

Table of Contents

USE OF THE INFORMATION ON THIS PAGE IS AT YOUR OWN RISK !	1
Notes on running CMSSW on ubuntu hardy heron	2
Running on ubuntu.....	2
Running kerberized CVS.....	2
ssh to lxplus using kerberos token forwarding.....	3
CRAB.....	3
Links.....	3
PYTHON.....	3
readline.....	3
Personal remarks.....	3
Misc.....	4
Mounting windows network filesystems.....	4
Links.....	4
USE OF THE INFORMATION ON THIS PAGE IS AT YOUR OWN RISK !	5

**USE OF THE INFORMATION ON THIS PAGE IS AT
YOUR OWN RISK !**

Notes on running CMSSW on ubuntu hardy heron

- on ubuntu, by default, when running `scramv1`, one sees errors like

```
[ : 71: ==: unexpected operator
```

This is due to the fact that `/bin/sh` is linked to `/bin/dash` instead of `/bin/bash` (see also this link [↗](#)). One can set this back to `bash` by running `sudo dpkg-reconfigure dash` and selecting the appropriate answer.

- `scramv1` complained about missing `Template.pm`. Doing `sudo apt-get install libtemplate-perl` seems to make this warning go away.
- At least with `CMSSW_1_6_12`, `scramv1 b` did not build anything even though it should have. Running `scramv1 --debug b` showed that it wanted to run `gmake` which does not exist on ubuntu. A `softlink ~/bin/gmake -> /usr/bin/make` (where `~/bin` is in the path) made the building work.
- To get from `cmstools import *` or `import readline` to work: After creating the project area (assuming `python2.4` is configured for the CMSSW version), `softlink` it from the local python installation, e.g. by doing:

```
ln -s /usr/lib/python2.4/lib-dynload/readline.so $CMSSW_BASE/lib/$SCRAM_ARCH/
```

Running on ubuntu

- To start `openafs`, do `sudo /etc/init.d/openafs-client start`. If the OpenAFS kernel module does not exist (for the kernel version in use), this will complain.
- To build the OpenAFS kernel module, follow the instructions in `/usr/share/doc/openafs-client/README.modules` which are essentially the following ones:

```
sudo apt-get install module-assistant
sudo module-assistant prepare openafs-modules
sudo module-assistant auto-build openafs-modules
```

You should get a message like: Done with `/usr/src/openafs-modules-....deb`. Install this deb file using `sudo dpkg -i /usr/src/openafs-modules-....deb`.

Note that this needs to be redone when you install a new kernel version.

Running kerberized CVS

- make sure the package `krb5-user` is installed (for `kinit`)
- Make sure `/etc/krb.conf` exists and contains the appropriate lines for CERN. If it doesn't exist, it's probably easiest to copy it from `lxplus/lx32slc4`.
- Copy `/etc/krb5.conf` and `/etc/krb.realms` from `lxplus/lx32slc4` after making a backup of your existing `/etc/krb5.conf` and `/etc/krb.realms`. (yes, it looks like we need `krb5` files despite only getting a ticket for `krb4...`)
- Copy `/usr/bin/cvs` from `lxplus/lx32slc4` and make sure you put it in a directory where it is found before the standard `cvs` (e.g. into `~/bin` and make sure that `~/bin` comes before `/usr/bin/` in your `PATH`).
- Get a token using `kinit -4 xyz@CERN.CH`.

- work with CVS

ssh to lxplus using kerberos token forwarding

- Obtain a kerberos token:

```
kinit xyz@CERN.CH
```

- At least on hardy heron, the ssh client is affected by a [bug](#) with round robin DNS hosts [such as lxplus](#).
 - ◆ In order to have token forwarding, one must use the option `GSSAPIDelegateCredentials yes` (either on the command line or put it in `~/.ssh/config`)
 - ◆ If you login to an explicit lxplus node (e.g. `lxplus123`), you won't be affected by the DNS round robin bug. Otherwise, use the option `GSSAPITrustDNS yes`. See also [this link](#) and [this link](#).

CRAB

- make sure you have a file `/etc/redhat-release`. If not, copy it from lxplus.
- setup the environment (for bash/zsh):

```
source /afs/cern.ch/cms/LCG/LCG-2/UI/cms_ui_env.sh
eval `scramv1 runtime -sh`
source /afs/cern.ch/cms/ccs/wm/scripts/Crab/crab.sh
```

(see also [here](#)).

- in order to get `voms-proxy-init` to work, one needs `libexpat.so.0` (e.g. `/usr/lib/libexpat.so.0` from `lx32slc4=` in my case). I copied this into the directory from which I submit the crab jobs and added `$PWD` at the end of the environment variable `LD_LIBRARY_PATH`.
- see also Storage element interaction in general
- see [here](#) for CERN storage element for CASTOR

Links

- CMSSW in chrooted SL4 (or SL3) system. Nice, but re-mounting the afs filesystem in the chrooted environment is somewhat ugly...
- installing CMSSW rpms on an unsupported distribution

PYTHON

readline

- copy `/usr/lib/libreadline.so.4` from `lx32slc4/lxplus`, into a directory which is contained in `$LD_LIBRARY_PATH$`.

Personal remarks

The following are not related to Ubuntu in general but more to my personal configuration.

- The auto library loader under root (FWLite) seems not to work in my (default) setup, this seems to be due to spurious libraries in the `LD_LIBRARY_PATH`. I get:

```
Error: Symbol G__exception is not defined in current scope (tmpfile):1:
```

Executing the following seems to fix it:

```
export LD_LIBRARY_PATH=$(echo $LD_LIBRARY_PATH | tr ':' '\n' | grep -E "/afs/cern.c
```

Misc.

- For my machine, copy binaries from `lx32slc4`, not from `lxplus`

Mounting windows network filesystems

Assume you want to mount your Windows (NICE) network home directory on your Linux machine. In the following example, the mount point is `/tmp/mnt` (which you must create if it does not exist). Assuming your Nice login name is `johndoe`, try mounting your home directory as:

```
sudo mount -t smbfs -o username='CERN\johndoe' -o uid=$(id -u) -o gid=$(id -g) '//cernhomej/j
```

(note that the servername is `cernhome` + the first letter of Nice login name).

Links

- CMSSW on Ubuntu in the CMS SWGuide

-- AndreHolzner - 10 Jul 2008

USE OF THE INFORMATION ON THIS PAGE IS AT YOUR OWN RISK !

This topic: Main > AndreHolznerCMSSWonUbuntu
Topic revision: r12 - 2009-11-12 - AndreHolzner



Copyright &© 2008-2022 by the contributing authors. All material on this collaboration platform is the property of the contributing authors.
or Ideas, requests, problems regarding TWiki? use [Discourse](#) or [Send feedback](#)