

LTFSArchiver 1.0 Interface specification

Summary

LTFSArchiver 1.0 Interface specification	1
Interface Specification	2
1. Managing the tapes (TapeManager)	3
Add command	3
GetStatus command	4
GetResult command	5
Cancel command	5
Withdraw command	5
2. Querying the tapes (QueryKnownTapes)	6
3. Querying free spaces (QueryFreeSpaces)	8
4. Archiving operations (WriteToLTO)	10
WriteFile and WriteFolder commands	10
Cancel command	11
Resubmit command	11
GetStatus command	12
GetResult command	12
5. Restoring operations (RestoreFromLTO)	15
RestoreFile command	15
GetStatus, Cancel and Resubmit commands	16
GetResult command	16
6. Giving access through LTFS direct access (MakeAvailable)	17
Mount command	17
GetStatus command	18
GetResult command	18
Cancel command	19
Unmount command	19
Appendix A - Status and substatus description	21

This document describes in detail the functionality of the interfaces used to interact with LTFSArchiver agent.

Interface Specification

The way of accessing the system is by an interface of *http* services that allows to:

- Manage the tapes
- Query the tapes inside the system
- Query the free space of each tape pool
- Book read/write requests and follow the current status and result of those requests
- Directly access the tapes content with LTFS (makeavailable)

1. Managing the tapes (TapeManager)

Service URL:

<http://<servername>/pprime/cgi-bin/TapeManager>

It is used to add / remove LTO tapes to / from tape pools.

In Table 4 are shown all the possible combinations of commands and parameters, N/A means that the parameter is not applicable for that command.

Command	Parameters			
	TapelD	PoolName	Format	TaskID
Add	Mandatory	Optional	Optional	N/A
Withdraw	Mandatory	Mandatory	N/A	N/A
GetStatus (Add only)	N/A	N/A	N/A	Mandatory
GetResult (Add only)	N/A	N/A	N/A	Mandatory
Cancel (Add only)	N/A	N/A	N/A	Mandatory

Table 1 - TapeManager commands and related parameters

All of the answers from the service are plain text, according to the following format:

RC<TAB>RESULT<TAB>Comment

Where RC will be one of the following:

200 -> Request satisfied

400 -> Bad request

500 -> System error

Add command

The Add command is used to assign a new LTO tape to a pool.

TapelD parameter

TapelD must be a unique string with a maximum length of eight characters. The system DOESN'T allow to add more tapes with the same label, an attempt to do so will result in an error.

PoolName parameter

If PoolName value is not supplied, the newly added tape will be automatically assigned to the special default pool named "default".

If the Poolname supplied does not yet exist, it will be automatically created.

Format parameter

The only valid values are "Y" (yes) and "F" (force), in any other case (unused option or option values different from "Y" and "F"), it will be set to "No"

Notes about format option

If you need to add a tape that already contains an LTFS file system, choose not to format it..

As soon as possible, the tape will be mounted to:

- Check Mechanical un-protection
- Evaluate actual available space (in case it has data in it)

If you need to add a blank tape to create a new LTFS files system on it, choose “Y”. As soon as possible, the tape will be mounted and an LTFS file system will be mounted to:

- Check mechanical un-protection
- Create an LTFS file system on it according to the configuration parameter PPRIMELTO_LTFSRULE (see LTFSArchiverConfiguration documentation)

If you want to add a tape regardless his current content and create a new LTFS files system on it, choose “F”.

As soon as possible, the tape will be mounted and an LTFS file system will be mounted to:

- Check mechanical un-protection
- Create / recreate an LTFS file system on it according to the configuration parameter PPRIMELTO_LTFSRULE (see LTFSArchiverConfiguration documentation)

Please note:

Don't choose “Yes” if the tape contains a LTFS file system, or the format will fail: as a security feature, the creation of the LTFS it's executed in “non-forced” mode, in order to avoid loss of pre-existing data. If you decide to use a previously LTFS formatted tape, please run the *unltfs* utility to clear it before adding the tape to a pool or set format parameter to “F”.

Example:

<http://servername/ltfsarchiver/cgi-bin/TapeManager?Command=Add&TapeID=NS04017B&PoolName=PoolB>

Will return

200<TAB>assigned-uuid<TAB>LTO with label: NS04017B added to system and assigned to pool: PoolB

where assigned-uuid is a unique id (e.g. 2f2437e2-7565-463c-aa0c-5755b254e24b) for the operation just taken in charge.

An 400/500 exit code with a description of which error occurred will be returned if the request should fail.

GetStatus command

This command returns the status of an Add operation supplying the assigned-uuid with the parameter TaskID.

Example:

<http://servername/pprime/cgi-bin/TapeManager?Command=GetStatus&TaskID=2f2437e2-7565-463c-aa0c-5755b254e24b>

The answer during the writing can be like:

```
200<TAB>starting<TAB>Tape being loaded o positioning
```

If the task has completed, the answer is:

```
200<TAB>completed
```

See appendix A for a comprehensive list of possible status and descriptions.

GetResult command

This command provides full information about the TapeManager/Add procedure completion.

Example:

<http://servername/pprime/cgi-bin/TapeManager?Command=GetStatus&TaskID=2f2437e2-7565-463c-aa0c-5755b254e24b>

```
200<TAB>Successful
```

Note that when a Add requests should fail, the requested tape is removed from pool, and a new Add request will have to be submitted to insert it in a pool.

Cancel command

This command allows to delete a previously submitted Add operation; assigned uuid must be supplied.

An operation can be deleted only if it is still waiting for some resource assignment.

Example:

<http://10.58.78.112/lfsarchiver/cgi-bin/TapeManager?Command=Cancel&TaskID=e55c4fed-55a8-4459-bf68-ffe2ce52dc8b>

```
200<TAB>e55c4fed-55a8-4459-bf68-ffe2ce52dc8b removed from working queue
```

A successful cancel request will also remove the added tape from the assigned Pool.

Withdraw command

The Withdraw command is used to remove a tape from a pool. Both LTOLabel and PoolName parameters are requested and must both match with an existing label and pool.

Removing a tape that is currently in use or that has been booked for further use is not allowed (a message error will be returned)

Example

<http://servername/pprime/cgi-bin/TapeManager?Command==Withdraw&TapeID=NS04017B&PoolName=PoolB>

200<TAB>LTO with label: NS04017B has been deleted from pool: PoolB

2. Querying the tapes (QueryKnownTapes)

Service URL:

<http://<servername>/ltfsarchiver/cgi-bin/QueryKnownTapes>

It is used to list the tapes and associated pools known by a specific LTFSArchiver instance. Retrieved information also includes the remaining free space in Mbytes for each tape.

This service allows the client to handle with several instances of LTFSArchiver that hence can be deployed in a more complex and larger context.

In this case the client application needs to know (with a suitable configuration) the URLs of all possible LTFSArchiver instances to be used.

Following Table2 reports the possible parameters.

Parameters		
TapeID	PoolName	Output
Optional	Optional	Optional

Table 2 -QueryKnownTapes commands and related parameters

TapeID parameter

Must be supplied to look for a specific tape, if only a part of the label is given it will be used as pattern matching. If not supplied, all the tapes are reported.

PoolName parameter

By supplying the PoolName parameter, the search will look only for tapes in the specified pool. The given value must be the exact pool name (no pattern matching is allowed). If not supplied, all the pools are considered.

If neither TapeID nor PoolName are given, all the tapes known by the system are reported.

Output parameter

If omitted, the output will be formatted in an HTML table.

Accepted other values are TEXT and JSON.

Examples of text output

<http://servername/pprime/cgi-bin/QueryKnownTapes?PoolName=TestExt&Output=TEXT>

200<TAB>(EX000001,140000,TestExt,LTO5)<TAB>(EX000002,90560,TestExt,LTO5)

Where:

200 means “I’ve found something”.

LTFSArchiver 1.0 – Interface specification

For each tape found, the following four values in the order are given: Label, Free space (in MBytes), Poolname, LTObtype

<http://servername/pprime/cgi-bin/QueryKnownTapes?PoolName=ThisDoesNotExist&Output=TEXT>

400<TAB>No tape found matching criteria

Samples of json output

<http://10.2.7.141/pprime/cgi-bin/QueryKnownTapes?Output=JSON>

```
{
  "exit_code": "200",
  "output": [
    {
      "TapeID": "000002L5",
      "FreeSpace": "1443596",
      "PoolName": "poolB",
      "LTObtype": "LT05"
    },
    {
      "TapeID": "EX000001",
      "FreeSpace": "1444049",
      "PoolName": "poolC",
      "LTObtype": "LT05"
    },
    {
      "TapeID": "000001L5",
      "FreeSpace": "1433041",
      "PoolName": "poolA",
      "LTObtype": "LT05"
    }
  ]
}
```

<http://10.2.7.141/pprime/cgi-bin/QueryKnownTapes?PoolName=PoolDoesntExist&Output=JSON>

```
{
  "exit_code": "400",
  "message": "No tape found matching criteria"
}
```

3. Querying free spaces (QueryFreeSpaces)

Service URL:

<http://<servername>/pprime/cgi-bin/QueryFreeSpaces>

It is used to get some information about minimum and maximum available space on pools.

In Table 3 are shown all the accepted parameters.

Parameters	
PoolName	Output
Optional	Optional

Table 3 - QueryFreeSpaces command related parameters

PoolName parameter

If PoolName is supplied, the command will retrieve information only about the specified pool (supplying a not existing pool name will generate an error message)
If omitted, the output gives information about all of the found pools.

Output parameter

If omitted, the output will be formatted in an HTML table.

Accepted other values are TEXT and JSON.

For each pool, the following information is returned:

- Number of tapes that have some free space (tapes without free space are not counted)
- Total amount of available free space on the pool
- Free space available on the “most full” tape of the pool
- Free space available on the “least full” tape of the pool

Examples:

<http://>servername>/ltsarchiver/cgi-bin/QueryFreeSpaces?Output=JSON>

```
{
  "exit_code": "200",
  "output": [
    {
      "Poolname": "poolA",
      "NumTapes": "3",
      "Total": "3840694",
      "Min": "911478",
      "Max": "1464608"
    },
    {
      "Poolname": "poolB",
      "NumTapes": "3",
      "Total": "3871334",
      "Min": "962796",
      "Max": "1464608"
    },
    {
      "Poolname": "poolD",
      "NumTapes": "1",

```


LTFSArchiver 1.0 – Interface specification

```
        "Total": "1464608",  
        "Min": "1464608",  
        "Max": "1464608"  
    }  
]  
}
```

<http://<servername>/ltsarchiver/cgi-bin/QueryFreeSpaces?Output=JSON&PoolName=poolB>

```
{"exit_code": "200", "output": [{"Poolname": "poolB", "NumTapes": "3", "Total": "38  
71334", "Min": "962796", "Max": "1464608"}]}
```

4. Archiving operations (WriteToLTO)

Service URL:

<http://<servername>/pprime/cgi-bin/WriteToLTO>

Used to archive a file or a directory (recursive) on an LTO Tape. In Table 4 are shown all the possible combinations of commands and parameters, N/A means that the parameter is not applicable for that command.

Command	Parameters				
	FileName	PoolName	TaskID	Output	MD5
WriteFile	Mandatory	Optional	Output	N/a	Optional
WriteFolder	Mandatory	Optional	Output	N/a	Optional
GetStatus	N/a	N/a	Mandatory	N/a	N/a
GetResult	N/a	N/a	Mandatory	Optional	N/a
Cancel	N/a	N/a	Mandatory	N/a	N/a
Resubmit	N/a	N/a	Mandatory	N/a	N/a

Table 4 - WriteLTO commands and related parameters

All of the answers are in plain text format, according to the following format:

```
RC<TAB>RESULT<TAB>Comment
```

The return code (RC) can be one of the following:

```
200 -> Request satisfied
400 -> Bad request
500 -> System error
```

A detailed description of the commands follows.

WriteFile and WriteFolder commands

WriteFile is used to ask for a single file archiving.

WriteFolder is used to ask for a whole directory archiving.

FileName parameter

Is the full path to the single file (or directory) to be archived. No pattern or partial path are allowed, the path must be in absolute format and local to the machine (e.g. /mnt/repository/trythis.mxf).

PoolName parameter

Is the name of the tape pool where LTFSSArchiver will try to find a tape with space enough to write the data. If PoolName is not specified, a default pool named “default” is used.

MD5 parameter

The only valid value is “Y” (capital letter) otherwise the parameter is ignored.

LTFSArchiver 1.0 – Interface specification

When set to Y, an MD5 checksum string will be calculated for the file (in case of a folder, for each contained file) copied to the tape. This checksum string is saved and made accessible through the GetResult command (see further).

Please note that performing MD5 checksum on a very large file or directory can heavily lower the performance of the archiving procedure.

Example:

<http://servername/pprime/cgi-bin/WriteToLTO?Command=WriteFile&FileName=/mnt/repository/TryThis.mxf&PoolName=ThisPool>

A successful request returns a message like:

```
200<TAB>assigned-uuid
```

where assigned-uuid is a unique id (e.g. 2f2437e2-7565-463c-aa0c-5755b254e24b) for the operation just taken in charge.

If the file or directory does not exist, the following answer is given

```
400<TAB>thisdoesntexist
```

If the object specified with the parameter FileName does not match with the specific archive command (e.g. when FileName value refers to a directory and the command used is WriteFile), a warning message is returned, but the request is satisfied anyway.

Cancel command

This command can be used to delete a request but only if it is in a wait state.

<http://servername/pprime/cgi-bin/WriteToLTO?Command=Cancel&TaskID=assigned-uuid>

An operation can be deleted only if it is still waiting for some resource assignment.

possible answer are:

```
200<TAB>assigned-uuid deleted
400<TAB>assigned-uuid doesn't exist
400<TAB>assigned-uuid is not in wait status
```

Resubmit command

This command can be used to re-submit a request that has failed.

<http://servername/pprime/cgi-bin/WriteToLTO?Command=Resubmit&TaskID=assigned-uuid>

An operation can be deleted only if it has previously failed.

possible answer are:

```
400<TAB>assigned-uuid doesn't exist
400<TAB>assigned-uuid is not in fallout status
```

GetStatus command

This command returns the status of a the operation specified by giving the assigned-uuid with the parameter TaskID.

This command also tries to guess the percentage of the data copied from disk to tape when the task is actually writing data. During the interval between the start of the archive command and the actual “first byte write”, the size of the archived data will be unavailable, and the command could hang for several seconds (the time taken from LTO drive to position the tape at the first free data block).

Example:

<http://servername/pprime/cgi-bin/WriteToLTO?Command=GetStatus&TaskID=2f2437e2-7565-463c-aa0c-5755b254e24b>

The answer during the writing can be like:

```
200<TAB>running 80
```

While when the task has completed, the answer is:

```
200<TAB>completed
```

See appendix A for a comprehensive list of possible status and status description.

GetResult command

This command provides full information about the archiving procedure completion.

Output parameter

Possible values are “TEXT” (default if not specified) and “JSON”.

Using a value rather than the other one not only affects the output formatting, but also affects the type of the information given.

Some examples follow:

Sample 1

a single file archiving request successfully completed with assigned uuid 334e67c-69ff-4494-a071-320a0f093229

The GetResult output in Text mode is asked with the url :

<http://servername/pprime/cgi-bin/WriteToLTO?Command=GetResult&TaskID=1334e67c-69ff-4494-a071-320a0f093229>

and is like

```
200<TAB>Success<TAB>lto-ltfs:000001L5:1334e67c-69ff-4494-a071-320a0f093229/test.mkv
```

This indicates that the archiving procedure was completed successfully.

The last field starting with “lto-ltfs” is the URN of the file that has to be used for a restore.

In this case no MD5 checksum information (even if requested when issuing the WriteFile command) is returned, to get it is necessary to use the JSON mode as explained later.

LTFSArchiver 1.0 – Interface specification

The GetResult output in Json mode is asked with the url:

<http://servername/pprime/cgi-bin/WriteToLTTO?Command=GetResult&Output=JSON&TaskID=1334e67c-69ff-4494-a071-320a0f093229>

and is like

```
{
  "exit_code": "200",
  "output": [
    {
      "FLocat": "lto-ltfs:000001L5:1334e67c-69ff-4494-a071-320a0f093229/test.mkv",
      "MD5": "b883f7995a1e79337bd2c4af7eac6f68"
    }
  ]
}
```

If the WriteFile command was issued with “MD5=Y” switch, or

```
{
  "exit_code": "200",
  "output": [
    {
      "FLocat": "lto-ltfs:000001L5:1334e67c-69ff-4494-a071-320a0f093229/test.mkv"
    }
  ]
}
```

If the WriteFile command was issued without activating the MD5 feature.

Example 2

A whole folder archiving request successfully completed with assigned uuid ffa1b18b-787c-475d-9cbe-034cf6b1f88b

The GetResult output in Text mode will be:

```
200<TAB>Success<TAB>lto-ltfs:000001L5:ffa1b18b-787c-475d-9cbe-034cf6b1f88b/testdir
```

Hence in text mode only the flocat of the entire folder is given while for getting the flocat of each archived file is necessary to use the JSON modality.

The same Command returns, if issued with JSON formatted output:

```
{
  "exit_code": "200",
  "output": [
    {
      "FLocat": "lto-ltfs:000001L5:ffa1b18b-787c-475d-9cbe-034cf6b1f88b/tests/type1.685859.mp4.format",
      "MD5": "8786a3def08196e047545ac53defbae8"
    },
    {
      "FLocat": "lto-ltfs:000001L5:ffa1b18b-787c-475d-9cbe-034cf6b1f88b/tests/type2.685290.mp4.format",
      "MD5": "313ebb9803ab3f6cf16e342bb7125c0d"
    }
  ]
}
```

LTFSArchiver 1.0 – Interface specification

```
      "FLocat": "lto-ltfs:000001L5:ffa1b18b-787c-475d-9cbe-034cf6b1f88b/tests/type1.685859.mp4.stderr",
      "MD5": "a09595ac5632488a0b63a7493307c7da"
    }
  ]
}
```

If the WriteFolder command was issued with “MD5=Y” switch, or

```
{
  "exit_code": "200",
  "output": [
    {
      "FLocat": "lto-ltfs:000001L5:ffa1b18b-787c-475d-9cbe-034cf6b1f88b/tests/type1.685859.mp4.format"
    },
    {
      "FLocat": "lto-ltfs:000001L5:ffa1b18b-787c-475d-9cbe-034cf6b1f88b/tests/type2.685290.mp4.format"
    },
    {
      "FLocat": "lto-ltfs:000001L5:ffa1b18b-787c-475d-9cbe-034cf6b1f88b/tests/type1.685859.mp4.stderr"
    }
  ]
}
```

If the WriteFolder command was issued without activating the MD5 switch.

This means that a single flocat will be available for each file archived, allowing a subsequent restore of a single one of them (as an alternative to restore the whole folder).

In case some error occurred, the answer is:

```
500<TAB>failure
```

in TEXT format or

```
{
  "exit_code": "500",
  "message": "failure"
}
```

in JSON format

5. Restoring operations (RestoreFromLTO)

Service URL:

<http://servername/pprime/cgi-bin/RestoreFromLTO>

Used to restore a file or a directory from an LTO tape but also to follow the ongoing restore operation and get the result. **The system leaves to the clients the responsibility to remember which file is on which tape** hence for restoring is necessary to specify the full flocat URN of the file or folder. Of course is possible to list the content of a specific tape to discover all the files stored in.

In Table 55 are shown all the possible combinations of commands and parameters, N/A means that the parameter is not applicable for that command.

Command	Parameters		
	FileName	DestPath	TaskID
RestoreFile	Mandatory	Mandatory	Output
GetStatus	N/a	N/a	Mandatory
GetResult	N/a	N/a	Mandatory
Cancel	N/a	N/a	Mandatory
Resubmit	N/a	N/a	Mandatory

Table 5 - RestoreFromLTO commands and related parameters

RestoreFile command

This command is used to ask for a recovery of a file or a directory previously archived.

FileName parameter

It is the flocat URN that identifies the file or the directory to restore. It is the one given as a result of a WriteToLTO operation.

DestPath parameter

It is the name of the destination file/directory of the restored object.

To avoid overwriting and/or creation of unwanted directories, the following checks are applied:

- The DestPath object MUST NOT exist
- The upper-level of DestPath MUST exist.

e.g. if DestPath is /var/mydir/mydata, the directory “var/mydir” must exist and it must not contain a file or directory named “mydata”.

If one of the check fails, an error message is returned:

```
400<TAB>Destination path DestPath exists
```

```
400<TAB>Destination upper level path nameofupperleveldir doesn't exist
```

If all checks has passed, the message will be:

200<TAB>assigned-uuid

Meaning that the operation has been queued for later execution.

GetStatus, Cancel and Resubmit commands

GetStatus, **Cancel**, and **Resubmit** commands have the same syntax, usage and meaning as for the WriteToLTO function.

GetResult command

The **GetResult** command returns the result of the operation, i.e. if the restore operation has been completed successfully or not.

In case of error, a specific description of the problem occurred can be retrieved using GetErrorDescr.

<http://servername/pprime/cgi-bin/GetErrorDescr?TaskID=assigned-uuid>

6. Giving access through LTFS direct access (MakeAvailable)

Service URL:

<http://servername/pprime/cgi-bin/MakeAvailable>

It is used to open/close a **read only** access to the whole content of a LTO. The tape will be locally mounted (using LTFS) on the controlling host and possibly made available remotely through a share.

The client application will be allowed to read data directly from the tape file system, thus allowing services to run processes on archived files without need to copy them locally. Examples of such use include partial extraction of big files (e.g. multimedia), services for regular data integrity check, storage or format migrations.

In Table 6 are shown all the possible combinations of commands and parameters, N/A means that the parameter is not applicable for that command.

Command	Parameters	
	TapelD	TaskID
Mount	Mandatory	Output
Unmount	Mandatory	Output
GetStatus	N/A	Mandatory
GetResult	N/A	Mandatory
Cancel	N/A	Mandatory
Resubmit	N/A	Mandatory

Table 6 - MakeAvailable commands and related parameters

All of the answers from the service are plain text, according to the following format:
RC<TAB>RESULT<TAB>Comment

Where RC will be one of the following:

```
200 -> Request satisfied
400 -> Bad request
500 -> System error
```

and RESULT value may vary according to the command issued.

Mount command

This command is used to get access (in read only mode) to the whole content of a single tape.

TapelD

Is the only needed (and mandatory) parameter to supply and is the label of the tape to get read only access to.

Example:

<http://servername/pprime/cgi-bin/MakeAvail?Command=Mount<OLabel=ThisTape>

If the TapeID supplied exists in one of the available pool, the answer is:

```
200<TAB>assigned-uuid
```

where assigned-uuid is the unique identifier assigned to the request.

If the supplied TapeID is not found in the system, the answer is:

```
400<TAB>LTO with label: ThisDoesntExists does not exists
```

GetStatus command

The GetStatus command is used to know if the request “assigned-uuid” is still waiting to be satisfied or it has been executed. TaskID must be supplied, and must match the uuid given as answer returned when issuing the Mount command.

<http://servername/pprime/cgi-bin/MakeAvail?Command=GetStatus&TaskID=assigned-uuid>

Possible answers are:

```
400<TAB>uuid doesn't exist
200<TAB>wait
200<TAB>running
200<TAB>completed
```

400 is self-explaining: the GetStatus has been issued on a non-existing operation.

The “wait” status means that the operation is waiting for its execution. The cause of this is in general that no free resources are available (tape in use by another operation, all of the tape drives are already in use) or, in case a manual operation required, that the physical tape loading has not been done and confirmed by the operator.

The “running” status means that the tape is in loading/positioning status. Normally the operation is going to be completed in few seconds.

The “completed” status means that the agent has completed the load / position / check operation. This status DOES NOT ASSURE that the tape has been made available successfully (It could have failed to an I/O error or FS error). To know the real result of the command it is necessary to issue the GetResult command (see further).

GetResult command

This command is used to know if a MakeAvailble request had an happy ending or not.

<http://servername/pprime/cgi-bin/MakeAvail?Command=GetResult&TaskID=assigned-uuid>

Possible answers are:

```
200<TAB>Success<TAB>local-path-to-LTFS
400<TAB>uuid doesn't exist
400<TAB>Not completed
```

LTFSArchiver 1.0 – Interface specification

500<TAB>Failure

200 Success means that the command was completed without errors, and the LTO content is accessible through the path “local-path-to-LTFS”

400 messages are self-explaining: the GetStatus has been issued on a non-existent or uncompleted operation

500 Means that something went wrong, e.g. FS errors, wrong tape loaded, etc.

To know exactly which error occurred, the generic GetErrorDescr service has to be called:

<http://servername/pprime/cgi-bin/GetErrorDescr?TaskID=assigned-uuid>

The answer will be one of the following:

400<TAB>assigned-uuid is not in fallout status

200<TAB>NumericErrorcode<TAB>Errordescription

Cancel command

This command can be used to delete a Makeavailable request.

An operation can be deleted only if it is still waiting for some resource assignment.

<http://servername/pprime/cgi-bin/MakeAvail?Command=Cancel&TaskID=assigned-uuid>

possible answer are:

200<TAB>assigned-uuid deleted

400<TAB>assigned-uuid doesn't exist

400<TAB>assigned-uuid is not in wait status

Unmount command

This command is used to break off the read only access previously given to a tape through the Mount command.

LTOLabel is the only needed (and mandatory) parameter to supply and is the label of the made available tape.

<http://servername/pprime/cgi-bin/MakeAvail?Command=Unmount&TapeID=ThisTape>

If the TapeID supplied refers to an LTO tape currently available as a local file system, the answer will be:

200<TAB>assigned-uuid

If the supplied TapeID is not currently in a makeavailable status, the answer is:

400<TAB>ThisTape is not a made available tape at time

The commands: **GetStatus**, **GetResult**, **Cancel** and **Resubmit** have the same meaning and behavior as when used in conjunction with a uuid assigned to a Mount operations.

Warning:

If a client is still accessing the made available LTO content, the Unmount request will fail and the request automatically requeued. Be sure to release every access to it before unmounting the tape.

Appendix A - Status and substatus description

All of the interfaces described (except Withdraw in TapeManager) support GetStatus command.

The GetStatus command output may slightly vary according to the interface used, but the common following rules as always applied:

- All the fields are separated by a <TAB> (Hex 09) character
- The first field is a numeric code (see table)
- The second field is a single word describing the main status
- The third field (if present) is used to supply the substatus or more specific information

Numeric code field

Value	Meaning
200	Valid query, message follows
400	Query referring an unexisting uuid
500	Failure

Status field

Status	Meaning
Wait	LTFSArchiver isn't still able to assign all of the requested resources (i.e. all of the tape drives are full)
Starting	LTFSArchiver is moving/preparing the requested tape
Running	LTFSArchiver is executing the requested action
Completed	LTFSArchiver has terminated the requested action

Describing field

Status	Possible values of describing field
Wait	Waiting to be dispatched
	Dispatched, waiting for tape device
	Dispatched, waiting for tape transferring from/to device
Starting	Tape being loaded o positioning
	Tape loaded and ready
Running	<i>Varying according to involved interface</i>
Completed	n/a