use a different folder on ‘/opt/rtapps’ for each OS (See [LIGO-T1000248](#)). Here is an example from ‘rtrc\_linux.sh’ on the LLO DAQ test stand.

```
# define apps folder
if [ -e /etc/gentoo-release ] ; then
    # define apps folder
    RTAPPSROOT=${RTAPPSBASEDIR}
    OS_ARCH=linux-x86_64
else
    uburel=`lsb_release -rs`
    if [[ "$uburel" == "12.*" ]]; then
        OS_ARCH=ubuntu12
    elif [[ "$uburel" == "11.*" ]]; then
        OS_ARCH=linux-x86_64
    elif [[ "$uburel" == "6.*" ]]; then
        OS_ARCH=sl6
    else
        OS_ARCH=linux-x86_64
    fi
    RTAPPSROOT=${RTAPPSBASEDIR}/${OS_ARCH}
fi
export RTAPPSROOT
export OS_ARCH
```

As done in the previous step, Ubuntu 12 will use ‘/opt/rtapps/ubuntu12’ for ‘RTAPPSROOT’. The rest of that file should use \$RTAPPSROOT

```
if [ -e ${RTAPPSROOT}/epics/etc/epics-user-env.sh ] ; then
    source ${RTAPPSROOT}/epics/etc/epics-user-env.sh
else
    echo "ERROR: No EPICS setup at $RTAPPSROOT/epics/etc"
fi
```

Note that we have been transitioning from stand-alone ‘framecpp’ package to ‘ldas-tools’ package containing framecpp. Due to this and a change in directory structure, the following is required to detect old and new set-ups

```
# - framecpp - choose ldas-tools if available
# new style is ${OS_ARCH}/(app)/etc, old style is (app)/linux-x86_64/etc
if [ -e ${RTAPPSROOT}/ldas-tools/etc/framecpp-user-env.sh ] ; then
    source ${RTAPPSROOT}/ldas-tools/etc/framecpp-user-env.sh
elif [ -e ${RTAPPSROOT}/ldas-tools/${EPICS_HOST_ARCH}/etc/framecpp-user-env.sh
$
    source ${RTAPPSROOT}/ldas-tools/${EPICS_HOST_ARCH}/etc/framecpp-user-env.sh
elif [ -e ${RTAPPSROOT}/framecpp/etc/framecpp-user-env.sh ] ; then
    source ${RTAPPSROOT}/framecpp/etc/framecpp-user-env.sh
elif [ -e ${RTAPPSROOT}/framecpp/${EPICS_HOST_ARCH}/etc/framecpp-user-env.sh ]
$
    source ${RTAPPSROOT}/framecpp/${EPICS_HOST_ARCH}/etc/framecpp-user-env.sh
fi
```

Note that you will also want to update things in the Python, Perl configuration scripts ('stddir.pl', 'stddir.py') found in '/opt/cdscfg/<site>' to set the apps root. In the Python script is the section:

```
SYSBIN = "/bin:/usr/bin"
if os.path.isfile("/etc/gentoo-release"):
    SYSLIB = "/lib64:/usr/lib64"
    OS_ARCH="linux-x86_64"
    PYTHONREV="2.6"
    APPSROOT=appsdir
else:
    if proctype == "x86_64":
        SYSLIB = "/lib64:/lib:/usr/lib64:/usr/lib"
        if uburel.startswith("12."):
            OS_ARCH="ubuntu12"
            PYTHONREV="2.7"
        elif uburel.startswith("6."):
            OS_ARCH="sl6"
            PYTHONREV="2.6"
        elif uburel.startswith("11."):
            OS_ARCH="ubuntull"
            PYTHONREV="2.6"
    else:
        OS_ARCH="linux-x86_64"
        PYTHONREV="2.6"
    else:
        OS_ARCH="linux-x86"
        SYSLIB = "/lib:/usr/lib"
        PYTHONREV="2.6"
    APPSROOT=os.path.join(appsdir,OS_ARCH)
```

These changes are included in the latest configuration script installer – See CDS Subversion at <https://redoubt.ligo-wa.caltech.edu/svn/controls/trunk/cdscfg>. Web interface is at <https://redoubt.ligo-wa.caltech.edu/websvn/>

### 5.3 Install and configure DAQ computer with new OS

Install the new operating system (i.e. Ubuntu 12.04) on the DAQ computer. CDS sys-admins typically use a PXE boot server to install a standard image, but one could start from an installation image or a disk cloned from another computer.

Much of the following is covered in [LIGO-T1500227](#). Configure the Ethernet ports on the computer. In Ubuntu 12, the IP info is found in `/etc/network/interfaces`, while the configuration matching devices/MAC addresses to Ethernet ports (eth0,1,2) is found in `/etc/udev/rules.d/70-persistent-net.rules`. Typically `eth0` is set to the FE LAN address, `eth1` to the LAN connection to the QFS server (if a frame-writer), while `eth2` is typically the DAQ OUT broadcast network using the 10G Myricom adapter. The Ubuntu 12 distribution has a built-in Myricom driver for the card.

For DAQ, the BIOS setting can mostly be left at the defaults. You want to make sure the IPMI interface (if one exists) is set to a static address so it does not flood the other LANs with DHCP requests. If you have a dual-CPU machine with at least 6 cores/CPU that you are using as a dedicated frame-writer, you may get more stable performance by turning off hyper-threading so the CPU-intensive frame writing threads don't interrupt threads ingesting data packets on the 10G network from the data concentrator.

## 5.4 NFS mounts of front-end system directories

Now we need to add NFS mounts of the front-end system directories. Under the `/opt` directory create the needed mount points (cdscfg,rtapps, rtcds). Then update `/etc/fstab` as below (using L1 as the example)

```
l1boot:/opt/cdscfg      /opt/cdscfg      nfs      defaults      0 0
l1boot:/opt/rtapps     /opt/rtapps     nfs      defaults      0
l1boot:/opt/rtcds     /opt/rtcds     nfs      defaults      0 0
```

Then mount the directories.

## 5.5 Create 'controls' user, customize BASH initialization

We need to create a `controls` user with the correct UID and `controls` group with the correct GID to match the rest of the front-end system

```
controls@l1daqfw1:~$ grep control /etc/passwd
controls:x:1001:1001::/home/controls:/bin/bash
controls@l1daqfw1:~$ grep control /etc/group
controls:x:1001:
```

Typically on these machines the controls user gets sudo privileges

```
controls@l1daqfw1:~$ sudo grep controls /etc/sudoers
controls    ALL=(ALL) NOPASSWD: ALL
```

In the `controls` home directory (which should be local to the machine), you need to add call to `rtsetup.sh` in `.bashrc`

```
controls@l1daqfw1:~$ more .bashrc
#
# hard-code to ubuntu RCG
#
export PRIVATE_RCG="/opt/rtcds/rtscore/uburelease"
#
source /opt/cdscfg/rtsetup.sh
```

Note that we are also defining use of a different RCG release area. More on that in the next step

## 5.6 Create ldconfig files for LIGO packages

The DAQ processes are started in ‘init’ scripts. Thus they typically don’t get use of the CDS environment configuration scripts to put libraries in their paths. To handle this, we add files in the ‘/etc/ld.so.conf.d’ directory, one for each package with libraries. Here is the list at L1:

```
controls@l1daqfw1:~$ ls -l /etc/ld.so.conf.d
total 24
-rw-r--r-- 1 root root 163 Apr 16 12:05 epics-x86_64.conf
-rw-r--r-- 1 root root 34 Feb 6 2015 framecpp-x86_64.conf
-rw-r--r-- 1 root root 29 Feb 6 2015 gds-x86_64.conf
-rw-r--r-- 1 root root 34 Feb 6 2015 libframe-x86_64.conf
-rw-r--r-- 1 root root 32 Feb 6 2015 libmetaio-x86_64.conf
-rw-r--r-- 1 root root 68 Apr 19 2012 x86_64-linux-gnu.conf
```

Each file lists directories containing library files. The EPICS one has multiple lines:

```
controls@l1daqfw1:~$ more /etc/ld.so.conf.d/epics-x86_64.conf
/opt/rtapps/ubuntu12/epics/base/lib/linux-x86_64
/opt/rtapps/ubuntu12/epics/modules/sncseq/lib/linux-x86_64
/opt/rtapps/ubuntu12/epics/extensions/lib/linux-x86_64
```

The others are one-liners

```
controls@l1daqfw1:~$ more /etc/ld.so.conf.d/framecpp-x86_64.conf
/opt/rtapps/ubuntu12/framecpp/lib
```

These should match the additions to LD\_LIBRARY\_PATH done in the user setup scripts for each package (i.e. ‘/opt/rtapps/ubuntu12/epics/etc/epics-user-env.sh’).

To update things, run ‘ldconfig’ with sudo privileges.

Example scripts for this and daqd are in the RCG distribution under ‘setup/ubudaq/etc/ld.so.conf.d’

## 5.7 Create RCG checkout, DAQ build area for Ubuntu 12

When building the DAQ executables ‘daqd’, ‘nds’, you need to do some things (build gds, bootstrap daqd, nds configure) in the RCG checkout directories. To avoid screwing up front-end and DAQ builds against the default OS (Gentoo 2.16.34/35), we need to use a separate checkout. This is done (acting as the ‘controls’ user)

```
cd /opt/rtcds/rtscore
mkdir ubutrunk
svn co https://redoubt.ligo-wa.caltech.edu/svn/advLigoRTS/trunk ubutrunk
ln -s ubutrunk uburelease.
```

We have to use ‘trunk’ or RCG releases 2.10 and above to build easily against different OSes.

You make sure the ‘controls’ user on the Ubuntu 12 machine use this as the default by setting PRIVATE\_RCG before calling the environment configuration script (‘/opt/cdscfg/rtsetup.sh’), as was done above.



You should make sure your `rtsetup` is updated to support `PRIVATE_RCG`. You should have the following in `/opt/cdscfg/<site>/<ifo>/rtrc.sh`:

```
# get RCG environment
if [ -n "${PRIVATE_RCG}" -a -f "${PRIVATE_RCG}/etc/rcg-user-env.sh" ] ; then
    source ${PRIVATE_RCG}/etc/rcg-user-env.sh
    alias rcgcode="pushd ${PRIVATE_RCG}"
elif [ -f "${RTCDSBASE}/rtscore/release/etc/rcg-user-env.sh" ] ; then
    source ${RTCDSBASE}/rtscore/release/etc/rcg-user-env.sh
    alias rcgcode="pushd ${RTCDSBASE}/rtscore/release"
fi
```

Now we can do the preliminaries to build the executables. We need to build RCG-specific GDS libraries. We also need to configure the `daqd`, `nds` source areas so the makefiles can find the correct packages. This is all covered in aLIGO DAQ hardware, software setup document ([LIGO-T1500227](#)), but I will repeat steps here:

Log into an Ubuntu12 DAQ machine as `controls` user

```
cd /opt/rtcds/rtscore/uburelease
cd src/gds
make clean
make
cd ../daqd
./bootstrap
cd ../nds
./bootstrap
cd /opt/rtcds/<site>/<ifo>/daqbuild
mkdir daqubu
cd daqubu
/opt/rtcds/rtscore/uburelease/configure
```

Now we are ready to build DAQ executables.

## 5.8 Install init scripts for `caRepeater`, `daqd`

The EPICS build makes an init script `S99caRepeater` that we can use

Acting with root privileges on the Ubuntu12 DAQ computer

```
cp /opt/rtapps/ubuntu12/epics/base/bin/linux-x86_64/S99caRepeater
/etc/init.d/caRepeater
cd /etc
cd rc2.d
ln -s ../init.d/caRepeater S20caRepeater
cd ../rc3.d
ln -s ../init.d/caRepeater S20caRepeater
cd ../rc4.d
ln -s ../init.d/caRepeater S20caRepeater
cd ../rc5.d
ln -s ../init.d/caRepeater S20caRepeater
```

Example scripts for this and `daqd` are in the RCG distribution under `setup/ubudaq/etc/init.d`

You also need to install a `daqd` init script, as well as a config script (`/etc/default/daqd`)

Acting with root privileges

```
cp /opt/rtcads/rtscore/uburelease/setup/ubudaq/etc/init.d/daqd /etc/default
cp /opt/rtcads/rtscore/uburelease/setup/ubudaq/etc/default/daqd /etc/default
```

You need to localize the target directory in `/etc/default/daqd`. Here is the L1 version

```
controls@l1daqfw1:/opt/rtcads/rtscore/uburelease/setup$ more /etc/default/daqd
TARGET_BASE=/opt/rtcads/l1o/l1/target

TARGET=$TARGET_BASE/`hostname`

LOGDIR=$TARGET/logs
```

We also use `monit`, so first customize the config file `/etc/monit/monitrc` for that machine.

You also should install the per-app configs, especially the one for `daqd`. These are also stored in the RCG release

```
cp /opt/rtcads/rtscore/uburelease/setup/ubudaq/etc/monit/conf.d/*
/etc/monit/conf.d
```

## 5.9 Build and install DAQ executables

We should finally be ready to build Ubuntu 12 DAQ executable and install them in the target areas. For instance, for a frame-writer on L1

Login as `controls` to an Ubuntu 12 DAQ machine

```
Cd /opt/rtcads/l1o/l1/daqbuild/daqubu
Make fw
Cp build/fw/daqd /opt/rtcads/l1o/l1/target/l1daqfw1/bin_archive/daqd_ubu
Cd /opt/rtcads/l1o/l1/target/l1daqfw1
<make any edits to the daqdrf file that are needed>
sudo /etc/init.d/monit stop
sudo /etc/init.d/daqd stop
cp -p bin_archive/daqd_ubu daqd
sudo /etc/init.d/monit start
```

You make have to access the `monit` web page for the machine to `'enable monitoring'` on `daqd` to get it going the first time.