Table of Contents

Instructions for creating a new Package in SVN

Video example (for the text below)

Example recorded for creating the new package RawEventFormat to sit in DBASE, using CMT. Note that a couple of the later svn commands contain typos, but you should get the general idea.

Instructions for creating a new Package

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 specialized: `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. Please note that the **core name (without the hat) of the package has to be unique** among the LHCb package list. You can find the full LHCb package list with the command

```
svn pg packages svn+ssh://svn.cern.ch/reps/lhcb
```

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>`

Make the package

Option 1: using CMT

```
cmt create FredComponents v1r0 Phys/
cd  Phys/FredComponents
mkdir doc
```

Option 2: personally

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

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

```
mkdir cmt
mkdir doc
mkdir src
```

Edit `requirements` and `CMakeLists.txt` (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, make sure the directories src cmt and doc are specified for your components package

```
cd ..
emacs CMakeLists.txt
```

Edit the CMakeLists.txt file

```
cd cmt
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 Subversion

IMPORTANT : Before you import anything into SVN, please discuss your new package with the Manager of your application (Brunel, DaVinci, Gauss, etc.) first, to decide the most appropriate project to commit your new package to.

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 $CMTCONFIG
rm -r genConf
rm  / ~
rm cmt/ .
rm cmt/ Make
```

And now import to Subversion

```
svn import -m "first import of FredComponents under Phys"
svn+ssh://svn.cern.ch/reps/lhcb/Analysis/trunk/Phys/FredComponents
svn mkdir -m "create tags and branches directories for Phys/FredComponents"
svn+ssh://svn.cern.ch/reps/lhcb/Analysis/tags/Phys/FredComponents
svn+ssh://svn.cern.ch/reps/lhcb/Analysis/branches/Phys/FredComponents
```

Note: if the `mkdir` command fails, you may need to create the intermediate directories (e.g. the hat "Phys") one by one. A version of the command line client more recent than the one on lxplus5 has the option "--parents" to simplify the process.

Note: Before creating the SVN directory in the "tags" subdirectory, the package must have been imported to the trunk. If not the pre-commit hook will reject the mkdir, with an error message: **Failure: Invalid operation on a tag**.

Add your new package in the list of known packages in the repository

On a machine where the pico editor is installed (e.g. lxplus, but not lxbuild)

```
cd $User_release_area
svn co -N svn+ssh://svn.cern.ch/reps/lhcb lhcb_repos
svn pe packages lhcb_repos
```

⚠ It's **mandatory** to check out a **fresh copy** of `lhcb_repos` or update the working copy with `svn update` **before** editing the property.

Add your package in the list:

Edit requirements and CMakeLists.txt (or copy them from asimilar package...) and configure the package

```
Phys/FredComponents Analysis
```

and commit the changes

```
svn ci -m "add Phys/FredComponents in the list of packages" lhcb_repos
rm -rf lhcb_repos
```

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 the repository

```
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
```

Tag the version as v1r0:

```
tag_package Phys/FredComponents v1r0
```

Finally, the new package can be added to the project

Inform the project manager that the new package is ready to be added to the project itself. Then add to the tag collector

Instructions for creating a new Data Package

Package creation

The constraints (unicity of the name etc) and the instructions are essentially the same for a normal package except that:

- You can create the package in the Application that will be using the package directly.

Once the package has been created and is tested, you can import the package in the project of your choice.

Another difference between data packages and code packages, is that in data packages there is no need for a `CMakeLists.txt`, but you must provide a file describing the environment required to use the package. The environment is described in an XML file with the name derived from the name of the package like `Hat_Package.xenv`, for example:

- `ParamFiles` needs `ParamFiles.xenv`
- `Det/SQLDDDB` needs `Det_SQLDDDB.xenv`

You can find examples in LbRelease/data/DataPkgEnvs.

Important: the changes required in the CMake configuration of a project to use a data package are different to those for regular packages. For a data package, you have to update the project CMakeLists.txt to extend the `DATA` section of the call to `gaudi_project` (or add one if it was missing). For example (from LHCb):

```
gaudi_project(LHCb v35r3
              USE Gaudi v23r5
              DATA Det/SQLDDDB VERSION v7r*
                   FieldMap
                   TCK/HltTCK)
```

Import the new package to Subversion

IMPORTANT : Before you import anything into SVN, please discuss your new package with the Core Software Team first, to decide the most appropriate project to commit your new package to: DBASE or PARAM (note that for PARAM, the SVN directory is called Param - i.e. in the examples below, replace occurrences of 'DBASE' by 'Param')

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

```
cd $User_release_area/Project_<version>/WG/MyDataPackage
```

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 $CMTCONFIG
rm -r genConf
rm  / ~
rm cmt/ .
rm cmt/ Make
```

And now import to Subversion, for example under DBASE

```
svn import -m "first import of MyDataPackage under DBASE"
svn+ssh://svn.cern.ch/reps/lhcb/DBASE/trunk/WG/MyDataPackage
svn mkdir -m "create tags and branches directories for WG/MyDataPackage"
svn+ssh://svn.cern.ch/reps/lhcb/DBASE/tags/WG/MyDataPackage
svn+ssh://svn.cern.ch/reps/lhcb/DBASE/branches/WG/MyDataPackage
```

Note: if the `mkdir` command fails, you may need to create the intermediate directories (e.g. the hat "WG") one by one. A version of the command line client more recent than the one on lxplus5 has the option "--parents" to simplify the process.

Add your new package in the list of known packages in the repository

As for normal packages

Check that everything is OK

As for normal packages

Adding the new Data package to

Send an email to Ben (ben.couturier@cernNOSPAMPLEASE.ch) in order to add the package to LbScripts

Make the package available in the nightlies

(example is for a `DBASE` package, `PARAM` is the same)

```
cd $LHCBDEV/nightlies/DBASE
getpack WG/MyDataPackage HEAD
```

If you don't have write access, ask a release manager to do it for you. When the package is officially released, it's a good idea to remove this checkout and replace it with a soft link to the release area.

-- MarcoCattaneo - 09-Jul-2013 -- MarcoClemencic - 16-Dec-2009
