1. Quick Start - LSST Cluster Orchestration

Just enough to get you going quickly. Read through the other pages in the list below the Orchestration topic for more details

Launching HTCondor jobs through runOrca.py is done on the machine lsst-dev. Log in there to execute your jobs.

1) Create a $HOME/.lsst directory with permissions 700.

```
$ mkdir $HOME/.lsst
$ chmod 700 $HOME/.lsst
```

2) Create a db-auth.py configuration file with your mysql host, user, password and mysql port information. This from must have permissions 600

```
config.database.authInfo["auth1"].host = "lsst10.ncsa.illinois.edu"
config.database.authInfo["auth1"].user = "juser"
config.database.authInfo["auth1"].password = "funkystuff"
config.database.authInfo["auth1"].port = 3306
```

```
$ chmod 600 $HOME/.lsst/db-auth.py
```

If you don't already have a MySQL user account on the lsst10.ncsa.illinois.edu MySQL server, you'll need to request one by sending the following to lsst-account@ncsa.illinois.edu:

- Your name
- Institution and LSST affiliation
- Your email address
- Desired account name

You'll also have to create a db-auth.paf file, because the pex_persistence package wasn't updated to use Config. This file also belongs in $HOME/.lsst, and requires permissions 600.

```
database: {
    authInfo: {
        host: lsst-db.ncsa.illinois.edu
        port: 3306
        user: <user>
        password: <password>
    }
}
```

3) Create a HTCondor configuration file in $HOME/.lsst/condor-info.py

```
root.platform["lsst"].user.name = "juser"

root.platform["lsst"].user.home = "/lsst/home/juser"
```

Yes, this looks like something you shouldn't have to specify. This is done for a consistent interface between platforms, since not all systems have consistent user names and home directories between sites or execution machines.

4) Create a directory named $HOME/condor_scratch

```
$ mkdir $HOME/condor_scratch
```

5) Setup ctrl_execute and ctrl_platform_lsst

```
$ setup ctrl_execute

$ setup ctrl_platform_lsst
```

6) execute runOrca.py with the command you want to run

```
runOrca.py -p lsst -c "processCcdSdss.py sdss /lsst7/stripe82/dr7-coadds/v5/run0/jbosch_2012_0710_192216/input --output .
/output" -i $HOME/short.input -e /lsst/DC3/stacks/gcc445-RH6/default
```

When you run this command, you'll be told which identifier was created for the run. In the example, this **jbosch_2012_0710_192216**. The command will create directories under $HOME/condor_scratch and (for the LSST platform as it's configured in lsst_ctrl_platform) under /lsst/DC3root, both named jbosch_2012_0710_192216.

This command says to run the command

```
processCcdSdss.py sdss /lsst7/stripe82/dr7-coadds/v5/run0/jbosch_2012_0710_192216/input --output .
/output
```

using ids from the file

```
$HOME/short.input
```

executing out of an LSST stack located in

```
/lsst/DC3/stacks/gcc445-RH6/default
```

on the lsst platform.

IMPORTANT NOTE: This takes the user's current EUPS environment and replicates it on the remote systems where to code is executed. The stack you're pointing to on the remote system must have all the packages available to it that you do when you launch the command.

The results from the HTCondor output are for this run are in $HOME/condor_scratch/jbosch_2012_0710_192216 and the command output is under /lsst /DC3root/jbosch_2012_0710_192216.