

LTFSArchiver 1.0

Configuration and Administration

In this document are described in detail the configuration of the system and its management. It is explained for example how to start and end the system service and how to keep track of what is going on through log files.

1. Configuration file(s)

The main configuration file of LTFSArchiver is

`/opt/pprime/conf/pprimelto.conf`

The config file has three sections:

Section 1: Runtime parameters and variables section

`PPRIMELTO_TITLE` and `PPRIMELTO_VERSION`

They are the strings the web interface shows in the main menu

default:

`PPRIMELTO_TITLE="PrestoPrime LTFS Archiver"`

`PPRIMELTO_VERSION="1.0 Beta"`

`PPRIMELTO_HOME` and `PPRIMELTO_LOGDIR`

Path to homedir and logdir of the LTFSArchiver application

default:

`PPRIMELTO_HOME="/opt/pprime"`

`PPRIMELTO_LOGDIR="$PPRIMELTO_HOME/logs"`

`PPRIMELTO_LOGLEVEL` and `PPRIMELTO_DEBUG`

`PPRIMELTO_LOGLEVEL` is used to set the verbosity of the log (0=minimum, 4=maximum)

If `PPRIMELTO_DEBUG` is set to "1", the log strings (those that have to be printed according to `PPRIMELTO_LOGLEVEL` value) will be also sent to standard output (useful when manually launching LTFSArchiver as a script, instead of service)

default:

`PPRIMELTO_LOGLEVEL=1`

`PPRIMELTO_DEBUG=0`

`PPRIMELTO_USER`, `PPRIMELTO_GROUP` and `PPRIMELTO_DB`

The first two parameters define the name of the user and group used as UID and GID during the ltfs mount procedure.

The `PPRIMELTO_USER` has also to be the owner of the database used by LTFSArchiver, whose name has to be set through the `PPRIMELTO_DB` variable

default:

```
PPRIMELTO_USER="pprime"  
PPRIMELTO_GROUP="pprime"  
PPRIMELTO_DB="ltfsarchiver"
```

PPRIMELTO_LTFSTIMEOUT

It's the number of seconds LTFSArchiver has to wait after ltfs' unmount command (from RW mount mode) has been successfully completed before ejecting/recovering tape.

After an unmount command, in fact, ltfs needs some time to physically update the index partition.

Removing the tape before the index update (even if physically allowed by the device) may cause a filesystem corruption(see also PPRIMELTO_LTFSSYNC parameter)

This parameter is ignored when unmounting a R/O ltfs.

Default:

```
PPRIMELTO_LTFSTIMEOUT=120
```

PPRIMELTO_MAXRESTORE_PRIORITY

Reserved for future uses, it will allow to set a priority rate between restore and other requests.

Default:

```
PPRIMELTO_MAXRESTORE_PRIORITY=0
```

PPRIMELTO_INTERRUN

It's the number of seconds between each agent run.

Default:

```
PPRIMELTO_INTERRUN=60
```

PPRIMELTO_LTFSSYNC

Determines the mode in which ltfs will execute the sync of the index partition (see also PPRIMELTO_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)

The unmount mode is preferred (is the default) because if the archive procedure has to write a lot of files, "time" and "close" values can cause frequent index updates and consequent long rewind/update/fast forward operations)

Using "unmount", the index update will be executed only once per run.

Default:

```
PPRIMELTO_LTFSSYNC="unmount"
```

PPRIMELTO_LTO5LOW

It represents the lower space limit (in MB) for a tape be considered available for a new archive request.

This is because ltfs doesn't allow to fill a tape more than 98-99%. If a 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:

```
PPRIMELTO_LTO5LOW=22000
```

PPRIMELTO_LTFSRULE

It is the string sent to the system command *mklufs* command when formatting a new tape. This is done to redirect the archiving of files smaller than what specified (1MB as default) to the index partition instead then data partition of the tape.

Please note that changing this parameter can lower the performances when archiving a whole folder.

Default:

PPRIMELTO_LTFSRULE='size=1M'

PPRIMELTO_MAXAVAIL

Max concurrent active instances of 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 runs. Please remember to leave at least one drive free to avoid blocking of the elaborations.

Default:

PPRIMELTO_MAXAVAIL=2

PPRIMELTO_MNTAVAIL

It's 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:

\$PPRIMELTO_MNTAVAIL/EX000000

Default:

PPRIMELTO_MNTAVAIL="/mnt/pprime/lto-ltfs"

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=\$PPRIMELTO_HOME/conf/guessed.conf

Section 2: Paths to command and db access

This section should'nt be modified, except if a non-default version of a command or utility has to be used.

MTX_CMD=`which mtx`

MT_CMD=`which mt`

PSQL_CMD=`which psql`

LTFS_CMD=`which ltfs`

RSYNC_CMD=`which rsync` " -va"

DBACCESS="\$PSQL_CMD -U \$PPRIMELTO_USER -d \$PPRIMELTO_DB -t -c "

Section 3: Operating mode

PPRIMELTO_MODE

Sets the operating mode used by LTFSArchiver,

Accepted values are:

"B": LTFSArchiver will 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, LTFSArchiver will try to find an external usable one (a tape not loaded into a library but known by the system). If such a tape is found, the request will be queued in "manual mode".

If also 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:

PPRIMELTO_MODE="M"

CHANGER_TYPE

Used to set the type of external library used.

The only accepted value is "MSL", standing for HP MSL-20xx tape libraries.

In the future it's expected to extend support to other libraries.

Default:

CHANGER_TYPE="MSL"

TAPE_TYPE

Used to set the type of tape device used.

The only accepted value is "LTO", that supports LTO5 devices.

This is kept for compatibility with older version of data archivers.

Default:

TAPE_TYPE="LTO"

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)

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

CHANGER_DEVICES

String array containing the 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:

CHANGER_DEVICES=(/dev/sg5 /dev/sg8)

Default:

CHANGER_DEVICES=(/dev/sg5)

CHANGER_SLOTS

String array containing the number of slot (Storage Element) available for each library.

A blank must be used as a separator.

Their order must be the same in which libraries has set in CHANGER_DEVICES array.

Referring to previous sample, suppose that sg5 library has 24 slot, while sg8 has 48 slots.

Array must appear as:

CHANGER_SLOTS=(24 48)

Default:

CHANGER_SLOTS=(24)

TAPE_SLOTS

String array containing the number of tape devices (Data Transfer Element) available for each library.

A blank must be used as a separator.

Their order must be the same in which libraries has set in CHANGER_DEVICES array.

Referring to previous sample, suppose that sg5 library has 2 tapes, while sg8 has 1 tape.

Array must appear as:

CHANGER_SLOTS=(2 1)

Default:

CHANGER_SLOTS=(1)

CHANGER_TAPE_DEV_X (where x= 0, 1... n)

Array containing the device address(es) of the tape devices associated to library X (where X is the zero-based index of array CHANGER_DEVICES indicating the library itself)

A blank must be used as a separator.

Referring to previous sample, suppose that sg5 library's tape are st6 and st7 has 2 tapes, while sg8 library single tape is st9.

Arrays must appear as:

CHANGER_TAPE_DEV_0=(/dev/st6 /dev/st7)

CHANGER_TAPE_DEV_1=(/dev/st9)

Default

CHANGER_TAPE_DEV_0=(/dev/st0)

MANUAL_TAPE_DEVICES

Array containing the device addresse(s) of ALL external tape device connected to the LTFSarchiver server.
A blank must be used as a separator.

Sample:

```
MANUAL_TAPE_DEVICES=( /dev/st2 /dev/st3 )
```

Default:

```
MANUAL_TAPE_DEVICES=( /dev/st1 )
```

Guessed_config file

Trying to make simpler to set parameters in the fourth section of the configuration file, it is possible to execute a script based on the sg_map system utility; please ensure that sg_map package has been installed or the script will fail.

Launching the script.

Login as root and run: `/opt/pprime/sbin/utis/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.

```
scsi device /dev/sg5 appears to be a library
scsi device /dev/st1 appears to be a tape owned by /dev/sg5 library
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:

```
-----
CHANGER_TAPE_DEV_0=( /dev/st1 )
CHANGER_DEVICES=( /dev/sg5 )
CHANGER_SLOTS=( 24 )
TAPE_SLOTS=( 1 )
MANUAL_TAPE_DEVICES=( /dev/st0 )
-----
```

Updating the configuration

If the guessed conf file doesn't exist or differs, a prompt will be issued, asking if the file has to be updated.

Answering "Y", the old GUESSED_CONF file will be overwritten by the new one.

2. Starting/stopping service

The service should be configured to be started at runlevels 3 and 5, using 90 as start priority (or, at least, at a lower priority than postgresql and apache daemons)

Status

To check the status of the agent, supply use the command:

```
service pprimelto status
```

Possible answers are:

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

RUNPID is the PID of the running instance

FOUNDPID is the PID found into the pidfile

Start

To start the agent, supply the command:

```
service pprimelto start
```

Start will be actually attempted only if no other instance is currently running AND pidfile doesn't exists

Possible answers are:

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

Stop

To stop the agent, supply the command:

```
service pprimelto 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:

```
pprimelto stopped
pprimelto already stopped
```

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

Please note that stopping the agent will NOT break currently active requests, but simply prevents LTFSArchiver from trying to satisfy further requests. The requests in wait status will be executed when the service will be started again.

3. Log files

LTFSArchiver logs its activities in using several files; all of them are placed in the directory specified in the `PPRIMELTO_LOGDIR` parameter.

`pprimelto.YYYYMMDD.log`

where `YYYYMMDD` is the date when `tape_agent` started

This is the main logfile.

It reports if some the incoming requests were found and if they are going to be processed or not (and why they're if not processed).

When a request is processed, its activity is reported into the log file:

`tape_agent_X_stY.YYYYMMDD.log`

where `X` staysnds for "type of request" (W=Archive,R=Restore,A=MakeAvalalble)

`stY` staysnds for basename of the tape device involved.

`YYYYMMDD` is the date when `tape_agent` started

So, for example an archive request run on `/dev/st0` tape device on August, 6th 2012 will be logged into the file

`tape_agent_W_st0.20120806.log` file

The log level is determined by the parameter `PPRIMELTO_LOGLEVEL`