

Ltools- Simple user tools based on the lcg- 2 software



provided and maintained by ZIB

User Guide

Version 1.4

Stefan Wollny
Hinnerk Stüben

10.May 2006

1. Ltools – Overview.....	3
2. Installation of Ltools.....	4
2.1 Mandatory Installation Steps.....	4
2.2. Optional Installation Steps	4
3. Ltools-commands in detail.....	5
3.1. lget.....	5
3.1.1. What lget does when it is started.....	5
3.1.2. Getting a configuration binary.....	5
3.1.3. Getting metadata of a configuration.....	6
3.1.3. Getting metadata of an ensemble.....	6
3.1.4. Mixing Options of lget.....	6
3.1.5. Example: Output of download of a configuration binary to be saved in / tmp/binary_example.....	7
3.2. lput.....	7
3.2.1. Overview of lput.....	7
3.2.2. Uploading binary and metadata.....	7
3.2.3. Registering a file that is already in an SE.....	8
3.2.4. Using or overriding the default path within the SE.....	9
3.3. lls.....	9
3.3.1. Showing the configurations of an ensemble.....	9
3.3.2. Showing all configurations in the MDC.....	9

3.3.3. Showing access information	9
3.3.4. Shortening the command using a prefix from the configuration file.....	10
3.3.5. Example: (Shortened) Output of presentation of all configurations of an ensemble:.....	10
3.4. linit.....	10
3.4.1. Creating a new ensemble-entry in the MDC.....	10
3.5. lupdate.....	11
3.5.1. Updating metadata in the MDC.....	11
3.5.2. Example Output of updating a configuration's metadata.....	11
3.6. lvalidate.....	11
3.6.1. Validating a file to be valid QCDML.....	11
3.6.2. Example: Output of validation of the metadata description of an ensemble..	12
3.7. ladm	12
3.7.1. Purpose of ladm	12
3.7.2. Synopsis of ladm	12
3.8. lid	13
3.8.1. Purpose of lid	13
3.8.2. Synopsis of lid.....	13
4. The Configuration File	14
4.1. Options for the main program behaviour.....	14
4.2. Grid/ILDG Defaults.....	14
4.2.1. Default VO.....	14
4.2.2. Default Ensemble prefix for Downloading	14
4.2.3. Default LFN prefix for Downloading	15
4.2.4. Default UPLOAD Storage element.....	15
5. Man pages.....	16
5.1. man page lget.....	16
5.2. man page lput.....	18
5.3. man page lls.....	20
5.4. man page linit.....	22
5.5. man page lupdate.....	23
5.6. man page lvalidate.....	24
5.7. man page ladm.....	25
5.7. man page lid.....	28
6. Typical Itools session.....	30

1. Ltools – Overview

The Ltools are a set of commands for using the LCG-2 software (see <http://lcg.web.cern.ch/LCG/> for information about the LHC computing grid project) in a more convenient and less error-prone way and also for accessing the central metadata-catalogue of a community. Although developed for use within the German Lattice Data Grid / International Lattice Data Grid (see <http://www-zeuthen.desy.de/latfor/ldg/>), these commands may be easily adopted to be used within other projects or other VOs too. They are intended to provide a simple usage for a simple, maybe even inexperienced user. That is, these commands shall to the greatest possible extent

- be tailored for easy usage within a specific scientific community
- offer a convenient access and hide the complexity of the underlying Grid, that means only a few necessary commands are presented that are easy to remember (of course more advanced users might as well still use lower-level commands for full functionality)
- prevent the user from making mistakes or producing inconsistencies
- be secure and reliable, therefore based on existing proven (sub-) commands
- ensure that there is always corresponding consistent metadata for uploaded binaries (this is done by "lputting" binary and metadata using one command and one transaction)

The commands are built upon the lcg-commands and also special client-software developed by DESY. Everything is bundled in the “LDG-software” and can be downloaded from the LDG-website (only a valid certificate is needed additionally). Usage of lcg-commands directly is not recommended, because it could lead to inconsistencies, e.g. having a binary in a SE but nothing in the MDC.

If needed, the ltools-commands will check whether a valid proxy exists, and initiate grid-proxy-init if not. All commands will look for a configuration file (\$HOME/.ltools). If this exists, the default VO among other things are defined to make usage more convenient. If not, usefull defaults are used or needed informations have to be given via command-line option.

This should lead to two advantages in real-world-use: firstly, the user should only need to type minimal information in standard commands (e.g. it is not necessary to specify the vo every time, although, of course, there is an option to specify a different vo than the default if needed). Secondly, tailored configuration files can be prepared by administrators for certain user groups for ease of use.

2. Installation of Ltools

At the moment, the ltools can be used on typical Linux-systems (like Scientific Linux, Suse Linux, Fedora Core). Ltools now are part of the LDG-software, which also includes all necessary lcg-2 software.

2.1 Mandatory Installation Steps

Install the LDG-software, which includes the ltools.

Download and Installation Information :

<http://www-zeuthen.desy.de/latfor/ldg/doc/> (see Software Installation).

2.2. Optional Installation Steps

You might copy the configuration file template.ltools from $\$LROOT/opt/ltools<Version>$ ¹ to your home $\$HOME/.ltools$ and modify it according to your needs (defaults and examples are proposed within that file).

If a certain value is not defined here, a useful system-default will be used.

¹ $\$LROOT$ is the installation path of LDG-software, see “Software Installation” on <http://www-zeuthen.desy.de/latfor/ldg/doc/>

3. Ltools-commands in detail

3.1. lget

3.1.1. What lget does when it is started

The first thing that lget is checking (unless you only asked for help using `-h / --help` or the version `--version`) is whether a valid grid proxy exists. Therefore the first two lines you will see (unless you told lget to be quite by putting `-q/--quite` or set `always_be_quite` to “yes” in the configuration file) are:

```
Welcome to the Ltool-command lget -  
Testing grid-proxy-init
```

If you do not have already a valid proxy, lget will try to initiate one by launching “grid-proxy-init” and you will see (hopefully) something like:

```
Trying to start grid-proxy-init...  
Your identity: /O=GermanGrid/OU=ZIB/CN=Stefan Wollny  
Enter GRID pass phrase for this identity:  
Creating proxy ..... Done  
Your proxy is valid until: Fri Apr 29 02:02:24 2005  
Proxy started ...  
grid-proxy...ok
```

If a valid proxy exists or could be started, lget will check next, whether a filename already exists that would be overwritten by the execution of the current command. If so, lget will stop with the message: “*Fatal Error: File exists. Use -f or --force to override !*”. So you should repeat the command with an additional `-f` or you might put “`default_overwrite`” to “yes” in the configuration file and lget will act like `-f` was given every time. If everything works fine, the last line you will see should be

```
Checking nonzero size of downloaded File ...ok.
```

The typical use cases are explained below; please see the man page for all options of lget.

3.1.2. Getting a configuration binary

The simplest way to use the lget command is to download a configuration binary by simply typing

```
lget <Logical Filename> .
```

This will get the configuration from any Storage-Element using the Replica Location

Service from the lcg-software². Executed like this, the binary will be saved in the current working directory in a file named as the Logical Filename (if the Logical Filename consists of slashes, than only the last part will be used).

If you want to save the binary in a different directory, then simply type

```
lget <Logical Filename> <my_path>.
```

Or, to save it with a different name, type

```
lget <Logical Filename> <Filename>      (current directory)
or   lget <Logical Filename> <Path/Filename>
```

3.1.3. Getting metadata of a configuration

If you do not want to download the configuration binary, but the metadata description of that configuration, simply add a “-m” flag to the command.

```
lget -m <Logical Filename>
```

and you will get a file <Logical Filename>.xml in your current working directory with the metadata (of course you can again specify a local filename and/or different directory like it is shown in 3.1.2.).

3.1.3. Getting metadata of an ensemble

If we are not talking about a configuration, but an ensemble please add an additional “-e” to get the metadata description of that ensemble (use the Markov-Chain-URI to specify the ensemble).

```
lget -m -e <MarkovChainURI>
```

3.1.4. Mixing Options of lget

Different options to lget can be given in a single string whenever it makes sense. This also holds for all other ltools commands.

E.g.

```
lget -mef <MarkovChainURI>
```

is exactly the same as typing

² the command that is issued is “lcg-cp lfn:<Logical Filename> --vo <vo> file:<ResultFilename>” where the <vo> is read from the configuration file \$HOME/.ltools (see Chapter 4. The Configuration File) and the <ResultFilename> is the current working directory plus the last part of the Logical Filename (LFN) if that is consisting of several slashes or whole LFN if no slashes occur.

```
lget -m -e -f <MarkovChainURI>
```

3.1.5. Example: Output of download of a configuration binary to be saved in /tmp/binary_example

```
~> lget qcdfs_b5p40kp13610-24x48_bqcd.561.1.1.00125.tar /tmp/binary_example
```

```
Welcome to the Ltool-command lget -  
Testing grid-proxy-init:  
grid-proxy...ok
```

```
Trying to get binary ...  
Virtual Organisation is ildg  
Executing lcg-cp lfn:qcdfs_b5p40kp13610-24x48_bqcd.561.1.1.00125.tar --vo  
ildg file:/tmp/binary_example
```

```
Checking nonzero size of downloaded File ...ok.  
~>
```

3.2. lput

3.2.1. Overview of lput

lput is the command to upload a binary to a so called Storage Element (SE) and the corresponding metadata to the Metadata-Catalogue (MDC). It is not allowed to place a binary without the corresponding (syntactically-)correct metadata, because the MDC and the data in the SE(s) should be consistent. Therefore, the operation will take place in one transaction, that is, all the data is stored by successfully finishing the operation or nothing will be stored if the operation fails. Configuration files will be evaluated to be correct QCDML, otherwise, they will be rejected and the transaction will fail.

“lputting” a configuration requires adequate access rights. Also, the corresponding ensemble must already exist in the MDC (see “linit” for initialising an ensemble).

The typical use case is explained below; please see the man page for all options of lput.

3.2.2. Uploading binary and metadata

In order to make the upload as convenient as possible, some assumptions/convention are made. E.g., if the binary file is saved in the current working directory and has as the filename the Logical Filename (LFN) or, if the LFN consists of slashes, the last part of the LFN, you simply need to type

```
lput <Metadatafile>
```

to upload the metadata to the MDC and upload the binary to the SE defined in the configuration file (see Chap. 4).

If the binary is located elsewhere, you have to give the absolute or relative path as additional parameter:

```
lput <Metadatafile> <Binaryfile>.
```

If the binary is located in a directory different than the current working directory, but is named as LFN (or last part of LFN), you can omit the filename and simply type the path to that directory:

```
lput <Metadatafile> <Path_to_directory>
```

Example: Output of upload of a configurations metadata and binary:

```
~> lput upload-test-conf.xml
```

```
Welcome to the Ltool-command lput -  
Testing grid-proxy-init  
grid-proxy...ok
```

```
Virtual Organisation is ildg  
Storage Element is dcache.zib.de  
Looking, whether ensemble www.lqcd.org/ildg/Testing/qcdsf/b50011100048 exists  
in MDC: Ok
```

```
MDC says:  
- Getting default credential  
- Authorization: HOST  
- Logging is working  
- Date: Thu Sep 01 15:11:26 CEST 2005  
- Version: Apache-XML-Security-J 1.0.4  
- Response Status: Ok.
```

```
...MDC Operation ok.
```

```
Trying to upload binary ...
```

```
Executing lcg-cr --vo ildg -l lfn:test_1_Sep_no2  
file:/home/wollny/Testing/test_1_Sep_no2 -d  
sfn://dcache.zib.de/pnfs/zib.de/data/ildg/Testing/qcdsf/b50011100048/test_1_Sep_no2  
guid:c876b918-91ca-473b-aa6b-542b73035254
```

```
Data successfully copied to Storage Element !  
LFN is: test_1_Sep_no2
```

3.2.3. Registering a file that is already in an SE

If it happens that you have a binary file already located in a SE but not registered in

the file-catalogue, you can give the SRM-path to that file as the second argument.

```
lput <Metadata> srm://Path/to/binary
```

In this case, the metadata will be uploaded as usual to the MDC and the corresponding binary in the SE will be registered in the file-catalogue. Concerning the access via lget there is no difference to an upload from a local filesystem.

3.2.4. Using or overriding the default path within the SE

Normally, lput will try to save the binary in a useful path in the given SE for convenient uploads. This is

```
srm://<SE>/<SE-specific-path>/<Collaboration>/<Project>/<Ensemble>/<Last_Part_of_LFN>
```

where <SE> is the Storage-Element (you might define a default SE for uploading in your .ltools file or give it with the cmd-line option --se), <SE-specific-path> is a system-specific part that is known for all SEs by lput and <Collaboration>/<Project>/<Ensemble>/<Last_Part_of_LFN> are taken from the metadata.

If you do not want this for any reason, you can specify the complete SRM-address for the data to be stored within the SE by putting

```
--path <SRM-Address>    and data will not be uploaded to default but to given path.
```

Specify full SRM-address!

3.3. lls

3.3.1. Showing the configurations of an ensemble

The lls command is used to list all configurations of an ensemble. You need to know the Markov-Chain-URI of that ensemble. LFNs are sorted alphabetically. To see all configurations of an ensemble simply type

```
lls <Markov-Chain-URI>
```

3.3.2. Showing all configurations in the MDC

If want to see all ensembles currently known by the MDC, just type:

```
lls --all
```

3.3.3. Showing access information

From version 1.4. on, access information is shown by putting “-l” (long). This will show the access-rights for a single ensemble by typing lls -l <Ensemble> or for all

ensembles by typing `lls --all -l` (see example in chap. 6).

3.3.4. Shortening the command using a prefix from the configuration file

You might also specify a prefix in your `.ltools` file and then give only the last part. E.g. if your prefix in `.ltools` is "www.lqcd.org/ildg/qcdsf/" and the Markov-Chain-URI is "www.lqcd.org/ildg/qcdsf/b5p25kp13575-24x48" then `lls` will show you all configurations by simply typing "`lls b5p25kp13575-24x48`" (see Chap. 5).

3.3.5. Example: (Shortened) Output of presentation of all configurations of an ensemble:

```
:~> lls www.lqcd.org/ildg/qcdsf/b5p40kp13610-24x48
```

```
Welcome to the Ltool-command lls -  
Testing grid-proxy-init:  
grid-proxy...ok
```

```
Virtual Organisation is ildg
```

```
The following configurations belong to  
www.lqcd.org/ildg/qcdsf/b5p40kp13610-24x48 :
```

```
qcdsf_b5p40kp13610-24x48_bqcd.561.1.1.00125.tar  
qcdsf_b5p40kp13610-24x48_bqcd.561.1.1.00130.tar  
qcdsf_b5p40kp13610-24x48_bqcd.561.1.1.00135.tar  
qcdsf_b5p40kp13610-24x48_bqcd.561.1.1.00140.tar  
qcdsf_b5p40kp13610-24x48_bqcd.561.1.1.00145.tar  
qcdsf_b5p40kp13610-24x48_bqcd.561.1.1.00150.tar
```

```
.  
.  
.
```

```
:~>
```

3.4. linit

3.4.1. Creating a new ensemble-entry in the MDC

A new ensemble with the name specified in the metadata xml description (Markov-Chain-URI) can be initialised with the following command:

```
linit <PID> <Metadatafilename>
```

Special administration rights are necessary and the metadata has to be valid QCDDL (otherwise it will be rejected by the MDC). PID is the ID of the project where the

ensemble belongs to. You can see the IDs of the projects by typing `ladm -show-projects`).

3.5. *lupdate*

3.5.1. Updating metadata in the MDC

In case you want to update only the metadata of a configuration or an ensemble, that is, updating entries in the MDC but changing nothing in the Storage-Elements, you can use the `lupdate` command. Simply update the metadata-xml-file of the configuration/ensemble and type:

```
lupdate <Updated_Metadadatafile>
```

Of course, adequate access rights for the MDC are required. Please note that the command is exactly the same for configuration and ensemble, because `lupdate` will see what kind of metadata to be uploaded by parsing the file.

3.5.2. Example Output of updating a configuration's metadata

```
~> lupdate qcdfs_b5p29kp13550-24x48_bqcd.525.1.1.00370.xml
```

```
Welcome to the Ltool-command lupdate -  
Testing grid-proxy-init  
grid-proxy...ok
```

```
MDC says:
```

- Getting default credential
- Authorization: HOST
- Logging is working
- Date: Mon Sep 05 14:39:34 CEST 2005
- Version: Apache-XML-Security-J 1.0.4
- Response Status: Ok.

```
...MDC Operation ok.
```

```
~>
```

3.6. *lvalidate*

3.6.1. Validating a file to be valid QCDML

In order to check whether a metadata-file contains a syntactically correct QCDML description of a configuration or an ensemble, you can use the `lvalidate` command:

```
lvalidate <QCDML_File>
```

Please note, that this command is exactly the same for a configuration and an ensemble metadata-description (lvalidate will simply validate the file against both schemata, and show a success message or in case of an error the error-message for both cases). The actual validation is done via webservice from the MDC, so it is assured that there is always the same result on client-side and server-side validation.

3.6.2. Example: Output of validation of the metadata description of an ensemble

```
~> lvalidate b5p25kp13575-24x48.xml

Welcome to the Ltool-command lvalidate -

Valid ! b5p25kp13575-24x48.xml
seems to be a valid QCDML-ensemble description !
~>
```

3.7. ladm

3.7.1. Purpose of ladm

ladm is used for showing/inserting/updating/deleting users, groups etc.; that is, all control/access information. In other words, ladm should be able to manage all aspects of access control.

3.7.2. Synopsis of ladm

Showing data: **ladm --show-admins**
 --show-managers
 --show-groups
 --show-groupmembers
 --show-grouprights
 --show-users
 --show-projects

where “ladm --show-admins“ prints all admins ...
 (Certificate-Subjectline and ID)

Inserting: **ladm --add-admin <Certificatesubjectline>**
 --add-manager <PrjID> <Certificatesubjectline>
 --add-project <Collaborationname> <Projectname>
 --add-group <PrjID> <Groupname>

```

--add-groupmember <GrID><Certificatesubjectline>

Deleting:      ladm --delete-admin    <CertID>
               --delete-manager <PrjID> <CertID>
               --delete-project  <PrjID>
               --delete-ensemble <Markov-Chain-URI>
               --delete-group    <GrID>
               --delete-groupmember <PrjID> <GrID> <CertID>
               --delete-user <CertID>

Updating :     ladm --chmod <EnsID> <GrID>{+w |-w |+r |-r }
               (please note there is no blank after <GrID> !)

               --publish <EnsID>
               (removes all "+r")

               --update-project <PrjID> <NewCollaboration>
                                   <NewProjectname>
               --update-group   <PrjID> <GrID>
                                   <NewGroupname>
               --update-ensemble <PrjID> <EnsID>
                                   <NewEnsembleURI>
               --update-user    <CertID> <NewSubjectLine>

```

3.8. lid

3.8.1. Purpose of lid

“lid” is just a simple tool to get an ID to a known pattern (e.g. a Name in real life). All matching lines in all tables are shown. Another possible usage is to put the option “--all” to print a “mapping” table, that is, a table that binds an ID to a user, ensemble,...

3.8.2. Synopsis of lid

```

Usage:      lid    <Pattern>
            (will look in Certificates, Ensemblenames, Projectnames and Groupnames)

            lid --all { ensembles | projects | groups | managers |
                        admins | users }
            (will show all ensembles, ensembles, ... with IDs)

```

4. The Configuration File

To avoid unnecessary typing without loosing flexibility, a central configuration file that is used by all the ltools-command can be placed in the user's home directory.

A sample file with a lot of explaining comments is included in the installation package (see Chap. 2). The Configuration file contains logically different parts, that are described in the following chapters.

4.1. Options for the main program behaviour

The first two options in the configuration file tell the ltools whether you want to overwrite existing files (default is “no”) and if you want the tools to suppress information output (default is “no”). Substitute these values to “yes” to change the behaviour of the commands. It is recommended not to change the default values here, but use “-f” and “-q” as command-line option if necessary.

4.2. Grid/ILDG Defaults

This section in the configuration file deals with grid-specific things, primarily to shorten frequently used commands.

4.2.1. Default VO

Please enter your Virtual Organisation. This string will be taken as your VO whenever this is needed by a command. You can always override this value with the “-vo”-command line option in the related ltools commands.

4.2.2. Default Ensemble prefix for Downloading

Use this option to define a prefix for downloading the metadata of an ensemble.

It will be added at the beginning if data is not found otherwise.

E.g. if prefix is "www.lqcd.org/ildg/" and the ensemble Markov-Chain-URI is www.lqcd.org/ildg/qcdsf/b5p25kp13575-24x48 then you can lget it with

```
> lget -m qcdsf/b5p25kp13575-24x48
```

If you are sure to only need data of a certain project, you might even specify this prefix as “www.lqcd.org/ildg/qcdsf” and then only type

```
> lget -m b5p25kp13575-24x48
```

Regardless of this prefix, you can always download the ensemble metadata using the full qualified Markov-Chain-URI, e.g. :

```
> lget www.lqcd.org/ildg/qcdsf/b5p25kp13575-24x48
```

4.2.3. Default LFN prefix for Downloading

As with the ensemble metadata, you also can specify a default prefix to be part of the Logical Filename (LFN), which is used to download a configuration binary or its metadata (using “-m”). If this string is not included in your argument if you try to download, it will be added at the beginning if data is not already found without that prefix (therefore, you can always download with typing the whole LFN regardless of what is defined here). E.g. if prefix is "www.lqcd.org/ildg/" and the LFN of a configuration is

www.lqcd.org/ildg/qcdsf/b5p25kp13575-24x48/bqcd.535.1.1.00180.tar then you can lget it with

```
>lget -m qcdsf/b5p25kp13575-24x48/bqcd.535.1.1.00180.tar or
```

```
>lget -m www.lqcd.org/ildg/qcdsf/b5p25kp13575-24x48/bqcd.535.1.1.00180.tar
```

4.2.4. Default UPLOAD Storage element

The default Upload Storage element is used, unless you specify an other using the “--se” option. E.g., if you are to upload data usually to dcache.zib.de, please set “default_upload_se dcache.zib.de”.

5. Man pages

As some people prefer to take a quick look at a man page rather than reading a long description, the man pages of the described commands are appended here additionally.

5.1. man page *lget*

`lget(1)` `Ltools` `lget(1)`

NAME

`lget` - getting a binary or the metadata for a binary
or an ensemble from the Grid

SYNOPSIS

`lget` [`Options`] `LogicalFilename` [`LocalFilename`]

`lget -m -e` [`Options`] `MarkovChainURI` [`LocalFilename`]

DESCRIPTION

The `lget`-command is build upon the `lcg-2` software.

It launches e.g. the `grid-proxy-init` and the `lcg-cp` command, therefore `lcg-2` has to be installed correctly.

"`LogicalFilename`" (`LFN`) should be the `LFN` as it is stored in the `MDC`. To get a configuration binary simply type "`lget LFN`" and it will be saved to the current working directory (`cwd`).

If the `LFN` consists of several parts divided by slashes (e.g. `www.lqcd.org/ildg/qcdsf/...`), the file will be saved as the last part (unless a `LocalFilename` is specified).

If the `-m` Flag is specified, you will get the metadata of a configuration instead of the binary. If you want to download the metadata of an ensemble, use "`-m -e <Markov-chain-URI>`". Metadata will have the suffix `.xml` additionally (and existing suffixes like `tar` or `zip` are cut).

You might specify a `LocalFilename` eventually with a leading path relativ to your `cwd` or absolut with a leading `/`.

If `LocalFilename` is a directory, the file will be saved in this directory.

The command reads `$HOME/.ltools` on start (if it exists). Use that configuration file to specify several options (see

documentation in template file).
See examples below.

OPTIONS

--version

Display the version of lget.

-h --help

Print a help message.

-q --quiet

No messages will be displayed.

--vo <vo>

Using the given vo, overriding the vo specified in ~/.ltools

-m --metadata

Getting the metadata instead of the binary (see above).

-e --ensemble

(together with -m/--metadata):Type Markov-chain-URI to get metadata of an ensemble.

-f --force

Overwrite existing files without prompting (by default user will be asked for confirmation).

BUGS

Please send to wollny@zib.de

SEE ALSO

lget, lput, lls, lupdate, linit, lrm, lvalidate

AUTHOR

Stefan Wollny, Hinnerk Stueben

COPYRIGHT

ENVIRONMENT

Uses \$HOME to locate \$HOME/.ltools

Uses \$LROOT to locate LTOOLS

EXAMPLES

lget <LFN>

Copies the file <LFN> from any Storage Element to the current directory.

lget -m -e <EnsembleID>

Copies the metadata for ensemble <EnsembleID> to the current directory.

```
lget -m -f <LFN> info.xml
```

Gets the metadata for the configuration with <LFN> from the MDC and saves it in the file info.xml in the current directory (any existing file info.xml is overwritten because of -f).

Stefan Wollny, Hinnerk Stueben 1.1 lget(1)

5.2. man page lput

lput(1) Ltools lput(1)

NAME

lput - putting a binary and the corresponding metadata of a configuration on the grid

SYNOPSIS

```
lput [Options] Configurationmetadata [Path_to_binary | Binaryfilename]
```

DESCRIPTION

Putting a binary file and the corresponding metadata for a configuration on the Grid. The Ensemble has to exist already. The metadata has to be valid QCDML.

Appropriate Access rights are necessary to upload the data to the MDC. Configurationdata and Binary will be uploaded in one transaction (that is, if one of these operation fails nothing will be stored).

lput will look for the binary in the following order:

If only the file with the configuration metadata and nothing additional is specified, lput will look in the current directory for a file named by the LFN (or last part of LFN if it contains slashes) specified in the configuration.

If additionally a directory is specified, lput will look for a file named by the LFN in that directory.

If additionally a filename is specified, lput will assume this as the binary (in the current directory or in a different directory if a path is specified).

If Binaryfilename is given and starts with "srm://", lput will try to register that file that is already in a SE (of course metadata will be uploaded to MDC as usual).

OPTIONS

--version

Display the version of lput.

-h --help

Print a help message.

-q --quiet

No messages will be displayed.

--vo <vo>

Using the given vo, overriding the vo specified in ~/.ltools Note: At the moment this is only implemented for the ildg vo !

--show_SEs

Shows all Storage Elements

--se --storage-element <SE>

Forcing the upload to the given storage element (also see .ltools)

---path <SRM-Path>

Do not write binary to default location but to SRM-Path (might even be a different SE). Specify full SRM-Path !

BUGS

Please send to wollny@zib.de

SEE ALSO

lget, lput, lls, lupdate, linit, lrm, lvalidate

AUTHOR

Stefan Wollny, Hinnerk Stueben

COPYRIGHT

ENVIRONMENT

Uses \$HOME to locate ~/.ltools

Uses \$LTOOLS_HOME to locate \$LTOOLS_HOME/Import/ltools.pm

(\$LTOOLS_HOME has to point to the directory where Ltools were installed)

EXAMPLES

lput qcdfs_b5p29kp13590-24x48.xml

Checks file to be valid QCDML. If yes, inserts the configuration in the MDC and uploads the binary to the default SE (specified in ~/.ltools; if no default SE is defined in .ltools or command-line option, lput will die). Binary named as LFN has to be in the current directory. If the complete transaction goes well, a success message will be shown, otherwise everything will be rolled back (hopefully).

```
lput --se dcache.fz-juelich.de qcdfs_b5p29kp13590-24x48.xml  
      /home/ildg/qcdfs/binaries/
```

Same as above, but binary will be taken from /home/ildg/qcdfs/binaries/ and uploaded to the specified SE.

```
lput qcdfs_b5p29kp13590-24x48.xml binary
```

Same as the first, but will take file >binary< from the current directory as the binary.

```
lput --path srm://<PATH> qcdfs_b5p29kp13590 binary.
```

Same as before, but will write binary not to default path but to srm://<PATH> .

```
lput qcdfs_b5p29kp13590-24x28.xml srm://<PATH>
```

Will not upload a local binary but register a binary that is already in a SE (local corresponding metadata is written to MDC)

Stefan Wollny, Hinnerk Stueben	1.1	lput(1)
--------------------------------	-----	---------

5.3. man page lls

lls(1)	Ltools	lls(1)
--------	--------	--------

NAME

lls - printing all LFNs of an ensemble or all ensembles

SYNOPSIS

```
lls [Options] MarkovchainURI | EID
```

```
lls --all    (showing all Ensembles)
```

DESCRIPTION

Lists all configurations of an ensemble. Type the Markov-chain-URI specified in the MDC. LFNs are sorted alphabetically. You might also specify a prefix in your .ltools file and then give only the last part.

E.g. if your prefix in .ltools is "www.lqcd.org/ildg/qcdfs/" and the ensemble ID is

"www.lqcd.org/ildg/qcdsf/b5p25kp13575-24x48" then lls will show you all configurations by simply typing lls b5p25kp13575-24x48 (see .ltools). You can also use the ID that is stored in the MDC for the ensemble (EID).

OPTIONS

--version

Display the version of lls.

-h --help

Print a help message.

-q --quiet

No messages will be displayed.

--vo <vo>

Using the given vo, overriding the vo specified in ~/.ltools Note: At the moment this is only implemented for the ildg vo !

--all Printing all ensembles currently in the MDC.

-l Printing also access rights for this ensemble (might be used together with --all for access rights of all ensembles).

BUGS Please send to wollny@zib.de

SEE ALSO

lget, lput, lls, lupdate, linit, lrm, lvalidate

AUTHOR

Stefan Wollny, Hinnerk Stueben

COPYRIGHT

ENVIRONMENT

Uses \$HOME to locate ~/.ltools

Uses \$LTOOLS_HOME to locate \$LTOOLS_HOME/Import/ltools.pm

(\$LTOOLS_HOME has to point to the directory where Ltools were installed)

EXAMPLES

lls www.lqcd.org/ildg/qcdsf/b5p25kp13575-24x48 prints the sorted LFNs of all configurations that belong to this ensemble.

Stefan Wollny, Hinnerk Stueben

1.1

lls(1)

5.4. man page *linit*

linit(1) Ltools linit(1)

NAME

linit - initialising an ensemble in the MDC

SYNOPSIS

linit [Options] Project-ID Ensemblemetadata

DESCRIPTION

A new ensemble with the name specified in the metadata will be initialised. Special administration rights are necessary. The metadata has to be valid QCDML. The new ensemble has to be added to a project (Project-ID). You can see existing projects in MDC with `ladm --show-projects`.

OPTIONS

`--version`

Display the version of linit.

`-h --help`

Print a help message.

`-q --quiet`

No messages will be displayed.

`--vo <vo>`

Using the given `vo`, overriding the `vo` specified in `~/.ltools`

Note: At the moment this is only implemented for the `ildg vo` !

BUGS Please send to wollny@zib.de

SEE ALSO

`lget`, `lput`, `lls`, `lupdate`, `linit`, `lrm`, `lvalidate`, `ladm`

AUTHOR

Stefan Wollny, Hinnerk Stueben

COPYRIGHT

ENVIRONMENT

Uses `$HOME` to locate `~/.ltools`

Uses \$LTOOLS_HOME to locate \$LTOOLS_HOME/Import/ltools.pm (\$LTOOLS_HOME has to point to the directory where Ltools were installed)

EXAMPLES

```
linit 8 qcdfs_b5p29kp1359024x48.xml
```

Checks file to be a valid QCDML Ensemble description. If yes, and if a project with ID 8 exists, creates the ensemble in the MDC.

5.5. man page lupdate

lupdate(1) Ltools lupdate(1)

NAME

lupdate - updating the metadata of a configuration or an ensemble in the grid

SYNOPSIS

```
lupdate [Options] {Configurationmetadata | Ensemblemetadata}
```

DESCRIPTION

Uploading updated metadata for an ensemble or a configuration on the Grid. The metadata in the file has to be valid QCDML.

Appropriate access rights are necessary to upload the data to the MDC (administration rights to update ensemble-metadata).

OPTIONS

--version

Display the version of lupdate.

-h --help

Print a help message.

-q --quiet

No messages will be displayed.

--vo <vo>

Using the given vo, overriding the vo specified in ~/.ltools Note: At the moment this is only implemented for the ildg vo !

BUGS Please send to wollny@zib.de

SEE ALSO

lget, lput, lls, lupdate, linit, lrm, lvalidate

AUTHOR

Stefan Wollny, Hinnerk Stueben

COPYRIGHT

ENVIRONMENT

Uses \$HOME to locate ~/.ltools

Uses \$LTOOLS_HOME to locate \$LTOOLS_HOME/Import/ltools.pm

(\$LTOOLS_HOME has to point to the directory where Ltools were installed)

EXAMPLES

```
lupdate qcdfs_b5p29kp13590-24x48.xml
```

Checks file to be valid QCDML. If yes, updates the configuration in the MDC.

5.6. man page lvalidate

lvalidate(1)

Ltools

lvalidate(1)

NAME

lvalidate - validating a configuration or ensemble metadata-description to be valid QCDML

SYNOPSIS

```
lvalidate [Options] LocalFilename
```

DESCRIPTION

The Filename should be the relative or absolute filename of an ensemble or a configuration metadata-description.

Actual validation is done by a webservice from the MDC.

If it is not valid configuration- or ensemble-metadata, an appropriate error-message will be shown.

OPTIONS

--version

Display the version of lvalidate.

-h --help
Print a help message.

-q --quiet
No messages will be displayed (just return code)

BUGS

Please send to wollny@zib.de

SEE ALSO

lget, lput, lls, lupdate, linit, lrm, lvalidate

AUTHOR

Stefan Wollny, Hinnerk Stueben

COPYRIGHT

ENVIRONMENT

Uses \$HOME to locate \$HOME/.ltools
USES \$LTOOLS_HOME to locate \$LTOOLS_HOME/Import/ltools.pm
(\$LTOOLS_HOME has to point to the directory where Ltools are installed)

EXAMPLES

lvalidate <Filename>
Validates the file <Filename> to be a valid QCDML-
configuration or ensemble metadata description.

Stefan Wollny, Hinnerk Stueben 1.1 lvalidate(1)

5.7. man page ladm

ladm(1) Ltools ladm(1)

NAME

ladm - administrating access control

SYNOPSIS

ladm [-q] {Option} {Option-specific Argument(s), see below }

DESCRIPTION

Command to show/insert/update/delete users, groups and so on.
As there is a quite long list of possible options and the meaning is quite self-

explanatory (e.g. to delete a user, type "ladm --delete-user <CID>"), please see options for details. Without privileges, the normal user will only be able to execute the `ladm --show-*` commands, e.g. `ladm --show-users`. See examples below.

OPTIONS

CID, GID, PID, EID stands for Certificate-, Group, Project, Ensemble Identifier (number); Certificate DN is the certificate distinguished name (sometimes also called "Subject-Line").

To show certain data type

```
--show-admins
--show-managers
--show-groups
--show-groupmembers
--show-grouprights
--show-projects
--show-users
```

To enter data type

```
--add-admin    <Certificate DN>
--add-manager  <PID> <Certificate DN>
--add-project  <Collaborationname> <Projectname>
--add-group    <PID> <Groupname>
--add-groupmember <GID> <Certificate DN>
```

To delete data type

```
--delete-admin    <CID>
--delete-manager  <PID> <CID>
--delete-project  <PID>
--delete-ensemble <EID>
--delete-group    <GID>
--delete-groupmember <PID> <GID> <CID>
--delete-user <CID>
```

To change access rights type

```
--chmod <GID>{+w | -w | +r | -r } <EID> [<EID2> <EID3> ...]
    (please note there is no blank after <GID> !)
```

So if you want to grant write-access to group 1 for ensemble 2 then type "--chmod 1+w 2"

To make an ensemble world-readable type

--publish <EID>

To update names etc. type

--update-project <PID> <NewCollaboration> <NewProjectname>

--update-group <PID> <GID> <NewGroupname>

--update-ensemble <PID> <EID> <NewEnsembleURI>

--update-user <CID> <New Certificate DN>

To get or set configurations of MDC type

--get-config

--set-config <Name> <Value> <Force>

These three arguments will be straightforward given to the MDC; You have to know what to do ...

BUGS

Please send to wollny@zib.de

SEE ALSO

lget, lput, lls, lupdate, linit, lrm, lvalidate, lid

AUTHOR

Stefan Wollny, Hinnerk Stueben

COPYRIGHT

ENVIRONMENT

Uses \$HOME to locate \$HOME/.ltools

Uses \$LTOOLS_HOME to locate \$LTOOLS_HOME/Import/ltools.pm

(\$LTOOLS_HOME has to point to the directory where Ltools are installed).

EXAMPLES

ladm --show-groups

Will show all groups in the MDC

ladm --delete-user 23

Will delete User 23

ladm --add-groupmember 5 <Certificate DN> will add user with <Certificate DN> to group 5 (but only if that user is member of the correct VO).

5.7. man page lid

lid(1) Ltools lid(1)

NAME

lid - printing IDs of users, groups and other

SYNOPSIS

lid <Pattern>

(will look in Certificates, Ensemblenames, Projectnames and Groupnames)

lid --all { ensembles | projects | groups | managers | admins | users }

(will show all ensembles, ensembles, ... with IDs)

DESCRIPTION

Gives the ID of a CertificateSubjectline, Ensemble, Project or Group as used in the MDC. Or lists all certificates/ensembles/projects/groups/users in the MDC

OPTIONS

--version

Display the version of lid

-h --help

Print a help message.

-q --quiet

No messages will be displayed.

--all Printing all UserIDs, GroupIDs ... currently in the MDC.

BUGS Please send to wollny@zib.de

SEE ALSO

lget, lput, lls, lupdate, linit, lrm, lvalidate, ladm

AUTHOR

Stefan Wollny, Hinnerk Stueben

COPYRIGHT

ENVIRONMENT

Uses \$HOME to locate ~/.ltools

Uses \$LTOOLS_HOME to locate \$LTOOLS_HOME/Import/ltools.pm
(\$LTOOLS_HOME has to point to the directory where Ltools were installed)

EXAMPLES

lid <User name>

prints all tables with ID that contain <User name> (kind of grep)

lid --all users

prints all Users with IDs

lid --all managers

prints all managers

Stefan Wollny, Hinnerk Stueben 1.1

lid(1)

6. Typical Itools session

Typical Itools-session (using example data):

You might start by looking which ensembles are stored in the MDC :

```
> lls --all

Welcome to the Ltool-command lls -

Showing all ensembles currently in the MDC:
=====

Markov-Chain-URI                                     EID      Read      Write
=====
mc://ldg/etmc/tmqcd_nf2/tlSym_b3.9_L24T48_k0.1609_mu0.0075 17      All      [6]
mc://ldg/gral/wilson_nf2/b5p6kp1575-14x32                16      [2,3,4]  [2,4]
mc://ldg/qcdsf/lover_nf2/b5p25kp13460-16x32              4       All      [2,3]
mc://ldg/qcdsf/lover_nf2/b5p25kp13575-24x48              5       All      [2,3]
mc://ldg/qcdsf/lover_nf2/b5p29kp13500-16x32              6       [2,3]    [2,3]
mc://ldg/qcdsf/lover_nf2/b5p29kp13550-12x32              7       All      [2,3]

...

```

Perhaps, you want to see the metadata of a particular ensemble.
So type:

```
> lget -m -e mc://ldg/qcdsf/lover_nf2/b5p29kp13590-16x32

Welcome to the Ltool-command lget -
Trying to get metadata for ensemble...

Validating downloaded File: OK.
> vi b5p29kp13590-16x32.xml

```

Then you might want to see which configurations belong to this ensemble and also which detailed access rights exist:

```
> lls -l mc://ldg/qcdsf/lover_nf2/b5p29kp13590-16x32

Welcome to the Ltool-command lls -

Details:
-----

Ensemble: mc://ldg/qcdsf/lover_nf2/b5p29kp13590-16x32
  Allowed to read:
    All
  Allowed to write:
    Group: 2
      O=GermanGrid,OU=<An Organisation>,CN=<A Name>
      O=GermanGrid,...
      O=GermanGrid,...

```

The following configurations belong to

```
mc://ldg/qcdsf/clover_nf2/b5p29kp13590-16x32 :
lfn://ldg/qcdsf/clover_nf2/b5p29kp13590-16x32/bqcd.546.1.1.00215.dat
lfn://ldg/qcdsf/clover_nf2/b5p29kp13590-16x32/bqcd.546.1.1.00220.dat
lfn://ldg/qcdsf/clover_nf2/b5p29kp13590-16x32/bqcd.546.1.1.00225.dat
...
```

Then, maybe you want to download metadata of the first configuration in your local folder ~/my_data and take a look at it:

```
> lget -m lfn://ldg/qcdsf/clover_nf2/b5p29kp13590-16x32/bqcd.546.1.1.00215.dat
~/my_data/

Welcome to the Ltool-command lget -
Trying to get metadata for configuration...

Validating downloaded File: OK.
> vi ~/my_data/bqcd.546.1.1.00215.xml
```

To get the binary, simply type

```
> lget lfn://ldg/qcdsf/clover_nf2/b5p29kp13590-16x32/bqcd.546.1.1.00215.dat ~/my_data

Welcome to the Ltool-command lget -
Testing grid-proxy-init:
grid-proxy...ok

Trying to get binary ...

Virtual Organisation is ildg
Executing leg-cp --vo ildg lfn:/grid/ildg/ldg/qcdsf/clover_nf2/b5p29kp13590-
16x32/bqcd.546.1.1.00215.dat file:/home/wollny/my_data/bqcd.546.1.1.00215.dat

Checking nonzero size of downloaded File ...ok.
```

(If you do not type the second argument "~/my_data", data will be stored in the current directory).

Let's suppose, finally you want to know which projects exist in the MDC. You remember that this will be shown using the "ladm" command, but you do not know the exact syntax.

So you type "-h" for help (this holds for all Ltools-commands):

```
> ladm -h

Usage: ladm [-q] {Option} {Option-specific Argument(s), see below}

Description
-
Some Options (see manpage also)

--show-admins
--show-managers
--show-groups
--show-groupmembers
--show-grouprights
--show-projects
--show-users

--add-admin      <Certificate DN>
--add-manager    <PID> <Certificate DN>
```

```

--add-project      <Collaborationname> <Projectname>
--add-group        <PID> <Groupname>
--add-groupmember <GID> <Certificate DN>

--delete-admin     <CID>
--delete-manager   <PID> <CID>
--delete-project   <PID>
--delete-ensemble  <EID>
--delete-group     <GID>
--delete-groupmember <GID> <CID>
--delete-user      <CID>

--chmod <GID>{+w |-w |+r |-r } <EID> [<EID2> <EID3> ...]
      (please note there is no blank after <GID> !)

--publish <EID>

--update-project   <PID> <NewCollaboration> <NewProjectname>
--update-group     <PID> <GID> <NewGroupname>
--update-ensemble  <EID> <NewEnsembleURI>
--update-user      <CID> <New Certificate DN>

--get-config
--set-config       <Name> <Value> <Force>

```

And so you type

```
> ladm --show-projects
```

```

Welcome to the Ltool-command ladm -
Testing grid-proxy-init:
grid-proxy...ok

```

```

PID Collaboration Projectname
=== =====
3   qcdfs          clover_nf2
4   gral           wilson_nf2
5   etmc           tmqcd_nf2
...

```