
LHCbDIRAC Documentation

Release v7

LHCbDIRAC Project.

Jul 03, 2018

Contents

1	Developers Guide	3
1.1	Guide for developing LHCbDIRAC (and DIRAC, for LHCb developers)	3
1.2	Developing DIRAC and LHCbDIRAC	5
1.3	HOW TOs	6
1.4	Browsing the code running in production	6
1.5	I developed something, I want it in the next release	6
1.6	Asking for a LHCbDIRAC patch	6
2	Administrator Guide	7
2.1	LHCbDIRAC Releases	7
2.2	Renewal of certificate for ONLINE machine	14
2.3	ONLINE steps	14
3	Certification	17
3.1	LHCbDIRAC Certification (development) Releases	17
3.2	The certification process	21
3.3	Acceptance test steps	24
4	Indices and tables	41



The **DIRAC** project is a complete Grid solution for a community of users such as the LHCb Collaboration. DIRAC forms a layer between a particular community and various compute resources to allow optimized, transparent and reliable usage:

- DIRAC documentation: <http://dirac.readthedocs.io/en/latest/index.html>
- DIRAC hosted repository: <https://github.com/DIRACGrid>

LHCbDIRAC is the LHCb extension to DIRAC:

- LHCbDIRAC documentation: <http://lhcb-dirac.readthedocs.io/en/latest/index.html>
- LHCbDIRAC hosted repository: <https://gitlab.cern.ch/lhcb-dirac>

1.1 Guide for developing LHCbDIRAC (and DIRAC, for LHCb developers)

A short, but hopefully comprehensive guide on developing in LHCbDIRAC, referencing DIRAC development model. For what are DIRAC and LHCbDIRAC doing, look elsewhere.

LHCbDIRAC is a DIRAC extension. This means that LHCbDIRAC cannot leave independently from DIRAC. There are a number of DIRAC extensions, maintained by various communities worldwide, and LHCbDIRAC is the most important out there, and the one that receives the most support by DIRAC itself. But it also means that DIRAC and LHCbDIRAC (as all the other DIRAC extensions) have different [release cycles](#) and [versioning](#), adopts different [version control systems](#), use different [tracking systems](#), and that the [code conventions](#) may slightly differ.

DIRAC can also have other extensions, independent from a VO. All these are hosted at [github](#).

1.1.1 Pre-requisites

Within this section we just look at what is necessary to know before looking at the code.

Releases

Naming

Both DIRAC and LHCbDIRAC follow the same naming conventions for releases, inherited by the LHCb convention:

vMrNpt

where:

- **M** stands for *major version*, or simply *version*
- **N** stands for *minor version*, or simply *release*

- **t** stands for *patch version*, or simply *patch*

with a special *pre-release* naming convention: **-preX**.

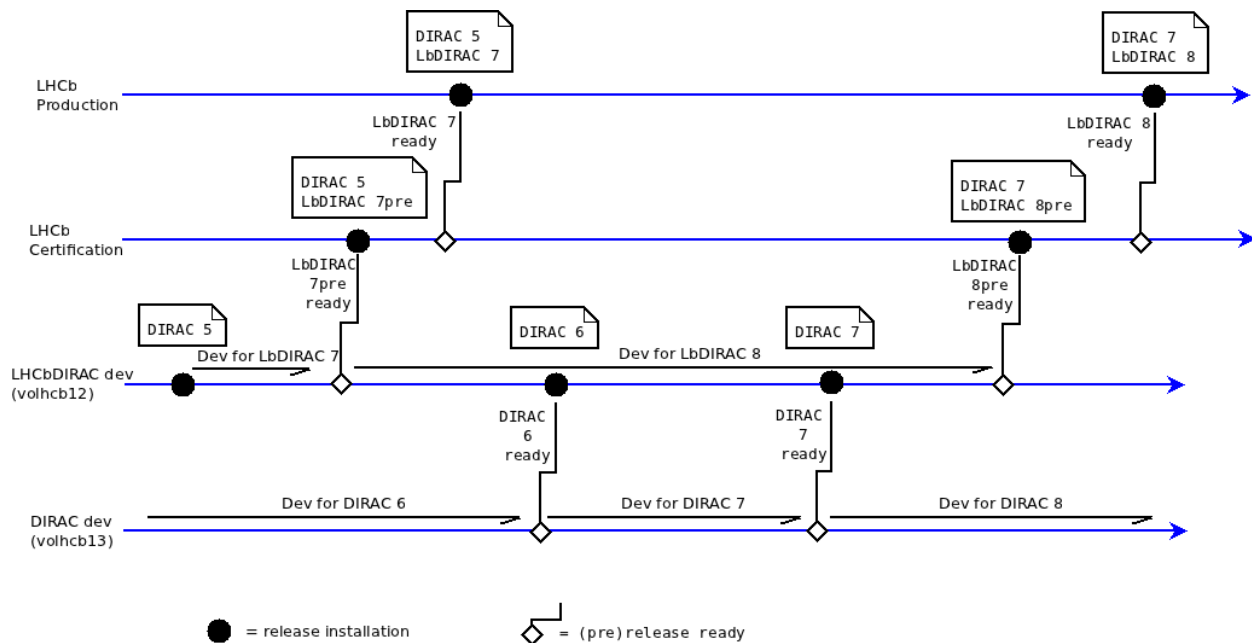
This will be clear with some examples:

- **v6r2p0** is the version 6, release 2, patch 0
- **v7r5p13** is the version 7, release 5, patch 13
- **v8r1-pre2** is the second pre-release of version 8, release 1

There are no pre-releases for patches.

Release cycle

When developing LHCbDIRAC, we need to consider that every LHCbDIRAC is developed on top of a DIRAC release. The following picture explains the model.



So, for example, there might be 2 or more LHCbDIRAC releases based on top of the same DIRAC release. Every LHCbDIRAC developer has to know which release of DIRAC its development is for. The major version of both DIRAC and LHCbDIRAC changes rarely, let's say every 2 years. The minor version changes more frequently in LHCbDIRAC with respect to DIRAC, but there is no strict advancement scheduling for none of the 2.

A pre-release is a release candidate that goes through a certification process.

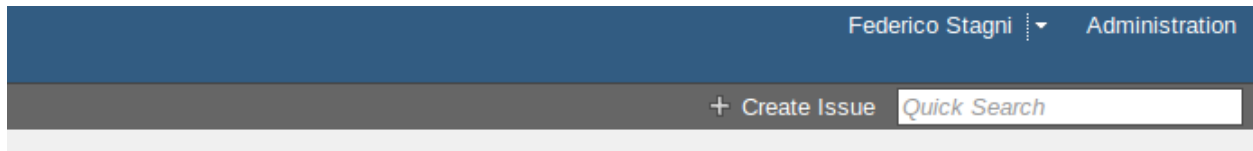
Version Control

LHCbDIRAC version control is based on GIT. GIT is a very popular distributed revision control system. The reader is supposed to be familiar with the way such systems work. The code is hosted in the [CERN GitLab](#).

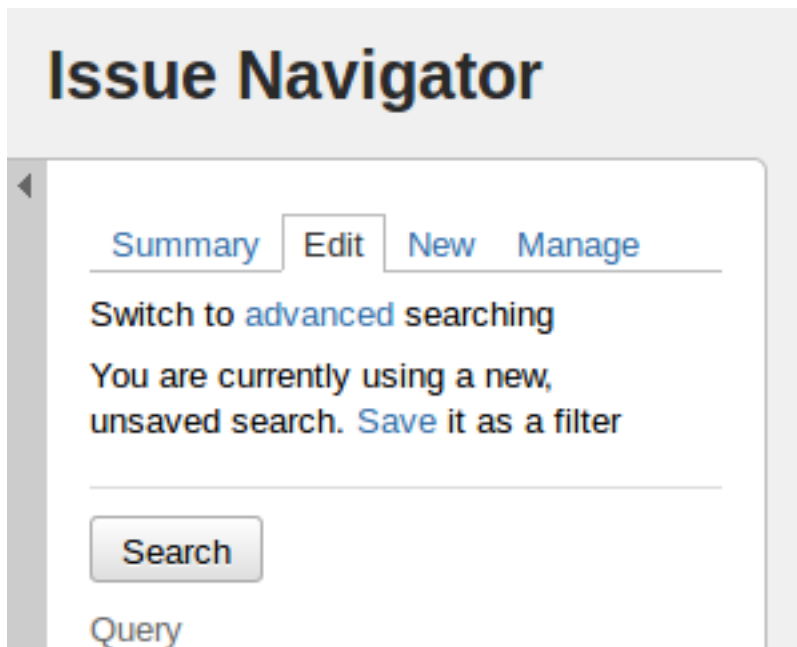
Tracking systems

The tracking system used for LHCbDIRAC is [jira](#). Jira is a fundamental tool for LHCbDIRAC, and its use is mandatory. Every development should be tracked there. Jira is a very powerful tool, but requires some time to master. Few notes/links:

- The *official documentation* is [here](#). You might also be interested in watching the first ~15 minutes of this [video](#).
- Issuing a new bug/task/story/etc. (there are many possible choices) is easy, just look at the top right of the screen:



- Remember to put a “component” when you make a new issue
- When you make a new research in the issue navigator, you can save the search: it will become useful later



Developer tools

You are free to choose the editor or IDE you prefer. I know [Emacs](#) is a great tool, and someone can’t just leave without it. And that also [vim](#) is great. [Eclipse](#) with [pydev](#) is another good choice. Other possibilities include [PyCharm](#) ([IntelliJIDEA](#)) and [atom](#)

1.2 Developing DIRAC and LHCbDIRAC

Developing the code is not just about editing. You also want to “run” something, usually for testing purposes. The DIRAC way of developing can be found [here](#) and it applies also to LHCbDIRAC. Please follow carefully especially what’s [here](#)

In general, if you are developing LHCbDIRAC, you should consider that:

- everything that applies to DIRAC development, also applies to LHCbDIRAC development, so, follow carefully the links above
- every LHCbDIRAC release has a strong dependency with a DIRAC release. See <https://gitlab.cern.ch/lhcb-dirac/LHCbDIRAC/blob/master/CONTRIBUTING.md> for more info.

1.3 HOW TOs

1.4 Browsing the code running in production

If you want to browse the DIRAC (and LHCbDIRAC) code running in production you'll first of all have to know which version is installed. Announcements of new deployments are done via the LHCb operations [eLog](#). The code is also always installed in the afs release area (`$LHCb_release_area/DIRAC/DIRAC_vX5rYpZ/DIRAC`) but you can normally use git to switch from one to another.

1.5 I developed something, I want it in the next release

Just open a merge request to the devel branch of LHCbDirac: all the releases (minor and major) are created branching from this branch.

1.6 Asking for a LHCbDIRAC patch

Just open a merge request to the master branch of LHCbDirac. If in a hurry, drop an e-mail to the [lhcb-dirac](#) mailing list.

This page is the work in progress. See more material here soon !

2.1 LHCbDIRAC Releases

The following procedure applies fully to LHCbDIRAC production releases, like patches. For pre-releases (AKA certification releases, there are some minor changes to consider).

2.1.1 Prerequisites

The release manager needs to:

- be aware of the LHCbDIRAC repository structure and branching as highlighted in the [contribution guide](#).
- have forked LHCbDIRAC on GitLab as a “personal project” (called “origin” from now on)
- have cloned origin locally
- have added <https://gitlab.cern.ch/lhcb-dirac/LHCbDIRAC> as “upstream” repository to the local clone
- have push access to the master branch of “upstream” (being part of the project “owners”)
- have DIRAC installed
- have been granted write access to <webService>
- have “lhcb_admin” or “diracAdmin” role.
- have a Proxy

The release manager of LHCbDIRAC has the triple role of:

1. creating the release
2. making basic verifications
3. deploying it in production

2.1.2 1. Creating the release

Unless otherwise specified, (patch) releases of LHCbDIRAC are usually done “on top” of the latest production release of DIRAC. The following of this guide assumes the above is true.

Creating a release of LHCbDIRAC means creating a tarball that contains the release code. This is done in 3 steps:

1. Merging “Merge Requests”
2. Propagating to the devel branch
3. Creating the release tarball, add uploading it to the LHCb web service

But before:

Pre

Verify what is the last tag of DIRAC:

```
# it should be in this list:
git describe --tags $(git rev-list --tags --max-count=10)
```

A tarball containing it should be already uploaded [here](#)

You may also look inside the .cfg file for the DIRAC release you’re looking for: it will contain an “Externals” version number, that should also be a tarball uploaded in the same location as above.

If all the above is ok, we can start creating the LHCbDIRAC release.

Merging “Merge Requests”

[Merge Requests \(MR\)](#) that are targeted to the master branch and that have been approved by a reviewer are ready to be merged

If there are no MRs, or none ready: please skip to the “update the CHANGELOG” subsection.

Otherwise, simply click the “Accept merge request” button for each of them.

Then, from the LHCbDIRAC local fork you need to update some files:

```
# if you start from scratch otherwise skip the first 2 commands
mkdir $(date +%Y%m%d) && cd $(date +%Y%m%d)
git clone https://:gitlab.cern.ch:8443/lhcb-dirac/LHCbDIRAC.git
git remote add upstream https://:gitlab.cern.ch:8443/lhcb-dirac/LHCbDIRAC.git
# update your "local" upstream/master branch
git fetch upstream
# create a "newMaster" branch which from the upstream/master branch
git checkout -b newMaster upstream/master
# determine the tag you're going to create by checking what was the last one from the
# following list (add 1 to the "p"):
git describe --tags $(git rev-list --tags --max-count=5)
# Update the version in the __init__ file:
vim LHCbDIRAC/__init__.py
# Update the version in the releases.cfg file:
vim LHCbDIRAC/releases.cfg
# For updating the CHANGELOG, get what's changed since the last tag
t=$(git describe --abbrev=0 --tags); git --no-pager log ${t}..HEAD --no-merges --
# pretty=format:'* %s';
# copy the output, add it to the CHANGELOG (please also add the DIRAC version)
```

(continues on next page)

(continued from previous page)

```
vim CHANGELOG # please, remove comments like "fix" or "pylint" or "typo"...
#If needed, change the versions of the packages
vim dist-tools/projectConfig.json
git add -A && git commit -av -m "<YourNewTag>"
```

Time to tag and push:

```
# make the tag
git tag -a <YourNewTag> -m <YourNewTag>
# push "newMaster" to upstream/master
git push --tags upstream newMaster:master
# delete your local newMaster
# before change your branch use git checkout "existing branch name"
git branch -d newMaster
```

Remember: you can use “git status” at any point in time to make sure what’s the current status.

Propagate to the devel branch

Now, you need to make sure that what’s merged in master is propagated to the devel branch. From the local fork:

```
# get the updates (this never hurts!)
git fetch upstream
# create a "newDevel" branch which from the upstream/devel branch
git checkout -b newDevel upstream/devel
# merge in newDevel the content of upstream/master
git merge upstream/master
```

The last operation may result in potential conflicts. If happens, you’ll need to manually update the conflicting files (see e.g. [this guide](#)). As a general rule, prefer the master fixes to the “HEAD” (devel) fixes. Remember to add and commit once fixed. Note: For porting the LHCbDIRAC.init.py from master to devel, we prefer the HEAD version (only for this file!!!)

Please fix the conflict if some files are conflicting. Do not forget to execute the following:

```
git add -A && git commit -m " message"
```

Conflicts or not, you’ll need to push back to upstream:

```
# push "newDevel" to upstream/devel
git push upstream newDevel:devel
# delete your local newDevel
git branch -d newDevel
# keep your repo up-to-date
git fetch upstream
```

Creating the release tarball, add uploading it to the LHCb web service

Login on lxplus, run

```
lb-run LHCbDirac/prod bash -norc

git archive --remote ssh://git@gitlab.cern.ch:7999/lhcb-dirac/LHCbDIRAC.git devel_
↪LHCbDIRAC/releases.cfg | tar -x -v -f - --transform 's|^LHCbDIRAC/||' LHCbDIRAC/
↪releases.cfg
```

(continues on next page)

(continued from previous page)

```
dirac-distribution -r v8r3p1 -l LHCb -C file:///`pwd`/releases.cfg (this may take_
↳some time)
```

Don't forget to read the last line of the previous command to copy the generated files at the right place. The format is something like:

```
( cd /tmp/joel/tmpxg8UuvDiracDist ; tar -cf - *.tar.gz *.md5 *.cfg ) | ssh_
↳lhcbprod@lxplus.cern.ch 'cd /afs/cern.ch/lhcb/distribution/DIRAC3/tars && tar -xvf_
↳- && ls *.tar.gz > tars.list'
```

And just copy/paste/execute it.

If you do not have access to lhcbprod, you can use your user name.

2.1.3 2. Making basic verifications

Once the tarball is done and uploaded, the release manager is asked to make basic verifications, via Jenkins, if the release has been correctly created.

At this [link](#) you'll find some Jenkins Jobs ready to be started. Please start the following Jenkins jobs and come back in about an hour to see the results for all of them.

1. https://lhcb-jenkins.cern.ch/jenkins/view/LHCbDIRAC/job/!RELEASE!__pylint_unit/ the !RELEASE! is the actual release for example: https://lhcb-jenkins.cern.ch/jenkins/view/LHCbDIRAC/job/v8r5__pylint_unit/

This job will: run pylint (errors only), run all the unit tests found in the system, assess the coverage. The job should be considered successful if:

- the pylint error report didn't increase from the previous job run
- the test results didn't get worse from the previous job run
- the coverage didn't drop from the previous job run

2. https://lhcb-jenkins.cern.ch/jenkins/view/LHCbDIRAC/job/!RELEASE!__pilot/

This job will simply install the pilot. Please just check if the result does not show in an “unstable” status

3. https://lhcb-jenkins.cern.ch/jenkins/view/LHCbDIRAC/job/!RELEASE!__/

TODO

2.1.4 3. Advertise the new release

Before you start the release you must write an Elog entry 1 hour before you start the deployment. You have to select Production and Release tick boxes. When the intervention is over you must notify the users (reply to the Elog message).

2.1.5 4. Deploying the release

Deploying a release means deploying it for the various installations:

```
* client
* server
* pilot
```

release for client

Please refer to this [TWIKI page](#) a quick test to validate the installation is to run the SHELL script \$LHCbRELEASE/LHCbDIRAC/LHCbDIRAC_vXrY/LHCbDiracSys/test/client_test.csh

go to this [web page](#) for asking to install the client release in AFS and CVMFS:

- in the field “Project list” put : “Dirac vNrMpK LHCbDirac vArBpC” (NOTE: If LHCbGRID is already released, please use only DIRAC and LHCbDIRAC: DIRAC vNrMpK LHCbDirac vArBpC)
- in the field “platforms” put : “x86_64-slc6-gcc48-opt x86_64-slc6-gcc49-opt”

Then click on the “BUILD” button

- within 10-15 min the build should start to appear in the nightlies page <https://lhcb-nightlies.cern.ch/release/>
- if there is a problem in the build, it can be re-started via the dedicated button (it will not restart by itself after a retag)

If it is the production release, and only in this case, once satisfied by the build, take note of the build id (you can use the direct link icon) and make the request via <https://sft.its.cern.ch/jira/browse/LHCBDEP>.

- **NOTE: If some package is already released, please do not indicate in the Jira task. For example: a Jira task when:**
 - DIRAC is not released, then the message in the JIRA task: Summary:Dirac v6r14p37 and LHCbDirac v8r2p50; Description: Please release Dirac and LHCbDirac in this order based on build 1526
 - DIRAC is released, then the message in the JIRA task: Summary:LHCbDirac v8r2p50; Description: Please release LHCbDirac based on build 1526

Server

To install it on the VOBOXes from lxplus:

```
lhcb-proxy-init -g diracAdmin
dirac-admin-sysadmin-cli --host volhcbXX.cern.ch
>update LHCbDIRAC-v8r3p32
>restart *
```

The (better) alternative is using the web portal or using the following script: LHCbDIRAC/LHCbDiracPolicy/scripts/create_vobox_update.

The recommended way is the following:

```
ssh lxplus
mkdir DiracInstall; cd DiracInstall
cp LHCbDIRAC/LHCbDiracPolicy/scripts/create_vobox_update .
cp LHCbDIRAC/LHCbDiracPolicy/scripts/skel_vobox_update .
python create_vobox_update v8r2p30
```

This command will create 6 files called “vobox_update_MyLetter” then you can run in 6 windows the recipe for one single machine like that:

```
ssh lxplus
cd DiracInstall ; lb-run LHCbDIRAC/prod bash -norc ; lhcb-proxy-init -g lhcb_admin;
↪dirac-admin-sysadmin-cli
and from the prompt ::
[host] : execfile vbox_update_MyLetter
[host] : quit
```

Note:

It is normal if you see the following errors:

→ Executing restart Framework SystemAdministrator [ERROR] Exception while reading from peer: (-1, 'Unexpected EOF')

In case of failure you have to update the machine by hand. Example of a typical failure:

```
--> Executing update v8r2p42
Software update can take a while, please wait ...
[ERROR] Failed to update the software
Timeout (240 seconds) for '['dirac-install', '-r', 'v8r2p42', '-t', 'server', '-e',
↪ 'LHCb', '-e', 'LHCb', '/opt/dirac/etc/dirac.cfg']' call
```

Login to the failing machine, become dirac, execute manually the update, and restart everything. For example:

```
ssh lbvobox11
sudo su - dirac
dirac-install -r v8r2p42 -t server -e LHCb -e LHCb /opt/dirac/etc/dirac.cfg
lhcb-restart-agent-service
runsvctrl t startup/Framework_SystemAdministrator/
```

Specify that this error can be ignored (but should be fixed !):

```
2016-05-17 12:00:00 UTC dirac-install [ERROR] Requirements installation script /opt/
↪ dirac/versions/v8r2p42_1463486162/scripts/dirac-externals-requirements failed.
↪ Check /opt/dirac/versions/v8r2p42_1463486162/scripts/dirac-externals-requirements.
↪ err
```

WebPortal

When the web portal machine is updated then you have to compile the WebApp:

```
ssh lhcb-portal-dirac.cern.ch
sudo su - dirac
dirac-install -r VERSIONTOBEINSTALLED -t server -l LHCb -e LHCb,LHCbWeb,WebAppDIRAC /
↪ opt/dirac/etc/dirac.cfg (for example: dirac-install -r v8r4p2 -t server -l LHCb -e
↪ LHCb,LHCbWeb,WebAppDIRAC /opt/dirac/etc/dirac.cfg)
dirac-webapp-compile
```

When the compilation is finished:

```
lhcb-restart-agent-service
runsvctrl t startup/Framework_SystemAdministrator/
```

TODO

When the machines are updated, then you have to go through all the components and check the errors. There are two possibilities:

1. Use the Web portal (SystemAdministrator)
2. Command line:

```
for h in $(grep 'set host' vbox_update_* | awk {'print $NF'}); do echo "show errors" | dirac-admin-
sysadmin-cli -H $h; done | less
```


Pilot

Use the following script (from, e.g., lxplus after having run *lb-run LHCbDIRAC tcsh*):

```
dirac-pilot-version -S v8r2p42
```

for checking and updating the pilot version. Note that you'll need a proxy that can write in the CS (i.e. lhcb-admin). This script will make sure that the pilot version is update BOTH in the CS and in the json file used by pilots started in the vacuum.

Basic instruction how to merging the devel branch into master (NOT for PATCH release)

Our developer model is to keep only two branches: master and devel. When we made a major release we have to merge devel to master. Before the merging please create a new branch based on master using the web interface of GitLab. This is for safety. After you can merge devel to master:

```
mkdir $(date +%Y%m%d) && cd $(date +%Y%m%d)
git clone ssh://git@gitlab.cern.ch:7999/lhcb-dirac/LHCbDIRAC.git
cd LHCbDIRAC
git remote rename origin upstream
git fetch upstream
git checkout -b newMaster upstream/master
git merge upstream/devel
git push upstream newMaster:master
```

2.1.6 5. Mesos cluster

Mesos is currently only used for the certification. In order to push a new version on the Mesos cluster, 3 steps are needed:

- Build the new image
- Push it the lhcbdirac gitlab repository
- Update the version of the running containers

All these functionalities have been wrapped up in a script (dirac-docker-mgmt), available on all the lbmesosadm* machines (01, 02)

The next steps are the following:

```
# build the new image
# this will download the necessary files, and build
# the image locally
dirac-docker-mgmt.py -v v8r5 --build

# Push it to the remote lhcbdirac registry
# Your credentials for gitlab will be asked
dirac-docker-mgmt.py -v v8r5 --release

# Update the version of the running containers
# The services and number of instances running
# will be preserved
dirac-docker-mgmt.py -v v8r5 --deploy
```

2.2 Renewal of certificate for ONLINE machine

Login as `lhcbprod` on `lbdirac.cern.ch` and generate the certificate request

```
openssl req -new -subj /CN=lbdirac.cern.ch -out newcsr.csr -nodes -sha1
```

Open in your browser the page <http://ca.cern.ch> cut the content of `newcsr.csr` (created in the previous step) in the web page and click on the submit button. Save the Base 64 encoded certificate as a file `newcert.cer`. Copy this file to `lbdirac.cern.ch`. Then convert the certificate in the correct format.

```
openssl pkcs12 -export -inkey privkey.pem -in newcert.cer -out myCertificate.pks (You
→ will have to type the PEM password you typed in the previous step. Type also an
→ export password, and don't forget it. Your certificate in PKCS12 format is ready in
→ file myCertificate.pks, you can delete the other files.)
openssl pkcs12 -in myCertificate.pks -clcerts -nokeys -out hostcert.pem
openssl pkcs12 -in myCertificate.pks -nocerts -out hostkey.pem.passwd
openssl rsa -in hostkey.pem.passwd -out hostkey.pem (remove the password)
```

If you want to test that the new host certificate is valid without any password, just do

```
dirac-proxy-init -C <cert> -K <key>
```

2.3 ONLINE steps

2.3.1 Installation of LHCbDirac

The machine running the transfers from the pit is `lbdirac`, and is in the online network. This machine runs:

- A complete RMS: ReqManager (url: RequestManagement/onlineGateway), a ReqProxy (known only from inside) and a RequestExecutingAgent
- The RAWIntegrity system: the RAWIntegrityHandler and RAWIntegrityAgent

A special catalog is defined in the local configuration in order to keep track of the files transferred:

```
RAWIntegrity
{
    AccessType = Read-Write
    Status = Active
}
```

We also have two special configuration for StorageElements:

```
# Setting it to NULL to transfer without
# checking the checksum, since it is already done by
# the DataMover and the RAWIntegrityAgent
# It should avoid the double read on the local disk
ChecksumType=NULL
# Setting this to True is dangerous...
# If we have a SRM_FILE_BUSY, we remove the file
# But we have enough safety net for the transfers from the pit
SRMBusyFilesExist = True
```

Finally, you need to overwrite the URLs of the RMS to make sure that they use the internal RMS:

```
URLs
{
  ReqManager = dips://lbdirac.cern.ch:9140/RequestManagement/ReqManager
  ReqProxyURLs = dips://lbdirac.cern.ch:9161/RequestManagement/ReqProxy
}
```

2.3.2 Workflow

The DataMover is the Online code responsible for the interaction with the BKK (register the run, the files, set the replica flag), to request the physical transfers, and to remove the file of the Online storage when properly transferred.

The doc is visible here: https://lbdokuwiki.cern.ch/online_user:rundb_onlinetoofflinedataflow

The DataMover registers the Run and the files it already knows about in the BKK. Then it creates for each file a request with a PutAndRegister operation. The target SE is CERN-RAW, the Catalog is RAWIntegrity. The RequestExecutingAgent will execute the copy from the local online storage to CERN-RAW, and register it in the RAWIntegrity DB.

The RAWIntegrityAgent looks at all the files in the DB that are in status 'Active'.

For each of them, it will check if the file is already on tape, and if so, compare the checksum.

If the checksum is correct, the file is registered in the DFC only, a removal request is sent to the local ReqManager, and the status set to 'Done'. This removal Request sends a signal to the DataMover, which will mark the file for removal (garbage collection), and the replica flag to yes in the BKK

If the checksum is not correct, the file is removed from CERN-RAW, the status set to 'Failed' and a retransfer request is put to the ReqManager (this last part will change soon because does not make sense).

In case of any error in the process (cannot get metadata, cannot send request, etc), no change is done to the RAWIntegrityDB concerning that file, and it will be part of the next cycle.

3.1 LHCbDIRAC Certification (development) Releases

The following procedure applies to pre-releases (AKA certification releases) and it is a simpler version of what applies to production releases.

This page details the duty of the release manager. The certification manager duties are detailed in the next page.

3.1.1 What for

The *release manager* of LHCbDIRAC has the role of:

1. creating the pre-release
2. making basic tests
3. deploying it in the certification setup

The *certification manager* would then follow-up on this by: 4. making even more tests

And, after several iterations of the above, before: 5. merging in the production branch

Points 4 and 5 won't anyway be part of this first document.

3.1.2 1. Creating the release

Unless otherwise specified, certification releases of LHCbDIRAC are done “on top” of the latest pre-release of DIRAC. The following of this guide assumes the above is true.

Creating a pre-release of LHCbDIRAC means creating a tarball that contains the code to certify. This is done in 2 steps:

1. Merging “Merge Requests”
2. Creating the release tarball, add uploading it to the LHCb web service

But before:

Pre

If you use a version of git prior to 1.8, remove the option `--pretty` in the command line

Verify what is the last tag of DIRAC:

```
# it should be in this list:
git describe --tags $(git rev-list --tags --max-count=10)
```

A tarball containing it should be already uploaded [here](#)

You may also look inside the `.cfg` file for the DIRAC release you're looking for: it will contain an "Externals" version number, that should also be a tarball uploaded in the same location as above.

If all the above is ok, we can start creating the LHCbDIRAC pre-release.

Merging "Merge Requests"

[Merge Requests \(MR\)](#) that are targeted to the `devel` branch and that have been approved by a reviewer are ready to be merged

If there are no MRs, or none ready: please skip to the "update the CHANGELOG" subsection.

Otherwise, simply click the "Accept merge request" button for each of them.

Then, from the LHCbDIRAC local fork you need to update some files:

```
# if you start from scratch otherwise skip the first 2 commands
mkdir $(date +%Y%m%d) && cd $(date +%Y%m%d)
git clone https://:gitlab.cern.ch:8443/lhcb-dirac/LHCbDIRAC.git
git remote add upstream https://:gitlab.cern.ch:8443/lhcb-dirac/LHCbDIRAC.git
# update your "local" upstream/master branch
git fetch upstream
# create a "newDevel" branch which from the upstream/devel branch
git checkout -b newDevel upstream/devel
# determine the tag you're going to create by checking what was the last one from the
# following list (add 1 to the "p"):
git describe --tags $(git rev-list --tags --max-count=5)
# Update the version in the __init__ file:
vim LHCbDIRAC/__init__.py
# Update the version in the releases.cfg file:
vim LHCbDIRAC/releases.cfg
# For updating the CHANGELOG, get what's changed since the last tag
# please use the proper LHCbDIRAC tag; replace v8r2p46
git log --pretty=oneline ${t}..HEAD | grep -Ev "($(git log --pretty=oneline ${t}..
# v8r2p46 | awk {'print $1'} | tr '\n' '|')BOOM)"
# copy the output, add it to the CHANGELOG (please also add the DIRAC version)
vim CHANGELOG # please, remove comments like "fix" or "pylint" or "typo"...
# If needed, change the versions of the packages
vim dist-tools/projectConfig.json
# Commit in your local newDevel branch the 3 files you modified
git add -A && git commit -av -m "<YourNewTag>"
```

Time to tag and push:

```
# make the tag
git tag -a <YourNewTag> -m <YourNewTag>
# push "newDevel" to upstream/devel
git push --tags upstream newDevel:devel
# delete your local newDevel
git branch -d newDevel
```

Remember: you can use “git status” at any point in time to make sure what’s the current status.

Creating the release tarball, add uploading it to the LHCb web service

Login on lxplus, run

```
lb-run LHCbDirac/prod bash -norc
git archive --remote ssh://git@gitlab.cern.ch:7999/lhcb-dirac/LHCbDIRAC.git devel_
↪LHCbDIRAC/releases.cfg | tar -x -v -f - --transform 's|^LHCbDIRAC/||' LHCbDIRAC/
↪releases.cfg
dirac-distribution -r v8r4-pre1 -l LHCb -C file:///`pwd`/releases.cfg (this may take_
↪some time)
```

Don’t forget to read the last line of the previous command to copy the generated files at the right place. The format is something like:

```
( cd /tmp/joel/tmpxg8UuvDiracDist ; tar -cf - *.tar.gz *.md5 *.cfg ) | ssh
↪$USER@lxplus.cern.ch 'cd /afs/cern.ch/lhcb/distribution/DIRAC3/tars && tar -xvf - &
↪& ls *.tar.gz > tars.list'
```

And just copy/paste/execute it.

3.1.3 2. Making basic verifications

Once the tarball is done and uploaded, the release manager is asked to make basic verifications, via Jenkins, if the release has been correctly created.

At this [link](#) you’ll find some Jenkins Jobs ready to be started. Please start the following Jenkins jobs and come back in about an hour to see the results for all of them.

1. https://lhcb-jenkins.cern.ch/jenkins/view/LHCbDIRAC/job/PRERELEASE__pylint_unit/

This job will: run pylint (errors only), run all the unit tests found in the system, assess the coverage. The job should be considered successful if:

- the pylint error report didn’t increase from the previous job run
- the test results didn’t get worse from the previous job run
- the coverage didn’t drop from the previous job run

2. https://lhcb-jenkins.cern.ch/jenkins/view/LHCbDIRAC/job/PRERELEASE__pilot/

This job will simply install the pilot. Please just check if the result does not show in an “unstable” status

3. https://lhcb-jenkins.cern.ch/jenkins/view/LHCbDIRAC/job/RELEASE__/

TODO

3.1.4 3. Deploying the release

Deploying a release means deploying it for some installation:

```
* client
* server
* pilot
```

release for client

Please refer to this [TWIKI page](#) a quick test to validate the installation is to run the SHELL script \$LHCbRELEASE/LHCbDIRAC/LHCbDIRAC_vXrY/LHCbDiracSys/test/client_test.csh

go to <https://jenkins-lhcb-nightlies.web.cern.ch/job/nightly-builds/job/release/build> page for asking to install the client release in AFS and CVMFS:

- in the field “Project list” put : “Dirac vNrMpK LHCbGrid vArB LHCbDirac vArBpC ” (LHCbGrid version can be found: <https://gitlab.cern.ch/lhcb-dirac/LHCbDIRAC/blob/devel/dist-tools/projectConfig.json>)
- in the field “platforms” put : “x86_64-slc6-gcc49-opt x86_64-slc6-gcc62-opt x86_64-centos7-gcc62-opt”
- in the field “build_tool” put : “CMake”
- in the field “scripts_version” put : “prepare-for-new-jenkins”

Then click on the “BUILD” button

- within 10-15 min the build should start to appear in the nightlies page <https://lhcb-nightlies.cern.ch/release/>
- if there is a problem in the build, it can be re-started via the dedicated button (it will not restart by itself after a retag)

When the release is finished <https://lhcb-nightlies.cern.ch/release/>, you can deploy to the client.

Note: Please execute the following commands sequentially.

The following commands used to prepare the RPMs:

```
ssh lhcb-archive
export build_id=1520
lb-release-rpm /data/artifacts/release/lhcb-release/$build_id
lb-release-rpm --copy /data/artifacts/release/lhcb-release/$build_id
```

If the rpm are created, you can deploy the release (Do not execute parallel the following commands):

```
ssh lxplus
cd /afs/cern.ch/lhcb/software/lhcb_rpm_dev
export MYITEROOT=/afs/cern.ch/lhcb/software/lhcb_rpm_dev
export MyProject=Dirac
export MyVersion=vArBpC
./lbpr rpm -- -ivh --nodeps /afs/cern.ch/lhcb/distribution/rpm/lhcb/${MyProject^^}_$
↪{MyVersion}*
export MyProject=LHCbDirac
export MyVersion=vArB-preC
./lbpr rpm -- -ivh --nodeps /afs/cern.ch/lhcb/distribution/rpm/lhcb/${MyProject^^}_$
↪{MyVersion}*

```


Server

To install it on the VOBOXes (certification only) from lxplus:

```
lhcb-proxy-init -g diracAdmin
dirac-admin-sysadmin-cli --host volhcbXX.cern.ch
>update LHCbDIRAC-v8r4-pre1
>restart *
```

The (better) alternative is using the web portal.

Pilot

Use the following script (from, e.g., lxplus after having run *lb-run -dev LHCbDIRAC bash*):

```
dirac-pilot-version
```

for checking and updating the pilot version. Note that you'll need a proxy that can write in the CS (i.e. lhcb-admin). This script will make sure that the pilot version is update BOTH in the CS and in the json file used by pilots started in the vacuum. The command to update is

```
dirac-pilot-version -S v8r4-pre1
```

Make sure that you are in the certification setup (e.g. check the content of your .dirac.cfg file)

3.2 The certification process

Certifying a release is a process. There are a number of steps to make to reach the point in which we can finally say that a release is at production level. Within LHCbDirac, we are trying to streamline and automatize this process as much as possible. Even with that, some tests still require manual intervention. We can split the process in a series of incremental tests.

Within the following sections we describe, step by step, all the actions needed.

The whole certification process varies from release to release. The list of things to do is maintained in trello boards.

3.2.1 Unit test

When a new release candidate is created from the devel branch, we first run pylint on the whole codebase, and all the unit tests. Jenkins automatizes this for us.

3.2.2 Integration and Regression tests

Run by Jenkins.

3.2.3 System tests

Even if it should not be considered strictly as a test, running all the agents and service within certification is an action to take. Agents and services spits errors and exceptions. While the second are obviously bugs, the first are not to be considered bugs until an expert look. Nonetheless, we have created a tool to easily identify all new exceptions and errors:

```
codeLocation=https://gitlab.cern.ch/lhcb-dirac/LHCbDIRAC/raw/devel/tests/System/
↪LogsParser/
mkdir /tmp/logTest
cd /tmp/logTest
wget -r -np -nH --cut-dirs=7 $codeLocation
/bin/bash logParser.sh
```

For testing that the RMS works, there is an ad-hoc test:

```
wget http://github.com/DIRACGrid/DIRAC/blob/integration/DataManagementSystem/test/
↪IntegrationFCT.py
python IntegrationFCT.py lhcb_user CERN-USER RAL-USER CNAF-USER
python IntegrationFCT.py lhcb_prod CERN-FAILOVER RAL-FAILOVER CNAF-FALIOVER
```

Those commands will create and put to the Request Management System two new requests:

1. for lhcb_user group, which should be banned from using the FTS system
2. for lhcb_prod or lhcb_prmgr group, which this should be executed using FTS

You could monitor their execution using *Request monitor* web page or by using CLI comamnd:

```
dirac-rms-show-request test<userName>-<userGroup>
```

The execution itself will take a while, but at the end both requests statuses should be set to ‘Done’.

Another test, again for the RMS, combined with FTS, is to simply use the following standard DIRAC scripts:

```
dirac-dms-create-replication-request CNAF_MC-DST /lhcb/certification/test/ALLSTREAMS.
↪DST/00000751/0000/00000751_00000014_1.allstreams.dst
```

Which will actually schedule the replication of such file using FTS. This will print an ID that can be used for the script

```
dirac-rms-show-request ID
```

That should show how the request goes (quickly) in status “Scheduled”, and then “Done”.

The following script, instead, will remove the copy just created.

```
dirac-dms-create-removal-request CNAF_MC-DST /lhcb/certification/test/ALLSTREAMS.DST/
↪00000751/0000/00000751_00000014_1.allstreams.dst
```

Again, monitoring is available as above.

For testing the replications and removals, use the following:

```
dirac-dms-add-replication --BKQuery=/validation/MC11a/Beam3500GeV-2011-MagDown-Nu2-
↪EmNoCuts/Sim05/Trig0x40760037Flagged/Reco12a/Stripping17Flagged/12463412/ALLSTREAMS.
↪DST --Plugin=ReplicateDataset --Test
```

That will just print out how many files can be replicated. If there is at least one file (for this particular query there should be 35), then you can start it with:

```
dirac-dms-add-replication --BKQuery=/validation/MC11a/Beam3500GeV-2011-MagDown-Nu2-
↪EmNoCuts/Sim05/Trig0x40760037Flagged/Reco12a/Stripping17Flagged/12463412/ALLSTREAMS.
↪DST --Plugin=ReplicateDataset --NumberOfReplicas=2 --SecondarySEs Tier1-DST --Start
```

You can monitor the advancement using:

```
dirac-dms-replica-stats --BKQuery=/validation/MC11a/Beam3500GeV-2011-MagDown-Nu2-
↳ EmNoCuts/Sim05/Trig0x40760037Flagged/Reco12a/Stripping17Flagged/12463412/ALLSTREAMS.
↳ DST
```

Which should tell you the replica statistics, something like:

```
[fstagni@lxplus0032 ~]$ dirac-dms-replica-stats --BKQuery=/validation/MC11a/
↳ Beam3500GeV-2011-MagDown-Nu2-EmNoCuts/Sim05/Trig0x40760037Flagged/Reco12a/
↳ Stripping17Flagged/12463412/ALLSTREAMS.DST
Executing BK query: {'Visible': 'Yes', 'ConfigName': 'validation',
↳ 'ConditionDescription': 'Beam3500GeV-2011-MagDown-Nu2-EmNoCuts', 'EventType':
↳ '12463412', 'FileType': 'ALLSTREAMS.DST', 'ConfigVersion': 'MC11a', 'ProcessingPass
↳ ': '/Sim05/Trig0x40760037Flagged/Reco12a/Stripping17Flagged', 'SimulationConditions
↳ ': 'Beam3500GeV-2011-MagDown-Nu2-EmNoCuts'}
```

34 files (0.0 TB) in directories:
 /lhcb/validation/MC11a/ALLSTREAMS.DST/00000654/0000 34 files
 34 files found with replicas

Replica statistics:

```
0 archives: 0 files
1 archives: 25 files
2 archives: 9 files
0 replicas: 0 files
1 replicas: 0 files
2 replicas: 0 files
3 replicas: 33 files
4 replicas: 0 files
5 replicas: 1 files
```

SE statistics:

```
    CERN-ARCHIVE: 15 files
    CNAF-ARCHIVE: 5 files
  GRIDKA-ARCHIVE: 11 files
    IN2P3-ARCHIVE: 1 files
    RAL-ARCHIVE: 8 files
    SARA-ARCHIVE: 3 files
  CERN_MC_M-DST: 34 files
    CNAF_MC-DST: 4 files
    CNAF_MC_M-DST: 8 files
    GRIDKA_MC-DST: 1 files
  GRIDKA_MC_M-DST: 3 files
    IN2P3_MC-DST: 9 files
    IN2P3_MC_M-DST: 6 files
    PIC_MC-DST: 5 files
    PIC_MC_M-DST: 4 files
    RAL_MC-DST: 20 files
    RAL_MC_M-DST: 6 files
    SARA_MC-DST: 3 files
    SARA_MC_M-DST: 1 files
```

Sites statistics:

```
    LCG.CERN.ch: 34 files
    LCG.CNAF.it: 12 files
  LCG.GRIDKA.de: 4 files
    LCG.IN2P3.fr: 15 files
    LCG.PIC.es: 9 files
```

(continues on next page)

(continued from previous page)

```
LCG.RAL.uk: 26 files
LCG.SARA.nl: 4 files
```

Later, when you see that at least 2 replicas exist, you can issue

```
dirac-dms-add-replication --BKQuery=/validation/MC11a/Beam3500GeV-2011-MagDown-Nu2-
↳EmNoCuts/Sim05/Trig0x40760037Flagged/Reco12a/Stripping17Flagged/12463412/ALLSTREAMS.
↳DST --Plugin=DeleteReplicas --NumberOfReplicas=1 --Start
```

3.3 Acceptance test steps

3.3.1 Installation of LHCbDirac

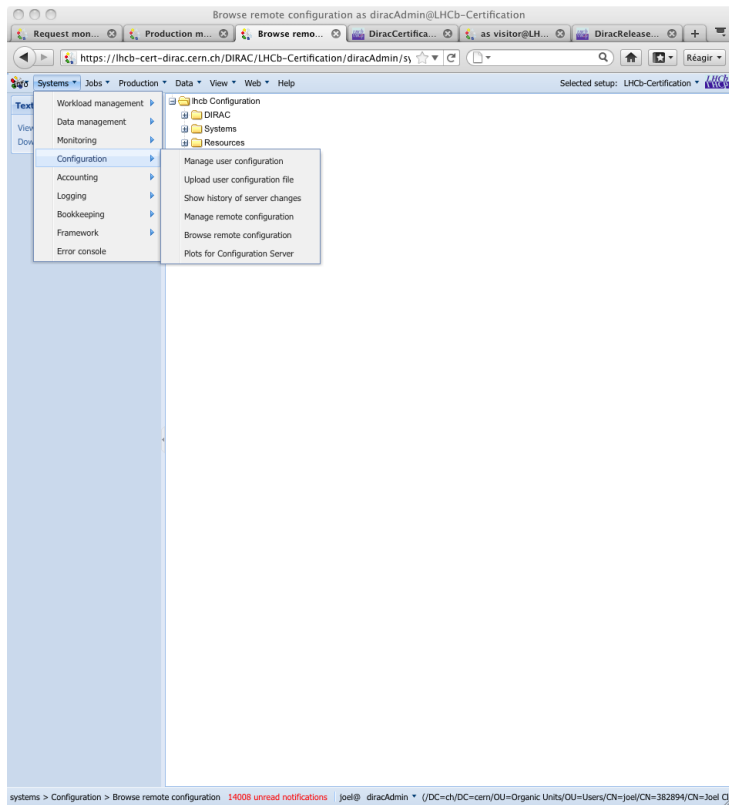
Login to a machine where LHCbDirac is already installed. Set the LHCbDirac environment, get a proxy with admin rights and launch the sysadmin CLI

```
lb-run LHCbDirac/prod bash
lhcb-proxy-init -g diracAdmin
dirac-admin-sysadmin-cli
```

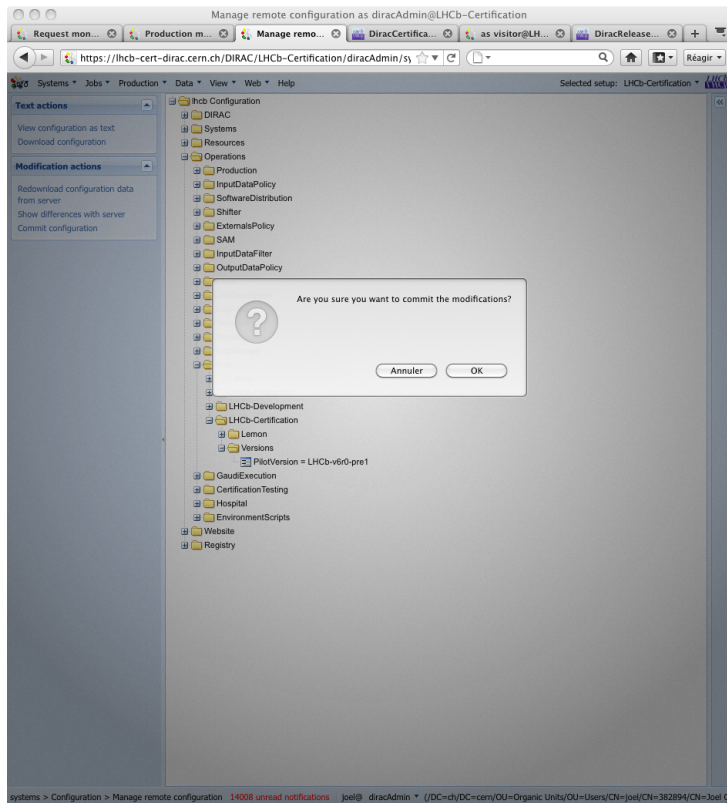
Update the LHCbDirac version and restart all the services

```
set host volhcbXX.cern.ch
update LHCb-vArBpC
restart *
```

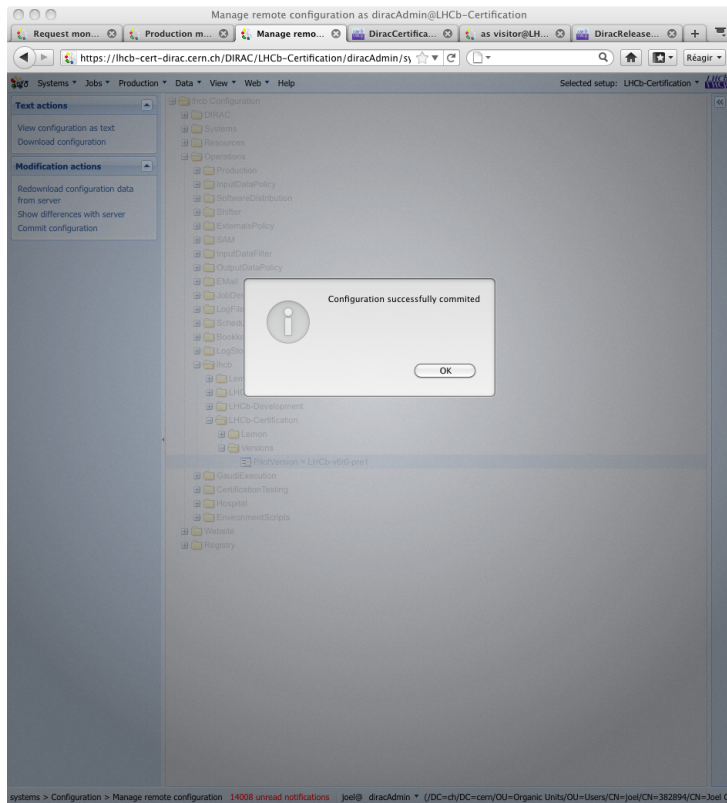
Change the version of the pilot in the CS. Go to the web portal, login with your certificate and the role **diracAdmin**. Click on **Systems, Configuration** and **Manage Remote configuration**.



The version is in the section /Operations/lhcb/LHCb-Certification/Versions/PilotVersion. Clicks on the **PilotVersion** and on change option value. Once you have changed the version number, click on **submit**. and do not forget to commit the change.



So you click on the left column on **Commit Configuration**



Now you should restart the task queue director

```
cd /opt/dirac/runit
runsvctrl d WorkloadManagement/TaskQueueDirector
runsvctrl u WorkloadManagement/TaskQueueDirector
```

3.3.2 Production test activity

Open your browser and connect to the certification instance of the LHCbDirac web portal (<http://lhcb-cert-dirac.cern.ch>) select the setup **LHCb-Certification** and load your certificate in the portal. Check that that your role is **lhcb_user**. Go to the tab **Production** and click on the **Requests** choice

Requests management as lhcb_user@LHCb-Certification

https://lhcb-cert-dirac.cern.ch/DIRAC/LHCb-Certification/lhcb_user/Production/ProductionRequest/display#

Systems Jobs Production Data View Web Help

Registered Production Requests

/ Requests

Id	Type	State	Priority	Name	Sim/Run conditions	Proc. pass	Event type	Events requested
18	Simulation	Accepted	2b	Template for certification	Beam3500GeV-VeloClosed-MagDown-I	2010-Sim08Trig0x002a002aReco07-wit	12143001	
17	Simulation	BK Check	2b	Template for certification	Beam3500GeV-VeloClosed-MagDown-I	2010-Sim08Trig0x002a002aReco07-wit	12143001	
14	Simulation	New	2b	Certification_MC_Test_Ignore	Beam3500GeV-VeloClosed-MagUp-Nu	CertificationTest01		20
11	Simulation	Accepted	2b	Certification_MC_Test_Ignore	Beam3500GeV-VeloClosed-MagUp-Nu	CertificationTest01		20
10	Simulation	Active	2b	(certification) MC Standard 3C	Beam450GeV-VeloClosed15mm-MagD	2009-Sim06Reco04-withTruth	30000000	1,000
9	Reconstruction	Accepted	2b	For testing Reconstruction on	Beam4000GeV-MagOff	Reco01-Brunel-v34r7-Online-Test	90000000	
8	Reconstruction	New	2b	For testing Reconstruction on	Beam450GeV-Mag-100%	Reco01-Brunel-v34r7-Online-Test	30000000	
6	Reconstruction	Accepted	2b	For testing Reconstruction on	ALL	Reco01-Brunel-v34r7-Online-Test	30000000	
5	Reconstruction	Submitted	2b	For testing Reconstruction on	ALL	Reco01	30000000	
4	Reconstruction	Submitted	2b	For testing Reconstruction on	ALL	Reco01	30000000	
3	Reconstruction	Submitted	2b	For testing Reconstruction on	ALL	Reco01	30000000	
2	Simulation	Active	2b	MC Standard 30000000 10 eV	Beam5TeV-VeloClosed-MagDown-Nu1	MC09-Sim06Reco02-withoutTruth	30000000	1,000
1	Simulation	Done	2b	MC Standard 30000000 10 eV	Beam5TeV-VeloClosed-MagDown-Nu1	MC09-Sim06Reco02-withoutTruth	30000000	1,000

Page 1 of 1 25 New request

production > Requests joel@ lhcb_user (/DC=ch/DC=cern/OU=Organic Units/OU=Users/

Click on the production which is defined label “template for certification” (nb = 28) and in the menu which appears select **Duplicate**

Requests management as lhcb_user@LHCb-Certification

https://lhcb-cert-dirac.cern.ch/DIRAC/LHCb-Certification/lhcb_user/Production/ProductionRequest/display#

Systems Jobs Production Data View Web Help

Registered Production Requests

/ Requests / 18

Id	Type	State	Priority	Name	Sim/Run conditions	Proc. pass	Event type	Events requested	Events in BK	Progress (%)
18	Simulation	Accepted	2b	Template for certification	Beam3500GeV-VeloClosed-MagDown-I	2010-Sim08Trig0x002a002aReco07-wit	12143001	100	0	0
17	Simulation	Accepted	2b	Template for certification	Beam3500GeV-VeloClosed-MagDown-I	2010-Sim08Trig0x002a002aReco07-wit	12143001	100	0	0
14	Simulation	New	2b	Certification_MC_Test_Ignore	Beam3500GeV-VeloClosed-MagUp-Nu	CertificationTest01		20,000	0	0
11	Simulation	Accepted	2b	Certification_MC_Test_Ignore	Beam3500GeV-VeloClosed-MagUp-Nu	CertificationTest01		20,000	0	0
10	Simulation	Active	2b	(certification) MC Standard 300000000	Beam450GeV-VeloClosed15mm-MagDi	2009-Sim06Reco04-withTruth	300000000	1,000,000	36,545,778	3654
9	Reconstruction	Accepted	2b	For testing Reconstruction on	Beam4000GeV-MagOff	Reco01-Brunel-v34r7-Online-Test		900000000	0	0
8	Reconstruction	New	2b	For testing Reconstruction on	Beam450GeV-Mag-100%	Reco01-Brunel-v34r7-Online-Test		300000000	0	0
6	Reconstruction	Accepted	2b	For testing Reconstruction on	ALL	Reco01-Brunel-v34r7-Online-Test		300000000	0	0
5	Reconstruction	Submitted	2b	For testing Reconstruction on	ALL	Reco01		300000000	0	0
4	Reconstruction	Submitted	2b	For testing Reconstruction on	ALL	Reco01		300000000	0	0
3	Reconstruction	Submitted	2b	For testing Reconstruction on	ALL	Reco01		300000000	0	0
2	Simulation	Active	2b	MC Standard 300000000	Beam5TeV-VeloClosed-MagDown-Nu1	MC09-Sim06Reco02-withoutTruth		300000000	1,000,000	0
1	Simulation	Done	2b	MC Standard 300000000	Beam5TeV-VeloClosed-MagDown-Nu1	MC09-Sim06Reco02-withoutTruth		300000000	1,000,000	0

Page 1 of 1 25 New request

production > Requests joel@ lhcb_user (/DC=ch/DC=cern/OU=Organic Units/OU=Users/

You are asked if you want to **Clear the processing pass in the copy**. Select **No**. This will keep all the steps which are pre-defined.

Requests management as lhcb_user@LHCb-Certification

https://lhcb-cert-dirac.cern.ch/DIRAC/LHCb-Certification/lhcb_user/Production/ProductionRequest/display#

Systems Jobs Production Data View Web Help

Registered Production Requests

/ Requests / 18

Id	Type	State	Priority	Name	Sim/Run conditions	Proc. pass	Event type	Events requested	Events in BK	Progress (%)
18	Simulation	Accepted	2b	Template for certification	Beam3500GeV-VeloClosed-MagDown-I	2010-Sim08Trig0x002a002aReco07-wit	12143001	100	0	0
17	Simulation	BK Check	2b	Template for certification	Beam3500GeV-VeloClosed-MagDown-I	2010-Sim08Trig0x002a002aReco07-wit	12143001	100	0	0
14	Simulation	New	2b	Certification_MC_Test_Ignore	Beam3500GeV-VeloClosed-MagUp-Nu	CertificationTest01		20,000	0	0
11	Simulation	Accepted	2b	Certification_MC_Test_Ignore	Beam3500GeV-VeloClosed-MagUp-Nu	CertificationTest01		20,000	0	0
10	Simulation	Active	2b	(certification) MC Standard 300000000	Beam450GeV-VeloClosed15mm-MagDi	2009-Sim06Reco04-withTruth	300000000	1,000,000	36,545,778	3654
9	Reconstruction	Accepted	2b	For testing Reconstruction on	Beam4000GeV-MagOff	Reco01-Brunel-v34r7-Online-Test		900000000	0	0
8	Reconstruction	New	2b	For testing Reconstruction on	Beam450GeV-Mag-100%	Reco01-Brunel-v34r7-Online-Test		300000000	0	0
6	Reconstruction	Accepted	2b	For testing Reconstruction on	ALL	Reco01-Brunel-v34r7-Online-Test		300000000	0	0
5	Reconstruction	Submitted	2b	For testing Reconstruction on	ALL	Reco01		300000000	0	0
4	Reconstruction	Submitted	2b	For testing Reconstruction on	ALL	Reco01		300000000	0	0
3	Reconstruction	Submitted	2b	For testing Reconstruction on	ALL	Reco01		300000000	0	0
2	Simulation	Active	2b	MC Standard 300000000	Beam5TeV-VeloClosed-MagDown-Nu1	MC09-Sim06Reco02-withoutTruth		300000000	1,000,000	0
1	Simulation	Done	2b	MC Standard 300000000	Beam5TeV-VeloClosed-MagDown-Nu1	MC09-Sim06Reco02-withoutTruth		300000000	1,000,000	0

Page 1 of 1 25 New request

production > Requests joel@ lhcb_user (/DC=ch/DC=cern/OU=Organic Units/OU=Users/CN=joel/CN=382894/CN=Joel Closier)

The new request is created and you get a number that will appear in the web page.

Requests management as lhcb_user@LHCb-Certification

https://lhcb-cert-dirac.cern.ch/DIRAC/LHCb-Certification/lhcb_user/Production/ProductionRequest/display#

Registered Production Requests

/ Requests / 18

Id	Type	State	Priority	Name	Sim/Run conditions	Proc. pass	Event type	Events requested	Events in BK	Progress (%)
19	Simulation	New	2b	Template for certification	Beam3500GeV-VeloClosed-MagDown-I	2010-Sim08Trig0x002a002aReco07-wit	12143001	100	0	0
18	Simulation	Accepted	2b	Template for certification	Beam3500GeV-VeloClosed-MagDown-I	2010-Sim08Trig0x002a002aReco07-wit	12143001	100	0	0
17	Simulation	BK Check	2b	Template for certification	Beam3500GeV-VeloClosed-MagDown-I	2010-Sim08Trig0x002a002aReco07-wit	12143001	100	0	0
14	Simulation	New	2b	Certification_MC_Test_Ignore	Beam3500GeV-VeloClosed-MagUp-Nu	CertificationTest01		20,000	0	0
11	Simulation	Accepted	2b	Certification_MC_Test_Ignore	Beam3500GeV-VeloClosed-MagUp-Nu	CertificationTest01		20,000	0	0
10	Simulation	Active	2b	(certification) MC Standard 3C	Beam450GeV-VeloClosed15mm-MagD	2009-Sim06Reco04-withTruth	30000000	1,000,000	36,545,778	3654
9	Reconstruction	Accepted	2b	For testing Reconstruction on	Beam4000GeV-MagOff	Reco01-Brunel-v34r7-Online-Test	90000000	0	0	
8	Reconstruction	New	2b	For testing Reconstruction on	Beam450GeV-Mag-100%	Reco01-Brunel-v34r7-Online-Test	30000000	0	0	
6	Reconstruction	Accepted	2b	For testing Reconstruction on	ALL	Reco01-Brunel-v34r7-Online-Test	30000000	0	0	
5	Reconstruction	Submitted	2b	For testing Reconstruction on	ALL	Reco01	30000000	0	0	
4	Reconstruction	Submitted	2b	For testing Reconstruction on	ALL	Reco01	30000000	0	0	
3	Reconstruction	Submitted	2b	For testing Reconstruction on	ALL	Reco01	30000000	0	0	
2	Simulation	Active	2b	MC Standard 30000000 10 ev	Beam5TeV-VeloClosed-MagDown-Nu1	MC09-Sim06Reco02-withoutTruth	30000000	1,000,000	0	0
1	Simulation	Done	2b	MC Standard 30000000 10 ev	Beam5TeV-VeloClosed-MagDown-Nu1	MC09-Sim06Reco02-withoutTruth	30000000	1,000,000	0	0

Request was successfully duplicated
New Request ID: 19
OK

Page 1 of 1 25 New request

No requests are registered

production > Requests joel@lhcb_user (/DC=ch/DC=cern/OU=Organic Units/OU=Users/CN=joel/CN=382894/CN=Joel Closier)

Click on the new request that you just created the step below and select the **edit** option

Requests management as lhcb_user@LHCb-Certification

https://lhcb-cert-dirac.cern.ch/DIRAC/LHCb-Certification/lhcb_user/Production/ProductionRequest/display#

Registered Production Requests

/ Requests / 19

Id	Type	State	Priority	Name	Sim/Run conditions	Proc. pass	Event type	Events requested	Events in BK	Progress (%)
19	Simulation	New	2b	Template for certification	Beam3500GeV-VeloClosed-MagDown-I	2010-Sim08Trig0x002a002aReco07-wit	12143001	100	0	0
18	Simulation	Accepted	2b	Template for certification	Beam3500GeV-VeloClosed-MagDown-I	2010-Sim08Trig0x002a002aReco07-wit	12143001	100	0	0
17	Simulation	BK Check	2b	Template for certification	Beam3500GeV-VeloClosed-MagDown-I	2010-Sim08Trig0x002a002aReco07-wit	12143001	100	0	0
14	Simulation	New	2b	Certification_MC_Test_Ignore	Beam3500GeV-VeloClosed-MagUp-Nu	CertificationTest01		20,000	0	0
11	Simulation	Accepted	2b	Certification_MC_Test_Ignore	Beam3500GeV-VeloClosed-MagUp-Nu	CertificationTest01		20,000	0	0
10	Simulation	Active	2b	(certification) MC Standard 3C	Beam450GeV-VeloClosed15mm-MagD	2009-Sim06Reco04-withTruth	30000000	1,000,000	36,545,778	3654
9	Reconstruction	Accepted	2b	For testing Reconstruction on	Beam4000GeV-MagOff	Reco01-Brunel-v34r7-Online-Test	90000000	0	0	
8	Reconstruction	New	2b	For testing Reconstruction on	Beam450GeV-Mag-100%	Reco01-Brunel-v34r7-Online-Test	30000000	0	0	
6	Reconstruction	Accepted	2b	For testing Reconstruction on	ALL	Reco01-Brunel-v34r7-Online-Test	30000000	0	0	
5	Reconstruction	Submitted	2b	For testing Reconstruction on	ALL	Reco01	30000000	0	0	
4	Reconstruction	Submitted	2b	For testing Reconstruction on	ALL	Reco01	30000000	0	0	
3	Reconstruction	Submitted	2b	For testing Reconstruction on	ALL	Reco01	30000000	0	0	
2	Simulation	Active	2b	MC Standard 30000000 10 ev	Beam5TeV-VeloClosed-MagDown-Nu1	MC09-Sim06Reco02-withoutTruth	30000000	1,000,000	0	0
1	Simulation	Done	2b	MC Standard 30000000 10 ev	Beam5TeV-VeloClosed-MagDown-Nu1	MC09-Sim06Reco02-withoutTruth	30000000	1,000,000	0	0

Request 19
View
Windowed view
History
Edit
Duplicate
Delete
Add subrequest

Page 1 of 1 25 New request

No requests are registered

production > Requests joel@lhcb_user (/DC=ch/DC=cern/OU=Organic Units/OU=Users/CN=joel/CN=382894/CN=Joel Closier)

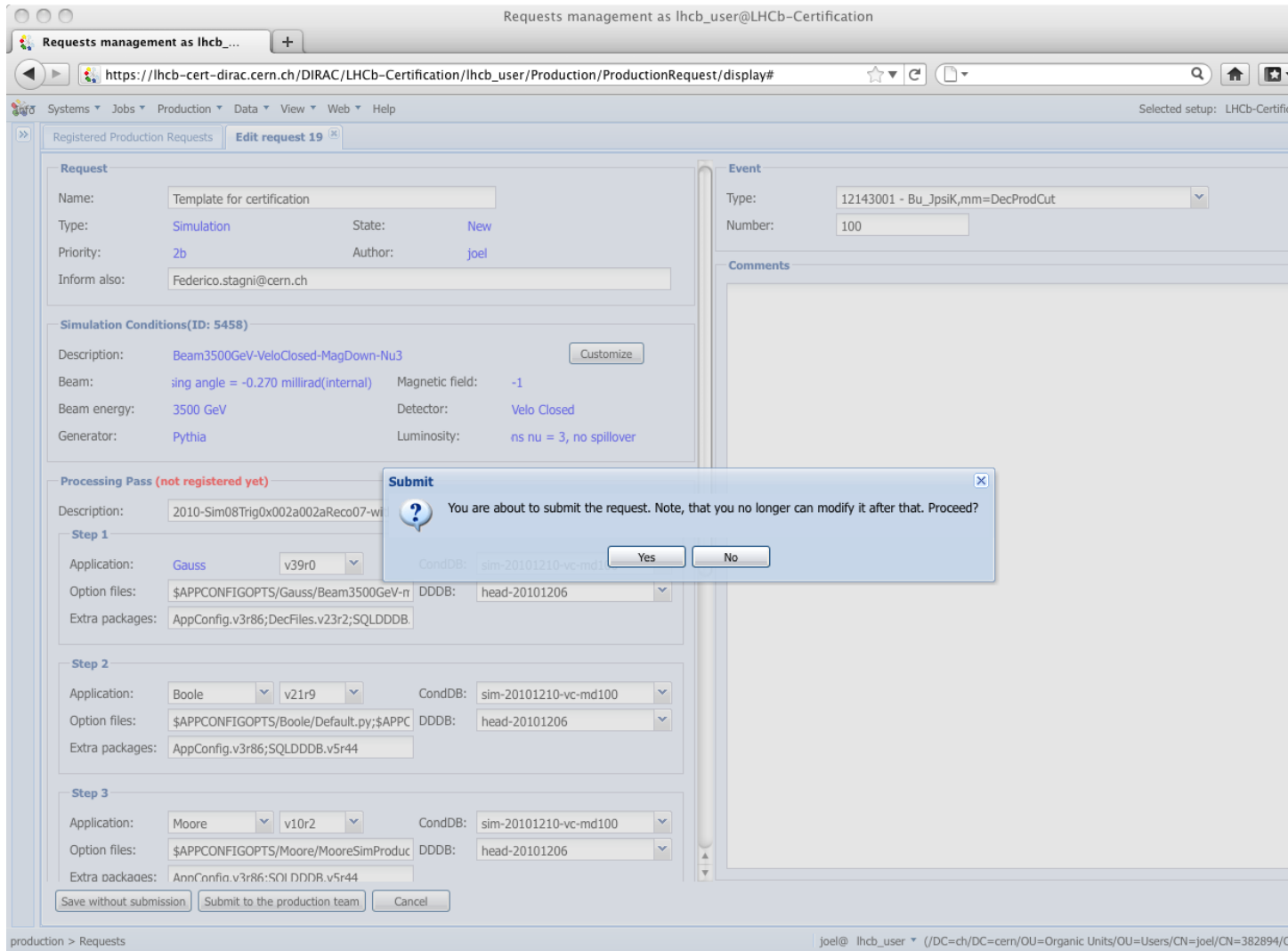
Then modify all the fields which needs a new value. Once you have finished, submit your request to the production team.

The screenshot shows the 'Requests management as lhcb_user@LHCb-Certification' web interface. The browser address bar shows the URL: `https://lhcb-cert-dirac.cern.ch/DIRAC/LHCb-Certification/lhcb_user/Production/ProductionRequest/display#`. The page title is 'Requests management as lhcb_user@LHCb-Certification'. The main content area is divided into several sections:

- Request:**
 - Name: Template for certification
 - Type: Simulation
 - State: New
 - Priority: 2b
 - Author: joel
 - Inform also: Federico.stagni@cern.ch
- Simulation Conditions (ID: 5458):**
 - Description: Beam3500GeV-VeloClosed-MagDown-Nu3
 - Beam: sing angle = -0.270 millirad(internal)
 - Magnetic field: -1
 - Beam energy: 3500 GeV
 - Detector: Velo Closed
 - Generator: Pythia
 - Luminosity: ns nu = 3, no spillover
- Processing Pass (not registered yet):**
 - Description: 2010-Sim08Trig0x002a002aReco07-withTruth-Test
- Step 1:**
 - Application: Gauss
 - CondDB: sim-20101210-vc-md100
 - Option files: \$APPCONFIGOPTS/Gauss/Beam3500GeV-m
 - DDDB: head-20101206
 - Extra packages: AppConfig.v3r86;DecFiles.v23r2;SQLDDDB
- Step 2:**
 - Application: Boole
 - CondDB: sim-20101210-vc-md100
 - Option files: \$APPCONFIGOPTS/Boole/Default.py;\$APPC
 - DDDB: head-20101206
 - Extra packages: AppConfig.v3r86;SQLDDDB.v5r44
- Step 3:**
 - Application: Moore
 - CondDB: sim-20101210-vc-md100
 - Option files: \$APPCONFIGOPTS/Moore/MooreSimProduc
 - DDDB: head-20101206
 - Extra packages: AppConfig.v3r86;SQLDDDB.v5r44

At the bottom of the form, there are three buttons: 'Save without submission', 'Submit to the production team', and 'Cancel'. On the right side of the form, there is an 'Event' section with a dropdown menu for 'Type' (selected: 12143001 - Bu_JpsiK,mm=DecProdCut) and a text field for 'Number' (100). Below the 'Event' section is a large 'Comments' area.

You have just to approve it.



Now you should change your role to become **lhcb_tech** and **lhcb_ppg** to validate the request. You click on the new request and in the menu you choose the option *sign*

Requests management as lhcb_tech@LHCb-Certification

https://lhcb-cert-dirac.cern.ch/DIRAC/LHCb-Certification/lhcb_tech/Production/ProductionRequest/display#

Systems Jobs Production Data View Web Help Selected setup: LHCb-Certification

Registered Production Requests

/ Requests / 19

ID	Type	State	Priority	Name	Sim/Run conditions	Proc. pass	Event type	Events requested	Events in BK	Progress (%)
19	Simulation	Submitted	2b	Template for certification	Beam3500GeV-VeloClosed-MagDown-I	2010-Sim08Trig0x002a002aReco07-wit	12143001	100	0	0
18	Simulation	Accepted	2b	Template for certification	3500GeV-VeloClosed-MagDown-I	2010-Sim08Trig0x002a002aReco07-wit	12143001	100	0	0
17	Simulation	BK Check	2b	Template for certification	3500GeV-VeloClosed-MagDown-I	2010-Sim08Trig0x002a002aReco07-wit	12143001	100	0	0
14	Simulation	New	2b	Certification_MC_Test	3500GeV-VeloClosed-MagUp-Nu	CertificationTest01		20,000	0	0
11	Simulation	Accepted	2b	Certification_MC_Test	3500GeV-VeloClosed-MagUp-Nu	CertificationTest01		20,000	0	0
10	Simulation	Active	2b	Certification_MC_Test	450GeV-VeloClosed15mm-MagDown-I	2009-Sim06Reco04-withTruth	30000000	1,000,000	36,545,778	3654
9	Reconstruction	Accepted	2b	For testing Reconstruction	4000GeV-MagOff	Reco01-Brunel-v34/7-Online-Test	90000000	0	0	0
8	Reconstruction	New	2b	For testing Reconstruction	450GeV-Mag-100%	Reco01-Brunel-v34/7-Online-Test	30000000	0	0	0
6	Reconstruction	Accepted	2b	For testing Reconstruction		Reco01-Brunel-v34/7-Online-Test	30000000	0	0	0
5	Reconstruction	Submitted	2b	For testing Reconstruction	ALL	Reco01	30000000	0	0	0
4	Reconstruction	Submitted	2b	For testing Reconstruction	ALL	Reco01	30000000	0	0	0
3	Reconstruction	Submitted	2b	For testing Reconstruction	ALL	Reco01	30000000	0	0	0
2	Simulation	Active	2b	MC Standard 30000000	Beam5TeV-VeloClosed-MagDown-Nu1	MC09-Sim06Reco02-withoutTruth	30000000	1,000,000	0	0
1	Simulation	Done	2b	MC Standard 30000000	Beam5TeV-VeloClosed-MagDown-Nu1	MC09-Sim06Reco02-withoutTruth	30000000	1,000,000	0	0

Page 1 of 1 25

No requests are registered

production > Requests joel@lhcb_tech (/DC=ch/DC=cern/OU=Organic Units/OU=Users/CN=joel/CN=382894/CN=Joel Closier)

Requests management as lhcb_ppg@LHCb-Certification

https://lhcb-cert-dirac.cern.ch/DIRAC/LHCb-Certification/lhcb_ppg/Production/ProductionRequest/display#

Systems Jobs Production Data View Web Help Selected setup: LHCb-Certification

Registered Production Requests

/ Requests / 19

ID	Type	State	Priority	Name	Sim/Run conditions	Proc. pass	Event type	Events requested	Events in BK	Progress (%)
19	Simulation	Tech OK	2b	Template for certification	Beam3500GeV-VeloClosed-MagDown-I	2010-Sim08Trig0x002a002aReco07-wit	12143001	100	0	0
18	Simulation	Accepted	2b	Template for certification	3500GeV-VeloClosed-MagDown-I	2010-Sim08Trig0x002a002aReco07-wit	12143001	100	0	0
17	Simulation	BK Check	2b	Template for certification	3500GeV-VeloClosed-MagDown-I	2010-Sim08Trig0x002a002aReco07-wit	12143001	100	0	0
14	Simulation	New	2b	Certification_MC_Test	3500GeV-VeloClosed-MagUp-Nu	CertificationTest01		20,000	0	0
11	Simulation	Accepted	2b	Certification_MC_Test	3500GeV-VeloClosed-MagUp-Nu	CertificationTest01		20,000	0	0
10	Simulation	Active	2b	Certification_MC_Test	450GeV-VeloClosed15mm-MagDown-I	2009-Sim06Reco04-withTruth	30000000	1,000,000	36,545,778	3654
9	Reconstruction	Accepted	2b	For testing Reconstruction	4000GeV-MagOff	Reco01-Brunel-v34/7-Online-Test	90000000	0	0	0
8	Reconstruction	New	2b	For testing Reconstruction	450GeV-Mag-100%	Reco01-Brunel-v34/7-Online-Test	30000000	0	0	0
6	Reconstruction	Accepted	2b	For testing Reconstruction		Reco01-Brunel-v34/7-Online-Test	30000000	0	0	0
5	Reconstruction	Submitted	2b	For testing Reconstruction	ALL	Reco01	30000000	0	0	0
4	Reconstruction	Submitted	2b	For testing Reconstruction	ALL	Reco01	30000000	0	0	0
3	Reconstruction	Submitted	2b	For testing Reconstruction	ALL	Reco01	30000000	0	0	0
2	Simulation	Active	2b	MC Standard 30000000	Beam5TeV-VeloClosed-MagDown-Nu1	MC09-Sim06Reco02-withoutTruth	30000000	1,000,000	0	0
1	Simulation	Done	2b	MC Standard 30000000	Beam5TeV-VeloClosed-MagDown-Nu1	MC09-Sim06Reco02-withoutTruth	30000000	1,000,000	0	0

Page 1 of 1 25

No requests are registered

production > Requests joel@lhcb_ppg (/DC=ch/DC=cern/OU=Organic Units/OU=Users/CN=joel/CN=382894/CN=Joel Closier)

You can sign or reject the request.

Requests management as lhcb_ppg@LHCb-Certification

https://lhcb-cert-dirac.cern.ch/DIRAC/LHCb-Certification/lhcb_ppg/Production/ProductionRequest/display#

Registered Production Requests **Edit request 19**

Request

Name: Template for certification
 Type: Simulation State: Tech OK
 Priority: 2b Author: joel
 Inform also: Federico.stagni@cern.ch

Simulation Conditions (ID: 5458)

Description: Beam3500GeV-VeloClosed-MagDown-Nu3
 Beam: sing angle = -0.270 millirad(internal) Magnetic field: -1
 Beam energy: 3500 GeV Detector: Velo Closed
 Generator: Pythia Luminosity: ns nu = 3, no spillover

Processing Pass (not registered yet)

Description: 2010-Sim08Trig0x002a002aReco07-withTruth-Test

Step 1

Application: Gauss v39r0 CondDB: sim-20101210-vc-md100
 Option files: e).opts;\$LBPYTHIAROOT/options/Pythia.py DDB: head-20101206
 Extra packages: xfig.v3r86;DecFiles.v23r2;SQLDDB.v5r44

Step 2

Application: Boole v21r9 CondDB: sim-20101210-vc-md100
 Option files: \$APPCONFIGOPTS/Boole/Default.py;\$APPC DDB: head-20101206
 Extra packages: AppConfig.v3r86;SQLDDB.v5r44

Step 3

Application: Moore v10r2 CondDB: sim-20101210-vc-md100
 Option files: \$APPCONFIGOPTS/Moore/MooreSimProd DDB: head-20101206
 Extra packages: AnnConfin.v3r86;SQL DDB: v5r44

Sign the request Reject the request (better first comment why) Cancel

production > Requests joel@ lhcb_ppg (/DC=ch/DC=cern/OU=Organic Units/OU=Users/CN=joel/CN=382894/CN=Joel Closer)

Once the request has been accepted by lhcb_ppg and lhcb_tech, the status become **accepted**. Choose now the role **lhcb_pmgr** and click on the request. Then choose the option *edit*

Requests management as lhcb_pmgr@LHCb-Certification

https://lhcb-cert-dirac.cern.ch/DIRAC/LHCb-Certification/lhcb_pmgr/Production/ProductionRequest/display#

Registered Production Requests

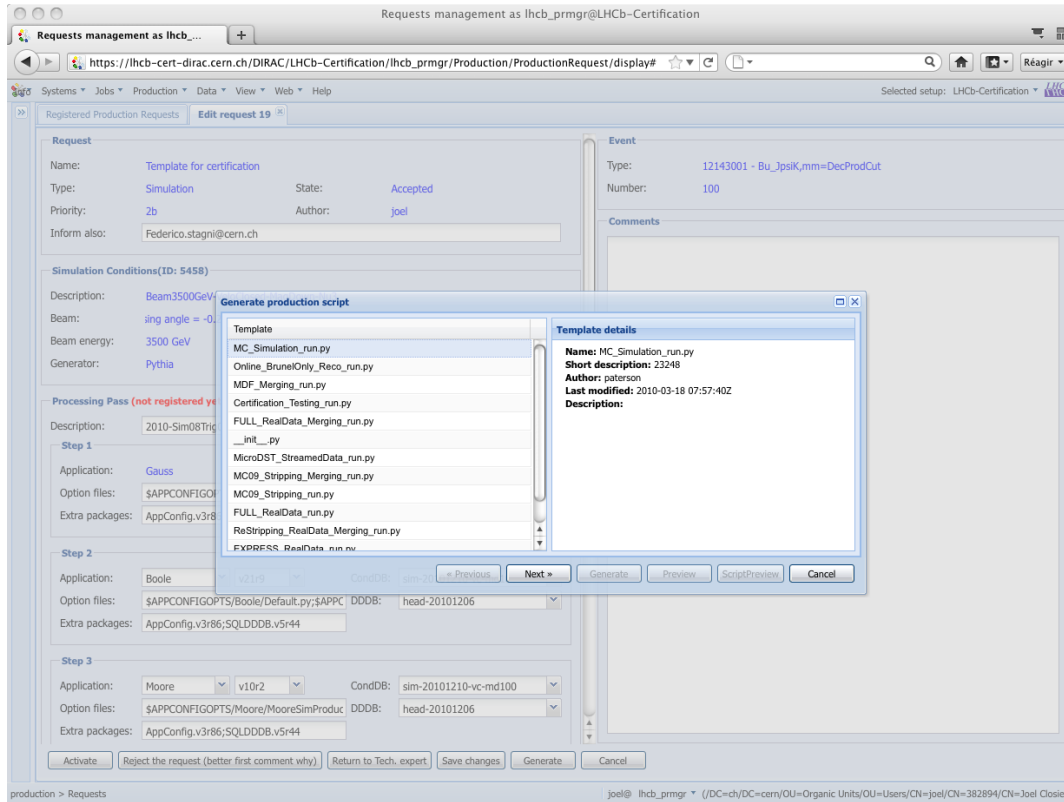
/ Requests / 19

Id	Type	State	Priority	Name	Sim/Run conditions	Proc. pass	Event type	Events requested	Events
19	Simulation	Accepted	2b	Template for certification	Beam3500GeV-VeloClosed-MagDown-I	2010-Sim08Trig0x002a002aReco07-wit	12143001	100	
18	Simulation	Accepted	2b	Template for certification	MagDown-I	2010-Sim08Trig0x002a002aReco07-wit	12143001	100	
17	Simulation	BK Check	2b	Template for certification	MagDown-I	2010-Sim08Trig0x002a002aReco07-wit	12143001	100	
14	Simulation	New	2b	Certification_MC_Test	MagUp-Nu	CertificationTest01		20,000	
11	Simulation	Accepted	2b	Certification_MC_Test	MagUp-Nu	CertificationTest01		20,000	
10	Simulation	Active	2b	(certification) MC Stan	mm-MagDi	2009-Sim06Reco04-withTruth	30000000	1,000,000	
9	Reconstruction	Accepted	2b	For testing Reconstru	Reco01-Brunel-v34r7-Online-Test		90000000	0	
8	Reconstruction	New	2b	For testing Reconstru	Reco01-Brunel-v34r7-Online-Test		30000000	0	
6	Reconstruction	Accepted	2b	For testing Reconstru	Reco01-Brunel-v34r7-Online-Test		30000000	0	
5	Reconstruction	Submitted	2b	For testing Reconstru	Reco01		30000000	0	
4	Reconstruction	Submitted	2b	For testing Reconstru	Reco01		30000000	0	
3	Reconstruction	Submitted	2b	For testing Reconstru	Reco01		30000000	0	
2	Simulation	Active	2b	MC Standard 3000000	Beam5TeV-VeloClosed-MagDown-Nu1	MC09-Sim06Reco02-withoutTruth	30000000	1,000,000	
1	Simulation	Done	2b	MC Standard 3000000	Beam5TeV-VeloClosed-MagDown-Nu1	MC09-Sim06Reco02-withoutTruth	30000000	1,000,000	

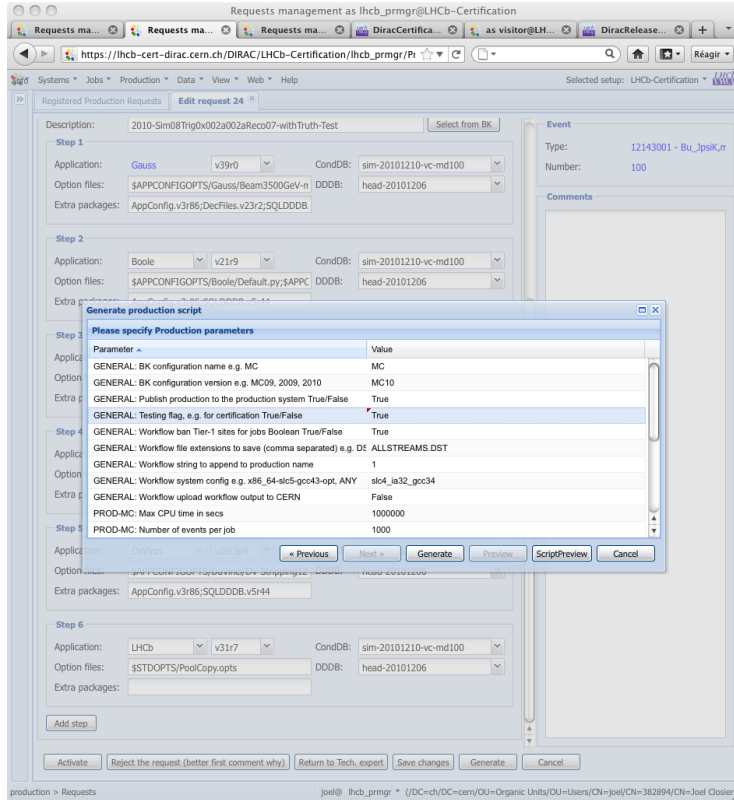
Page 1 of 1 25

production > Requests joel@ lhcb_pmgr (/DC=ch/DC=cern/OU=Organic Units/OU=Users/CN=joel/CN=382894/CN=Joel Closer)

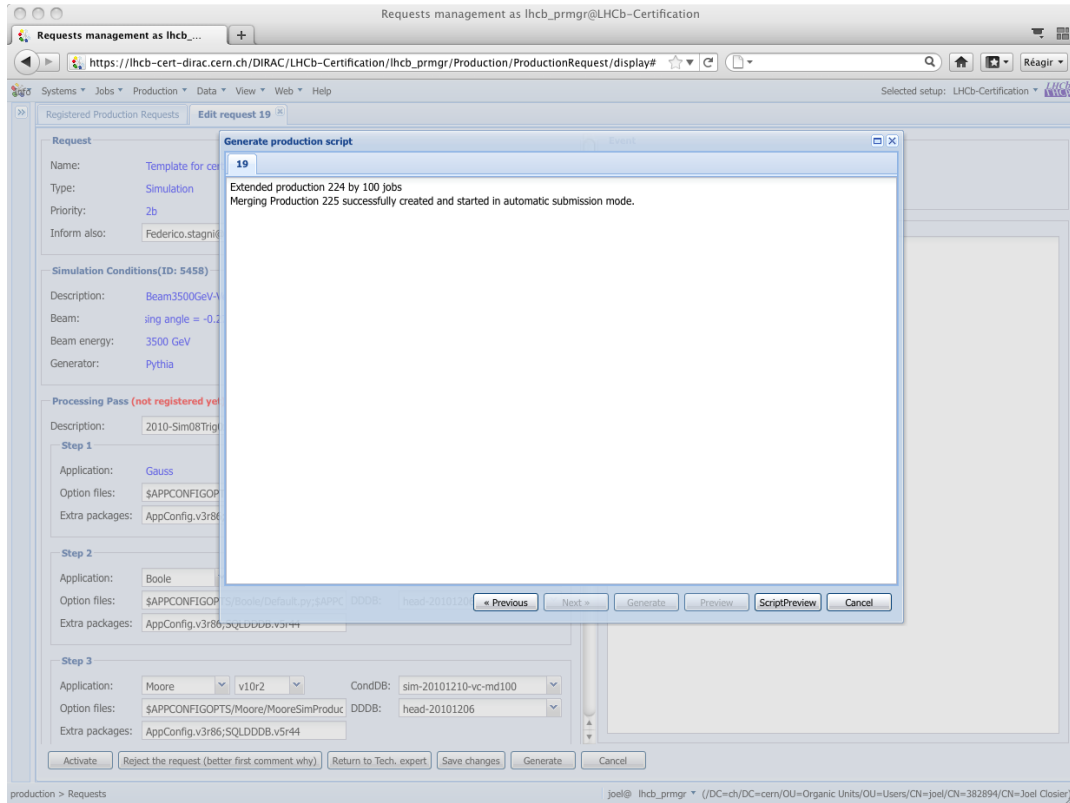
You give the correct Event Type and number of Events. Then you click on **Generate**. At this stage you are asked to choose which template should be used. In our case we will choose “MC_Simulation_run.py” and click on the **next** button.



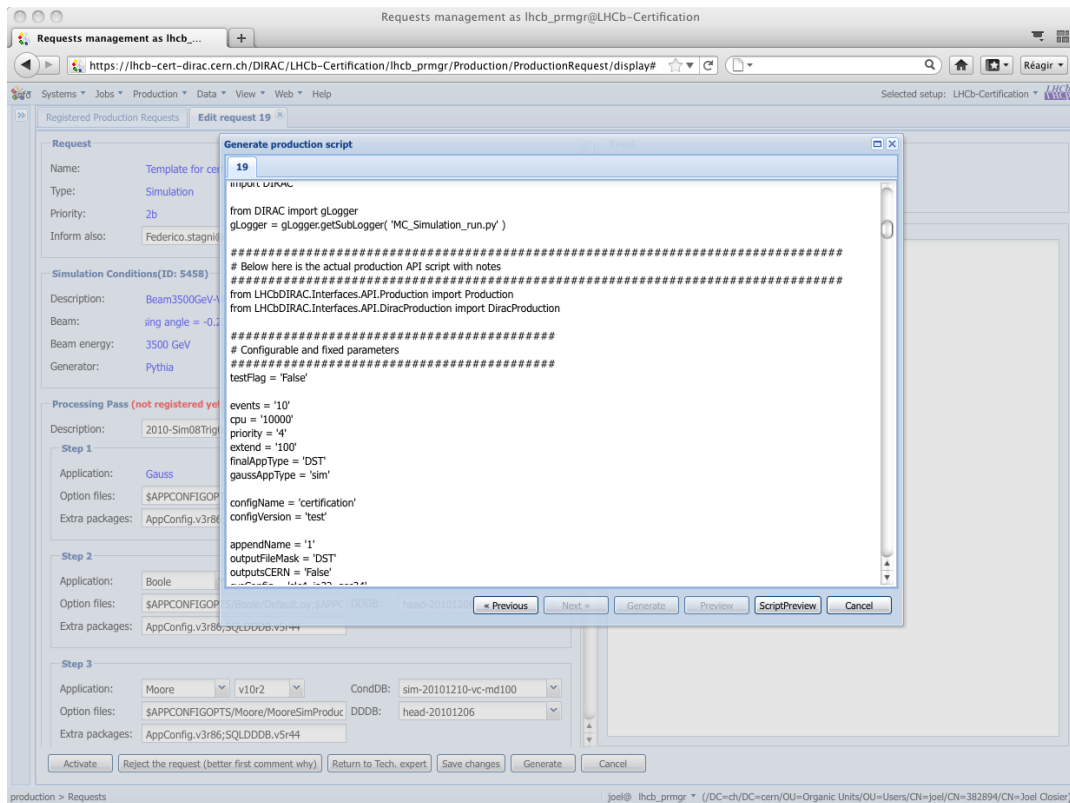
You get now the list of value that you could change before submitting the production. For the certification purpose you should change the value for “MC configuration name” to be **certification**, the “configuration version” should be **test**. Verify which plugin you want to use, the number of event that you want to process, the cputimelimit,... Once you have finished, click on the **generate** button.



After the generation of the production you will get in a new window the production ID and the number of jobs generated. If you want you can see and save the script which will generate this production by clicking on the **script preview** button.



This is the window of the python script which could be used to generate again the production. To exit this window click on **cancel**



If you click on the request and you choose **production monitor** you will be re-direct to the production monitor.

The screenshot shows the 'Requests management as lhcb_prmgr@LHCb-Certification' interface. The main table lists 19 requests. A context menu for 'Request 19' is open, showing options: View, Windowed view, History, Edit, Duplicate, Productions, and Production monitor. The 'Production monitor' option is highlighted.

Id	Type	State	Priority	Name	Sim/Run conditions	Proc. pass	Event type	Events requested	Events in BK	Progress (%)
19	Simulation	Accepted	2b	Template	Beam3500GeV-VeloClosed		x002a002aReco07-wit	12143001	100	0
18	Simulation	Accepted	2b	Template	Beam3500GeV-VeloClosed		x002a002aReco07-wit	12143001	100	0
17	Simulation	BK Check	2b	Template	Beam3500GeV-VeloClosed		x002a002aReco07-wit	12143001	100	0
14	Simulation	New	2b	Certificat	Beam3500GeV-VeloClosed				20,000	0
11	Simulation	Accepted	2b	Certificat	Beam3500GeV-VeloClosed				20,000	0
10	Simulation	Active	2b	(certificat)	Beam4500GeV-VeloClosed1		j04-withTruth	300000000	1,000,000	36,545,778
9	Reconstruction	Accepted	2b	For testir	Beam4000GeV-MagOff		347-Online-Test	900000000	0	0
8	Reconstruction	New	2b	For testir	Beam4500GeV-Mag-100%		347-Online-Test	300000000	0	0
6	Reconstruction	Accepted	2b	For testir	ALL		347-Online-Test	300000000	0	0
5	Reconstruction	Submitted	2b	For testir	ALL		347-Online-Test	300000000	0	0
4	Reconstruction	Submitted	2b	For testir	ALL		347-Online-Test	300000000	0	0
3	Reconstruction	Submitted	2b	For testir	ALL		Reco01	300000000	0	0
2	Simulation	Active	2b	MC Stan	Beam5TeV-VeloClosed-MagDown-Nu1		MC09-Sim06Reco02-withoutTruth	300000000	1,000,000	0
1	Simulation	Done	2b	MC Stan	Beam5TeV-VeloClosed-MagDown-Nu1		MC09-Sim06Reco02-withoutTruth	300000000	1,000,000	0

Production monitor with the fresh generated productions.

The screenshot shows the 'ProductionMonitor' interface. On the left, there are filters for Status, AgentType, Type, Group, Plugin, Date, ProductionID, and RequestID. The main table lists production jobs for Request 19.

ID	Status	AgentType	Type	Name	Files	Processed (%)	Created	Submitted	Waiting	Done
226	Active	Automatic	Replication	Request_19_ALLS	0	0	0	0	0	0
225	Active	Automatic	Merge	ALLSTREAMS.DS	0	0	0	0	0	0
224	Active	Automatic	MCSimulation	MC_Beam3500GeV	0	-	100	100	0	0

dirac-bookkeeping-production-informations 830 -o /DIRAC/Setup=LHCb-Certification

```

lxplus448] x86_64-slc5-gcc46-opt /afs/cern.ch/user/j/joel> dirac-bookkeeping-
↳production-informations 830 -o /DIRAC/Setup=LHCb-Certification
Production Info:
Configuration Name: LHCb
Configuration Version: Collision11
Event type: 91000000
-----
StepName: merging MDF
ApplicationName      : mergeMDF
ApplicationVersion   : None
OptionFiles          : None
DDDB                 : None
CONDDB               : None
ExtraPackages        :None
-----
Number of Steps      1
Total number of files: 2
    LOG:1
    RAW:1
Number of events
File Type            Number of events   Event Type            EventInputStat
RAW                  30988                91000000              30988
Path: /LHCb/Collision11/Beam3500GeV-VeloClosed-MagDown/Real Data/Merging
/LHCb/Collision11/Beam3500GeV-VeloClosed-MagDown/Real Data/Merging/91000000/RAW

```

You can then check the produced files:

```

nsls -l /castor/cern.ch/grid/lhcb/certification/test/ALLSTREAMS.DST/00000225/0000
dirac-dms-lfn-replicas /lhcb/certification/test/ALLSTREAMS.DST/00000225/0000/00000225_
↳00000001_1.allstreams.dst
dirac-dms-add-replication --Production 259:268 --FileType RADIATIVE.DST --Plugin_
↳LHCbMCDSTBroadcastRandom --Request 30
dirac-dms-add-replication --Production 239 --FileType ALLSTREAMS.DST --Plugin_
↳LHCbMCDSTBroadcastRandom --Request 29
Transformation 273 created
Name: Replication-ALLSTREAMS.DST-239-Request29 , Description:_
↳LHCbMCDSTBroadcastRandom of ALLSTREAMS.DST for productions 239
BK Query: {'FileType': ['ALLSTREAMS.DST'], 'ProductionID': ['239'], 'Visibility': 'Yes
↳'}
3 files found for that query
Plugin: LHCbMCDSTBroadcastRandom
RequestID: 29
[lxplus433] x86_64-slc5-gcc43-opt /afs/cern.ch/lhcb/software/DEV/LHCBDIRAC/LHCBDIRAC_
↳v6r0-pre12> dirac-bookkeeping-production-informations 239Production Info::
    Configuration Name: certification
    Configuration Version: test
    Event type: 12143001

StepName: MCMerging10
ApplicationName      : LHCb
ApplicationVersion   : v31r7
OptionFiles          : $STDOPPTS/PoolCopy.opts
DDB                  : head-20101206
CONDDB               : sim-20101210-vc-md100
ExtraPackages        :None

```

(continues on next page)

(continued from previous page)

```

Number of Steps      4
Total number of files: 8
      LOG:4
      ALLSTREAMS.DST:4
Number of events
File Type            Number of events    Event Type            EventInputStat
ALLSTREAMS.DST       540                12143001                540
Path: /certification/test/Beam3500GeV-VeloClosed-MagDown-Nu3/MC10Sim01-
↳Trig0x002e002aFlagged/Reco08/Stripping12Flagged
/certification/test/Beam3500GeV-VeloClosed-MagDown-Nu3/MC10Sim01-
↳Trig0x002e002aFlagged/Reco08/Stripping12Flagged/12143001/ALLSTREAMS.DST
dirac-bookkeeping-production-files 239 ALLSTREAMS.DST
FileName
↳                Size            GUID                                Replica
/lhcb/certification/test/ALLSTREAMS.DST/00000239/0000/00000239_00000044_1.allstreams.
↳dst                14515993    165DD5A9-1D40-E011-AD80-003048F1E1E0    Yes
/lhcb/certification/test/ALLSTREAMS.DST/00000239/0000/00000239_00000045_1.allstreams.
↳dst                2971054     988731FC-1C40-E011-AFCD-90E6BA442F3B    Yes
/lhcb/certification/test/ALLSTREAMS.DST/00000239/0000/00000239_00000074_1.allstreams.
↳dst                202748580   E2BAF0A1-A340-E011-BF97-003048F1B834    Yes
/lhcb/certification/test/ALLSTREAMS.DST/00000239/0000/00000239_00000076_1.allstreams.
↳dst                2804277     F086C525-EB43-E011-96F9-001EC9D8B181    Yes

[lxplus433] x86_64-slc5-gcc43-opt /afs/cern.ch/lhcb/software/DEV/LHCBDIRAC/LHCBDIRAC_
↳v6r0-pre12> dirac-dms-lfn-replicas /lhcb/certification/test/ALLSTREAMS.DST/00000239/
↳0000/00000239_00000044_1.allstreams.dst
{'Failed': {},
 'Successful': {'/lhcb/certification/test/ALLSTREAMS.DST/00000239/0000/00000239_
↳00000044_1.allstreams.dst': {'CERN_MC_M-DST': 'srm://srm-lhcb.cern.ch/castor/cern.
↳ch/grid/lhcb/certification/test/ALLSTREAMS.DST/00000239/0000/00000239_00000044_1.
↳allstreams.dst'}}}
[lxplus433] x86_64-slc5-gcc43-opt /afs/cern.ch/lhcb/software/DEV/LHCBDIRAC/LHCBDIRAC_
↳v6r0-pre12> dirac-dms-lfn-replicas /lhcb/certification/test/ALLSTREAMS.DST/00000239/
↳0000/00000239_00000045_1.allstreams.dst
{'Failed': {},
 'Successful': {'/lhcb/certification/test/ALLSTREAMS.DST/00000239/0000/00000239_
↳00000045_1.allstreams.dst': {'CNAF_MC_M-DST': 'srm://storm-fe-lhcb.cr.cnaf.infn.it/
↳tld1/lhcb/certification/test/ALLSTREAMS.DST/00000239/0000/00000239_00000045_1.
↳allstreams.dst'}}}
[lxplus433] x86_64-slc5-gcc43-opt /afs/cern.ch/lhcb/software/DEV/LHCBDIRAC/LHCBDIRAC_
↳v6r0-pre12> dirac-dms-lfn-replicas /lhcb/certification/test/ALLSTREAMS.DST/00000239/
↳0000/00000239_00000074_1.allstreams.dst
{'Failed': {},
 'Successful': {'/lhcb/certification/test/ALLSTREAMS.DST/00000239/0000/00000239_
↳00000074_1.allstreams.dst': {'CERN_MC_M-DST': 'srm://srm-lhcb.cern.ch/castor/cern.
↳ch/grid/lhcb/certification/test/ALLSTREAMS.DST/00000239/0000/00000239_00000074_1.
↳allstreams.dst'}}}
[lxplus433] x86_64-slc5-gcc43-opt /afs/cern.ch/lhcb/software/DEV/LHCBDIRAC/LHCBDIRAC_
↳v6r0-pre12> dirac-dms-lfn-replicas /lhcb/certification/test/ALLSTREAMS.DST/00000239/
↳0000/00000239_00000076_1.allstreams.dst
{'Failed': {},
 'Successful': {'/lhcb/certification/test/ALLSTREAMS.DST/00000239/0000/00000239_
↳00000076_1.allstreams.dst': {'CNAF_MC_M-DST': 'srm://storm-fe-lhcb.cr.cnaf.infn.it/
↳tld1/lhcb/certification/test/ALLSTREAMS.DST/00000239/0000/00000239_00000076_1.
↳allstreams.dst'}}}

```

How to enable/disable FTS channel ? To check TFS transfer, look at the log for DataManagement/FTSSubmitAgent

3.3.3 Specific tests

Every release is somewhat special, and introduce new features that should be tested. It has to be noted that developers should always participate in the testing of very specific new developments, anyway the certification manager should look into if these tests have been done.

Within Jira, there is a special board, named [ready for integration](#). that contain tasks marked as “Resolved”, but not yet “Done”. Dragging tasks from left to right will mark them as “Done”.

So, the certification manager can decide to investigate directly, by submitting tests, if know, or ask the developer to confirm the task can be closed.

CHAPTER 4

Indices and tables

- `genindex`
- `search`