Using CMT

Murrough Landon – 4 February 2002

`http://www.hep.ph.qmul.ac.uk/~landon/talks`

Overview

- Introduction
- Progress so far
- CMT basics
- Directory layouts
- Documentation
- Next steps
- PS

Introduction

- ATLAS has now officially chosen CMT for Offline computing
- The DataCollection group has been using it for a while
- The Online group is converting to CMT and will(?) make its next release with CMT
- CMT will be useful to us as we develop more packages
- Now (or earlier!) is the time to move before we get set in the old ways

Progress so far

- Using latest version of CMT: v1r10p20011126 (!)
- Adopted package directory layout from the Online group
- Copied/modified various scripts, mostly from the DataCollection group into a new `scripts` package
- Created/copied some CMT control packages
- Created wrapper for HDMC as an external package (for the moment)
- Moved all my packages into CMT (renaming them with L1Calo suffixes to avoid clashes with Online packages)
- Tested checkout, compile, link at QMW and RAL – all OK.
- End result: run controllers still work (at QMW).

Scripts package

CMT related

- `getpkg`: fetch a single named package
- `getallpkg`: fetch all our packages, uses:
- `pkginfo.pl`: info about all our packages
- `latest`: finds latest tagged version from CVS (requires special CMT hook in CVS repository)

Others

- `fetchonline.pl`: fetch Online software version
- `fetchexternal.pl`: fetch external libs for Online
- `downloadcmt.pl`: download and install CMT
- `setup.sh`: setup script for L1Calo software

CMT Basics (1)

Package structure

- Directory structure is `packageName/packageTag/subdirs`
- CMT control files live in `cmt` subdirectory.
- The `cmt/requirements` file specifies which applications, libraries and other documents are provided by the package.
- The `cmt/requirements` file is used to generate Makefile fragments.
- You can also edit the standard Makefile for each package and declare fragments to be used by all packages. Eg the Online group have written Makefile fragments for installing all packages into a common area and some special handling for Java.
- Packages can `use` other packages and inherit all their settings, special fragments, etc.

CMT Basics (2)

CMT command

- `cmt config` configure a package (once).
- `make` make the current package (from `cmt` subdirectory).
- `cmt broadcast <cmd>`: execute `cmd` for the current package and all its used packages. Eg `cmt broadcast make` run from the L1CaloRelease package will make all our packages.
- Many others...

Directory layout: package

Typical structure of an Online package (though they vary a bit).

`.../<pkg>/<pkg-tag>/`

`/cmt`	CMT control directory
`/<pkg>`	C++ header files
`/src`	C++ source or libraries
`/jsrc`	Java source for jar files
`/bin`	Source for applications
`/test`	Source for test programs
`/dox`	Files for Doxygen
`/linux-gcc`	Binaries via gcc
`/linux-egcs`	Binaries via egcs

Directory layout: users work area

You need to check out all packages into a work area (eg `~/l1calo`):

`workarea`	`/L1CaloPolicy`	common setup for all our packages
	`/L1CaloRelease`	uses all our other packages
	`/OnlinePolicy`	common setup for Online packages
	`/confL1Calo`	config/etc database libraries
	`/dbFiles`	database schema and data files
	`/hdmcExternal`	wrapper package for HDMC
	`/iguiL1Calo`	panels for the IGUI (java)
	`/isL1Calo`	generated code of IS variables
	`/rcL1Calo`	run controllers
	`/installed`	installation area
	`/scripts`	utility scripts (not a CMT package)

Directory layout: installation area

The Online group install all required applications, libraries, headers and other files from all package binary directories to a single common installation area:

`installed`	`/include`		include files
		`pkg1`	headers for pkg1
		`pkg2`	headers for pkg2
	`/share`		arch/compiler indep files
	`/java`		jar files
	`/linux-gcc`		arch/compiler dep files
		`/bin`	applications
		`/lib`	libraries
		`/include/pkg`	arch dep headers
	`/linux-egcs`		arch/compiler dep files

Documentation

The CMT manual is available at the CMT website:

`http://www.lal.in2p3.fr/SI/CMT/CMT.htm`

The Online group have a detailed web page describing their setup:

`http://akazarov.home.cern.ch/akazarov/cmt`

I have added a CMT page to our software website:

`http://www.hep.ph.qmul.ac.uk/l1calo/sweb/software/cmt.html`

and one detailing how to set up the L1Calo software:

`http://www.hep.ph.qmul.ac.uk/l1calo/sweb/software/setup.html`

(but I have not yet updated software note 008).

List of our packages with links to a page for each one:

`http://www.hep.ph.qmul.ac.uk/l1calo/sweb/packages`

Descriptions of the scripts in the scripts package:

`http://www.hep.ph.qmul.ac.uk/l1calo/sweb/packages/scripts.html`

Next steps

- Work out how to write CMT style make fragments to cope with Qt. Need to generate (and compile) `moc` files. This is required for HDMC. Would also like to generate C++ from Qt designer (unless/until noddy trigger editor is abandoned).
- Move other packages (simulation, testVectors, moduleServices)
- Also HDMC? Especially if it is to be made available as part of ROD crate DAQ.

PS: News from the Online group

- Current release, 0.0.16, works OK. On RedHat 7.2, hostname command always returns the short name which must be the one in the list of workstations in the database.
- Additional functionality in `is_generator` provided for me in a special patch (will be in the next release).
- They are reviewing all their user requirements documents. These are mostly very short and snappy – quick to write similar things for our packages?!
- An Online task force to discuss the requirements for the conditions database will be set up soon.