LTFSArchiver 1.0

Specification of the interface

The interface is made of a set of *http* services that allows to:

- Manage the tapes
- Query the tapes inside the system
- Book read/write requests and follow the current status and the final result of those requests
- Directly access the tapes content with LTFS

1. Managing the tapes (TapeManager)

Accessible at the URL:
http://<servername>/pprime/cgi-bin/TapeManager

Is used to add / remove LTO tapes to / from tape pools.
In Table 3 are shown all the possible combinations of commands and parameters, N/a means that the parameter is not applicable for that command.

Command	**Parameters**			
	TapeID	**PoolName**	**LTOType**	**Format**
Add	Mandatory	Optional	Mandatory	Optional
Withdraw	Mandatory	Mandatory	N/A	N/A

Table 1 - TapeManager commands and related parameters

Add command

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

TapeID parameter
TapeID must be an unique string of a maximum 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.

LTOType parameter
Must be “LTO5”. This parameter has been created for further use (i.e. future coming LTO6).

Format parameter

The only valid value is “Y” in this case LTFSArchiver will mark as unusable the tape, proceeding as soon as possible to format it according to the configuration parameter PPRIMELTO_LTFSRULE (*)

If omitted or different from “Y” the newly added tape will be immediately available to the system, in this case it is supposed that the tape has been previously formatted. (e.g. with the mkltfs utility that is part of the LTFS software).

Withdraw command

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

2. Querying the tapes (QueryKnownTapes)

The service is accessible at the URL:
http://<servername>/pprime/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 permits the client to handle with several instances of LTFS-Archiver that hence can be deployed in a more complex and larger context.
In this case the client application will need to know (and be configured consequently) the URLs of all possible LTFS Archiver instances to be used.

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

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

Output parameter
Used to specify the output format, Accepted values are “HTML” (default), “TEXT” and “JSON”.

Examples of text output

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

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

Where:
200 means “I've found something”.
For each tape found, the following three values are given in parenthesis:
Label
Free space (in MBytes)
Poolname

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"
      },
      {
         "TapeID":"EX000001",
         "FreeSpace":"1444049",
         "PoolName":"poolC"
      },
      {
         "TapeID":"000001L5",
         "FreeSpace":"1433041",
         "PoolName":"poolA"
      }
   ]
}
```

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

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

3. Archiving operations (WriteToLTO)

The service url is *http://<servername>/pprime/cgi-bin/WriteToLTO* where servername is the name (or ip address) of the host where the software is installed.

Used to archive a file or a directory (recursive) on an LTO Tape. In Table 3 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 3 - WiteLTO 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 the command used to ask for a single file archiving.
WriteFolder is the command 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 LTFS Archiver 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.
When set to Y, an MD5 checksum string will be calculated for the file (or, 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 severely 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     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 waiting request.
A request can be deleted only if its status is “wait”.
http://servername/pprime/cgi-bin/WriteToLTO?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
```

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

possible answer are:

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

GetStatus

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
```

GetResult

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, in case of inquiry about a successfully completed archiving process, 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.
No information about MD5 checksum (even if requested when issuing the WriteFile command) is returned, to get it is necessary to use the JSON mode as explained later.

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

http://servername/pprime/cgi-bin/WriteToLTO?Command=GetResult&Output=JSONTaskID=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 fiolder 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 otput:


```
{
   "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"
      },
```

```
      {
         "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.

That 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

4. Restoring operations (RestoreFromLTO)

The service is accessible at the URL:
http://servername/pprime/cgi-bin/RestoreFromLTO

Used to restore a file or a directory from an LTO Tape. 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 if necessary.

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	**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 4 - RestoreFromLTO commands and related parameters

RestoreFile

This command is used to request e restore 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 name 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 DesPath is /var/mydir/mydata, the directory “var/mydir” has to be an existing one, buat it has not to 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 check has passed, the message will be:

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

GetStatus, Cancel and Resubmit

GetStatus, **Cancel,** and **Resubmit** commands have the same meaning they have in WriteToLTO function.

GetResult

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, the exact error message can be retrieved using GetErrorDescr.

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

5. Giving access through LTFS direct access (MakeAvailable)

The service is accessible at the 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 (in LTFS) on the controlling host and possibly made available remotely through a share. The mounted folder tree is going to be “read-only”.
The client application will be allowed to read data directly from the tape file system.
This will allow services to run processes on archived files without having to copy them locally. Examples of such use include DRACMA (in access), services for regular data integrity check, storage or format migration.

Command	**Parameters**	
	TapeID	**TaskID**
Mount	Mandatory	Output
Unmount	Mandatory	Output
GetStatus	N/a	Mandatory
GetResult	N/a	Mandatory
Cancel	N/a	Mandatory
Resubmit	N/a	Mandatory

Table 5 - MakeAvailable commands and related parameters

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

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

RC will be one of the following:

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

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.

TapeID

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&LTOLabel=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

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-existeingnt 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

“how” the request has been completed, the user has 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
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 cgi 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 waiting makeavailable request.
A request can be deleted only if its status is “wait”. If it is in “running” or “completed” it cannot be deleted.

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 finish the access (in read only mode) to the whole content of a single tape previously made accessible by 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 wasn't in 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.

Caution:
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.