LTFSArchiver 1.1 Configuration and Administration

Summary

Scope:

This document describes in detail the configuration of a LTFSArchiver system and its management including how to start and stop the main service and how to check the most relevant log files.

1. Configuration file(s)

The main configuration file of LTFSArchiver is the plain text file

/opt/ltfsarchiver/conf/ltfsarchiver.conf

Lines beginning with the character "#" are comments while useful lines contain the assigning of specific values to name variables.

The configuration file has four sections each of which defining specific parameters.

Section 1: Operating mode

LTFSARCHIVER_MODE
This parameter sets the operating mode used by LTFSArchiver, accepted values are:

"B": this corresponds to the Mixed mode, LTFSArchiver will hence use both internal (library managed) and external (manually managed) LTO tape devices.

Effects: Each archive request will be evaluated to verify if an existing internal tape (loaded into a managed library) can be used to write the data on it. If a tape is found, the request will be queued in "library mode". If no internal tape can be used, the system will try to find an external usable one (a tape not loaded into a library but known to the system). If such a tape is found, the request will be queued in "manual mode". If even this search fails, the request will be refused (fallout status).

"C": LTFSArchiver will use only internal (library managed) LTO tape devices.

Effects: Each archive request will be evaluated to verify if an existing internal tape can be used to write the data on it. If a tape is found, the request will be queued in "library mode". If no internal tape can be used, the request will be refused (fallout status).

"M": LTFSArchiver will use only external (manually managed) LTO tape devices.

Effects: Each archive request will be evaluated to verify if an existing external tape can be used to write the data on it. If a tape is found, the request will be queued in "manual mode". If no external tape can be used, the request will be refused (fallout status).

Default:
LTFSARCHIVER_MODE="M"

HW_CHANGER_TYPE
Used to set the type of external library used.
For now the only accepted value is "MSL" standing for HP MSL-xxxx tape libraries, possibly will be expanded in the future to support to other libraries.

Default:
HW_CHANGER_TYPE="MSL"

CHANGER_ADDRESSES (Not yet implemented)

Array containing the URLs of the administrator web interface for MSL libraries. If more than one library is used, a blank must be used as a separator. Useful only if the library(ies) will be network connected.

Default:
```
CHANGER_ADDRESSES=( http://198.162.0.2 )
```

`HW_TAPE_TYPE`
Used to set the type of tape device used.
The only accepted value is "`LTO`", that supports LTO5 and LTO6 devices.
This is done for possible future use with other tape types.

Deafult:
```
HW_TAPE_TYPE="LTO"
```

Section 2: Runtime parameters and variables section

`LTFSARCHIVER_TITLE` and `LTFSARCHIVER_VERSION`
These strings are shown on the web interface to present the tool.

default:
```
LTFSARCHIVER_TITLE="LTFS Archiver"
LTFSARCHIVER_VERSION="1.1"
```

`LTFSARCHIVER_HOME` and `LTFSARCHIVER_LOGDIR`
These are the paths to home and log directories

default:
```
LTFSARCHVER_HOME="/opt/ltfsarchiver"
LTFSARCHIVER_LOGDIR="$LTFSARCHIVER_HOME/logs"
```

`LTFSARCHIVER_LOGLEVEL` and `LTFSARCHIVER_DEBUG`
`LTFSARCHIVER_LOGLEVEL` is used to set the verbosity of the log (0=minimum, 4=maximum)
`LTFSARCHIVER_DEBUG` if is set to "1", the log strings (those that have to be printed according to `LTFSARCHIVER_LOGLEVEL` value) will be also sent to standard output (useful when manually launching LTFSArchiver as a script, instead of service)

Defaults:
```
LTFSARCHIVER_LOGLEVEL=1
LTFSARCHIVER_DEBUG=0
```

`LTFSARCHIVER_USER`, `LTFSARCHIVER_GROUP` and `LTFSARCHIVER_DB`
The first two parameters define the user name and group used as UID and GID during the ltfs mount procedure. The `LTFSARCHIVER_USER` has also to be the owner of the database used by LTFSArchiver, whose name has to be set through the `LTFSARCHIVER_DB` variable.

default:
```
LTFSARCHIVER_USER="pprime"
LTFSARCHIVER_GROUP="pprime"
LTFSARCHIVER_DB="ltfsarchiver"
```

`LTFSARCHIVER_LTFSTIMEOUT`
Is the time in seconds LTFSArchiver waits after ltfs unmount command (if mount issued in RW mode) before considering successfully completed the eject/recovery tape.
After an umount command, in fact, ltfs needs some time to physically update the index partition. Removing the tape before the index update [1](even if physically allowed by the device) may cause a filesystem corruption (see also `LTFSARCHIVER_LTFSSYNC` parameter).
This parameter is ignored when unmountig a read only ltfs. Please be careful when considering to lower the default that is set to 120 seconds.

Default:
```
LTFSARCHIVER_LTFSTIMEOUT=120
```

`LTFSARCHIVER_MAXRESTORE_PRIORITY`
Reserved for future use, it is intended to set a priority level on requests.

Default:
```
LTFSARCHIVER_MAXRESTORE_PRIORITY=0
```

`LTFSARCHIVER_INTERRUN`
It is the number of seconds between each agent run. Lower values assure more promptness of the system but also a higher overhead.

Default:
```
LTFSARCHIVER_INTERRUN=60
```

`LTFSARCHIVER_LTFSSYNC`
Determines the mode ltfs will executes the sync of the index partition (see also `LTFSARCHIVER_LTFSTIMEOUT` parameter). Used only when the LTO Tape is mounted in R/W. Accepted values are the same accepted by the system command ltfs mount: *time, close, unmount.*
Usually the *unmount* mode is preferred (the default) because if the archive procedure has to write a lot of files, *time* and *close* modalities can cause frequent index updates and consequently long rewind/update/fast forward operations).
Using *unmount* instead*,* the index update will be executed only once per run.

Default:
```
LTFSARCHIVER_LTFSSYNC="unmount"
```

`LTFSARCHIVER_RULESIZE` and `LTFSARCHIVER_RULEEXTS`
Set these variable to determine the files that should be stored into the index partition (“a” partition) of a LTO5/LTFS tape.

[1] It has been proved that at low level the tape eject command (at least with the HP driver) does not care the fact that some writing is ongoing and can cause severe data corruptions on the tape.

`LTFSARCHIVER_RULESIZE` sets the maximum size of a file that, as a rule, will be stored in the index partition.

`LTFSARCHIVER_ RULEEXTS` array sets the extension(s) of the files that, as a rule, will be stored in the index partition.

Note: In order for a file to be stored in the index partition all the two previous conditions has to be satisfied contemporarily.

Example:
```
LTFSARCHIVER_RULESIZE="2M"
LTFSARCHIVER_RULEEXTS=( txt htm html )
```

This rule will cause storing into the index partition all the files with extension txt, htm and html only when smaller than 2 MBytes in size. Files with different extension or bigger will be stored into the data partition

Default:
```
LTFSARCHIVER_RULESIZE="5M"
LTFSARCHIVER_RULEEXTS=( txt xml jpg )
```

Caveat:
The same criteria will be applied during a directory archiving process to optimize the performance of the process itself. Changing these parameters after having formatted one o more tape may lead to slower archive operations.

`LTFSARCHIVER_RULEOVER` (not implemented yet)
If set to (Y)es mkltfs will format tape allowing a future override of formatting rules.
If set to (N)o mkltfs will format tape without allowing future override.
For future use

Default:
```
LTFSARCHIVER_RULEOVER="n"
```

`LTFSARCHIVER_MAXAVAIL`
This establishes the maximum number of concurrent makeavailable requests. When a tape is loaded due to a “makeavailable” request, it will lock both the tape and the drive device during the time it stays online. If the system has more than one LTO driver, this can be conveniently set to a smaller number, in order to avoid blocking the other queued tasks.

Default:
```
LTFSARCHIVER_MAXAVAIL=2
```

`LTFSARCHIVER_MNTAVAIL`
Is the “base” mountpoint where the makeavailable access point will be created e.g., if the tape “`EX000000`“ is made available, its mount point will be:
```
$LTFSARCHIVER_MNTAVAIL/EX000000
```

Default:
```
LTFSARCHIVER_MNTAVAIL="/mnt/pprime/lto-ltfs"
```

`LTFSARCHIVER_LOCK_LABEL`
A dummy label used to put a tape device in unusable status through the web interface (see "LTFSArchiverWebGUI"). An attempt to use Tapemanager/Add command to add a tape with this label will be refused.

Default:
```
LTFSARCHIVER_LOCK_LABEL="donotuse"
```

`LTFSARCHIVER_DAYSKEEP`
Sets the "time to live" (in days) of log files and database entries.
A background service (script located in /etc/cron.daily) removes from the database the records of successfully completed instances (failed ones are NOT removed) and log files if older than the specified number of days.
This parameter should be set according to specific tracking needs.

Default:
```
LTFSARCHIVER_DAYSKEEP=10
```

`GUESSED_CONF`
It's the name of the auto-configuration file generated by the script `/opt/pprime/sbin/utils/guess_conf.sh` (see further).

Default:
```
GUESSED_CONF=$LTFSARCHIVER_HOME/conf/guessed.conf
```

`LTO_ALLOWED_CODES` and `LTO_ALLOWED_TYPES`
These arrays contain the Density Code(s) that identify the different LTO types and their official names. The n^{th} density code of the first array must match with the n^{th} LTO name.
Default:
```
LTO_ALLOWED_CODES=( 0x58 0x5A )
LTO_ALLOWED_TYPES=( LTO5 LTO6 )
```

`LTO_WATERMARK`
This array must have the same number of elements as the previous two.
Each element represents the minimum available space (in MB) for a tape (according to his known type) to be considered available for a new archive request. This is done because ltfs does not allow to fill a tape more than 98-99% of its declared capacity. If a LTO5 tape has less than about 22 GB of free space, the tape will be mounted in R/O mode even if R/W was specified.

Default:
```
LTO_WATERMARK=( 20200 30200 ) # 30200 for LTO6 is estimated
```

Section 3: Paths to command and db access

This section includes the major paths to the OS commands used for the core activities, should not be modified, except if a non-default version of a command or utility has to be used.

```
CMD_MTX=`which mtx`
CMD_MT=`which mt`
CMD_PSQL=`which psql`
CMD_LTFS=`which ltfs`
CMD_RSYNC=`which rsync`" -va"
CMD_DB="$CMD_PSQL -U $LTFSARCHIVER_USER -d $LTFSARCHIVER_DB -t -c "
CMD_DB_HTML="$CMD_PSQL -U $LTFSARCHIVER_USER -d $LTFSARCHIVER_DB -H -c "
```

Section 4: Library and tape devices (see usage of guess_config further)

`CONF_CHANGER_DEVICES`
String array containing the physical device name(s) of the used library(ies), a blank must be used as a separator.
Sample of a system with two libraries with device name sg5 and sg8:
`CONF_CHANGER_DEVICES=( /dev/sg5 /dev/sg8 )`

Default:
`CONF_CHANGER_DEVICES=( /dev/sg5 )`

`CONF_CHANGER_SLOTS`
String array containing the number of slots (Storage Elements) available for each library.
A blank must be used as a separator. The order must be the same used for libraries with the `CHANGER_DEVICES` array.
Referring to previous sample, and supposing that the sg5 library has 24 slot and the sg8 has 48 slots, the array must be:
`CONF_CHANGER_SLOTS=( 24 48 )`

Default:
`CONF_CHANGER_SLOTS=( 24 )`

`CONF_CHANGER_TAPES`
Array containing the number of tape devices (Data Transfer Element) available for each library. A blank must be used as a separator.
The order must be the same used for libraries with the `CHANGER_DEVICES` array.
Referring to previous sample and supposing that the sg5 library has 2 tapes and the sg8 has 1 tape, the array must be:
`CONF_CHANGER_TAPES=( 2 1 )`

Deafult:
`CONF_CHANGER_TAPES=( 1 )`

`CONF_CHANGER_TAPEDEV_X` (where x= 0, 1... n)
Arrays containing the device address(es) of the tape devices associated to library X (where X is the zero-based index of the library array `CHANGER_DEVICES`). A blank must be used as a separator.
Referring to previous sample and supposing that sg5 library has two tape drives st6 and st7 and sg8 library has a single tape drive st9, the arrays must appear as:

```
CONF_CHANGER_TAPEDEV_0=( /dev/st6 /dev/st7 )
CONF_CHANGER_TAPEDEV_1=( /dev/st9 )
```

Default
```
CONF_CHANGER_TAPEDEV_0=( /dev/st0 )
```

CONF_MANUAL_TAPEDEV
Is the array containing the device address(es) of ALL the external tape devices connected to the LTFSarchiver system. A blank must be used as a separator.

Sample:
```
CONF_MANUAL_TAPEDEV=( /dev/st2 /dev/st3 )
```

Default:
```
CONF_MANUAL_TAPEDEV =( /dev/st1 )
```

CONF_BACKEND
Array containing information about the “backend emulator” that will be used by LTFS commands. Because different LTFS builds use different backends and device default values, a correct setting of this array is necessary for LTFSArchiver to manage correctly the tape devices from different vendors.
The array data are stored in the form *“scsi-tapedevice-address” “backend-value”*
See Appendix B for a deeper discussion about backend and device name.

Sample:
```
CONF_BACKENDS=( /dev/st0 ibmtape /dev/st1 ltotape /dev/st2 orcltape)
```

Default :
```
CONF_BACKENDS=( /dev/st0 ltotape /dev/st1 ltotape )
```

Guessed_config file

Trying to make simpler the parameters set up in section 4 of the configuration file, it is possible to execute a script based on the *sg_map* system utility, please ensure that *sg_map* rpm package is installed otherwise the script will fail.

Launching the script.

Login as root user and run: `/opt/pprime/sbin/utils/guess_conf.sh`

Each time the script finds a device that announce itself as a library or as a tape, a message will be printed. An example follows:

```
Step 1: Tape changers
Tape loader(s) found: 1
scsi device /dev/sg6 appears to be a library
Step 2: Tape devices
Tape device(s) found: 2
Step 3: Changers/Tape associations
scsi device /dev/st1 appears to be a tape owned by /dev/sg6 library
Step 4: Standalone tapes
scsi device /dev/st0 appears to be an external tape
```

Finally, the guessed configuration found will be compared with the existing (if existing) guessed configuration file (see `GUESSED_CONF` variable) and shown:

```
The guessed conf doesn't differ from running conf:
--------------------------------------------------
CONF_BACKENDS=( /dev/st0 ltotape /dev/st1 ltotape )
CONF_CHANGER_DEVICES=( /dev/sg6 )
CONF_CHANGER_SLOTS=( 48 )
CONF_CHANGER_TAPEDEV_0=( /dev/st1 )
CONF_CHANGER_TAPES=( 2 )
CONF_MANUAL_TAPEDEV=( /dev/st0 )
--------------------------------------------------
```

Updating the configuration

If the guessed conf file doesn't exist or differs from active configuration, the user is asked whether to update the configuration or not. Answering “Y”, the old `GUESSED_CONF` file will be overwritten by the new one. Please pay attention to this delicate configuration step and refer to Appendix A for more information.

2. Starting/stopping the service

The real actions like read and write operations are performed by a single system service called *ltfsarchiver* that works sequentially by carrying out jobs queued on the database.
The service should be configured to be started at run-levels 3 and 5, using 90 as start priority (at least at a lower priority than Postgresql and Apache daemons).

Status

To check the status of the agent use the command:

```
service lftsarchiver status
```

Possible answers are:

```
ltfsarchiver is running
ltfsarchiver is stopped
[WARNING] ltfsarchiver (RUNPID) is running, but pidfile is missing.
[WARNING] ltfsarchiver not running, but pidfile (FOUNDPID) exixts
```

RUNPID is the process id (PID) of the running instance.
FOUNDPID is the process id (PID) found into the pidfile.

Start

To start the agent use the command:

```
service lftsarchiver start
```

Start will be actually attempted only if no other instance is currently running AND pidfile does not exist.

Possible answers are:

```
ltfsarchiver started
ltfsarchiver (RUNPID) is already running
[WARNING] ltfsarchiver not running, but pidfile (FOUNDPID) exixts
[WARNING] ltfsarchiver (RUNPID ) is running, but pidfile is missing
```

Stop

To stop the agent, supply the command:

```
service lftsarchiver stop
```

Stop will be actually attempted only if an instance is currently running AND pidfile matches the one of the running instance, possible answers are:

```
ltfsarchiver stopped
ltfsarchiver already stopped
[WARNING] ltfsarchiver not running, but pidfile exists... removing
[WARNING] Running process pid (RUNPID) didn't match pidfile (FOUNDPID)
[WARNING] ltfsarchiver (RUNPID ) is running, but pidfile is missing
```

NOTE: stopping the agent will NOT break currently active requests, but simply prevents LTFSArchiver from serving further requests. Pending requests (wait status) at the moment of the switch off are not lost but executed when the service will be re-started. If a tape has been put in "available online" status, it will be left mounted even if an umount command has been issued but still has not been executed at the time of the switch off.

3. Log files

LTFSArchiver logs its activities using several files, all of them are placed in the directory specified with the `LTFSARCHIVER_LOGDIR` parameter.

The main logfile has the following name:

`ltfsarchiver.YYYYMMDD.log`

where `YYYYMMDD` is the date when tape_agent started.
It reports new incoming requests and if they are going to be processed or not (and the reason in the last case). When a request is processed the corresponding activity is reported in a separate log file named:

`tape_agent_sssssssss.log`

where **`sssssssss`** is the time when tape agent started in "seconds since 1970/01/01"

The log verbosity level is determined by the parameter `LTFSARCHIVER_LOGLEVEL`

4. Appendix A – Guessed configuration

The installing operator is advised that the result of the `guess_conf` script may, in some cases, lead to wrong configurations.

It happened with a HP MSL4048 library with two tape drives, to appear with a single drive, while the second internal tape was detected as an external one.

This happens because the library arm is accessible through the same FC bus assigned to the first drive OR the second tape device.

Here follows an example of a real `guess_config` output obtained with the guessing utility on a server with an HP MSL 4048 with two LTO drives (`/dev/st0 and /dev/st1`) and an external HP desktop LTO drive (`/dev/st2`).

```
CONF_CHANGER_TAPEDEV_0=( /dev/st0 )
CONF_CHANGER_DEVICES=( /dev/sg6 )
CONF_CHANGER_SLOTS=( 48 )
CONF_CHANGER_TAPES=( 2 )
CONF_MANUAL_TAPEDEV=(  /dev/st1 /dev/st2 )
```

While the right configuration should be:

```
CONF_BACKENDS=( /dev/st0 ltotape /dev/st1 ltotape /dev/st2 ltotape )
CONF_CHANGER_TAPEDEV_0=( /dev/st0 /dev/st1 )
CONF_CHANGER_DEVICES=( /dev/sg6 )
CONF_CHANGER_SLOTS=( 48 )
CONF_CHANGER_TAPES=( 2 )
CONF_MANUAL_TAPEDEV=( /dev/st2 )
```

When the script detects that the number of tape devices associated to a media changer is lower that the value declared by the changer itself, a warning message will be displayed:

```
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
WARNING!
Changer /dev/sg6 has 2 tape device(s),
but the guessed conf script detected 1 tape device(s)
maybe one of the following device(s) are actually managed by /dev/sg6
--> /dev/st1
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
```

Consider the guess_conf script as an aid to configure the system, not as fully automatic configuration tool.

5. Appendix B – LTFS devname and backend

According to LTFS documentation, LTFS needs to know “who” and “how” to interact with a tape device.

“who” is the devname: the tape SCIS name mapped from generic /dev/sgX (e.g. /dev/stX)

“how” is the backend: the manufacturer’s specific driver library (e.g. ltotape)

Several manufacturer like HP or IBM have released their own LTFS version, each using different default values for these parameters.

Additionally, IBM uses a different device naming (/dev/IBMtapeX instead of /dev/stX), and a specific rpm package (lin_tape) is needed to create the correct device name.

The default devname and backend values used by some LTFS distribution are reported:

Distribution	Devname	Backend
HP	/dev/stX	ltotape
IBM	/dev/IBMtapeX	ibmtape
Oracle	/dev/stX	orcltape
Quantum	/dev/stX	ltotape

The guess_config script tries to get the tape vendor name through a “tapeinfo” call and finally creates the CONF_BACKENDS array according to the returned values.

6. Appendix C - Compatibility tests

Interoperability tests (even not exhaustive) have been done considering the tape devices, media changers and LTFS “flavors” reported in the below tables where also some important notes appear.

LTFS Versions

Distribution	LTFS version	Index Spec. Version	Note
HP	1.2.2	2.0.0	No support for orcltape
	2.0.0	2.0.0	
IBM	1.3.0	2.0.1	No support for ltotape/orcltape
Oracle	1.2.0	2.0.0	
Quantum	2.0.0	2.0.0	No support for orcltape

LTO Tape devices

Manifacturer	Model (reported from tapeinfo)	Revision	Conn.	Notes
HP	Ultrium 5-SCSI	I3BW	FC	Internal - MSL G3 Series
HP	Ultrium 5-SCSI	Z3ED	SAS	Desktop model
IBM	ULT3580-HH5	BBNF	SAS	Desktop model

Media changers

Manifacturer	Model (reported from tapeinfo)	Revision	Conn.	Notes
HP	MSL G3 Series	5.00	FC	MSL 2024 (1 drive)
HP	MSL G3 Series	8.10	FC	MSL 4048 (2 drives)