Table of Contents

SetupProject User Guide

Introduction

SetupProject has been introduced to address the need of a tool to prepare a customized environment in which run an LHCb application.

Since its initial version (end of July, 2006), the *simple* tool grew in functionality and flexibility to address more and more use cases.

Disclaimer: all the versions of the projects presented in this page are those available at the moment of writing and there is no implicit or explicit suggestion about which version to use or not to use. They are just examples.

Basic Usage

The simplest use-case one can imagine, is the configuration of the runtime environment for a released version of a project, so that one can call directly the application passing the preferred options, without having to check out from CVS the top level package of the application.

To be able to use SetupProject, you need the minimal LHCb environment that you get when you log on lxplus.cern.ch or after having sourced the ExtCMT script in a local installation.

The minimal information that you need to pass to SetupProject is the name of the application (or project) you want to use. Obviously you may want to specify the version, but if you do not do it, SetupProject will use the latest available. An example is:

```
[lxplus] ~ > SetupProject DaVinci
Configuring DaVinci v20r0 from /afs/cern.ch/lhcb/software/releases/DAVINCI/DAVINCI_v20r0
Using CMTPROJECTPATH = '/afs/cern.ch/user/m/marcocle/cmtuser:/afs/cern.ch/lhcb/software/releases:
Environment for DaVinci v20r0 ready.
(taken from DaVinci v20r0 from /afs/cern.ch/lhcb/software/releases/DAVINCI/DAVINCI_v20r0)
```

As you can see, the output is pretty self explanatory. You have the version of the application, the `CMTPROJECTPATH` used to locate dependent projects and where the binaries are actually taken from.

To specify the version of the application that you need, just add it to the command line:

```
[lxplus] ~ > SetupProject DaVinci v19r14
Configuring DaVinci v19r14 from /afs/cern.ch/lhcb/software/releases/DAVINCI/DAVINCI_v19r14
Using CMTPROJECTPATH = '/afs/cern.ch/user/m/marcocle/cmtuser:/afs/cern.ch/lhcb/software/releases:
Environment for DaVinci v19r14 ready.
(taken from DaVinci v19r14 from /afs/cern.ch/lhcb/software/releases/DAVINCI/DAVINCI_v19r14)
```

If you do not remember or do not know which versions are available, you have two choices: you can tell SetupProject to list the versions it can use or let it propose the list from which to chose a version. The list is produced with the command line option `--list-versions`, that can be abbreviated with `--list`:

```
[lxplus] ~ > SetupProject DaVinci --list
v12r18 in /afs/cern.ch/lhcb/software/releases
v14r5 in /afs/cern.ch/lhcb/software/releases
v19r5 in /afs/cern.ch/lhcb/software/releases
v19r6 in /afs/cern.ch/lhcb/software/releases
v19r7 in /afs/cern.ch/lhcb/software/releases
v19r8 in /afs/cern.ch/lhcb/software/releases
v19r9 in /afs/cern.ch/lhcb/software/releases
```

```
v19r10 in /afs/cern.ch/user/m/marcocle/cmtuser
v19r11 in /afs/cern.ch/lhcb/software/releases
v19r12 in /afs/cern.ch/lhcb/software/releases
v19r13 in /afs/cern.ch/user/m/marcocle/cmtuser
v19r14 in /afs/cern.ch/lhcb/software/releases
v20r0 in /afs/cern.ch/lhcb/software/releases
```

(you can notice that most of the proposed versions come from the standard release area, while two come from my cmtuser directory, the reason of it is explained later in this page). If you prefer to chose the version from a list of possibilities, the command line option is `--ask`:

```
[lxplus] ~ > SetupProject DaVinci --ask
Please enter your choice (v12r18 v14r5 v19r5 v19r6 v19r7 v19r8 v19r9 v19r10 v19r11 v19r12 v19r13
Trying version 'v19r8'
Configuring DaVinci v19r8 from /afs/cern.ch/lhcb/software/releases/DAVINCI/DAVINCI_v19r8
Using CMTPROJECTPATH = '/afs/cern.ch/user/m/marcocle/cmtuser:/afs/cern.ch/lhcb/software/releases:
Environment for DaVinci v19r8 ready.
(taken from DaVinci v19r8 from /afs/cern.ch/lhcb/software/releases/DAVINCI/DAVINCI_v19r8)
```

In this case, at the prompt, you can press enter without typing anything to accept the default (equivalent to a call to SetupProject without explicit version), type `q` to avoid changes to your environment or select a different version.

You may need to try versions that are not in production yet, like the build available in the special release are called LHCBDEV. Since SetupProject doesn't propose those version by default to avoid possible problems, you can enable LHCBDEV with the command line option `--dev`:

```
[lxplus] ~ > SetupProject Panoramix --list
v9r8 in /afs/cern.ch/lhcb/software/releases
v15r8 in /afs/cern.ch/lhcb/software/releases
v15r9 in /afs/cern.ch/lhcb/software/releases
v15r10 in /afs/cern.ch/lhcb/software/releases
v15r11 in /afs/cern.ch/lhcb/software/releases
v15r13 in /afs/cern.ch/lhcb/software/releases
v15r14 in /afs/cern.ch/lhcb/software/releases
v15r15 in /afs/cern.ch/lhcb/software/releases
[lxplus] ~ > SetupProject Panoramix --dev --list
v9r8 in /afs/cern.ch/lhcb/software/releases
v15r8 in /afs/cern.ch/lhcb/software/releases
v15r9 in /afs/cern.ch/lhcb/software/releases
v15r10 in /afs/cern.ch/lhcb/software/releases
v15r11 in /afs/cern.ch/lhcb/software/releases
v15r13 in /afs/cern.ch/lhcb/software/releases
v15r14 in /afs/cern.ch/lhcb/software/releases
v15r15 in /afs/cern.ch/lhcb/software/releases
v15r16 in /afs/cern.ch/lhcb/software/DEV
v16r0 in /afs/cern.ch/lhcb/software/DEV
```

Note that the version that would be taken by default is the last one of the list.

There is another way of using a DEV directory, `--dev-dir`, which allows you to specify something different from the standard LHCBDEV or more than one "dev" directory (not a very common use case, I agree). An example of a dev directory is the location of the LHCb nightly builds:

```
[lxplus] ~ > SetupProject Brunel --dev-dir /afs/cern.ch/lhcb/software/nightlies/lhcb2/Wed
Configuring Brunel HEAD from /afs/cern.ch/lhcb/software/nightlies/lhcb2/Wed/BRUNEL/BRUNEL_HEAD
Using CMTPROJECTPATH = '/afs/cern.ch/lhcb/software/nightlies/lhcb2/Wed:/afs/cern.ch/user/m/marcoc
Environment for Brunel HEAD ready.
(taken from Brunel HEAD from /afs/cern.ch/lhcb/software/nightlies/lhcb2/Wed/BRUNEL/BRUNEL_HEAD)
```

User Local Projects

The supported way (within LHCb) of developing software in the context of an application/project is to create a "user project" in which you can check out packages from CVS or create your new ones. You do not need a deep knowledge of the internal structure of projects managed with CMT, SetupProject will do the dirty job of creating the project directory if you use the option `--build-env`. You have to specify the name of the project and the version you want to use. If the version is omitted, SetupProject will ask which one to use (this differs from the behavior described before to maintain the same behavior of a previous tool used for the same purpose). Example:

```
[lxplus] ~ > SetupProject LHCb --build-env v25r0
Build-time environment for LHCb v25r0 ready.
Created user project in /afs/cern.ch/user/m/marcocle/cmtuser/LHCb_v25r0
Current directory is '/afs/cern.ch/user/m/marcocle/cmtuser/LHCb_v25r0'.
Using CMTPROJECTPATH = '/afs/cern.ch/user/m/marcocle/cmtuser:/afs/cern.ch/lhcb/software/releases:
[lxplus] ~/cmtuser/LHCb_v25r0 >
```

After the call to SetupProject, you will be in the directory of the local project, ready to call `getpack` or to create your package. If the project directory is already present, it will not be modified (you can tell the difference because SetupProject will not print the line `Created user project in ...`). The location in which the user project is created is defined by the variable `User_release_area`, if you want the project to be created somewhere else, just change its value (the default is `~/cmtuser`).

It must be clear that when called with `--build-env`, SetupProject cannot prepare a runtime environment. The reason is simple: if you want to build some code, most probably the libraries you may want to use do not exist yet in your local project, so there is nothing you can use yet.

To prepare the runtime environment using your user project, you don't need to do anything special: SetupProject always look for projects in your `User_release_area`, as you can notice with --list:

```
[lxplus] ~ > SetupProject LHCb --list
v23r5 in /afs/cern.ch/lhcb/software/releases
v23r6 in /afs/cern.ch/lhcb/software/releases
v23r7 in /afs/cern.ch/lhcb/software/releases
v24r0 in /afs/cern.ch/lhcb/software/releases
v24r1 in /afs/cern.ch/lhcb/software/releases
v25r0 in /afs/cern.ch/user/m/marcocle/cmtuser
```

The fact that the user project is used, is also clear when setting up the runtime environment:

```
[lxplus] ~ > SetupProject LHCb v25r0
Configuring LHCb v25r0 from /afs/cern.ch/user/m/marcocle/cmtuser/LHCb_v25r0
Using CMTPROJECTPATH = '/afs/cern.ch/user/m/marcocle/cmtuser:/afs/cern.ch/lhcb/software/releases:
Environment for LHCb v25r0 ready.
(taken from LHCb v25r0 from /afs/cern.ch/user/m/marcocle/cmtuser/LHCb_v25r0)
```

If you do not want that your local project is picked up, you have to unset `User_release_area` (or to rename the user project directory).

Advanced Usage

Use Non-standard Packages

When SetupProject is asked to prepare the runtime environment for a project, it uses all the packages that belong to the requested project plus those that are used by them, directly or indirectly. This means that setting up *Brunel* will not give you access to all the packages in *LHCb*.

There is at least one case where the behavior could be annoying: the CondDBBrowser. The graphical user interface to the Conditions Database needs some libraries that are made available only if you use the package `Tools/CondDBUI`, which is part of *LHCb*. Of course, you can do `SetupProject LHCb` and get the browser, but you may be working with Panoramix and do not want to leave of modify (maybe corrupt) the environment to be able to start the browser, so SetupProject gives you the possibility of getting the environment specific to packages that are not in the project with the option `--use`:

```
[lxplus] ~ > SetupProject Panoramix --use Tools/CondDBUI
Configuring Panoramix v15r15 from /afs/cern.ch/lhcb/software/releases/PANORAMIX/PANORAMIX_v15r15
Using CMTPROJECTPATH = '/afs/cern.ch/user/m/marcocle/cmtuser:/afs/cern.ch/lhcb/software/releases:
Environment for Panoramix v15r15 ready.
(taken from Panoramix v15r15 from /afs/cern.ch/lhcb/software/releases/PANORAMIX/PANORAMIX_v15r15)
[lxplus] ~ > echo $CONDDBUIROOT
/afs/cern.ch/lhcb/software/releases/LHCB/LHCB_v24r0/Tools/CondDBUI/v2r10
```

The usefulness of `--use` is not limited to add a package to the list of the used ones. With special projects like DBASE and PARAM, you have many versions of a package available at the same time, but only the latest one us used. If you need a special version of one of these packages, for example the version of `DecFiles` has to be fixed for productions of simulated data, you can add it to the use parameter:

```
[lxplus] ~ > SetupProject Gauss v33r2 --use "Gen/DecFiles v14r4"
Configuring Gauss v33r2 from /afs/cern.ch/lhcb/software/releases/GAUSS/GAUSS_v33r2
Using CMTPROJECTPATH = '/afs/cern.ch/user/m/marcocle/cmtuser:/afs/cern.ch/lhcb/software/releases:
Environment for Gauss v33r2 ready.
(taken from Gauss v33r2 from /afs/cern.ch/lhcb/software/releases/GAUSS/GAUSS_v33r2)
[lxplus] ~ > echo $DECFILESROOT
/afs/cern.ch/lhcb/software/releases/DBASE/Gen/DecFiles/v14r4
```

Please, note the quotes around `"Gen/DecFiles v14r4"`: they are needed.

More Advanced usage

Quick Reference

Set up the runtime environment for a released application
- `SetupProject DaVinci`
- `SetupProject Brunel v33r1`
- `SetupProject Panoramix --ask`

Set up the runtime environment for an application in the nightly builds
- `SetupProject DaVinci --nightly lhcb2`
- `SetupProject DaVinci --nightly lhcb2 Wed`

Prepare a user local project
- `SetupProject --build-env LHCb v25r0`
- `setenvLHCb v25r0`
  (an alias to the previous command)

Custom Project Search Path (CMTPROJECTPATH)

Since beginning of October 2008, SetupProject does not use anymore the defined CMTPROJECTPATH (the environment variable used to declare where to find projects), but it always constructs it from `User_release_area` (by default `~/cmtuser`), `LHCBPROJECTPATH` (the default basic project search path) and the command line.

The reason for the change of behavior is needed because the old behavior could lead to some inconsistencies across calls to SetupProject. Of course, something similar to the old behavior is still possible with the command line option `--keep-CMTPROJECTPATH`.

Here I describe how to have SetupProject constructing the CMTPROJECTPATH for you. The basic logic is that the minimal `CMTPROJECTPATH` is given by the environment variable `LHCBPROJECTPATH`, the options to modify the `CMTPROJECTPATH` are prepended to it and the `User_release_area` must always be the first one to be searched. The options that modify the `CMTPROJECTPATH` are:

- `--dev`
- `--dev-dir <dir>`
- `--nightly <slot> [<day>]`

The basic call to SetupProject, without any special option will set `CMTPROJECTPATH` to `${User_release_area}:${LHCBPROJECTPATH}`.

The option `--dev` prepends to the `CMTPROJECTPATH` the content of the environment variable `LHCBDEV`, so that a call

```
SetupProject LHCb --dev
```

will set `CMTPROJECTPATH` to `${User_release_area}:${LHCBDEV}:${LHCBPROJECTPATH}`.

`--dev-dir` is used to specify a different "dev" area, so `--dev` is actually equivalent to `--dev-dir ${LHCBDEV}`. The call

```
SetupProject LHCb --dev-dir /afs/cern.ch/user/m/marcocle/public
```

will result in the value `${User_release_area}:/afs/cern.ch/user/m/marcocle/public:${LHCBPROJECTPATH}` for `CMTPROJECTPATH`.

The option `--nightly` allows to prepare the environment to use the standard LHCb nightly build directories. It accepts a mandatory argument defining the slot to use (e.g. `lhcb1` or `lhcb2`) and an optional argument for the day (3 chars abbreviation), which is defaulted to the current day. With this option, SetupProject checks the existence of the configuration file of the nightly build and uses the information in it to prepare the `CMTPROJECTPATH` (needed for some special configuration of the nightly build slot). The call

```
SetupProject LHCb --nightly lhcb1
```

will set `CMTPROJECTPATH` to `${User_release_area}:/afs/cern.ch/lhcb/software/nightlies/lhcb1/Fri:...:${LHCBPROJECTPATH}` (assuming today is Friday), where the `...` will contain the special entries required by the slot.

The options described can be combined for a more complex `CMTPROJECTPATH`. The specified directories are added to the `CMTPROJECTPATH` in the order specified on the command line, after `User_release_area` and before `LHCBPROJECTPATH`. So from

```
SetupProject LHCb --dev-dir /afs/cern.ch/user/m/marcocle/public --nightly lhcb1 Mon --dev --dev-d
```

you should expect `CMTPROJECTPATH` featuring, in this order,

- `${User_release_area}`
- `/afs/cern.ch/user/m/marcocle/public`
- `/afs/cern.ch/lhcb/software/nightlies/lhcb1/Mon`
- `${LHCBDEV}`
- `/afs/cern.ch/user/s/somebody/public`
- `${LHCBPROJECTPATH}`.

Caveats

There are 2 main usages for SetupProject: the fist one is to setup the full environment of a project and the second one is to prepare the environment for the software development. This second usage which is performed by using the `--build-env` option (or equivalently by using the alias setenvProject) doesn't setup the full environment. It only updates the environment variable LHCBPROJECTPATH.

This is all you need to build your software. For example, here is a normal usage of SetupProject for both development and run:

1. Prepare the environment which the custom directory from the nightly builds:

```
SetupProject --build-env --nightly lhcb2 DaVinci v24r0  (setenvProject --night
```

2. checkout the package you would like to modifiy

```
getpack MyPack
```

3. Modify your package
4. Build it

```
cd  MyPack/cmt
cmt make
```

5. Do the full setup to run it

```
SetupProject --nightly lhcb2 DaVinci v24r0
```

It is important that the build is done before the full setup is done. Otherwise you could end up with a package that doesn't build in the normal environment.

Bug reporting

Please report bugs to the SetupProject category of the lhcbscripts Savannah bug tracker

Appendix

SetupProject online help

This is the help message you can get calling `SetupProject --help`:

```
Usage: SetupProject.py [options] <project_name> [version|--ask] [options] [externals]

Options:
  --version             show program's version number and exit
  -h, --help            show this help message and exit
  --site SITE           enable site specific defaults
  --ask                 ask for the version of the project to use (overrides
                        the version specified)
  --disable-CASTOR      remove CASTOR from the added dependencies
  --tag_add TAG_ADD     specify extra CMT tags
  --use USE             add a CMT use statement
  --verbose             be a bit more verbose
  --debug               output useful for debugging
  --ignore-missing      do not fail if some externals are missing, just
                        complain
  --ignore-context      do not use CMTUSERCONTEXT even if it should be used
  --list-versions       print available versions of the specified project and
                        exit (all other options are ignored)
  --build-env           sets only the build time environment for the project
```

```
--external-only       sets only the environment for the externals (the
                      project is used only to select the version of LCG)
--dev                 prepend $LHCBDEV to the search path. Note: the
                      directories are searched in the order specified on the
                      command line.
--dev-dir DEVDIR      prepend DEVDIR to the search path Note: the
                      directories are searched in the order specified on the
                      command line.
-v VERSION            must be used after the name of an external to specify
                      a non default version for it
--set-CMTPATH         Set CMTPATH to the value used internally by CMT
                      (DANGEROUS)
--runtime-project PROJECT [VERSION]
                      Add a project to the runtime environment
--overriding-project PROJECT [VERSION]
                      Add a project to override packages
--no-auto-override    Do not automatically prepend the projects
                      [('ExtraPackages', [])]
--use-grid            Enable auto selection of LHCbGrid project
-q, --silent          Removes message printout during setup
--keep-CMTPROJECTPATH
                      Do not override the value of the environment variable
                      CMTPROJECTPATH
--nightly SLOT [DAY]  Add the required slot of the LHCb nightly builds to
                      the list of DEV dirs. DAY must be a 3 digit
                      abbreviation of the weekday, by default the current
                      day. Special settings of the CMTPROJECTPATH needed for
                      the nightly build slot are taken into account.
```

How SetupProject works

-- MarcoClemencic - 20 Oct 2008

This topic: LHCb > SetupProject
Topic revision: r9 - 2009-11-25 - HubertDegaudenzi