

Table of Contents

Instructions for creating a new CMT package.....	1
Choose a name and location for your package.....	1
Work in the appropriate project directory.....	1
Create the package root directory.....	1
Create the package directory structure.....	1
Edit requirements (or copy them from a similar package...) and configure the package.....	1
Write some code and build a library.....	1
Import the new package to CVS.....	2
Add your new package in the list of CVS modules.....	2
Check that everything is OK.....	2
Finally, add the new package to the project.....	2

Instructions for creating a new CMT package

These instructions are for creating a new package in CVS. Since all CVS packages will soon be migrated to SVN, it is recommended to create new packages directly in SVN.

Choose a name and location for your package

Your package should have a unique name (unique to the LHCb software), so choose a name that is both descriptive and specialised: `Components` is a bad name, `FredComponents` is good. You should also choose an LHCb subsystem ("Hat") among existing ones, e.g. `Phys` if your components are for doing a physics analysis. Finally, you must choose a "Project" among existing ones within which your package should be released, e.g. `Analysis`

In what follows, let us assume you want to write a components package called `FredComponents` for subsystem `Phys`, in `Analysis` project

Work in the appropriate project directory

```
SetupProject --build-env Analysis
```

You will be prompted for a version number, choose the most recent one. This puts you in the directory `$User_release_area/Analysis_<version>`

Create the package root directory

```
mkdir -p Phys/FredComponents
cd Phys/FredComponents
```

Create the package directory structure

For a components package, just `cmt`, `doc`, `src` are needed (no public includes)

```
mkdir cmt doc src
```

Edit `requirements` (or copy them from a similar package...) and configure the package

```
cd doc
emacs release.notes
```

Add some meaningful `release notes` to explain the purpose of the package, the author etc...

```
cd ../cmt
emacs requirements
```

Edit the `requirements` file.

```
cmt config
```

Configure the package

Write some code and build a library

```
cd ../src
```

Add `.cpp` files

```
cd ../cmt
cmt make
```

component library is built

Import the new package to CVS

Note: the `CVSROOT` variable is not defined in the default LHCb environment. It has therefore to be added explicitly to commands which are not using the local CVS/Root file, using the `-d CVS` option. This is the case for example for `cvs import` or plain `cvs checkout`

Important: Please make sure that the `import` command is run from inside the directory you want to import:

```
cd $User_release_area/Analysis_<version>/Phys/FredComponents
```

Remove first all files you do not wish to import: binary directories, backup copies (`~`), generated files in `cmt` directory (only requirements is needed)

```
rm -r x86
rm / ~
rm cmt/ .
rm cmt/ Make
```

And now import to CVS

```
cvs -d :ext:isscvcs.cern.ch:/local/reps/lhcb import -m " first import of FredComponents
under Phys" Phys/FredComponents myid v0r0
```

`myid` is a "vendor-tag" in the CVS terminology. It is recommended to use your AFS user name.

Add your new package in the list of CVS modules

```
cd $User_release_area
cvs -d :ext:isscvcs.cern.ch:/local/reps/lhcb co CVSROOT/modules
cd CVSROOT
emacs modules
```

Add your package in the list:

```
Phys/FredComponents      Phys/FredComponents
```

And put the changes into CVS

```
cvs -d :ext:isscvcs.cern.ch:/local/reps/lhcb commit -m "add Phys/FredComponents in the
list of CVS modules"
cd ..
rm -r CVSROOT
```

Check that everything is OK

First rename the original package

```
cd $User_release_area/Analysis_<version>/Phys
mv FredComponents _FredComponents
```

Then check out your package from CVS

```
cd $User_release_area/Analysis_<version>
getpack Phys/FredComponents head
cd Phys/FredComponents/cmt
cmt show uses
cmt make
```

If everything looks fine you can remove the original version

```
cd ../../
rm -r _FredComponents
```

Finally, add the new package to the project

```
cd $User_release_area/Analysis_<version>
getpack AnalysisSys head
cd AnalysisSys
emacs cmt/requirements
```

Add the new package:

```
use FredComponents v Phys
```

Update the release notes

```
emacs doc/release.notes
```

Commit and tag the changes

```
cvs -d :ext:isscvcs.cern.ch:/local/repos/lhcb commit -m "added package Phys/FredComponents"
```

Then tag and add to the tag collector

-- MarcoCattaneo - 2009-09-14

This topic: LHCb > CreateNewPackage

Topic revision: r7 - 2011-06-22 - PatrickSKoppenburg



Copyright &© 2008-2019 by the contributing authors. All material on this collaboration platform is the property of the contributing authors.

Ideas, requests, problems regarding TWiki? Send feedback