
Production shifter tutorial

v1, June 2024

Aaron Higuera
Elisabetta Pennacchio



RICE UNIVERSITY



- The procedure for initiating and producing simulation, reconstruction, or data processing campaigns is described in the document *Production and Processing Campaign Policy* available on DocDB [DUNE-doc-29278-v4](#)
- These slides aim to describe how to configure, test and run a production campaign by following the policy document ([DUNE-doc-29278-v4](#))
- In particular the role of the production shifter and the actions to be taken are highlighted.

Pre-requisites:

- justIN: Workflow management <https://justin.dune.hep.ac.uk/docs/>
- Brief introduction to RUCIO, MetaCat and justIN [DUNE-doc-30145-v9](#)

Further reading:

- RUCIO: location management <https://rucio.github.io/>
- MetaCat Data catalog management <https://metacat.readthedocs.io/en/latest/>

Production organization

The procedure for initiating and running a campaign consists mainly of 5 sequential steps. Each step has to be completed before moving to the next one:

1. Setup the campaign
2. Test the campaign
3. Validation sample production
4. Full scale production
5. Close out

These 5 steps are described in the following slides

1. Setup the campaign

- The production request is submitted by a physics working group (requester), and presented at the production meeting
- The production shifter is appointed
- An entry in the [production](#) wiki page is created
- The shifter log (google doc) is created as well, its link has to be added to the wiki page. It has to be accessible by everybody
- The shifter takes care of describing the production request in the shifter log by providing the information listed in the next page. All the entries of the list are specified in the production request.

Information to be added to the google doc:

1. Code version to be used (tagged version and from a DUNE official repo)
2. Workflow description: list of processing steps and fcl file to be used for each step, number of events/job
3. Validation sample description
4. The list of output files to be copied on tape. The filesize has also to be provided. *Example: for the campaign xyz , reconstruction output has to be copied, and each file has a size of 1 GB. 100k files will have to be copied for a total expected size of 100TB*
5. Estimation of resources : CPU, memory and storage
6. List of samples/runs to be processed, ordered by priority
7. Number of events/sample(run)
8. List of foreseen final output datasets

- Based on this information (in particular points 4,5 and 6), a “workflows submission plan” (*how many workflows have to submitted, how many jobs/workflow*) can be drafted by the shifter
- This plan, including production of both validation and full samples, has to be discussed at the weekly production meeting, and agreed with production and data management team. The name of the campaign has to be agreed as well at the production meeting

Now the shifter, if necessary with the help of an experienced member of the production team, can start organizing the campaign. First steps are:

1. define the working dir on a dunegpvm machine (connect as dunepro, not with a personal account). Add this information to the shifter log
2. prepare the jobscript. This can be done by modifying jobscripts already used in previous campaigns. Jobscript documentation is available [here](#). Metadata creation has to be checked carefully with the production and data management team, and discussed at the production meeting, and may require some time. So, while discussing/finalizing metadata, it is suggested to start testing the campaign

2. Test the campaign

1. run the jobscript interactively on a dunegpvm machine. A couple of events are enough. What to check:
 - 1.1. the jobscript runs with no bash/Art/larsoft errors
 - 1.2. output files look as expected
 - 1.3. metadata file(s) is (are) created. Metadata may not be finalized at this stage
2. once previous step validated, test jobscript submission on justIN. At testing stage **ALWAYS** use `- scope usertests` option
submit a few jobs (~10), with 1 or 2 events/ job: the goal of this step is to check that the workflow is correctly implemented in justIN, so a fast feedback is important. What to check:
 - 2.1. verify exit code of the jobs (see next pages) → all zero ok, if not check error codes. The goal is to identify the origin of the error and fix it.
 - 2.2. check metadata (see next pages)

Exit code verification

Workflow ID	Description	User	Scope	State	MQL
2251		epennacc@fnal.gov	vd-protodune	finished	monte-carlo 500

Workflow 2251

Description	
State	finished
Message	Finished as all input files are in terminal states
MQL	monte-carlo 500
Scope	vd-protodune
HTCondor Group	group_dune.prod_mcsim
Finding	once
Submitted by	epennacc@fnal.gov
Created	2024-06-07 09:48:36
Submitted	2024-06-07 09:48:36
Started	2024-06-07 09:52:18
Finished	2024-06-07 11:03:27
Events for this workflow	

Events

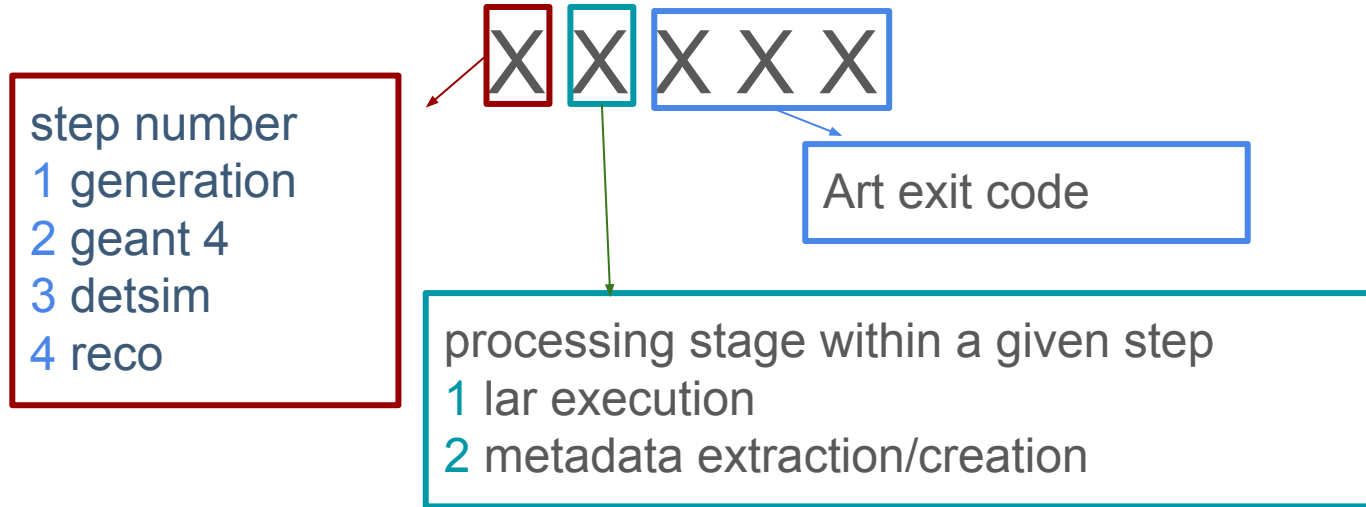
Event Type: ANY Workflow ID: Stage ID: File DID: Jobsub ID:

Jobscript exit: 0 Site: ANY Entry: ANY RSE: ANY **Filter**

jobscript error code

Following what implemented in PDVD, an error code ID system is put in place

The main idea is to define a 5 digits exit code:



output files in metacat

Workflow 2251

Description	
State	finished
Message	Finished as all input files are in terminal states
MQL	monte-carlo 500
Scope	vd-protodune
HTCondor Group	group_dune.prod_mcsim
Finding	once
Submitted by	epennacc@fnal.gov
Created	2024-06-07 09:48:36
Submitted	2024-06-07 09:48:36
Started	2024-06-07 09:52:18
Finished	2024-06-07 11:03:27
Events for this workflow	

1

File states per stage

Stage ID	Files	Finding	Unallocated	Allocated	Outputting	Processed	Not found	Failed
1	500	0	0	0	0	500	0	0

2

Files for Workflow 2251, Stage 1

3

Page: 1 Next

DID	Attempts	State	Allocated			
			Time	RSE	Site	Jobs
monte-carlo-002251-000500	1 / 6	processed	2024-06-07 10:22:44	MONTECARLO	UK_Brunel	219
monte-carlo-002251-000499	1 / 6	processed	2024-06-07 10:22:44	MONTECARLO	UK_Brunel	219

File monte-carlo-002251-000500

Workflow ID	2251
Stage ID	1
Size	0 bytes (0.00GB)
State	processed
Processed	2024-06-07 10:52:29
Replicas	MONTECARLO: 000500
Allocations	1 / 6
Jobs	219948.124@justin-prod-sched01.dune.hep.c.uk 2024-06-07 10:22:44 (finished UK_Brunel MONTECARLO)
List file events	

4

Jobscript	Real time	30m (1847s)
	CPU time	24m (1441s = 78%)
Outputting started	2024-06-07 10:49:59	
Output files	vd-protodune:prod_beam_protodunevd_20240607T102244Z_000500_gen_g4_stage1.root vd-protodune:prod_beam_protodunevd_20240607T102244Z_000500_gen_g4_stage1_g4_stage2.root vd-protodune:prod_beam_protodunevd_20240607T102244Z_000500_gen_g4_stage1_g4_stage2_detsim.root vd-protodune:prod_beam_protodunevd_20240607T102244Z_000500_gen_g4_stage1_g4_stage2_detsim_reco.root	

File vd-protodune:prod_beam_protodunevd_20240607T102244Z_000500_gen_g4_stage1_g4_stage2_detsim_reco.root

Workflow ID	2251
Size	669419480 bytes (0.67GB)
State	output
Processed	
Replicas	
Allocations	0 / 6
Jobs	
List file events	Find in MetaCat



DUNE/ProtoDUNE MetaCat Production

[users](#)
[roles](#)
[namespaces](#)
[datasets](#)
[find file](#)
[categories](#)
[query](#)
[named queries](#)
[filters](#)

How to check that metadata are correct

- 1) login as dunepro on a dunepgvm machine
→ The script to be run is in preparation

Results of these checks have to be discussed at the production meeting, with production and data management team

- After metadata validation (fields are correctly filled) , one has to check Metacat dataset creation, to verify that **ALL** needed metadata are present. This test has to be run and results have to be checked by experimented people in the production and data management teams. Validation is done by running the script `refactored_CollectionCreator.py` in test mode, by providing an input json file.

Summary of tests to be performed before moving to the production of the validation sample.

1. run interactively (dunepro, on dunegpmv machine)
 - 1.1. josbscript run and metadata are created
2. submit quick workflows (scope usertests)
 - 2.1. workflow is implemented correctly
 - 2.2. metadata are uploaded in Metacat
 - 2.3. Metacat dataset can be created

The results of these tests have to be documented in the shifter log. Before moving to next step, they have to be validated by the production and data management teams

3. Validation sample production

- Once the preliminary tests done and results validated, the production of the validation sample (*pag.6, point 3*) can start, accordingly to the “workflows submission plan” (pag. 7)
- **Remember:**
 - to set the correct number of events in the jobscript
 - the scope is no longer usertests
- Before launching jobs at full speed, submit a workflow with few jobs. A mistake is always possible, but one has to realize that removing wrong files (not in *usertests*) from a bad workflow requires a lot of work to data management people, so it is better/suggested to run one more test in the final configuration
- If results are ok, inform the production group, and start the production of the validation sample

Suggestion: prepare all the *justin simple-workflow* you will need to run in advance, and write them in the shifter notes

example taken from LBL2 reco campaign

```

workflow$ justin simple-workflow --jobscript dec2023_FD_MC_production.jobscript --env DETPROD=HD --env
HCPROD=FHC --env INPUT_TAR_DIR_LOCAL="$INPUT_TAR_DIR_LOCAL" --mql "files from
higuera:fardet-hd__fd_mc_2023a__mc__hit-reconstructed__prodgenie_nu_dune10kt_1x2x6.fcl__v09_78_01d01_
_preliminary skip 6000 limit 1000" --rss-mb 4000 --max-distance 30 --scope fardet-hd --output-pattern
"*reco2.root:fardet-hd-reco2_ritm1780305_020124_r0" --output-pattern
"*reco2_ana.root:fardet-hd-reco2ana_ritm1780305_020124_r0"
    
```

with skip and limit values defined in a table like this:

Input Dataset -- HD, nu		Total Files: 20784							
HD, nu, FHC	Output Dataset Template	Skip	Limit	Remaining	Workflow ID	State	Missing Replicas		Notes
	fardet-hd-{type}_ritm1780305_nu_fhc_skip0_limit6000_1650	0	6000	14784	1650	Finished			Output dataset needs triage: contains various errors due to memory issue
	fardet-hd-{type}_ritm1780305_nu_fhc_skip6000_limit1000_1638	6000	1000	13784	1638	Finished			Original submission was 1513 and dataset fardet-hd-{type}_ritm1780305_
	fardet-hd-{type}_ritm1780305_nu_fhc_skip7000_limit5000_1630	7000	5000	8784	1630	Finished			1 Not Found and 308 Failed files: resubmitted with more memory and to r
	fardet-hd-{type}_ritm1780305_nu_fhc_skip12000_limit5000_1631	12000	5000	3784	1631	Finished			1 Not Found and 257 Failed files resubmitted with more memory and to re
	fardet-hd-{type}_ritm1780305_nu_fhc_skip17000_end_1632	17000	-1	0	1632	Finished			9 Failed files resubmitted with more memory and to replace missing files

The table can be filled during workflow submissions, to keep record of what has been done

- Once the validation sample is completed, stop workflows submission
- Report this at the production meeting by preparing a short summary containing the following information:
 - # of jobs, % jobs with exit code=0, # of produced events
- The campaign is paused, one or more datasets have to be produced, and advertised to the requesters, who has to provide a feedback.
- The procedure to be followed in order to validate this pre-production sample and move to full production is described in the *Production and Processing Campaign Policy* ([DUNE-doc-29278-v4](#))

4. Full scale production

- Once the sample validated, the request to initiate the full-scale production is approved. Full production can start, following the *workflows submission plan*
- Keep shifter logs updated, and report campaign progress at the production meeting
- While a campaign is running, a constant monitoring is needed. This means checking at least twice per day how jobs execution is progressing.
During first days of full scale production it may be needed to check more often
When the campaign is progressing steadily, two times per day may become enough
- The following pages describe how to do monitoring

Monitoring production jobs

While running a campaign, it is crucial to monitor constantly the progression of jobs execution, to identify problems

From justin dashboard:

1. check that submitted workflows are not put in “paused” status: this means that more than 10 jobs have gone wrong (jobscript wrong, problems with input files...)
2. for a given running workflow, check jobs exit status, and in case of errors, check if they are happening to a particular site or RSE → if it is the case, send a message on slack
3. once a workflow is finished, check how many jobs have exit code=0, and report it in the shifter notes. In case of errors, quantify numbers of different error code

Events

Event Type: ANY Workflow ID: 2299 Stage ID: File DID: Jobsub ID:

Jobscript exit: Site: ANY Entry: ANY RSE: ANY **Filter**

A short report has to be prepared and presented at the weekly production meeting.

The report should contains:

1. A summary table as:

workflow id	# jobs	status	# jobs error code=0	success rate	# of events	comments
12345	10000	finished	9800	98%	98000	
23456	10000	running				

with a description of the errors (in case of high number of failed jobs)

2. the progress status of the campaign (1000 workflows finished, 200 still to be run,....)
3. distributions from monitoring links (see next 2 pages)

Monitoring links

1. running jobs per user (last 24 hours), select *dunejustin*

<https://lsvip.fnal.gov/monitor/d/000000053/experiment-batch-details?orgId=1&var-experiment=dune&var-pool=dune-global&var-pool=fifebatch&viewPanel=1>

The idea is to check how many jobs are running: while running a production, this number can be kept stable, and if it is decreasing, it means that a new workflow can be submitted. The reference number of maximum number of jobs to be run simultaneously has to be discussed/checked with production and data management team

also check:

2. held jobs (last 24 hours)

<https://lsvip.fnal.gov/monitor/d/000000053/experiment-batch-details?orgId=1&var-experiment=dune&var-pool=dune-global&var-pool=fifebatch&viewPanel=13>

3. offsite running jobs (last 24 hours), for a global overview of the system

<https://lsvip.fnal.gov/monitor/d/000000053/experiment-batch-details?orgId=1&var-experiment=dune&viewPanel=10>

5. Close out

- Once all workflows finished, compute the number of produced events in absolute terms and as a percentage of the total number initially requested. Example 99400 events produced out of 10K requested, that means 99.4%
- prepare a list of the output datasets to be produced.
- During a production meeting it can be decided to tag the campaign as finished accordingly to *Production and Processing Campaign Policy* [DUNE-doc-29278-v4](#).
- In case the campaign cannot be closed, some actions will be defined (example: some more workflows to submit are identified)
- Procedures to follow in case the campaign can be closed are described in the pages

- Ask data management people to create datasets and RUCIO containers
- Create a pdf version of the shifter log, and upload it on the wiki
- A production report has to be prepared → template in preparation
- Commit the jobscript file and any input file used to the production [git](#) repository
- The production report will be presented during a production meeting, and then will be uploaded (with the pdf version of the shifter log) on DocDB

Conclusions (1) : 5 main steps

Set up the campaign

*creation of shifter log
jobscript setting up*

ok

Test the campaign

*Interactive tests
Workflows submission (usertests)
Understand and fix errors (if any)
Metadata validation; check dataset creation*

ok

Validation sample production

*Workflow submission
monitoring*

ok

Full scale production

*Workflow submission
monitoring*

ok

Close out

*commit scripts to git
production report preparation*

Conclusions (2) : discussions and information exchanges are fundamental!

