

## Locating Files on the Grid

All files produced by either ILD, SID or CLIC are stored in the DiracFileCatalog. This allows for several data operations like searching for files, getting them, removing them, etc. There are several interfaces available: The Command Line Utility, the web interface, the Python API, and a script.

### The `dirac-ilc-find-in-FC` script

Besides the `dirac-dms-user-lfns` script, there is another script that used metadata to print a list of files. The syntax is the following:

```
dirac-ilc-find-in-FC /ilc/prod/ EvtClass=higgs_ffh Datatype=DST-MERGED
```

The output of this can be redirected to a text file for easy manipulation later.

### The Command Line Interface

It is invoked using

```
dirac-dms-filecatalog-cli
```

Once started you'll get a shell that looks like this:

```
Starting FileCatalog client
File Catalog Client $Revision: 1.1 $Date:
FC: />
```

Here, you can get help by typing `help`:

```
FC: /> help
Documented commands (type help <topic>):
=====
add          chgrp      exit      guid      meta      register  rm         stats
ancestor    chmod     find      id        mkdir     repair    rmdir     unregister
ancestorset chown     get       lcd       pwd       replicas  rmreplica user
cd           descendent group     ls        rebuild   replicate size

Undocumented commands:
=====
help

FC: />
```

As you can see, you can get help with

```
FC: /> help meta
Metadata related operations
```

Usage:

```
meta index [-d|-f|-r] <metaname> [<metatype>] - add new metadata index. Possible types
                                                'int', 'float', 'string', 'date';
-d directory metadata
-f file metadata
-r remove the specified metadata index

meta set <path> <metaname> <metavalue> - set metadata value for directory or file
meta remove <path> <metaname> - remove metadata value for directory or file
meta get [-e] [<path>] - get metadata for the given directory or file
meta tags <path> <metaname> where <meta_selection> - get values (tags) of the given met
```

the metadata selection

meta show - show all defined metadata indice

FC:/&gt;

⚠ Some commands are considered unsafe like the rm and rmdir and should not be used. As this system is being improved, this will be made safe.

From the help message above, one can already see what can be done with data manipulation. In particular, it's possible to set searchable index using the first command. I do not recommend this to everyone, as it should be left to production managers or service managers. It is also possible to set meta data values for directories and files using meta set. If a metadata is not part of the searchable fields, it will be set as documentation. Getting the searchable fields is done with

FC:/&gt; meta show

```
FileMetaFields : {'PolarizationB2': 'VARCHAR(128)', 'GenProcessID': 'INT', 'BeamParticle1':
'BeamParticle2': 'VARCHAR(128)', 'PolarizationB1': 'VARCHAR(128)'}
DirectoryMetaFields : {'EvtType': 'VARCHAR(128)', 'NumberOfEvents': 'int', 'BXoverlaid': 'int',
'Datatype': 'VARCHAR(128)', 'Energy': 'VARCHAR(128)', 'MachineParams': 'VARCHAR(128)', 'DetectorT
'Machine': 'VARCHAR(128)', 'EvtClass': 'VARCHAR(128)', 'Owner': 'VARCHAR(128)', 'SoftwareTag': 'V
'DetectorModel': 'VARCHAR(128)', 'JobType': 'VARCHAR(128)', 'ProdID': 'int'}
```

The output is a bit ugly but will be improved.

It is also possible to get all values for a meta tag (⚠ for directory level tags only for the moment) using

FC:/&gt; meta tags /ilc EvtClass

Possible values for EvtClass:

```
1f
1f_3f
2f
2f_Z_bhabhag
2f_Z_hadronic
2f_Z_leptonic
3f
4f
4f_singleW_leptonic
4f_singleW_semibleptonic
4f_singleZee_leptonic
4f_singleZee_semibleptonic
4f_singleZnu_nu_leptonic
4f_singleZnu_nu_semibleptonic
4f_singleZsingleWMix_leptonic
4f_WW_hadronic
4f_WW_leptonic
4f_WW_semibleptonic
4f_ZZWWMix_hadronic
4f_ZZWWMix_leptonic
4f_ZZ_hadronic
4f_ZZ_leptonic
4f_ZZ_semibleptonic
5f
6f
6f_eeWW
6f_llWW
6f_ttbar
6f_vvWW
6f_xxWW
6f_xxxxZ
6f_yyyyZ
aa_2f
aa_4f
aa_lowpt
```



The `.` indicates that one as to look for compatible files in the current directory, and can be replaced by a path, e.g `/ilc/prod/ilc/mc-dbd/ild` to make sure the files only come from this directory and its sub-directories.

It's always possible to have access to the defined meta data of a directory:

```
FC:> meta get /ilc/prod/ilc/mc-dbd/ild/dst-merged/1000-B1b_ws/higgs_ffh/ILD_o1_v05/v01-16-p03/
      *Datatype : DST-MERGED
      *Energy : 1tev
      *MachineParams : B1b_ws
      *DetectorType : ILD
      *Machine : ilc
      *EvtClass : higgs_ffh
      !SoftwareTag : v01-16-p03
      *DetectorModel : ILD_o1_v05
FC:>
```

Where `*` indicate inherited meta data and `!` mean local meta data. Meta data that have no indication are non searchable.

Of course, getting the meta data of a file is possible:

```
FC:> meta get /ilc/prod/ilc/mc-dbd/ild/dst-merged/1000-B1b_ws/higgs_ffh/ILD_o1_v05/v01-16-p03/rv01-16-p03.sv01-16-p03
      NumberOfEvents : 1000
      RandomSeed : None
      BeamParticle1 : e1
      BeamParticle2 : E1
      GenProcessID : 37594
      StartEvent : 0
      PolarizationB1 : L
      PolarizationB2 : R
FC:>
```

Here, there are no indication (for the moment) of the nature of the meta data (searchable or not) and the user should rely on the `meta show` output.

A last functionality worth advertising here is the ancestor/daughter relationships. For this, the following examples should be enough.

```
FC:> ancestor /ilc/prod/ilc/mc-dbd/ild/dst-merged/1000-B1b_ws/higgs_ffh/ILD_o1_v05/v01-16-p03/rv01-16-p03.sv01-16-p03
1 /ilc/prod/ilc/mc-dbd/ild/dst-merged/1000-B1b_ws/higgs_ffh/ILD_o1_v05/v01-16-p03/rv01-16-p03.sv01-16-p03
1 /ilc/prod/ilc/mc-dbd/ild/dst/1000-B1b_ws/higgs_ffh/ILD_o1_v05/v01-16-p03/rv01-16-p03.sv01-16-p03
1 /ilc/prod/ilc/mc-dbd/ild/dst/1000-B1b_ws/higgs_ffh/ILD_o1_v05/v01-16-p03/rv01-16-p03.sv01-16-p03
```

This gives the direct parents.

The descendants of a file can be also obtained:

```
FC:> descendent /ilc/prod/ilc/mc-dbd/ild/dst/1000-B1b_ws/higgs_ffh/ILD_o1_v05/v01-16-p03/rv01-16-p03.sv01-16-p03
1 /ilc/prod/ilc/mc-dbd/ild/dst-merged/1000-B1b_ws/higgs_ffh/ILD_o1_v05/v01-16-p03/rv01-16-p03.sv01-14-01-p03
```

There are other useful utilities in this interface that should be looked upon, but the `help` is enough to grasp the concepts. In particular, users can get their files using `get`.

## The web portal

There is an interface to the File Catalog on the DIRAC web portal at [https://ilcdirc.cern.ch/DIRAC/ILC-Production/ilc\\_user/data/MetaCatalog/display](https://ilcdirc.cern.ch/DIRAC/ILC-Production/ilc_user/data/MetaCatalog/display) but it's still in

development and does not allow easy manipulation of data as the CLI permits. It's still possible to interact with it to get file informations.

This interface will be more detailed once it's fully fonctionnal. Until then, people should get the files using

## The Python API

This interface uses directly the underlying python API to get the information. It's to be used when submitting jobs as it's the easiest way of passign the files from the catalog to the job definition. I only demonstrate here the mean to perform a data query.

```
from DIRAC.Core.Base import Script
Script.parseCommandLine()
from DIRAC.Resources.Catalog.FileCatalogClient import FileCatalogClient

fc = FileCatalogClient()

meta = {}
meta['EvtClass'] = 'higgs_ffh'
meta['Datatype'] = 'DST-MERGED'

res = fc.findFilesByMetadata(meta)
if not res['OK']:
    print res['Message']

lfns = res['Value']

print "Found %s files" % len(lfns)
for lfn in lfns:
    print lfn
```

This will print on screen the result of the meta data query indicated by the dictionary `meta`. This is typically used when defining a job: its `setInputSandbox` would contain `"LFN: "+lfn` to tell DIRAC to get the file before running the job.

-- AndreSailer - 21 Feb 2014 Moved from DiracForUsers

---

This topic: CLIC > IlcdircObtainLFNs

Topic revision: r1 - 2014-02-21 - AndreSailer



Copyright &© 2008-2020 by the contributing authors. All material on this collaboration platform is the property of the contributing authors.

Ideas, requests, problems regarding TWiki? Send feedback