Table of Contents

Data Management commands

This page describes the usage of the LHCbDirac data management (DMS) commands. As they interface may vary with time and introduction of new facilities, note that the `--help` option can be used to get the syntax.

The LHCbDirac DMS relies on Logical File Names (LFNs) that are always of the form `/lhcb/.....`, and on LHCbDirac Storage Elements (SE) for which the first part always indicates the site where it is located. The instance of a file at an SE is called a *replica*. Note that some replicas may be *inactive*, i.e. cannot be used for accessing files. This is always the case for archive replicas (SE name of the form <Site>-ARCHIVE), furthermore some SEs may be temporarily unavailable or files not accessible (e.g. in case a disk server is down at a site).

A list of active SEs can be obtained with `dirac-dms-show-se-status`.

A replica has a unique SRM URL (SURL) that is used to query the Storage Resource Manager (SRM) for getting informations on the replica or getting a Transport URL (tURL), a.k.a. Physical File name (PFN) to be used by Gaudi.

Common command options

The DM scripts are based on a shared command line options parsing tool (DMScript). These options are split into sets for different purposes:

List of files

The parsing of file names is trying to extract LFNs out of any string that contains an LFN (e.g. SURL, tURL, output of other commands, Gaudi options). It works in most cases, as long as there is a separator after the LFN. In the case the LFNs are in a file or from the terminal, lines not containing an LFN are ignored.

Note that if the command is piped with the output of another command, the list of LFNs is supposed to come from the pipe unless otherwise specified. This allows chaining DMS or other commands through pipes in a handy way.

```
  -    --File=           : File containing list of LFNs
  -l:  --LFNs=           : List of LFNs (comma separated)
  -    --Terminal        : LFNs are entered from stdin (--File /dev/stdin)
```

If a list of files was used through a pipe, the `--File` or `--Terminal=` option above, the last list of files used may be re-used using `--LastLFNs`. Note that this may not always work, depending on how the software is installed. This should however be fixed in a coming release (July 2019).

Here is an example of piping commands:

```
[localhost, Patch] ~ $ dirac-transformation-debug 24847 --Status Processed --Info files | dirac-b
 Transformation 24182 (Active) : Request_11636_DataStripping_Stripping20r0p1_EventType_90000000_F
BKQuery: {'StartRun': 111181L, 'ConfigName': 'LHCb', 'EndRun': 133785L, 'EventType': 90000000L, '

5 files found

/lhcb/LHCb/Collision12/FULL.DST/00024847/0000/00024847_00000001_1.full.dst - Run: 115327 - Status
/lhcb/LHCb/Collision12/FULL.DST/00024847/0000/00024847_00000094_1.full.dst - Run: 132338 - Status
/lhcb/LHCb/Collision12/FULL.DST/00024847/0000/00024847_00000020_1.full.dst - Run: 124129 - Status
/lhcb/LHCb/Collision12/FULL.DST/00024847/0000/00024847_00000069_1.full.dst - Run: 124915 - Status
/lhcb/LHCb/Collision12/FULL.DST/00024847/0000/00024847_00000031_1.full.dst - Run: 124214 - Status
  .
```

SE and sites

```
-g:  --Sites=          :   Sites to consider [ALL] (comma separated list)
-S:  --SEs=            :   SEs to consider [ALL] (comma separated list)
```

For Tier1s, the name of the site can be a short name (CERN, CNAF, GRIDKA, IN2P3, NIKHEF, PIC, RAL, SARA). When the command requires a list of SEs and a Site is specified, all SEs at that site are considered.

Storage Elements can also be a StorageElementGroup if defined in the Configuration of LHCbDirac, for example Tier1-DST, Tier1-BUFFER etc...

Bookkeeping query switches

```
-P:  --Productions=    :    Production ID to search (comma separated list)
-f:  --FileType=       :    File type (comma separated list, to be used with --Production) [All
-    --ExceptFileType= :    Exclude the (list of) file types when all are requested
-B:  --BKQuery=        :    Bookkeeping query path
-r:  --Runs=           :    Run or range of runs (r1:r2)
-    --DQFlags=        :    DQ flag used in query
-    --StartDate=      :    Start date for the BK query
-    --EndDate=        :    End date for the BK query
-    --Invisible       :    See also invisible files
```

The `--BKQuery` option consists of a path constructed as follows:

`/DataType/Activity/Conditions/ProcessingPass/EventType/FileType`

- `DataType` : this is either the Online partition for real data (in particular `LHCb`) or `MC` for simulated data (mandatory)
- `Activity` : this is the generic activity, for example `Collision12` or `MC2012` (mandatory)
- `Conditions` : data taking or simulation conditions, for example `Beam4000GeV-VeloClosed-MagDown` (if omitted: any)
- `ProcessingPass` : this is the level of processing, for example `RealData/Reco14/Stripping20` (if omitted: any)
- `EventType` : numeric event type id (or comma-separated list of ids), for example `90000000` (if omitted: any for MC, 90000000 for real data)
- `FileType` : file type of comma-separated list of file types
    - ♦ Omitted : all file types
    - ♦ `ALL.DST` or `ALL.MDST` : all DST or MDST files except those specified in `--ExceptFileType`
        - ◊ By default `--ExceptFileTypes` are all xxx.ETC files, all xxxHIST files
    - ♦ `ALL` : all file types except those specified in `--ExceptFileType`
    - ♦ `ALL.HIST` : all histogram file types

Some examples:

- `/LHCb/Collision12//RealData/Reco14` : all files from Reco14 of 2012 real data
- `/LHCb/Collision12//RealData/Reco14/Stripping20` : all files for Reco14/Stripping20 of 2012 data
- `/LHCb/Collision12//RealData/Reco14/Stripping20//ALL.MDST` : all µDST files
- `/LHCb/Collision12//RealData/Reco14/Stripping20//ALL.MDST` with `--Except PID.MDST`: all µDST files except PID.MDST
- `/LHCb/Collision12////BHADRON.MDST` : all existing BHADRON.DST files from 2012 data (i.e. all processings and strippings)
- `/LHCb/Collision12//RealData/91000000/RAW` : all Express stream RAW data of 2012
- `/MC/2012/Beam4000GeV-2012-MagDown-Nu2.5-Pythia6/Sim08/Digi13/Trig0x409f0045/Reco14/Strippi` : a long list of event types of MC

Information on files

dirac-dms-lfn-replicas

Shows where files have replicas and reports the SURL registered in the file catalog

```
$ dirac-dms-lfn-replicas --help

    Show replicas for a (set of) LFNs

Usage:
  dirac-dms-lfn-replicas [option|cfgfile] [<LFN>] [<LFN>...]
General options:
  -o:  --option=         : Option=value to add
  -s:  --section=        : Set base section for relative parsed options
  -c:  --cert=           : Use server certificate to connect to Core Services
  -d   --debug           : Set debug mode (-dd is extra debug)
  -h   --help            : Shows this help

Options:
  -    --File=           : File containing list of LFNs
  -l:  --LFNs=           : List of LFNs (comma separated)
  -    --Terminal        : LFNs are entered from stdin (--File /dev/stdin)
  -   --LastLFNs              : Use last set of LFNs
  -a   --All             :   Also show inactive replicas
```

dirac-dms-lfn-metadata

Shows metadata of a logical file, i.e. not related to any physical replica.

```
[localhost, Patch] ~ $ dirac-dms-lfn-metadata --help
  Get the metadata of a (list of) LFNs from the FC
Usage:
  dirac-dms-lfn-metadata [option|cfgfile] ... [LFN[,LFN2[,LFN3...]]]
Arguments:
  LFN:      Logical File Name or file containing LFNs
General options:
  -o:  --option=         : Option=value to add
  -s:  --section=        : Set base section for relative parsed options
  -c:  --cert=           : Use server certificate to connect to Core Services
  -d   --debug           : Set debug mode (-dd is extra debug)
  -h   --help            : Shows this help

Options:
  -    --File=           : File containing list of LFNs
  -l:  --LFNs=           : List of LFNs (comma separated)
  -    --Terminal        : LFNs are entered from stdin (--File /dev/stdin)
  -   --LastLFNs              : Use last set of LFNs
  -g:  --Sites=          :   Sites to consider [ALL] (comma separated list)
  -S:  --SEs=            :   SEs to consider [ALL] (comma separated list)
```

dirac-dms-pfn-metadata

Shows the metadata of physical replica(s), possibly at selected SEs/Sites.

```
[localhost, Patch] ~ $ dirac-dms-pfn-metadata  --help

  Gets the metadata of a (list of) LHCb LFNs/PFNs given a valid DIRAC SE.
  Only the LFN contained in the PFN is considered, unlike the DIRAC similar script

Usage:
  dirac-dms-pfn-metadata [option|cfgfile] ... [URL[,URL2[,URL3...]]] SE[ SE2...]
```

```
Arguments:
  URL:      Logical/Physical File Name or file containing URLs
  SE:       Valid DIRAC SE
General options:
  -o:  --option=         : Option=value to add
  -s:  --section=        : Set base section for relative parsed options
  -c:  --cert=           : Use server certificate to connect to Core Services
  -d   --debug           : Set debug mode (-dd is extra debug)
  -h   --help            : Shows this help

Options:
  -    --File=           : File containing list of LFNs
  -l:  --LFNs=           : List of LFNs (comma separated)
  -    --Terminal        : LFNs are entered from stdin (--File /dev/stdin)
  -   --LastLFNs              : Use last set of LFNs
  -g:  --Sites=          :   Sites to consider [ALL] (comma separated list)
  -S:  --SEs=            :   SEs to consider [ALL] (comma separated list)
  -    --Check           :    Checks the PFN metadata vs LFN metadata
  -    --Exists          :    Only reports if the file exists
```

For backward compatibility the list of URLs and SEs can be provided as positional arguments, but it is recommended to use option switches. If a site is selected, the metadata of all replicas at that site are reported. If no SE and no sites are specified, the metadata of all replicas is reported.

dirac-dms-pfn-exists

Shortcut for `dirac-dms-pfn-metadata --Exists`

```
$ dirac-dms-pfn-exists --help

Check existence of a (list of) LHCb LFNs/PFNs given a valid DIRAC SE (or for all replicas)
  Only the LFN contained in the PFN is considered, unlike the DIRAC similar script

Usage:
  dirac-dms-pfn-exists [option|cfgfile] ... [URL[,URL2[,URL3...]]] SE[ SE2...]
Arguments:
  URL:      Logical/Physical File Name or file containing URLs
  SE:       Valid DIRAC SE

General options:
  -o  --option <value>         : Option=value to add
  -s  --section <value>        : Set base section for relative parsed options
  -c  --cert <value>           : Use server certificate to connect to Core Services
  -d  --debug                  : Set debug mode (-ddd is extra debug)
  -   --autoreload             : Automatically restart if there's any change in the module
  -   --license                : Show DIRAC's LICENSE
  -h  --help                   : Shows this help

Options:
  -   --File=                 : File containing list of LFNs
  -l  --LFNs <value>           : List of LFNs (comma separated)
  -   --Terminal               : LFNs are entered from stdin (--File /dev/stdin)
  -   --LastLFNs               : Use last set of LFNs
  -g  --Sites <value>          :   Sites to consider [ALL] (comma separated list)
  -S  --SEs <value>            :   SEs to consider [ALL] (comma separated list)
  -   --Summary                :    Only prints a summary on existing files
```

dirac-dms-user-lfns

This command allows a user to get a list of all files they own on the Grid, within directories or with data/time criteria.

```
$ dirac-dms-user-lfns --help
```

```
Get the list of all the user files.
Usage:
  dirac-dms-user-lfns [option|cfgfile] ...
General options:
  -o:  --option=          : Option=value to add
  -s:  --section=         : Set base section for relative parsed options
  -c:  --cert=            : Use server certificate to connect to Core Services
  -d   --debug            : Set debug mode (-dd is extra debug)
  -h   --help             : Shows this help

Options:
  -D:  --Days=            : Match files older than number of days [0]
  -M:  --Months=          : Match files older than number of months [0]
  -Y:  --Years=           : Match files older than number of years [0]
  -w:  --Wildcard=        : Wildcard for matching filenames [*]
  -b:  --BaseDir=         : Base directory to begin search (default /[vo]/user/[initial]/[username
  -e   --EmptyDirs        : Create a list of empty directories
```

Accessing files

dirac-dms-lfn-accessURL

This command returns tURLs for files at a given SE or site. This tURL is optimised for fast and efficient access to the data. Note that any other tURL may either be inefficient or its usage may be discontinued at some point. One can select for which protocol one wants to get a URL using `--Protocol=` (default is `xrootd`). In case the file is on tape storage, the tURL can currently only be obtained once the file has been staged, therefore the command may in this case timeout.

```
[localhost] ~ $ dirac-dms-lfn-accessURL --help

Retrieve an access URL for an LFN replica given a valid DIRAC SE.

Usage:
  dirac-dms-lfn-accessURL [option|cfgfile] ... [LFN[,LFN2[,LFN3...]]] SE[,SE2...]
Arguments:
  LFN:      Logical File Name or file containing LFNs
  SE:       Valid DIRAC SE

General options:
  -o  --option <value>         : Option=value to add
  -s  --section <value>        : Set base section for relative parsed options
  -c  --cert <value>           : Use server certificate to connect to Core Services
  -d  --debug                  : Set debug mode (-ddd is extra debug)
  -   --autoreload             : Automatically restart if there's any change in the module
  -   --license                : Show DIRAC's LICENSE
  -h  --help                   : Shows this help

Options:
  -   --File=                 : File containing list of LFNs
  -l  --LFNs <value>           : List of LFNs (comma separated)
  -   --Terminal               : LFNs are entered from stdin (--File /dev/stdin)
  -   --LastLFNs               : Use last set of LFNs
  -g  --Sites <value>          :   Sites to consider [ALL] (comma separated list)
  -S  --SEs <value>            :   SEs to consider [ALL] (comma separated list)
  -   --Protocol=              :    Define the protocol for which a tURL is requested (default:ro
```

Positional arguments are supported for backward compatibility, but option switches are highly recommended.
If no SE is specified, all allowed SEs are used.

dirac-dms-get-file

This command copies a file or a set of files to the local directory. The options of this command are going to evolve soon, giving additional possibilities.

```
[localhost, Certif] ~ $ dirac-dms-get-file --help

  Retrieve a single file or list of files from Grid storage to the current directory.

Usage:
  dirac-dms-get-file [option|cfgfile] [<LFN>] [<LFN>...]
General options:
  -o:  --option=          : Option=value to add
  -s:  --section=         : Set base section for relative parsed options
  -c:  --cert=            : Use server certificate to connect to Core Services
  -d   --debug            : Set debug mode (-dd is extra debug)
  -h   --help             : Shows this help

Options:
  -    --File=            : File containing list of LFNs
  -l:  --LFNs=            : List of LFNs (comma separated)
  -    --Terminal         : LFNs are entered from stdin (--File /dev/stdin)
  -   --LastLFNs               : Use last set of LFNs
  -D:  --Directory=       :    Directory to download to (default = /home/phicharp)
```

Replicating files

dirac-dms-replicate-lfn

Replicates an LFN to a (list of) SE. It requires write access to the FC directory.

```
[localhost, Certif] ~ $ dirac-dms-replicate-lfn --help

  Replicate a (list of) existing LFN(s) to (set of) Storage Element(s)

Usage:
  dirac-dms-replicate-lfn [option|cfgfile] ...  [LFN1[,LFN2,[...]]] Dest[,Dest2[,...]] [Source [C
Arguments:
  Dest:     Valid DIRAC SE(s)
  Source:   Valid DIRAC SE
  Cache:    Local directory to be used as cache
General options:
  -o:  --option=          : Option=value to add
  -s:  --section=         : Set base section for relative parsed options
  -c:  --cert=            : Use server certificate to connect to Core Services
  -d   --debug            : Set debug mode (-dd is extra debug)
  -h   --help             : Shows this help

Options:
  -    --File=            : File containing list of LFNs
  -l:  --LFNs=            : List of LFNs (comma separated)
  -    --Terminal         : LFNs are entered from stdin (--File /dev/stdin)
  -   --LastLFNs               : Use last set of LFNs
```

dirac-dms-add-file

Uploads to a Grid storage element and registers in the file catalog a local file. In order to properly register the GUID of the file as the one contained in the file itself, it is mandatory to setup the ROOT environment as well. Easiest way is:

```
$ lb-run -c best LHCbDirac/prod [bash --norc|tcsh|...]
```

If you are on lxplus or on a machine that has access to EOS or Castor at CERN, the "local file" can be a Castor file (in the form `/castor/cern.ch/...`) or an EOS file (in the form `/eos/lhcb/...`), of course not yet a Grid file.

The LFN for a user file must be in the form
`/lhcb/user/<initial>/<username>/whatever-you-like-as-a-path`.

Command usage:

```
dirac-dms-add-file --help

Upload a file to the grid storage and register it in the File Catalog
Usage:
  dirac-dms-add-file [option|cfgfile] ... LFN Path SE [GUID]
Arguments:
  LFN:      Logical File Name
  Path:     Local path of the file
  SE:       DIRAC Storage Element
  GUID:     GUID to use in the registration (optional)

 ++ OR ++

Usage:
  dirac-dms-add-file [option|cfgfile] ... LocalFile
Arguments:
  LocalFile: Path to local file containing all the above, i.e.:
  lfn1 localfile1 SE [GUID1]
  lfn2 localfile2 SE [GUID2]

General options:
  -o  --option <value>         : Option=value to add
  -s  --section <value>        : Set base section for relative parsed options
  -c  --cert <value>           : Use server certificate to connect to Core Services
  -d  --debug                  : Set debug mode (-dd is extra debug)
  -h  --help                   : Shows this help
```

GUID-related issues

Gaudi requires when reading a file through a catalog that the GUID in the file catalog matches the GUID (a.k.a. FID) stored in the file. If the file was uploaded to the Grid and registered without care, it may be that this is not the case. It can be checked and eventually fixed using the following commands:

dirac-lhcb-get-root-guid

This script will read the file GUID from the actual replica.

```
[localhost, PatchFull] ~ $ dirac-lhcb-get-root-guid --help

Get the GUID of a (set of) ROOT file
The file can be either local, an LFN or an xrootd URL (root:...)

Usage:
  dirac-lhcb-get-root-guid [option|cfgfile] file1 [file2 ...]

General options:
  -o  --option <value>         : Option=value to add
  -s  --section <value>        : Set base section for relative parsed options
  -c  --cert <value>           : Use server certificate to connect to Core Services
  -d  --debug                  : Set debug mode (-dd is extra debug)
  -h  --help                   : Shows this help
```

dirac-lhcb-fix-file-guid

This script will read the file GUID from the actual file and update the file catalog if needed with the correct GUID.

```
[localhost, PatchFull] ~ $ dirac-lhcb-fix-file-guid --help

Fix incorrect file GUIDs

Usage:
  dirac-lhcb-fix-file-guid [option|cfgfile] [OldLFN]

General options:
  -o  --option <value>         : Option=value to add
  -s  --section <value>        : Set base section for relative parsed options
  -c  --cert <value>           : Use server certificate to connect to Core Services
  -d  --debug                  : Set debug mode (-dd is extra debug)
  -h  --help                   : Shows this help

Options:

  -f  --OldLFN <value>         : LFN of existing file to be fixed.

  -n  --NewLFN <value>         : Optional: specify a new LFN for the file (retaining the existing

  -D  --Directory <value>      : Optional: directory to download file (defaults to TMPDIR then PW

  -k  --Keep                   : Optional: specify this switch to retain the local copy of the do

  -m  --SafeMode               : Optional: specify this switch to run the script in safe mode (wi
```

Removing replicas or files

dirac-dms-remove-replicas

Allows to remove a replica at a given SE. This is only allowed for authorised files of course. Note also that it is not possible to remove the last replica of a file. Use `dirac-dms-remove-files` instead.

```
[localhost, Certif] ~ $ dirac-dms-remove-replicas --help

  Remove the given file replica or a list of file replicas from the File Catalog
  and from the storage.

  Usage:
     dirac-dms-remove-replicas <LFN | fileContainingLFNs> SE [SE]

General options:
  -o:  --option=          : Option=value to add
  -s:  --section=         : Set base section for relative parsed options
  -c:  --cert=            : Use server certificate to connect to Core Services
  -d   --debug            : Set debug mode (-dd is extra debug)
  -h   --help             : Shows this help

Options:
  -    --File=            : File containing list of LFNs
  -l:  --LFNs=            : List of LFNs (comma separated)
  -    --Terminal         : LFNs are entered from stdin (--File /dev/stdin)
  -   --LastLFNs               : Use last set of LFNs
  -g:  --Sites=           :   Sites to consider [ALL] (comma separated list)
  -S:  --SEs=             :   SEs to consider [ALL] (comma separated list)
  -v   --Verbose          :  use this option for verbose output [False]
  -n   --NoLFC            :  use this option to force the removal from storage of replicas not in
```

The option `--NoLFC` allows to remove replicas even though they are not present in the LFC (dark data). In this case an attempt is made to remove the file from physical storage only.

dirac-dms-remove-files

Allows to remove all replicas of a (list of) file(s) as well as the entry in the file catalog.

```
 [localhost, Certif] ~ $ dirac-dms-remove-files --help

  Remove the given file or a list of files from the File Catalog and from the storage

Usage:
  dirac-dms-remove-files [option|cfgfile] [<LFN>] [<LFN>...]
General options:
  -o:  --option=          : Option=value to add
  -s:  --section=         : Set base section for relative parsed options
  -c:  --cert=            : Use server certificate to connect to Core Services
  -d   --debug            : Set debug mode (-dd is extra debug)
  -h   --help             : Shows this help

Options:
  -P:  --Productions=     :    Production ID to search (comma separated list)
  -f:  --FileType=        :    File type (comma separated list, to be used with --Production) [All
  -    --ExceptFileType=  :    Exclude the (list of) file types when all are requested
  -B:  --BKQuery=         :    Bookkeeping query path
  -r:  --Runs=            :    Run or range of runs (r1:r2)
  -    --DQFlags=         :    DQ flag used in query
  -    --StartDate=       :    Start date for the BK query
  -    --EndDate=         :    End date for the BK query
  -    --Invisible        :    See also invisible files
  -    --File=            : File containing list of LFNs
  -l:  --LFNs=            : List of LFNs (comma separated)
  -    --Terminal         : LFNs are entered from stdin (--File /dev/stdin)
  -   --LastLFNs               : Use last set of LFNs
  -    --SetProcessed     :   Forced to set Removed the files in status Processed (default:not res
```

The list of files may be the result of a Bookkeeping query. This is only allowed for data managers and should be used with great care.

If files are used by some transformations, they are set to "Removed" status unless their current status is "Processed". "Processed" files can be forced to be set "Removed" using the switch =--SetProcessed+.

Bookkeeping commands

They can be found here.

This topic: LHCb > LHCbDiracCLI
Topic revision: r12 - 2019-07-05 - PhilippeCharpentier