XenData Tape Format

XenData software archives files to data tape using an extension of the POSIX tar format. The approach take by XenData ensures that files can be restored from tape by a wide range of third party software, including native Unix and Linux operating systems as well as Windows Services for Unix.

The extension to POSIX tar supports file version control, spanning of files across two or more data tapes, partial restore of files, file renaming and file deletion. Each tape is self-describing and the XenData extension supports a tape contents catalog which allows tapes to be quickly transferred between archive systems.

This paper gives a short overview of XenData's extended POSIX tar format.

XenData

About XenData

XenData is a software company that focuses or digital archive solutions - archiving to and restoring from data tape such as LTO. Based on a standard file system, XenData Archive Series software makes the digital archive appear as a Windows logical drive letter which provides easy integration with other standards-based systems. Files are written to data tape using the open standard POSIX tar format. This is important for long term archives because it means that files can be restored for decades to come using a wide range of native Linux and Unix operating systems, Microsoft Services for Unix, as well as using XenData software.

Last updated: 11 November 2009

1. POSIX tar Format – the Basics

The term 'tar' is derived from '**t**ape **ar**chive' and is both a format for writing files to data tape and the name of a program used to handle such tapes. The tar format first appeared in Version 7 of AT&T UNIX in 1979 and subsequently a number of variants were developed. The IEEE developed a standardized software program called 'pax' that is an interchange utility which defines POSIX tar. The pax utility is specified in the IEEE 1003.1 standard, 2004 Edition. Today POSIX tar, as defined by pax, is the preferred implementation because it is widely used and supports files and file names of unlimited size as well as Unicode file names.

The POSIX tar format defines how a file of any type will be written to a data tape drive. The POSIX tar format includes a header with file system metadata such as pathname, owner, permissions, etc. The end of the tar archive is indicated by two records consisting entirely of zero bytes. The POSIX tar format can be used to archive and restore any file type to data tape, ranging from a MOV file to a MXF file or even a Word .doc file.

POSIX tar is the leading interchange standard for data tapes because files can be written and restored by a large number of Linux and Unix native operating systems. Windows Server running Services for Unix also writes and restores files to data tape using the POSIX tar format. An example of file interchange is provided in Appendix A which describes the steps involved when using Red Hat Linux to restore files from an LTO-4 tape cartridge, written using XenData Archive Series software.

2. XenData Extensions - Overview

XenData has extended the POSIX tar format to provide a number of significant advantages:

- Partial file restore
- Ability to span a file across multiple data tapes
- Comprehensive file version control
- Support for file deletion and rename
- Tape contents catalog for rapid transfer of tapes between archives systems

These capabilities are added in such a way that files written to a data tape using XenData extended tar can be fully restored using any program capable of reading the standard POSIX tar format.

A data tape written using the XenData extensions is self describing: no additional data is required to recover the contents and present them in the portion of the file system that was archived to that specific tape. When a tape contents catalog is implemented, each tape is not only self describing but may be transferred to another archive system very quickly.

3. XenData Extensions

3.1 Option for File Fragmentation Added

The term 'file fragmentation' refers to the process of breaking large files into smaller, more manageable units, termed 'fragment files'. The XenData extensions add the option to fragment files and to select the fragment size. This capability supports partial file restore and the ability to span files across multiple data tapes.

3.2 Extension to the File Name Added

The XenData extensions extend the file name written to tape using the following convention: **-f-g-v**

where

- f is the file fragment number. (This is always 0 for non-fragmented files)
- g is the generation number. (This is 0 for files unless a file of the same name and path has previously been deleted.)
- v is the version number. (This is always 1 for the first version of a file.)

This capability supports file fragmentation and file version control.

3.3 Rename and Delete Records Added

The XenData extensions support the writing of file rename and delete records as XML files written to tape using the POSIX tar format. This capability supports file system file rename and file delete functions.

3.4 Tape Contents Catalog Added

The XenData extensions support the writing of a tape contents catalog to the end of a tape. The contents catalog describes the file system metadata for all files and rename and delete records written to the tape. This capability supports rapid transfer of tapes between archive systems.

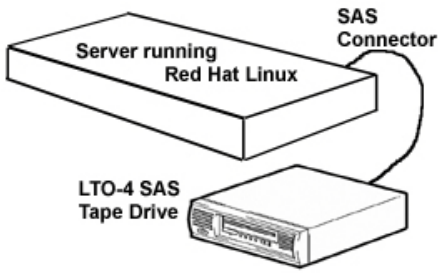
The tape contents catalog is written to tape when the tape is finalized. After finalization, no additional files may be written to the tape.

Appendix A: Restoring Files from a XenData Tape Cartridge Using Native Linux

This appendix describes the steps involved when using Red Hat Linux to restore files from an LTO-4 tape cartridge, written using XenData Archive Series software.

Example Linux System Configuration Used for File Restores

The system configuration that was used to restore files written using the XenData extended POSIX tar format is illustrated below.



In our tests we used the following:

- Server hardware: IBM x3655 server with LSI SAS3801X SAS host bus adapter
- Server Operating System: Linux Red Hat v 5.3 64 bit with the mt-st-0.9b-2.2.2.x86_64.rpm package installed. The MT-ST package is required for tape drive access and control.
- Tape drive: Dell LTO-4 external SAS (manufactured for Dell by IBM)
- Tape cartridge: LTO-4 tape written on a Windows server using XenData Archive Series software

We checked to ensure that the tape drive was recognized by the Linux OS using the following command:

```
#cat /proc/scsi/scsi
```

We obtained the following output which confirmed that the tape drive was recognized:

```
Host: scsi7 Channel: 00 Id: 00 Lun: 00

Vendor: IBM      Model: ULTRIUM-HH4      Rev: 8191

Type:      Sequential-Access                    ANSI SCSI revision: 03
```

We then checked the status of the drive using the following command;

```
#mt -f /dev/nst0 status
```

This resulted in the following output:

```
SCSI 2 tape drive:

File number=0, block number=0, partition=0

Tape block size 0 bytes.  Density code 0x46 (no translation).
```

Soft error count since last status=0

General status bits on (41010000):

BOT ONLINE IM_REP_EN

Example File Restore Procedure

We first created a directory for file restores and moved to that directory, as follows:

#mkdir /tape

#cd /tape

We then calculated the blocking factor. This is the block size used to write to the tape in bytes divided by 512. In our case, a 64 KB block size had been used by the XenData software which meant that the blocking factor was 128.

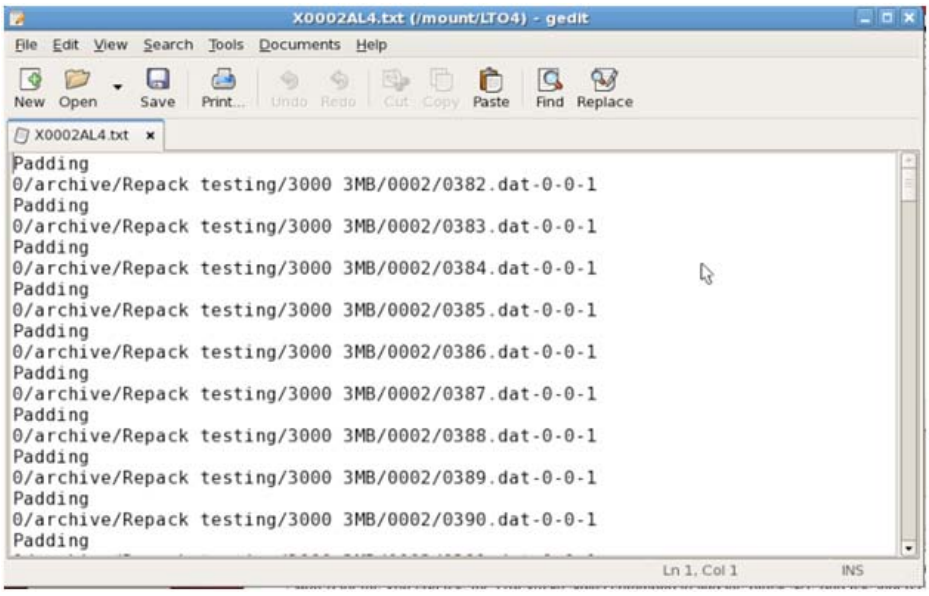
The LTO-4 tape that we had inserted in the drive had a barcode X0002AL4. We created a file called 'X0002AL4.txt' that contained a list of the contents of the tape using the following command:

#tar --blocking-factor=128 -tf /dev/nst0 > X0002AL4.txt

Note if the blocking factor is incorrect, you will obtain an error message: 'Cannot read: Cannot allocate memory'.

After the tape contents had been read, we rewound the tape: #mt -f /dev/nst0 rewind

We viewed the contents file (X0002AL4.txt in our case) using gedit, as shown below:



The XenData software appends characters to the file name using the following convention: -f-g-v

where

f is the file fragment number. (This is always 0 for non-fragmented files)

g is the generation number. (This is 0 for files unless a file of the same name and path has previously been deleted.)

v is the version number. (This is always 1 for the first version of a file.)

We then selected and restored a file (the first generation and first version of 0390.dat in the \archive\Repack testing\3000 3MB\ folder) using the following command:

#tar --blocking-factor=128 -xvf /dev/nst0 “0/archive/Repack testing/3000 3MB/0002/0390.dat-0-0-1”

The quotes must be used if the folder names contain spaces. As this command will scan the whole tape looking for all files with this combination of folder and file name, a Ctl-C can be used to stop the process after the file has been restored.

We then restored all files on the tape using the following command:

#tar --blocking-factor=128 -xvf /dev/nst0

Concluding Comments

This technical note describes the steps that should be employed using the Red Hat Linux OS to restore files from a tape written using XenData software. Of course, the restore process is not as convenient and straight forward as using XenData software. However, variants of Linux and UNIX capable of restoring POSIX tar files are likely to be available for decades to come, ensuring that tapes written using XenData Archive Series software may be restored using third party software for many years in the future.