



An Oracle White Paper  
May 2011

# Redefining Tape Usage with StorageTek Tape Tiering Accelerator and StorageTek In-Drive Reclaim Accelerator

Introduction .....	1
The Tape Storage Space Problem .....	3
The Tape Organization Problem.....	4
The Tape Access Problem .....	5
Why Previous Options Don't Work As Well.....	5
The StorageTek In-Drive Reclaim Accelerator Solution .....	5
Tape Reclamation with the StorageTek In-Drive Reclaim Accelerator	7
The StorageTek Tape Tiering Accelerator Solution for Tape Organization and Access.....	8
StorageTek In-Drive Reclaim Accelerator and StorageTek Tape Tiering Accelerator Implementation .....	11
Conclusion .....	13

## Introduction

This paper explains how Oracle's StorageTek Tape Tiering Accelerator and StorageTek In-Drive Reclaim Accelerator work in conjunction with Oracle's StorageTek T10000C tape drive to address performance and scalability challenges typically encountered in today's tape archive environment. It describes in detail the challenges of tape storage as a serial medium, the limitations created by common organization formats such as TAR, and tape performance itself.

Existing technologies define linear tape as a single serial stream of stored information and data. This serial stream winds across thousands of meters of tape in a serpentine fashion and is not designed to support efficient random access. For applications that are not sequential in nature, this fundamental characteristic presents a significant challenge, and can result in long access times and inefficient use of space.

Partitions are a tape technology that have been around for many years, and have been used to segment tape media into a few areas that act like independent tape volumes. Typically, one of these segments is dedicated to metadata or system information and the other segment is used for user data. Unfortunately, efficient partition sizing requires that applications have some knowledge of how much storage space is required before it is allocated. In today's dynamic storage environments, with virtualization, data transformations and de-duplication, this becomes an impossible task.

Existing tape media can store more than a TB (terabyte) of user data. In the near future, a single tape media will hold tens of terabytes, with 100 TB capacities in the foreseeable future. Managing these multi-terabyte tape cartridges requires a new approach for managing data on tape. To address this need, Oracle has developed a new tape storage format with the StorageTek T10000C tape drive. This new format utilizes an innovative partitioning architecture, which allows the addition or removal of storage space as needed; a capability that will make the management of high-capacity tape much easier for future applications,

accommodate new storage paradigms for the greenest storage technology available, and help solve the problems caused by the digital data explosion.

## The Tape Storage Space Problem

Current technology manages tape media as a single stream of serial storage. This serial stream of data has a single beginning of data location and single end of data (EOD) location. Subsequent write operations define a new single EOD, which move down the length of the tape as more data is stored to the tape cartridge. When tape files are updated or modified, the original file is left on tape and the new modified file is added to the end of the serial stream. This process is shown in Figure 1, below. If “File B” must be modified, tape format conventions add the modified File B (designated File B’) to the end of the serial stream.

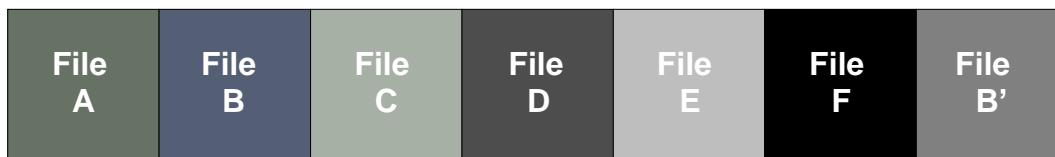


Figure 1: Modified File B written to tape after F

While this approach works fine for sequentially written data that may have a common expiration date for the entire cartridge, it is not optimal for random access data or cases when multiple files are written to tape with different expiration dates. For example, this method results in wasted space since the original File B is left in place. To periodically reclaim this wasted space, the files between B and EOD must be rewritten with the updated File B’ replacing the original File B. To avoid data loss, files that will be over written are read from tape (Files C-F) and stored elsewhere. Then every file from the original “File B” location to the end of the serial data stream (EOD) is rewritten with the old “File B” replaced as shown in Figure 2 below. Note: Files C-F designated with a “\*” do not change but they are rewritten.

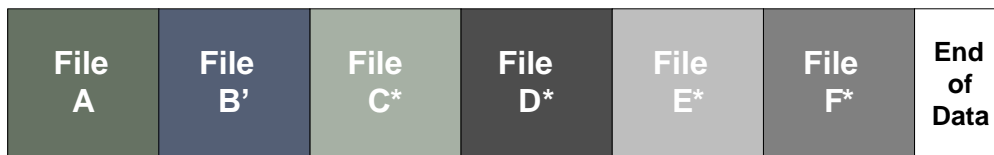


Figure 2: To reclaim space on tape, all files from B to F are rewritten

One can easily see that with tape cartridges exceeding 1 TB in capacity, this reclaim operation can consume a large amount of storage and compute resources. Typically, this process requires several hours per tape. To use tape for archive data, a better architecture is required.

## The Tape Organization Problem

The common formats used to organize files stored on tape are based on Tape Archive Record (TAR) formats. This format or modern derivatives of it are the standard for storing tape data. A typical TAR based format contains a header that describes the set of files being archived and a trailer (Tape Marks) indicating the end of the set of files. In multi-record TARs, often referred to as TAR balls, each file in the TAR set has a header, which describes that file set. This archive of a set of files ends with two trailers (Tape Marks). An example is shown in Figure 3.

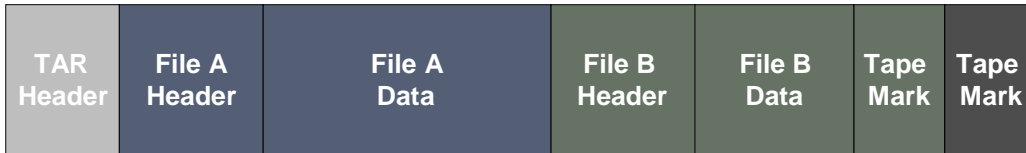


Figure 3: A TAR Dataset (TAR ball) containing file sets A and B

Since tape is managed as a single serial stream, each TAR is written as a standalone file system with the metadata needed to understand the files contained in each TAR. While associated TAR sets may originally be physically located together, they become separated as updates take place and the new modified data sets are added to the end of the serial data stream.



Figure 4: TAR A3 which is associated with TARs A1 & A2 is written at EOD

In addition to the space issue, the applications must read all the TARs A1-A3, extract the TAR metadata and then use that information to associate files and file updates. Since the metadata can be spread across the tape sequentially with the data, it can mean having to move from the beginning of tape sequentially to near the end of the tape to collect the metadata. Most tape vendors add additional information to optimize and streamline this process, but the basics of how tape is managed remains the same.

While there have been advances in tape format, they have dealt mainly with file size, use of compression and other application specific enhancements to the standard TAR. Recently, LTO-5 tape drives embedded a tape file system on their tapes. This is the next logical step in modernizing tape storage. While this advancement is logical, it does not fundamentally change how data is stored on tape. These new tape file systems organize file indexes and metadata in one partition and user data files in a second partition. Instead of locating to the EOD to determine the location of needed files, these systems typically locate to the end of the file metadata partition. An example of this organization is shown below.

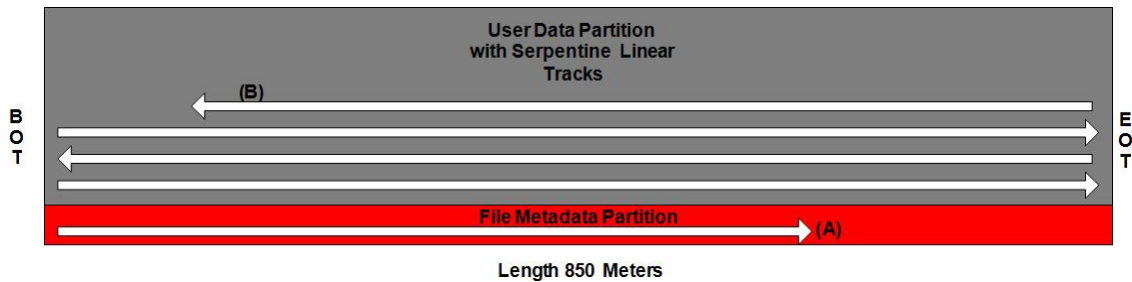


Figure 5: Modern tape with a file system

In Figure 5, the beginning of tape (BOT) is on the left hand side and the end of tape (EOT) is on the right hand side. The length of the tape is 850 meters. Point (A) is the EOD for the file partition and point (B) is the file that will be read. Movement from the file information metadata at point (A) to user data file at point (B) still requires long movements across tape with the added operation of changing partitions. The main benefit of this tape file system is to make tape look like disk to external applications.

## The Tape Access Problem

The final problem addressed in this paper is tape performance as a direct consequence of tape organization. Since file updates and additions are always added at the end of the serial data stream, the distance between the original files and their updates can impact data access performance significantly. Some incremental improvement in access performance is seen with a tape file system because the file metadata is stored in a relatively small partition, but it is not a paradigm shift in tape organization. Moving from the original file set or metadata to updated or related file sets can take minutes. Figures 1, 4 and 5 are typical examples of how existing tape organization contributes to this access problem.

## Why Previous Options Don't Work As Well

Present-day solutions to the problems outlined above are use of more expensive, less green disk storage to buffer tape files as they are unbundled, organized and accessed for use by disk centric applications. The serial nature of tape storage and data retrieval relegates tape to backup and disaster recovery. Previous tape file systems only partially solved this problem and did not fundamentally change tape organization and access. As a result, many organizations have not been able to fully realize the total cost of ownership advantage inherent in tape.

## The StorageTek In-Drive Reclaim Accelerator Solution

To address these problems, and to help optimize tape for archive applications, Oracle has developed an innovative new method for using tape partitions. We believe adopting this new StorageTek In-Drive Reclaim Accelerator format will revolutionize tape storage for the 21st century. This solution allows applications to break up tape's single serial data stream into smaller more manageable divisions of storage. The format is implemented using physical partitions that are linked together and managed as a

doubly linked list. This simple concept supports removing or adding physical partitions as storage requirements evolve over time.

In the following illustrations “File” refers to TARs. Figure 6 illustrates how several physical partitions are linked together to store Files A-I. File A is the beginning of this logical volume and is located in physical partition 1. The end of the logical volume is located in physical partition 3. The storage space used for any one logical volume is obtained by linking together any number of physical partitions.

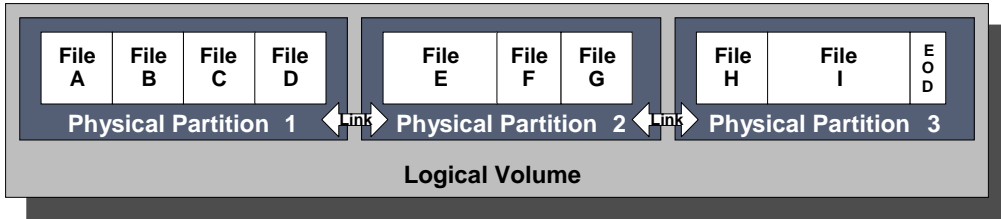


Figure 6: Three partitions linked together

Since this is tape, file access within each logical volume must be handled as a single serial stream. However, each physical partition is accessed randomly and dynamically linked to the logical volume during write. If Files E-G are obsolete and we wish to free that space; that can be accomplished by the application changing the partition map it maintains, as shown below.

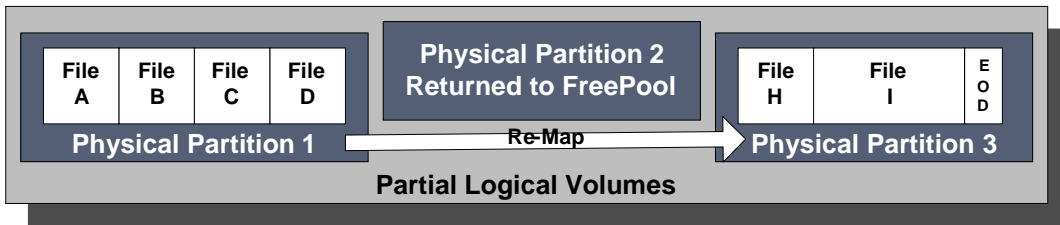


Figure 7: Physical Partition 2 is removed from the logical volume

If we are increasing the size of the logical volume that is accomplished by adding a link from physical partition 3 to physical partition 4, as shown in Figure 8.

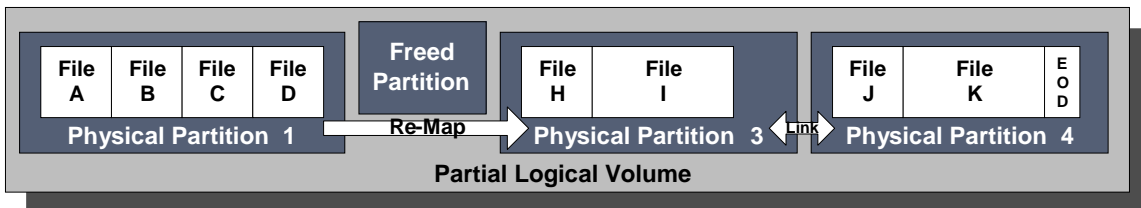


Figure 8: Physical Partition 4 is added to the logical volume



Linking together relatively small physical partitions and creating logical volumes changes the way tape is managed. To grow any logical volume, the application continues writing past the end of physical partition 3. This is shown in the transition between Figure 7 and Figure 8. The next available physical partition is automatically linked by the tape drive into the logical volume space.

Note, the actual link on tape does not change once a physical partition has been linked to a logical volume. This information is embedded in the data format of each physical partition and does not change until that partition is rewritten. For this reason partition maps need to be retained by the application so that it can skip over partitions that have been rewritten.

## Tape Reclamation with the StorageTek In-Drive Reclaim Accelerator

With Oracle’s new StorageTek In-Drive Reclaim Accelerator feature, there is no need to leave old files on tape. Reclaiming space is as simple as re-linking a single physical partition, as shown in Figure 7. The example is interesting, but usually obsolete files don’t fit exactly within a physical partition boundary. The more likely scenario is addressed in Figures 9 through 12, which illustrate modifying only File F and then reclaiming a partition with a new File J.

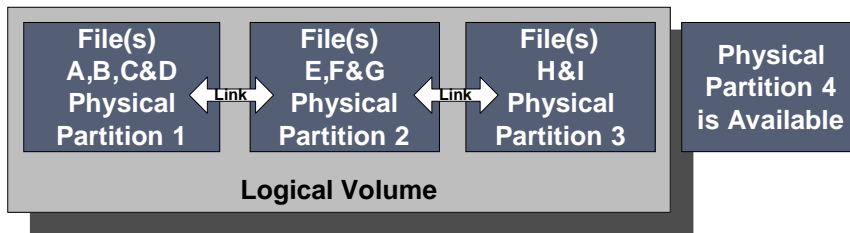


Figure 9: Files A-I are in one Logical Volume, the same as Figure 3

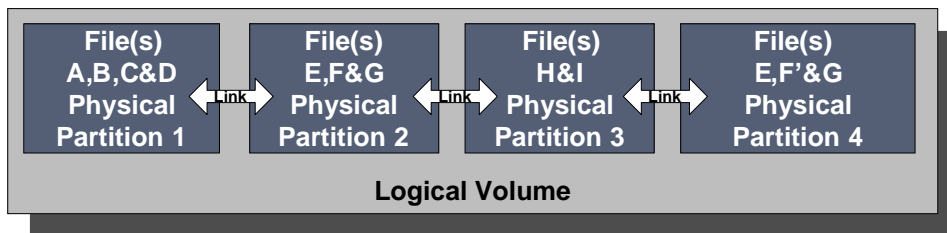


Figure 10: Files E&G are copied with a modified File F' to the end of the Logical Volume

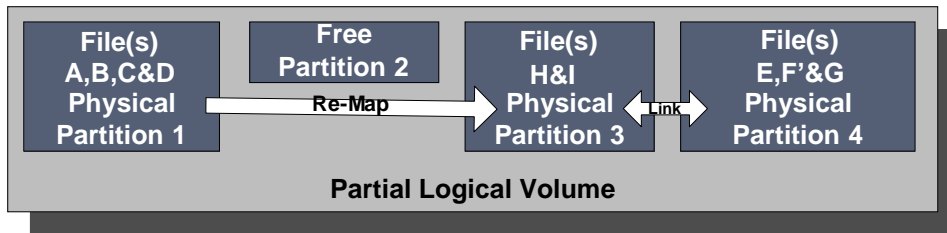


Figure 11: Physical partition 2 is removed from the Logical Volume

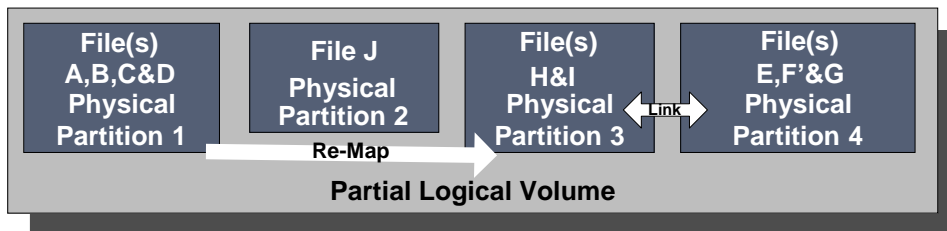


Figure 12: Physical partition 2 is reclaimed with File J

While not as simple as disk, automatically linking during write, and remapping these links manually, allows applications to manage tape using randomly accessible physical partitions. File modification and storage space recovery are only constrained by the time and resources needed to copy and re-map a single physical partition. The reclaim operation may be performed incrementally, and limits the impact on valuable business resources.

## The StorageTek Tape Tiering Accelerator Solution for Tape Organization and Access

Modern linear tape sizes range from 850 to 1100 meters in length. This often results in metadata information located hundreds of meters from associated user files. With the StorageTek Tape Tiering Accelerator, users control the physical location of associated metadata and data on tape and can overcome the single serial stream nature of tape formats.

An Oracle partitioned tape is organized in physical partitions and sections, as shown in Figure 13. This representation shows 80 physical partitions beginning on the lower left hand side of the tape that serpentine through the tape ending at the BOT side on the upper left hand corner of the tape. There is a guard band between each section so track turns are possible for movement between physical partitions in the same section. In addition, the drive can move between any section and any physical partition. This is a representation for illustration only. The characteristics of an actual partitioned tape will be discussed later in this paper.

	Section 0	Section 1	Section 2	Section 3	Section 4	
B O T	Partition 79	Partition 78	Partition 77	Partition 76	Partition 75	E O T
	Partition 70	Partition 71	Partition 72	Partition 73	Partition 74	
	Partition 69	Partition 68	Partition 67	Partition 66	Partition 65	
	Partition 60	Partition 61	Partition 62	Partition 63	Partition 64	
	Partition 59	Partition 58	Partition 57	Partition 56	Partition 55	
	Partition 50	Partition 51	Partition 52	Partition 53	Partition 54	
	Partition 49	Partition 48	Partition 47	Partition 46	Partition 45	
	Partition 40	Partition 41	Partition 42	Partition 43	Partition 44	
	Partition 39	Partition 38	Partition 37	Partition 36	Partition 35	
	Partition 30	Partition 31	Partition 32	Partition 33	Partition 34	
	Partition 29	Partition 28	Partition 27	Partition 26	Partition 25	
	Partition 20	Partition 21	Partition 22	Partition 23	Partition 24	
	Partition 19	Partition 18	Partition 17	Partition 16	Partition 15	
	Partition 10	Partition 11	Partition 12	Partition 13	Partition 14	
	Partition 9	Partition 8	Partition 7	Partition 6	Partition 5	
Partition 0	Partition 1	Partition 2	Partition 3	Partition 4		

Figure 13: Representation of an Oracle partitioned tape.

To format a partitioned tape for maximum sustained throughput, a logical volume is constructed horizontally down tape from BOT to EOT. This type of logical volume would contain physical partitions 0 through 4 and is highlighted in pink in Figure 14.

	Section 0	Section 1	Section 2	Section 3	Section 4	
B O T	Partition 79	Partition 78	Partition 77	Partition 76	Partition 75	E O T
	Partition 70	Partition 71	Partition 72	Partition 73	Partition 74	
	Partition 69	Partition 68	Partition 67	Partition 66	Partition 65	
	Partition 60	Partition 61	Partition 62	Partition 63	Partition 64	
	Partition 59	Partition 58	Partition 57	Partition 56	Partition 55	
	Partition 50	Partition 51	Partition 52	Partition 53	Partition 54	
	Partition 49	Partition 48	Partition 47	Partition 46	Partition 45	
	Partition 40	Partition 41	Partition 42	Partition 43	Partition 44	
	Partition 39	Partition 38	Partition 37	Partition 36	Partition 35	
	Partition 30	Partition 31	Partition 32	Partition 33	Partition 34	
	Partition 29	Partition 28	Partition 27	Partition 26	Partition 25	
	Partition 20	Partition 21	Partition 22	Partition 23	Partition 24	
	Partition 19	Partition 18	Partition 17	Partition 16	Partition 15	
	Partition 10	Partition 11	Partition 12	Partition 13	Partition 14	
	Partition 9	Partition 8	Partition 7	Partition 6	Partition 5	
Partition 0	Partition 1	Partition 2	Partition 3	Partition 4		

Figure 14: Physical partitions 0-4 form one logical volume.

This type of partition is constructed when the user desires to copy an older tape (e.g. LTO-2) to a higher capacity modern tape and maintain its access characteristics for legacy applications. Within the pink logical volume the partitioned format has the same characteristics as a single tape volume. The first block in physical partition 0 is the logical BOT and the last block in physical partition 4 is logical end of data EOD.

One quickly sees other possibilities with partitioned tape. Assume a medical application must store all files generated during the life of a patient and group them for quick access. Figure 15 shows a format supporting this association.

	Section0	Section 1	Section2	Section3	Section4	
B O T	Partition 79	Partition 78	Partition 77	Partition 76	Partition 75	E O T
	Partition 70	Partition 71	Partition 72	Partition 73	Partition 74	
	Partition 69	Partition 68	Partition 67	Partition 66	Partition 65	
	Partition 60	Partition 61	Partition 62	Partition 63	Partition 64	
	Partition 59	Partition 58	Partition 57	Partition 56	Partition 55	
	Partition 50	Partition 51	Partition 52	Partition 53	Partition 54	
	Partition 49	Partition 48	Partition 47	Partition 46	Partition 45	
	Partition 40	Partition 41	Partition 42	Partition 43	Partition 44	
	Partition 39	Partition 38	Partition 37	Partition 36	Partition 35	
	Partition 30	Partition 31	Partition 32	Partition 33	Partition 34	
	Partition 29	Partition 28	Partition 27	Partition 26	Partition 25	
	Partition 20	Partition 21	Partition 22	Partition 23	Partition 24	
	Partition 19	Partition 18	Partition 17	Partition 16	Partition 15	
	Partition 10	Partition 11	Partition 12	Partition 13	Partition 14	
	Partition 9	Partition 8	Partition 7	Partition 6	Partition 5	
Partition 0	Partition 1	Partition 2	Partition 3	Partition 4		

Figure 15: Physical partitions 0, 9, 10, 19, 20, 29, 30, 39, 40, 49, 50, 59, 60, 69, 70 & 79 form one logical volume

All the patient’s files are located in the first section of tape (Section 0) so any file can be accessed in seconds. The mechanisms for allocating physical partitions will be discussed later.

In the next example, a bank needs to save check images for individual bank accounts, associated by month. Figure 15 shows how partitioning can support this data organization on tape.

	Section0	Section 1	Section2	Section3	Section4	
B O T	Partition 79	Partition 78	Partition 77	Partition 76	Partition 75	E O T
	Partition 70	Partition 71	Partition 72	Partition 73	Partition 74	
	Partition 69	Partition 68	Partition 67	Partition 66	Partition 65	
	Partition 60	Partition 61	Partition 62	Partition 63	Partition 64	
	Partition 59	Partition 58	Partition 57	Partition 56	Partition 55	
	Partition 50	Partition 51	Partition 52	Partition 53	Partition 54	
	Partition 49	Partition 48	Partition 47	Partition 46	Partition 45	
	Partition 40	Partition 41	Partition 42	Partition 43	Partition 44	
	Partition 39	Partition 38	Partition 37	Partition 36	Partition 35	
	Partition 30	Partition 31	Partition 32	Partition 33	Partition 34	
	Partition 29	Partition 28	Partition 27	Partition 26	Partition 25	
	Partition 20	Partition 21	Partition 22	Partition 23	Partition 24	
	Partition 19	Partition 18	Partition 17	Partition 16	Partition 15	
	Partition 10	Partition 11	Partition 12	Partition 13	Partition 14	
	Partition 9	Partition 8	Partition 7	Partition 6	Partition 5	
Partition 0	Partition 1	Partition 2	Partition 3	Partition 4		

Figure 16: Three separate logical volumes

The check images for January are in the pink logical volume; February check images are in the dark grey logical volume; and March images are in the maroon logical volume. February has the most check images making it the largest logical volume, and January has the least check images making it the smallest logical volume. Since the physical partitions are only linked when written, exactly the amount of tape storage needed for the monthly check images is used. It takes a little longer to locate to the March (maroon) check images, since they are more than a third of the way down tape. However, once the drive reaches physical partition 2 accessing any of the March check images only takes seconds.

One can easily imagine many different file associations. Since Oracle’s partitioning is dynamic, it can be optimized for almost any association that makes tape a more useful storage option. The checkerboard of logical volumes shown in the following figure illustrates many different data associations.

	Section0	Section 1	Section2	Section3	Section4
B O T	Partition 79	Partition 78	Partition 77	Partition 76	Partition 75
	Partition 70	Partition 71	Partition 72	Partition 73	Partition 74
	Partition 69	Partition 68	Partition 67	Partition 66	Partition 65
	Partition 60	Partition 61	Partition 62	Partition 63	Partition 64
	Partition 59	Partition 58	Partition 57	Partition 56	Partition 55
	Partition 50	Partition 51	Partition 52	Partition 53	Partition 54
	Partition 49	Partition 48	Partition 47	Partition 46	Partition 45
	Partition 40	Partition 41	Partition 42	Partition 43	Partition 44
	Partition 39	Partition 38	Partition 37	Partition 36	Partition 35
	Partition 30	Partition 31	Partition 32	Partition 33	Partition 34
	Partition 29	Partition 28	Partition 27	Partition 26	Partition 25
	Partition 20	Partition 21	Partition 22	Partition 23	Partition 24
	Partition 19	Partition 18	Partition 17	Partition 16	Partition 15
	Partition 10	Partition 11	Partition 12	Partition 13	Partition 14
Partition 9	Partition 8	Partition 7	Partition 6	Partition 5	
Partition 0	Partition 1	Partition 2	Partition 3	Partition 4	

Figure 17: Many different data associations on a tape.

Figure 17 shows both access (vertical) and streaming (horizontal) data associations on tape. The single Partition 9 is also a logical volume, which contains all the metadata needed to describe this tape. It is ideally located at the beginning of tape for fast access. This illustration also shows logical volumes with physical partitions in different sections of tape (fragmented). The light pink logical volume uses physical partitions in sections 0, 1 and 4. Since logical volumes can be fragmented, similar to disk fragmentation, storage space must periodically to be cleaned up to optimize performance. Fortunately, this defragmentation can be done incrementally and minimize the impact on IT resources.

## StorageTek In-Drive Reclaim Accelerator and StorageTek Tape Tiering Accelerator Implementation

Oracle developed this partitioning format for ease of use and flexibility. A set of vendor unique commands support both SCSI Fibre Channel and the FICON interface on the StorageTek Virtual Storage Manager (VSM) system. Also, an API has been developed for Open Systems applications that prefer a C Library interface.

The magic of Oracle’s partitioning format is that logical volumes do not need to be completely allocated up front. A partitioned tape drive links a physical partition to the logical volume only when it is used. Any unused physical partition can be re-linked at any time to any logical volume.

Implementation of Oracle’s partitioning format is based on maps defining which physical partitions are used to build a logical volume. To build a logical volume, physical partitions are structured as a doubly linked list. These maps tell the drive which physical partitions to automatically link as a logical volume grows during write operations. For example, to construct the pink logical volume in Figure 16 the application provides a partition map reserving physical partitions 0, 9, 10, 19, 20, 29, 30, 39, 40, 49, 50, 59, 60, 69, 70 and 79. Note, this original partition map includes all the partitions in section 0. As the drive writes data to the pink logical volume it automatically links together physical partitions in ascending order starting with the lowest numbered physical partition in the map. The logical volume starts in physical partition 0. When physical partition 0 fills the drive creates a link from physical





To illustrate how this organization improves access with partitioning, the following table shows how access characteristics are improved when performing five random accesses on a non-partitioned tape and a partitioned tape. The following formulas are used in these calculations:

- L – Tape (~1095m) or Section length (~210m)
- S – Locate Speed (11 m/s, average over the tape)
- R – Time to ramp to S (3 seconds average over tape)

Average Access Time =  $((L \div 3) \div S) + R$

Maximum Access Time =  $(L \div S) + R$

Section Average Access Time =  $((210m \div 3) \div 11m/s) + 3s = 9.4$  seconds

Section Maximum Access Time =  $(210m \div 11 m/s) + 3s = 22.1$  seconds

Full Tape Average Access Time =  $((1095m \div 3) \div 11m/s) + 3s = 36.2$  seconds

Full Tape Maximum Access Time =  $(1095m \div 11m/s) + 3s = 102.5$  seconds

NOTE: These calculations do not include the time from tape load to ready status.

Partitioned tape average access times difference with Non-partitioned tape	Section	Section	Section	Section	Section
	0	1	2	3	4
1st random record after move to section	-26.8	-3.8	16.2	36.2	56.2
2nd random record within section	-53.6	-30.6	-10.6	9.4	29.4
3rd random record within section	-80.5	-57.5	-37.5	-17.5	2.5
4th random record within section	-107.3	-84.3	-64.3	-44.3	-24.3
5th random record within section	-134.1	-111.1	-91.1	-71.1	-51.1

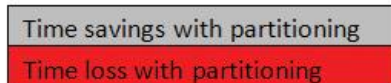


Figure 19: Average access time improvement with partitioning

Since the tape heads must locate to a section before the first access within that section, the first access in section 2 and higher takes longer than the average access time on a non-partitioned tape. As one would expect, the access times for the first two sections on a partitioned tape are much faster than on a non-partitioned tape. The surprising result is that even in the last section (Section 4) of a partitioned tape, access times improve after the third random file access.

## Conclusion

By breaking up tape’s single serial data stream into smaller, more manageable divisions of storage, the StorageTek In-drive Reclaim Accelerator simplifies the reclamation process and eliminates the need to leave obsolete files on tape. The StorageTek Tape Tiering Accelerator provides a simple and flexible means of managing files for fast access through the use of an innovative partition format. Together, the StorageTek In-drive Reclaim Accelerator and the StorageTek Tape Tiering Accelerator enable users

to overcome the serial nature of data management on tape, making it possible to fully realize its inherent total cost of ownership advantage.





Redefining Tape Usage With  
StorageTek Tape Tiering Accelerator and  
StorageTek In-Drive Reclaim Accelerator  
May 2011

Oracle Corporation  
World Headquarters  
500 Oracle Parkway  
Redwood Shores, CA 94065  
U.S.A.

Worldwide Inquiries:  
Phone: +1.650.506.7000  
Fax: +1.650.506.7200  
oracle.com



Oracle is committed to developing practices and products that help protect the environment

Copyright © 2011, Oracle and/or its affiliates. All rights reserved. This document is provided for information purposes only and the contents hereof are subject to change without notice. This document is not warranted to be error-free, nor subject to any other warranties or conditions, whether expressed orally or implied in law, including implied warranties and conditions of merchantability or fitness for a particular purpose. We specifically disclaim any liability with respect to this document and no contractual obligations are formed either directly or indirectly by this document. This document may not be reproduced or transmitted in any form or by any means, electronic or mechanical, for any purpose, without our prior written permission.

Oracle and Java are registered trademarks of Oracle and/or its affiliates. Other names may be trademarks of their respective owners.

AMD, Opteron, the AMD logo, and the AMD Opteron logo are trademarks or registered trademarks of Advanced Micro Devices. Intel and Intel Xeon are trademarks or registered trademarks of Intel Corporation. All SPARC trademarks are used under license and are trademarks or registered trademarks of SPARC International, Inc. UNIX is a registered trademark licensed through X/Open Company, Ltd. 0511

**Hardware and Software, Engineered to Work Together**