

Table of Contents

Getting Productions to 100%.....	1
Analysing a reconstruction request.....	1
Looking why the Reconstruction production 8178 still has 2 files in status "Assigned".....	2
Looking why Merging production 8187 still had 30 files in status "Assigned".....	3
Looking why there is a quite important mismatch in the number of files merged or to be merged and those produced by the production.....	3
A couple of useful queries.....	4
Productions with some Unused files, which have run number = zero in production DB.....	7
Productions with some 'Unused' files as they only have a failover replica in the LFC but no 'regular' replica.....	7
Productions with some jobs in 'Rescheduled' status since many days.....	8
Production with some Assigned files, though the Dirac job is reported as Done.....	9
Production with some Assigned files and corresponding Tasks have external status Failed.....	9
Stripping productions with some Unused files after several days.....	10
Corrupted or lost input files.....	11
File in status Assigned and relative job is Done: input data missing from JDL.....	11
Some comments and warnings.....	11
dirac-transformation-cli only update some status.....	12
No tool available to set the files status to Processed.....	12
No tool to retrieve all the failed jobs for a given LFN.....	12

Getting Productions to 100%

Assuming there is no external problems, when a production starts, and Data Quality shifters follow it, It is usually easy to reach a 95% of processed files without much intervention. Anyway, there are still cases where a human intervention is required. Throughout this page I'll give some examples.

WARNING WARNING WARNING WARNING WARNING WARNING WARNING WARNING WARNING WARNING

- Applying what is written here is potentially dangerous and should be made only by experts.

WARNING WARNING WARNING WARNING WARNING WARNING WARNING WARNING WARNING WARNING

Analysing a reconstruction request

The "workflow" of a reconstruction request could be summarised as follows:

1. Before a RAW file is even reconstructed, a Data Quality flag ("OK") has to be made by the DQ shifter.
2. There is a "DataReconstruction" production that take RAW files as input and produces stream DST.
3. The outputs of this first step have to get a Data Quality check. This flag is registered in the bookkeeping.
4. In case the DQ flag is "OK", the streams are merged. For the example that follows, there are 11 "Merging" productions.
5. The merged files are distributed by one or more "Replication" production. Note that this step is not always mandatory.

For the purposes of these examples, we'll have a look at request 1926:

Request: 1926						
<input type="checkbox"/>	8191		Active	Automatic	Replication	STREAM_Replication_SETC_Request1926_Reco07-Stripping11_90000000_1
<input type="checkbox"/>	8190		Active	Automatic	Replication	STREAM_Replication_DST_Request1926_Reco07-Stripping11_90000000_1
<input type="checkbox"/>	8189		Active	Automatic	Merge	SEMILEPTONIC_Merging_Request1926_Reco07-Stripping11_90000000_1
<input type="checkbox"/>	8188		Active	Automatic	Merge	RADIATIVE_Merging_Request1926_Reco07-Stripping11_90000000_1
<input type="checkbox"/>	8187		Active	Automatic	Merge	MINIBIAS_Merging_Request1926_Reco07-Stripping11_90000000_1
<input type="checkbox"/>	8186		Active	Automatic	Merge	EW_Merging_Request1926_Reco07-Stripping11_90000000_1
<input type="checkbox"/>	8185		Active	Automatic	Merge	DIMUON_Merging_Request1926_Reco07-Stripping11_90000000_1
<input type="checkbox"/>	8184		Active	Automatic	Merge	DIELECTRON_Merging_Request1926_Reco07-Stripping11_90000000_1
<input type="checkbox"/>	8183		Active	Automatic	Merge	CHARMCONTROL_Merging_Request1926_Reco07-Stripping11_90000000_1
<input type="checkbox"/>	8182		Active	Automatic	Merge	CALIBRATION_Merging_Request1926_Reco07-Stripping11_90000000_1
<input type="checkbox"/>	8181		Active	Automatic	Merge	BHADRON_Merging_Request1926_Reco07-Stripping11_90000000_1
<input type="checkbox"/>	8180		Active	Automatic	Merge	LEPTONIC_Merging_Request1926_Reco07-Stripping11_90000000_1
<input type="checkbox"/>	8179		Active	Automatic	Merge	CHARM_Merging_Request1926_Reco07-Stripping11_90000000_1
<input type="checkbox"/>	8178		Active	Automatic	DataReconstruction	FULL_Request1926_Reco07-Stripping11_90000000_1

Shifters and GEOCs can get quite a lot of information by simply looking at the Production monitor page [?](#).

Looking why the Reconstruction production 8178 still has 2 files in status "Assigned".

Where do I get this information? Very simple: I just look at the "File Status" for that production. Two files are still in "Assigned" state, but I want to look if there were real jobs that tried to process it: to do that the fastest thing is to connect directly to the database (which today is the ProductionDB on volhcb22), but you can get this information also by looking at the production and jobs pages (the job name of every production is composed by the production number and the "TaskID" number).

```
mysql> SELECT * FROM TransformationTasks WHERE TransformationID = 8178 AND TaskID IN (22103, 21849)
+-----+-----+-----+-----+-----+-----+-----+
| TaskID | TransformationID | ExternalStatus | ExternalID | TargetSE | CreationTime | LastUpdate |
+-----+-----+-----+-----+-----+-----+-----+
| 21849 | 8178 | Done | 12614650 | IN2P3-RAW | 2010-11-08 14:12:43 | 2010-11-08 14:12:43 |
| 22103 | 8178 | Done | 12616481 | IN2P3-RAW | 2010-11-08 14:15:07 | 2010-11-08 14:15:07 |
+-----+-----+-----+-----+-----+-----+-----+
2 rows in set (0.00 sec)
```

I got the TaskIDs from the web page simply looking at "File status" for that production. First thing to notice, is that these ran at IN2P3, so it seems to be the usual case of the payload killed at IN2P3 that anyway exits with status = 0. This happens because IN2P3 has its own batch system (BQS) which in any case will be replaced hopefully soon.

I have a look at the jobs in the Jobs monitoring, and they have no logs. This is normal, because the applications could not finalize, so the logs were not uploaded.

Also, they have no pilot output, but this might be just the case where the LB lost track of the pilot:

```
[lxplus303] ~/Jobs $ dirac-admin-get-pilot-output 12614650
ERROR 12614650: Failed to determine owner for pilot 12614650
[lxplus303] ~/Jobs $ dirac-admin-get-pilot-output 12616481
ERROR 12616481: Failed to determine owner for pilot 12616481
```

Anyway, I could get the std.out:

```
[lxplus303] ~/Jobs $ dirac-wms-job-get-output 12616481
2010-11-11 10:57:08 UTC dirac-wms-job-get-output/DiracAPI INFO: Files retrieved and extracted in
Job output sandbox retrieved in 12616481/
```

By looking at it, it ends abruptly. It really seems that it has been killed. Same for the other one.

I make a last check:

```
[lxplus303] ~/Jobs $ dirac-bookkeeping-get-file-descendants /lhcb/data/2010/RAW/FULL/LHCb/COLLISION10/81356/081356_0000000280.raw:
Successful:
/lhcb/data/2010/RAW/FULL/LHCb/COLLISION10/81356/081356_0000000280.raw:
Failed: []
[lxplus303] ~/Jobs $ dirac-bookkeeping-get-file-descendants /lhcb/data/2010/RAW/FULL/LHCb/COLLISION10/81356/081356_000000023.raw:
Successful:
/lhcb/data/2010/RAW/FULL/LHCb/COLLISION10/81356/081356_000000023.raw:
Failed: []
```

I conclude that the file has not been processed: I then set their status as "Unused":

```
mysql> SELECT * FROM TransformationFiles WHERE TransformationID = 8178 AND TaskID IN (22103, 21849)
+-----+-----+-----+-----+-----+-----+-----+
| TransformationID | FileID | Status | ErrorCount | TaskID | TargetSE | UsedSE | LastUpdate |
+-----+-----+-----+-----+-----+-----+-----+
| 8178 | 3865218 | Assigned | 0 | 21849 | Unknown | IN2P3-RAW | 2010-11-08
```

```
|          8178 | 3864794 | Assigned |          0 | 22103 | Unknown | IN2P3-RAW | 2010-11-08
+-----+-----+-----+-----+-----+-----+-----+
2 rows in set (0.06 sec)
```

BE VERY -VERY- CAREFUL WHEN DOING THIS OPERATION!!!

```
mysql> UPDATE TransformationFiles SET Status = 'Unused' WHERE TransformationID = 8178 AND TaskID = 22103;
Query OK, 2 rows affected (0.07 sec)
Rows matched: 2  Changed: 2  Warnings: 0
```

Looking why Merging production 8187 still had 30 files in status "Assigned".

This is a different case from the one before. There is one merging job that tried to merge these files:

```
mysql> SELECT DISTINCT TaskID FROM TransformationFiles WHERE Status = 'Assigned' AND TransformationID = 8187;
+-----+
| TaskID |
+-----+
| 239    |
+-----+
1 row in set (0.02 sec)
```

Then I have a look at which DIRAC Job ID tried to merge these files:

```
mysql> SELECT * FROM TransformationTasks WHERE TaskID = 239 AND TransformationID = 8187;
+-----+-----+-----+-----+-----+-----+-----+
| TaskID | TransformationID | ExternalStatus | ExternalID | TargetSE | CreationTime          | LastUpdate
+-----+-----+-----+-----+-----+-----+-----+
| 239    | 8187             | Failed         | 12631051   | RAL-DST  | 2010-11-08 20:20:45 | 2010-11-08 20:20:45
+-----+-----+-----+-----+-----+-----+-----+
1 row in set (0.00 sec)
```

I discover this job is stalled. The job application ran to completion in the RAL queue when it was already in draining state before a downtime. Then it started to finalize, but I assume that, because the SEs were banned, the output went to Failover, from which it was picked up lately without problems. So, the output is now replicated and registered in the bookkeeping. The merged file even has a DataQuality Flag = 'OK'. However, I don't know why but the job was still in the queue, and stalled about 36 hours later, because the pilot was aborted.

So, what I did was again to mark these files as "Processed".

The reason why there wasn't a new job created for this case is because the DataRecoveryAgent checks if the file has already descendants registered in the bookkeeping before creating a new job. When it happens, it just stops and print out a line in the log.

Looking why there is a quite important mismatch in the number of files merged or to be merged and those produced by the production

Probably there are some runs that didn't go to merging status... is this because they have not been yet flagged or not? First of all, let's see who's left over:

```
mysql> SELECT * FROM TransformationRuns WHERE TransformationID = 8178 AND RunNumber NOT IN (SELECT RunNumber FROM TransformationFiles WHERE TransformationID = 8178);
+-----+-----+-----+-----+-----+
| TransformationID | RunNumber | SelectedSite | Status | LastUpdate          |
+-----+-----+-----+-----+-----+
| 8178             | 0         | LCG.IN2P3.fr | Active | 2010-10-28 08:50:16 |
| 8178             | 81597    | LCG.SARA.nl  | Active | 2010-10-29 15:12:27 |
+-----+-----+-----+-----+-----+
```

Production100 < LHCb < TWiki

```
|          8178 |          81605 | LCG.GRIDKA.de | Active | 2010-10-29 15:34:03 |
|          8178 |          81606 | LCG.GRIDKA.de | Active | 2010-10-29 15:34:04 |
|          8178 |          81610 | LCG.GRIDKA.de | Active | 2010-10-30 09:25:03 |
|          8178 |          81611 | LCG.GRIDKA.de | Active | 2010-10-30 09:25:04 |
|          8178 |          81608 | LCG.GRIDKA.de | Active | 2010-10-30 09:57:02 |
|          8178 |          81609 | LCG.GRIDKA.de | Active | 2010-10-30 09:57:03 |
|          8178 |          81356 | LCG.IN2P3.fr  | Active | 2010-11-08 14:12:31 |
+-----+-----+-----+-----+-----+
9 rows in set (0.00 sec)
```

(8178 is the Reconstruction prod, 8179 is the merging one)

Just to check, I looked at also the others merging and list of runs is the same. The run with number 0 does not seem to be a problem (re-check!).

Then, I look at the bookkeeping for the run files:

```
[lxplus303] ~/Jobs $ dirac-bookkeeping-run-files 81609
```

Get the descendents of one of these:

```
[lxplus303] ~/Jobs $ dirac-bookkeeping-get-file-descendants /lhcb/data/2010/RAW/FULL/LHCb/COLLIS
```

And check if they have been flagged:

```
[lxplus303] ~/Jobs $ dirac-bookkeeping-file-metadata /lhcb/data/2010/DST/00008178/0001/00008178_0
```

I discovered they were "Unchecked" so I inform the DQ manager (Marco Adinolfi) that the runs have not been yet flagged.

A couple of useful queries

This last section is meant to give some practical suggestions to experts. First of all we strongly recommend to avoid to run any SQL UPDATE statement w/o running **beforehand** the equivalent SELECT statement. You have to be 100% sure to act on the desired rows. Furthermore we strongly suggest to avoid to run any SQL UPDATE statement by trying to modify more than one field at the time. Better to split the changes in two update statements than trying to do all in once (this is a known SQL bug and you might screw up your table).

It might happen often that the number of files in Assigned is so large that is not convenient jumping from a table (TransformationTasks) to another (TransformationFiles) selecting TaskID from one side and copy them in a select query against the other... Isolating files in Status Assigned in one table must be the input for querying the ExternalTasks status. Concatenating select queries is expensive and it happened that we killed the DB for several hours leaving it running (w/o results after all). This is why this query (just change the TransformationID) is useful for investigating on large number of files not Processed.

```
select TransformationFiles.FileID, TransformationTasks.TaskID, TransformationFiles.Status, Transf
```

An even more heavy and complicated query (but this saves a lot of time if one has just to check the descendants of the input files for the various "problematic" tasks) is something like that (please note that I put also ExternalStatus = Completed or Done as it happens that the large majority of the cases of files in Assigned but job completed is just due to the file status not changed (diset issue) while the file has produced descendant.

```
mysql> select DataFiles.LFN, TransformationFiles.FileID, TransformationTasks.TaskID, Transformati
```

```
+-----+-----+-----+-----+-----+
| LFN                                     | FileID | TaskID | Status
+-----+-----+-----+-----+-----+
```

Looking why there is a quite important mismatch in the number of filesmerged or to be merged and those pr

Production100 < LHCb < TWiki

/lhcb/LHCb/Collision11/SDST/00010883/0000/00010883_00000258_1.sdsm	12699074	4513	Assign
/lhcb/LHCb/Collision11/SDST/00010883/0000/00010883_00005178_1.sdsm	13057230	3152	Assign
/lhcb/LHCb/Collision11/SDST/00010883/0000/00010883_00005230_1.sdsm	13057257	3191	Assign
/lhcb/LHCb/Collision11/SDST/00010883/0000/00010883_00005257_1.sdsm	13057268	3157	Assign
/lhcb/LHCb/Collision11/SDST/00010883/0000/00010883_00005813_1.sdsm	13057279	3322	Assign
/lhcb/LHCb/Collision11/SDST/00010883/0000/00010883_00005831_1.sdsm	13057291	3420	Assign
/lhcb/LHCb/Collision11/SDST/00010883/0000/00010883_00005893_1.sdsm	13057335	3334	Assign
/lhcb/LHCb/Collision11/SDST/00010883/0000/00010883_00005923_1.sdsm	13057349	3400	Assign
/lhcb/LHCb/Collision11/SDST/00010883/0000/00010883_00005203_1.sdsm	13058177	3222	Assign
/lhcb/LHCb/Collision11/SDST/00010883/0000/00010883_00005210_1.sdsm	13058182	3155	Assign
/lhcb/LHCb/Collision11/SDST/00010883/0000/00010883_00005248_1.sdsm	13058197	3178	Assign
/lhcb/LHCb/Collision11/SDST/00010883/0000/00010883_00005250_1.sdsm	13058199	3147	Assign
/lhcb/LHCb/Collision11/SDST/00010883/0000/00010883_00005808_1.sdsm	13058206	3444	Assign
/lhcb/LHCb/Collision11/SDST/00010883/0000/00010883_00005840_1.sdsm	13058219	3429	Assign
/lhcb/LHCb/Collision11/SDST/00010883/0000/00010883_00005867_1.sdsm	13058224	3304	Assign
/lhcb/LHCb/Collision11/SDST/00010883/0000/00010883_00005887_1.sdsm	13058228	3360	Assign
/lhcb/LHCb/Collision11/SDST/00010883/0000/00010883_00005917_1.sdsm	13058242	3393	Assign
/lhcb/LHCb/Collision11/SDST/00010883/0000/00010883_00005943_1.sdsm	13058252	3384	Assign
/lhcb/LHCb/Collision11/SDST/00010883/0000/00010883_00005976_1.sdsm	13058263	3414	Assign
/lhcb/LHCb/Collision11/SDST/00010883/0000/00010883_00005991_1.sdsm	13058268	3339	Assign
/lhcb/LHCb/Collision11/SDST/00010883/0000/00010883_00004539_1.sdsm	13121700	9192	Assign
/lhcb/LHCb/Collision11/SDST/00010883/0000/00010883_00004100_1.sdsm	13123716	6213	Assign
/lhcb/LHCb/Collision11/SDST/00010883/0000/00010883_00009523_1.sdsm	13123914	6374	Assign
/lhcb/LHCb/Collision11/SDST/00010883/0001/00010883_00012664_1.sdsm	13129031	7069	Assign
/lhcb/LHCb/Collision11/SDST/00010883/0001/00010883_00012670_1.sdsm	13129037	7071	Assign
/lhcb/LHCb/Collision11/SDST/00010883/0001/00010883_00012686_1.sdsm	13129053	7059	Assign
/lhcb/LHCb/Collision11/SDST/00010883/0001/00010883_00012694_1.sdsm	13129061	6947	Assign
/lhcb/LHCb/Collision11/SDST/00010883/0001/00010883_00012701_1.sdsm	13129068	7051	Assign
/lhcb/LHCb/Collision11/SDST/00010883/0001/00010883_00012722_1.sdsm	13129089	7070	Assign
/lhcb/LHCb/Collision11/SDST/00010883/0001/00010883_00012730_1.sdsm	13129097	7066	Assign
/lhcb/LHCb/Collision11/SDST/00010883/0001/00010883_00012731_1.sdsm	13129098	7054	Assign
/lhcb/LHCb/Collision11/SDST/00010883/0001/00010883_00012732_1.sdsm	13129099	6897	Assign
/lhcb/LHCb/Collision11/SDST/00010883/0001/00010883_00012740_1.sdsm	13129107	7068	Assign
/lhcb/LHCb/Collision11/SDST/00010883/0001/00010883_00012760_1.sdsm	13129127	7090	Assign
/lhcb/LHCb/Collision11/SDST/00010883/0001/00010883_00012786_1.sdsm	13129153	7057	Assign
/lhcb/LHCb/Collision11/SDST/00010883/0001/00010883_00012794_1.sdsm	13129161	7062	Assign
/lhcb/LHCb/Collision11/SDST/00010883/0001/00010883_00012797_1.sdsm	13129164	7061	Assign
/lhcb/LHCb/Collision11/SDST/00010883/0001/00010883_00012802_1.sdsm	13129169	7037	Assign
/lhcb/LHCb/Collision11/SDST/00010883/0001/00010883_00012805_1.sdsm	13129172	7049	Assign
/lhcb/LHCb/Collision11/SDST/00010883/0001/00010883_00012808_1.sdsm	13129175	6910	Assign
/lhcb/LHCb/Collision11/SDST/00010883/0001/00010883_00012815_1.sdsm	13129182	7076	Assign
/lhcb/LHCb/Collision11/SDST/00010883/0001/00010883_00012850_1.sdsm	13129217	6950	Assign
/lhcb/LHCb/Collision11/SDST/00010883/0001/00010883_00012871_1.sdsm	13129238	6966	Assign
/lhcb/LHCb/Collision11/SDST/00010883/0000/00010883_00002142_1.sdsm	13129945	7351	Assign
/lhcb/LHCb/Collision11/SDST/00010883/0000/00010883_00002143_1.sdsm	13129946	7444	Assign
/lhcb/LHCb/Collision11/SDST/00010883/0000/00010883_00002146_1.sdsm	13129949	7459	Assign
/lhcb/LHCb/Collision11/SDST/00010883/0000/00010883_00002167_1.sdsm	13129970	7322	Assign
/lhcb/LHCb/Collision11/SDST/00010883/0000/00010883_00002170_1.sdsm	13129973	7428	Assign
/lhcb/LHCb/Collision11/SDST/00010883/0000/00010883_00002171_1.sdsm	13129974	7432	Assign
/lhcb/LHCb/Collision11/SDST/00010883/0000/00010883_00002173_1.sdsm	13129976	7318	Assign
/lhcb/LHCb/Collision11/SDST/00010883/0000/00010883_00002177_1.sdsm	13129980	7437	Assign
/lhcb/LHCb/Collision11/SDST/00010883/0000/00010883_00002178_1.sdsm	13129981	7330	Assign
/lhcb/LHCb/Collision11/SDST/00010883/0000/00010883_00002179_1.sdsm	13129982	7347	Assign
/lhcb/LHCb/Collision11/SDST/00010883/0000/00010883_00002185_1.sdsm	13129988	7449	Assign
/lhcb/LHCb/Collision11/SDST/00010883/0000/00010883_00002186_1.sdsm	13129989	7287	Assign
/lhcb/LHCb/Collision11/SDST/00010883/0000/00010883_00002187_1.sdsm	13129990	7324	Assign
/lhcb/LHCb/Collision11/SDST/00010883/0000/00010883_00002193_1.sdsm	13129996	7420	Assign
/lhcb/LHCb/Collision11/SDST/00010883/0000/00010883_00002194_1.sdsm	13129997	7294	Assign
/lhcb/LHCb/Collision11/SDST/00010883/0000/00010883_00002201_1.sdsm	13130004	7407	Assign
/lhcb/LHCb/Collision11/SDST/00010883/0000/00010883_00002202_1.sdsm	13130005	7303	Assign
/lhcb/LHCb/Collision11/SDST/00010883/0000/00010883_00002208_1.sdsm	13130011	7278	Assign
/lhcb/LHCb/Collision11/SDST/00010883/0000/00010883_00002209_1.sdsm	13130012	7410	Assign
/lhcb/LHCb/Collision11/SDST/00010883/0000/00010883_00002218_1.sdsm	13130021	7368	Assign
/lhcb/LHCb/Collision11/SDST/00010883/0000/00010883_00002221_1.sdsm	13130024	7441	Assign
/lhcb/LHCb/Collision11/SDST/00010883/0000/00010883_00002223_1.sdsm	13130026	7451	Assign
/lhcb/LHCb/Collision11/SDST/00010883/0000/00010883_00002225_1.sdsm	13130028	7336	Assign
/lhcb/LHCb/Collision11/SDST/00010883/0000/00010883_00002229_1.sdsm	13130032	7399	Assign


```
| 26706102 | 2012-08-01 09:16:48 | 24570 | Assigned | Rescheduled | 35205201 |
| 26706103 | 2012-08-01 09:16:48 | 24570 | Assigned | Rescheduled | 35205201 |
...
```

Productions with some Unused files, which have run number = zero in production DB

Sometimes it happened that some files, even after flushing a production, are 'Unused', because their run number is zero in the ProductionDB.TransformationFiles table (though in the Bookkeeping it's a valid number!). It's a race condition: the TransformationAgent picks up the files from ProductionDB before the BookkeepingWatchAgent had inserted the run number.

As an example: this [elog](#). What to do:

```
$ dirac-transformation-debug 16771 --Status Unused

=====
Transformation 16771 : RADIATIVE.DST_Merging_Request7299_Stripping17b_90000000_1.xml of type Mer
BKQuery: {'FileType': 'RADIATIVE.DST', 'ProductionID': 16764L, 'DataQualityFlag': 'OK'}
9 files found with status Unused

9 files have run number 0, use --FixIt to get this fixed

$ dirac-transformation-debug 16771 --Status Unused --FixIt

=====
Transformation 16771 : RADIATIVE.DST_Merging_Request7299_Stripping17b_90000000_1.xml of type Mer
BKQuery: {'FileType': 'RADIATIVE.DST', 'ProductionID': 16764L, 'DataQualityFlag': 'OK'}
9 files found with status Unused

Successfully fixed run number for 9 files
```

Productions with some 'Unused' files as they only have a failover replica in the LFC but no 'regular' replica

It can happen that a file fails to be replicated to a regular SE and only has a failover replica. In this case it will not be processed by the transformation, and the file will stay in Unused status. Example:

```
dirac-dms-lfn-replicas -a
/lhcb/LHCb/Collision11/CHARMCOMPLETEEVENT.DST/00016983/0001/00016983_00017390_1.CharmCompleteEven
Successful :
    /lhcb/LHCb/Collision11/CHARMCOMPLETEEVENT.DST/00016983/0001/00016983_00017390_1.CharmComplete
    RAL-FAILOVER : (-)
srm://srm-lhcb.gridpp.rl.ac.uk/castor/ads.rl.ac.uk/prod/lhcb/failover/lhcb/LHCb/Collision11/CHARM
```

to see what is the SE where it should be replicated, get the run number for these file:

```
mysql> select RunNumber from TransformationFiles where FileID = (select FileID from DataFiles whe
+-----+
| RunNumber |
+-----+
|    98269 |
+-----+
```

and look where are the other files of this run (runs are at one site only):

```
mysql> select distinct UsedSE from TransformationFiles where TransformationID=16986 and RunNumbe
+-----+
| UsedSE |
```

```
+-----+
| GRIDKA-DST |
+-----+
1 row in set (0.09 sec)
```

how to fix it: check in the RequestDB if there is any pending request for that transfer, and try to understand why it is not happening. In this case, the request was not there. So the solution is to replicate manually:

```
> dirac-dms-replicate-lfn /lhcb/LHCb/Collision11/CHARMCOMPLETEEVENT.DST/00016983/0001/00016983_0
**** Set replica flag on Bookkeeping/BookkeepingManager
{'Failed': {}},
'Successful': {'/lhcb/LHCb/Collision11/CHARMCOMPLETEEVENT.DST/00016983/0001/00016983_00017390_1.
```

finally flush the production, and a new job should be created.

Productions with some jobs in 'Rescheduled' status since many days

Checking why some files are still in status 'Assigned', I found a job reported in ProductionDB as Rescheduled, since more than one month:

```
mysql> SELECT * FROM TransformationTasks WHERE TransformationID = 16994 AND TaskID=615;
+-----+-----+-----+-----+-----+-----+-----+
| TaskID | TransformationID | ExternalStatus | ExternalID | TargetSE | CreationTime | LastUpdateT |
+-----+-----+-----+-----+-----+-----+-----+
| 615 | 16994 | Rescheduled | 31411895 | GRIDKA-DST | 2012-04-12 14:36:55 | 2012-04-12 14:36:55 |
+-----+-----+-----+-----+-----+-----+-----+
```

not clear why, but might be a problem with Gridka batch system, as another job is found in same situation at the time:

```
mysql> SELECT * FROM TransformationTasks WHERE TransformationID = 16997 AND TaskID=632;
+-----+-----+-----+-----+-----+-----+-----+
| TaskID | TransformationID | ExternalStatus | ExternalID | TargetSE | CreationTime | LastUpdateT |
+-----+-----+-----+-----+-----+-----+-----+
| 632 | 16997 | Rescheduled | 31405549 | GRIDKA-DST | 2012-04-12 08:42:43 | 2012-04-12 16:31:19 |
+-----+-----+-----+-----+-----+-----+-----+
```

what to do: kill the jobs, then the input files will be automatically reset Unused.

Problem killing the job through the JobMonitor interface, sets the status 'killed' in the JobDB, but for some reason the status is not propagated to the ProductionDB, where it is still Rescheduled. The responsible of the propagation of job status is WorkflowTaskAgent and usually takes few minutes.

Temporary Solution Finally, it was not possible to understand why these jobs are stuck in status 'Rescheduled', however a practical solution is to update the ProductionDB and set the task's ExternalStatus = killed by hand:

```
mysql> update TransformationTasks set ExternalStatus='Killed' WHERE TransformationID = 16994 AND
```

shortly after, the files are reset as 'Unused' and a new task is created, and then a new job submitted.

fixed on 31/08/2012 solution here [↗](#)

Production with some Assigned files, though the Dirac job is reported as Done

In production 18310 (Aug 2012) and also in prod. 18117 (see <http://lblogbook.cern.ch/Operations/12057>) some files are in Assigned status and the relative jos status is Done:

```
mysql> select ExternalStatus,ExternalID,TargetSE,CreationTime,LastUpdateTime from TransformationFiles
+-----+-----+-----+-----+-----+
| ExternalStatus | ExternalID | TargetSE | CreationTime           | LastUpdateTime         |
+-----+-----+-----+-----+-----+
| Done           | 32608697  | CERN-RAW | 2012-05-20 06:13:41   | 2012-05-21 22:03:13   |
| Done           | 32612616  | CERN-RAW | 2012-05-20 10:30:27   | 2012-05-20 16:16:59   |
| Done           | 32675342  | CERN-RAW | 2012-05-21 21:55:34   | 2012-05-22 14:33:06   |
| Done           | 32675345  | CERN-RAW | 2012-05-21 21:55:35   | 2012-05-22 14:33:06   |
| Failed         | 33680504  | RAL-RAW  | 2012-06-17 17:19:43   | 2012-06-19 10:26:16   |
+-----+-----+-----+-----+-----+
5 rows in set (0.00 sec)
```

3 of the Done jobs at CERN (32608697,32675342,32675345) are in Done status but the application status is 'Brunel Exited With Status 137!' (though in Brunel_00018310_00015957_1.log no error message is reported). In std.out the error status is detected:

```
2012-05-22 14:14:11 UTC dirac-jobexec/GaudiApplication INFO: Status after the application execution
2012-05-22 14:14:12 UTC dirac-jobexec/GaudiApplication ERROR: Brunel execution completed with error
2012-05-22 14:14:12 UTC dirac-jobexec/GaudiApplication ERROR: =====
2012-05-22 14:14:12 UTC dirac-jobexec/GaudiApplication ERROR: StdError:
2012-05-22 14:14:12 UTC dirac-jobexec/GaudiApplication ERROR:
2012-05-22 14:14:12 UTC dirac-jobexec/GaudiApplication ERROR: /bin/sh: line 3: 28327 Killed
gaudirun.py $APPCONFIGOPTS/Brunel/DataType-2012.py prodConf_Brunel_00018310_00015957_1.py
2012-05-22 14:14:12 UTC dirac-jobexec/GaudiApplication ERROR: Brunel Exited With Status 137
2012-05-22 14:14:12 UTC dirac-jobexec/GaudiApplication INFO: ===== Terminating $Id: GaudiApplication
2012-04-17 15:54:09Z fstagni $ =====
2012-05-22 14:16:24 UTC dirac-jobexec/AnalyseLogFile INFO: ===== Executing $Id: AnalyseLogFile.py
2012-04-17 14:13:28Z fstagni $ =====
```

However, check if there are descendants and if not, the jobs can be set to Killed in production db and this should reset the files to unused.

Explanation: these status (completed, done) are set by the JobWrapper, not by the Workflow. The workflow is indeed correct, it reports errors and so on. But the process was killed, in all these cases, so this points to a bug in the Watchdog. The difficult part is that for debugging it would be VERY useful to have the pilot output, which unfortunately does not exist. Opened a github issue (04/09/2012).

Temporary Solution In such case, the input files of the job should be reset to Unused using the command: `dirac-transformation-reset-files`. This will of course create an inconsistent situation in Production DB because the relative job is reported as Done.

waiting for the fix.

Production with some Assigned files and corresponding Tasks have external status Failed

Example, in production 18979 we found two files in status Assigned, but the corresponding task is reported as Failed:

```
mysql> select TransformationFiles.FileID, TransformationFiles.RunNumber,TransformationFiles.LastUpdateTime
+-----+-----+-----+-----+-----+
| TransformationFiles.FileID | TransformationFiles.RunNumber | TransformationFiles.LastUpdateTime |
```

Production100 < LHCb < TWiki

FileID	RunNumber	LastUpdate	TaskID	Status	ExternalStatus	ExternalID
25982109	120199	2012-07-10 15:30:26	5423	Assigned	Failed	34596253
26192544	120539	2012-07-13 10:43:12	15506	Assigned	Failed	34638025

Why aren't the input files reset to Unused?

DataRecoveryAgent (DRA) clearly skips this job:

```
2012-08-28 10:27:42 UTC Transformation/DataRecoveryAgent INFO: Removing jobID 34596253 from cons
```

it removes the job from the list of jobs to be treated because it has pending requests which are not in status 'Done'. Looked in JobMonitor and in fact the jobs has 2 diset requests, but in status Done. So it is not clear why the DRA agent thinks are pending!

TO BE UNDERSTOOD

Moreover, the two files have descendants, so they were actually processed:

```
> dirac-transformation-debug --Info=Files,Jobs --Status=Assigned 18979
=====
Transformation 18979 : DataReconstruction_Request8240_Reco13c_90000000_1.xml of type DataReconstr
BKQuery: {'StartRun': 119956L, 'ConfigName': 'LHCb', 'EndRun': 123803L, 'EventType': 90000000L, '
2 files found with status Assigned

LFN: /lhcb/data/2012/RAW/FULL/LHCb/COLLISION12/120199/120199_0000000139.raw - Run: 120199 - Statu
LFN: /lhcb/data/2012/RAW/FULL/LHCb/COLLISION12/120539/120539_0000000168.raw - Run: 120539 - Statu
List of jobs found:
34638025 34596253
```

For the one of the files, DataRecoveryAgent (DRA) reports that it can't be reset to Unused because it has descendants for that production:

```
2012-09-03 14:13:23 UTC Transformation/DataRecoveryAgent INFO: !!!!!!!! Note that transformation
2012-09-03 14:13:23 UTC Transformation/DataRecoveryAgent INFO: Job 34638025, Files ['/lhcb/data/
```

for the other file, no mention in DRA. To be understood why.

Solution there is no solution for this case, unless updating by hand the status of the files in ProductionDB (strongly not recommended). Possible solution: make DRA update the status of the file in ProductionDB to Processed, or add a functionality to some command line tool (like dirac-transformation-debug) to check if a file in status = Processed has descendants for the given production and in such case update the status of the file in the ProductionDB. Currently, there is no procedure to fix this case.

TO BE FIXED

Stripping productions with some Unused files after several days

It can happen that some stripping production have some files Unused, even after several days that the files have been assigned to the production. This can happen also if the production has been created with the ByRunWithFlush plugin, which in principle should schedule jobs also for files that are less than the group size (usually we set 2 files per job for stripping and 5 files per job for the femto stripping, but it can change).

In order to unblock the situation it is necessary to select the production and Flush it. Important: Selecting a single run and flushing it will have no effect.

Production with some Assigned files and correspondingTasks have external status Failed

Corrupted or lost input files

It may happen that some jobs repeatedly fail due to missing or corrupted input file. This happens often for merging jobs, which have one replica, and as it is only little statistics, it is preferable to set the file problematic, instead of reproducing it. See this example [\[1\]](#), the file is corrupted so it should be set problematic also in the LFC, not only for this particular production. The file has only one replica:

```
> dirac-dms-lfn-replicas /lhcb/LHCb/Collision12/BHADRONCOMPLETEEVENT.DST/00019167/0002/00019167_0
Successful :
    /lhcb/LHCb/Collision12/BHADRONCOMPLETEEVENT.DST/00019167/0002/00019167_00024559_1.BhadronComp
    IN2P3-BUFFER : srm://ccsrn.in2p3.fr/pnfs/in2p3.fr/data/lhcb/buffer/lhcb/LHCb/Collision1
```

so the command to set it problematic is:

```
> dirac-dms-set-problematic-files /lhcb/LHCb/Collision12/BHADRONCOMPLETEEVENT.DST/00019167/0002/0
Now processing 1 files
Getting replicas from FC (chunks of 1000): .
Checking with BK
Checking with Transformation system (chunks of 100): .

Replicas set (P) in FC for 1 files

Replica flag removed in BK for 1 files

1 files were set Problematic in the transformation system
    1 files set Problematic for transformation 19169
Execution completed in 12.83 seconds
```

as it's the only replica, the Bookkeeping replica is set to No:

```
> dirac-bookkeeping-file-metadata /lhcb/LHCb/Collision12/BHADRONCOMPLETEEVENT.DST/00019167/0002/0
FileName
/lhcb/LHCb/Collision12/BHADRONCOMPLETEEVENT.DST/00019167/0002/00019167_00024559_1.BhadronComple
```

replica is set problematic in the LFC:

```
> dirac-dms-lfn-replicas -a /lhcb/LHCb/Collision12/BHADRONCOMPLETEEVENT.DST/00019167/0002/000191
Successful :
    /lhcb/LHCb/Collision12/BHADRONCOMPLETEEVENT.DST/00019167/0002/00019167_00024559_1.BhadronComp
    IN2P3-BUFFER : (P) srm://ccsrn.in2p3.fr/pnfs/in2p3.fr/data/lhcb/buffer/lhcb/LHCb/Collis
```

and in ProductionDB too.

File in status Assigned and relative job is Done: input data missing from JDL

In a stripping19c merging production was found a rare case of a file in status Assigned, whose relative job was Done. The problem was that the InputData field in the JDL was empty! so the job didn't actually process the input file, and produced an output file with no relevant data in it. In the Bkk summary there was no relation with the input file, so it was not registered as descendant of the input. See [elog \[2\]](#)

Solution remove the output file with `dirac-dms-remove-file`, and reset to Unused the input.

Some comments and warnings

dirac-transformation-cli only update some status

This tool can only set some file status but not all. However, it will not give a warning! e.g. if a file needs to be set Unused it WILL NOT work:

```
(Cmd) setFileStatus 19770 /lhcb/LHCb/Collision12/FULL.DST/00019785/0002/00019785_00027742_1.full
Updated file status to Unused
```

but in reality the files is still in MaxReset in production db:

```
mysql> select Status, LastUpdate from TransformationFiles where FileID=(select FileID from DataFi
+-----+-----+
| Status   | LastUpdate |
+-----+-----+
| MaxReset | 2012-09-18 12:55:16 |
+-----+-----+
```

TO BE FIXED or at least should not print a misleading message.

the file can be reset with:

```
> dirac-transformation-reset-files 19770 MaxReset --LFNs=/lhcb/LHCb/Collision12/FULL.DST/0001978
1 files were reset Unused in transformation 19770
```

No tool available to set the files status to Processed

Sometimes files are processed, but their status is not correctly updated in ProductionDB, see this example [?](#). So, you need to update the status of 150 files to Processed. The dirac-transformation-cli only allows to update one file at the time, and the dirac-transformation-reset-files only reset to Unused. Maybe this script could be adapted to update the status also to Processed.

TO BE DONE

No tool to retrieve all the failed jobs for a given LFN

When a production has some files in MaxReset it is important to check the reason of failure of all the jobs. Currently, no tool can provide that. The dirac-transformation-debug prints out only the last failed job. In order to have all the jobs, a query has to be done to the ProductionDB.

would be useful to have a tool able to report the full list of jobs for a given LFN. It would be even better if the tool could print out also the minor status and application status of the jobs. In this way one can avoid entering the jobs in job monitor, and would save a lot of time

-- FedericoStagni - 15-Nov-2010

This topic: LHCb > Production100

Topic revision: r28 - 2018-09-23 - MarcoCattaneo



Copyright &© 2008-2019 by the contributing authors. All material on this collaboration platform is the property of the contributing authors.

Ideas, requests, problems regarding TWiki? Send feedback