Tape Tiering Accelerator (TTA)

Z/OS User Specification

Part Number E25377-01

June 3, 2011

Tape Tiering Accelerator (TTA) Z/OS User Specification

Part Number E25377-01

Oracle welcomes your comments and suggestions for improving this book. Contact us at STP_FEEDBACK_US@ORACLE.COM. Please include the title, part number, issue date, and revision.

Contents

1. Overview 4
2. TTA Programming Methods for Z/OS 4
2.1 Set Host Type 5
2.2 Detect Current TTA State 5
2.3 Map TTA Partitions 6
2.4 Locate TTA Partition 6
2.5 Report TTA Linkage 7
2.6 Create TTA Tape 7
2.7 Start New Logical Volume 8
2.8 Convert a TTA Tape to Standard Tape 8
3. TTA Examples 9
3.1 TTA Example to Save Space 9
3.2 TTA Example to Improve Access 11
4. Drive Operation 13
4.1 Logical Volume 13
4.2 Partial Logical Volume 13
4.3 Writing 13
4.4 Reading 14
4.5 Spacing and Locating 14
5. TTA Layout 16
6. TTA Capacity 16
7. Potential TTA Benefits and Effects 17
7.1 Increased Access Times 17
7.2 More Efficient Use of Tape Capacity 17
8. Theory of Operation 17
8.1 Initializing an TTA Tape 17
8.2 Loading an TTA Tape 17
8.3 Positioning to a Host File 18
8.4 Reading a Host File 18
8.5 Appending a Host File 18
8.6 Writing a Host File to a free TTA 18
9 TTA Data Definitions 19

1. Overview

Today, tapes hold hundreds to thousands of gigabytes of data. Typically, the data is "stacked" on the media as different data sets and each data set has different expirations. When the expiration occurs, space is wasted on the tape. Over time the wasted space becomes large enough where customers must reclaim the tape, which can consume many hours. In order to save our customers time and money, a method to capture the wasted space was created called Tape Tiering Accelerator (TTA). Using this method, the tape drives were created in a hard disk like format such that more of the tape can be used. Tape tiering or partitioning (Non Linked) has existed for many years. However many of today's tape drives do not implement them because developers didn't think they were useful. This implementation goes beyond tiers and creates automatically linked partitions where the data can span across tiers or partitions and be non-contiguous on tape. Furthermore the tape drive handles much of this formatting without host intervention.

The T10000C drive TTA Specifications:

- Cartridge Requirement: Long Fuji
- Number Partitions: 480
- Partition Size: 9gb

This User Guide describes how to format a TTA Cartridge and access it but excludes details about how to manage the partitions or the data in them. The T10000C tape drive stores the Logical Volume start and linkage status in the MIR but does not store Read/Write access status of partitions.

2. TTA Programming Methods for Z/OS

The T10000C tape drive introduces a new set of command codes and extensions to other command formats on Z/OS. To implement these new command codes, you need to use EXCPVR; EXCPVR is documented in the following IBM publications:

- SC26-7400 DFSMSdfp Advanced Services
- SC26-7408 Z/OS Macro Instructions for Data Sets
- SA22-7605 MVS Programming: Assembler Services Guide

Figure 1 illustrates an example of TTA partition allocation. In Figure 1 the TTA tape contains 3 files, A, B and C. File A occupies a single partition (0). File B occupies three partitions that are linked consecutively (1, 2, 3). File C occupies 8 non-consecutively linked partitions (4, 13, 14, 15, 16, 17, 18, 19) indicating that partitions (5, 6, 7, 8, 9, 10, 11, 12) were not writable at the time file C was written to the tape.

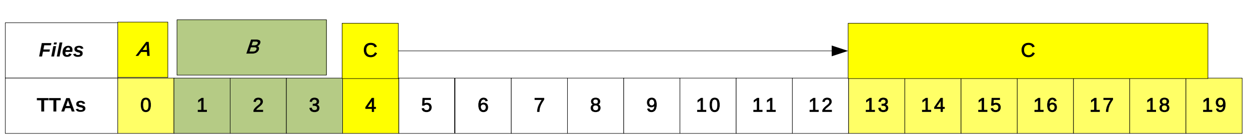

Figure 1: Example TTA Partition allocation

The command procedures and their command formats are summarized below:

2.1 Set Host Type

Command X'77', 20 byte parameter data

Use the PSF Set Host to identify the drive as a STK/Oracle drive. The following parameter data byte must be set:

- Byte 0 = X'40'
- Byte 2 = X'08'

The Set Host Type command must be issued before any other TTA commands. The other TTA commands will be recognized only after this command is issued.

2.2 Detect Current TTA State

Command X'77', suborder X'18', 12 byte parameter data (prep for RSSD)

Command x'3E', 40 byte buffer to receive RSSD response, chained to prep for RSSD

The RSSD command is used to return the TTA State of the T10000C drive and loaded tape. This command indicates if the drive is TTA capable, TTA enabled, and TTA Tape loaded and returns the current partition number at which the tape is positioned. To issue this command, the correct host type command must have been issued and the RSSD CCW must be chained to a PSF command with the Prep RSSD parameter data specified. The PSF command must be the first CCW in the chain.

Prep for RSSD parameter data supplied:

X'180000000000030000000000'

The RSSD command returns the following TTA related status fields:

- Byte 14 – Cartridge status, X'02' indicates and TTA formatted cartridge
- Byte 15 – Drive status, X'02' indicates the drive is in TTA mode
- Byte 16-17 – 2 byte current partition position
- Byte 24 – Drive capability, X'02' indicates the drive is TTA capable

2.3 Map TTA Partitions

Command X'77', suborder X66', 132 byte parameter data to enable partitions for write

The command Perform Subsystem Function (PSF) suborder x'66' is used to define the partition map which will enable individual partitions for Read-Only or Write authority. The map size is 128 bytes with a 4 byte header for a total parameter length of 132 bytes. The parameter data format is as follows:

- Bytes 0-3 – X'66008000'
- Bytes 4-126 – Partition authorization bits

The T10000C drive supports 480 partitions represented as 0-479. Each bit position starting with the high order bit of byte 4 through the low order bit of byte 64 represents the corresponding partition number's state. A B'1' represents write authorization and a B'0' represents Read-Only authorization. The remaining bits are reserved for future use, setting those bits to B'1' will result in a Command Reject status, parameter exception. For example:

X'6600800080' – Partition 1 is enabled for write; the remaining bytes are set to zero.
X'66008000C0' – Partitions 1 and 2 are enabled for write; the remaining bytes are set to zero.
X'66008000FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF' – Every partition is enabled for write; the remaining bytes must be set to zero; otherwise Command Reject status is returned.
X'66008000' – Partitions are enabled for read only, all partition bytes are set to zero. This is the default state when a TTA Tape is loaded.

When issuing the TTA Partition Map command the tape must be positioned at Beginning of Tape (BOT). Otherwise Command Reject status is returned.

2.4 Locate TTA Partition

Command X'BF', 2 byte parameter data

The Locate TTA command moves the tape into position on the target tape drive so that the application can write or read on a specific partition on tape. The tape drive positions the tape to the nearside of the first block of the partition.

This command requires a data length of 2 with the parameter data containing a 2 byte binary partition number (0-479). A TTA formatted tape must be loaded and ready; if any of these conditions are not true a Command Reject (ERPA_code 27) will result. Initial status of Channel End (CE) and Device End (DE) will be displayed upon the command's completion.

The 2 byte parameter data will be boundary checked and a Command Reject (CR) will be issued if the value is outside the boundary supported by the loaded tape.

Once the tape is located to a partition then normal commands are used to process the blocks within the partition, such as write, read, tape mark, sync, read RBID, etc.,. Use the RSSD command to return the current partition. Note that the current partition is relative to blocks actually written on the physical tape. A SYNC command will flush the cache to tape in order to obtain an accurate TTA partition number.

2.5 Report TTA Linkage

Command X'B2', return 2048 bytes of information

The Report TTA Linkage command returns the current TTA mode tape linkage (mapping). The command returns 2048 bytes of information. The information will represent how the partitions are forward linked together. The information will represent each of the partitions (2 bytes of data for each partition). The 2 bytes of data are the next partition number in the link or one of 4 special values. The 4 special values are:

X'FFFF' - partition not linked or no more TTA Partitions linked to this partition.
X'FFFE' - partition not used or the TTA Partition isn't used in the current tape format.
X'FFFD' - partition link unknown or the drive doesn't know if the TTA Partition is linked to something else. This only happens when a tape is loaded and the drive is power cycled or receives an unexpected reset condition.
X'FFFC' - partition blank or the TTA Partition hasn't been written since the last time the tape was converted to a TTA tape.

2.6 Create TTA Tape

This section describes the steps required to create a TTA Tape from a standard tape. Once the required steps are completed the tape will be formatted as a TTA Tape with a Logical Volume starting at TTA partition 0. The drive will recognize the tape as a TTA tape when it is loaded.

A TTA formatted tape cannot be processed as a standard tape once formatted as a TTA Tape, Z/OS will detect a permanent I/O error during open. Additionally, when creating a TTA tape from a standard tape you must bypass label processing by specifying BLP on the DD Label parameter or use other means to bypass label processing.

Once the drive is loaded, ready, and the tape is at Beginning of Tape (BOT), perform the following sequence of commands:

1. Set Host Type described in section 2.1 to allow the TTA commands to be recognized.
2. Detect Current TTA State described in section 2.2 to verify that the drive is TTA capable and a Standard (not TTA) is loaded and the TTA Mode bit should be zero.
3. Map TTA Partitions described in section 2.3; partition 0 must be set to write.
4. Write at least one block and non buffered sync or one non buffered tape mark. Note any number of blocks can be written such as labels followed by a non-buffered Tape Mark.
5. Optionally Detect Current TTA State may be performed and the TTA Mode bit will be set but the TTA Cartridge bit will not be set until the tape is reloaded.

At this point the tape is a TTA formatted tape and the Map TTA Partitions and Locate TTA Partition can be used to map and position the tape to the desired partition. Normal I/O command processing is available to process the tape. Note in some cases general utilities may or may not work once a TTA Formatted Tape is loaded and positioned to a partition greater than 0.

2.7 Start New Logical Volume

To start a new Logical Volume a TTA Formatted Tape must be loaded or created. Each partition can potentially be a standalone 9.68gb logical volume or multiple partitions contiguous or non-contiguously linked to form a logical volume.

1. Map TTA Partitions described in section 2.3; partition 0 must be set to write. This step may have been performed already. It only needs to be done if the map needs to be changed.
2. Locate TTA Partition described in section 2.4 to position the tape at the start of the new Logical Volume.
3. Write at least one block and non buffered sync or one non buffered tape mark. Note any number of blocks can be written such as labels followed by a non-buffered Tape Mark.

2.8 Convert a TTA Tape to Standard Tape

Command X'77',suborder X'67', 4 byte parameter data to convert to Standard Tape

This command sequence will convert a TTA formatted tape back to a Standard formatted tape. However this process is not recommended except in situations where a Standard Tape was accidently formatted as a TTA tape and never used. The process of positioning and processing TTA formatted tapes may cause ware and debris to form around partition start and could degrade the tape's performance as a standard format tape.

Perform Subsystem Function (PSF) suborder x'67' is used to remove the partitions and revert to Standard Format. The parameter data format is as follows:

1. PSF Command with parm bytes 0-3 equal to X'67000000' reverts to standard format.
2. Write at least one block and non buffered sync or one non buffered tape mark. Note any number of blocks can be written such as labels followed by a non-buffered Tape Mark.

3. TTA Examples

These examples demonstrate typical uses for TTA to save space and improve access time.

3.1 TTA Example to Save Space

To format a TTA tape, the application issues a "Set TTA Mode" Command. The tape should be empty or a scratch tape since initializing it may prevent access to data previously written on the tape. Formatting a TTA tape divides the tape into a fixed number of empty TTAs with a guard band allocated between each partition. Next the application sets a write mask by issuing a "Map TTA Partitions" Command; this command will tell the tape drive which TTAs to link together during subsequent write operations. In figure 2 the first 20 TTAs on tape have been mapped for writing. Each block is an empty TTA that can be used by the application.

Files	**Empty – No data has been written to the tape**																			
ALPs	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19

Figure 2: An empty TTA tape

The application now sends a "Locate TTA" command to TTA #1 and a "Start New Logical Volume" command sequence. The application can now begin writing to this logical volume on tape using standard tape write commands. This process may occur over multiple mount and dismount cycles and across several drives.

In figure 3 below the application writes nine files. Those files span all 20 TTAs. The data shown in the top row are application files and the bottom row indicates the physical relationship of the 20 TTAs to those files. Note that the files can span one or more TTAs or be contained within a single TTA.

Files	*A*			*B*		*C*		*D*	*E*		*F*		*G*		*H*				*I*	
ALPs	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19

Figure 3: Host files and TTA allocation

Over time the files that are written to tape expire and become obsolete or invalid. As the expired files become obsolete the tape begins to resemble "Swiss Cheese", with holes created throughout the tape. In figure 4 host files B and F expire and become obsolete which frees TTAs 2, 3, 4, 9 and 10 for space reclamation.

Files	A			Expired		C	D		E	Expired			G		H			I		
ALPs	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19

Figure 4: Host files becoming invalid

The application can now reuse the TTAs containing the expired files by starting a second logical volume and over writing them. First the application should execute a new “Set Writeable TTAs” Command sequence that maps partitions 2, 3, 4, 9 and 10 as writeable. Then the application positions to the first TTA that was freed when record B expired. In this case it is TTA 2. To locate to partition 3 use the “Locate TTA” command. The application can now write files M and N. As these files are written to tape the drive automatically positions to the next available TTA. This process may occur over multiple mount and dismount cycles and across several drives. Although read access will still be permitted to all TTAs, it will only be possible to write to partitions that forms the current writable TTAs (2, 3, 4, 9, and 10) shown as green blocks below. File N links from partition 4 to partition 9 and then continues into partition 11. Note the space between file A and M (end of partition 1); N and C (beginning of partition 5); E and N (end of partition 8) are wasted space.

This space is not empty as it contains the residual data from the expired Files B and F. If the application attempts to read the old file B data, the beginning of that file will be read and then drive will report an EOD error at the end of ALP 1. As a result, the application will need to stop reading at the end of file A. This process is discussed in detail below.

Files	A		M		N	C	D		E	N			G		H			I		
ALPs	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19

Figure 5: New Files written to freed partitions

The application will have to retain a record of the start of each record (Host Block ID and ALP number) to navigate through logical volumes where files have expired and partitions reallocated to a new logical volume. To read the first logical volume (yellow) in figure 5 do the following.

1. Locate to TTA #0.
2. Locate to the start block ID for record A.
3. Read record A.
4. Locate to TTA#5.
5. Locate to the start block ID for record C.
6. Read record C, D and E.
7. Locate to TTA#11.
8. Locate to the start block ID for record G.
9. Read record G, H and I.

The user should also note that it will be necessary for the application to track which “Set Writeable TTA” command map corresponds to each logical volume. There will be a different write map for the first (yellow) and second (green) logical volumes.

Now assume files D and H shown in figure 5 expire. Following the same process as described above the application can now start a third logical volume (purple). Again, start by sending a new "Set Writeable TTAs" Command sequence that maps partitions 6, 7, 14 and 15, a "Locate TTA" command to TTA #6. As shown in figure 6, the host can then write this third logical volume with files S and T.

Files	A		M		N	C	S	T	E	N		G			T		I			
ALPs	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19

Figure 6: New files written for third logical volume

To read the second logical volume (green) in figure 6

1. Locate to TTA#2.
2. Read files M and N.

To read the third logical volume (purple) in figure 6

1. Locate to TTA#6.
2. Read files S and T.

It is worth noting that the application only needs to track TTA# and Host Block ID when space for expired files is reclaimed and used by another logical volume. In logical volumes that have not had space reclaimed like the green and purple examples in figure 6, the TTA feature of the tape drive will move from TTA to TTA as if the logical volume was written sequentially in a single tape partition.

3.2 TTA Example to Improve Access

This example describes features that are only supported in the T10000C tape drive. The following representation of a TTA formatted tape that has 5 sections with 5 TTAs in each section, for a total of 25 ALPs. This format does not actually exist and it is for illustration purposes only. The actual T10000C ALP format has 480 TTAs and is too large to show in a small drawing. In this illustration, TTA 0 is located at the Beginning of Tape (BOT) at the bottom left side and the ALPs serpentine through the tape in a linear fashion from BOT to End of Tape (EOT) and back again.

TOB (left) / TOE (right)

Section 0	Section 1	Section 2	Section 3	Section 4
← ALP 39	← ALP 38	← ALP 37	← ALP 36	← ALP 35
ALP 30 →	ALP 31 →	ALP 32 →	ALP 33 →	ALP 34 →
← ALP 29	← ALP 28	← ALP 27	← ALP 26	← ALP 25
ALP 20 →	ALP 21 →	ALP 22 →	ALP 23 →	ALP 24 →
← ALP 19	← ALP 18	← ALP 17	← ALP 16	← ALP 15
ALP 10 →	ALP 11 →	ALP 12 →	ALP 13 →	ALP 14 →
← ALP 9	← ALP 8	← ALP 7	← ALP 6	← ALP 5
ALP 0 →	ALP 1 →	ALP 2 →	ALP 3 →	ALP 4 →

Figure 7: Representation of a TTA formatted tape.

Section 0 in Figure 7 contains TTAs 0, 9, 10, 19, 20, 29, 30 and 39. Since all of these TTAs are in a single section, the average access time to locate to any TTA in section 0 is relatively short. A T10000C tape drive has an average access time of about 10 seconds for any TTA in the same section.

If the application requires fast tape access, it can be achieved by constructing logical volumes within the same section. To build a logical volume in section 0, the following commands should be sent to the drive.

1. Load a T10000C ALP formatted tape.
2. At BOT Issue a "Set Writeable TTAs" command sequence that maps TTAs 0, 9, 10, 19, 20, 29, 30 & 39.
3. Issue a "Start New Logical Volume" Command sequence.
4. Begin writing this Logical Volume.

As files are written to this logical volume the drive will automatically link the mapped TTAs starting with the lowest numbered ALP. If application files are written to the first three TTAs in the logical volume to the TTAs shaded in yellow will contain the logical volume as shown below. A wrap turn will be performed at the end of TTA 0 and TTA 9 will be written in the opposite direction. At the end of TTA 9 another warp turn will be performed and direction reverses again.

TOB (left) / TOE (right)

Section 0	Section 1	Section 2	Section 3	Section 4
← ALP 39	← ALP 38	← ALP 37	← ALP 36	← ALP 35
ALP 30 →	ALP 31 →	ALP 32 →	ALP 33 →	ALP 34 →
← ALP 29	← ALP 28	← ALP 27	← ALP 26	← ALP 25
ALP 20 →	ALP 21 →	ALP 22 →	ALP 23 →	ALP 24 →
← ALP 19	← ALP 18	← ALP 17	← ALP 16	← ALP 15
ALP 10 →	ALP 11 →	ALP 12 →	ALP 13 →	ALP 14 →
← ALP 9	← ALP 8	← ALP 7	← ALP 6	← ALP 5
ALP 0 →	ALP 1 →	ALP 2 →	ALP 3 →	ALP 4 →

Figure 8: TTAs 0, 9, and 10 are written in section 0

If it is desired to set a write mask for an entire section, the mask table below can be referenced for the binary values. For any section, the mask values will be a repeating binary sequence indicated by the values show in the table below. For example: To mask the entirety of section 0 as writable, a writable mask containing 60 bytes with the hexadecimal values of 80, 60, 18, 06, 01, 80, 60, 18, 06, 01……should be written to the drive.

4. Drive Operation

The following figure illustrates the above TTA example. For the discussion of drive operations refer to figure 9. It may also be helpful to refer to the TTA example to see how the tape got into this condition.

Files	*A*		*M*		*N*	*C*	*S*	*T*	*E*	*N*			*G*		*T*		*I*			
ALPs	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19

Figure 9: Three different logical volumes with ALP

4.1 Logical Volume

A logical volume is defined as a group of TTAs linked together that have both a block 0 and a physical EOD. In the above figure, there are two complete logical volumes. The first logical volume maps TTA 2, 3, 4, 9 and 10 (files M and N) and the second logical volume maps TTAs 7, 8, 15, 16 (files S and T).

4.2 Partial Logical Volume

A partial logical volume is a group of TTAs linked together with either no block 0 or no physical End of Data (EOD), or neither a block 0 or an EOD. In figure 9, there are 5 partial logical volumes. TTA 0 and 1 are a partial logical volume with a block 0 but no physical EOD. TTAs 16, 17, 18, 19 make up a partial volume with no block 0 but it does contain an EOD. TTA 5, TTA 8 and TTAs 11, 12, 13 are partial volumes without a block 0 or a physical EOD. These partial logical volumes contain the files that were not expired for the originally written logical volume shown in figure 3. When combined they still represent that logical volume.

4.3 Writing

When writing, the drive only allows writes on TTAs identified with the Set Writable TTAs using the Map TTA Partition command sequence. On load the tape defaults to no TTAs being writable. The host should give the set writable TTAs by using the Map TTA Partitions command mask after load which must be done at BOT. If the Set Writable TTAs command isn't issued, the tape will only operate in a read only mode.

The drive always chooses the lowest TTA after the current TTA when linking TTAs. The drive does not allow the next TTA to wrap around the writable list. For example, TTA 255 will not be forward linked to TTA 20.

The drive will report LEOV on the last free TTA in a manner similar to a normal tape. The amount of space between LEOV and PEOV has not changed. When starting a write the drive uses the current block as the append point, just like in normal tape operation, with one exception, if the write command was proceeded by a start new logical volume, the drive writes at block 0 at the start of the current TTA. The host should be located at the start of the TTA before issuing the start new logical volume command. Regardless of where the host is logically positioned, the write will take place at the start of the TTA.

When the application is finished writing a file, the application should issue a "Detect Current TTA State" command sequence and TTA number where the last block was written will be in bytes 16 and 17.

4.4 Reading

When reading the tape, the drive automatically moves to TTAs linked together. For example, if the host is positioned to read TTA 11, the drive transitions from TTA 4 to 5, 10, and 11 with no host intervention.

It is assumed that the host does not read old host files as a matter of course. But if the host requests to read the start of host file B in TTA 2 (for example) the data is returned until the end of TTA 2 at which point the drive returns EOD.

4.5 Spacing and Locating

The drive operates on (partial) logical volumes when processing space block/file and locates commands. When spacing/locating prior to first block in a partial volume the drive reports back that BOT has been crashed. For example, if record G in TTA 11 starts at block 10,000, then any operation that ends up before 10,000 reports a Unit Check(UC) BOT.

Similarly when spacing/locating beyond the last block in a partial volume EOD is reported. For example, if record E in TTA 8 ends at block 1,500 then any operation that ends beyond 1,500 reports a UC EOD.

Let's use figure 3 and figure 5 as examples. In figure 3, File A starts in TTA 0 and for this example the starting block id for TTA 0 is zero. File B spans TTA 2 and for this example the starting block id in TTA 2 is 20,000. Now, suppose file B expires. In figure 5 the user decided to write file M starting in TTA 2. The result of putting file M into ALP 2 caused an EOD to be put down at the end file A.

Starting block	**0**	**10,000**	**20,000**
Files	**A**	**B**	**C**
TTAs	**0**	**1**	**2**

Figure 10: Three files written into one logical volume.

EOD

Starting block	**0**		**20,000**
Files	**A**		**C**
TTAs	**0**	**1**	**2**

Figure 11: File B expires but nothing has reclaimed that space yet.

EOD

Starting block	**0**	**0**	**20,000**
Files	**A**	**M**	**C**
TTAs	**0**	**1**	**2**

Figure 12: File M reclaims the space that file B used to occupy.

EOD **EOD** **EOD**

Normal spacing and Locating...In figure 10, if the user can locate to any blocks in the range of 0 thru 29,999. It is no problem because the TTAs are linked. If the user locates to 10,000 then the tape will end up in TTA 1.

UC into EOD example…Figure 10 shows Files A, B and C. TTAs 0, 1, 2 are all linked together into one logical volume. As you can see, the EOD is at the end of TTA 2. In figure 11, file B expires at the host. In figure 12, the host reclaims TTA 1. When that reclaim occurs the links are broken and each TTA has its own EOD now. If the user now tries to locate to block 10,000 then the tape drive will generate a crash into EOD because TTA 0 is no longer forward linked to TTA 1.

UC into BOT example.... Figure 10 shows Files A, B and C. TTAs 0, 1, 2 are all linked together into one logical volume. As you can see, the EOD is at the end of TTA 2. In figure 11, file B expires at the host. In figure 12, the host reclaims TTA 1. When that reclaim occurs the links are broken and each TTA has its own EOD now. If the user locates to TTA 2 with the Locate TTA command, then the tape drive will be positioned in TTA 2. If the user now tries to locate to block 10,000, then the tape drive will generate a UC into BOT because TTA 1 is no longer backward linked to TTA 2.

5. TTA Layout

A T10000C tape formatted for TTA is represented in the T10000C format specification. It has 5 sections of 96 TTAs for a total of 480 TTAs. The TTAs are organized in a linear serpentine pattern starting at ALP #1 at the Beginning of Tape (BOT) at the bottom left side increasing sequentially down the length of tape to End of Tape (EOT) on the right side and then returning to BOT. Refer to figure 8 to see this pattern.

6. TTA Capacity

For the T10000C each TTA is approximately 9GB. (Note: 480 * 9 GB is not equal to the full 5000GB that can fit on a T10000C tape) This loss of full capacity is needed to allow the TTAs to be written in any order.

7. Potential TTA Benefits and Effects

7.1 Increased Access Times

If an application identifies data for frequent access, that data can be located in TTAs at the beginning of the tape for fast access. The host can control where data is located on tape by setting writable TTAs.

Like data sets can be located together, so that access to multiple files of like data can be performed more efficiently.

7.2 More Efficient Use of Tape Capacity

Depending on the application much of the data on a particular tape is obsolete unnecessary data. This data can now be reclaimed without the need to rewrite valid data on the tape. With tape capacities ever increasing, the effective tape capacity will continue to get worse.

8. Theory of Operation

While there are many valid ways of using the set of commands for meeting an individual application's requirements, this section shows how we envisioned the commands will be used.

8.1 Initializing an TTA Tape

To initialize a TTA tape, the only thing required is to issue the “Create TTA Tape” command sequence on a tape that is in the free pool (Recall that the activate process is destructive). Best practices would also be to set the writable TTA mask for Byte 0, bit 7 to a logical 0x1 and write some sort of label or tape mark at TTA 0. TTA 0 must be initialized before volumes are created or accessed that utilize remaining TTAs.

8.2 Loading an TTA Tape

After loading a TTA tape, a Map TTA Partitions command should be issued so that the tape can be written. The writable mask is kept by the host application based on which TTAs no longer hold valid data.

8.3 Positioning to a Host File

The host is responsible for keeping track of where each host file is on tape in the form of TTA and block id (RFID). Positioning to a host file is as easy as issuing a locate TTA to the proper TTA and then issuing a standard tape locate command to get to the start of the host file.

TTA tape read position byte level references are relative to a start of volume position for a particular partition. If for example, a second or third partition is created at TTA 10 by issuing a start volume at that partition 10, then once a file is written beginning at TTA 10, any subsequent read position information (RFID) for that partition will be relative to the volume beginning at TTA 10. The same is true regardless of how many partitions are created. Issuing a start volume at a particular TTA does not immediately set the beginning RFID to RFID 0; a write to that volume must occur first.

8.4 Reading a Host File

After positioning to a host file, reading is as simple as issuing standard tape read commands; the drive will automatically transition to next TTAs as needed.

8.5 Appending a Host File

When appending a new host file to an existing one, the host will first position the drive to the end of the existing host file. Next the host should issue a Detect Current TTA State command sequence and note the value at bytes 16-17 in the host's file location structure. Next the host should issue a standard read position command to get the RFID, this should also be noted in the host's file location structure. These values are used later when the new host file is accessed or for determining when a TTA is free to be reused. Next the host writes the new file. Another Detect Current TTA State command sequence is issued to determine the current TTA. Based on the starting TTA and current TTA the host can determine which TTAs the file resides in. This information is kept in the host file location structure and later used to determine when a TTA is free to reuse.

8.6 Writing a Host File to a free TTA

When starting a new logical volume the first command to issue is a locate TTA to the first free TTA which is kept by the host file location structure. Next the host file location structure is updated with the first free TTA and a host block id of 0. Next the host initiates a start new logical volume command sequence and writes the new file. A Detect Current TTA State command sequence is issued to determine the current TTA. Based on the starting TTA and current TTA the host can determine which ALPs the file resides in. This information is kept in the host file location structure and later used to determine when a TTA is free to reuse.

9 TTA Data Definitions

RSSD data definition.

```
         MACRO
         RSSDMAP ,
.*****************************************************************
****
.*
*
.*  Copyright (c) 2011, Oracle and/or its affiliates.
*
.*  All rights reserved.
*
.*
*
.*  Dsect map for RSSD information returned for Generic Host
*
.*
*
.*  Revised 6-01-2011
*
.*
*
.*****************************************************************
****
*
*        RSSD Map
*
RSSDMAP  DSECT
RSSDLEN  DC    XL2'0'              Length of this RSSD Message
RSSDMID  DC    XL1'0'              Message ID
RSSDNMSG EQU   X'00'               .. No Message
RSSDIDST EQU   X'02'               .. ID Status
RSSDSTK  EQU   X'E0'               .. STK Message ID
RSSDMCD  DC    XL1'0'              Message Code
RSSDMCD1 EQU   X'01'               Dependent Format
RSSDMSGI DC    XL4'0'              Message ID
RSSDFLG1 DC    XL1'0'              Flag Byte
RSSDNNS  EQU   X'01'               .. Notify Non-supported
RSSDSOLM EQU   X'02'               .. Solicited Message
RSSDRSV2 DC    XL5'0'              Dependent on format and msg
code
RSSDCART DC    XL1'0'              Cartridge Status
RSSDTTAC EQU   X'02'               .. TTA Cartridge is loaded
```

```
RSSDVOLS EQU   X'80'                  .. Volsafe Cartridge is loaded
RSSDFET1 DC    XL1'0'                 Drive Enabled Features
RSSDRB1E EQU   X'01'                  .. Reserved
RSSDTTAE EQU   X'02'                  .. TTA Drive Mode is Enabled
RSSDRB4E EQU   X'04'                  .. Reserved
RSSDBFTM EQU   X'08'                  .. Buffered Tape Marks
RSSDSTRA EQU   X'10'                  .. Stop read Ahead
RSSD32BI EQU   X'20'                  .. 3490 32bit BlockID
Emulation
RSSD3590 EQU   X'40'                  .. 3590 Emulation
RSSDVLSE EQU   X'80'                  .. Volsafe Enabled
RSSDPART DC    XL2'0'                 TTA Partition Number
RSSDRSV3 DC    XL6'0'                 Reserved
RSSDCAP  DC    XL1'0'                 Drive Capibilities
RSSDRS1F EQU   X'01'                  .. Reserved
RSSDTTAF EQU   X'02'                  .. TTA Capable
RSSDRS4F EQU   X'04'                  .. Reserved
RSSDBTMF EQU   X'08'                  .. Write Buffered TM
RSSDSRAF EQU   X'10'                  .. Stop Read Ahead
RSSD349F EQU   X'20'                  .. 3490 Capable
RSSD359F EQU   X'40'                  .. 3590 Capable
RSSDVLSF EQU   X'80'                  .. Volsafe Capable
RSSDRSV4 DC    XL8'0'                 Reserved
RSSDHOST DC    XL1'0'                 Host Type ID
RDDSRSV4 DC    XL10'0'                Reserved
RSSDSIZE EQU   *-RSSDMAP              Size of area
         MEND
```