

- Command Reference

Contents

- [htcaas-job-submit](#)
- [htcaas-job-status](#)
- [htcaas-job-log](#)
- [htcaas-job-cancel](#)
- [htcaas-job-resubmit](#)
- [htcaas-file-put](#)
- [htcaas-file-get](#)
- [htcaas-file-list](#)
- [htcaas-mgjob-submit](#)

• htcaas-job-submit

Submit a metajob (general job)

- `htcaas-job-submit [--help or -h]`
- `htcaas-job-submit [-f <jsdl file>]`

- Options:

- `-h` or `--help` : Display usage information
- `-f <jsdl file>` : Submit a metajob in accordance with '`<jsdl file>`'
- `-t <estimated time(sec)>` : To be used for backfill function

- Exit Status:

- `htcaas-job-submit` will exit with a `metajob id` upon success, and it will exit with error message upon failure

- Example:

To submit a metajob using `hello.jsdl`, (correct the jsdl file first)

```
% htcaas-job-submit ./hello.jsdl
or
% htcaas-job-submit /somewhere/hello.jsdl
```

• htcaas-job-status

Get a status of metajob

- `htcaas-job-status [--help or -h]`
- `htcaas-job-status [-m <metajob id>] [-s <sequence>] [custom options]`

- Options:

- `-h` or `--help` : Display usage information
- `-m <metajob id>` : Shows status of metajob having `<metajob id>`
- `-s <sequence>` : Shows status of a specific subjob of `<metajob id>`. If put empty value on `<sequence>`, it shows all subjobs' info of the metajob
- **Custom options:**
 - `--all` or `-a` : shows all metajobs' status and information
 - `--info` or `-i` : shows information in detail
 - `--progress` or `-p` : shows summary

- Exit Status:

- `htcaas-job-status` will return information and status upon success, and it will exit with error message upon failure

- Example:

To see status of all subjobs in a metajob (`id` is `1234`)

```
% htcaas-job-status -m 1234 -s
```

To see status of specific subjob(e.g. 1st one) of the metajob

```
% htcaas-job-status -m 1234 -s 1
```

To see only status for all subjobs in the metajob,

```
% htcaas-job-status -m 1234 -s | grep Status
Status: done
```

The `-p` option displays the summary of a metajob

```
% htcaas-job-status -m 1234 -p
=====
```

```
MetaJob ID : 1234
Total Jobs : 10
Waiting Jobs : 2
Preparing Jobs : 0
Running Jobs : 4
Done Jobs : 4
Failed Jobs : 0
canceled Jobs : 0
```

The `-i` option displays detailed information of a metajob

```
% htcaas-job-status -m 1234 -i
or
% htcaas-job-status -m 1234 -s 1 -i
```

• **htcaas-job-log**

Get a log(standard out & error) of metajob

- `htcaas-job-log [--help or -h]`
- `htcaas-job-log [-m <metajob id>] [-s <sequence>]`
- **Options:**
 - `-h` or `--help` : Displays usage information
 - `-m <metajob id>` : Shows log of metajob having `<metajob id>`
 - `-s <sequence>` : Shows log of a specific subjob of `<metajob id>`
- **Exit Status:**
 - `htcaas-job-log` will show log message upon success, and it will exit with error message upon failure
- **Example:**

To see status of specific subjob(e.g. 1st one) of a metajob (`id` is `1234`)

```
% htcaas-job-log -m 1234 -s 1
```

• **htcaas-job-cancel**

Cancel a metajob

- `htcaas-job-cancel [--help or -h]`
- `htcaas-job-cancel [-m <metajob id>]`
- **Options:**
 - `-h` or `--help` : Displays usage information
 - `-m <metajob id>` : Cancels a metajob having `<metajob id>`
- **Exit Status:**
 - `htcaas-job-cancel` will show its metajob id upon success, and it will exit with error message upon failure
- **Example:**

To cancel a metajob (`id` is `1234`)

```
% htcaas-job-cancel -m 1234
```

• **htcaas-job-resubmit**

Resubmit subjob(s)

- `htcaas-job-resubmit [--help or -h]`
- `htcaas-job-resubmit [-m <metajob id>] [-s <sequence >]`
- `htcaas-job-resubmit [-m <metajob id>] [-c <status >]`
- **Options:**
 - `-h` or `--help` : Displays usage information
 - `-m <metajob id>` : Resubmits a metajob having `<metajob id>`
 - `-s <sequence>` : Sequence of subjob(e.g., 1,2,..). One or multiple (e.g. 1 or 1-100 or 1,3,5-7)
 - `-c <status>` : Status constraint to be resubmitted (`waiting` | `failed` | `preparing` | `running` | `done`)
- **Exit Status:**
 - `htcaas-job-resubmit` will show its metajob id upon success, and it will exit with error message upon failure
- **Example:**

To resubmit specific subjob(s)(e.g. 1,2,3 and 10th) of a metajob (`id` is `1234`)

```
% htcaas-job-resubmit -m 1234 -s 1-3,10
```

To resubmit subjob(s) having specific status (e.g. failed) of a metajob (`id` is `1234`)

```
% htcaas-job-resubmit -m 1234 -c failed
```

- **htcaas-file-put**

Upload(put) file(s) or directory to a Server

- `htcaas-file-put [--help or -h]`
- `htcaas-file-put [-f <file name>] [-r <remote location>] [-i <username>]`
- `htcaas-file-put [-l <local directory>] [-r <remote location>] [-i <username>]`
- **Options:**
 - `-h` or `--help` : Displays usage information
 - `-f <file name with absolute path>`: Sends *<file name>* to destination. It should be absolute path
 - `-l <local directory>` : Sends entire *<local directory>* to destination
 - `-r <remote location>` : Send file or directory to *<remote location>*
 - `-i <username>` : Transferred file or directory is owned by *<username>*

- **Exit Status:**

- `htcaas-file-put` will return no message upon success, and it will exit with error message upon failure

- **Example:**

To send a file *sample.dat* owned by *someuser* to a working directory on the server

```
% htcaas-file-put -f /somewhere/sample.dat -r /server/workingdir/ -i someuser
or
% htcaas-file-put -f ${PWD}/sample.dat -r /server/workingdir/ -i someuser
```

To send a directory owned by *someuser* to a working directory on the server

```
% htcaas-file-put -d /somewhere/thisDIR -r /server/workingdir/ -i someuser
or
% htcaas-file-put -d ${PWD}/thisDIR -r /server/workingdir/ -i someuser
```

If you want to send a file or directory to **NEWDIR** which does not exist on working directory of server,

```
% htcaas-file-put -f /somewhere/sample.dat -r /server/workingdir/NEWDIR -i someuser
or
% htcaas-file-put -d /somewhere/thisDIR -r /server/workingdir/NEWDIR -i someuser
```

- **htcaas-file-get**

Download(get) file(s) or directory from a Server

- `htcaas-file-get [--help or -h]`
- `htcaas-file-get [-l <filename with absolute localpath>] [-r <file name with remote location>] [-i <username>]`
- `htcaas-file-get [-l <dir with absolute localpath>] [-r <file name with remote location>] [-i <username>]`
- **Options:**
 - `-h` or `--help` : Display usage information
 - `-l <path+{file or dir}>` : *file* or *directory* name to be downloaded will be located on *path*
 - `-r <remote path+{file or dir}>` : *file* or *directory* name on *source(remote)* location
 - `-i <username>` : Transferred file or directory is owned by *<username>*

- **Exit Status:**

- `htcaas-file-get` will return no message upon success, and it will exit with error message upon failure

- **Example:**

To get a file *sample.dat* owned by *someuser* from a working directory on the server to local directory *somewhere*

```
% htcaas-file-get -l /somewhere/sample.dat -r /server/workingdir/sample.dat -i someuser
or
% htcaas-file-get -l ${PWD}/sample.dat -r /server/workingdir/sample.dat -i someuser
```

To get a directory owned by *someuser* from a working directory on the server to local directory *somewhere*

```
% htcaas-file-get -l /somewhere/thisDIR -r /server/workingdir/thisDIR -i someuser
or
% htcaas-file-get -l ${PWD}/thisDIR -r /server/workingdir/thisDIR -i someuser
```

If you want to send a file or directory to **NEWDIR** which does not exist on local,

```
% htcaas-file-get -l /somewhere/NEWDIR/sample.dat -r /server/workingdir/sample.dat -i someuser
or
% htcaas-file-get -l /somewhere/NEWDIR/thisDIR -r /server/workingdir/thisDIR -i someuser
```

- **htcaas-file-list**

List directory contents on server

- `htcaas-file-list [--help or -h]`
- `htcaas-file-list [-p or -pwd <FTP password>] [-r <remote directory>]`
- **Options:**
 - `-h` or `--help` : Display usage information
 - `-p <FTP password>` : Server *password* for this user
 - `-r <remote directory >` : It lists *<remote directory >*, and it returns the list of user's *home directory* if don't use `-r` option,
- **Exit Status:**
 - `htcaas-file-list` will return list of preferring directory upon success, and it will exit with error message upon failure

- **Example:**

To get list of a directory *somewhere* on server,

```
% htcaas-file-list -p pass123 -r /somewhere
[Directory]: MG5_aMC_v2_1_2
[Directory]: MG5_workspace
[Directory]: lib
[Directory]: output
[Directory]: workspace
[File]: PI_sub.py
[File]: PI_sub.sh
[File]: env.sh
```

To get list of user home directory

```
% htcaas-file-list -p pass123
```

- **htcaas-mgjob-submit**

Submit a metajob for [MadGraph5](https://launchpad.net/mg5amcnlo) (<https://launchpad.net/mg5amcnlo>)

- `htcaas-mgjob-submit [--help or -h]`
- `htcaas-mgjob-submit [-d or --dir <working directory>] [-e or --exec <executable(s)>] [-a <arguments>]`
- **Options:**
 - `-h` or `--help` : Display usage information
 - `-d <working directory>` : a submitted metajob will be run on *the working directory*
 - `-e <executable(s)>` : executable(s) for this metajob. Multiple executables are separated with semi-colon(;). **e.g. *ajob1;ajob2;ajob3***
 - `-a <arguments>` : argument(s) used for all subjob(s). Multiple arguments are separated with equal symbol(=). **e.g. *abc=123=xyz***
- **Exit Status:**
 - `htcaas-mgjob-submit` will exit with a *metajob id* upon success, and it will exit with error message upon failure

- **Example:**

To submit a MG5 metajob with working directory */somewhere/wd*, one executable ***ajob1*** and arguments ***2 F 0***

```
% htcaas-mgjob-submit -d /somewhere/wd -e ajob1 -a 2=F=0
```

If working directory is current directory

```
% htcaas-mgjob-submit -d $PWD -e ajob1 -a 2=F=0
```

If don't need arguments

```
% htcaas-mgjob-submit -d $PWD -e ajob1
```

To submit a MG5 Metajob having multiple subjobs with different executables(*ajob1, ajob2, ajob3*) each other.

```
% htcaas-mgjob-submit -d /somewhere/wd -e ajob1;ajob2;ajob3 -a 2=F=0
```

In above case, each subjob will be executed on some computing nodes like the followings

```
% /bin/bash /somewhere/wd/ajob1 2 F 0
% /bin/bash /somewhere/wd/ajob2 2 F 0
% /bin/bash /somewhere/wd/ajob3 2 F 0
```