# Table of Contents

**Running the online alignment in run 2** ........................................................................................................... 1

- How to run the Alignment from the PVSS panel ............................................................................................. 1
- Running without the autopilot .......................................................................................................................... 3
- See and select runs available on the HLT farm to run the alignment ................................................................. 4
- See which runs are used for the alignment ....................................................................................................... 5
- Software development in the satellite area (online) ......................................................................................... 5
- Compile the alignment framework online ....................................................................................................... 5
- Function to be used ........................................................................................................................................... 5
- Alignment constants in.db and ONLINE.db ...................................................................................................... 6

**Check and change the CONDDB and DDDB tags used by the Alignment** .................................................. 6

- Check the CONDDB and DDDB tags currently used by Alignment ................................................................. 7

**Monitoring the automatic procedure** .......................................................................................................... 11

- XML files location ........................................................................................................................................... 11
- Histograms location .......................................................................................................................................... 12
- Location of Automatically produced plots ...................................................................................................... 12
- Make the Automatically produced plots yourself .......................................................................................... 13
- Make trend plots .............................................................................................................................................. 13

**Update reference histograms in the presenter** ............................................................................................... 13

**Known issues and workaronds** ..................................................................................................................... 14

- Create a new release area ................................................................................................................................ 14
- Troubleshooting................................................................................................................................................. 16

**See the logfiles** ............................................................................................................................................ 16

**Warnings and errors to shifters** .................................................................................................................. 16

- Where in the code ............................................................................................................................................ 16
- How to see the dim service .............................................................................................................................. 17

**Use the debugger** ......................................................................................................................................... 17

**Avoid to insert the online password many times** ......................................................................................... 17
Running the online alignment in run 2

How to run the Alignment from the PVSS panel

Open the panel from an ui or plus machine:

```bash
ssh -Y lbgw
ssh -Y ui
/group/online/ecs/Shortcuts315/LHCb/ECS/ECS_UI_FSM.sh
```

Open the LHCb_Align tree (right click).

The first thing to do is to write your name in the appropriate field and click on Reserve Alignment so that people know who to contact in case of need.

⚠️ Usually the farm should be taken by the run control in shared mode. If this is not the case you may have to:

1. take the project: click on the grey open lock and choice "take";
2. Allocate: click on the State menu and select Allocate.

Then select the activity from the menu on the right of the Run Info panel. Normally, alignment activities are performed in this order:

1. VELO
2. Tracker
3. RICH
4. Muon
Now you should select the run range to use in the alignment.

To select the run range, click on the *Choose Runs for Alignment* button. Select the runs from the list and click *Ok*. The runs corresponding to the last fill are in the bottom of the table. It is possible to select runs from any other type of data for a given alignment activity (by default it proposes the data corresponding to the activity).

Now open the error log to monitor what is going on. On a shell, type:

```
errorLog LHCbA
```

Useful commands: Ctrl-S to block the screen, Ctrl-Q to resume, Ctrl-C to close.

Tips:

- this is the error log for the whole LHCb-A online partition, so if errors not containing "Alig" appear, do not panic.
- the color code is determined according to the content of the message: for example, messages containing *contribution to hit error* do not represent errors, but are still colored in red.
You are now ready to run the alignment: **turn on the Auto Pilot** and wait that it does the **Configure** and **Start run**. This will take several minutes; when it will be done everything should be in **ready** and the autopilot should be off.

(to check- and modify in case- the function and the conditions that are used see sections "Function to be used" and "Alignment constants")

**Running without the autopilot**

From the LHCb_Align PVSS panel, click on the menu at the top and choose **Configure**. This can take a few minutes.

If the HLT subtree goes in **Error** state, you can open it and exclude the nodes or farms in error using the arrows in the **Quick actions** panel. To locate the troubling nodes, have a look at the **PARTAlign** tree.
Once the top tree is in state **ready**, you can click on **Start run**. Enjoy!

**See and select runs available on the HLT farm to run the alignment**

There is a button on the main panel "Run for alignment" Clicking on it a new window is open. It shows for the selected activity fill number, run number, number of events collected for that activity in that run (This number is not precise to 1 event but the order is correct, and it should be read as at least. This list is available only for all the fill starting from 3962)

See and select runs available on the HLT farm to run the alignment
See which runs are used for the alignment

From the top panel click on RunInfo, then in the panel that appears click on Trigger. In the panel that appears the information can be found under Trigger.DeferredRuns.

Software development in the satellite area (online)

AlignmentOnline is installed in the satellite area and a specific branch is devoted to development in there. The suggested workflow is the following:

```
<make changes until sensible workpoint>
git add <files>
git commit -m '<a meaningful message>'
<make changes until sensible workpoint>
git add <files>
git commit -m '<a meaningful message>'
<ready to go to the main repository without breaking the nightlies>
git pull origin master
<fix any conflict (hopefully none) and check that things work as expected>
git push origin satellite
<open merge request from given link or gitlab webpage>
```

If some modification in Alignment is needed, the corresponding package can be developed with the workflow described in Git4LHCb:

```
git lb-use Alignment
git lb-checkout Alignment/master Some/Package
<make changes and test>
git add Some/Package/src/MyStuff.cpp
git commit -m 'fixing feature abc (JIRATICKET-123)?
git lb-push Alignment JIRATICKET-123
```

Don't forget to open a merge request in Alignment after pushing.

Compile the alignment framework online

The code used online for the alignment can be found in:

```
/group/online/dataflow/cmtuser/AlignmentRelease/ (Link to the appropriate AlignmentOnlineDev_vXYrZ, e.g. AlignmentOnlineDev_v11r0 for 2016 or AlignmentOnlineDev_v11r4p1 in 2017). To recompile the code after having made some changes, login to plus as online (ui does not work) and follow the prescription and troubleshooting help here:
```

You are now ready to compile (as online)

Function to be used

First, follow the instructions for the installation explained at the step before. The functions that can be used to run the tracker alignment are defined under:

```
/group/online/dataflow/cmtuser/AlignmentRelease/Alignment/TAlignment/python/TAlignment/AlignmentScenarios.py
```

E.g. configureEarlyDataAlignment (for early 2012 alignment)

configureTrackerAlignment (default 2016/2017 alignment)

configureFirstTrackerAlignment (early runs 2016 alignment)
configureTrackerAlignmentITInternal (additional dof for internal layers of the IT)

The different elements that can be aligned are defined in the script `Alignables.py` that is present in the same directory. Here, \( T_x T_y T_z = \) Translation around the pivot point in X, Y and Z axis (positive or negative), \( R_x R_y R_z = \) Rotation around X, Y and Z axis.

The function that one want to run can be found in

```
/group/online/dataflow/cmtuser/AlignmentRelease/AlignmentOnline/AlignOnline/python/AlignmentConfigurations/TrackerAlignment.py
```

Just uncomment the related lines.

After modifying any of these scripts, do again `do_install`.

### Alignment constants in .db and ONLINE.db

In addition to the `[det]Global` and `[det]Modules` xml files, the IT and TT also support an `Elements.xml` that contains values taken as constant over time, and thus are not rewritten by the automatic procedure. These parameters were used to be stored in `LHCbCond`.

Since run 1, the database logic has changed. The alignment constants are no longer stored in `LHCbCond` but in the Online database that does not have tags and only supports intervals of validity. On top of this, a database named `CalibOff.db` may be used to hold different tags: its behavior is the same of `LHCbCond`, but the structure is that of the `Online` database.

The `/group/online/alignment` folder holds the latest alignment results used by the HLT.

For the tracker, the important info are contained into

```
/group/online/alignment/IT/ITModules
/group/online/alignment/IT/ITGlobal
```

and similarly for TT and OT. If non stated explicitly, the latest version is used. To select a different version it is useful to look into this twiki page, where a (non-up-to-date) list can be found. Important: we do not align for ITModules thus we never update these xml.

To change the xml used, use the script `newAlignmentStart.py` like

```
python newAlignmentStart.py --IT versionModules versionGlobal
```

and similarly for IT and OT. With the option --dry-run, the script only print the version used without changing it.

### Check and change the CONDDB and DDDDDB tags used by the Alignment

The CONDDB (LHCBCOND) and DDDDDB tags used by the LHCb_Align project are currently (April-May 2016) not automatically propagated from the tags used in the HLT. Therefore when there is a change in the tags used by the HLT, we need to update the tags used by the online Alignment as well.
Check the CONDDB and DDDB tags currently used by Alignment

On the LHCb_Align panel click on the "View.." button next to the Trigger Config

The window that you now visualize, shows you the database tags currently used, but you cannot change them from this panel (see all the fields are grey and no "Save" button at the bottom of the window). To change the tags, see next point.
Change the CONDDB and DDDB tags to be used by the Alignment

On the LHCb_Align panel click on "RunInfo"
and in the "RunInfo" panel, select Trigger Configuration, Create/Edit

Change the CONDDB and DDDB tags to be used by the Alignment
From this panel you can select the cond db and dddb tags needed, and Save the update with the button at the bottom of the window. (you will be asked to confirm your choice before the change is saved).

⚠️ **NOTE:** It is possible that you also have to change the version on Moore to the latest version.
\textbf{NOTE:} Once you have saved the new tags, for the given trigger configuration (PassThrough in this case), you need to select again the Trigger configuration (PassThrough in this case) in the LHCb Align TOP level panel.

\textbf{NOTE:} You can only choose the database tags which appear on the menu when you click on the arrow next to "Tags CONDDB (DDDB)". This is because you can only choose the tags which are already available for the HLT. (The HLT in order to make the tags available on the panels needs to put a snapshot into a specific location). For the Alignment we will always have to select a tag which is already available for the HLT, because we have to be consistent in the tags we use, we will never need a tag not already available for the HLT.

**Monitoring the automatic procedure**

The results of each iteration are saved, this includes xml files and histograms (more details below). For the moment the monitoring should be performed manually running few scripts.

**Xml files location**

- The xml files used in the job while it is running are in /group/online/AlignWork/running
- At the end of each iteration the results are copied in Iter1 (Iter2,...), while the results of last iteration will remain in xml
- All results present in the directory running are copied into a new directory located under /group/online/AlignWork/ACTIVITY/RunXXXX (ACTIVITY=Velo, Tracker, Muon XXX=run number) when the alignment converged or after the maximum number of iteration (currently 10). If you run the job twice on the same run and the same activity the results are overwritten. In this
directory it is possible to find also the logfile of the alignment.

- For each iteration, various subfolders are created, one per aligned subdetector. For each subdetector two sets of alignment results are created: [det]Global and [det]Modules.

- If the job converged and the new alignment constants significantly differ by the others, they are copied in /group/online/alignment/XX (XX: Velo, TT, IT, OT) as vN.xml (N=previous version+1)
- The alignment job for your activity uses the latest version in /group/online/alignment/XX for the subdetectors that you are aligning for.
- In some panel of PVVS one can see (and set) the version that should be used in hlt. Currently it is done manually, in future will be automatic.
- For each run all the information about calibration and alignment version used in hlt can be found in /group/online/hlt/conditions/LHCb/2015/XXXX (XXXX=run number)
  - xx.xml (XX: Velo, TT, IT, OT) contains the link to the version used
  - When a run is ready for hlt2, the alignment/calibration constants has to be copied to be used offline. An hlt task is run and a temporary directory /group/online/hlt/tasks/LHCb/2015/XXXX/OfflineConds is created. It contains the xml with the full list of calibration and alignment constants (with the "offline" header/footer). An additional script copy these xml files to the online db.
  - The constants are available in the online.db for all the runs declared ready for hlt2. Time validity time of the start of the run to the next run with valid constants (if no further run, infinite).

- The location of the Survey info can be found in:
  - /group/online/dataflow/cmtuser/AlignmentRelease/Alignment/SurveyConstraints.py
  - and it is called by AlignmentScenarios.py (in the same directory)

### Histograms location

The histograms are produced automatically when running the alignment. We have one root file for each iteration, they are at: the histograms are at /hist/Savesets/"YEAR"/LHCbA/ Nomenclature convention: ex. 

/ hist/Savesets/2015/LHCbA/AligWrk_Muon/07/10/AligWrk_Muon-15691601-20150710T100630-EOR.root

- Muon is the name of the activity (Muon, Tracker or Velo)
- 156916 is the first run number in the run list
- 01 is the number of iteration
- 20150710T100630 is the time when the file has been generated.

To run root on plus:

/group/online/dataflow/cmtuser/AlignmentRelease/build.x86_64-centos7-gcc62-opt/run bash

or (not working in 2017):

lb-run root root.exe FILENAME.root

### Location of Automatically produced plots

A pdf containing monitoring and convergency plots is produced after the alignment (only a pdf is produced per run analysed). If the alignment needs updating or if the variation of the alignment constants is too big the pdf is uploaded to the logbook. If no alignment update is needed the pdf is not attach to the logbook but only a message appears. All the pdfs can be found here. The naming convention is FIRST_RUN_NUMBER.pdf if the update was not triggered or vVERSION_FIRST_RUN_NUMBER.pdf
Make the Automatically produced plots yourself

To make the automatically produced plots you can use the scripts in Alignment/AlignmentMonitoring/scripts/.

- moniPlots.py to produce the plots for the Velo
- moniPlots_Tracker.py to produce the plots for the Tracker
- moniPlots_Muon.py to produce the plots for the Muon

To see how to use them call them with the \(-h\) option.

E.g. for the Tracker use:

```
/group/online/dataflow/cmtuser/AlignmentRelease/build.x86_64-centos7-gcc62-opt/run bash
/group/online/dataflow/cmtuser/AlignmentDev_v11r4p1/Alignment/AlignmentMonitoring/scripts/moniPlots_Tracker.py
```

Make trend plots

To make the trend plots you can use the scripts Alignment/AlignmentMonitoring/scripts/trendPlots.py call it with the option \(-h\) to see the options available. If you want to produce the "standard plots " for Velo, Tracker and Muon you can use the script Alignment/AlignmentMonitoring/examples/makePublicityPlots.sh, it just calls the previous one with the appropriate options. You may want to open it and change some options like the min and max run number or the date in the label.

```
lb-dev Alignment_v11r4p1
cd AlignmentDev_v11r4p1
git lb-use Alignment
git lb-checkout Alignment/master Alignment/AlignmentMonitoring
make configure
make -j8
./run Alignment/AlignmentMonitoring/examples/makePublicityPlots.sh
```

Update reference histograms in the presenter

Instructions to change the reference histograms in the presenter can be found in this page.

Using the hlt1 conditions instead of the ones in the online.db

If you want to run on plus using the same hlt1 (and hlt2) conditions and not the one in the online.db you should use add the following options (example for a Brunel job). Main reasons: runs not yet processed by hlt2 or check the constants in the online.db are the same ones used in hlt.

```
#Need to check that the CondDBtag and DDDBtag were the same used during the data taking.
Brunel().DDDBtag = "dddb-20150526"
Brunel().CondDBtag = "cond-20150617"
```

from Configurables import CondDB
conddb = CondDB()
conddb.IgnoreHeartBeat = True
conddb.EnableRunChangeHandler = True
Known issues and workaronds

- The folder running should not be present in the /group/online/AligWork directory before the start of the alignment. If it remained there from previous unsuccessful tests it should be removed. To remove it one must log in as online user. Changing the name of the directory also works.

- It is possible that the job become stuck with the PARTAlign_Master in running and the various nodes (e.g. HLTA05_A) in ready. In this case from the top tree select again start run.

- If only few nodes are in the "Included Nodes and Removed Nodes" in the panel Quick Actions of the window LHCbA _HLT: TOP, try to DEALLOCATE and then ALLOCATE again from the Top panel.

- The histograms created before 21st July have a bug for all iterations except the first one (the histograms were not reset before filling in the second (or more) iterations, this results in histograms including entry of first iteration and second iteration (and all the previous for the other iterations). This problem was fixed on 21st July, thus re-running would produce the correct histograms.

- A list of solutions of common problem to be used by HLT piquets can be found in AlignmentPiquetInstruction.

Create a new release area

A new release area can be created with the following steps.

First a clone of the Alignment project is needed:

```
user@plus $ cd /group/online/dataflow/cmtuser
user@plus $ git clone ssh://git@gitlab.cern.ch:7999/lhcb/Alignment.git
user@plus $ mv Alignment AlignmentDev_vXrY
user@plus $ cd AlignmentDev_vXrY
user@plus $ git checkout vXrY
user@plus $ git checkout -b satellite_vXrY
user@plus $ lb-project-init
user@plus $ git update-index --assume-unchanged CMakeLists.txt
user@plus $ chmod g+rw ../AlignmentDev_vXrY # to allow colleagues to make changes in the directory
```

And modify its CMakeLists.txt by changing Alignment to AlignmentDev

```
# Declare project name and version
```

Using the hlt1 conditions instead of the ones in the online.db
Then clone the AlignmentOnline project:

```bash
user@plus $ cd /group/online/dataflow/cmtuser
user@plus $ git clone ssh://git@gitlab.cern.ch:7999/lhcb/AlignmentOnline.git
user@plus $ mv AlignmentOnline AlignmentOnlineDev_vXrY
user@plus $ cd AlignmentOnlineDev_vXrY
user@plus $ git checkout vXrY
user@plus $ git checkout satellite_vXrY
user@plus $ lb-project-init
user@plus $ git update-index --assume-unchanged CMakeLists.txt
user@plus $ chmod g+rw ../AlignmentOnlineDev_vXrY # to allow colleagues to make changes in the directory
```

Then modify the CMakeLists.txt by adding the following lines at its top putting the appropriate version of the Alignment and Online packages:

```bash
set(AlignmentDev_DIR /group/online/dataflow/cmtuser/AlignmentDev_vXrY/InstallArea/$ENV{CMTCONFIG})
set(OnlineDev_DIR /group/online/dataflow/cmtuser/OnlineDev_vXXrYY/InstallArea/$ENV{CMTCONFIG})
```

and change the dependency to the OnlineDev and AlignmentDev release:

```bash
gaudi_project(AlignmentOnlineDev vXrY
  USE AlignmentDev vXrY
  OnlineDev vXXrYY)
```

You are now ready to compile (as online):

```bash
online@plus $ dataflowcmt
online@plus $ cd $User_release_area
online@plus $ export CMTPROJECTPATH=/group/online/dataflow/cmtuser:$CMTPROJECTPATH
online@plus $ cd AlignmentOnlineDev_vXrY
online@plus $ rm -rf build.x* # (carefully)
online@plus $ do_configure
online@plus $ do_action
online@plus $ cmsetup # (one should call this after every installation)
online@plus $ cd AlignmentOnlineDev_vXrY
online@plus $ rm -rf build.x* # (carefully)
online@plus $ do_configure
online@plus $ do_action
online@plus $ cmsetup # (one should call this after every installation)
```

**Beware** that this procedure could fail with CMake failing to find OnlineDev project. This may happen if /group/online/dataflow/cmtuser is not in the CMTPROJECTPATH. To fix this:

```bash
online@plus $ export CMTPROJECTPATH=$CMTPROJECTPATH:/group/online/dataflow/cmtuser
```

Once finished modify the link to the current and old version of AlignmentOnline:

```bash
user@plus $ rm AlignmentRelease_old
user@plus $ mv AlignmentRelease AlignmentRelease_old
user@plus $ ln -s AlignmentOnlineDev_vXrY AlignmentRelease
```
Troubleshooting

A typical error is due to CMake missing boost dependencies. This happens as an error at configuring time. To solve it, simply add the following lines to `searchPath.cmake`:

```cmake
set(CMAKE_PREFIX_PATH "/group/online/dataflow/cmtuser" "/group/online/dataflow/SwData" "/cvmfs/lhcb.cern.ch/lib/lcg/app/releases" "/cvmfs/lhcb.cern.ch/lib/lcg/external" ${CMAKE_PREFIX_PATH})
list(INSERT CMAKE_PREFIX_PATH 0 "$ENV{User_release_area}")
```

It may sometimes happen that the above recipe won't work. In that case manual compilation of the opt and dbg executables is possible:

```bash
online@plus $ make purge
online@plus $ export OnlineDev_DIR=/group/online/dataflow/cmtuser/OnlineDev_v5r24/InstallArea/x86_64-slc6-gcc48-opt # <-- this should point to the online version in CMakeLists.txt
online@plus $ export LD_LIBRARY_PATH=$LD_LIBRARY_PATH:${SW_LCG}/releases/LCG_79/Boost/1.55.0_python2.7/x86_64-slc6-gcc48-opt/lib # <-- this solves the boost compilation error
online@plus $ make -j 5 configure
online@plus $ make -j 5 install
```

n.b.: if `SW_LCG` is not defined, look for lcg in the configuration output (usually it is
/cvmfs/lhcb.cern.ch/lib/lcg)

The same procedure needs to be repeated for the dbg executables by repeating it after logging in with

```bash
online@plus $ LbLogin -c $CMTDEB
```

beware to setup properly the environment variables to point to the dbg install area.

Finally, call

```bash
online@plus $ cmsetup
```

To setup the system.

See the logfiles

The logfile of the day is `/clusterlogs/partitions/LHCbA/daq/LHCbA.log`. To see all the messages for a particular node do something like: grep hlt02 /clusterlogs/partitions/LHCbA/daq/LHCbA.log Old logs are zipped and moved to the folder `old`; there you can do something like less /clusterlogs/partitions/LHCbA/daq/old/LHCbA.log.2016-05-11:03h.bz2 | grep hlt02 | less

Warnings and errors to shifters

In `AlignmentOnline/AlignOnline/src/AlignOnlineIterator.cpp` it's define a DIM service to communicate the outcome of the alignment procedure and raise warning and alarms.

Where in the code

The DIM service publish the content of the string `m_align_message` that has the following structure:

"SeverityLevel: Subdetector message | further informations" where `SeverityLevel` can be INFO, WARNING or ERROR and the part = | further informations= is optional. In case of WARNING or ERROR an email will be sent to lhcb-onlinealignmentcalibration@cernNOSPAMPLEASE.ch; in case of ERROR a message will also appear in the alarm screen and require immediate action. In the alarm message only the Subdetector message will appear while the email will contain also the further informations. The error message will disappear at the next run of the alignment on the same subdetector when an INFO message is found.
How to see the dim service

From a machine of in the online network do:

did -dns=hlt01

A window will appear, there in the View menu select Servers by Node and select hlt02. If you are running the alignment, among the services you should see LHCbA_HLT02_AligDrv_0, click on it and select Services. From the list select /Publisher/AlignmentMessenger and click on View/Send. A new window will open where you can see the string written from AlignOnlineIterator.cpp. Click on Subscribe (On Change).

Use the debugger

as online user ssh into hlt02

ssh -Y online@hlt02

Follow the instructions above for running the alignment online until the job is configured. Just after having started the analysis

ps x | grep AligDrv

to find the pid of the iterator. Then

/home/beat/scripts/debug --pid [pid]

When the debugger finished configured press c to make the process continue. In case of crash one can have the information on what caused the crash by typing where.

Avoid to insert the online password many times

To avoid inserting the online password many times it is possible to create a private/public key pair following the instructions in http://cvs.web.cern.ch/cvs/howto.php#accessing-sshlinux.

Just once one has to do:

ssh-keygen --help
cp .ssh/id_rsa.pub .ssh/authorized_keys
ssh-agent > ~/.agentinfo
. ~/.agentinfo
ssh-add

and then each time:

eval `ssh-agent`
ssh-add

-- ElenaGraverini - 2015-06-12
Avoid to insert the online password many times