Table of Contents

Rerunning a production job locally starting from a LogSE link

The original request for this page came from the following log link which will be cleaned at some point soon after the creation of this twiki page. The log page is visible here for completeness:

Log files for Job 00006556_00000117

Environment_Dump_DaVinci_v25r4p3_Step2.log
DaVinci_00006556_00000117_2.log
Environment_Dump_ErrorLogging_Step2.log
Environment_Dump_ErrorLogging_Step1.log
Error_Log_DaVinci_v25r4p3_2.log
Brunel_00006556_00000117_1.log
Environment_Dump_Brunel_v37r2p2_Step1.log
Error_Log_Brunel_v37r2p2_1.log
std.out
authentication.xml
pool_xml_catalog.xml
jobDescription.xml
dblookup.xml
bookkeeping_00006556_00000117_1.xml
DaVinci_v25r4p3_Run_2.sh
Brunel_v37r2p2_Run_1.sh
job.info

Job 00006556_00000117 corresponds to WMS JobID 8844527 executed at LCG.IN2P3.fr.

Parameter summary for job 00000117_00006556

Parameter Name	Parameter Value
InputData	/lhcb/data/2010/RAW/FULL/LHCb/COLLISION10/72328 /072328_0000000182.raw
JobType	DataReconstruction
LogFilePath	/lhcb/data/2010/LOG/00006556/0000/00000117
LogLevel	verbose
MaxCPUTime	600000
ProductionOutputData	/lhcb/data/2010/SDST/00006556/0000/00006556_00000117_1.sdst; /lhcb/data/2010/HIST/00006556 /0000/Brunel_00006556_00000117_1_Hist.root;/lhcb/data/2010/DST /00006556/0000/00006556_00000117_2.dst;/lhcb/data/2010/HIST /00006556/0000/DaVinci_00006556_00000117_2_Hist.root
SoftwarePackages	AppConfig.v3r58;Brunel.v37r2p2;AppConfig.v3r58;DaVinci.v25r4p3
SystemConfig	slc4_ia32_gcc34

From the log link above you see numbers padded with zeroes like:

```
<PRODUCTION_ID>_< PRODUCTION_JOB_ID>
```

meaning that in the case above the production ID is 6556 and the production job ID is 117. The WMS job ID that can be looked at in the job monitoring is also listed (as well as the site it was executed on). This corresponds to a production with the following details (available on the production monitoring page when pressing "Show Details" in the menu of the production ID.

```
====> Brunel v37r2p2 Step0
  Brunel Option Files:
    $APPCONFIGOPTS/Brunel/earlyData.py
    $APPCONFIGOPTS/Brunel/DataType-2010.py
    $APPCONFIGOPTS/UseOracle.py
    $APPCONFIGOPTS/DisableLFC.py
  ExtraPackages: AppConfig.v3r58
====> DaVinci v25r4p3 Step1
```

```
  DaVinci Option Files:
    $APPCONFIGOPTS/DaVinci/DVMonitor-RealData.py
    $APPCONFIGROOT/options/DaVinci/DVStrippingDST-RealData-noV0.py
    $APPCONFIGOPTS/DaVinci/StrippingLinesPrescales-1005.py
    $APPCONFIGROOT/options/DaVinci/DataType-2010.py
    $APPCONFIGROOT/options/DaVinci/InputType-SDST.py
    $APPCONFIGOPTS/UseOracle.py
    $APPCONFIGOPTS/DisableLFC.py
  ExtraPackages: AppConfig.v3r58

BK Input Data Query:
    ConfigName = LHCb
    EventType = 90000000
    FileType = RAW
    ProcessingPass = Real Data
    DataQualityFlag = EXPRESS_OK
    ConfigVersion = Collision10
    DataTakingConditions = Beam3500GeV-VeloClosed-MagDown
```

The same information is also available elsewhere but allows further digging in the DIRAC web portal.

Some useful conventions to be aware of about the log index page include:

- Files with extension log are always standard output from part of a step
- By convention the log file name per Gaudi application process is constructed from the application name, production ID and production job ID
- Application log files always have the command listed as the first line e.g. DaVinci_00006556_00000117_2.log followed by printouts of the LD_LIBRARY_PATH, PATH and PYTHONPATH e.g.

```
================================================
Log file from execution of: gaudirun.py    /afs/in2p3.fr/grid/toolkit/lhcb/lib/lhcb/DBASE/AppConf
================================================
LD_LIBRARY_PATH is:
/afs/in2p3.fr/grid/toolkit/lhcb/lib/lhcb/DAVINCI/DAVINCI_v25r4p3/InstallArea/slc4_ia32_gcc34/lib
/afs/in2p3.fr/grid/toolkit/lhcb/lib/lhcb/HLT/HLT_v10r3/InstallArea/slc4_ia32_gcc34/lib
/afs/in2p3.fr/grid/toolkit/lhcb/lib/lhcb/ANALYSIS/ANALYSIS_v4r4/InstallArea/slc4_ia32_gcc34/lib
/afs/in2p3.fr/grid/toolkit/lhcb/lib/lhcb/REC/REC_v9r2p1/InstallArea/slc4_ia32_gcc34/lib
/afs/in2p3.fr/grid/toolkit/lhcb/lib/lhcb/LBCOM/LBCOM_v9r2p1/InstallArea/slc4_ia32_gcc34/lib
/afs/in2p3.fr/grid/toolkit/lhcb/lib/lhcb/PHYS/PHYS_v9r5/InstallArea/slc4_ia32_gcc34/lib
/afs/in2p3.fr/grid/toolkit/lhcb/lib/lhcb/LHCB/LHCB_v31r0p1/InstallArea/slc4_ia32_gcc34/lib
/afs/in2p3.fr/grid/toolkit/lhcb/lib/lhcb/GAUDI/GAUDI_v21r9/InstallArea/slc4_ia32_gcc34/lib
/afs/in2p3.fr/grid/toolkit/lhcb/lib/lcg/external/Grid/myproxy/3.6-VDT-1.6.0/slc4_ia32_gcc34/globu
/afs/in2p3.fr/grid/toolkit/lhcb/lib/lcg/external/qt/4.4.2/slc4_ia32_gcc34/lib
/afs/in2p3.fr/grid/toolkit/lhcb/lib/lhcb/COMPAT/COMPAT_v1r5/CompatSys/slc4_ia32_gcc34/lib
/afs/in2p3.fr/grid/toolkit/lhcb/lib/lcg/external/dcache_client/1.9.3p1/slc4_ia32_gcc34/dcap/lib
...
================================================
PYTHONPATH is:
/afs/in2p3.fr/grid/toolkit/lhcb/lib/lhcb/DAVINCI/DAVINCI_v25r4p3/InstallArea/python.zip
/afs/in2p3.fr/grid/toolkit/lhcb/lib/lhcb/HLT/HLT_v10r3/InstallArea/python.zip
/afs/in2p3.fr/grid/toolkit/lhcb/lib/lhcb/ANALYSIS/ANALYSIS_v4r4/InstallArea/python.zip
/afs/in2p3.fr/grid/toolkit/lhcb/lib/lhcb/REC/REC_v9r2p1/InstallArea/python.zip
/afs/in2p3.fr/grid/toolkit/lhcb/lib/lhcb/LBCOM/LBCOM_v9r2p1/InstallArea/python.zip
/afs/in2p3.fr/grid/toolkit/lhcb/lib/lhcb/PHYS/PHYS_v9r5/InstallArea/python.zip
...
================================================
PATH is:
/afs/in2p3.fr/grid/toolkit/lhcb/lib/lhcb/HLT/HLT_v10r3/InstallArea/scripts
/afs/in2p3.fr/grid/toolkit/lhcb/lib/lhcb/LHCB/LHCB_v31r0p1/InstallArea/scripts
/afs/in2p3.fr/grid/toolkit/lhcb/lib/lhcb/GAUDI/GAUDI_v21r9/InstallArea/scripts
/afs/in2p3.fr/grid/toolkit/lhcb/lib/lhcb/ANALYSIS/ANALYSIS_v4r4/InstallArea/slc4_ia32_gcc34/bin
/afs/in2p3.fr/grid/toolkit/lhcb/lib/lhcb/PHYS/PHYS_v9r5/InstallArea/slc4_ia32_gcc34/bin
/afs/in2p3.fr/grid/toolkit/lhcb/lib/lhcb/LHCB/LHCB_v31r0p1/InstallArea/slc4_ia32_gcc34/bin
```

```
/afs/in2p3.fr/grid/toolkit/lhcb/lib/lhcb/GAUDI/GAUDI_v21r9/InstallArea/slc4_ia32_gcc34/bin
...
================================================
...
```

- Files starting with Environment are dumps of the environment for each step
- Files starting with ErrorLog are the output of the core software logErr script'
- Files starting with bookkeeping are the BK records of each step
- The pool_xml_catalog.xml file is the POOL XML slice for the job and looks like:

```
<?xml version="1.0" encoding="UTF-8" standalone="no" ?>
<!-- Edited By PoolXMLCatalog.py -->
<!DOCTYPE POOLFILECATALOG SYSTEM "InMemory">
<POOLFILECATALOG>


  <File ID="cc332e7c-67c6-11df-bfa2-00188b8565ca">
     <physical>
       <pfn filetype="MDF" name="/scratch/lhcb04912088.ccwl9100/tmp/https_3a_2f_2fwms216.cern.ch_
     </physical>
     <logical>
       <lfn name="/lhcb/data/2010/RAW/FULL/LHCb/COLLISION10/72328/072328_0000000182.raw"/>
     </logical>
   </File>

  <File ID="06201B31-856D-DF11-A36E-001D0967E002">
    <physical>
      <pfn filetype="ROOT_All" name="00006556_00000117_1.sdst"/>
    </physical>
    <logical/>
  </File>

</POOLFILECATALOG>
```

- The pool_xml_catalog.xml file is accessed via the usual Gaudi options below but notice that this file is site specific!

```
FileCatalog().Catalogs= ["xmlcatalog_file:pool_xml_catalog.xml"]
```

*The POOL XML slice can be regenerated if desired using genXMLCatalog e.g. this avoids having to supply a new EventSelector.Input... option.

```
$ genXMLCatalog --help
Usage: genXMLCatalog <options> <config-files>
Options:
  -s, --site <site>: site name (default=CERN)
  -d, --depth <depth>: depths for ancestors in BK (default=1)
  -f, -p, --catalog <catalog-name>: XML file catalog name (default=./pool_xml_catalog.xml)
  -n, --newoptions <config-file>: generate a new config file (no catalog is created)
  -o, --options <config-file>: python config file to be parsed (for backward compatibility)
  -i, --ignore: ignore missing files
  <config-files>: list of python config files
  -v: verbose output
```

- Files ending in sh allow to rerun a step from the directory on the worker node that the job was initially running and can serve as a guide (removing the site specific paths for example) e.g. looking at DaVinci_v25r4p3_Run_2.sh and omitting the environment dump

```
#!/bin/sh
#################################################################
# Dynamically generated script to reproduce execution environment.
#################################################################
# $Id: LogSERunJobs.txt,v 1.2 2010/06/11 15:21:45 Peter_2eL_2eJones_40cern_2ech Exp $
```

```
################################################################
export CALORECOOPTS="/afs/in2p3.fr/grid/toolkit/lhcb/lib/lhcb/REC/REC_v9r2p1/Calo/CaloReco/option
export LOKIMCROOT="/afs/in2p3.fr/grid/toolkit/lhcb/lib/lhcb/LHCB/LHCB_v31r0p1/Phys/LoKiMC"
export COMMONPARTICLESROOT="/afs/in2p3.fr/grid/toolkit/lhcb/lib/lhcb/ANALYSIS/ANALYSIS_v4r4/Phys/
export JOBID="8844527"
...
export HEPMCROOT="/afs/in2p3.fr/grid/toolkit/lhcb/lib/lcg/external/LCGCMT/LCGCMT_58c/LCG_Interfac
export LCGCMTVERS="58c"
echo "================================================="
echo "Log file from execution of: gaudirun.py    $APPCONFIGOPTS/DaVinci/DVMonitor-RealData.py  $A
echo "================================================="
echo "LD_LIBRARY_PATH is:"
echo $LD_LIBRARY_PATH | tr ":" "
"
echo "================================================="
echo "PYTHONPATH is:"
echo $PYTHONPATH | tr ":" "
"
echo "================================================="
echo "PATH is:"
echo $PATH | tr ":" "
"
echo "================================================="
env | sort >> Environment_Dump_DaVinci_v25r4p3_Step2.log
gaudirun.py    $APPCONFIGOPTS/DaVinci/DVMonitor-RealData.py  $APPCONFIGROOT/options/DaVinci/DVStr
declare -x appstatus=$?
if [ -e core.* ]
 then  gdb python core.* >> DaVinci_Step2_coredump.log << EOF
where
quit
EOF
fi
exit $appstatus
```

- The most useful file for obtaining the options used to run the job is std.out which will be described further below.

Broadly speaking the *.sh scripts show you how to run the gaudirun.py command but do not show you the precise LbLogin and SetupProject calls as well as the Gaudi options generated by DIRAC for the step.

In this example, looking at the std.out the environment was set via:

```
2010-06-02 01:36:13 UTC dirac-jobexec.py/ProductionEnvironment  INFO: Attempting to run: /scratch
2010-06-02 01:36:14 UTC dirac-jobexec.py/ProductionEnvironment  INFO: Attempting to run: /scratch
2010-06-02 01:36:29 UTC dirac-jobexec.py/ProductionEnvironment  INFO: LbLogin.sh, SetupProject.sh
2010-06-02 01:36:29 UTC dirac-jobexec.py/ProductionEnvironment  VERB: APPCONFIGROOT found, will o
2010-06-02 01:36:29 UTC dirac-jobexec.py/CondDBAccess  VERB: Running at site: LCG.IN2P3.fr, CondD
2010-06-02 01:36:29 UTC dirac-jobexec.py/ProductionEnvironment  VERB: Successfully obtained Oracl
2010-06-02 01:36:29 UTC dirac-jobexec.py/ProductionEnvironment  VERB: Removing CMTPROJECTPATH fro
2010-06-02 01:36:29 UTC dirac-jobexec.py/ProductionEnvironment  INFO: Setting MYSITEROOT to /scra
2010-06-02 01:36:29 UTC dirac-jobexec.py/ProductionEnvironment  INFO: Setting CMTCONFIG to slc4_i
Command = gaudirun.py    $APPCONFIGOPTS/DaVinci/DVMonitor-RealData.py  $APPCONFIGROOT/options/DaV
2010-06-02 01:36:29 UTC dirac-jobexec.py/GaudiApplication  VERB: Created debug script DaVinci_v25
2010-06-02 01:36:29 UTC dirac-jobexec.py/GaudiApplication  INFO: Running DaVinci v25r4p3 step 2
2010-06-02 01:36:29 UTC dirac-jobexec.py/GaudiApplication  VERB: setJobApplicationStatus(8844527,
2010-06-02 01:36:29 UTC dirac-jobexec.py  WARN: Server is not who it's supposed to be Connecting
2010-06-02 05:53:55 UTC dirac-jobexec.py/GaudiApplication  INFO: Status after the application exe
2010-06-02 05:53:55 UTC dirac-jobexec.py/GaudiApplication ERROR: DaVinci execution completed with
```

You can also see exactly the options that were used for the step, the application name and version are always printed like the below:

```
2010-06-02 01:36:13 UTC dirac-jobexec.py/GaudiApplication  INFO: Extra options generated for DaVi
```

```
#//////////////////////////////////////////////////////
# Dynamically generated options in a production or analysis job

from Gaudi.Configuration import *
HistogramPersistencySvc().OutputFile = "DaVinci_00006556_00000117_2_Hist.root"
from DaVinci.Configuration import *
DaVinci().EvtMax=-1
DaVinci().HistogramFile = "DaVinci_00006556_00000117_2_Hist.root"
from Configurables import SelDSTWriter
SelDSTWriter.OutputFileSuffix = '00006556_00000117_2'
LHCbApp().DDDBtag = "head-20100407"
LHCbApp().CondDBtag = "head-20100509"
ApplicationMgr().EvtMax = -1
def forceOptions():
  MessageSvc().Format = "%u % F%18W%S%7W%R%T %0W%M"
  MessageSvc().timeFormat = "%Y-%m-%d %H:%M:%S UTC"
appendPostConfigAction(forceOptions)
EventSelector().Input=[ "DATAFILE='LFN:00006556_00000117_1.sdst' TYP='POOL_ROOTTREE' OPT='READ'"]

FileCatalog().Catalogs= ["xmlcatalog_file:pool_xml_catalog.xml"]

ApplicationMgr().EvtMax = -1
```

For rerunning a job the above options (using POOL XML catalog or not) must be adapted to run at your local site. Any intermediate job outputs as in the case above are uploaded to the DEBUG SE based at the following path in Castor:

```
/castor/cern.ch/grid/lhcb/debug/ < BK Config Version e.g. Collision10> / <File Type e.g. SDST> /

$ nsls -l /castor/cern.ch/grid/lhcb/debug/Collision10/SDST/00006556/0000/00006556_00000117_1.sdst
-rw-r--r--   1 lhcbprod z5               1006322858 Jun 02 07:56 /castor/cern.ch/grid/lhcb/debug/
```

the path is also printed in the standard output:

```
2010-06-02 05:54:26 UTC dirac-jobexec.py/AnalyseLogFile  INFO: Attempting: rm.putAndRegister("000
2010-06-02 05:54:27 UTC dirac-jobexec.py  VERB: LcgFileCatalogClient.__getACLInformation: /lhcb/d
2010-06-02 05:54:27 UTC dirac-jobexec.py  INFO: ReplicaManager.putAndRegister: Checksum informati
2010-06-02 05:54:38 UTC dirac-jobexec.py  INFO: ReplicaManager.putAndRegister: Checksum calculate
2010-06-02 05:54:39 UTC dirac-jobexec.py  VERB: StorageElement.isValid: Determining whether the S
2010-06-02 05:54:39 UTC dirac-jobexec.py  WARN: StorageElement.isValid: The 'operation' argument
2010-06-02 05:54:39 UTC dirac-jobexec.py  VERB: StorageElement.getStorageElementName: The Storage
2010-06-02 05:54:39 UTC dirac-jobexec.py  VERB: StorageElement.__executeFunction: Attempting to p
2010-06-02 05:54:39 UTC dirac-jobexec.py  VERB: StorageElement.isValid: Determining whether the S
2010-06-02 05:54:39 UTC dirac-jobexec.py  VERB: StorageElement.isLocalSE: Determining whether CER
2010-06-02 05:54:39 UTC dirac-jobexec.py  VERB: StorageElement.__executeFunction: Generating 1 pr
2010-06-02 05:54:39 UTC dirac-jobexec.py  VERB: StorageElement.__executeFunction: Attempting to p
2010-06-02 05:54:41 UTC dirac-jobexec.py  INFO: SRM2Storage.__putFile: Executing transfer of file
2010-06-02 05:56:18 UTC dirac-jobexec.py  INFO: SRM2Storage.__putFile: Successfully put file to s
2010-06-02 05:56:19 UTC dirac-jobexec.py  VERB: ReplicaManager.registerFile: Attempting to regist
2010-06-02 05:56:19 UTC dirac-jobexec.py  VERB: StorageElement.isValid: Determining whether the S
2010-06-02 05:56:19 UTC dirac-jobexec.py  WARN: StorageElement.isValid: The 'operation' argument
2010-06-02 05:56:19 UTC dirac-jobexec.py  VERB: StorageElement.getStorageElementName: The Storage
2010-06-02 05:56:19 UTC dirac-jobexec.py  VERB: StorageElement.getProtocols: Obtaining all protoc
2010-06-02 05:56:19 UTC dirac-jobexec.py ERROR: StorageElement.getPfnForProtocol: Requested proto
2010-06-02 05:56:19 UTC dirac-jobexec.py  VERB: ReplicaManager.__registerFile: Resolved 1 files f
2010-06-02 05:56:21 UTC dirac-jobexec.py  INFO: ReplicaManger.putAndRegister: Sending accounting
2010-06-02 05:56:21 UTC dirac-jobexec.py/AnalyseLogFile  INFO: {'OK': True, 'Value': {'Successful
```

as usual with Gaudi applications care must be taken to set CondDB, DDDB and other application step specific options (about which I'm sure the experts know more than us).

To summarise the recipe would be to look for the appropriate commands for:

- LbLogin
- SetupProject
- Options passed to gaudirun.py (not including the automatically generated gaudi_extra_options.py described above)

and adapt the specific step options to work with the data uploaded to the DEBUG SE. Since the CondDB may be used it also means lhcb-proxy-init should be executed (implying the !--use-grid should be set in the SetupProject call).

This topic: LHCb > LogSERunJobs
Topic revision: r2 - 2010-06-11 - PeterJones