



## **LTFS and SpaceLTO on Tape Layouts Comparison**

***Matthew Daubney***

*Chief Engineer, GBLabs*

### **Summary**

LTFS is an on tape file system designed by IBM to make the transfer of large files on tape between systems easier. The method it uses to achieve this is similar to an on disk file system, but altered to take in account the limitations of the tape. Unfortunately the standard leaves a lot of decisions up to the client implementation which can be dangerous to the integrity of the original tape. The SpaceLTO on tape layout is a much simpler system designed to preserve the tapes for a long period as part of an archive.

## **LTFS**

An LTFS tape can be led out in either one or two partitions, most current implementations use a dual partition layout. The first partition of the tape, the index partition, contains only metadata, but can contain some small files as well. The second partition is the data partition, this contains data extents, but will also contain some index files.

At the beginning of the first partition is an XML file that describes the tape as a whole and includes some metadata on the implementation that created the initial layout.

### ***LTFS Label***

This is an XML document that describes the LTFS volume and the partition that it is contained on. It contains;

1. The version of the LTFS specification that this tape conforms too
2. The name of the manufacturer and program that created the LTFS volume
3. The time that the tape had been formatted
4. The block size for the partition
5. The partition layout of the tape
6. The identifier of the partition that contains this file

### ***Index Files***

The index file is an XML document that describes the data on the LTO.

Each index file contains a few pieces of information to identify itself and ensure consistency of the LTFS volume. These are;

1. The generation number of the index
2. A self pointer containing the partition and block number of where this index file is
3. A back pointer.

The generation number records the age of the index file in comparison to other indexes on the tape.

Each index XML file contains a full copy of the contents of the tape. Folder structures are also stored inside the index file. It also contains any extended metadata that the client implementation can access. Should a different client not recognise this extra metadata it is to pass it silently.

After each write operation is completed a file marker is placed and then an index file is added. The index file is also placed in the index partition. If an index file is on the data partition it should have a back pointer to the previous index file on the data partition. If the index file is the last index file on the index partition, it should have a back pointer to the index of equivalent generation in the data partition.

### ***Data Extents***

Data extents are just areas of the tape that contain the raw bytes of a file. The maximum size of the extents is set when the file system is created and is described in the LTFS label at the beginning of the partition. Data extents can only be placed on the data partition. Data extents can be shared among files.

## ***Implementation Notes***

A lot of how LTFS might function is left to the implementation and therein lies its weakness. According to Wikipedia, at the time of writing there are 5 different implementations of LTFS;

- IBM LTFS-SDE (Single Drive Edition)
- IBM LTFS-LTE (Library Tape Edition)
- Oracle Storagetek Linear Tape File System, Open Edition
- HP Linear Tape File System
- Quantum LTFS

Of these, only IBM-LTFS-SDE supports Microsoft Windows, all of them support Linux and Mac with the exception of Oracle which only supports Oracle Linux.

In order to reduce wear on the tapes, the client implementation must implement caching of the index file, only writing it to tape when required. Each file is also kept separate, gaining an item in the index file. This would require significant amounts of resources to contain indexes of many file/folder file systems or a database back end to the client implementation.

The extent based layout can lead to fragmentation over time. It would be up to an implementation to resolve this.

Shared blocks can lead to the tape having to move forward and backwards significantly, this could lead to scrubbing of the tape.

## **SpaceLTO**

A SpaceLTO tape is split up into tar files. The last tar file on the tape contains an index file formatted as an XML document. The layout of the files into tar files is based on rules designed around the needs of the media industry.

### ***Index Files***

The index file is an XML document that describes the data on the LTO. This file will contain;

- The name of the tape
- The date the tape was written
- The serial number of the system that wrote the tape
- The version of the XML layout
- The blocksize of the tape
- A description of each file contained on the tap
- Any tags associated with the tape.

### ***Tar Files***

Each file is stored inside an individual tar file with the following exceptions;

- Any folder structure that is recognised as the contents of a P2 card
- Any folder structure that is recognised as XDCam HD or Ex
- Image sequences and dpx files

These alterations are designed to reduce time to copy to and from the tape and to reduce the wear on the tape itself. It also ensures that related files do not get scattered across the tape.

### ***Implementation Notes***

When a SpaceLTO tape is placed into a member of the SpaceLTO family it immediately reads the index file and adds the tape to the library of that unit. From that point on it will use it's internal database rather than the on tape index to address the files.

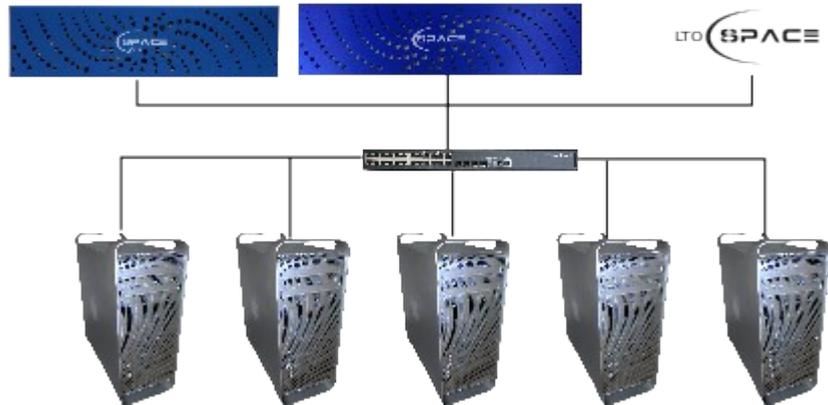
Data is written onto the tape as close to the recommended speed of the LTO drive as is possible from the source location. If this is the internal RAID of a SpaceLTO device or from a Space storage system, this should be close to the limit of the drive. This helps keep the tape in optimum condition.

## ***Workflow Examples***

### **Notes**

In order to show off the strengths and weaknesses of these LTO layouts the following example workflows have been created. These workflows may not be applicable to your own situation.

## Workflow 1: Archiving Old Footage



*Illustration 1: Simple Space Archiving Setup*

### **Workflow With Space**

Footage is edited on the Mac Pro's from the Space unit, backed up onto the Space Echo. Once the footage is finished with, it is archived onto an LTOSpace system and the tape archived safely.

When the footage needs to be restored, it is requested through the LTOSpace and the relevant tape is inserted. It is then restored straight back onto the Space head unit.

### **Workflow With LTFS**

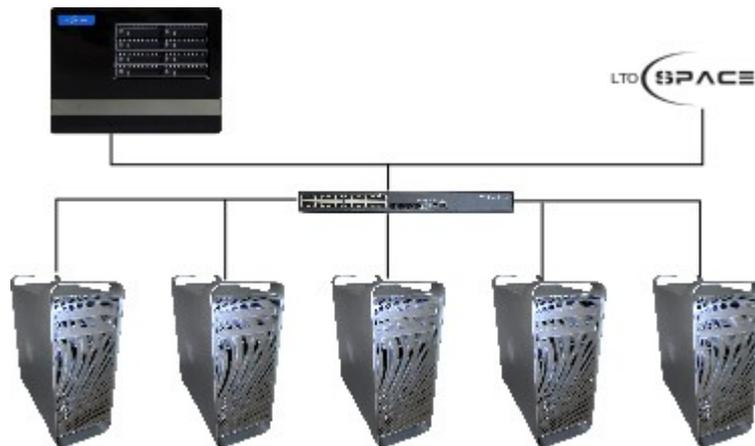
Footage is edited on the Mac Pro's from the Space unit, backed up onto the Space Echo. Once the footage is finished with, it is archived onto an LTO drive locally connected to one of the Mac Pro's.

When the footage needs to be restored, the tape has to be found through a manual process and restored through the Mac Pro.

### **Notes**

In this situation, the native Space LTO format is better. The data is kept in the LTOSpace's internal database and the tape can be found and restored quickly and easily. The tricks implemented in the LTOSpace format ensure that the tape is treated as carefully as possible, ensuring the longevity of the LTO tape.

## Workflow 2: Transferring Footage From One Office to Another



*Illustration 2: Small Branch Suite*

### **Workflow With Space**

Footage is edited on the Mac Pro's from the MiniSpace unit. Once the product is complete, it is placed onto an Mini Space LTO system and the tape is shipped to the other site.

When the other site receives the tape it is either archived, or loaded into another LTOSpace product and placed onto other storage.

### **Workflow With LTFS**

Footage is edited on the Mac Pro's from the MiniSpace unit. Once the footage is finished with, it is archived onto an LTO drive locally connected to one of the Mac Pro's.

The LTO is then shipped to the other site and loaded via another device.

### **Notes**

In this situation, LTFS is better. It would still need to be archived by the other site, and if the tape is to be reused for another transfer, it would be advisable to archive it's contents and then reformat the tape.

<b>Term</b>	<b>Definition</b>
Data Extent	A grouping of data relating to a file. Equivalent to a data block on a regular file system.
Metadata	Data that describes data.
XML	Extensible Markup Language
LTFS	Linear Tape File System
Scrubbing of the tape	The process of a tape degrading from being pushed backwards and forwards across the reading head.

Document Control  
Sheet

19/11/2012  
Document Created by Matt Daubney  
Suggested changes by Tom Sheldon

Creator Signature

Design Signoff

Technical Signoff

Corporate  
Signoff