October 6th, 2020

IBM® TS7700 Series Statistical Data Format White Paper Version 5.1

Author:
Kousei Kawamura (e35223@jp.ibm.com)
Enterprise Storage Development
Tokyo, Japan

Target Audience

This document provides the definition of the TS7700 statistics records. The document is targeted for the following:

- System Administrators
- Product Field Engineers
- Statistics Analysis tool writers

Introduction

The IBM TotalStorage 3494 Virtual Tape Server (VTS) revolutionized the way zSeries customers utilized their tape resources. To help the customer monitor the performance of the VTS various statistics were developed. These statistics appeared in two forms, hourly records sent to the hosts known as SMF94 records and periodic real-time statistics available via the Library's web specialist or in an XML file for use by the Total Productivity Center (TPC).

For the next generation of VTS, the TS7700, the statistics design has been revisited. The useful statistics from the VTS have been retained and new statistics relevant for the TS7700 have been added. Also, both point-in-time (PIT) and historical statistics are recorded. The point-in-time records present data from the most recent interval providing speedometer like statistics. The historical statistics provide statistics where historical trends can be observed. These statistics are available to a host via the Bulk Volume Information Retrieval (BVIR) facility. Refer to the IBM® TS7700 Series Bulk Volume Information Retrieval Function User's Guide for more information concerning the BVIR facility as it relates to the TS7700 statistics.

It is assumed the reader of this document is familiar with the Virtual Tape Server and the TS7700. This document defines the statistics records, both point-in-time and historical. Some of the records are fixed length, while others vary in length depending upon the number of virtual devices, number of underlying physical libraries, number of clusters in a grid, and so forth. The Vnode statistics are presented first, both point-in-time and historical, followed by the Hnode statistics, both point-in-time and historical. Each Cluster in a Grid has its own set of Vnode and Hnode PIT and historical statistics.

For each record type, the table that describes the fields includes a column for when the data is updated or sampled. If the corresponding column entry is blank, then the field is updated or sampled at the interval of the statistics type being reported (15 seconds for point in time and 15 minutes for historical).

Please note that the statistical records described within this white paper may contain records, fields or field expansions that are not supported in the current release of the product. In general, those fields have been identified in the field descriptions, but you should also validate the availability of the function with your IBM representative.

SUMMARY OF CHANGES

V 1.0

- Initial release.

V 1.1

- Updated text to include that JB media is now supported.

V1.5

- Updated with new statistics delivered in Release 1.5
- Various clarifications

V1.6

- Remove unpopulated fields and change to reserved.
  - Hnode Grid Historical Record
  - Hnode Grid Point-In-Time (PIT) Record
- Various clarifications including:
  - "Awaiting Replication to Available Clusters" field in Hnode HSM Historical Record
  - Maximum number of clusters in a grid in now 4

V2.0

- Add updates for Release 2.0
  - Increase media type from 1 byte to 4 bytes in the Hnode Library Historical record
  - Add tape drive serial number and tape drive format
  - Add CPU usage percentage to Hnode HSM records (PIT and HIS)
  - Add throughput delays to Vnode Virtual Device records (PIT and HIS)
  - Minor clarifications and corrections

V2.0a

- Add updates for Release 2.0 PGA1
  - Update for TS1140 drives and media types
  - Remove 3590 Physical media types
  - The Hnode Library Historical record, library-pooling media containers, are no longer index based.
  - Update the Hnode Library Historical Record, pooling media containers to reflect that they are now packed.
  - Minor clarifications and corrections

V2.1

- Add updates for Release 2.1
  - Updates for sync mode copy
  - Added sync mode, immediate and deferred copy and count and MB transferred the Hnode Grid Historical record.
  - Removed references to the library manager, updated comments for Library container in the Hnode Library Historical record.
  - Changed the labels 'GB' to be '1000 MB' to match the definition
  - Added Hnode Export/Import Historical Record (Should have been added long ago)

V2.1a

- Additional updates for Release 2.1
  - Changed CPU Usage fields to reflect that it actually is the larger of CPU Usage % and TVC Cache throughput %.

V3.0

- Update for Release 3.0 PGA1
- Nomenclature changes, MB to MiB and KB to KiB
- Added Delay Interval Percent to the Historical Virtual Device record
- Added Maximum Disk Usage percentage, Write Overrun Throttling Reasons, Copy Throttling Reasons and Deferred Copy Throttling Reasons to the PIT HSM record
- Added Maximum CPU Usage Percentage, Average Maximum Disk Usage Percentage, Maximum Disk Usage Percentage, Write Overrun Throttle Reasons, Copy Throttle Reasons and Deferred Copy Throttle Reasons to the Historical HSM record
- Added the MI Historical record

V3.1

- Updated for release 3.1
- Add Ahead Count and Behind Count to the Virtual Device Point-In-Time record
- Add average and maximum ahead and behind counts to the Virtual Device Historical record
- Add Average time delayed copy queue age to Hnode Grid Point-In-Time record
- Add Total used cache and Total used Flash cache to Hnode HSM Historical Record
- Add Removed time delayed copies average age and Time delayed copies removal count to Hnode HSM Historical Record
- Add Time delayed copy queue to Hnode Grid Historical Record
- Updated to correct versions
- Minor corrections

V3.2

- Changed number of virtual device containers to be variable from 256 to 496 by 16 increments
- Updated historical statistics for release 3.2
  - Added “Extended HSM-Cache” container to Hnode HSM Historical record
  - Added “Extended HSM-Cache-Partition” container to Extended HSM-Cache container
  - Added “Extended HSM-Cache-Partition-Preference Group” containers to Extended HSM-Cache container
  - Added “Migrated Data” field to HSM-Cache-Partition container
- Updated point-in-time statistics for release 3.2
  - Added “Extended HSM” container to Hnode HSM Point-In-Time (PIT) record
  - Added “Offset to Extended HSM Container” to HSM container
- Removed unused MI Historical record

V3.2a

- Additional updates for Release 3.2
  - Added Hankie adapter type
  - Corrected a typographical error

V3.3

- Added new Device Class ID and Media Format to HSM Point-In-Time (PIT) record for supporting new E08 drives.
- Added new Physical Media Types, Device Class, Physical Media Identifier and First/Second Media Types to Library Historical record for supporting new E08 drives and JY, JD, JZ and JL media.
- Added Sunset Media Reclaim Threshold to Hnode Library Historical record. For supporting heterogeneous drive type support.
- Added description about GGM copy activity to "Data Transferred From a Cluster's Cache To Other Clusters as part of a Copy Operation" field.
- Updated Library Historical record version from 4 to 5.

V4.0

- Added physical device container for future use.
- Updated Library Historical record version from 5 to 6.

V4.0a

- Corrected errors.

V4.1.2

- Added compression container.

V4.2

- No change to statistical data format itself.
- Added some descriptions related to cloud tiering.

V5.0

- Added 64 bit statistics fields for V5000 support.
- Added a statistics field for DS8000 Object Data Offload support.
- Added temporary premigration throttle statistics.
- Added longest copy queue age statistics fields.
- Added the packet retransmission rate field.

V5.1

- Added Cloud Historical record.

Overview

This document outlines the information which the TS7700 returns for the request of statistical information. The information is divided into Point-in-Time (PIT) and Historical (HIS) data. The PIT information is intended to supply information about what the system is doing the instant the request is made to the system. This information is not persistent on the system, and is updated on a 15 second interval. This information focuses on the individual components of the system and their current activity.

The HIS information is intended to help with capacity planning, and tracking system use over an extended period of time. The information focuses more on the system as a whole, and the movement of data through the system. This information is kept on the system for 90 days, and is collected on a 15-minute interval basis.

With the support for Cluster Paired configured systems, some of the HSM records account for the fact that the two clusters are sharing the backend tape drive, library and media resources. For the PIT HSM records, there is an indicator as to which cluster is in control of the physical drive. While both clusters have access to each drive, and hence create records for all of them, only one cluster actively controls that drive at a time. Historical records for clusters in a pair configuration will report activity on a per cluster basis (i.e. individually).

Record Types and Sizes

This section provides a table indicating the record types, a description of the record type, the length of each record and the number of records produced.

Data Type (hexadecimal)	Description	Record Length (in bytes)	Number of Records
x01	Vnode Virtual Device Point in Time Record	96 + (number-of-virtual-devices-in-this-cluster x 32)	1 per Vnode
x02	Vnode Adapter Point in Time Record	384	1 per Vnode
x20	Vnode Virtual Device Historical Record	192	1 per Vnode
x21	Vnode Adapter Historical Record	384	1 per Vnode
x10	Hnode HSM Point in Time Record	96 + (number-of-physical-libraries-attached-to-this-cluster x 1568) + 32	1 per Hnode
x11	Hnode Grid Point in Time Record	96 + (number-of-clusters-in-the-grid x 128)	1 per Hnode
x30	Hnode HSM Historical Record	4688 + (number-of-compression-methods x 32)	1 per Hnode
x31	Hnode Export/Import Historical Record	128	1 per Hnode
x32	Hnode Library Historical Record	10048	1 per physical library attached to a Cluster
x33	Hnode Grid Historical Record	160 + (number-of-clusters-in-the-grid x 256)	1 per Hnode
x35	Hnode Cloud Historical Record	128 + (number-of-cloud-pools x 256)	1 per Hnode

Vnode Virtual Device Point In Time (PIT) Record

This Vnode Point-In-Time record has the following nested structure:

- Header
- General Information Container
- Virtual Device Container
    - Virtual Device 0 info
    - Virtual Device 1 info
    - .
    - .
    - .
    - Virtual Device xxx info

Bytes	Name	Description	When Data is Sampled/Updated
0-1	Length	This 2 byte hexadecimal field contains the length of this record. The length includes these 2 bytes.	
2	Version	This 1 byte hexadecimal field contains the version of the data presented in this record. The current version is set to x06.	
3	Data Type	This 1 byte hexadecimal field indicates the type of data contained in this record. For this record the value is set to x01 indicating this is a Vnode Virtual Device Point-In-Time record.	
4	Node ID	This 1 byte hexadecimal field indicates the Vnode ID which this interval's data represents. Valid values are x00 – x0F.	
5	Cluster ID	This 1 byte hexadecimal field indicates the Cluster ID which this Vnode is a part of. Valid values are x00 – x07.	
6-7	Interval Duration	This 2 byte hexadecimal field indicates the interval in seconds that this interval's data was taken over.	
8-11	Time Stamp	This 4 byte hexadecimal field indicates the end time of the interval this data was taken over. This value is the time in seconds since the Epoch (00:00:00 UTC, January 1, 1970)	
12-15	Machine Type	This 4 byte EBCDIC field contains this node's machine type. The field is left justified padded with EBCDIC blanks. Initially this field will be set to "3957".	
16-18	Machine Model	This 3 byte EBCDIC field contains this node's machine model. The field is left justified padded with EBCDIC blanks. Initially this field will be set to "V06".	
19-26	Machine Serial Number	This 8 character EBCDIC field contains the serial number of this node. This field is left justified and padded with EBCDIC blanks. The format is XX-YYYYY where XX is the plant of manufacture and the YYYYY is the sequence number of the node's machine. The dash character (-) is fixed.	
27-34	Code Level	This 8 byte hexadecimal field contains the code level of the TS7700. The 8 bytes are actually four 2-byte fields. Each 2-byte field represents a portion of the code level. The VE code level is expressed as Version.Release.Modification.Fix in a decimal form. For example the code level of 8.0.0.104 would be represented in the 8 bytes as: x0008000000000068.	
35-39	Grid Library Sequence Number	This 5 character EBCDIC field contains the Library Sequence Number of the Grid (Composite) library.	

<table>
<tr><th>Bytes</th><th>Name</th><th>Description</th><th>When Data is Sampled/Updated</th></tr>
<tr><td>40-44</td><td>Distributed Library Sequence Number</td><td>This 5 character EBCDIC field contains the Distributed Library Sequence Number for this Distributed Library ID</td><td></td></tr>
<tr><td>45-63</td><td>Reserved</td><td>All bytes set to x00</td><td></td></tr>
<tr><td colspan="3"><b>Vnode PIT General Information Container</b>
<b>Bytes 64-95</b>
<b>The fields below provide information concerning the configuration of the Vnode whose data is being reported.</b></td><td></td></tr>
<tr><td>64</td><td>Node State</td><td>This one byte hexadecimal field indicates the state of the Vnode at the end of this interval. Possible values are:
<table>
<tr><th>Value</th><th>Description</th></tr>
<tr><td>x00</td><td>Offline</td></tr>
<tr><td>x01</td><td>Online</td></tr>
<tr><td>x02</td><td>Going Offline</td></tr>
<tr><td>x03</td><td>Going Online</td></tr>
<tr><td>x04-xFE</td><td>Reserved</td></tr>
<tr><td>xFF</td><td>Node is not working at all</td></tr>
</table></td><td></td></tr>
<tr><td>65-66</td><td>Configured Maximum Throughput</td><td>This 2 byte hexadecimal field contains the maximum throughput for this Vnode. The value is expressed in MiB/Sec. (1 MiB = 1024x1024 bytes). This field is set to x0000 if there is no restriction for the maximum throughput. This is the value at the end of the interval.</td><td>Set to x0000 for first release.</td></tr>
<tr><td>67-68</td><td>Installed Virtual Devices</td><td>This 2 byte hexadecimal field indicates the number of installed virtual devices in this Vnode. This field can be used to determine how many Virtual Device containers will be attached to this record. This is the value at the end of the interval. This value varies from 256 to 496 by 16 increments.</td><td></td></tr>
<tr><td>69-72</td><td>Throughput delay</td><td>This 4 byte hexadecimal field indicates the throughput delay for this Vnode. This is the time delay that was used to limit the throughput so it doesn’t exceed the configured maximum throughput. This value is the sum of all the delays that occurred during the interval in milliseconds.</td><td>This value is updated every 15 seconds.</td></tr>
<tr><td>73-76</td><td>Ahead count</td><td>This 4 byte hexadecimal field indicates how many times the cluster was waiting on the FICON channel. This is the value at the end of the interval.
• When the write path drains faster than the host can fill it (empty buffer), the ahead count is increased by one. The 7700 is ahead of the host/channel.
• The opposite occurs during reads. The ahead count is incremented when the buffer fills with ready to read data.</td><td></td></tr>
<tr><td>77-80</td><td>Behind count</td><td>This 4 byte hexadecimal field indicates how many times the FICON channel TS7700 was waiting on the cluster. This can be used to determine if the cluster is busy. This is the value at the end of the interval.
• When the write path fills the buffer faster than it can drain to disk cache, the channel is CCR'ed (Channel Command Retry), the behind count is increased by one and the CCR event is freed once enough space frees up.
• The opposite occurs during reads. The behind count is incremented when the buffer is empty (nothing to give to the channel).</td><td></td></tr>
<tr><td>81-95</td><td>Reserved</td><td>All bytes set to x00</td><td></td></tr>
</table>

Bytes	Name	Description	When Data is Sampled/Updated
Virtual Device Container **Bytes from 96 and total length (N x 32), where N is the number of installed virtual devices and can be obtained from Installed Virtual Devices field in Vnode PIT General Information container. For example, if there are 128 installed virtual devices there will be 128 sets of data with 32 bytes each which totals 4096 bytes. There is a maximum of 496 virtual devices per Vnode.** **This next segment of the record contains one set of data for each virtual device installed in the Vnode as defined in bytes 67-68 above. Each set of data contains 32 bytes. The following fields define the 32 bytes of data and are numbered starting with byte 0. The first virtual device's data can be found in bytes 96-127, the second device's data can be found in bytes 128-159, and so forth.**			
0-9	Mounted Volume	This 10 byte EBCDIC field contains the volser of the logical volume, if any, that is currently mounted in the virtual device or was most recently mounted in the device. For device mount states that indicate a mount in progress or is mounted (x01, x02, x03, x06 – see byte 11) this field will contain the volser of the volume that is in the process of being mounted. For device mount states that indicate a device isn't mounted or in the process of being mounted (x00, x04, x05 – see byte 11) this field will contain the volser of the last successfully mounted volume, if any, or will be filled with EBCDIC blanks. This field is left justified and padded with EBCDIC blanks. This is the value at the end of the interval.	Updated whenever the logical volume changes in the virtual device.
10	Cluster Access Point	This 1 byte hexadecimal field indicates the Cluster ID which is sourcing or has most recently sourced the logical volume for a mount. In the case of the most recently sourced Cluster, the current device mount state will indicate "Device unloaded, failed or cancelled". (see byte 11 below) Valid values for this field are x00 – x07. The subsystem has the ability to access a logical volume in any Cluster from any Cluster. Where a logical volume is sourced from is based on a set of criteria including volume consistency, access policies, and so forth. This is the value at the end of the interval.	Updated for each virtual mount.

<table>
<tr><th>Bytes</th><th>Name</th><th>Description</th><th>When Data is Sampled/Updated</th></tr>
<tr><td>11</td><td>Device Mount State</td><td>This 1 byte hexadecimal field indicates the mount state of the virtual device. Valid values are as follows:
<table>
<tr><th>Value</th><th>Description</th></tr>
<tr><td>x00</td><td>Device unloaded. This indicates the previous mount completed successfully and the virtual device is now unloaded. This is also reported when the virtual device has not mounted any volume yet and isn’t in the process of performing a mount.</td></tr>
<tr><td>x01</td><td>Mount request being processed</td></tr>
<tr><td>x02</td><td>Mount accepted and initial status received</td></tr>
<tr><td>x03</td><td>Mount in progress (Device mount command received)</td></tr>
<tr><td>x04</td><td>Mount failed. This indicates the previous mount failed (error reported by device) and the virtual device is unloaded.</td></tr>
<tr><td>x05</td><td>Mount cancelled. This indicates the previous mount request was cancelled before the mount occurred. (Demount received before Unload) The virtual device is now unloaded.</td></tr>
<tr><td>x06</td><td>Mounted</td></tr>
<tr><td></td><td>All other values are reserved</td></tr>
</table>
For a non-configured or non-installed device, this field will indicate Device unloaded.

This is the value at the end of the interval.</td><td>Updated whenever the virtual device mount state changes.</td></tr>
</table>

<table>
<tr><th>Bytes</th><th>Name</th><th>Description</th><th>When Data is Sampled/Updated</th></tr>
<tr><td>12-13</td><td>Device Flags</td><td>This 2 byte field contains 16 single bit flags related to the current state of the virtual device. Several bits can be set to 1 at the same time. The bit mask values are defined as follows:
<table>
<tr><th>Mask Value</th><th>Description</th></tr>
<tr><td>x8000</td><td>Device is ready. Provides information as to the mount state of the virtual drive. If the bit is set, then a virtual volume is loaded into the drive and is ready for host I/O activity.</td></tr>
<tr><td>x4000</td><td>Device is write-protected. This flag indicates the logical write protect state of the virtual drive/virtual volume. This prevents any modifications to a virtual volume loaded into this drive.</td></tr>
<tr><td>x2000</td><td>Write data is in the buffer. This flag indicates that the control unit has write data for this virtual volume held in the controller’s memory, and not yet committed to the virtual device. This is normal for streaming write data to the device.</td></tr>
<tr><td>x1000</td><td>Write mode – The last IO to the device was a write operation. Indicates that the virtual device is in write mode. The device enters write mode upon the first write I/O operation, and stays in this mode until a non-write command (position change, read, etc) is encountered.</td></tr>
<tr><td>x0800</td><td>Volume is in the LEOP (Logical End of Partition). Indicates that the current block position of the virtual tape is in the LEOP region of the virtual volume, and the host should be performing end of volume processing on the virtual volume.</td></tr>
<tr><td>x0400</td><td>Volume is at BOT (Beginning of Tape). Indicates the virtual volume loaded is currently at the beginning of the virtual volume. This would be the tape position after a mount request, or if the host repositioned the volume back to block 0.</td></tr>
<tr><td>x0200</td><td>Device is fenced. The control unit presents unit check status with associated sense data indicating ERA 47, Volume Fenced, if a condition has occurred which has resulted in the loss of volume integrity due to lost positioning or assignment. The control unit prevents further access to the tape volume by generating deferred unit checks with associated sense data indicating ERA 47, Volume Fenced, for all eligible commands until the condition is reset or until the cartridge is unloaded. The original condition which subsequently caused ERA 47, Volume Fenced, to be presented has already been indicated by a previous unit check and associated sense data.</td></tr>
<tr><td>x0100</td><td>Device is in stand alone mount mode. This indicates that the virtual volume in this virtual drive was loaded as part of an operator request through the Management Interface, and not via a connected host. Standalone mounts are required to be able to IPL the operating system image from a virtual tape device, without the operating system having to have tape mount capabilities.</td></tr>
<tr><td>x0080</td><td>Sync Mode Copy mode. This indicates the virtual volume in this virtual drive has sync mode copy in effect.</td></tr>
<tr><td></td><td>All other bits are reserved</td></tr>
</table>
For a non-configured or non-installed device, this field will be set to x0000.

This is the value at the end of the interval.</td><td>Updated whenever the virtual device mount state changes.</td></tr>
</table>

Bytes	Name	Description	When Data is Sampled/Updated
14	Buffer CCR Conditions	This one byte hexadecimal field contains the number of times the virtual device had to CCR (Channel Command Retry) the channel due to a buffer condition during this interval. If more than 255 buffer CCRs occur during the interval this field will indicate 255 buffers CCRs (xFF). This field can be used in conjunction with the Device Flags field, Write mode bit, to determine if the device was in write mode (buffer full condition) or read mode (buffer empty condition) during the interval. This value is reset to 0 at the beginning of the interval.	Count is incremented whenever a Buffer CCR condition occurs.
15-16	Channel Bytes Read	This 2 byte hexadecimal field contains the number of bytes transferred from the virtual device to the channel for a read from this device. The value is reported in increments of 100KiB (100 x 1024). Any residual data will cause the value to be rounded up to the next higher value. This value is reset to 0 at the beginning of the interval	Count is incremented for each read of a logical volume's data.
17-18	Channel Bytes Written	This 2 byte hexadecimal field contains the number of bytes transferred from the channel to the virtual device for a write to this device. The value is reported in increments of 100KiB (100 x 1024). Any residual data will cause the value to be rounded up to the next higher value. This value is reset to 0 at the beginning of the interval.	Count is incremented for each write of data to a logical volume.
19	Synchronized Cluster Access Point	This 1 byte hexadecimal field indicates the cluster ID which is synchronized to the cluster which is sourcing or has most recently sourced the logical volume for a mount with sync mode copy enabled. In the case of the most recently sourced cluster, the current device mount state will indicate "Device unloaded, failed or cancelled". (See byte 11 above) . If a sync mode copy is not in progress, or was not in effect for the last mount, then this value will be set to xFF. Valid values for this field are x00 – x07 and xFF. The subsystem has the ability to access a logical volume in any cluster from any cluster. Where a logical volume is sourced from is based on a set of criteria including volume consistency, access policies, and so forth. This is the value at the end of the interval.	This value is set at mount time.
20-31	Reserved	All bytes set to x00.	

Vnode Adapter Point-In-Time (PIT) Record

This Vnode Point-In-Time record has the following nested structure:

- Header
- Adapter Container
  - Adapter 0 general information
    - Port 0 information
    - Port 1 information
  - Adapter 1 general information
    - Port 0 information
    - Port 1 information
  - Adapter 2 general information
    - Port 0 information
    - Port 1 information
  - Adapter 3 general information
    - Port 0 information
    - Port 1 information

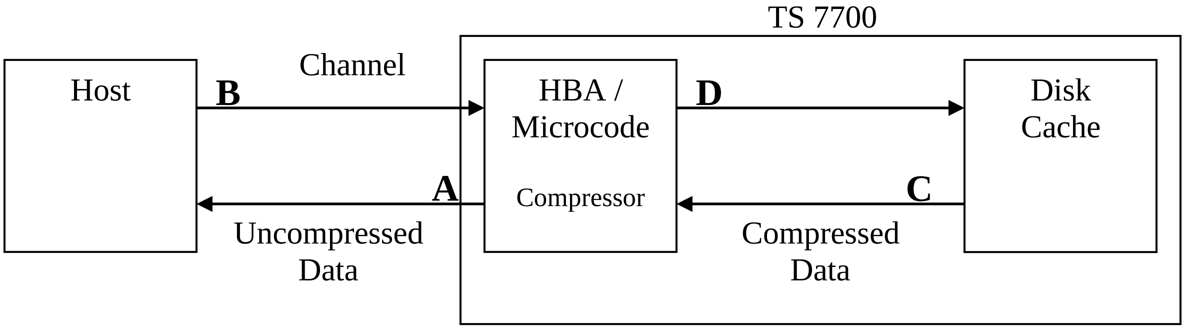
Bytes	Name	Description	When Data is Sampled/Updated
0-1	Length	This 2 byte hexadecimal field contains the length of this record. The length includes these 2 bytes.	
2	Version	This 1 byte hexadecimal field contains the version of the data presented in this record. The current version is set to x02.	
3	Data Type	This 1 byte hexadecimal field indicates the type of data contained in this record. For this record the value is set to x02 indicating this is a Vnode Adapter Point-In-Time record.	
4	Node ID	This 1 byte hexadecimal field indicates the Vnode ID which this interval’s data represents. Valid values are x00 – x0F.	
5	Cluster ID	This 1 byte hexadecimal field indicates the Cluster ID which this Vnode is a part of. Valid values are x00 – x07.	
6-7	Interval Duration	This 2 byte hexadecimal field indicates the interval in seconds that this interval’s data was taken over.	
8-11	Time Stamp	This 4 byte hexadecimal field indicates the end time of the interval this data was taken over. This value is the time in seconds since the Epoch (00:00:00 UTC, January 1, 1970)	
12-15	Machine Type	This 4 byte EBCDIC field contains this node’s machine type. The field is left justified padded with EBCDIC blanks. Initially this field will be set to “3957”.	
16-18	Machine Model	This 3 byte EBCDIC field contains this node’s machine model. The field is left justified padded with EBCDIC blanks. Initially this field will be set to “V06”.	
19-26	Machine Serial Number	This 8 character EBCDIC field contains the serial number of this node. This field is left justified and padded with EBCDIC blanks. The format is XX-YYYYY where XX is the plant of manufacture and the YYYYY is the sequence number of the node’s machine. The dash character (-) is fixed.	
27-34	Code Level	This 8 byte hexadecimal field contains the code level of the TS7700. The 8 bytes are actually four 2-byte fields. Each 2-byte field represents a portion of the code level. The VE code level is expressed as Version.Release.Modification.Fix in a decimal form. For example the code level of 8.0.0.104 would be represented in the 8 bytes as: x0008000000000068.	

<table>
<tr><th>Bytes</th><th>Name</th><th>Description</th><th>When Data is Sampled/Updated</th></tr>
<tr><td>35-39</td><td>Grid Library Sequence Number</td><td>This 5 character EBCDIC field contains the Library Sequence Number of the Grid (Composite) library.</td><td></td></tr>
<tr><td>40-44</td><td>Distributed Library Sequence Number</td><td>This 5 character EBCDIC field contains the Distributed Library Sequence Number for this Distributed Library ID</td><td></td></tr>
<tr><td>45-63</td><td>Reserved</td><td>All bytes set to x00</td><td></td></tr>
<tr><td colspan="3"><h2>Adapter Container</h2><b>Bytes 64-383 (4 sets of data x 80 bytes/set = 320 bytes)</b>

<b>This next set of bytes contains information for up to 4 host bus adapters (HBA). For each adapter there is data for up to 2 ports on the adapter. Each set of data contains 80 bytes. The following fields define the 80 bytes of data and are numbered starting with byte 0. The first set of data is for adapter 0, the second for adapter 1, and so forth. The first HBA’s data can be found in bytes 64-143, the second adapter’s data can be found in bytes 144-223, and so forth.</b></td><td></td></tr>
<tr><td>0</td><td>Adapter Type</td><td>This 1 byte hexadecimal field identifies the type of Host Bus Adapter (HBA). The possible values are:
<table>
<tr><th>Value</th><th>Description</th></tr>
<tr><td>x00</td><td>No adapter installed</td></tr>
<tr><td>x01-x08</td><td>Reserved</td></tr>
<tr><td>x09</td><td>FICON – 1 Port (Arctic Circle)</td></tr>
<tr><td>x0A</td><td>FICON – 2 Port (Yukon)</td></tr>
<tr><td>x10</td><td>FICON – 4 Port (Hankie)</td></tr>
<tr><td></td><td>All other values are reserved</td></tr>
</table>
This is the value at the end of the interval.</td><td></td></tr>
<tr><td>1</td><td>Adapter State</td><td>This 1 byte hexadecimal field identifies the current state of the adapter. The possible values are:
<table>
<tr><th>Value</th><th>Description</th></tr>
<tr><td>x00</td><td>No adapter installed</td></tr>
<tr><td>x01</td><td>The adapter is online</td></tr>
<tr><td>x02</td><td>The adapter is offline</td></tr>
<tr><td>x03</td><td>The adapter is not working at all</td></tr>
<tr><td>x04</td><td>The adapter is reloading itself</td></tr>
<tr><td>x05</td><td>The adapter is in a Check1 condition</td></tr>
<tr><td></td><td>All other values are reserved</td></tr>
</table>
This is the value at the end of the interval.</td><td></td></tr>
<tr><td>2</td><td>HBA Drawer</td><td>This 1 byte hexadecimal field indicates which drawer the HBA is located in. The possible values are:
<table>
<tr><th>Value</th><th>Description</th></tr>
<tr><td>x00</td><td>The HBA is in the left drawer when looking at the drawers from the back which is the side the cables plug into.</td></tr>
<tr><td>x01</td><td>The HBA is in the right drawer when looking at the drawers from the back which is the side the cables plug into.</td></tr>
<tr><td></td><td>All other values are reserved.</td></tr>
</table>
This is the value at the end of the interval.</td><td></td></tr>
<tr><td>3</td><td>HBA Slot Number</td><td>This 1 byte hexadecimal field indicates the physical slot number of the HBA within its drawer. This is the value at the end of the interval.</td><td></td></tr>
</table>

<table>
<tr><th>Bytes</th><th>Name</th><th>Description</th><th>When Data is Sampled/Updated</th></tr>
<tr><td>4</td><td>Host throttle</td><td>This 1 byte hexadecimal field indicates if host throttle is being applied. This is the value at the end of the interval.<table><tr><th>Value</th><th>Description</th></tr><tr><td>x00</td><td>No host throttle is applied</td></tr><tr><td>x01</td><td>A throttle is being applied to Host IO</td></tr><tr><td></td><td>All other values are reserved.</td></tr></table></td><td></td></tr>
<tr><td>5-15</td><td>Reserved</td><td>All bytes set to x00.</td><td></td></tr>
<tr><td colspan="3"><b>Adapter-Port Container</b>
<b>Relative bytes 16 – 79 (2 sets of data x 32 bytes/set = 64 bytes)</b>
<b>This next set of bytes contains information for up to 2 ports on the HBA. Each set of data contains 32 bytes. The following fields define the 32 bytes and are numbered starting with byte 0. The first port’s data can be found in relative bytes 16-47 and the second port’s data can be found in relative bytes 48-79.</b></td><td></td></tr>
<tr><td>0-1</td><td>RCD Interface ID</td><td>This 2 byte hexadecimal field contains the internal ID of the HBA port that is reported in the RCD (Read Configuration Data), General NEQ (Node Element Qualifier) record. This is the value at the end of the interval.</td><td></td></tr>
<tr><td>2-4</td><td>Reserved</td><td>All bytes set to x00</td><td></td></tr>
<tr><td>5-8</td><td>Bytes Read by the Channel</td><td>This 4 byte hexadecimal field contains the number of bytes transferred to the channel from this HBA port as part of a read operation. This is the value after the data has been decompressed by the HBA. The value is reported in increments of 4KiB (4 x 1024). Any residual data will cause the value to be rounded up to the next higher value.
In the diagram below, Bytes Read by the Channel is indicated by the <b>A</b> label.
This value is reset to 0 at the beginning of the interval.</td><td>Count is incremented for each block of data read.</td></tr>
<tr><td>9-12</td><td>Bytes Written by the Channel</td><td>This 4 byte hexadecimal field contains the number of bytes transferred from the channel to this HBA port as part of a write operation. This is the value before the effect of the HBA compression. The value is reported in increments of 4KiB (4 x 1024). Any residual data will cause the value to be rounded up to the next higher value.
In the diagram below, Bytes Written by the Channel is indicated by the <b>B</b> label.
This value is reset to 0 at the beginning of the interval.</td><td>Count is incremented for each block of data written.</td></tr>
<tr><td>13-16</td><td>Bytes Read from the Disk Cache</td><td>This 4 byte hexadecimal field contains the number of bytes transferred from disk cache to this HBA port as part of a read operation. The value is for data previously compressed by the HBA or microcode. The value is reported in increments of 4KiB (4 x 1024). Any residual data will cause the value to be rounded up to the next higher value.
In the diagram below, Bytes Read from Disk Cache is indicated by the <b>C</b> label.
This value is reset to 0 at the beginning of the interval.</td><td>Count is incremented for each block of data read.</td></tr>
</table>

Bytes	Name	Description	When Data is Sampled/Updated
17-20	Bytes Written to Disk Cache	This 4 byte hexadecimal field contains the number of bytes transferred to disk cache from this HBA port as part of a write operation. The value is for data compressed by the HBA or microcode. The value is reported in increments of 4KiB (4 x 1024). Any residual data will cause the value to be rounded up to the next higher value. In the diagram below, Bytes Written to Disk Cache is indicated by the **D** label. This value is reset to 0 at the beginning of the interval.	Count is incremented for each block of data written.
21-31	Reserved	All bytes set to x00.	



Vnode Virtual Device Historical Record

This Vnode historical record has the following nested structure:

- Header
- Virtual Device Container

Bytes	Name	Description	When Data is Sampled/Updated
0-1	Length	This 2 byte hexadecimal field contains the length of this record. The length includes these 2 bytes.	
2	Version	This 1 byte hexadecimal field contains the version of the data presented in this record. The current version is set to x05.	
3	Data Type	This 1 byte hexadecimal field indicates the type of data contained in this record. For this record the value is set to x20 indicating this is a Vnode Virtual Device Historical record.	
4	Node ID	This 1 byte hexadecimal field indicates the Vnode ID which this interval's data represents. Valid values are x00 – x0F.	
5	Cluster ID	This 1 byte hexadecimal field indicates the Cluster ID which this Vnode is a part of. Valid values are x00 – x07.	
6-7	Interval Duration	This 2 byte hexadecimal field indicates the interval in seconds that this interval's data was taken over.	
8-11	Time Stamp	This 4 byte hexadecimal field indicates the end time of the interval this data was taken over. This value is the time in seconds since the Epoch (00:00:00 UTC, January 1, 1970)	
12-15	Machine Type	This 4 byte EBCDIC field contains this node's machine type. The field is left justified padded with EBCDIC blanks. Initially this field will be set to "3957".	
16-18	Machine Model	This 3 byte EBCDIC field contains this node's machine model. The field is left justified padded with EBCDIC blanks. Initially this field will be set to "V06".	
19-26	Machine Serial Number	This 8 character EBCDIC field contains the serial number of this node. This field is left justified and padded with EBCDIC blanks. The format is XX-YYYYY where XX is the plant of manufacture and the YYYYY is the sequence number of the node's machine. The dash character (-) is fixed.	
27-34	Code Level	This 8 byte hexadecimal field contains the code level of the TS7700. The 8 bytes are actually four 2-byte fields. Each 2-byte field represents a portion of the code level. The VE code level is expressed as Version.Release.Modification.Fix in a decimal form. For example the code level of 8.0.0.104 would be represented in the 8 bytes as: x0008000000000068.	
35-39	Grid Library Sequence Number	This 5 character EBCDIC field contains the Library Sequence Number of the Grid (Composite) library.	
40-44	Distributed Library Sequence Number	This 5 character EBCDIC field contains the Distributed Library Sequence Number for this Distributed Library ID	
45-63	Reserved	All bytes set to x00	

Bytes	**Name**	**Description**	**When Data is Sampled/Updated**
Vnode Virtual Device Container **Bytes 64-191**			
64-65	Installed Virtual Devices	This 2 byte hexadecimal field indicates the number of installed virtual devices in this Vnode. This is the value at the end of the interval.	
66-69	Virtual Device Type	This 4 byte EBCDIC field indicates the device type emulated by the virtual devices. Initially this is set to “3490”. This field is left justified and padded with blanks.	
70-72	Virtual Device Model	This 3 byte EBCDIC field indicates the device model emulated by the virtual devices. Initially this is set to “C2A”. This field is left justified and padded with blanks. This is the value at the end of the interval.	
73-80	Channel Blocks Written 1-2048 byte range	This 8 byte hexadecimal field indicates the number of channel blocks written to all the virtual devices in this Vnode that had a size of between 1 and 2048 bytes inclusive for this interval. This value is reset to 0 at the beginning of the interval.	Count is incremented for each block of data written that fits the size.
81-88	Channel Blocks Written 2049-4096 byte range	This 8 byte hexadecimal field indicates the number of channel blocks written to all the virtual devices in this Vnode that had a size of between 2049 and 4096 bytes inclusive for this interval. This value is reset to 0 at the beginning of the interval.	Count is incremented for each block of data written that fits the size.
89-96	Channel Blocks Written 4097-8192 byte range	This 8 byte hexadecimal field indicates the number of channel blocks written to all the virtual devices in this Vnode that had a size of between 4097 and 8192 bytes inclusive for this interval. This value is reset to 0 at the beginning of the interval.	Count is incremented for each block of data written that fits the size.
97-104	Channel Blocks Written 8193-16384 byte range	This 8 byte hexadecimal field indicates the number of channel blocks written to all the virtual devices in this Vnode that had a size of between 8193 and 16384 bytes inclusive for this interval. This value is reset to 0 at the beginning of the interval.	Count is incremented for each block of data written that fits the size.
105-112	Channel Blocks Written 16385-32768 byte range	This 8 byte hexadecimal field indicates the number of channel blocks written to all the virtual devices in this Vnode that had a size of between 16385 and 32768 bytes inclusive for this interval. This value is reset to 0 at the beginning of the interval.	Count is incremented for each block of data written that fits the size.
113-120	Channel Blocks Written 32769-65536 byte range	This 8 byte hexadecimal field indicates the number of channel blocks written to all the virtual devices in this Vnode that had a size of between 32769 and 65536 bytes inclusive for this interval. This value is reset to 0 at the beginning of the interval.	Count is incremented for each block of data written that fits the size.
121-128	Channel Blocks Written above 65536 byte range	This 8 byte hexadecimal field indicates the number of channel blocks written to all the virtual devices in this Vnode that had a size of 65537 or higher for this interval. This value is reset to 0 at the beginning of the interval.	Count is incremented for each block of data written that fits the size.

<table>
<tr><th>Bytes</th><th>Name</th><th>Description</th><th>When Data is Sampled/Updated</th></tr>
<tr><td>129-130</td><td>Configured Maximum Throughput</td><td>This 2 byte hexadecimal field contains the current maximum throughput for this Vnode. The value is expressed in MiB/Sec. (1 MiB = 1024x1024 bytes). This field is set to x0000 if there is no restriction for the maximum throughput. This is the value at the end of the interval.</td><td>This will be set to x0000 for the first release.</td></tr>
<tr><td>131-132</td><td>Minimum Virtual Devices Mounted</td><td>This 2 byte hexadecimal field indicates the minimum number of virtual devices that were mounted at the same time over the interval.</td><td>The count of mounted virtual devices is sampled every 15 seconds. The min/max/avg is updated over the interval.</td></tr>
<tr><td>133-134</td><td>Maximum Virtual Devices Mounted</td><td>This 2 byte hexadecimal field indicates the maximum number of virtual devices that were mounted at the same time over the interval.</td><td>The count of mounted virtual devices is sampled every 15 seconds. The min/max/avg is updated over the interval.</td></tr>
<tr><td>135-136</td><td>Average Virtual Devices Mounted</td><td>This 2 byte hexadecimal field indicates the average number of virtual devices that were mounted at the same time over the interval. The average is calculated by recording the number of mounted devices on a periodic basis then averaging it over the interval.</td><td>The count of mounted virtual devices is sampled every 15 seconds. The min/max/avg is updated over the interval.</td></tr>
<tr><td>137-140</td><td>Maximum Delay</td><td>This 4 byte hexadecimal field indicates the maximum total throughput delay over the 15 minute interval. The total throughput delay is accumulated for each 15 second period within the 15 minute interval and the 15 second period with the largest delay is displayed. This value is in milliseconds.</td><td></td></tr>
<tr><td>141-144</td><td>Average Delay</td><td>This 4 byte hexadecimal field indicates the average total throughput delay over the 15 minute interval. The total throughput delay is accumulated for each 15 second period within the 15 minute interval and the average of these 15 second periods is displayed. This value is in milliseconds.</td><td></td></tr>
<tr><td>145</td><td>Delay Interval Percentage</td><td>This 1 byte hexadecimal field indicates the percentage of 15 second periods within the 15 minute interval which contained a non-zero delay. It does not represent the weight of the delay. It simply represents how often some level of delay is active.
<table>
<tr><th>Value</th><th>Description</th></tr>
<tr><td>0 - 100</td><td>Percentage of time host throttling had been applied.</td></tr>
</table></td><td></td></tr>
<tr><td>146-149</td><td>Maximum ahead count</td><td>This 4 byte hexadecimal field indicates the maximum number of times that the cluster was ahead of the FICON channel during the interval.
• When the write path drains faster than the host can fill it (empty buffer), the ahead count is increased by one. The 7700 is ahead of the host/channel.
• The opposite occurs during reads. The ahead count is incremented when the buffer fills with ready to read data.</td><td>This is the maximum of the samples over the past 15 minutes. Samples are taken every 15 seconds.</td></tr>
</table>

Bytes	**Name**	**Description**	**When Data is Sampled/Updated**
150-153	Average ahead count	This 4 byte hexadecimal field indicates the average number of times that the cluster was ahead of the FICON channel during the interval. • When the write path drains faster than the host can fill it (empty buffer), the ahead count is increased by one. The 7700 is ahead of the host/channel. • The opposite occurs during reads. The ahead count is incremented when the buffer fills with ready to read data.	This is the average of the samples over the past 15 minutes. Samples are taken every 15 seconds.
154-157	Maximum behind count	This 4 byte hexadecimal field indicates the maximum number of times that the cluster was behind the FICON channel during the interval. • When the write path fills the buffer faster than it can drain to disk cache, the channel is CCR'ed (Channel Command Retry), the behind count is increased by one and the CCR event is freed once enough space frees up. • The opposite occurs during reads. The behind count is incremented when the buffer is empty (nothing to give to the channel).	This is the maximum of the samples over the past 15 minutes. Samples are taken every 15 seconds.
158-161	Average behind count	This 4 byte hexadecimal field indicates the average number of times that the cluster was behind the FICON channel during the interval. • When the write path fills the buffer faster than it can drain to disk cache, the channel is CCR'ed (Channel Command Retry), the behind count is increased by one and the CCR event is freed once enough space frees up. • The opposite occurs during reads. The behind count is incremented when the buffer is empty (nothing to give to the channel).	This is the average of the samples over the past 15 minutes. Samples are taken every 15 seconds.
162-191	Reserved	All bytes set to x00.	

Vnode Adapter Historical Record

This Vnode historical record has the following nested structure:

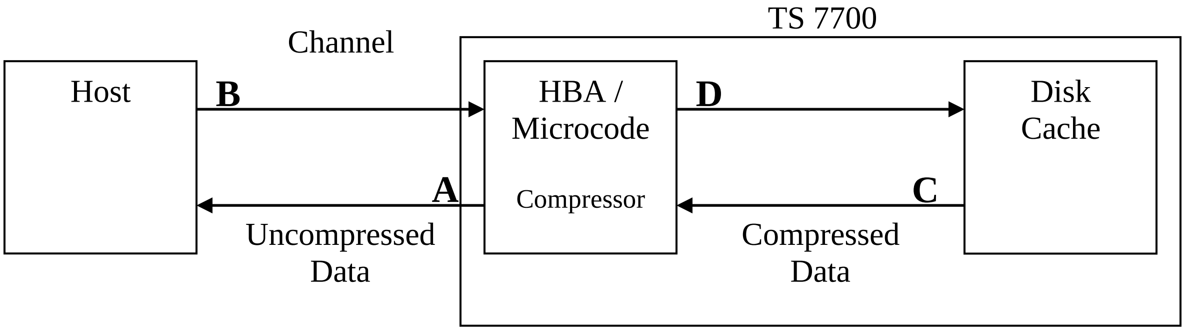
- Header
- Adapter 0 Container
    - Adapter-Port 0 Container
    - Adapter-Port 1 Container
- Adapter 1 Container
    - Adapter-Port 0 Container
    - Adapter-Port 1 Container
- Adapter 2 Container
    - Adapter-Port 0 Container
    - Adapter-Port 1 Container
- Adapter 3 Container
    - Adapter-Port 0 Container
    - Adapter-Port 1 Container

Bytes	Name	Description	When Data is Sampled/Updated
0-1	Length	This 2 byte hexadecimal field contains the length of this record. The length includes these 2 bytes.	
2	Version	This 1 byte hexadecimal field contains the version of the data presented in this record. The current version is set to x02.	
3	Data Type	This 1 byte hexadecimal field indicates the type of data contained in this record. For this record the value is set to x21 indicating this is a Vnode Adapter Historical record.	
4	Node ID	This 1 byte hexadecimal field indicates the Vnode ID which this interval's data represents. Valid values are x00 – x0F.	
5	Cluster ID	This 1 byte hexadecimal field indicates the Cluster ID which this Vnode is a part of. Valid values are x00 – x07.	
6-7	Interval Duration	This 2 byte hexadecimal field indicates the interval in seconds that this interval's data was taken over.	
8-11	Time Stamp	This 4 byte hexadecimal field indicates the end time of the interval this data was taken over. This value is the time in seconds since the Epoch (00:00:00 UTC, January 1, 1970)	
12-15	Machine Type	This 4 byte EBCDIC field contains this node's machine type. The field is left justified padded with EBCDIC blanks. Initially this field will be set to "3957".	
16-18	Machine Model	This 3 byte EBCDIC field contains this node's machine model. The field is left justified padded with EBCDIC blanks. Initially this field will be set to "V06".	
19-26	Machine Serial Number	This 8 character EBCDIC field contains the serial number of this node. This field is left justified and padded with EBCDIC blanks. The format is XX-YYYYY where XX is the plant of manufacture and the YYYYY is the sequence number of the node's machine. The dash character (-) is fixed.	
27-34	Code Level	This 8 byte hexadecimal field contains the code level of the TS7700. The 8 bytes are actually four 2-byte fields. Each 2-byte field represents a portion of the code level. The VE code level is expressed as Version.Release.Modification.Fix in a decimal form. For example the code level of 8.0.0.104 would be represented in the 8 bytes as: x0008000000000068.	

<table>
<tr><th>Bytes</th><th>Name</th><th>Description</th><th>When Data is Sampled/Updated</th></tr>
<tr><td>35-39</td><td>Grid Library Sequence Number</td><td>This 5 character EBCDIC field contains the Library Sequence Number of the Grid (Composite) library.</td><td></td></tr>
<tr><td>40-44</td><td>Distributed Library Sequence Number</td><td>This 5 character EBCDIC field contains the Distributed Library Sequence Number for this Distributed Library ID</td><td></td></tr>
<tr><td>45-63</td><td>Reserved</td><td>All bytes set to x00</td><td></td></tr>
<tr><td colspan="3"><h1>Vnode Adapter Container</h1><p><b>Bytes 64-383 (4 sets x 80 bytes/set = 320 bytes)</b></p><p><b>This next set of bytes contains information for 4 host bus adapters (HBA). For each adapter there is data for up to 2 ports. Each set of data contains 80 bytes. The following fields define the bytes of data and are numbered starting with byte 0. The first set of data is for the adapter 0, the second for adapter 1, and so forth. The first adapter’s data can be found in bytes 64-143, the second adapter’s data can be found in bytes 144-223, and so forth.</b></p></td><td></td></tr>
<tr><td>0</td><td>Adapter Type</td><td>This 1 byte hexadecimal field identifies the type of Host Bus Adapter (HBA) this data is for. The possible values are:
<table>
<tr><th>Value</th><th>Description</th></tr>
<tr><td>x00</td><td>No adapter installed</td></tr>
<tr><td>x02-x08</td><td>Reserved</td></tr>
<tr><td>x09</td><td>FICON – 1 Port (Arctic Circle)</td></tr>
<tr><td>x0A</td><td>FICON – 2 Port (Yukon)</td></tr>
<tr><td>x10</td><td>FICON – 4 Port (Hankie)</td></tr>
<tr><td></td><td>All other values are reserved</td></tr>
</table>
This is the value at the end of the interval.</td><td></td></tr>
<tr><td>1</td><td>Adapter State</td><td>This 1 byte hexadecimal field identifies the current state of the adapter. The possible values are:
<table>
<tr><th>Value</th><th>Description</th></tr>
<tr><td>x00</td><td>No adapter installed</td></tr>
<tr><td>x01</td><td>The adapter is online</td></tr>
<tr><td>x02</td><td>The adapter is offline</td></tr>
<tr><td>x03</td><td>The adapter is not working at all</td></tr>
<tr><td>x04</td><td>The adapter is reloading itself</td></tr>
<tr><td>x05</td><td>The adapter is in a Check1 condition</td></tr>
<tr><td></td><td>All other values are reserved</td></tr>
</table>
This is the value at the end of the interval.</td><td></td></tr>
<tr><td>2</td><td>HBA Drawer</td><td>This 1 byte hexadecimal field indicates which drawer the HBA is located in. The possible values are:
<table>
<tr><th>Value</th><th>Description</th></tr>
<tr><td>x00</td><td>The HBA is in the left drawer when looking at the drawers from the back which is the side the cables plug into.</td></tr>
<tr><td>x01</td><td>The HBA is in the right drawer when looking at the drawers from the back which is the side the cables plug into.</td></tr>
<tr><td></td><td>All other values are reserved.</td></tr>
</table>
This is the value at the end of the interval.</td><td></td></tr>
<tr><td>3</td><td>HBA Slot Number</td><td>This 1 byte hexadecimal field indicates the physical slot number of the HBA within its drawer. This is the value at the end of the interval.</td><td></td></tr>
<tr><td>4-15</td><td>Reserved</td><td>All bytes set to x00.</td><td></td></tr>
</table>

Bytes	**Name**	**Description**	**When Data is Sampled/Updated**
Vnode Adapter-Port Container **Relative bytes 16-79 (2 sets x 32 bytes/set = 64 bytes)** **This next set of bytes contains information for up to 2 ports on the HBA. Each set of data contains 32 bytes. The following fields define the 32 bytes and are numbered starting with byte 0. The first port’s data can be found in relative bytes 16-47 and the second port’s data can be found in relative bytes 48-79.**			
0-1	RCD Interface ID	This 2 byte hexadecimal field contains the internal ID of the HBA port that is reported in the RCD (Read Configuration Data), General NEQ (Node Element Qualifier) record. This is the value at the end of the interval.	
2	Maximum Data Rate	This 1 byte hexadecimal field indicates the maximum data rate the FICON port is capable of at the end of the interval. The value is reported in Giga-bits (Gb) per second.	.
3	Actual Data Rate	This 1 byte hexadecimal field indicates the actual data rate of the FICON port at the end of the interval. The value is reported in Giga-bits (Gb) per second. A value of x00 in this field indicates that auto-negotiate is enabled.	
4-7	Bytes Read by the Channel	This 4 byte hexadecimal field contains the number of bytes transferred to the channel from this HBA port as part of a read operation. This is the value after the data has been decompressed by the HBA. The value is reported in increments of 4KiB (4 x 1024). Any residual data will cause the value to be rounded up to the next higher value. In the diagram below, Bytes Read by the Channel is indicated by the **A** label. This value is reset to 0 at the beginning of the interval.	Count is incremented for each block of data read.
8-11	Bytes Written by the Channel	This 4 byte hexadecimal field contains the number of bytes transferred from the channel to this HBA port as part of a write operation. This is the value before the effect of the HBA compression. The value is reported in increments of 4KiB (4 x 1024). Any residual data will cause the value to be rounded up to the next higher value. In the diagram below, Bytes Written by the Channel is indicated by the **B** label. This value is reset to 0 at the beginning of the interval.	Count is incremented for each block of data written
12-15	Bytes Read from Disk Cache	This 4 byte hexadecimal field contains the number of bytes transferred from disk cache to this HBA port as part of a read operation. The value is for data previously compressed by the HBA or microcode. The value is reported in increments of 4KiB (4 x 1024). Any residual data will cause the value to be rounded up to the next higher value. In the diagram below, Bytes Read from Disk Cache is indicated by the **C** label. This value is reset to 0 at the beginning of the interval.	Count is incremented for each block of data read.

Bytes	Name	Description	When Data is Sampled/Updated
16-19	Bytes Written to Disk Cache	This 4 byte hexadecimal field contains the number of bytes transferred to disk cache from this HBA port as part of a write operation. The value is for data compressed by the HBA or microcode. The value is reported in increments of 4KiB (4 x 1024). Any residual data will cause the value to be rounded up to the next higher value. In the diagram below, Bytes Written to Disk Cache is indicated by the **D** label. This value is reset to 0 at the beginning of the interval.	Count is incremented for each block of data written.
20	Selective Resets	This 1 byte hexadecimal field indicates the number of selective resets this port received during the interval. This field is set to xFF when the number of selective resets is greater than 255 for the interval. This value is reset to 0 at the beginning of the interval.	Count is incremented for each selective reset.
21	System Resets	This 1 byte hexadecimal field indicates the number of system resets this port received during the interval. This field is set to xFF when the number of system resets is greater than 255 for the interval. This value is reset to 0 at the beginning of the interval.	Count is incremented for each system reset.
22-31	Reserved	All bytes set to x00.	



Hnode HSM Point-In-Time (PIT) Record

This Hnode Point-In-Time record is generated if the cluster is tape attached.
The record has the following nested structure:

- Header
- HSM Container
    - HSM-Library 0 Container
        - HSM-Library-Physical Device 0 Container
        - HSM-Library-Physical Device 1 Container
        - .
        - .
        - .
        - HSM-Library-Physical Device 31 Container
    - HSM-Library 1 Container (If installed)
        - Same sub-containers as HSM-Library 0 Container
    - HSM-Library 2 Container (If installed)
        - Same sub-containers as HSM-Library 0 Container
    - HSM-Library 3 Container (If installed)
        - Same sub-containers as HSM-Library 0 Container
- Extended HSM Container

Bytes	Name	Description	When Data is Sampled/Updated
0-1	Length	This 2 byte hexadecimal field contains the length of this record. The length includes these 2 bytes.	
2	Version	This 1 byte hexadecimal field contains the version of the data presented in this record. The current version is set to x05.	
3	Data Type	This 1 byte hexadecimal field indicates the type of data contained in this record. For this record the value is set to x10 indicating this is an Hnode HSM Point-In-Time record.	
4	Node ID	This 1 byte hexadecimal field indicates the Hnode ID which this interval's data represents. Valid values are x00 – x01.	
5	Cluster ID	This 1 byte hexadecimal field indicates the Cluster ID which this Hnode is a part of. Valid values are x00 – x07.	
6-7	Interval Duration	This 2 byte hexadecimal field indicates the interval in seconds that this interval's data was taken over.	
8-11	Time Stamp	This 4 byte hexadecimal field indicates the end time of the interval this data was taken over. This value is the time in seconds since the Epoch (00:00:00 UTC, January 1, 1970)	
12-15	Machine Type	This 4 byte EBCDIC field contains this node's machine type. The field is left justified padded with EBCDIC blanks. Initially this field will be set to "3957".	
16-18	Machine Model	This 3 byte EBCDIC field contains this node's machine model. The field is left justified padded with EBCDIC blanks. Initially this field will be set to "V06".	
19-26	Machine Serial Number	This 8 character EBCDIC field contains the serial number of this node. This field is left justified and padded with EBCDIC blanks. The format is XX-YYYYY where XX is the plant of manufacture and the YYYYY is the sequence number of the node's machine. The dash character (-) is fixed.	

Bytes	Name	Description	When Data is Sampled/Updated
27-34	Code Level	This 8 byte hexadecimal field contains the code level of the TS7700. The 8 bytes are actually four 2-byte fields. Each 2-byte field represents a portion of the code level. The VE code level is expressed as Version.Release.Modification.Fix in a decimal form. For example the code level of 8.0.0.104 would be represented in the 8 bytes as: x0008000000000068.	
35-39	Grid Library Sequence Number	This 5 character EBCDIC field contains the Library Sequence Number of the Grid (Composite) library.	
40-44	Distributed Library Sequence Number	This 5 character EBCDIC field contains the Distributed Library Sequence Number for this Distributed Library ID	
45-63	Reserved	All bytes set to x00	
HSM Container **Bytes from 64 and length (32 + N x 1568), where N is the number of physical libraries attached and can be obtained from Number of Physical Libraries field in this container.** **This container provides information concerning HSM related items. The total length depends upon the number of physical libraries attached to this Hnode.**			
64-65	Recalls in Queue	This two byte hexadecimal field contains the current number of queued recall operations at the end of the interval.	This count is updated as recalls added or removed from the queue.
66-67	Pre-migrates in Queue	This two byte hexadecimal field contains the current number of queued pre-migrate operations at the end of the interval.	This count is updated with the current value every 30 seconds.
68-71	Host Write Throttle on Cache Partition 0	This 4 byte hexadecimal field contains the host write throttling value on cache partition 0 over the interval. The value is reported in thousandths of a second. This is the value at the end of the interval.	This count is updated with the current value every 30 seconds.
72-75	Copy Throttle on Cache Partition 0	This 4 byte hexadecimal field contains the copy throttling value on cache partition 0 over the interval. The value is reported in thousandths of a second. This is the value at the end of the interval.	This count is updated with the current value every 30 seconds.
76	Number of Physical Libraries	This 1 byte hexadecimal field indicates the number of physical libraries this Hnode is attached to. This field can be used to determine how many HSM-Library containers will be attached to this record. This is the value at the end of the interval. **Note:** Only one physical library is supported at this time.	
77-80	Deferred Copy Throttle on Cache Partition 0	This 4 byte hexadecimal field contains the current copy throttling value on cache partition 0 over the interval. This is for throttling where the Cluster was prioritizing Host and Immediate Copies over sourcing Deferred Copies due to a constrained resource. The value is reported in thousandths of a second. This is the value at the end of the interval.	

<table>
<tr><th>Bytes</th><th>Name</th><th>Description</th><th>When Data is Sampled/Updated</th></tr>
<tr><td>81</td><td>CPU Usage percentage</td><td>This 1 byte hexadecimal field indicates TS7700 server CPU percentage at the end of the interval.
Prior to R3.0 this field reflected the Average CPU/Cache Utilization Percentage where the larger of the two is reported.</td><td>This field is updated every 30 seconds.</td></tr>
<tr><td>82</td><td>Maximum Disk Cache Usage percentage</td><td>This 1 byte hexadecimal field indicates the highest disk cache usage percentage of all the disks in the subsystem at the end of the interval.</td><td>This field is updated every 30 seconds.</td></tr>
<tr><td>83-84</td><td>Host Write Throttle Reason(s) on Cache Partition 0</td><td>This 2 byte hexadecimal field indicates the reason(s) for host write throttling on cache partition 0 during the interval.
<table>
<tr><th>Value</th><th>Description</th></tr>
<tr><td>x00</td><td>No throttling during the interval</td></tr>
<tr><td>x01</td><td>Premigration steady state (PMTHLVL)</td></tr>
<tr><td>x02</td><td>Low on cache free space</td></tr>
<tr><td>x04</td><td>Immediate copy throttling</td></tr>
<tr><td>x08</td><td>Excess cached content for copy</td></tr>
<tr><td>x10</td><td>Grid premigration steady state (throttling outbound copies because the target cluster is premigration throttling)</td></tr>
<tr><td></td><td>All other values are reserved</td></tr>
</table>
This value is reset to 0 at the beginning of the interval.</td><td>This field is updated every 30 seconds.</td></tr>
<tr><td>85-86</td><td>Copy Throttle Reason(s) on Cache Partition 0</td><td>This 2 byte hexadecimal field indicates the reason(s) for copy throttling on cache partition 0 during the interval.
<table>
<tr><th>Value</th><th>Description</th></tr>
<tr><td>x00</td><td>No throttling during the interval</td></tr>
<tr><td>x01</td><td>Premigration steady state (PMTHLVL)</td></tr>
<tr><td>x02</td><td>Low on cache free space</td></tr>
<tr><td></td><td>All other values are reserved</td></tr>
</table>
This value is reset to 0 at the beginning of the interval.</td><td>This field is updated every 30 seconds.</td></tr>
<tr><td>87-88</td><td>Deferred Copy Throttle Reason(s) on Cache Partition 0</td><td>This 2 byte hexadecimal field indicates the reason(s) for deferred copy throttling on cache partition 0 during the interval.
<table>
<tr><th>Value</th><th>Description</th></tr>
<tr><td>x00</td><td>No throttling during the interval</td></tr>
<tr><td>x03</td><td>Heavy host I/O or high resource utilization</td></tr>
<tr><td>x04</td><td>Immediate copy throttling</td></tr>
<tr><td></td><td>All other values are reserved</td></tr>
</table>
This value is reset to 0 at the beginning of the interval.</td><td>This field is updated every 30 seconds.</td></tr>
<tr><td>89-90</td><td>Reserved</td><td>All bytes set to x00</td><td></td></tr>
<tr><td>91-92</td><td>Offset to Extended HSM Container</td><td>This 2 byte hexadecimal field contains the offset to Extended HSM container. The offset is measured from the beginning of this container.</td><td></td></tr>
<tr><td>93-95</td><td>Reserved</td><td>All bytes set to x00</td><td></td></tr>
</table>

<table>
<tr><th>Bytes</th><th>Name</th><th>Description</th><th>When Data is Sampled/Updated</th></tr>
<tr><td colspan="3">

HSM-Library Container

Bytes from 96 and length (N x 1568), where N is the number of physical libraries attached and can be obtained from Number of Physical Libraries field in HSM container. For example, if there is just one physical library attached to this node there will be 1 set of data with 1568 bytes. There is a maximum of 4 physical libraries attached to a single Cluster.

This next segment of the record contains one set of data for each physical library attached to this Hnode as defined in byte 76 above. Each set of data contains 1568 bytes. The data for the first library is found in bytes 96-1663; the second library's data (if the library exists) is found in bytes 1664-3231, and so forth.

These fields contain information concerning the underlying automation.

</td><td></td></tr>
<tr><td>0-15</td><td>Library Sequence Number</td><td>This 16 byte EBCDIC field indicates the Library Sequence Number of the underlying automation. This field is left justified and padded with EBCDIC blanks. This is the value at the end of the interval.</td><td></td></tr>
<tr><td>16-31</td><td>Reserved</td><td>All bytes set to x00.</td><td></td></tr>
<tr><td colspan="3">

HSM-Library-Physical Device Container

Relative bytes 32-415 (32 sets of device data x 48 bytes/set = 1536 bytes)

For a system with just 1 class of physical device (homogeneous), the data for the devices is found in the first 16 sets of data. For a system with 2 classes of physical devices (heterogeneous), the data for one class of drives is found in the first 16 sets of data, and the data for the second class of devices is found in the second 16 sets of data.

There are 32 sets of data per possible library, one for each of 32 possible physical devices per physical library. The data for the first device of the first device class is in relative bytes 0-47; the second device's data of the first device class is in bytes 48-95, and so forth. For the second device class, the first device's data is found in relative bytes 768-815, the second device's data is in bytes 816-863, and so forth.

</td><td></td></tr>
<tr><td>0</td><td>Device Class ID</td><td>This one byte hexadecimal field contains this device's device class identifier.
<table>
<tr><th>Value</th><th>Description</th></tr>
<tr><td>x00</td><td>No device installed</td></tr>
<tr><td>x20</td><td>3592 Model J1A. This also includes a 3592-E05 that is emulating a 3592-J1A device.</td></tr>
<tr><td>x22</td><td>3592 Model E05. This is for a 3592-E05 that is behaving as a 3590-E05.</td></tr>
<tr><td>x23</td><td>3592 Model E05 (encryption configured)</td></tr>
<tr><td>x24</td><td>3592 Model E06</td></tr>
<tr><td>x25</td><td>3592 Model E07</td></tr>
<tr><td>x26</td><td>3592 Model E08</td></tr>
<tr><td></td><td>All other values are reserved.</td></tr>
</table>
This is the value at the end of the interval.</td><td></td></tr>
<tr><td>1-10</td><td>Physical Volume</td><td>This 10 byte EBCDIC field contains the volser of the volume that is loaded in this device at the end of the interval, if any. This field is left justified and padded with EBCDIC blanks. This field is filled with EBCDIC blanks when there is no volume in the device.</td><td>Volser is updated whenever there is a change.</td></tr>
<tr><td>11</td><td>Volume Pool</td><td>This 1 byte hexadecimal field indicates the pool associated with the volume, if any, loaded in the device at the end of the interval. This field is set to x00 when there isn't a volume loaded in the device. Values 1-32 are also valid in this field.</td><td>Pool is updated whenever there is a change.</td></tr>
</table>

<table>
<tr><th>Bytes</th><th>Name</th><th>Description</th><th>When Data is Sampled/Updated</th></tr>
<tr><td>12</td><td>Device State</td><td>This 1 byte hexadecimal field indicates the state of the device at the end of the interval. Possible values are:
<table>
<tr><th>Value</th><th>Description</th></tr>
<tr><td>x00</td><td>Device is online</td></tr>
<tr><td>x01</td><td>Device is offline</td></tr>
<tr><td></td><td>All other values are reserved.</td></tr>
</table></td><td>State is updated whenever there is a change.</td></tr>
<tr><td>13</td><td>Device Role</td><td>This 1 byte hexadecimal field contains the role of the device at the end of the interval. Possible values are:
<table>
<tr><th>Value</th><th>Description</th></tr>
<tr><td>x00</td><td>Idle</td></tr>
<tr><td>x01</td><td>Recalling</td></tr>
<tr><td>x02</td><td>Migrating</td></tr>
<tr><td>x03</td><td>Reclaim Source</td></tr>
<tr><td>x04</td><td>Importing</td></tr>
<tr><td>x05</td><td>Exporting</td></tr>
<tr><td>x06</td><td>Data Security Erase</td></tr>
<tr><td>x07</td><td>Reclaim target</td></tr>
<tr><td>x08</td><td>Rekey</td></tr>
<tr><td></td><td>All other values are reserved.</td></tr>
</table></td><td>Role is updated whenever there is a change.</td></tr>
<tr><td>14-23</td><td>Logical Volume</td><td>This 10 byte EBCDIC field contains the volser of the logical volume, if any, that is being processed by the physical device at the end of the interval. This field is left justified and padded with EBCDIC blanks. This field contains all EBCDIC blanks when there isn’t a logical volume being processed at the end of the interval.</td><td>Volser updated whenever there is a change.</td></tr>
<tr><td>24-27</td><td>Data Read From Device</td><td>This 4 byte hexadecimal field indicates the number of bytes transferred from the physical device during this interval. The value is reported in increments of 100KiB (100 x 1024). Any residual data will cause the value to be rounded up to the next higher value.
This value is reset to 0 at the beginning of the interval.</td><td>Count is incremented after every read.</td></tr>
<tr><td>28-31</td><td>Data Written to Device</td><td>This 4 byte hexadecimal field indicates the number of bytes transferred to the physical device during this interval. The value is reported in increments of 100KiB (100 x 1024). Any residual data will cause the value to be rounded up to the next higher value.
This value is reset to 0 at the beginning of the interval.</td><td>Count is incremented after each write.</td></tr>
<tr><td>32-43</td><td>Device Serial Number</td><td>This 12 byte ASCII field contains the serial number of the physical tape drive. This field is left justified and blank filled.</td><td></td></tr>
</table>

<table>
<tr><th>Bytes</th><th>Name</th><th>Description</th><th>When Data is Sampled/Updated</th></tr>
<tr><td>44</td><td>Media Format</td><td>This one byte hexadecimal field contains the current media format.
<table>
<tr><th>Value</th><th>Description</th></tr>
<tr><td>x00</td><td>No device installed</td></tr>
<tr><td>x20</td><td>3592 Model J1A. This also includes a 3592-E05 that is emulating a 3592-J1A device.</td></tr>
<tr><td>x22</td><td>3592 Model E05. This is for a 3592-E05 that is behaving as a 3590-E05</td></tr>
<tr><td>x23</td><td>3592 Model E05 encrypted</td></tr>
<tr><td>x24</td><td>3592 Model E06.</td></tr>
<tr><td>x25</td><td>3592 Model E06 encrypted</td></tr>
<tr><td>x26</td><td>3592 Model E07</td></tr>
<tr><td>x27</td><td>3592 Model E07 encrypted</td></tr>
<tr><td>x28</td><td>3592 Model E08</td></tr>
<tr><td>x29</td><td>3592 Model E08 encrypted</td></tr>
<tr><td></td><td>All other values are reserved.</td></tr>
</table></td><td></td></tr>
<tr><td>45-47</td><td>Reserved</td><td>All bytes set to x00.</td><td></td></tr>
<tr><td colspan="3"><h2>Extended HSM Container</h2>
<b>Bytes from (64 + N) and length 32 bytes, where N is the offset to this Extended HSM container measured from the beginning of HSM container and can be obtained from Offset to Extended HSM container field in HSM container.</b>

<b>This container provides additional information concerning HSM related items.</b></td><td></td></tr>
<tr><td>0-1</td><td>Length</td><td>This 2 byte hexadecimal field contains the length of this container. The length includes these 2 bytes.</td><td></td></tr>
<tr><td>2-5</td><td>Host Write Throttle on Tape Attached Cache Partitions</td><td>This 4 byte hexadecimal field contains the host write throttling value on tape attached cache partitions over the interval. The value is reported in thousandths of a second.

This is the value at the end of the interval.</td><td>This count is updated with the current value every 30 seconds.</td></tr>
<tr><td>6-9</td><td>Copy Throttle on Tape Attached Cache Partitions</td><td>This 4 byte hexadecimal field contains the copy throttling value on tape attached cache partitions over the interval. The value is reported in thousandths of a second.

This is the value at the end of the interval.</td><td>This count is updated with the current value every 30 seconds.</td></tr>
<tr><td>10-13</td><td>Deferred Copy Throttle on Tape Attached Cache Partitions</td><td>This 4 byte hexadecimal field contains the current copy throttling value on tape attached cache partitions over the interval. This is for throttling where the Cluster was prioritizing Host and Immediate Copies over sourcing Deferred Copies due to constrained resources. The value is reported in thousandths of a second.

This is the value at the end of the interval.</td><td></td></tr>
</table>

<table>
<tr><th>Bytes</th><th>Name</th><th>Description</th><th>When Data is Sampled/Updated</th></tr>
<tr><td>14-15</td><td>Host Write Throttle Reason(s) on Tape Attached Cache Partitions</td><td>This 2 byte hexadecimal field indicates the reason(s) for host write throttling on tape attached cache partitions during the interval.
<table>
<tr><th>Value</th><th>Description</th></tr>
<tr><td>x00</td><td>No throttling during the interval</td></tr>
<tr><td>x01</td><td>Premigration steady state (PMTHLVL)</td></tr>
<tr><td>x02</td><td>Low on cache free space</td></tr>
<tr><td>x04</td><td>Immediate copy throttling</td></tr>
<tr><td>x08</td><td>Excess cached content for copy</td></tr>
<tr><td>x10</td><td>Grid premigration steady state (throttling outbound copies because the target cluster is premigration throttling)</td></tr>
<tr><td></td><td>All other values are reserved</td></tr>
</table>
This value is reset to 0 at the beginning of the interval.</td><td>This field is updated every 30 seconds.</td></tr>
<tr><td>16-17</td><td>Copy Write Throttle Reason(s) on Tape Attached Cache Partitions</td><td>This 2 byte hexadecimal field indicates the reason(s) for copy throttling on tape attached cache partitions during the interval.
<table>
<tr><th>Value</th><th>Description</th></tr>
<tr><td>x00</td><td>No throttling during the interval</td></tr>
<tr><td>x01</td><td>Premigration steady state (PMTHLVL)</td></tr>
<tr><td>x02</td><td>Low on cache free space</td></tr>
<tr><td></td><td>All other values are reserved</td></tr>
</table>
This value is reset to 0 at the beginning of the interval.</td><td>This field is updated every 30 seconds.</td></tr>
<tr><td>18-19</td><td>Deferred Copy Write Throttle Reason(s) on Tape Attached Cache Partitions</td><td>This 2 byte hexadecimal field indicates the reason(s) for deferred copy throttling on tape attached cache partitions during the interval.
<table>
<tr><th>Value</th><th>Description</th></tr>
<tr><td>x00</td><td>No throttling during the interval</td></tr>
<tr><td>x01</td><td>Preferring Host I/O</td></tr>
<tr><td>x02</td><td>Cluster resource usage</td></tr>
<tr><td>x04</td><td>Immediate copy throttling</td></tr>
<tr><td></td><td>All other values are reserved</td></tr>
</table>
This value is reset to 0 at the beginning of the interval.</td><td>This field is updated every 30 seconds.</td></tr>
<tr><td>20-31</td><td>Reserved</td><td>All bytes set to x00</td><td></td></tr>
</table>

Hnode Grid Point-In-Time (PIT) Record

This Hnode Point-In-Time record has the following nested structure:

- Header
- Grid Container
  - Grid-Cluster 0 Container
  - Grid-Cluster 1 Container (If installed)
  - Grid-Cluster 2 Container (If installed)
  - Grid-Cluster 3 Container (If installed)
  - Grid-Cluster 4 Container (If installed)
  - Grid-Cluster 5 Container (If installed)
  - Grid-Cluster 6 Container (If installed)
  - Grid-Cluster 7 Container (If installed)

Bytes	Name	Description	When Data is Sampled/Updated
0-1	Length	This 2 byte hexadecimal field contains the length of this record. The length includes these 2 bytes.	
2	Version	This 1 byte hexadecimal field contains the version of the data presented in this record. The current version is set to x03.	
3	Data Type	This 1 byte hexadecimal field indicates the type of data contained in this record. For this record the value is set to x11 indicating this is an Hnode Grid Point-In-Time record.	
4	Node ID	This 1 byte hexadecimal field indicates the Hnode ID which this interval's data represents. Valid values are x00 – x01.	
5	Cluster ID	This 1 byte hexadecimal field indicates the Cluster ID which this Hnode is a part of. Valid values are x00 – x07.	
6-7	Interval Duration	This 2 byte hexadecimal field indicates the interval in seconds that this interval's data was taken over.	
8-11	Time Stamp	This 4 byte hexadecimal field indicates the end time of the interval this data was taken over. This value is the time in seconds since the Epoch (00:00:00 UTC, January 1, 1970)	
12-15	Machine Type	This 4 byte EBCDIC field contains this node's machine type. The field is left justified padded with EBCDIC blanks. Initially this field will be set to "3957".	
16-18	Machine Model	This 3 byte EBCDIC field contains this node's machine model. The field is left justified padded with EBCDIC blanks. Initially this field will be set to "V06".	
19-26	Machine Serial Number	This 8 character EBCDIC field contains the serial number of this node. This field is left justified and padded with EBCDIC blanks. The format is XX-YYYYY where XX is the plant of manufacture and the YYYYY is the sequence number of the node's machine. The dash character (-) is fixed.	
27-34	Code Level	This 8 byte hexadecimal field contains the code level of the TS7700. The 8 bytes are actually four 2-byte fields. Each 2-byte field represents a portion of the code level. The VE code level is expressed as Version.Release.Modification.Fix in a decimal form. For example the code level of 8.0.0.104 would be represented in the 8 bytes as: x0008000000000068.	
35-39	Grid Library Sequence Number	This 5 character EBCDIC field contains the Library Sequence Number of the Grid (Composite) library.	

Bytes	Name	Description	When Data is Sampled/Updated
40-44	Distributed Library Sequence Number	This 5 character EBCDIC field contains the Distributed Library Sequence Number for this Distributed Library ID	
45-63	Reserved	All bytes set to x00	

Grid Container

Bytes 64 and up

This container provides information concerning the Grid aspects of this Cluster. The length of the message depends on the number of Clusters in the Grid. There are 32 fixed bytes plus 32 bytes for each Cluster in the Grid.

64-67	Immediate Copy Queue	This 4 byte hexadecimal field indicates the number of logical volumes in the immediate copy queue targeted for this Cluster at the end of the interval.	Count is updated every 5 minutes.
68-71	Deferred Copy Queue	This 4 byte hexadecimal field indicates the number of logical volumes in the deferred copy queue targeted for this Cluster at the end of the interval.	Count is updated every 5 minutes.
72-75	Active Copies	This 4 byte hexadecimal field indicates the number of active copies that are targeting this Cluster at the end of the interval.	Count is updated every 5 minutes.
76-78	Reserved	Reserved for future use.	
79	Number of Clusters	This 1 byte hexadecimal field indicates the number of Clusters in the Grid. This field can be used to determine how many Grid-Cluster containers will be attached to this record. There is a maximum of 8 Clusters. This is the value at the end of the interval.	
80-83	Average time delayed copy queue age	This 4 byte hexadecimal field indicates the average age that copies in the timed delay state are in the copy queue. Logical volumes in the timed delay state are not yet eligible for the actual copy until their defined time-delays are expired.	
84-87	Overall Object Data Transferred into Cache from DS8Ks	This 4 byte field contains the total number of bytes transferred to this Cluster's cache from all of the DS8K connected to this Cluster as part of DS8K offload function during this interval. The value is reported in increments of 100KiB (100 x 1024). Any residual data will cause the value to be rounded up to the next higher value. This value is reset to 0 at the beginning of the interval.	
88-91	Overall Object Data Transferred from Cache to DS8Ks	This 4 byte field contains the total number of bytes transferred from this Cluster's cache to all of the DS8K connected to this Cluster as part of DS8K offload function during this interval. The value is reported in increments of 100KiB (100 x 1024). Any residual data will cause the value to be rounded up to the next higher value. This value is reset to 0 at the beginning of the interval.	
92-95	Reserved	All bytes set to x00.	

<table>
<tr><th>Bytes</th><th>Name</th><th>Description</th><th>When Data is Sampled/Updated</th></tr>
<tr><td colspan="3"><h2>Grid-Cluster Container</h2><p><b>Bytes 96 and up (Number of Clusters x 128 bytes/set) For example, if there are 3 Clusters in the Grid there will be 3 sets of data. There is a maximum of 8 Clusters in a Grid.</b></p><p><b>This next segment of the record contains one set of data for each Cluster in the Grid as defined in byte 79 above. Each set of data contains 128 bytes. The data for the first Cluster (Cluster 0) can be found in bytes 96-223; the second Cluster's (Cluster 1) data can be found in bytes 224-351, and so forth.</b></p></td><td></td></tr>
<tr><td>0</td><td>Cluster Link State</td><td><p>This 1 byte hexadecimal field indicates the state of the link between this Cluster and the others in the Grid at the end of the interval.</p>
<table>
<tr><th>Value</th><th>Description</th></tr>
<tr><td>x00</td><td>All links are fully operational. There are no detected error conditions between the Clusters in the Grid. For a Grid with only a single Cluster, this is the value always reported in this field.</td></tr>
<tr><td>X01</td><td>Degraded. One or more of the logical communication paths between the Clusters is not operational.</td></tr>
<tr><td>X02</td><td>Failed. All of the logical communication paths between this Cluster and the other Clusters in the Grid are not working.</td></tr>
<tr><td></td><td>All other values are reserved.</td></tr>
</table>
<p>This field represents the overall link state as determined by the Cluster. The actual connection may be comprised of several physical connections and underlying infrastructure which is transparent to the system.</p></td><td>Updated whenever the link state changes.</td></tr>
<tr><td>1-4</td><td>Data Transferred into a Cluster's Cache from other Clusters</td><td><p>This 4 byte field contains the number of bytes transferred to this Cluster's cache from other Clusters as part of a remote file access during this interval.</p><p>The value is reported in increments of 100KiB (100 x 1024). Any residual data will cause the value to be rounded up to the next higher value.</p><p>This value is reset to 0 at the beginning of the interval.</p></td><td>Count is incremented every time a block of data is written to this Cluster.</td></tr>
<tr><td>5-32</td><td>Reserved</td><td>All bytes set to x00.</td><td></td></tr>
<tr><td>33-36</td><td>Data Transferred From a Cluster's Cache To Other Clusters</td><td><p>This 4 byte field contains the number of bytes transferred from this Cluster's cache to other Clusters as part of a copy and remote file access during this interval.</p><p>The value is reported in increments of 100KiB (100 x 1024). Any residual data will cause the value to be rounded up to the next higher value.</p><p>This value is reset to 0 at the beginning of the interval.</p></td><td>Count is incremented every time a block of data is read from this cluster</td></tr>
<tr><td>37-127</td><td>Reserved</td><td>All bytes set to x00.</td><td></td></tr>
</table>

Hnode HSM Historical Record

This Hnode historical record has the following nested structure:

- Header
- HSM Container
    - HSM-Disaster Recovery Container
    - HSM-Cache Container
        - HSM-Cache-Partition 0 Container
            - HSM-Cache-Partition-Preference Group 0 Container
            - HSM-Cache-Partition-Preference Group 1 Container
        - HSM-Cache-Partition 1 Container
            - HSM-Cache-Partition-Preference Group 0 Container
            - HSM-Cache-Partition-Preference Group 1 Container
        - HSM-Cache-Partition 2 Container
            - HSM-Cache-Partition-Preference Group 0 Container
            - HSM-Cache-Partition-Preference Group 1 Container
        - HSM-Cache-Partition 3 Container
            - HSM-Cache-Partition-Preference Group 0 Container
            - HSM-Cache-Partition-Preference Group 1 Container
        - HSM-Cache-Partition 4 Container
            - HSM-Cache-Partition-Preference Group 0 Container
            - HSM-Cache-Partition-Preference Group 1 Container
        - HSM-Cache-Partition 5 Container
            - HSM-Cache-Partition-Preference Group 0 Container
            - HSM-Cache-Partition-Preference Group 1 Container
        - HSM-Cache-Partition 6 Container
            - HSM-Cache-Partition-Preference Group 0 Container
            - HSM-Cache-Partition-Preference Group 1 Container
        - HSM-Cache-Partition 7 Container
            - HSM-Cache-Partition-Preference Group 0 Container
            - HSM-Cache-Partition-Preference Group 1 Container
    - Extended HSM-Cache Container
        - Extended HSM-Cache-Partition 0 Container
            - Extended HSM-Cache-Partition-Preference Group 0 Container
            - Extended HSM-Cache-Partition-Preference Group 1 Container
        - Extended HSM-Cache-Partition 1 Container
            - Extended HSM-Cache-Partition-Preference Group 0 Container
            - Extended HSM-Cache-Partition-Preference Group 1 Container
        - Extended HSM-Cache-Partition 2 Container
            - Extended HSM-Cache-Partition-Preference Group 0 Container
            - Extended HSM-Cache-Partition-Preference Group 1 Container
        - Extended HSM-Cache-Partition 3 Container
            - Extended HSM-Cache-Partition-Preference Group 0 Container
            - Extended HSM-Cache-Partition-Preference Group 1 Container
        - Extended HSM-Cache-Partition 4 Container
            - Extended HSM-Cache-Partition-Preference Group 0 Container
            - Extended HSM-Cache-Partition-Preference Group 1 Container
        - Extended HSM-Cache-Partition 5 Container
            - Extended HSM-Cache-Partition-Preference Group 0 Container
            - Extended HSM-Cache-Partition-Preference Group 1 Container
        - Extended HSM-Cache-Partition 6 Container

    - Extended HSM-Cache-Partition-Preference Group 0 Container
    - Extended HSM-Cache-Partition-Preference Group 1 Container
  - Extended HSM-Cache-Partition 7 Container
    - Extended HSM-Cache-Partition-Preference Group 0 Container
    - Extended HSM-Cache-Partition-Preference Group 1 Container
- Compression Container
  - Compression Method 0 Container
  - Compression Method 1 Container
  - …

Bytes	Name	Description	When Data is Sampled/Updated
0-1	Length	This 2 byte hexadecimal field contains the length of this record. The length includes these 2 bytes.	
2	Version	This 1 byte hexadecimal field contains the version of the data presented in this record. The current version is set to x08.	
3	Data Type	This 1 byte hexadecimal field indicates the type of data contained in this record. For this record the value is set to x30 indicating this is an Hnode HSM Historical record.	
4	Node ID	This 1 byte hexadecimal field indicates the Hnode ID which this interval's data represents. Valid values are x00 – x01.	
5	Cluster ID	This 1 byte hexadecimal field indicates the Cluster ID which this Hnode is a part of. Valid values are x00 – x07.	
6-7	Interval Duration	This 2 byte hexadecimal field indicates the interval in seconds that this interval's data was taken over.	
8-11	Time Stamp	This 4 byte hexadecimal field indicates the end time of the interval this data was taken over. This value is the time in seconds since the Epoch (00:00:00 UTC, January 1, 1970)	
12-15	Machine Type	This 4 byte EBCDIC field contains this node's machine type. The field is left justified padded with EBCDIC blanks. Initially this field will be set to "3957".	
16-18	Machine Model	This 3 byte EBCDIC field contains this node's machine model. The field is left justified padded with EBCDIC blanks. Initially this field will be set to "V06".	
19-26	Machine Serial Number	This 8 character EBCDIC field contains the serial number of this node. This field is left justified and padded with EBCDIC blanks. The format is XX-YYYYY where XX is the plant of manufacture and the YYYYY is the sequence number of the node's machine. The dash character (-) is fixed.	
27-34	Code Level	This 8 byte hexadecimal field contains the code level of the TS7700. The 8 bytes are actually four 2-byte fields. Each 2-byte field represents a portion of the code level. The VE code level is expressed as Version.Release.Modification.Fix in a decimal form. For example the code level of 8.0.0.104 would be represented in the 8 bytes as: x0008000000000068.	
35-39	Grid Library Sequence Number	This 5 character EBCDIC field contains the Library Sequence Number of the Grid (Composite) library.	
40-44	Distributed Library Sequence Number	This 5 character EBCDIC field contains the Distributed Library Sequence Number for this Distributed Library ID	
45-63	Reserved	All bytes set to x00	

<table>
<tr><th>Bytes</th><th>Name</th><th>Description</th><th>When Data is Sampled/Updated</th></tr>
<tr><td colspan="3"><h2>HSM Container</h2>
<b>This record contains information pertinent to the Hydra Storage Manager.</b></td><td></td></tr>
<tr><td colspan="3"><h2>HSM – Disaster Recovery Container</h2>
<b>Bytes 64 – 95</b>
<b>This set of 32 bytes contains information concerning HSM Disaster Recovery (DR).</b></td><td></td></tr>
<tr><td>64-73</td><td>Disaster Recovery Volser</td><td>This 10 byte EBCDIC field contains the volser of the physical volume that contains the latest subsystem backup information. This field is set to all EBCDIC blanks if a disaster recovery volume does not exist. This field is left justified and padded with EBCDIC blanks. This is the value at the end of the interval.
For the TS7700 models without backend tape, this field will be EBCDIC blanks</td><td>Volser is updated whenever it changes.</td></tr>
<tr><td>74-95</td><td>Reserved</td><td>All bytes set to x00.</td><td></td></tr>
<tr><td colspan="3"><h2>HSM – Cache Container</h2>
<b>Bytes 96 – 1215</b>
<b>This set of bytes contains information concerning the Tape Volume Cache (TVC) and this Hnode.</b></td><td></td></tr>
<tr><td>96</td><td>Active Hnode</td><td>This 1 byte hexadecimal field indicates if this Hnode was the active node in charge of Tape Volume Cache (TVC) management at the end of the interval being reported.
<table>
<tr><th>Value</th><th>Description</th></tr>
<tr><td>x00</td><td>This is not the active Hnode</td></tr>
<tr><td>x01</td><td>This is the active Hnode</td></tr>
<tr><td></td><td>All other values are reserved</td></tr>
</table></td><td>This value is set to x00 for the first release.</td></tr>
<tr><td>97-100</td><td>TVC Size (32 bit)</td><td>This 4 byte hexadecimal field indicates the current size of the Cluster’s TVC in increments of 1000MiB (1000 x 1024 x 1024).
This field is filled in by the Hnode that is currently in charge of TVC management. The Hnode that isn’t in charge of the TVC management reports x00000000 in this field. Refer to the Active Hnode field.
This is the value at the end of the interval.
This field is deprecated and remains just for the backward compatibility. It still stores the value, but it shows xFFFFFFFF when the value hit the limit of the 32 bit maximum size. Refer to the new 8 byte field in the Extended HSM-Cache container to be ready for the cache larger than 4 PiB.</td><td></td></tr>
</table>

Bytes	Name	Description	When Data is Sampled/Updated
101	Percent Host Write Throttle on Cache Partition 0	This 1 byte hexadecimal field indicates the percentage of 30-second periods where there was at least one throttling value greater than zero on cache partition 0. This field is filled in by the Hnode that is currently in charge of TVC management. The Hnode that isn't in charge of the TVC management reports x00 in this field. Refer to the Active Hnode field. This value is recomputed at the end of the interval based on the data from the interval.	The current throttle value is sampled every 30 seconds. The percentage is computed over the interval.
102-105	Average Host Write Throttle on Cache Partition 0	This 4 byte hexadecimal field indicates the average host write throttle value on cache partition 0 during the interval. The value presented is the average of the non-zero throttling values. The value is reported in thousandths of a second. This field is filled in by the Hnode that is currently in charge of TVC management. The Hnode that isn't in charge of the TVC management reports x00000000 in this field. Refer to the Active Hnode field. This value is recomputed at the end of the interval based on the data from the interval.	The current throttle value is sampled every 30 seconds. The average is computed over the interval.
106	Percent Copy Throttle on Cache Partition 0	This 1 byte hexadecimal field indicates the percentage of 30-second periods where there was at least one throttling value greater than zero on cache partition 0 and that copy was the predominant reason for throttling. This field is filled in by the Hnode that is currently in charge of TVC management. The Hnode that isn't in charge of the TVC management reports x00 in this field. Refer to the Active Hnode field. This value is recomputed at the end of the interval based on the data from the interval.	The current throttle value is sampled every 30 seconds. The percentage is computed over the interval.
107-110	Average Copy Throttle on Cache Partition 0	This 4 byte hexadecimal field indicates the average copy throttle value on cache partition 0 during the interval. The value presented is the average of the non-zero throttling values where copy was the predominant reason for throttling. The value is reported in thousandths of a second. This field is filled in by the Hnode that is currently in charge of TVC management. The Hnode that isn't in charge of the TVC management reports x00000000 in this field. Refer to the Active Hnode field. This value is recomputed at the end of the interval based on the data from the interval.	The current throttle value is sampled every 30 seconds. The average is computed over the interval.

Bytes	**Name**	**Description**	**When Data is Sampled/Updated**
111-114	Average Overall Throttle on Cache Partition 0	This 4 byte hexadecimal field indicates the average of all throttling values on cache partition 0 during the interval. The calculation includes samples for periods where throttling was both zero and non-zero. The value is reported in thousandths of a second. This field is filled in by the Hnode that is currently in charge of TVC management. The Hnode that isn't in charge of the TVC management reports x00000000 in this field. Refer to the Active Hnode field. This value is recomputed at the end of the interval based on the data from the interval.	The current throttle values are sampled every 30 seconds. The average is computed over the interval.
115	Percent Deferred Copy Throttle on Cache Partition 0	This 1 byte hexadecimal field indicates the percentage of 30-second periods where deferred copy throttle from cache partition 0 of this Hnode was active (non-zero). This field is filled in by the Hnode that is currently in charge of TVC management. The Hnode that isn't in charge of the TVC management reports x00 in this field. Refer to the Active Hnode field. This value is recomputed at the end of the interval based on the data from the interval.	
116-119	Average Deferred Copy Throttle on Cache Partition 0	This 4 byte hexadecimal field indicates the average deferred copy throttle value on cache partition 0 during the interval. The value presented is the average of 30-second intervals of the deferred copy throttle value over the historical record interval. The value is reported in thousandths of a second. This value is recomputed at the end of the interval based on the data from the interval.	
120-123	Base Deferred Copy Throttle	This 4 byte hexadecimal field indicates the base deferred copy throttle value which would be used if deferred copy throttle were to be applied. The value is reported in thousandths of a second.	
124-127	Pre-migration Throttle Threshold	This 4 byte hexadecimal field indicates the current threshold of the pre-migration throttle in increments of 1000MiB (1000 x 1024 x 1024). This field represents amount of un-premigrated data in cache, at which the system will begin throttling the host I/O in order to keep the value below this level. This field is filled in by the Hnode that is currently in charge of TVC management. The Hnode that isn't in charge of the TVC management reports x00000000 in this field. Refer to the Active Hnode field. This is the value at the end of the interval.	
128	Average CPU Usage percentage	This 1 byte hexadecimal field indicates the CPU usage percentage at the end of the interval. This value can be used to indicate how busy the system was during the interval. This value is recomputed at the end of the interval based on the data from the interval.	The CPU usage is sampled every 30 seconds.

<table>
<tr><th>Bytes</th><th>Name</th><th>Description</th><th>When Data is Sampled/Updated</th></tr>
<tr><td>129</td><td>Maximum CPU Usage Percentage</td><td>This 1 byte hexadecimal field indicates the maximum value of the CPU usage percentage during the interval.

This value is recomputed at the end of the interval based on the data from the interval.</td><td>The CPU usage is sampled every 30 seconds.</td></tr>
<tr><td>130</td><td>Average Maximum Disk Usage Percentage</td><td>This 1 byte hexadecimal field indicates the average value of the highest disk usage percentage at the end of the interval. For each disk in the system, the disk usage percentages are compared to determine the highest value. The highest values are then averaged over the interval.

This value is recomputed at the end of the interval based on the data from the interval.</td><td>The disk usage is sampled every 30 seconds.</td></tr>
<tr><td>131</td><td>Maximum Disk Usage Percentage</td><td>This 1 byte hexadecimal field indicates the maximum value of the disk usage percentage during the interval.

This value is recomputed at the end of the interval based on the data from the interval.</td><td>The disk usage is sampled every 30 seconds.</td></tr>
<tr><td>132-133</td><td>Host Write Throttle Reason(s) on Cache Partition 0</td><td>This 2 byte hexadecimal field indicates the reason(s) for write overrun (host) throttling on cache partition 0 during the interval.
<table>
<tr><th>Value</th><th>Description</th></tr>
<tr><td>x00</td><td>No throttling during the interval</td></tr>
<tr><td>x01</td><td>Premigration steady state (PMTHLVL)</td></tr>
<tr><td>x02</td><td>Low on cache free space</td></tr>
<tr><td>x04</td><td>Immediate copy throttling</td></tr>
<tr><td>x08</td><td>Excess cached content for copy</td></tr>
<tr><td>x10</td><td>Grid premigration steady state (throttling outbound copies because the target cluster is premigration throttling)</td></tr>
<tr><td></td><td>All other values are reserved</td></tr>
</table></td><td>This value is updated every 30 seconds.</td></tr>
<tr><td>134-135</td><td>Copy Throttle Reason(s) on Cache Partition 0</td><td>This 2 byte hexadecimal field indicates the reason(s) for copy throttling on cache partition 0 during the interval.
<table>
<tr><th>Value</th><th>Description</th></tr>
<tr><td>x00</td><td>No throttling during the interval</td></tr>
<tr><td>x01</td><td>Premigration steady state (PMTHLVL)</td></tr>
<tr><td>x02</td><td>Low on cache free space</td></tr>
<tr><td></td><td>All other values are reserved</td></tr>
</table></td><td>This value is updated every 30 seconds.</td></tr>
<tr><td>136-137</td><td>Deferred Copy Throttle Reasons on Cache Partition 0</td><td>This 2 byte hexadecimal field indicates the reasons for deferred copy throttling on cache partition 0 during the interval.
<table>
<tr><th>Value</th><th>Description</th></tr>
<tr><td>x00</td><td>No throttling during the interval</td></tr>
<tr><td>x03</td><td>Heavy host I/O or high resource utilization</td></tr>
<tr><td>x04</td><td>Immediate copy throttling</td></tr>
<tr><td></td><td>All other values are reserved</td></tr>
</table></td><td>This value is updated every 30 seconds.</td></tr>
<tr><td>138-147</td><td>Reserved</td><td>All bytes set to x00.</td><td></td></tr>
<tr><td>148-151</td><td>Total used cache (32 bit)</td><td>This 4 byte hexadecimal field indicates the amount of cache used in the system. This value is in gigabytes (1000 x 1000 x 1000).

This field is deprecated and remains just for the backward compatibility. It still stores the value, but it shows xFFFFFFFF when the value hit the limit of the 32 bit maximum size. Refer to the new 8 byte field in the Extended HSM-Cache container to be ready for the cache larger than 4 PiB.</td><td></td></tr>
</table>

Bytes	**Name**	**Description**	**When Data is Sampled/Updated**
152-159	Reserved	All bytes set to x00.	
160-163	Total used flash cache (32 bit)	This 4 byte hexadecimal field indicates the amount of flash copy cache used in the system. This value is in gigabytes (1000 x 1000 x 1000). This field is deprecated and remains just for the backward compatibility. It still stores the value, but it shows xFFFFFFFF when the value hit the limit of the 32 bit maximum size. Refer to the new 8 byte field in the Extended HSM-Cache container to be ready for the cache larger than 4 PiB.	
164-167	Temporary Pre-migration Throttle Threshold	This 4 byte hexadecimal field indicates the current temporary threshold of the pre-migration throttle in increments of 1000MiB (1000 x 1024 x 1024). TS7700 will begin throttling the host I/O when the amount of un-premigrated data reach the threshold in order to keep the amount below the level. The value is set to 0 by default. Setting PRETHDEG configuration to DISABLE is available with Library Request command to disable Pre-migration Throttle when Physical Tape Library or the Cloud enters degraded state. This temporary threshold is set when degraded Physical Tape Library or the cloud gets back to operational and PRETHDEG is DISABLE. The value varies depending on conditions as shown below. Suppose PMTH_TMP is the Temporary Pre-migration Throttle Threshold value, PMTHLVL is the Pre-migration Throttle Threshold value, PMPRIOR is the Pre-migration Priority Threshold value, Q_OP is the amount of un-premigrated data when the Library became operational, and Q_NOW is the current amount of un-premigrated data. ● If Q_OP <= Q_NOW, then PMTH_TMP = PMTHLVL + Q_OP. ● If PMPRIOR <= Q_NOW < Q_OP, then PMTH_TMP = PMTHLVL + Q_NOW. ● If Q_NOW < PMPRIOR, then PMTH_TMP = 0. (Temporary Pre-migration Throttle Threshold is disabled) This is the value at the end of the interval.	

Bytes	Name	Description	When Data is Sampled/Updated
168-171	Temporary Pre-migration Priority Threshold	This 4 byte hexadecimal field indicates the current temporary threshold of the pre-migration task prioritization in increments of 1000MiB (1000 x 1024 x 1024). TS7700 will begin increasing the number of pre-migration tasks, which would compete with host I/O for cache and processor resources, when the amount of un-premigrated data reach the threshold. The value is set to 0 by default. Setting PRETHDEG configuration to DISABLE is available with Library Request command to disable Pre-migration Throttle when Physical Tape Library or the Cloud enters degraded state. Additionally, setting PMPRIOR configuration to ENABLE is available with Library Request command to enable to slide Pre-migration Priority Threshold to Temporary Pre-migration Priority Threshold when Physical Tape Library or the Cloud gets back to operational state. This temporary threshold is set when degraded Physical Tape Library gets back to operational, PRETHDEG is DISABLE, and SLDPMPRI is ENABLE. The value varies depending on conditions as shown below. Suppose PMPRIOR_TMP is the Temporary Pre-migration Priority Threshold value, PMPRIOR is the Pre-migration Priority Threshold value, Q_OP is the amount of un-premigrated data when the Library became operational, and Q_NOW is the current amount of un-premigrated data. ● If Q_OP <= Q_NOW, then PMPRIOR_TMP = PMPRIOR + Q_OP. ● If PMPRIOR <= Q_NOW and Q_NOW < Q_OP, then PMPRIOR_TMP = PMPRIOR + Q_NOW. ● If Q_NOW < PMPRIOR, then PMPRIOR_TMP = 0. (Temporary Pre-migration Priority Threshold is disabled) This is the value at the end of the interval.	
172-191	Reserved	All bytes set to x00.	

HSM – Cache – Partition Container

Bytes 192 – 1215 (8 x 128 bytes = 1024 bytes)

This next set of bytes contains information for up to 8 cache partitions for the Cluster. Each set of data contains 128 bytes. The following fields define the 128 bytes and are numbered starting with byte 0. The first cache partition's data can be found in relative bytes 0-127 and the second cache partition's data can be found in relative bytes 128-255, and so forth.

Bytes	Name	Description	When Data is Sampled/Updated
0-3	Partition Size (32 bit)	This 4 byte hexadecimal field indicates the amount of cache assigned to this partition. The value is reported in increments of 1000MiB (1000 x 1024 x 1024). This field is filled in by the Hnode that is currently in charge of TVC management. The Hnode that isn't in charge of the TVC management reports x00000000 in this field. Refer to the Active Hnode field. This is the value at the end of the interval. This field is deprecated and remains just for the backward compatibility. It still stores the value, but it shows xFFFFFFFF when the value hit the limit of the 32 bit maximum size. Refer to the new 8 byte field in the Extended HSM-Cache-Partition container to be ready for the cache larger than 4 PiB.	The size is updated when it changes.
4-5	Fast Ready Mounts	This 2 byte hexadecimal field indicates the number of mount requests completed using the Fast Ready method during this interval. A mount is accredited to the interval when the mount is completed. This value is reset to 0 at the beginning of the interval.	The count is incremented when the mount-complete message is received.
6-9	Average Fast Ready Mount Time	This 4 byte hexadecimal field indicates the average time, in milliseconds, taken to complete Fast-Ready mounts during the interval. Mount time is accrued from the time the mount request is accepted by the system until the mount is completed. The mount time is averaged into the interval's time when the mount is completed. This value is reset to 0 at the beginning of the interval.	The time is incremented for each mount and averaged at the end of the interval.
10-11	Cache Hit Mounts	This 2 byte hexadecimal field indicates the number of mount requests completed that the data was resident in the Tape Volume Cache (TVC) during this interval. A mount is accredited to the interval when the mount is completed. If two cluster access points were used to satisfy a mount with sync mode copy enabled, then this count is incremented only when the data was resident in the TVC of both access points. This value is reset to 0 at the beginning of the interval.	The count is incremented when the mount-complete message is received.
12-15	Average Cache Hit Mount Time	This 4 byte hexadecimal field indicates the average time, in milliseconds, taken to complete Cache Hit mounts during the interval. Mount time is accrued from the time the mount request is accepted by the system until the mount is completed. The mount time is averaged into the interval's time when the mount is completed. This value is reset to 0 at the beginning of the interval.	The time is incremented for each mount and averaged at the end of the interval.
16-17	Cache Miss Mounts	This 2 byte hexadecimal field indicates the number of mount requests completed that required recall from a stacked volume during this interval. A mount is accredited to the interval when the mount is completed. If two cluster access points were used to satisfy a mount with sync mode copy enabled, then this count is incremented only when the data was resident in the TVC of both access points. This value is reset to 0 at the beginning of the interval.	The count is incremented when the mount-complete message is received.

Bytes	Name	Description	When Data is Sampled/Updated
18-21	Average Cache Miss Mount Time	This 4 byte hexadecimal field indicates the average time, in milliseconds, taken to complete Cache Miss mounts during the interval. Mount time is accrued from the time the mount request is accepted by the system until the mount is completed. The mount time is averaged into the interval's time when the mount is completed. This value is reset to 0 at the beginning of the interval.	The time is incremented for each mount and averaged at the end of the interval.
22-23	Sync level mounts	This 2 byte hexadecimal field indicates the number of mount requests completed using the sync mode copy method during this interval. Only mounts using both the primary cluster access point and the secondary cluster access point are included in this count. A mount is accredited to the interval when the mount is completed. This value is reset to 0 at the beginning of the interval.	The count is incremented when the mount-complete message is received.
24-27	Sync level mount time	This 4 byte hexadecimal field indicates the average time, in milliseconds, taken to complete Sync level mounts during the interval. Mount time is accrued from the time the mount request is accepted. By the system until the mount is completed. The mount time is averaged into the interval's time when the mount is completed. This value is reset to 0 at the beginning of the interval.	The time is incremented for each mount and averaged at the end of the interval.
28-31	Total Size of Migrated Data (32 bit)	This 4 byte hexadecimal field contains the total size of lvols which are in migrated state. The value is reported in increments of 1000 MiB (1024 x 1024 x 1000). Any residual data will cause the value to be rounded down to the next smaller value. This is the value at the end of the interval. This field is deprecated and remains just for the backward compatibility. It still stores the value, but it shows xFFFFFFFF when the value hit the limit of the 32 bit maximum size. Refer to the new 8 byte field in the Extended HSM-Cache-Partition container to be ready for the cache larger than 4 PiB.	

HSM – Cache – Partition – Preference Group Container

Relative bytes 32 – 127 (2 x 48 bytes = 96 bytes)
These bytes are relative to the HSM – Cache - Partition Container.

This next set of bytes contains information for 2 preference groups for the cache partition. Each set of data contains 48 bytes. The following fields define the 48 bytes and are numbered starting with byte 0. The first preference group's (PG0) data can be found in bytes 0-47 and the second preference group's (PG1) data can be found in relative bytes 48-95.

Note that values for the resident partition are stored only in Preference Group 1 container.

0-3	Virtual Volumes in Cache	This 4 byte hexadecimal field contains the number of virtual volumes in the Tape Volume Cache (TVC) partition that are assigned to the preference group this data is for. This is the value at the end of the interval.	

Bytes	Name	Description	When Data is Sampled/Updated
4-7	Data Resident in Cache (32 bit)	This 4 byte hexadecimal field contains the amount of data in the TVC partition whose volumes are assigned to the preference this data is for. The value is reported in increments of 1 MiB (1024 x 1024). Any residual data will cause the value to be rounded up to the next higher value. This is the value at the end of the interval. This field is deprecated and remains just for the backward compatibility. It still stores the value, but it shows xFFFFFFFF when the value hit the limit of the 32 bit maximum size. Refer to the new 8 byte field in the Extended HSM-Cache-Partition-Preference Group container to be ready for the cache larger than 4 PiB.	
8-11	4 Hour Average Cache Age	This 4 byte hexadecimal field contains the average age, in minutes, of the oldest logical volume in cache, excluding outliers, from the previous 4 hourly samples. Each hourly sample discards “outliers” that are small numbers of logical volumes that are not representative of the cache as a whole. This value is for volumes that were assigned to the preference group this data is for. Each sample is rounded up to the nearest minute. This value is calculated once an hour, on the hour. The data for this field is calculated at the end of the interval. For the model VEA or VEB, or for a partition which is defined not to migrate out to tape or cloud, this time represents the average cache age based on when the volumes are expired, rather than migrated.	The age is updated once an hour, on the hour.
12-15	Volumes Migrated Last 4 Hours	This 4 byte hexadecimal field contains the number of virtual volumes migrated from the cache partition over the past 4 hours that are assigned to the preference group this data is for. This data is calculated once an hour, on the hour. The data for this field is calculated at the end of the interval.	The count is updated once an hour, on the hour.
16-19	48 Hour Average Cache Age	This 4 byte hexadecimal field contains the average age, in minutes, of the oldest logical volume in cache, excluding outliers, from the previous 48 hourly samples. Each hourly sample discards “outliers” that are small numbers of logical volumes that are not representative of the cache as a whole. This value is for volumes that were assigned to the preference group this data is for. Each sample is rounded up to the nearest minute. This value is calculated once an hour, on the hour. The data for this field is calculated at the end of the interval. For the model VEA or VEB, or for a partition which is defined not to migrate out to tape or cloud, this time represents the average cache age based on when the volumes are expired, rather than migrated.	The age is updated once an hour, on the hour.
20-23	Volumes Migrated Last 48 Hours	This 4 byte hexadecimal field contains the number of virtual volumes migrated from the cache partition over the past 48 hours that are assigned to the preference group this data is for. This data is calculated once an hour, on the hour. The data for this field is calculated at the end of the interval.	The count is updated once an hour, on the hour.

Bytes	Name	Description	When Data is Sampled/Updated
24-27	35 Day Average Cache Age	This 4 byte hexadecimal field contains the average age, in minutes, of the oldest logical volume in cache, excluding outliers, from the previous 35 days worth of hourly samples. Each hourly sample discards "outliers" that are small numbers of logical volumes that are not representative of the cache as a whole. This value is for volumes that were assigned to the preference group this data is for. Each sample is rounded up to the nearest minute. This value is calculated once an hour, on the hour. The data for this field is calculated at the end of the interval. For the model VEA or VEB, or for a partition which is defined not to migrate out to tape or cloud, this time represents the average cache age based on when the volumes are expired, rather than migrated.	The age is updated once an hour, on the hour.
28-31	Volumes Migrated Last 35 Days	This 4 byte hexadecimal field contains the number of virtual volumes migrated from the cache partition over the past 35 days that are assigned to the preference group this data is for. This data is calculated once an hour, on the hour. The data for this field is calculated at the end of the interval.	The count is updated once an hour, on the hour.
32-35	Un-premigrated Data (32 bit)	This 4 byte hexadecimal field contains the amount of data in the TVC partition whose volumes are assigned to this preference group, and are not yet premigrated to physical tape or cloud (cache only). The value is reported in increments of 1 MiB (1024 x 1024). Any residual data will cause the value to be rounded up to the next higher value. This is the value at the end of the interval. For the model VEA or VEB, or for a partition which is defined not to migrate out to tape or cloud, all active data should be in the un-premigrated state. This field is deprecated and remains just for the backward compatibility. It still stores the value, but it shows xFFFFFFFF when the value hit the limit of the 32 bit maximum size. Refer to the new 8 byte field in the Extended HSM-Cache-Partition-Preference Group container to be ready for the cache larger than 4 PiB.	
36-39	Awaiting Replication to available Clusters (32 bit)	This 4 byte hexadecimal field contains the amount of data in the TVC partition whose volumes are assigned to this preference group, and are awaiting replication to other available clusters. Data to be replicated to clusters which are either not available (service or offline) or are blocked from receiving copies (Host Console Request) are not counted. This field depicts data that resides in cache. Data to be replicated that exists on tape or cloud only is not included. The value is reported in increments of 1 MiB (1024 x 1024). Any residual data will cause the value to be rounded up to the next higher value. This is the value at the end of the interval. This field is deprecated and remains just for the backward compatibility. It still stores the value, but it shows xFFFFFFFF when the value hit the limit of the 32 bit maximum size. Refer to the new 8 byte field in the Extended HSM-Cache-Partition-Preference Group container to be ready for the cache larger than 4 PiB.	
40-43	Removed copies average age	This 4 byte hexadecimal field contains the average age of the removed copies. The age is in minutes.	Updated every 4 hours

Bytes	Name	Description	When Data is Sampled/Updated
44-47	Copies removal count	This 4 byte hexadecimal field contains the count of copy volumes removed over the last 4 hours.	Updated every 4 hours

Extended HSM – Cache Container

Bytes 1216 – 4671

This set of bytes contains additional information concerning the Tape Volume Cache (TVC) and this Hnode.

0-1	Length	This 2 byte hexadecimal field contains the length of this container. The length includes these 2 bytes.	
2-3	Offset to Cache Partition 0	This 2 byte hexadecimal field contains the offset to cache partition 0 container. The offset is measured from beginning of this container.	
4-5	Offset to Cache Partition 1	This 2 byte hexadecimal field contains the offset to cache partition 1 container. The offset is measured from beginning of this container.	
6-7	Offset to Cache Partition 2	This 2 byte hexadecimal field contains the offset to cache partition 2 container. The offset is measured from beginning of this container.	
8-9	Offset to Cache Partition 3	This 2 byte hexadecimal field contains the offset to cache partition 3 container. The offset is measured from beginning of this container.	
10-11	Offset to Cache Partition 4	This 2 byte hexadecimal field contains the offset to cache partition 4 container. The offset is measured from beginning of this container.	
12-13	Offset to Cache Partition 5	This 2 byte hexadecimal field contains the offset to cache partition 5 container. The offset is measured from beginning of this container.	
14-15	Offset to Cache Partition 6	This 2 byte hexadecimal field contains the offset to cache partition 6 container. The offset is measured from beginning of this container.	
16-17	Offset to Cache Partition 7	This 2 byte hexadecimal field contains the offset to cache partition 7 container. The offset is measured from beginning of this container.	
18-31	Reserved	All bytes set to x00	
32	Percent Host Write Throttle on Tape/Cloud Attached Cache Partitions	This 1 byte hexadecimal field indicates the percentage of 30-second periods where there was at least one throttling value greater than zero on tape/cloud attached cache partitions. This field is filled in by the Hnode that is currently in charge of TVC management. The Hnode that isn't in charge of the TVC management reports x00 in this field. Refer to the Active Hnode field in HSM – Cache container. This value is recomputed at the end of the interval based on the data from the interval.	The current throttle value is sampled every 30 seconds. The percentage is computed over the interval.
33-36	Average Host Write Throttle on Tape/Cloud Attached Cache Partitions	This 4 byte hexadecimal field indicates the average host write throttle value on tape/cloud attached cache partitions during the interval. The value presented is the average of the non-zero throttling values. The value is reported in thousandths of a second. This field is filled in by the Hnode that is currently in charge of TVC management. The Hnode that isn't in charge of the TVC management reports x00000000 in this field. Refer to the Active Hnode field in HSM – Cache container. This value is recomputed at the end of the interval based on the data from the interval.	The current throttle value is sampled every 30 seconds. The average is computed over the interval.

Bytes	Name	Description	When Data is Sampled/Updated
37	Percent Copy Throttle on Tape/Cloud Attached Cache Partitions	This 1 byte hexadecimal field indicates the percentage of 30-second periods where there was at least one throttling value greater than zero on tape/cloud attached cache partitions and that copy was the predominant reason for throttling. This field is filled in by the Hnode that is currently in charge of TVC management. The Hnode that isn't in charge of the TVC management reports x00 in this field. Refer to the Active Hnode field in HSM – Cache container. This value is recomputed at the end of the interval based on the data from the interval.	The current throttle value is sampled every 30 seconds. The percentage is computed over the interval.
38-41	Average Copy Throttle on Tape/Cloud Attached Cache Partitions	This 4 byte hexadecimal field indicates the average copy throttle value on tape/cloud attached cache partitions during the interval. The value presented is the average of the non-zero throttling values where copy was the predominant reason for throttling. The value is reported in thousandths of a second. This field is filled in by the Hnode that is currently in charge of TVC management. The Hnode that isn't in charge of the TVC management reports x00000000 in this field. Refer to the Active Hnode field in HSM – Cache container. This value is recomputed at the end of the interval based on the data from the interval.	The current throttle value are sampled every 30 seconds. The average is computed over the interval.
42-45	Average Overall Throttle on Tape/Cloud Attached Cache Partitions	This 4 byte hexadecimal field indicates the average of all throttling values on tape/cloud attached cache partitions during the interval. The calculation includes samples for periods where throttling was both zero and non-zero. The value is reported in thousandths of a second. This field is filled in by the Hnode that is currently in charge of TVC management. The Hnode that isn't in charge of the TVC management reports x00000000 in this field. Refer to the Active Hnode field in HSM – Cache container.	The current throttle values are sampled every 30 seconds. The average is computed over the interval.
46	Percent Deferred Copy Throttle on Tape/Cloud Attached Cache Partitions	This 1 byte hexadecimal field indicates the percentage of 30-second periods where deferred copy throttle from tape/cloud attached cache partitions of this Hnode was active (non-zero). This field is filled in by the Hnode that is currently in charge of TVC management. The Hnode that isn't in charge of the TVC management reports x00 in this field. Refer to the Active Hnode field in HSM – Cache container. This value is recomputed at the end of the interval based on the data from the interval.	
47-50	Average Deferred Copy Throttle on Tape/Cloud Attached Cache Partitions	This 4 byte hexadecimal field indicates the average of 30-second intervals of the deferred copy throttle value on tape/cloud attached cache partitions over the historical record interval. The value is reported in thousandths of a second. This value is recomputed at the end of the interval based on the data from the interval.	

<table>
<tr><th>Bytes</th><th>Name</th><th>Description</th><th>When Data is Sampled/Updated</th></tr>
<tr><td>51-54</td><td>Base Deferred Copy Throttle on Tape/Cloud Attached Cache Partitions</td><td>This 4 byte hexadecimal field indicates the base deferred copy throttle value on tape/cloud attached cache partitions which would be used if deferred copy throttle were to be applied. The value is reported in thousandths of a second.</td><td></td></tr>
<tr><td>55-56</td><td>Host Write Throttle Reason(s) on Tape/Cloud Attached Cache Partitions</td><td>This 2 byte hexadecimal field indicates the reason(s) for write overrun (host) throttling on tape/cloud attached cache partitions during the interval.
<table>
<tr><th>Value</th><th>Description</th></tr>
<tr><td>x00</td><td>No throttling during the interval</td></tr>
<tr><td>x01</td><td>Premigration steady state (PMTHLVL)</td></tr>
<tr><td>x02</td><td>Low on cache free space</td></tr>
<tr><td>x04</td><td>Immediate copy throttling</td></tr>
<tr><td>x08</td><td>Excess cached content for copy</td></tr>
<tr><td>x10</td><td>Grid premigration steady state (throttling outbound copies because the target cluster is premigration throttling)</td></tr>
<tr><td></td><td>All other values are reserved</td></tr>
</table></td><td>This value is updated every 30 seconds.</td></tr>
<tr><td>57-58</td><td>Copy Throttle Reason(s) on Tape/Cloud Attached Cache Partitions</td><td>This 2 byte hexadecimal field indicates the reason(s) for copy throttling on tape/cloud attached cache partitions during the interval.
<table>
<tr><th>Value</th><th>Description</th></tr>
<tr><td>x00</td><td>No throttling during the interval</td></tr>
<tr><td>x01</td><td>Premigration steady state (PMTHLVL)</td></tr>
<tr><td>x02</td><td>Low on cache free space</td></tr>
<tr><td></td><td>All other values are reserved</td></tr>
</table></td><td>This value is updated every 30 seconds.</td></tr>
<tr><td>59-60</td><td>Deferred Copy Throttle Reason(s) on Tape/Cloud Attached Cache Partitions</td><td>This 2 byte hexadecimal field indicates the reasons for deferred copy throttling on tape/cloud attached cache partitions during the interval.
<table>
<tr><th>Value</th><th>Description</th></tr>
<tr><td>x00</td><td>No throttling during the interval</td></tr>
<tr><td>x01</td><td>Preferring Host I/O</td></tr>
<tr><td>x02</td><td>Cluster resource usage</td></tr>
<tr><td>x04</td><td>Immediate copy throttling</td></tr>
<tr><td></td><td>All other values are reserved</td></tr>
</table></td><td>This value is updated every 30 seconds.</td></tr>
<tr><td>61-68</td><td>TVC Size (64 bit)</td><td>This 8 byte hexadecimal field indicates the current size of the Cluster’s TVC in increments of 1000MiB (1000 x 1024 x 1024).
This field is filled in by the Hnode that is currently in charge of TVC management. The Hnode that isn’t in charge of the TVC management reports x00000000 in this field. Refer to the Active Hnode field.
This is the value at the end of the interval.
The original 4 byte field in the HSM-Cache container still stores the value, but it shows xFFFFFFFF when the value hit the limit of the 32 bit maximum size. Refer to this 8 byte field to be ready for the cache larger than 4 PiB.</td><td></td></tr>
<tr><td>69-76</td><td>Total used cache (64 bit)</td><td>This 8 byte hexadecimal field indicates the amount of cache used in the system. This value is in gigabytes (1000 x 1000 x 1000).
The original 4 byte field in the HSM-Cache container still stores the value, but it shows xFFFFFFFF when the value hit the limit of the 32 bit maximum size. Refer to this 8 byte field to be ready for the cache larger than 4 PiB.</td><td></td></tr>
</table>

Bytes	Name	Description	When Data is Sampled/Updated
77-84	Total used flash cache (64 bit)	This 8 byte hexadecimal field indicates the amount of flash copy cache used in the system. This value is in gigabytes (1000 x 1000 x 1000). The original 4 byte field in the HSM-Cache container still stores the value, but it shows xFFFFFFFF when the value hit the limit of the 32 bit maximum size. Refer to this 8 byte field to be ready for the cache larger than 4 PiB.	
85-127	Reserved	All bytes set to x00	

Extended HSM – Cache – Partition Container

Bytes 1344 – 4671 (8 x 416 bytes = 3328 bytes)

This next set of bytes contains additional information for up to 8 cache partitions for the Cluster.

0-1	Length	This 2 byte hexadecimal field contains the length of this container. The length includes these 2 bytes.	
2-9	Partition Size (64 bit)	This 8 byte hexadecimal field indicates the amount of cache assigned to this partition. The value is reported in increments of 1000MiB (1000 x 1024 x 1024). This field is filled in by the Hnode that is currently in charge of TVC management. The Hnode that isn’t in charge of the TVC management reports x00000000 in this field. Refer to the Active Hnode field. This is the value at the end of the interval. The original 4 byte field in the HSM-Cache-Partition container still stores the value, but it shows xFFFFFFFF when the value hit the limit of the 32 bit maximum size. Refer to this 8 byte field to be ready for the cache larger than 4 PiB.	
10-17	Total Size of Migrated Data (64 bit)	This 8 byte hexadecimal field contains the total size of lvols which are in migrated state. The value is reported in increments of 1000 MiB (1024 x 1024 x 1000). Any residual data will cause the value to be rounded down to the next smaller value. This is the value at the end of the interval. The original 4 byte field in the HSM-Cache-Partition container still stores the value, but it shows xFFFFFFFF when the value hit the limit of the 32 bit maximum size. Refer to this 8 byte field to be ready for the cache larger than 4 PiB.	
18-31	Reserved	All bytes set to x00	

Extended HSM – Cache – Partition – Preference Group Container

Relative bytes 32 – 415 (2 x 192 bytes = 384 bytes)
These bytes are relative to the Extended HSM – Cache - Partition Container.

This next set of bytes contains additional information for 2 preference groups for the cache partition.

Note that values for the resident partition are stored only in Preference Group 1 container.

0-1	Length	This 2 byte hexadecimal field contains the length of this container. The length includes these 2 bytes.	

Bytes	Name	Description	When Data is Sampled/Updated
2-5	Number of Prefer Keep Volumes	This 4 byte hexadecimal field contains the number of virtual volumes that are assigned to prefer-keep. This is the value at the end of the interval.	
6-9	Total Size of Prefer Keep Volumes (32 bit)	This 4 byte hexadecimal field contains the amount of data in the TVC partition whose volumes are assigned to prefer-keep. The value is reported in increments of 1000 MiB (1024 x 1024 x 1000). Any residual data will cause the value to be rounded up to the next higher value. This is the value at the end of the interval. This field is deprecated and remains just for the backward compatibility. It still stores the value, but it shows xFFFFFFFF when the value hit the limit of the 32 bit maximum size. Refer to the new 8 byte field in this container to be ready for the cache larger than 4 PiB.	
10-13	Number of Prefer Remove Volumes	This 4 byte hexadecimal field contains the number of virtual volumes that are assigned to prefer-remove. This is the value at the end of the interval.	
14-17	Total Size of Prefer Remove Volumes (32 bit)	This 4 byte hexadecimal field contains the amount of data in the TVC partition whose volumes are assigned to prefer-remove. The value is reported in increments of 1000 MiB (1024 x 1024 x 1000). Any residual data will cause the value to be rounded up to the next higher value. This is the value at the end of the interval. This field is deprecated and remains just for the backward compatibility. It still stores the value, but it shows xFFFFFFFF when the value hit the limit of the 32 bit maximum size. Refer to the new 8 byte field in this container to be ready for the cache larger than 4 PiB.	
18-21	Number of Pinned Volumes	This 4 byte hexadecimal field contains the number of virtual volumes that are assigned to pinned. This is the value at the end of the interval.	
22-25	Total Size of Pinned Volumes (32 bit)	This 4 byte hexadecimal field contains the amount of data in the TVC partition whose volumes are assigned to pinned. The value is reported in increments of 1000 MiB (1024 x 1024 x 1000). Any residual data will cause the value to be rounded up to the next higher value. This is the value at the end of the interval. This field is deprecated and remains just for the backward compatibility. It still stores the value, but it shows xFFFFFFFF when the value hit the limit of the 32 bit maximum size. Refer to the new 8 byte field in this container to be ready for the cache larger than 4 PiB.	
26-29	4 Hour Average Cache Age by Delayed Premigration	This 4 byte hexadecimal field contains the average age, in minutes, of the oldest logical volume in cache by delayed premigration, excluding outliers, from the previous 4 hourly samples. Each hourly sample discards “outliers” that are small numbers of logical volumes that are not representative of the cache as a whole. This value is for volumes that were assigned to the preference group this data is for. Each sample is rounded up to the nearest minute. This value is calculated once an hour, on the hour. The data for this field is calculated at the end of the interval. For the model VEA or VEB, or for a partition which is defined not to migrate out to tape, this time represents the average cache age based on when the volumes are expired, rather than migrated.	

Bytes	Name	Description	When Data is Sampled/Updated
30-33	Volumes Migrated Last 4 Hours by Delayed Premigration	This 4 byte hexadecimal field contains the number of virtual volumes migrated from the cache partition by delayed premigration over the past 4 hours that are assigned to the preference group this data is for. This data is calculated once an hour, on the hour. The data for this field is calculated at the end of the interval.	
34-37	48 Hours Average Cache Age by Delayed Premigration	This 4 byte hexadecimal field contains the average age, in minutes, of the oldest logical volume in cache by delayed premigration, excluding outliers, from the previous 48 hourly samples. Each hourly sample discards “outliers” that are small numbers of logical volumes that are not representative of the cache as a whole. This value is for volumes that were assigned to the preference group this data is for. Each sample is rounded up to the nearest minute. This value is calculated once an hour, on the hour. The data for this field is calculated at the end of the interval. For the model VEA or VEB, or for a partition which is defined not to migrate out to tape, this time represents the average cache age based on when the volumes are expired, rather than migrated.	
38-41	Volumes Migrated Last 48 Hours by Delayed Premigration	This 4 byte hexadecimal field contains the number of virtual volumes migrated from the cache partition by delayed premigration over the past 48 hours that are assigned to the preference group this data is for. This data is calculated once an hour, on the hour. The data for this field is calculated at the end of the interval.	
42-45	35 Days Average Cache Age by Delayed Premigration	This 4 byte hexadecimal field contains the average age, in minutes, of the oldest logical volume in cache by delayed premigration, excluding outliers, from the previous 35 days worth of hourly samples. Each hourly sample discards “outliers” that are small numbers of logical volumes that are not representative of the cache as a whole. This value is for volumes that were assigned to the preference group this data is for. Each sample is rounded up to the nearest minute. This value is calculated once an hour, on the hour. The data for this field is calculated at the end of the interval. For the model VEA or VEB, or for a partition which is defined not to migrate out to tape, this time represents the average cache age based on when the volumes are expired, rather than migrated.	
46-49	Volumes Migrated Last 35 Days by Delayed Premigration	This 4 byte hexadecimal field contains the number of virtual volumes migrated from the cache partition by delayed premigration over the past 35 days that are assigned to the preference group this data is for. This data is calculated once an hour, on the hour. The data for this field is calculated at the end of the interval.	
50-53	Un-premigrated Vols	Number of un-premigrated virtual volumes. Delayed premigration volumes are excluded.	
54-57	Average Waiting Time of Delayed Premigration Volumes	The average waiting time of delayed premigration volumes during the interval. The value is reported in seconds. This field is filled in by the Hnode that is currently in charge of TVC management. The Hnode that isn’t in charge of the TVC management reports x00000000 in this field. Refer to the Active Hnode field in HSM – Cache container. This value is recomputed at the end of the interval based on the data from the interval.	

Bytes	Name	Description	When Data is Sampled/Updated
58-61	Total Size of Resident Volumes Waiting for Delayed Premigration (32 bit)	The amount of data in the TVC partition whose volumes are resident on TVC waiting for delayed premigration. The value is reported in increments of 1 MiB (1024 x 1024). Any residual data will cause the value to be rounded up to the next higher value. This is the value at the end of the interval. This field is deprecated and remains just for the backward compatibility. It still stores the value, but it shows xFFFFFFFF when the value hit the limit of the 32 bit maximum size. Refer to the new 8 byte field in this container to be ready for the cache larger than 4 PiB.	
62-65	Resident Volumes Waiting for Delayed Premigration	Number of resident volumes on TVC waiting for delayed premigration.	
66-73	Data Resident in Cache (64 bit)	This 8 byte hexadecimal field contains the amount of data in the TVC partition whose volumes are assigned to the preference this data is for. The value is reported in increments of 1 MiB (1024 x 1024). Any residual data will cause the value to be rounded up to the next higher value. This is the value at the end of the interval. The original 4 byte field in the HSM-Cache-Partition-Preference Group container still stores the value, but it shows xFFFFFFFF when the value hit the limit of the 32 bit maximum size. Refer to this 8 byte field to be ready for the cache larger than 4 PiB.	
74-81	Un-premigrated Data (64 bit)	This 8 byte hexadecimal field contains the amount of data in the TVC partition whose volumes are assigned to this preference group, and are not yet premigrated to physical tape or cloud (cache only). The value is reported in increments of 1 MiB (1024 x 1024). Any residual data will cause the value to be rounded up to the next higher value. This is the value at the end of the interval. For the model VEA or VEB, or for a partition which is defined not to migrate out to tape or cloud, all active data should be in the un-premigrated state. The original 4 byte field in the HSM-Cache-Partition-Preference Group container still stores the value, but it shows xFFFFFFFF when the value hit the limit of the 32 bit maximum size. Refer to this 8 byte field to be ready for the cache larger than 4 PiB.	

Bytes	Name	Description	When Data is Sampled/Updated
82-89	Awaiting Replication to available Clusters (64 bit)	This 8 byte hexadecimal field contains the amount of data in the TVC partition whose volumes are assigned to this preference group, and are awaiting replication to other available clusters. Data to be replicated to clusters which are either not available (service or offline) or are blocked from receiving copies (Host Console Request) are not counted. This field depicts data that resides in cache. Data to be replicated that exists on tape or cloud only is not included. The value is reported in increments of 1 MiB (1024 x 1024). Any residual data will cause the value to be rounded up to the next higher value. This is the value at the end of the interval. The original 4 byte field in the HSM-Cache-Partition-Preference Group container still stores the value, but it shows xFFFFFFFF when the value hit the limit of the 32 bit maximum size. Refer to this 8 byte field to be ready for the cache larger than 4 PiB.	
90-97	Total Size of Prefer Keep Volumes (64 bit)	This 8 byte hexadecimal field contains the amount of data in the TVC partition whose volumes are assigned to prefer-keep. The value is reported in increments of 1000 MiB (1024 x 1024 x 1000). Any residual data will cause the value to be rounded up to the next higher value. This is the value at the end of the interval. The original 4 byte field in this container still stores the value, but it shows xFFFFFFFF when the value hit the limit of the 32 bit maximum size. Refer to this 8 byte field to be ready for the cache larger than 4 PiB.	
98-105	Total Size of Prefer Remove Volumes (64 bit)	This 8 byte hexadecimal field contains the amount of data in the TVC partition whose volumes are assigned to prefer-remove. The value is reported in increments of 1000 MiB (1024 x 1024 x 1000). Any residual data will cause the value to be rounded up to the next higher value. This is the value at the end of the interval. The original 4 byte field in this container still stores the value, but it shows xFFFFFFFF when the value hit the limit of the 32 bit maximum size. Refer to this 8 byte field to be ready for the cache larger than 4 PiB.	
106-113	Total Size of Pinned Volumes (64 bit)	This 8 byte hexadecimal field contains the amount of data in the TVC partition whose volumes are assigned to pinned. The value is reported in increments of 1000 MiB (1024 x 1024 x 1000). Any residual data will cause the value to be rounded up to the next higher value. This is the value at the end of the interval. The original 4 byte field in this container still stores the value, but it shows xFFFFFFFF when the value hit the limit of the 32 bit maximum size. Refer to this 8 byte field to be ready for the cache larger than 4 PiB.	

Bytes	Name	Description	When Data is Sampled/Updated
114-121	Total Size of Resident Volumes Waiting for Delayed Premigration (64 bit)	The amount of data in the TVC partition whose volumes are resident on TVC waiting for delayed premigration. The value is reported in increments of 1 MiB (1024 x 1024). Any residual data will cause the value to be rounded up to the next higher value. This is the value at the end of the interval. The original 4 byte field in this container still stores the value, but it shows xFFFFFFFF when the value hit the limit of the 32 bit maximum size. Refer to this 8 byte field to be ready for the cache larger than 4 PiB.	
122-129	Objects in Cache	This 8 byte hexadecimal field contains the number of objects in the Tape Volume Cache (TVC) partition that are assigned to the preference group this data is for. This is the value at the end of the interval.	
130-191	Reserved	All bytes set to x00	

HSM – Compression – Compression Container

This next set of bytes contains information for compression.

Bytes	Name	Description	When Data is Sampled/Updated
0-1	Length	This 2 byte hexadecimal field contains the length of this container. The length includes these 2 bytes.	
2	Number of Compression Methods	This 2 byte hexadecimal field contains the number of Compression Method currently supported. This field can be used to determine how many Compression Method containers will be attached to this record.	
3-15	Reserved	All bytes set to x00.	

HSM – Compression – Compression Method Container

This next set of bytes contains information for compression method.

Number of this container attached to this record can be determined by Number of Compression Methods field in the parent Compression container.

Bytes	Name	Description	When Data is Sampled/Updated
0-1	Length	This 2 byte hexadecimal field contains the length of this container. The length includes these 2 bytes.	
2	Compression Method	This 1 byte hexadecimal field contains the value to indicate Compression Method. **Value** **Description** x01 FICON compression x02 LZ4 compression x03 ZSTD compression All other values are reserved	
3-6	Uncompressed Read Bytes	This 4 byte hexadecimal field contains the number of bytes read from LVOLs on the local cluster during the last interval. The value is reported in increments of 1 MiB (1024x1024). This value is reset to 0 at beginning of the interval.	Updated when LVOL is unmounted.
7-10	Uncompressed Write Bytes	This 4 byte hexadecimal field contains the number of bytes written to LVOLs on the local cluster during the last interval. The value is reported in increments of 1 MiB (1024x1024). This value is reset to 0 at beginning of the interval.	Updated when LVOL is unmounted.
11-14	Compressed Read Bytes	This 4 byte hexadecimal field contains the number of compressed bytes read from LVOLs on the local cluster during the last interval. The value is reported in increments of 1 MiB (1024x1024). This value is reset to 0 at beginning of the interval.	Updated when LVOL is unmounted.

Bytes	Name	Description	When Data is Sampled/Updated
15-18	Compressed Write Bytes	This 4 byte hexadecimal field contains the number of compressed bytes written to LVOLs on the local cluster during the last interval. The value is reported in increments of 1 MiB (1024x1024). This value is reset to 0 at beginning of the interval.	Updated when LVOL is unmounted.
19-31	Reserved	All bytes set to x00.	

Hnode Export/Import Historical Record

This Hnode historical record has the following nested structure:

- Header
- Export/Import Container

Bytes	Name	Description	When Data is Sample/Updated
0-1	Length	This 2 byte hexadecimal field contains the length of this record. The length includes these 2 bytes.	
2	Version	This 1 byte hexadecimal field contains the version of the data presented in this record. The current version is set to x02.	
3	Data Type	This 1 byte hexadecimal field indicates the type of data contained in this record. For this record the value is set to x31 indicating this is an Hnode Export/Import Historical record.	
4	Node ID	This 1 byte hexadecimal field indicates the Hnode ID which this interval's data represents. Valid values are x00 – x01.	
5	Cluster ID	This 1 byte hexadecimal field indicates the Cluster ID which this Hnode is a part of. Valid values are x00 – x07.	
6-7	Interval Duration	This 2 byte hexadecimal field indicates the interval in seconds that this interval's data was taken over.	
8-11	Time Stamp	This 4 byte hexadecimal field indicates the end time of the interval this data was taken over. This value is the time in seconds since the Epoch (00:00:00 UTC, January 1, 1970)	
12-15	Machine Type	This 4 byte EBCDIC field contains this node's machine type. The field is left justified padded with EBCDIC blanks. Initially this field will be set to "3957".	
16-18	Machine Model	This 3 byte EBCDIC field contains this node's machine model. The field is left justified padded with EBCDIC blanks. Initially this field will be set to "V06".	
19-26	Machine Serial Number	This 8 character EBCDIC field contains the serial number of this node. This field is left justified and padded with EBCDIC blanks. The format is XX-YYYYY where XX is the plant of manufacture and the YYYYY is the sequence number of the node's machine. The dash character (-) is fixed.	
27-34	Code Level	This 8 byte hexadecimal field contains the code level of the TS7700. The 8 bytes are actually four 2-byte fields. Each 2-byte field represents a portion of the code level. The VE code level is expressed as Version.Release.Modification.Fix in a decimal form. For example the code level of 8.0.0.104 would be represented in the 8 bytes as: x0008000000000068.	
35-39	Grid Library Sequence Number	This 5 character EBCDIC field contains the Library Sequence Number of the Grid (Composite) library.	
40-44	Distributed Library Sequence Number	This 5 character EBCDIC field contains the Distributed Library Sequence Number for this Distributed Library ID	
45-63	Reserved	All bytes set to x00	

Bytes	**Name**	**Description**	**When Data is Sample/Updated**
Export/Import Container **Bytes 64-127 (64 bytes)** **This set of bytes contains information pertinent to Export/Import operations performed by this Hnode.**			
64-65	Physical Volumes Imported	This 2 byte hexadecimal field indicates the number of physical volumes processed as part of an Import operation during the interval. This value is reset to 0 at the beginning of the interval.	Import isn't supported at this time.
66-67	Physical Volumes Exported	This 2 byte hexadecimal field indicates the number of physical volumes processed as part of an Export operation during the interval. This value is reset to 0 at the beginning of the interval.	Updated as each physical volume is exported.
68-71	Logical Volumes Imported	This 4 byte hexadecimal field indicates the number of logical volumes successfully imported during the interval. This value is reset to 0 at the beginning of the interval.	Import isn't supported at this time.
72-75	Logical Volumes Exported	This 4 byte hexadecimal field indicates the number of logical volumes successfully exported during the interval. This value is reset to 0 at the beginning of the interval.	Updated as each logical volume is exported.
76-79	Amount of data imported	This 4 byte hexadecimal value contains the amount of data imported during import operations that completed during the interval. Logical volumes being imported with import operations of SCRATCH or INITIALIZE are not added to the count because no customer data is moved. The number of MiBs (1024x1024) imported is the sum of the MiB integer values of the data imported from each Exported Stacked Volume. The MiB integer value for the data imported from each Exported Stacked Volume is the full count by byte divided by 1024x1024 (1 MiB). If the result is less than 1, the MiB integer becomes 1, and if greater than 1 MiB, the result is truncated to the integer value (rounded down) before being added to the count. This value is reset to 0 at the beginning of the interval.	Import isn't supported at this time.
80-83	Amount of data exported	This 4 byte hexadecimal value contains the amount of data exported during export operations that completed during the interval. The number of MiB (1024x1024) imported is the sum of the MiB integer values of the data exported to each Exported Stacked Volume. The MiB integer value for the data exported to each Exported Stacked Volume is the full count by byte divided by 1024x1024 (1 MiB). If the result is less than 1, the MiB integer becomes 1, and if greater than 1 MiB, the result is truncated to the integer value (rounded down) before being added to the count. This value is reset to 0 at the beginning of the interval.	Updated as each physical volume is exported.
84-127	Reserved	All bytes set to x00.	

Hnode Library Historical Record

This Hnode historical record is generated if the cluster is tape attached.
The record has the following nested structure:

- Header
- Library Container
    - Library Device Type 0 Usage Container
    - Library Device Type 1 Usage Container
    - Library Device Type 2 Usage Container
    - Library Device Type 3 Usage Container
    - Library-Pooling Container
        - Library-Pooling-CSP Media Type 0 Container
        - Library-Pooling-CSP Media Type 1 Container
        - Library-Pooling-CSP Media Type 2 Container
        - Library-Pooling-CSP Media Type 3 Container
        - Library-Pooling-CSP Media Type 4 Container
        - Library-Pooling-CSP Media Type 5 Container
        - Library-Pooling-CSP Media Type 6 Container
        - Library-Pooling-CSP Media Type 7 Container
        - Library-Pooling-GUP 1 Container
            - Library-Pooling-GUP-Media Type 0 Container
            - Library-Pooling-GUP-Media Type 1 Container
            - Library-Pooling-GUP-Media Type 2 Container
            - Library-Pooling-GUP-Media Type 3 Container
            - Library-Pooling-GUP-Media Type 4 Container
            - Library-Pooling-GUP-Media Type 5 Container
            - Library-Pooling-GUP-Media Type 6 Container
            - Library-Pooling-GUP-Media Type 7 Container
            - Library-Pooling-GUP-Reclaim Container
            - Library-Pooling-GUP-Properties Container
        - Library-Pooling-GUP 2 Container
            - Same sub-containers as GUP 1
        - .
        - .
        - .
        - Library-Pooling-GUP 32 Container
            - Same sub-containers as GUP 1
    - Physical Device 0 Container (for future use)
    -
    -
    -
    - Physical Device 31 Container (for future use)

Bytes	Name	Description	When Data is Sample/Updated
0-1	Length	This 2 byte hexadecimal field contains the length of this record. The length includes these 2 bytes.	
2	Version	This 1 byte hexadecimal field contains the version of the data presented in this record. The current version is set to x06.	

Bytes	Name	Description	When Data is Sample/Updated
3	Data Type	This 1 byte hexadecimal field indicates the type of data contained in this record. For this record the value is set to x32 indicating this is an Hnode Library Historical record.	
4	Node ID	This 1 byte hexadecimal field indicates the Hnode ID which this interval’s data represents. Valid values are x00 – x01.	
5	Cluster ID	This 1 byte hexadecimal field indicates the Cluster ID which this Hnode is a part of. Valid values are x00 – x07.	
6-7	Interval Duration	This 2 byte hexadecimal field indicates the interval in seconds that this interval’s data was taken over.	
8-11	Time Stamp	This 4 byte hexadecimal field indicates the end time of the interval this data was taken over. This value is the time in seconds since the Epoch (00:00:00 UTC, January 1, 1970)	
12-15	Machine Type	This 4 byte EBCDIC field contains this node’s machine type. The field is left justified padded with EBCDIC blanks. Initially this field will be set to “3957”.	
16-18	Machine Model	This 3 byte EBCDIC field contains this node’s machine model. The field is left justified padded with EBCDIC blanks. Initially this field will be set to “V06”.	
19-26	Machine Serial Number	This 8 character EBCDIC field contains the serial number of this node. This field is left justified and padded with EBCDIC blanks. The format is XX-YYYYY where XX is the plant of manufacture and the YYYYY is the sequence number of the node’s machine. The dash character (-) is fixed.	
27-34	Code Level	This 8 byte hexadecimal field contains the code level of the TS7700. The 8 bytes are actually four 2-byte fields. Each 2-byte field represents a portion of the code level. The VE code level is expressed as Version.Release.Modification.Fix in a decimal form. For example the code level of 8.0.0.104 would be represented in the 8 bytes as: x0008000000000068.	
35-39	Grid Library Sequence Number	This 5 character EBCDIC field contains the Library Sequence Number of the Grid (Composite) library.	
40-44	Distributed Library Sequence Number	This 5 character EBCDIC field contains the Distributed Library Sequence Number for this Distributed Library ID	
45-63	Reserved	All bytes set to x00	
Library Container **Bytes 64-127 (64 bytes)** **This set of bytes contains information pertinent to the Library operations associated with this Hnode.**			
64-69	Library Machine Type	This 6 character EBCDIC field contains the machine type of the underlying automation. The field is left justified and padded with EBCDIC blanks. Initially this field will be set to “3494” or “3584”. This is the value at the end of the interval.	
70-72	Library Model Number	This 3 character EBCDIC field contains the model number of the underlying automation. The field is left justified and padded with EBCDIC blanks. Initially this field will be set to “L10” when attached to a 3494 or “L22” when attached to a 3584. This is the value at the end of the interval.	

Bytes	Name	Description	When Data is Sample/Updated
73-75	Library Manufacturer	This 3 byte EBCDIC field contains an abbreviation of the manufacturer of the underlying automation. The field is left justified and padded with EBCDIC blanks. Initially this will be set to “IBM”. This is the value at the end of the interval.	
76-77	Library Plant of Manufacture	This 2 byte EBCDIC field contains the library plant of manufacture of the underlying automation. The field is left justified and padded with EBCDIC blanks. This is the value at the end of the interval.	
78-93	Library Sequence Number	This 16 byte EBCDIC field contains the sequence number of the underlying automation. The field is left justified and padded with EBCDIC blanks. This is the value at the end of the interval.	
94-127	Reserved	All bytes set to x00.	

Library – Tape Device Usage (TDU) Container

Bytes 128-255 (4 sets of data x 32 bytes/set = 128 bytes)

This container contains 4 sets of data. The 4 sets of data allow up to 4 device types/models to be attached to the Hnode. Each set of data contains 32 bytes. The following fields define these bytes and are numbered starting with 0. Data for the first device type is found in bytes 128-159, the data for the second device type is found in bytes 160-191, and so forth.

Each Hnode reports this data from its perspective.

<table>
<tr><td>0</td><td>Device Class ID</td><td>This one byte hexadecimal field contains this device’s device class identifier.
<table>
<tr><th>Value</th><th>Description</th></tr>
<tr><td>x00</td><td>No device installed</td></tr>
<tr><td>x20</td><td>3592 Model J1A. This also includes a 3592-E05 that is emulating a 3592-J1A device.</td></tr>
<tr><td>x22</td><td>3592 Model E05. This is for a 3592-E05 that is behaving as a 3590-E05.</td></tr>
<tr><td>x23</td><td>3592 Model E05 (encryption configured)</td></tr>
<tr><td>x24</td><td>3592 Model E06</td></tr>
<tr><td>x25</td><td>3592 Model E07</td></tr>
<tr><td>x26</td><td>3592 Model E08</td></tr>
<tr><td></td><td>All other values are reserved.</td></tr>
</table>
This is the value at the end of the interval.</td><td></td></tr>
<tr><td>1</td><td>Installed Physical Devices</td><td>This 1 byte hexadecimal field contains the number of physical devices, of the device class indicated, that are installed at the end of the interval.</td><td></td></tr>
<tr><td>2</td><td>Available Physical Devices</td><td>This 1 byte hexadecimal field contains the number of physical devices, of the device class indicated, that are available for use at the end of the interval.</td><td>Count is updated when the number of available physical devices changes.</td></tr>
<tr><td>3</td><td>Maximum Physical Devices Mounted</td><td>This 1 byte field contains the maximum number of physical devices, of the device class indicated, that were concurrently mounted during the interval.
This value is reset to 0 at the beginning of the interval.</td><td>Count is updated whenever the number of mounted physical devices changes.</td></tr>
<tr><td>4</td><td>Minimum Physical Devices Mounted</td><td>This 1 byte field contains the minimum number of physical devices, of the device class indicated, that were concurrently mounted during the interval.
This value is reset to 0 at the beginning of the interval.</td><td>Count is updated whenever the number of mounted physical devices changes.</td></tr>
</table>

Bytes	Name	Description	When Data is Sample/Updated
5	Average Physical Devices Mounted	This 1 byte field contains the average number of physical devices, of the device class indicated, that were concurrently mounted during the interval. The average is calculated by recording the number of mounted devices on a periodic basis then averaging it over the interval. This value is reset to 0 at the beginning of the interval.	The count of mounted physical devices is sampled every 15 seconds. The average is computed over the interval.
6-7	Maximum Physical Mount Time	This 2 byte field contains the maximum time, in seconds, that it took to complete the execution of a mount request for a physical device, of the device class indicated, over the interval. Mount time is accrued from the time the mount request is sent until the mount complete is received. The mount time is accredited to the interval it was completed. This value is reset to 0 at the beginning of the interval.	The time for each mount is examined when the mount completes.
8-9	Minimum Physical Mount Time	This 2 byte field contains the minimum time, in seconds, that it took to complete the execution of a mount request for a physical device, of the device class indicated, over the interval. Mount time is accrued from the time the mount request is sent until the mount complete is received. The mount time is accredited to the interval it was completed. This value is reset to 0 at the beginning of the interval.	The time for each mount is examined when the mount completes.
10-11	Average Physical Mount Time	This 2 byte field contains the average time, in seconds, that it took to complete the execution of a mount request for a physical device, of the device class indicated, over the interval. Mount time is accrued from the time the mount request is sent until the mount complete is received. The mount time is accredited to the interval it was completed. This value is reset to 0 at the beginning of the interval.	The time for each mount is examined when the mount completes.
12-13	Physical Recall Mounts	This 2 byte hexadecimal field contains the number of physical mount requests completed by the library during the interval to satisfy recall mounts for the device class indicated. A mount is accredited to the interval the mount is completed. This value is reset to 0 at the beginning of the interval.	Count is incremented each time a recall mount is completed.
14-15	Physical Pre-Migrate Mounts	This 2 byte hexadecimal field contains the number of physical mount requests completed by the library during the interval to satisfy pre-migrate mounts for the device class indicated. A mount is accredited to the interval the mount is completed. This value is reset to 0 at the beginning of the interval.	Count is incremented each time a pre-migrate mount is completed.
16-17	Physical Reclaim Mounts	This 2 byte hexadecimal field contains the number of physical mount requests completed by the library during the interval to satisfy reclaim mounts for the device class indicated. A mount is accredited to the interval the mount is completed. This value is reset to 0 at the beginning of the interval.	Count is incremented each time a reclaim mount, source or target, is completed.
18-19	Physical Security Data Erase Mounts	This 2 byte hexadecimal field contains the number of physical mount requests completed by the library during the interval to satisfy Security Data Erase mounts for the device class indicated. A mount is accredited to the interval the mount is completed. This value is reset to 0 at the beginning of the interval.	Count is incremented each time a Security Data Erase mount is completed.
20-31	Reserved	All bytes set to x00.	

<table>
<tr><th>Bytes</th><th>Name</th><th>Description</th><th>When Data is Sample/Updated</th></tr>
<tr><td colspan="3"><b>Library - Pooling Container</b>
<b>Bytes 256-8511</b>
<b>The data from each Hnode will be the same within a Cluster.</b></td><td></td></tr>
<tr><td colspan="3"><b>Library - Pooling – Common Scratch Pool (CSP) Media Container</b>
<b>Bytes 256-319 (8 sets of data x 8 bytes/set = 64 bytes)</b>
<b>These fields contain 8 sets of data describing the physical types in the CSP. Each set of data contains 8 bytes. The data for the first media type can be found in bytes 256-263; the data for the second media type can be found in bytes 264-271, and so forth. All 8 sets will be populated starting with the first media type that has a non-zero Physical Media Identifier. Only media types with a non-zero Physical Media Identifier will occupy a set, then the remainder of the 8 sets will be have its Physical Media Identifier set to no media. If only 3 media types are used in the CSP, then the first 3 sets will be populated with no unused sets. The remaining 5 sets will have their Physical Media Types set to zero, indicating they are not used. The first Physical Media Type that is zero indicates that there are no more used Physical Media types in the subsequent sets</b></td><td></td></tr>
<tr><td>0</td><td>Physical Media Type</td><td>This 1 byte hexadecimal field contains the identifier for the media type associated with following common scratch pool volume counts. The value is recorded at the end of the interval. The following are values for this field:
<table>
<tr><th>Value</th><th>Description</th></tr>
<tr><td>x00</td><td>No media type for this set of data</td></tr>
<tr><td>x10</td><td>3590 J media (reserved)</td></tr>
<tr><td>x11</td><td>3590 K media (reserved)</td></tr>
<tr><td>x20</td><td>3592 JA media</td></tr>
<tr><td>x21</td><td>3592 JW media (reserved for future use)</td></tr>
<tr><td>x22</td><td>3592 JJ media</td></tr>
<tr><td>x23</td><td>3592 JR media (reserved for future use)</td></tr>
<tr><td>x24</td><td>3592 JB Media</td></tr>
<tr><td>x25</td><td>3592 JX Media (reserved for future use)</td></tr>
<tr><td>x26</td><td>3592 JC Media</td></tr>
<tr><td>x27</td><td>3592 JY Media (reserved for future use)</td></tr>
<tr><td>x28</td><td>3592 JK Media</td></tr>
<tr><td>x29</td><td>3592 JD Media</td></tr>
<tr><td>x2A</td><td>3592 JZ Media (reserved for future use)</td></tr>
<tr><td>x2B</td><td>3592 JL Media</td></tr>
<tr><td></td><td>All other values are reserved.</td></tr>
</table>
This is the value at the end of the interval.</td><td>Type is updated every 2 minutes.</td></tr>
<tr><td>1-2</td><td>Physical Media Count</td><td>This two byte hexadecimal field contains the number of scratch stacked volumes, of the type identified, assigned to the common scratch pool. This is the value at the end of the interval.</td><td>Count is updated every 2 minutes.</td></tr>
<tr><td>3</td><td>Media type container count</td><td>This one byte hexadecimal field contains the number of physical media types that have a non-zero Physical Media Count in the common scratch pool. If more than 8 physical media types are non-zero, only the first 8 will be returned.</td><td></td></tr>
<tr><td>4-7</td><td>Reserved</td><td>All bytes set to x00.</td><td></td></tr>
</table>

Bytes	Name	Description	When Data is Sample/Updated
Library - Pooling – General Use Pool (GUP) Container **Bytes 320-8511 (32 sets of data x 256 bytes/set = 8192 bytes)** **These fields contain 32 sets of data describing each of the 32 General Use Pools. Each set of data contains 256 bytes. The following fields define these bytes and are numbered starting with 0. The data for GUP 1 can be found in bytes 320-575; the data for GUP 2 can be found in bytes 576-831, and so forth.**			
0-3	Active Logical Volumes	This 4 byte hexadecimal field contains the number of logical volume images resident in the volume pool at the end of the interval. The number is updated and the reported value is the snapshot of that count when statistics are calculated at the end of the interval. To be considered resident in a pool, the logical volume must be on one of the physical volumes assigned to the pool. Cache resident only volumes, although assigned to the pool, are not included.	Count is updated every 60 minutes.
4-7	Active Data	This 4 byte hexadecimal field contains the number of MiBs of logical volume image data after compression managed in the volume pool. The number is updated dynamically and the reported value is the snapshot of that count when statistics are calculated at the end of the interval. To be considered resident in a pool, the logical volume must be on one of the physical volumes assigned to the pool. Cache resident only volumes, although assigned to the pool, are not included. The value is reported in increments of 1 MiB (1024x1024). Any volume with a count of less than 1 MiB is rounded up to 1 MiB.	Count is updated every 60 minutes.
8-11	Data Written to Pool	This 4 byte hexadecimal field represents the number bytes written to the media associated with this pool during the last interval. This is data pre-migrated from the Tape Volume Cache (TVC), and does not include data moved as part of reclamation. The value is reported in increments of 1 MiB (1024x1024). This value is reset to 0 at the beginning of the interval.	Count is incremented when a logical volume's data is written to a pool.
12-15	Data Read from Pool	This 4 byte hexadecimal field represents the number bytes read from the media associated with this pool during the last interval. This is recall data written to the Tape Volume Cache (TVC), and does not include data moved as part of reclamation. The value is reported in increments of 1 MiB (1024x1024). This value is reset to 0 at the beginning of the interval.	Count is incremented when a logical volume's data is read from a pool.

<table>
<tr><th>Bytes</th><th>Name</th><th>Description</th><th>When Data is Sample/Updated</th></tr>
<tr><td>16</td><td>Device Class</td><td>This 1 byte hexadecimal field indicates the device class identifier for the pool.
<table>
<tr><th>Value</th><th>Description</th></tr>
<tr><td>x00</td><td>No device</td></tr>
<tr><td>x20</td><td>3592 Model J1A. This also includes a 3592-E05 that is emulating a 3592-J1A device.</td></tr>
<tr><td>x22</td><td>3592 Model E05. This is for a 3592-E05 that is behaving as a 3590-E05.</td></tr>
<tr><td>x23</td><td>3592 Model E05 (encryption configured)</td></tr>
<tr><td>x24</td><td>3592 Model E06</td></tr>
<tr><td>x25</td><td>3592 Model E07</td></tr>
<tr><td>x26</td><td>3592 Model E08</td></tr>
<tr><td></td><td>All other values are reserved.</td></tr>
</table>
This is the value at the end of the interval.</td><td></td></tr>
<tr><td>17-18</td><td>Average time since data began expiring on volumes.</td><td>This 2 byte hexadecimal field contains the average age, in days, of the residual data that resides on the stacked volumes assigned to the pool. Statically assigned and borrowed volumes are included in this calculation. A physical volume has residual data on it if it is not full. This value is calculated based on the date a volume transitions to not full and the current date.

This data does not include stacked volumes that are in Read-Only-Recovery or are Unavailable.

The data for this field is reported at the end of the interval.</td><td>Average age is updated every 360 minutes.</td></tr>
<tr><td>19-20</td><td>Maximum time since data began expiring on volumes.</td><td>This 2 byte hexadecimal field contains the maximum age, in days, of the residual data that resides on the stacked volumes assigned to the pool. Statically assigned and borrowed volumes are included in this calculation. A physical volume has residual data on it if it is not full. This value is calculated based on the date a volume transitions to not full and the current date.

This data does not include stacked volumes that are in Read-Only-Recovery or are Unavailable.

The data for this field is reported at the end of the interval.</td><td>Maximum age is updated every 360 minutes.</td></tr>
<tr><td>21-22</td><td>Average Age of Full Private Volumes</td><td>This 2 byte hexadecimal field contains the average age, in days, of private stacked volumes in the pool. Statically assigned and borrowed volumes are included in this calculation. This value is calculated based when the volume is marked as full until it is reclaimed.

This data does not include stacked volumes that are in Read-Only-Recovery or are Unavailable.

The data for this field is reported at the end of the interval.</td><td>Average age is updated every 360 minutes.</td></tr>
</table>

Bytes	Name	Description	When Data is Sample/Updated
23-24	Maximum Age of Full Private Volumes	This 2 byte hexadecimal field contains the maximum age, in days, of private stacked volumes in the pool. Statically assigned and borrowed volumes are included in this calculation. This value is calculated based when the volume is marked as full until it is reclaimed. This data does not include stacked volumes that are in Read-Only-Recovery or are Unavailable. The data for this field is reported at the end of the interval.	Maximum age is updated every 360 minutes.
25-31	Reserved	All bytes set to x00.	

Library - Pooling – GUP - Media Container

Relative bytes 32-223 (8 sets of data x 24 bytes/set = 192 bytes)
These bytes are relative to the Library - Pooling – GUP Container.

These fields contain 8 sets of data describing up to 8 physical media types in the pool. The first media type is referred to as Media Type 0; the second is referred to as Media Type 1 and so forth. Each set of data contains 24 bytes. The following fields define these bytes and are numbered starting with 0. The data for media type 0 can be found in relative bytes 32-55; the data for media type 1 can be found in relative bytes 56-79, and so forth. All 8 sets will be populated starting with the first media type that has a non-zero Physical Media Identifier. Only media types with a non-zero Physical Media Identifier will occupy a set, then the remainder of the 8 sets will be have its Physical Media Identifier set to no media. If only 3 media types are used in this pool, then the first 3 sets will be populated, with no unused sets. The remaining 5 sets will have their Physical Media Types set to zero, indicating they are not used. The first Physical Media Identifier that is zero indicates that there are no more used Physical Media types in the subsequent sets

<table>
<tr><td>0</td><td>Physical Media Identifiers</td><td>This 1 byte hexadecimal field contains the identifier for the media type associated with following general use pool (GUP) volume counts. The following are values for this field:
<table>
<tr><th>Value</th><th>Description</th></tr>
<tr><td>x00</td><td>No media type for this set of data</td></tr>
<tr><td>x10</td><td>3590 J media (reserved)</td></tr>
<tr><td>x11</td><td>3590 K media (reserved)</td></tr>
<tr><td>x20</td><td>3592 JA media</td></tr>
<tr><td>x21</td><td>3592 JW media (reserved for future use)</td></tr>
<tr><td>x22</td><td>3592 JJ media</td></tr>
<tr><td>x23</td><td>3592 JR media (reserved for future use)</td></tr>
<tr><td>x24</td><td>3592 JB media</td></tr>
<tr><td>x25</td><td>3592 JX media (reserved for future use)</td></tr>
<tr><td>x26</td><td>3592 JC Media</td></tr>
<tr><td>x27</td><td>3592 JY Media (reserved for future use)</td></tr>
<tr><td>x28</td><td>3592 JK Media</td></tr>
<tr><td>x29</td><td>3592 JD Media</td></tr>
<tr><td>x2A</td><td>3592 JZ Media (reserved for future use)</td></tr>
<tr><td>x2B</td><td>3592 JL Media</td></tr>
<tr><td></td><td>All other values are reserved.</td></tr>
</table>
This is the value at the end of the interval.</td><td></td></tr>
</table>

Bytes	Name	Description	When Data is Sample/Updated
1-2	Scratch Volume Count	This 2 byte hexadecimal field indicates the number of scratch stacked volumes associated with this pool of the media type indicated. The count includes volumes statically assigned to the pool at the end of the interval. It does not include any volumes that are borrowed from the CSP or are waiting to be erased due to a Security Data Erase. This data does not include stacked volumes that are in Read-Only-Recovery or are Unavailable. This is the value at the end of the interval.	Count is updated every 60 minutes.
3-4	Private Volume Count	This 2 byte hexadecimal field indicates the number of private stacked volumes associated with this pool of the media type indicated. The count includes volumes statically assigned to the pool at the end of the interval. It does not include any volumes that are borrowed from the CSP or are waiting to be erased due to a Security Data Erase. This data does not include stacked volumes that are in Read-Only-Recovery or are Unavailable. This is the value at the end of the interval.	Count is updated every 60 minutes.
5-6	Waiting for Security Data Erase	This 2 byte hexadecimal field indicates the number of stacked volumes associated with this pool of the media type indicated that are waiting for Security Data Erase. The count includes volumes statically assigned to the pool and any volumes borrowed from the CSP. This data does not include stacked volumes that are in Read-Only-Recovery or are Unavailable. This is the value at the end of the interval.	Count is updated every 60 minutes.
7-8	Borrowed Scratch Volume Count	This 2 byte hexadecimal field indicates the number of stacked volumes associated with this pool of the media type indicated that are in scratch status that have been borrowed from the CSP. This data does not include stacked volumes that are in Read-Only-Recovery or are Unavailable. This is the value at the end of the interval.	Count is updated every 60 minutes.
9-10	Borrowed Private Volume Count	This 2 byte hexadecimal field indicates the number of stacked volumes associated with this pool of the media type indicated that are in private status that have been borrowed from the CSP. This data does not include stacked volumes that are in Read-Only-Recovery or are Unavailable. This is the value at the end of the interval.	Count is updated every 60 minutes.
11-12	Read Only Recovery Volume Count	This 2 byte hexadecimal field indicates the number of stacked volumes associated with this pool of the media type indicated that are in read only recovery status. The count includes stacked volumes that are both statically assigned to the pool and are borrowed stacked volumes. This is the value at the end of the interval.	Count is updated every 60 minutes.

Bytes	Name	Description	When Data is Sample/Updated
13-14	Unavailable Volume Count	This 2 byte hexadecimal field indicates the number of stacked volumes associated with this pool of the media type indicated that are in unavailable status. The count includes stacked volumes that are both statically assigned to the pool and are borrowed stacked volumes. This is the value at the end of the interval.	Count is updated every 60 minutes.
15	Media type container count	This one byte hexadecimal field contains the number of physical media types that have a non-zero Physical Media Count in this pool. If more than 8 physical media types are non-zero, only the first 8 will be returned.	
16-23	Reserved	All bytes set to x00.	

Pooling – GUP - Reclaim Container

Relative bytes 224-239 (1 set of data x 16 bytes/set = 16 bytes)
These bytes are relative to the Library - Pooling – GUP Container.

These fields contain Reclaim information for this GUP.

224	Reclaim Threshold	This 1 byte hexadecimal field contains the reclaim threshold percentage for the pool as defined at the end of the interval.	Threshold is updated when its value changes.
225	Reclaim Pool	This 1 byte hexadecimal field contains the reclaim pool for the pool as defined at the end of the interval.	Pool is updated when its value changes.
226-227	Last Access Policy	This 2 byte hexadecimal field indicates, in days, when a physical volume is eligible for reclaim based on last access. A volume is eligible for reclaim when the number of days specified has elapsed since any data on the volume has been accessed because of a recall. If this field contains a value of 0, it is not used as criteria for reclaim. This is the value at the end of the interval.	Policy is updated when its value changes.
228-229	Last Written Policy	This 2 byte hexadecimal field indicates, in days, when a physical volume is eligible for reclaim based on when it was last written to. A volume is eligible for reclaim when the number of days specified has elapsed since any data has been written to the volume. If this field contains a value of 0, it is not used as criteria for reclaim. This is the value at the end of the interval.	Policy is updated when its value changes.
230-231	Last Data Invalidation Policy	This 2 byte hexadecimal field indicates, in days, when a physical volume is eligible for reclaim based on when data was last invalidated on it. A volume is eligible for reclaim when the number of days specified has elapsed since any data has invalidated on the volume. If this field contains a value of 0, it is not used as criteria for reclaim. This is the value at the end of the interval.	Policy is updated when its value changes.
232	Minimum Active Data Percentage Policy	This 1 byte hexadecimal field indicates the minimum active data percentage a physical volume's active data must fall below before it can be reclaimed using the days since last data invalidation reclamation policy. This is the value at the end of the interval.	Policy is updated when its value changes.
233-234	Force Erasure Policy	This 2 byte hexadecimal field indicates the number of days before a physical volume must complete the erase process. The time starts when the first data is invalidated on the volume. Supported values are 0 to 365. If this field contains a value of 0, it is not used as criteria for reclaim. When this field contains a non-zero value, the pool is operating in the secure data erasure mode. Any volume that is reclaimed in the pool is physically erased before being returned to scratch. This is the value at the end of the interval.	Policy is updated when its value changes.

<table>
<tr><th>Bytes</th><th>Name</th><th>Description</th><th>When Data is Sample/Updated</th></tr>
<tr><td>235</td><td>Sunset Media Reclaim Threshold</td><td>This 1 byte hexadecimal field contains the reclaim threshold percentage for sunset media in the pool as defined at the end of the interval.</td><td>Threshold is updated when its value changes.</td></tr>
<tr><td>236-239</td><td>Reserved</td><td>All bytes set to x00.</td><td></td></tr>
<tr><td colspan="3"><h2>Pooling – GUP - Properties Container</h2><b>Relative bytes 240-255 (1 set of data x 16 bytes/set = 16 bytes)
These bytes are relative to the Library - Pooling – GUP Container.</b>

<b>These fields contain pool property information for this GUP.</b></td><td></td></tr>
<tr><td>240</td><td>Pool Type</td><td>This 1 byte hexadecimal field indicates the type of the pool. The valid bit masks are as follows:<table><tr><th>Value</th><th>Description</th></tr><tr><td>x80</td><td>Reserved</td></tr><tr><td>x40</td><td>Reserved</td></tr><tr><td>x20</td><td>Export Pool</td></tr><tr><td></td><td>All other values are reserved.</td></tr></table>This is the value at the end of the interval.</td><td>Pool type is updated when its value changes.</td></tr>
<tr><td>241</td><td>Return Borrowed Volumes</td><td>This 1 byte hexadecimal field indicates whether volumes borrowed from the CSP should be returned to the CSP after they are no longer needed by the pool. The valid values are:<table><tr><th>Value</th><th>Description</th></tr><tr><td>x00</td><td>Don’t return borrowed volumes</td></tr><tr><td>x01</td><td>Return borrowed volumes</td></tr><tr><td></td><td>All other values are reserved.</td></tr></table>This is the value at the end of the interval.</td><td>Policy is updated when its value changes.</td></tr>
</table>

<table>
<tr><th>Bytes</th><th>Name</th><th>Description</th><th>When Data is Sample/Updated</th></tr>
<tr><td>242-245</td><td>First Media Types to Borrow</td><td>This 4 byte bit-mapped field indicates the first media types to be borrowed from the CSP if additional scratch physical volumes are needed by the pool. When a bit is set to one (1) then the media type defined by the corresponding Library-Pooling-GUP-Media – Physical Media Identifier set can be used as one of the first media types to borrow from the CSP. When a bit is set to zero (0) the corresponding physical media type, if any, is not one of the first media types to borrow.

For example, if the first Physical Media Identifier in the Library-Pooling-GUP-Media container had indicated J media (now reserved). The second Media Indicator indicates K media (also now reserved). The third indicates JA media, and the fourth indicates JW media, and so forth. A value of x20000000 in this field would indicate the media type identified in the third set of 24 bytes should be borrowed first. In this example JA media will be borrowed first.

A value of x00 indicates borrowing is turned off for this pool.
<table>
<tr><th>Value</th><th>Description</th></tr>
<tr><td>x80000000</td><td>The first media type defined by the Physical Media Identifier. In previous levels this indicated J media, this is now reserved.</td></tr>
<tr><td>x40000000</td><td>The second media type defined by the Physical Media Identifier. In previous levels this indicated K media, this is now reserved.</td></tr>
<tr><td>x20000000</td><td>The third media type (JA) defined by the Physical Media Identifier can be borrowed as a first choice.</td></tr>
<tr><td>x10000000</td><td>The fourth media type (JW) defined by the Physical Media Identifier can be borrowed as a first choice.</td></tr>
<tr><td>x08000000</td><td>The fifth media type (JJ) defined by the Physical Media Identifier can be borrowed as a first choice.</td></tr>
<tr><td>x04000000</td><td>The sixth media type (JR) defined by the Physical Media Identifier can be borrowed as a first choice.</td></tr>
<tr><td>x02000000</td><td>The seventh media type (JB) defined by the Physical Media Identifier can be borrowed as a first choice.</td></tr>
<tr><td>x01000000</td><td>The eighth media type (JX) defined by the Physical Media Identifier can be borrowed as a first choice.</td></tr>
<tr><td>x00800000</td><td>The ninth media type (JC) defined by the Physical Media Identifier can be borrowed as a first choice.</td></tr>
<tr><td>x00400000</td><td>The tenth media type (JY) defined by the Physical Media Identifier can be borrowed as a first choice.</td></tr>
<tr><td>x00200000</td><td>The eleventh media type (JK) defined by the Physical Media Identifier can be borrowed as a first choice.</td></tr>
<tr><td>x00100000</td><td>The twelfth media type (JD) defined by the Physical Media Identifier can be borrowed as a first choice.</td></tr>
<tr><td>x00080000</td><td>The thirteenth media type (JZ) defined by the Physical Media Identifier can be borrowed as a first choice.</td></tr>
<tr><td>x00040000</td><td>The fourteenth media type (JL) defined by the Physical Media Identifier can be borrowed as a first choice.</td></tr>
<tr><td></td><td>All other values are reserved</td></tr>
</table>
This is the value at the end of the interval.</td><td>Media types is updated when its value changes.</td></tr>
</table>

<table>
<tr><th>Bytes</th><th>Name</th><th>Description</th><th>When Data is Sample/Updated</th></tr>
<tr><td>246-249</td><td>Second Media Types to Borrow</td><td>This 4 byte bit-mapped field indicates the second media types to be borrowed from the CSP if additional scratch physical volumes are needed by the pool. When a bit is set to one (1) then the media type defined by the corresponding Library-Pooling-GUP-Media – Physical Media Identifier set can be used as one of the second media types to borrow from the CSP. When a bit is set to zero (0) the corresponding physical media type, if any, is not one of the second media types to borrow.

For example, if the first Physical Media Identifier in the Library-Pooling-GUP-Media container indicates J media (now reserved), the second indicates K media (also now reserved). The third indicates JA media, and the fourth indicates JW media. A value of x08000000 in this field would indicate the media type JJ will be borrowed second.

A value of x00 indicates a second media type to borrow is not specified.
<table>
<tr><th>Value</th><th>Description</th></tr>
<tr><td>x80000000</td><td>The first media type defined by the Physical Media Identifier. In previous levels this indicated J media, this is now reserved.</td></tr>
<tr><td>x40000000</td><td>The second media type defined by the Physical Media Identifier. In previous levels this indicated K media, this is now reserved.</td></tr>
<tr><td>x20000000</td><td>The third media type (JA) defined by the Physical Media Identifier can be borrowed as a second choice.</td></tr>
<tr><td>x10000000</td><td>The fourth media type (JW) defined by the Physical Media Identifier can be borrowed as a second choice.</td></tr>
<tr><td>x08000000</td><td>The fifth media type (JJ) defined by the Physical Media Identifier can be borrowed as a second choice.</td></tr>
<tr><td>x04000000</td><td>The sixth media type (JR) defined by the Physical Media Identifier can be borrowed as a second choice.</td></tr>
<tr><td>x02000000</td><td>The seventh media type (JB) defined by the Physical Media Identifier can be borrowed as a second choice.</td></tr>
<tr><td>x01000000</td><td>The eighth media type (JX) defined by the Physical Media Identifier can be borrowed as a second choice.</td></tr>
<tr><td>x00800000</td><td>The ninth media type (JC) defined by the Physical Media Identifier can be borrowed as a second choice.</td></tr>
<tr><td>x00400000</td><td>The tenth media type (JY) defined by the Physical Media Identifier can be borrowed as a second choice.</td></tr>
<tr><td>x00200000</td><td>The eleventh media type (JK) defined by the Physical Media Identifier can be borrowed as a second choice.</td></tr>
<tr><td>x00100000</td><td>The twelfth media type (JD) defined by the Physical Media Identifier can be borrowed as a second choice.</td></tr>
<tr><td>x00080000</td><td>The thirteenth media type (JZ) defined by the Physical Media Identifier can be borrowed as a second choice.</td></tr>
<tr><td>x00040000</td><td>The fourteenth media type (JL) defined by the Physical Media Identifier can be borrowed as a second choice.</td></tr>
<tr><td></td><td>All other values are reserved</td></tr>
</table>
This is the value at the end of the interval.</td><td>Media types is updated when its value changes..</td></tr>
<tr><td>250-255</td><td>Reserved</td><td>All bytes set to x00.</td><td></td></tr>
</table>

<table>
<tr><th>Bytes</th><th>Name</th><th>Description</th><th>When Data is Sample/Updated</th></tr>
<tr><td colspan="3"><b>Library – Physical Device Container (for future use)</b>

<b>Bytes 8512-10047 (32 sets of data x 48 bytes/set = 1536 bytes)</b>

<b>These fields are currently filled by zero except for “Length” field. The container specification would be changed in a later release.</b></td><td></td></tr>
<tr><td>0-1</td><td>Length</td><td>This 2 byte hexadecimal field contains the length of this container. The length includes these 2 bytes.</td><td></td></tr>
<tr><td>2</td><td>Device Class ID</td><td>This one byte hexadecimal field contains this device’s device class identifier.
<table>
<tr><th>Value</th><th>Description</th></tr>
<tr><td>x00</td><td>No device installed</td></tr>
<tr><td>x20</td><td>3592 Model J1A. This also includes a 3592-E05 that is emulating a 3592-J1A device.</td></tr>
<tr><td>x22</td><td>3592 Model E05. This is for a 3592-E05 that is behaving as a 3590-E05.</td></tr>
<tr><td>x23</td><td>3592 Model E05 (encryption configured)</td></tr>
<tr><td>x24</td><td>3592 Model E06</td></tr>
<tr><td>x25</td><td>3592 Model E07</td></tr>
<tr><td>x26</td><td>3592 Model E08</td></tr>
<tr><td></td><td>All other values are reserved.</td></tr>
</table>
This is the value at the end of the interval.</td><td></td></tr>
<tr><td>3-14</td><td>Device Serial Number</td><td>This 12 byte ASCII field contains the serial number of the physical tape drive. This field is left justified and blank filled.</td><td></td></tr>
<tr><td>15-18</td><td>Data Read From Device</td><td>This 4 byte hexadecimal field indicates the number of bytes transferred from the physical device during this interval. The value is reported in increments of 1MiB (1024 x 1024). Any residual data will cause the value to be rounded up to the next higher value.

This value is reset to 0 at the beginning of the interval.</td><td></td></tr>
<tr><td>19-22</td><td>Data Written to Device</td><td>This 4 byte hexadecimal field indicates the number of bytes transferred to the physical device during this interval. The value is reported in increments of 1MiB (1024 x 1024). Any residual data will cause the value to be rounded up to the next higher value.

This value is reset to 0 at the beginning of the interval.</td><td></td></tr>
<tr><td>23-26</td><td>Maximum Read Throughput</td><td>This 4 byte hexadecimal field contains the maximum read throughput recorded during this interval. The value is expressed in KiB/Sec (1 KiB = 1024 bytes). This is the value at the end of the interval.</td><td></td></tr>
<tr><td>27-30</td><td>Average Read Throughput</td><td>This 4 byte hexadecimal field contains the average read throughput recorded during this interval. The value is expressed in KiB/Sec (1 KiB = 1024 bytes). This is the value at the end of the interval.</td><td></td></tr>
<tr><td>31-34</td><td>Maximum Write Throughput</td><td>This 4 byte hexadecimal field contains the maximum write throughput recorded during this interval. The value is expressed in KiB/Sec (1 KiB = 1024 bytes). This is the value at the end of the interval.</td><td></td></tr>
<tr><td>35-38</td><td>Average Write Throughput</td><td>This 4 byte hexadecimal field contains the average write throughput recorded during this interval. The value is expressed in KiB/Sec (1 KiB = 1024 bytes). This is the value at the end of the interval.</td><td></td></tr>
<tr><td>39-47</td><td>Reserved</td><td>All bytes set to x00.</td><td></td></tr>
</table>

Hnode Grid Historical Record

This Hnode historical record has the following nested structure:

- Header
- Grid Container
    - Grid-Cluster 0 Container
    - Grid-Cluster 1 Container (If installed)
    - Grid-Cluster 2 Container (If installed)
    - Grid-Cluster 3 Container (If installed)
    - Grid-Cluster 4 Container (If installed)
    - Grid-Cluster 5 Container (If installed)
    - Grid-Cluster 6 Container (If installed)
    - Grid-Cluster 7 Container (If installed)
- Extended Grid Container

Bytes	**Name**	**Description**	**When Data is Sampled/Updated**
0-1	Length	This 2 byte hexadecimal field contains the length of this record. The length includes these 2 bytes.	
2	Version	This 1 byte hexadecimal field contains the version of the data presented in this record. The current version is set to x05.	
3	Data Type	This 1 byte hexadecimal field indicates the type of data contained in this record. For this record the value is set to x33 indicating this is an Hnode Grid Historical record.	
4	Node ID	This 1 byte hexadecimal field indicates the Hnode ID which this interval's data represents. Valid values are x00 – x01.	
5	Cluster ID	This 1 byte hexadecimal field indicates the Cluster ID which this Hnode is a part of. Valid values are x00 – x07.	
6-7	Interval Duration	This 2 byte hexadecimal field indicates the interval in seconds that this interval's data was taken over.	
8-11	Time Stamp	This 4 byte hexadecimal field indicates the end time of the interval this data was taken over. This value is the time in seconds since the Epoch (00:00:00 UTC, January 1, 1970)	
12-15	Machine Type	This 4 byte EBCDIC field contains this node's machine type. The field is left justified padded with EBCDIC blanks. Initially this field will be set to "3957".	
16-18	Machine Model	This 3 byte EBCDIC field contains this node's machine model. The field is left justified padded with EBCDIC blanks. Initially this field will be set to "V06".	
19-26	Machine Serial Number	This 8 character EBCDIC field contains the serial number of this node. This field is left justified and padded with EBCDIC blanks. The format is XX-YYYYY where XX is the plant of manufacture and the YYYYY is the sequence number of the node's machine. The dash character (-) is fixed.	

Bytes	Name	Description	When Data is Sampled/Updated
27-34	Code Level	This 8 byte hexadecimal field contains the code level of the TS7700. The 8 bytes are actually four 2-byte fields. Each 2-byte field represents a portion of the code level. The VE code level is expressed as Version.Release.Modification.Fix in a decimal form. For example the code level of 8.0.0.104 would be represented in the 8 bytes as: x0008000000000068.	
35-39	Grid Library Sequence Number	This 5 character EBCDIC field contains the Library Sequence Number of the Grid (Composite) library.	
40-44	Distributed Library Sequence Number	This 5 character EBCDIC field contains the Distributed Library Sequence Number for this Distributed Library ID	
45-63	Reserved	All bytes set to x00	
Grid Container **The data from each Hnode will be the same within a Cluster.**			
64-67	Logical Volumes for Copy	This 4 byte hexadecimal field indicates the number of logical volumes that are scheduled to be copied to this Cluster. This is the value at the end of the interval.	Count is updated every 5 minutes.
68-71	Data to Copy	This 4 byte hexadecimal field indicates the amount of data that is scheduled to be copied to this Cluster. This represents the amount of data contained in the logical volumes that are scheduled to be copied. The value is reported in increments of 1 MiB (1024x1024). Any residual data is rounded up to 1 MiB. This is the value at the end of the interval.	Count is updated every 5 minutes.
72-75	Average Deferred Queue Age	This 4 byte hexadecimal field indicates the average age, in seconds, of the logical volumes in the deferred copy queue destined to be copied to this Cluster. This is the value at the end of the interval.	Age is updated every 5 minutes.
76-79	Average Immediate Queue Age	This 4 byte hexadecimal field indicates the average age, in seconds, of the logical volumes in the immediate copy queue destined to be copied to this Cluster. This is the value at the end of the interval.	Age is updated every 5 minutes.
80	Number of Clusters	This 1 byte hexadecimal field indicates the number of Clusters in the Grid. This field can be used to determine how many Grid-Cluster containers will be attached to this record. There is a maximum of 8 Clusters. This is the value at the end of the interval.	
81-84	Average Time delayed copy queue Age	This 4 byte hexadecimal field indicates the average age, in seconds, of the logical volumes in the timed delay state that are in the copy queue. Logical volumes in the timed delay state are not yet eligible for the actual copy until their defined time-delays are expired.	
85-88	Overall Object Data Transferred into Cache from DS8Ks	A 4-byte field indicating the number of bytes transferred to the Cluster this record is for from all of the DS8K connected to this Cluster as a part of DS8K offload function. The value is reported in increments of 1 MiB (1024x1024). Any residual data is rounded up to 1 MiB. This value is reset to 0 at the beginning of the interval.	

Bytes	Name	Description	When Data is Sampled/Updated
89-92	Overall Object Data Transferred from Cache to DS8Ks	A 4-byte field indicating the number of bytes transferred from the Cluster this record is for to all of the DS8K connected to this Cluster as a part of DS8K offload function. The value is reported in increments of 1 MiB (1024x1024). Any residual data is rounded up to 1 MiB. This value is reset to 0 at the beginning of the interval.	
93	Packet Retransmission Rate	A 1-byte field indicating the percentage of packets retransmission over the packets sent.	
94-95	Reserved	All bytes set to x00.	
Grid-Cluster Container **Bytes 96 and up (Number of Clusters x 256 bytes/set). For example, if there are 3 Clusters in the Grid there will be 3 sets of data. There is a maximum of 8 Clusters.** **This next segment of the record contains one set of data for each Cluster in the Grid as defined in byte 80 above. Each set of data contains 256 bytes. The data for the first Cluster (Cluster 0) can be found in bytes 96-351; the second Cluster's (Cluster 1) data can be found in bytes 352-447, and so forth.**			
0-3	Data Transferred into a Cluster's Cache from other Clusters as part of a Copy Operation	A 4-byte field indicating the number of bytes transferred to the Cluster this record is for from another Cluster this container is for as part of a copy operation (immediate, deferred) during this interval. The Cluster transferred from is determined by the relative position of the container segment within the record. A value of x00000000 is indicated for a Cluster's own value. The value is reported in increments of 1 MiB (1024x1024). Any residual data is rounded up to 1 MiB. This value is reset to 0 at the beginning of the interval. For a performance reason this field does not have a valid value (filled by zero).	Count is incremented every time a block of data is transferred to this cluster.
4-31	Reserved	All bytes set to x00.	

Bytes	Name	Description	When Data is Sampled/Updated
32-35	Data Transferred From a Cluster's Cache To Other Clusters as part of a Copy Operation	A 4-byte field indicating the number of bytes transferred from the Cluster this record is for to another Cluster this container is for as part of a copy operation (immediate, deferred) during this interval. A field indicating data transfer from this Cluster's cache to this Cluster's cache shows data size transferred from this Cluster's cache through GGM copy activity if the Cluster is used as a GGM copy source. The Cluster transferred to is determined by the relative position of the container segment within the record. A value of x00000000 is indicated for a Cluster's own value except GGM copy activity. The value is reported in increments of 1 MiB (1024x1024). Any residual data is rounded up to 1 MiB. This value is reset to 0 at the beginning of the interval.	Count is incremented every time a block of data is transferred.
36-63	Reserved	All bytes set to x00.	
64-67	Logical Mounts Directed to other Clusters	A 4-byte field indicating the number of logical mounts from all Vnodes in this Cluster which were satisfied by accessing another Cluster. The Cluster receiving the mounts is determined by the relative position of the container segment within the record. A value of x00000000 is indicated for a Cluster's own value. A logical mount is counted when the mount is completed. This value is reset to 0 at the beginning of the interval.	Count is incremented when a logical mount is completed.
68-95	Reserved	All bytes set to x00.	
96-99	Data Transferred into a Cluster's Cache from other Clusters as part of a Remote Write Operation	A 4-byte field indicating the number of bytes transferred to the Cluster this record is for from another Cluster this container is for as part of a remote write operation including sync mode copy during this interval. A sync mode copy into this cluster from another cluster is considered a remote mount for write and is thus included in this count. The source Cluster is determined by the relative position of the container segment within the record. A value of x00000000 is indicated for a Cluster's own value. The value is reported in increments of 1 MiB (1024x1024). Any residual data is rounded up to 1 MiB. This value is reset to 0 at the beginning of the interval.	Count is incremented every time a block of data is transferred.
100-127	Reserved	All bytes set to x00.	

Bytes	Name	Description	When Data is Sampled/Updated
128-131	Data Transferred from a Cluster's Cache To Other Clusters as part of a Remote Read operation	A 4-byte field indicating the number of bytes transferred from the Cluster this record is for to another Cluster this container is for as part of a remote read operation during this interval. The Cluster doing the remote read is determined by the relative position of the container segment within the record. A value of x00000000 is indicated for a Cluster's own value. The value is reported in increments of 1 MiB (1024x1024). Any residual data is rounded up to 1 MiB. This value is reset to 0 at the beginning of the interval.	Count is incremented every time a block of data is transferred.
132-159	Reserved	All bytes set to x00	
160-163	Data Transferred into a cluster's Cache from other clusters as part of an Immediate copy operation	A 4-byte field indicating the total number of bytes transferred to the cluster's cache from other clusters as part of an immediate copy operation during this interval. The cluster this container is for is determined by the relative position of the container segment within the record. This field has a valid value only if the cluster this container is for is the same one as that this record is for. The value is reported in increments of 1 MiB (1024x1024). Any residual data is rounded up to 1 MiB. This value is reset to 0 at the beginning of the interval.	Count is incremented when a copy of logical volume is completed.
164-165	Number of immediate copies that have completed	A 2-byte field indicating the number of immediate copies that completed which transferred data to the cluster's cache from another cluster during this interval. The cluster this container is for is determined by the relative position of the container segment within the record. This field has a valid value only if the cluster this container is for is the same one as that this record is for. This value is reset to 0 at the beginning of the interval.	Count is incremented when a copy of logical volume is completed.
166-169	Data Transferred into a cluster's Cache from Other clusters as part of a deferred copy operation	A 4-byte field indicating the total number of bytes transferred to the cluster from other clusters as part of a deferred copy operation during this interval. The cluster this container is for is determined by the relative position of the container segment within the record. This field has a valid value only if the cluster this container is for is the same one as that this record is for. The value is reported in increments of 1 MiB (1024x1024). Any residual data is rounded up to 1 MiB. This value is reset to 0 at the beginning of the interval.	Count is incremented when a copy of logical volume is completed.

Bytes	Name	Description	When Data is Sampled/Updated
170-171	Number of deferred copies that have completed	A 2-byte field indicating the number of deferred copies that transferred data to the cluster's cache from another cluster during this interval. The cluster this container is for is determined by the relative position of the container segment within the record. This field has a valid value only if the cluster this container is for is the same one as that this record is for. This value is reset to 0 at the beginning of the interval.	Count is incremented when a copy of logical volume is completed.
172-175	Data Transferred into a cluster's Cache from Other clusters as part of a sync mode copy operation	A 4-byte field indicating the number of bytes transferred to this cluster's cache from another cluster as part of a sync mode copy operation during this interval. The cluster transferred from is determined by the relative position of the container segment within the record. A value of x00000000 is indicated for a cluster's own value. The value is reported in increments of 1 MiB (1024x1024). Any residual data is rounded up to 1 MiB. This value is reset to 0 at the beginning of the interval. This field is always 0.	Count is incremented every time a block of data is transferred.
176-177	Number of sync mode copies that have completed	A 2-byte field indicating the number of sync mode copies that transferred data to this cluster's cache from another cluster during this interval. The cluster transferred from is determined by the relative position of the container segment within the record. A value of x0000 is indicated for a cluster's own value. This value is reset to 0 at the beginning of the interval. This field is always 0.	Count is incremented when a copy is completed.
178-255	Reserved	All bytes set to x00.	
Extended Grid Container			
0-1	Length	This 2 byte hexadecimal field contains the length of this container. The length includes these 2 bytes.	
2-5	Longest Time Delayed Copy Queue Age	This 4 byte hexadecimal field indicates the longest age in seconds that copies in the timed delay state are in the copy queue. Logical volumes in the timed delay state are not yet eligible for the actual copy until their defined time-delays are expired.	
6-9	Longest Family Deferred Copy Queue Age	This 4 byte hexadecimal field indicates the longest age in seconds that copies in the family deferred state are in the copy queue.	
10-13	Longest Copy Queue Age	This 4 byte hexadecimal field indicates the longest age in seconds that copies are in the copy queue.	
14-63	Reserved	All bytes set to x00.	

Hnode Cloud Historical Record

This Hnode historical record is generated if cloud is enabled on the cluster.
This Hnode historical record has the following nested structure:

- Header
- Tier Container
  - Pool 0 Container (if exists)
  - Pool 1 Container (if exists)
  - ...
  - Pool 15 Container (if exists)

Bytes	Name	Description	When Data is Sampled/Updated
0-1	Length	This 2 byte hexadecimal field contains the length of this record. The length includes these 2 bytes.	
2	Version	This 1 byte hexadecimal field contains the version of the data presented in this record. The current version is set to x01.	
3	Data Type	This 1 byte hexadecimal field indicates the type of data contained in this record. For this record the value is set to x35 indicating this is an Hnode Cloud Historical record.	
4	Node ID	This 1 byte hexadecimal field indicates the Hnode ID which this interval's data represents. Valid values are x00 – x01.	
5	Cluster ID	This 1 byte hexadecimal field indicates the Cluster ID which this Hnode is a part of. Valid values are x00 – x07.	
6-7	Interval Duration	This 2 byte hexadecimal field indicates the interval in seconds that this interval's data was taken over.	
8-11	Time Stamp	This 4 byte hexadecimal field indicates the end time of the interval this data was taken over. This value is the time in seconds since the Epoch (00:00:00 UTC, January 1, 1970)	
12-15	Machine Type	This 4 byte EBCDIC field contains this node's machine type. The field is left justified padded with EBCDIC blanks. Initially this field will be set to "3957".	
16-18	Machine Model	This 3 byte EBCDIC field contains this node's machine model. The field is left justified padded with EBCDIC blanks. Initially this field will be set to "V06".	
19-26	Machine Serial Number	This 8 character EBCDIC field contains the serial number of this node. This field is left justified and padded with EBCDIC blanks. The format is XX-YYYYY where XX is the plant of manufacture and the YYYYY is the sequence number of the node's machine. The dash character (-) is fixed.	
27-34	Code Level	This 8 byte hexadecimal field contains the code level of the TS7700. The 8 bytes are actually four 2-byte fields. Each 2-byte field represents a portion of the code level. The VE code level is expressed as Version.Release.Modification.Fix in a decimal form. For example the code level of 8.0.0.104 would be represented in the 8 bytes as: x0008000000000068.	
35-39	Grid Library Sequence Number	This 5 character EBCDIC field contains the Library Sequence Number of the Grid (Composite) library.	

<table>
<tr><th>Bytes</th><th>Name</th><th>Description</th><th>When Data is Sampled/Updated</th></tr>
<tr><td>40-44</td><td>Distributed Library Sequence Number</td><td>This 5 character EBCDIC field contains the Distributed Library Sequence Number for this Distributed Library ID</td><td></td></tr>
<tr><td>45-63</td><td>Reserved</td><td>All bytes set to x00</td><td></td></tr>
<tr><td colspan="3"><h2>Tier Container</h2><b>The data from each Hnode will be the same within a Cluster.</b></td><td></td></tr>
<tr><td>0-3</td><td>Length</td><td>This 4 byte hexadecimal field contains the length of this container. The length includes these 2 bytes.</td><td></td></tr>
<tr><td>4-5</td><td>Number of Cloud Pools</td><td>This 2 byte hexadecimal field contains the number of cloud pools currently defined. This field can be used to determine how many Cloud Pool Containers are attached to this container.</td><td></td></tr>
<tr><td>6-63</td><td>Reserved</td><td>All bytes set to x00.</td><td></td></tr>
<tr><td colspan="3"><h2>Pool X Container</h2><b>The data from each Hnode will be the same within a Cluster. Up to 16 pool containers will be recorded.</b></td><td></td></tr>
<tr><td>0-1</td><td>Length</td><td>This 2 byte hexadecimal field contains the length of this container. The length includes these 2 bytes.</td><td></td></tr>
<tr><td>2-20</td><td>ID</td><td>This EBCDIC field contains the ID of the cloud pool. The field is left justified and padded with EBCDIC blanks. This is the value at the end of the interval.</td><td></td></tr>
<tr><td>21-28</td><td>Nickname</td><td>This EBCDIC field contains the nickname of the cloud pool. The field is left justified and padded with EBCDIC blanks. This is the value at the end of the interval.</td><td></td></tr>
<tr><td>29-36</td><td>Number of Objects</td><td>This 8 byte hexadecimal field contains the number of latest version lvols in the cloud pool.</td><td></td></tr>
<tr><td>37-44</td><td>Total Size of Objects</td><td>This 8 byte hexadecimal field contains the total size of latest version lvols in the cloud pool in GiB.</td><td></td></tr>
<tr><td>45-52</td><td>Number of Retained Objects</td><td>This 8 byte hexadecimal field contains the number of retained lvols in the cloud pool. Retained lvols mean the lvols which are older versions but not deleted yet because volume version retention is enabled by setting retention duration to the cloud pool. This number excludes the latest version lvols in the cloud pool.</td><td></td></tr>
<tr><td>53-60</td><td>Total Size of Retained Objects</td><td>This 8 byte hexadecimal field contains the total size of retained lvols in the cloud pool in GiB. Retained lvols mean the lvols which are older versions but not deleted yet because volume version retention is enabled by setting retention duration to the cloud pool. This size excludes the latest version lvols in the cloud pool.</td><td></td></tr>
<tr><td>61</td><td>Retention Type</td><td>This field indicates how the volume version is retained in the pool.
<table>
<tr><th>Value</th><th>Description</th></tr>
<tr><td>x00</td><td>Volume version retention is disabled.</td></tr>
<tr><td>x01</td><td>The number of days to retain volume versions is specified.</td></tr>
<tr><td></td><td>All other values are reserved</td></tr>
</table>
This is the value at the end of the interval.</td><td></td></tr>
<tr><td>62-65</td><td>Retention Duration</td><td>The number of days to retain versions of data.</td><td></td></tr>
</table>

<table>
<tr><th>Bytes</th><th>Name</th><th>Description</th><th>When Data is Sampled/Updated</th></tr>
<tr><td>66</td><td>State</td><td>This field indicates the access status of the pool.
<table>
<tr><th>Value</th><th>Description</th></tr>
<tr><td>x00</td><td>READ-WRITE.</td></tr>
<tr><td>x01</td><td>READ-ONLY</td></tr>
<tr><td></td><td>All other values are reserved</td></tr>
</table>
This is the value at the end of the interval.</td><td></td></tr>
<tr><td>67-74</td><td>Total Size of Objects Written</td><td>This 8 byte hexadecimal field contains the total size of lvols which are written to the cloud pool during the interval in KiB</td><td>Data is updated when a lvol is premigrated to the cloud pool.</td></tr>
<tr><td>75-82</td><td>Number of Objects Written</td><td>This 8 byte hexadecimal field contains the number of lvols which are written to the cloud pool during the interval.</td><td>Data is updated when a lvol is premigrated to the cloud pool.</td></tr>
<tr><td>83-90</td><td>Total Size of Objects Read</td><td>This 8 byte hexadecimal field contains the total size of lvols which are read from the cloud pool during the interval in KiB</td><td>Data is updated when a lvol is recalled from the cloud pool.</td></tr>
<tr><td>91-98</td><td>Number of Objects Read</td><td>This 8 byte hexadecimal field contains the number of lvols which are read from the cloud pool during the interval.</td><td>Data is updated when a lvol is recalled from the cloud pool.</td></tr>
<tr><td>99-106</td><td>Number of Objects Deleted</td><td>This 8 byte hexadecimal field contains the number of lvols which are deleted from the cloud pool during the interval.</td><td>Data is updated when a lvol is deleted from the cloud pool.</td></tr>
<tr><td>107-114</td><td>Number of Object Look-ups</td><td>This 8 byte hexadecimal field contains the number of lvols which are looked up to check if they exist in the cloud pool during the interval.</td><td>Data is updated when a lvol is checked if it exists in the cloud pool.</td></tr>
<tr><td>115-122</td><td>Total Size of Objects Eligible to be Deleted within 6 hours</td><td>This 8 byte hexadecimal field contains the total size of retained lvols which are eligible to be deleted from the cloud pool within 6 hours in GiB. Retained lvols mean the lvols which are older versions but not deleted yet because volume version retention is enabled by setting retention duration to the cloud pool. Therefore, this is total size of retained lvols whose retention durations expire within 6 hours.</td><td></td></tr>
<tr><td>123-130</td><td>Number of Objects Eligible to be Deleted within 6 hours</td><td>This 8 byte hexadecimal field contains the number of retained lvols which are eligible to be deleted from the cloud pool within 6 hours. Retained lvols mean the lvols which are older versions but not deleted yet because volume version retention is enabled by setting retention duration to the cloud pool. Therefore, this is number of retained lvols whose retention durations expire within 6 hours.</td><td></td></tr>
<tr><td>131-138</td><td>Total Size of Objects Eligible to be Deleted within 24 hours</td><td>This 8 byte hexadecimal field contains the total size of retained lvols which are eligible to be deleted from the cloud pool within 24 hours in GiB. Retained lvols mean the lvols which are older versions but not deleted yet because volume version retention is enabled by setting retention duration to the cloud pool. Therefore, this is total size of retained lvols whose retention durations expire within 24 hours.</td><td></td></tr>
</table>

Bytes	Name	Description	When Data is Sampled/Updated
139-146	Number of Objects Eligible to be Deleted within 24 hours	This 8 byte hexadecimal field contains the number of retained lvols which are eligible to be deleted from the cloud pool within 24 hours. Retained lvols mean the lvols which are older versions but not deleted yet because volume version retention is enabled by setting retention duration to the cloud pool. Therefore, this is number of retained lvols whose retention durations expire within 24 hours.	
147-154	Total Size of Objects Eligible to be Deleted within 36 hours	This 8 byte hexadecimal field contains the total size of retained lvols which are eligible to be deleted from the cloud pool within 36 hours in GiB. Retained lvols mean the lvols which are older versions but not deleted yet because volume version retention is enabled by setting retention duration to the cloud pool. Therefore, this is total size of retained lvols whose retention durations expire within 36 hours.	
155-162	Number of Objects Eligible to be Deleted within 36 hours	This 8 byte hexadecimal field contains the number of retained lvols which are eligible to be deleted from the cloud pool within 36 hours. Retained lvols mean the lvols which are older versions but not deleted yet because volume version retention is enabled by setting retention duration to the cloud pool. Therefore, this is number of retained lvols whose retention durations expire within 36 hours.	
163-170	Total Size of Objects Eligible to be Deleted within 48 hours	This 8 byte hexadecimal field contains the total size of retained lvols which are eligible to be deleted from the cloud pool within 48 hours in GiB. Retained lvols mean the lvols which are older versions but not deleted yet because volume version retention is enabled by setting retention duration to the cloud pool. Therefore, this is total size of retained lvols whose retention durations expire within 48 hours.	
171-178	Number of Objects Eligible to be Deleted within 48 hours	This 8 byte hexadecimal field contains the number of retained lvols which are eligible to be deleted from the cloud pool within 48 hours. Retained lvols mean the lvols which are older versions but not deleted yet because volume version retention is enabled by setting retention duration to the cloud pool. Therefore, this is number of retained lvols whose retention durations expire within 48 hours.	
179-186	Total Size of Objects Eligible to be Deleted within 72 hours	This 8 byte hexadecimal field contains the total size of retained lvols which are eligible to be deleted from the cloud pool within 72 hours in GiB. Retained lvols mean the lvols which are older versions but not deleted yet because volume version retention is enabled by setting retention duration to the cloud pool. Therefore, this is total size of retained lvols whose retention durations expire within 72 hours.	
187-194	Number of Objects Eligible to be Deleted within 72 hours	This 8 byte hexadecimal field contains the number of retained lvols which are eligible to be deleted from the cloud pool within 72 hours. Retained lvols mean the lvols which are older versions but not deleted yet because volume version retention is enabled by setting retention duration to the cloud pool. Therefore, this is number of retained lvols whose retention durations expire within 72 hours.	
195-255	Reserved	All bytes set to x00.	

References

IBM White Paper - TS7700 Bulk Volume Information Retrieval Function User's Guide

Disclaimers:

Product data has been reviewed for accuracy as of the date of initial publication. Product data is subject to change without notice. This information could include technical inaccuracies or typographical errors. IBM may make improvements and/or changes in the product(s) and/or programs(s) at any time without notice.

References in this document to IBM products, programs, or services does not imply that IBM intends to make such products, programs or services available in all countries in which IBM operates or does business. Any reference to an IBM Program Product in this document is not intended to state or imply that only that program product may be used. Any functionally equivalent program, that does not infringe IBM's intellectually property rights, may be used instead. It is the user's responsibility to evaluate and verify the operation of any non-IBM product, program or service.

THE INFORMATION PROVIDED IN THIS DOCUMENT IS DISTRIBUTED "AS IS" WITHOUT ANY WARRANTY, EITHER EXPRESS OR IMPLIED. IBM EXPRESSLY DISCLAIMS ANY WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE OR NON INFRINGEMENT. IBM shall have no responsibility to update this information. IBM products are warranted according to the terms and conditions of the agreements (*e.g.*, IBM Customer Agreement, Statement of Limited Warranty, International Program License Agreement, etc.) under which they are provided. IBM is not responsible for the performance or interoperability of any non-IBM products discussed herein. The customer is responsible for the implementation of these techniques in its environment.

Information concerning non-IBM products was obtained from the suppliers of those products, their published announcements or other publicly available sources. Unless otherwise noted, IBM has not tested those products in connection with this publication and cannot confirm the accuracy of performance, compatibility or any other claims related to non-IBM products. Questions on the capabilities of non-IBM products should be addressed to the suppliers of those products.

The provision of the information contained herein is not intended to, and does not, grant any right or license under any IBM patents or copyrights. Inquiries regarding patent or copyright licenses should be made, in writing, to:

IBM Director of Licensing
IBM Corporation
North Castle Drive
Armonk, NY 10504-1785
U.S.A.

Trademarks

The following are trademarks or registered trademarks of International Business Machines in the United States, other countries, or both.

IBM, TotalStorage, DFSMS/MVS, S/390, z/OS, and zSeries.

Other company, product, or service names may be the trademarks or service marks of others.