

-- AndreiTsaregorodtsev - 17 Nov 2008

## DIRAC Production Management

This page describes the procedures and tools to define and follow the LHCb productions within the DIRAC framework

### Workflow Framework

The **Workflow Framework** provides a mechanism to describe sequences of execution units of arbitrary complexity. The execution units in the Workflow Framework are called Modules. The Modules can be combined in Steps. Steps can be combined in complex sequences forming an execution DAG.

### Production Template

The **Production Template** defines the workflow of LHCb production jobs using the Workflow Framework. It is a generic description of one class of production jobs. Several Production Templates cover all the possible LHCb productions. Combined with the parameters provided with the **Production Request**, the Production Template forms the **Production Workflow** which is a full description of the LHCb applications that are executed in the production runs.

Each Step in the LHCb production procedure is created to execute one Application together with possible auxiliary modules to verify the Application software and results of the Application execution. The final Job Finalization Step takes care of the Job results bookkeeping, data uploading and creating any failover requests as needed.

### Production Request

**Production Request** is an object which allows to capture the details of a user request to produce a set of data. It also serves as a token allowing the user to follow the progress of the request execution and to navigate to the corresponding production information.

**Production Request** is composed of the following parameters:

- ID and a Name
- Identity of the requestor
- Creation Date
- Priority
- Type
- Simulation Conditions or Data Taking Conditions ( depending on the Type of the Request )
- Processing Pass specifying the details of the Applications to be used
- Event Type
- Input Data specification In case of Data Processing requests

The Production Request is supposed to be created by Physics coordinators. The creation is followed by validation, testing and final approval by the LHCb Production Manager. After the Production Request is validated, the corresponding Production is created and put into operation. The Production Request monitor allow users to follow the progress of execution of the corresponding Productions.

### Production

**Production** is a logical object which encapsulates all the necessary information necessary to create a workload corresponding to a **Production Request**. The workload is created as a number of jobs associated

with the **Production**. Each job has a full set of parameters necessary for its execution, including the Application Workflow, input data, etc.

The **Production** manipulation includes the following operations provided that the corresponding **Production Request** and **Workflow** are defined:

- Production creation.
- Production manipulation:
  - ◆ Starting, stopping production
  - ◆ Production job submission
  - ◆ Production extension
  - ◆ Creation of derived productions
- Production finalization.
- Production monitoring.

### Production creation

**Production** is created with *createProduction* CLI command ( or with the eventual Web interface ). At creation time the following parameters are provided:

- Workflow XML file path *or* the name of a workflow from the **Workflow Repository**;
- Production Request reference;
- For "Processing" type Productions - specification of the input data in one of the following forms:
  - ◆ Bookkeeping Query in a form that can be passed to the Bookkeeping Service in order to obtain the list of input files
  - ◆ Regular expression filter that can be applied to file LFNs
- For "Processing" type Productions - specification of how many input files are grouped together for a single job

As a result, the new **Production** is created with the *New* status. It is created with no input data associated and no jobs generated.

As soon as the new Production is created a special ProductionInputData Agent will attempt to resolve all the available input data files according to the Production definition by either querying the Bookkeeping service or looking for the eligible LFNs in the ProductionDB internal catalog. After the ProductionInputData Agent performed the input data resolution it sets a LastInputUpdate time stamp. This time stamp is used to invoke the ProductionInputData Agent later at regular predefined intervals to update the set of input files. All the files added to the **Production** by the ProductionInputData Agent are given initially *Unused* status.

### Production Parameters

**Production** definition can be complemented by a number of parameters. Currently the following parameters can be defined:

- MaxNumberOfJobs - this parameter is applicable to the *Simulation* type Productions. It will not allow to extend the number of jobs for the Production beyond this limit
- AncestorDepth - this parameter is applicable to the *Processing* type Productions. It will trigger getting all the ancestors for each input file and take the ancestor locations into account when creating jobs.
- PluginType - this parameter determines which algorithm will be used to create jobs by the Transformation Agent ( see below ).
- SubmissionType - this parameter can take values *Automatic / Manual* and controls the way how the job are submitted ( see below ).

### Creation of a Derived Production

It is sometimes necessary to update considerably the Production definition and process the remaining files with the new settings. This is achieved by defining a *Derived* Production. It is defined by the *createDerivedProduction* CLI command which is similar to the *createProduction* command with an additional argument specifying the original Production. Once the Derived Production is created the original Production is moved to the *Finished* status if not yet there. All the jobs in the original Production that are not in any of the final states ( Done, Completed, Failed or Killed ) are either killed ( for Running jobs ) or deleted.

The Derived Production inherits from the original Production all its files with the following rules applied:

- Files in *Processed* and *Problematic* states are inheriting this state as well as the WMS JobID of jobs that processed them;
- Files in all the other states ( *Unused* , *Assigned* ) are moved as *Unused*.

## Production manipulation

### Starting, Stopping Production

**Production** is initially created in the *New* state. This state does not allow to create jobs associated with the **Production**. To create jobs the **Production** should be moved to the *Active* state. This is done by the *start* command of the CLI or with a corresponding command in the Web interface. The **Production** in the *Active* state is allowing to create jobs. The creation of jobs is different for Simulation and Processing type Productions

- For the *Simulation* type productions the new jobs are created by *extendProduction* CLI command. This will create a specified number of new jobs.
- In the current version of the ProductionDB for the *Processing* type productions the new jobs are created by a Transformation Agent. The Transformation Agent runs periodically and for each *Active* production checks the files in *Unused* state to see if there are enough files for grouping on the same site in order to create jobs. It creates jobs if it finds enough eligible files.

**Production** can pass to *Stopped* state using *stop* CLI command or with a corresponding command in the Web interface. For the Production in the *Stopped* state no new jobs can be created and submitted.

### Job Submission

Job submission for the given **Production** is performed either manually or automatically depending on the *SubmissionType* parameter value

- Manual job submission is performed by *submitJobs* CLI command which allows to specify
  - ◆ Production ID
  - ◆ Number of jobs to submit
  - ◆ Destination site (optionally)
- Automatic job submission is performed by ProductionJob Agent which runs periodically and for each active Production attempts to submit a predefined number of jobs (50).

### Production Finalization

The Production finalization consists in submission of jobs for the remaining non-processed files. For that the Production is passed to the *Flushed* state using *setProductionStatus* CLI command. For the Productions in this state the Transformation Agent creates jobs for all the remaining *Unused* files irrespective to the specified number of files per job. Once the jobs are created, the Production passes to *Finished* state. For the Productions in this state no new files are added and no new jobs can be created. The Production in *Finished* state can not be restarted again.

---

This topic: LHCb > DIRACProductionManagement

Topic revision: r4 - 2009-02-18 - AndreiTsaregorodtsev



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