Table of Contents

Building LHCb Software on OS X

Introduction

This is a short guide to building the LHCb software stack on OS X. For the moment this means OS X 10.5 (Leopard). OS X 10.6 (Snow Leopard) has not yet been tested.

Prerequisites

As there are no tarballs (yet) for OS X, and since building all the LCG externals from source is a lot of work, we are going to use the LCG externals from AFS where the OS X 10.5 gcc 4.0.1 binaries are provided. You can get AFS from http://www.openafs.org/ . Note that this requirement implicitly means you must have an active network connection whilst using the software.

We are also going to make extensive use of the CERN CVS and SVN repositories. Therefore you will need to setup remote access (just as you would on any SLC installation outside CERN). See CVSUsageGuidelines and SVNUsageGuidelines for details on how to do this.

Software Environment

The basic CMT environment is set up on OS X in exactly the same way as on linux.

```
  > source /afs/cern.ch/lhcb/software/releases/LBSCRIPTS/prod/InstallArea/scripts/LbLogin.sh
```

Which, all going well, should give you some output similar to

```
Configuring LbScripts v4r3 from /afs/cern.ch/lhcb/software/releases/LBSCRIPTS/LBSCRIPTS_v4r3
*********************************************************************************
*                           ---- LHCb Login v4r3 ----                           *
*           Building with gcc401 on osx105_ia32 system                          *
*********************************************************************************
 --- CMTROOT is set to /afs/cern.ch/sw/contrib/CMT/v1r20p20090520
 --- CMTCONFIG is set to osx105_ia32_gcc401
 --- User_release_area is set to /Users/jonrob/cmtuser
 --- CMTPROJECTPATH is set to $User_release_area:$LHCb_release_area:$Gaudi_release_area:$LCG_rele
 --- projects will be searched in $CMTPROJECTPATH
---------------------------------------------------------------------------------
Using CMTPROJECTPATH = '/afs/cern.ch/lhcb/software/releases:/afs/cern.ch/sw/Gaudi/releases:/afs/c
Environment for LbScripts v4r3 ready.
(Compat v1r3 from /afs/cern.ch/lhcb/software/releases/COMPAT/COMPAT_v1r3,
 LbScripts v4r3 from /afs/cern.ch/lhcb/software/releases/LBSCRIPTS/LBSCRIPTS_v4r3,
 LCGCMT 57a from /afs/cern.ch/sw/lcg/app/releases/LCGCMT/LCGCMT_57a,
 COMPAT v1r3 from /afs/cern.ch/lhcb/software/releases/COMPAT/COMPAT_v1r3)
```

Building A Software Release

Now we are ready to build a release. For the purposes of this guide I will use DaVinci v24r4, but the procedure is similar for any other release or project. Note that at present Gauss and Panoramix are problematic since they make much more extensive use of external packages. I recommend staying away from these at present, unless **very** keen.

First, set up your DaVinci release area

```
  > SetupProject DaVinci v24r4 --build-env
```

Next, we are going to recursively getpack all the LHCb/Gaudi packages ! This is actually easier than it sounds. You just need to know the release versions of each project (which can be got from the releases notes for DaVinci).

```
> getpack -r GaudiRelease v21r5
> getpack -r LHCbSys v28r1
> getpack -r LbcomSys v7r7
> getpack -r HltSys v7r1
> getpack -r RecSys v7r7
> getpack -r AnalysisSys v3r11
> getpack -r PhysSys v8r11
```

There are other slightly different ways to do the above. One reason I do it this way is I prefer to place each project in its own private release area, since then you can share projects that are used by more than more higher level projects. However, this is a slightly more complicated use case which for simplicity I don't go into here.

The next step is to try and build everything.

```
> cd ~/cmtuser/DaVinci_v24r4/Phys/DaVinci/cmt
> cmt br cmt make
```

Which will take some time. If you have a multi-core machine, try using `make -j X` where X is the number of cores.

Note that whilst in practise most things will build just fine, there is no guarantee of this. If something does fail it is usually though quite easy to fix. If you are not sure just contact me (ChrisRJones).

All going well you will eventually have a working DaVinci application.

Note, before you can use your shiny new DaVinci build, you will need to setup the runtime environment with

```
> SetupProject DaVinci v24r4
```

Just as you would on any other platform.

One final comment. Due to the way `SetupProject` strips empty paths from `PYTHONPATH` and `LD_LIBRARY_PATH` this has to be done after building (since then your private `InstallArea` is no long empty).

Known Issues

- At the moment castor access to data does not work. Just copy a few files locally.
- Nothing has been tested on Snow Leopard as yet. As the LCG area does not provide binaries for the native compiler on OS X 10.6 (gcc 4.2) the best bet at the moment might be to try and pretend to be OS X 10.5, by manually setting the `CMTCONFIG` environment variable to `osx105_ia32_gcc401`. I suspect this will not be enough, and some additional work will then be needed to convince CMT to use GCC 4.0.1.

-- ChrisRJones - 11-Nov-2009

This topic: LHCb > LHCbSoftOnOSX
Topic revision: r4 - 2010-01-22 - MarcoCattaneo