

sgi.



## Release Notes for StorHouse Release 5.6

Publication Number  
007-6315-001

StorHouse Release 5.6

November 19, 2013

StorHouse®



© 2013 Silicon Graphics International Corp. All Rights Reserved; provided portions may be copyright in third parties, as indicated elsewhere herein. No permission is granted to copy, distribute, or create derivative works from the contents of this electronic documentation in any manner, in whole or in part, without the prior written permission of SGI.

Publication Number: 007-6315-001

#### LIMITED RIGHTS LEGEND

The software described in this document is "commercial computer software" provided with restricted rights (except as to included open/free source) as specified in the FAR 52.227-19 and/or the DFAR 227.7202, or successive sections. Use beyond license provisions is a violation of worldwide intellectual property laws, treaties and conventions. This document is provided with limited rights as defined in 52.227-14.

#### TRADEMARKS AND ATTRIBUTIONS

SGI, SGI InfiniteStorage, the SGI logo, Supportfolio, SGI Trusted Edge, and SGI StorHouse are trademarks or registered trademarks of Silicon Graphics International Corp. or its subsidiaries in the United States and other countries. All other trademarks mentioned herein are the property of their respective owners.



# Contents

- Welcome ..... 1
  - Intended Audience.....1
  - Document Organization.....1
  - Related Documentation.....2
- Chapter 1: Release Enhancements..... 3
  - Detailed Feature List.....3
  - Support for Red Hat Linux Operating Systems .....9
  - Enhancements to Level F Storage.....10
    - Support for Multiple Level F Volume Sets .....10
    - Commands That Do Not Affect Level F .....14
    - Centera as Level F .....14
    - Hitachi Content Archive Platform as Level F.....14
    - Cloud Storage as Level F .....15
    - Device Scan Recovery .....15
  - Enhancements to Level L Storage.....16
    - MAID Support .....16
    - LTO Drive and Media Support.....17
    - Support for Other New Media/Recording Types .....18
  - File Retention.....19
    - About File Retention .....19
    - Setting Retention Periods .....20
    - Uncataloging Volumes with Retained Files.....23

File Replication.....	23
About Replication .....	24
Implementing Replication.....	25
System Parameter Changes .....	26
RETENTION_MODE System Parameter.....	26
SNMP_TRAPS System Parameter.....	28
SQL_DROP_HOLD System Parameter .....	28
SQL_FTPS_ONLY.....	29
Changes to System Parameters .....	29
User Log Changes.....	30
security_file Record .....	31
security_cmd Record .....	32
command_exec Record .....	32
device_error Record .....	33
file_copy Record .....	33
New Status Codes .....	33
Operator Messages.....	36
New Messages .....	36
Updated Operator Messages.....	38
<b>Chapter 2: Changes to the StorHouse Interface .....</b>	<b>39</b>
StorHouse Command Language .....	39
CATALOG DEVICE .....	40
CREATE FILE.....	43
CREATE FSET .....	45
CREATE VSET .....	48
EXTRACT DIRECTORY.....	50
IMPORT .....	52
MIGRATE .....	53
PUT.....	59
RECOVER DEVICE .....	60
RECOVER VOLUME.....	63



RELOCATE.....	64
REMOVE FILE.....	66
REPLICATE .....	67
SCHEDULE .....	73
SET DEVICE.....	76
SET FILE .....	79
SET FSET .....	82
SET VOLUME .....	85
SET VSET.....	86
SHOW CONFIGURATION .....	88
SHOW DEVICE .....	92
SHOW FILE .....	93
SHOW FSET .....	102
SHOW VOLUME .....	104
SHOW VSET.....	111
STAGE.....	113
VALIDATE DEVICE.....	114
VALIDATE FILE .....	118
VALIDATE VOLUME .....	122
Absolute Time Specification .....	132
Miscellaneous Changes.....	133
Generic Callable Interface.....	134
LSMCO - Create Open.....	134
LSMOS – Open Sequential .....	135
Mainframe Callable Interface .....	138
CREATE-OPEN .....	138
OPEN-SEQ .....	140





## Welcome

Welcome to StorHouse® Release 5.6. This document details the content of Release 5.6. Release 5.6 supports all capabilities of previous releases except if noted otherwise in this document.

## Intended Audience

These release notes are intended for StorHouse users who are familiar with the StorHouse software and for new users who want a summary of system changes.

## Document Organization

This document contains two chapters:

- Chapter 1, “Release Enhancements,” lists the enhancements and features in StorHouse Release 5.6, explains new concepts and user log information, and describes system parameter changes and new StorHouse messages.
- Chapter 2, “Changes to the StorHouse Interface,” describes the StorHouse Release 5.6 modifications to Command Language commands, Generic Callable Interface functions, and mainframe Callable Interface functions.



## Related Documentation

The StorHouse User Document Set contains the following manuals:

- *Host Installation and Operations Guide*, publication number 900011 for IBM MVS hosts, 900051 for UNIX→ hosts, and 900052 for DOS hosts, explains how to install the StorHouse host software. These guides are intended for system administrators and system programmers.
- *StorHouse Glossary*, publication number 900027, defines technical terminology used in all SGI StorHouse publications.
- *Callable Interface Programmer's Guide*, publication number 900013 for mainframe hosts and the *Generic Callable Interface Programmer's Guide*, publication number 900046 for all other hosts, are references for programmers who write applications that invoke the StorHouse Callable Interface. These guides explain the functions of the Callable Interface and contain sample programs.
- *Command Language Reference Manual*, publication number 900005, is a general reference for Command Language, the standard command interface between StorHouse and all host computers. It contains descriptions of commands and related concepts.
- *Messages and Codes Manual*, publication number 900011, describes the messages and return codes generated by the StorHouse host software and the StorHouse software. It lists the messages by status code, gives the meaning of each message, and indicates any actions to take as a result of the messages.
- *User Log Format*, publication number 900028, describes the format of the StorHouse User Log. It is intended for programmers who write applications to generate reports from statistical information contained in the User Log. Refer to these documents for more information about your StorHouse system.

Refer to these documents for more information about your StorHouse system.





## Release Enhancements

This chapter describes enhancements and features in StorHouse Release 5.6 through software deliveries 241 (IBM AIX), 289 (Sun Solaris), and 117 (Linux). As of this version of Release 5.6, StorHouse no longer supports HP-UX platforms. Refer to Chapter 2, “Changes to the StorHouse Interface,” for details about the new and/or modified StorHouse Command Language commands and Generic Callable Interface and mainframe Callable Interface functions.

## Detailed Feature List

StorHouse Release 5.6 contains the following additions and modifications:

- Support for the Red Hat Linux Release 5.x (5.0 to 5.6) operating system installed on X86 platforms.
- Enhancements to level F storage:
  - Support for multiple level F volume sets.
  - Device scan recovery for directory and disaster (extended) recovery of level F devices.
  - Support for Centera as part of the StorHouse level F storage.

- New media specifications (MJA and MJB) for Centera.
- Support for the Hitachi Content Archive Platform (HCAP).
- New media specification MKA for HCAP.
- Enhancements to level L storage:
  - Support for COPAN Systems MAID technology as part of StorHouse level L storage.
  - New media specifications (PAA) for COPAN Systems Revolution 200T MAID device.
  - Support for the Sun STK SL8500 I and SL3000 libraries.
  - Support for LTO drives and media.
    - New media specifications (TFB and TFC) for LTO3 media.
    - New media specification (TFD) for LTO4 media.
    - New media specification (TFE) for LTO5 media.
  - New media specifications (TDC and TDD) for the 9840 drive.
  - New media specifications (TIC, TID, TIE, TIF, and TIZ) for the T10000B drive.
  - New media specifications (THC, THD, THE, THF, THG, THH, THI, and THJ) for the IBM 3592 drive and library.
  - New media specifications (THK, THL, THM, and THN) for the Spectra Logic library
- Support for file retention at the file and file set level.
- Support for application-independent file replication from one StorHouse system to another.
- New StorHouse Command Language commands:
  - CATALOG DEVICE - implements device scan recovery of level F devices.
  - REPLICATE - duplicates files from one StorHouse system to another.
  - SHOW CONFIGURATION - displays system configuration information.
  - VALIDATE DEVICE - verifies all file extents on a level L device are readable.
  - VALIDATE FILE - verifies all extents of a file can be read.



- Modifications to existing StorHouse Command Language commands:
  - CREATE FILE: New /RETENTION modifier.
  - CREATE FSET: New /FORCE\_RETENTION, /RETENTION and /RPL\_CLASS modifiers.
  - CREATE VSET: When creating level F volume sets, /LIBRARY specifies the Fxx portion of the corresponding level F free pool volume set.
  - EXTRACT DIRECTORY:
    - New /RETENTION modifier.
    - Now extracts file set and volume set information for files on level F devices.
  - IMPORT: New /VALIDATE\_DATE modifier.
  - MIGRATE:
    - /BY\_VSET Always creates noncontiguous destination file sets.
    - Added new filename parameter and associated parameter modifiers: /GROUP, /NOSPECIAL, /PASSWORDS, /REPORT, /VERSION, and /WAIT.
  - PUT: New /RETENTION modifier.
  - RECOVER DEVICE: New BYPASS\_LABEL, /LABEL, and /VALIDATE\_DATE modifiers.
  - RECOVER VOLUME:
    - Now supports the /ERASE modifier.
    - Now reports the extents and bytes copied.
    - Allows users to execute more than one RECOVER VOLUME command at a time.
  - RELOCATE:
    - The /VOLUME modifier allows the specification of level F devices.
    - New parameter modifiers /USED and /UNUSED.
  - REMOVE FILE: New /BEFORE modifier.

- RESTORE DIRECTORY: Now restores file set and volume set information for level F devices.
- RETIRE VOLUME:
  - Now reports the extents and bytes copied.
  - Allows users to execute more than one RETIRE VOLUME command at a time.
- SCHEDULE:
  - SCHEDULE supports more Command Language commands. Only these Command Language commands cannot be scheduled: CREATE FILE, EXECUTE STH\_LOAD, GET, LOGOFF, MONITOR, PUT, SCHEDULE, SET ACCOUNT, and SIGNOFF. All other commands may be scheduled. 1
  - New modifiers: /EMAIL, /TEST\_EMAIL, /ON\_ERROR, and /DISABLED.
- SET DEVICE:
  - New /ACCESSIBLE modifier to change the mode of a drive in an STK SL8500 library from INACCESSIBLE to ACCESSIBLE.
  - New /CLEANED modifier to support changing the mode of a level L drive.
  - New /NEEDS\_CLEANING modifier to set the needs\_cleaning mode for the drive.
  - New /SIZE modifier to support configuring space on Centera clusters and filesystem level F drives (for example, network attached storage [NAS] devices).
  - New /UNWRITELOCKED modifier to change the mode of a level F drive.
  - New /WRITELOCKED modifier to set the mode of a level F drive to writelocked.
- SET FILE: New /FSET, /RETENTION, /NOREPLICATED, /RPL\_CLASS, and /WAIT modifiers.
- SET FSET: New /FORCE\_RETENTION, /REDUCE, /RPL\_CLASS, and /RETENTION modifiers.
- SET VOLUME: New /VALIDATE\_DATE modifier.



- SET VSET:
  - New /REDUCE modifier.
  - /LIBRARY allows specification of a level F device.
- SHOW DEVICE:
  - New mode of INACCESSIBLE to indicate a drive in an STK SL8500 library is inaccessible because a robot is unavailable to access it.
  - New device modes of WRITE\_LOCK and NEEDS\_CLEANING.
  - New /FULL modifier shows the volume set, directory, and size of a level F device.
  - The output of a command for a library device may include one of these two new messages: INTERVENTION\_REQD (when a library is waiting on an XROMIR operator request) and RESET\_REQD (when a library is waiting on an XRORST operator request).
- SHOW FILE:
  - Command modifications consisting of a new /NAME command modifier; new /ARCHIVE\_EXISTS, /BACKUP\_EXISTS, /BKP\_ATTR, /BUFFERED, /EXPIRED, /MAXEXTENTS, /MINEXTENTS, /MINSIZE, /NOARCHIVE\_EXISTS, /NO\_BACKUP\_EXISTS, /PHYSICAL, /REPLICA\_EXISTS, RESIDENT, /RETAINED, /RPL\_CLASS, /SAFE\_COPIES, and /USED parameter modifiers; and expanded definition of the /DAMAGED parameter modifier. In addition, /VOLUME can specify a level F volume ID, and /EXTENT displays the actual media and recording type for level F volumes.
  - SHOW FILE/FULL display modifications consisting of a new descriptor field (REPLICATED) and two new fields (RETENTION and RPL\_CLASS).
- SHOW FSET: New display that includes one new STATE (FORCE\_RETENTION) and two new fields (RETENTION and RPL\_CLASS).
- SHOW SCHEDULE: The /START modifier now accepts a value of NONE.
- SHOW VOLUME:
  - New /FIXED modifier to indicate a level F volume.
  - New /DISMOUNTED, /FIRST\_ALLOCATED, /LAST\_ALLOCATED, /NOTDISMOUNTED, /USED, /VALIDATED, and /NOTVALIDATED modifiers.

- The command now displays information for level F volumes under any one of the following conditions:
  - The specified volume ID begins with the letter “M.”
  - /FIXED is specified.
  - /VSET specifies a level F VSET.
  - /FREE\_POOL specifies a free pool that begins with the letter “F.”
  - /LOCATION specifies a location that begins with the letter “F.”
- SHOW VSET: New /FREE\_POOL modifier enables information for free pool volume sets to be displayed.
- STAGE: New /FORCE modifier to permit file staging to the performance buffer even when the staging operation triggers a migration.
- VALIDATE VOLUME: Enhanced command functionality, including support for wildcarded volume IDs, automatic periodic validation of removable volumes, and the /NODATA command modifier.
- The absolute date/time specification now accepts the following date keywords: YESTERDAY, TODAY, TOMORROW, LAST, and NEXT. These keywords can be used in place of the date component. For example, NEXT:20 indicates the next 8:00 P.M, which could be today or tomorrow, depending on the time the command was run.
- New and modified system parameters:
  - New RETENTION\_MODE system parameter for setting a system-wide retention enforcement level.
  - New SNMP\_TRAPS system parameter for controlling the StorHouse SNMP trap facility.
  - New SQL\_DROP\_HOLD system parameter for specifying the number of days that must pass between a StorHouse/RM DROP TABLE and PURGE table operation.
  - New SQL\_FTPS\_ONLY system parameter for specifying whether StorHouse disallows insecure (encrypted) logins or data transfers.
  - Increased maximum for the XFR\_COUNT system parameter.
  - Increased maximum and default for VRAM\_CACHE\_MAX.



- SSPNO system parameter is now a dynamic parameter with a range of 0 to 100.
- XFR\_BUFSIZ now ranges from 0 to 1240 with a default value of 620.
- XFR\_BUFSIZ\_D now ranges from 0 through 1240 with a default value of 0.
- XFR\_MEMADR\_D now has a 0 range and a default of 0.
- XFR\_MEMSIZ\_D now has a 0 range and a default of 0.
- SQL\_BKUP\_LIMIT now has a slightly modified description, a range of 1 through 11000 (about 30 years of daily backups), and a default of 366 (about 7 years of weekly backups).
- Changes to the Generic Callable Interface LMSOS and LSMCO functions to support file retention.
- Changes to the mainframe Callable Interface OPEN-SEQ and CREATE-OPEN functions to support file retention.
- User Log
  - Changes to five StorHouse user log records: security\_file, security\_cmd, command\_exec, device\_error, and file\_copy.
  - New “SHOW CONFIGURATION - 816” entry in the *User Log Format*, Table A-11, “StorHouse Command IDs,” for the SHOW CONFIGURATION command.
  - New StorHouse status codes and operator messages.

StorHouse Release 5.6 requires Sun™ Solaris™ 8 or higher.

## Support for Red Hat Linux Operating Systems

The StorHouse software now runs on the Red Hat Linux Release 5.x (5.0 to 5.6) operating system installed on X86 platforms.

## Enhancements to Level F Storage

StorHouse Release 5.6 supports the following enhancements to level F storage:

- Multiple level F volumes sets and media types
- Centera (fixed content object-based disk) as part of level F storage
- Hitachi Content Archive Platform (HCAP) as part of level F storage
- Support for Cloud storage Level F devices on Linux
- Device scan recovery for directory and disaster recovery of level F devices.

## Support for Multiple Level F Volume Sets

StorHouse supports any combination of the following storage for level F:

- Directly attached JBOD and RAID raw devices
- Filesystems mounted on directly attached JBOD and RAID devices
- Filesystems mounted on NAS devices
- Centera.

Before StorHouse Release 5.6, if a Centera cluster was configured on a StorHouse system, no RAID or filesystem devices could be used for level F (including the performance buffer). Similarly, if RAID was configured as a raw device, no Centera or filesystem devices could be used. In addition, level F supported only one volume set, MAGDISK, which contained the performance buffer file set, \$\$BUFFER. StorHouse Release 5.6 removes these restrictions by adding support for multiple level F media types and volume sets on a single StorHouse system.

Note: To StorHouse, a level F volume and a level F device are the same. In the discussion that follows, these words are interchangeable.

### Configuring Level F

Configuring level F is SGI's responsibility. Prior to installation, SGI configures the MAGDISK volume set and the \$\$BUFFER file set (the performance buffer). SGI also sets up StorHouse level F devices for customers and allocates them to free pool



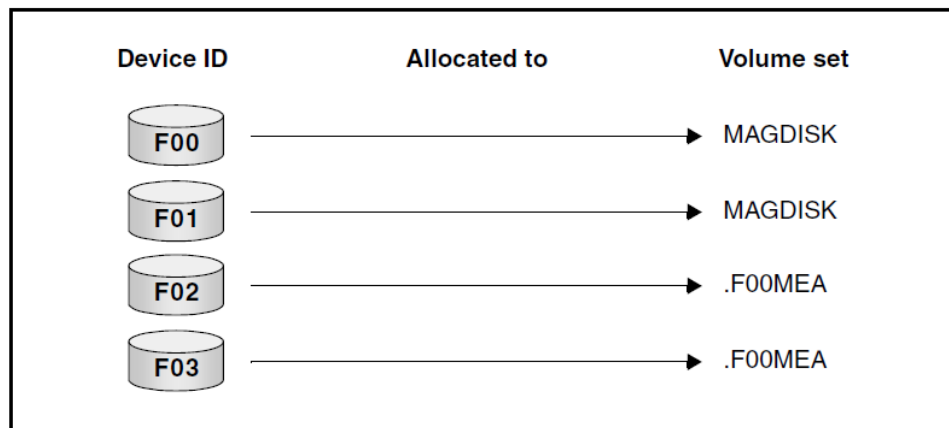
volume sets. All devices allocated to a level F free pool volume set have the same media and recording type.

Level F free pool volume sets have the following naming convention:

`{.Fxx}{mmr}`

where .F is a constant, xx is a value expressed in hexadecimal 00 through FF, mm is the media type, and r is the recording type of the devices in the free pool. Example level F free pool volume set names are .F00MIA and .F01MEA.

The Fxx portion of the free pool volume set name does not refer or relate to any specific level F device. For example, Figure 1-1 illustrates a StorHouse configuration with four level F devices: F00, F01, F02, and F03. Prior to installation, SGI allocated F00 and F01 to the MAGDISK volume set and F02 and F03 to the free pool volume set .F00MEA. Volumes F02 and F03 are available for allocation to user level F volume sets.

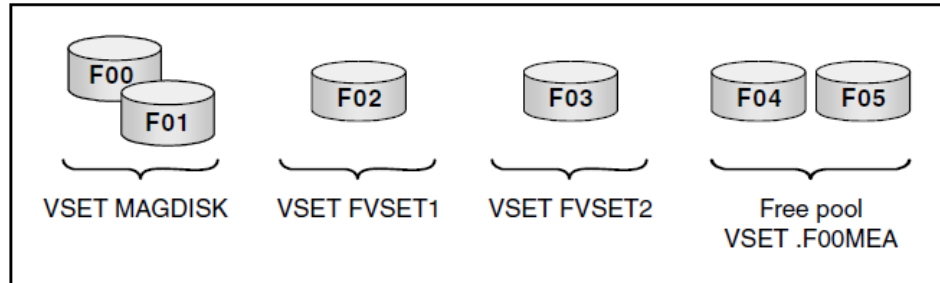


**Figure 1-1: Sample StorHouse Level F Configuration**

Figure 1-2 illustrates a StorHouse configuration with three level F volume sets (MAGDISK, FVSET1, and FVSET2) and one level F free pool (.F00MEA).

- Devices F00 and F01 are allocated to the MAGDISK volumes set.
- Device F02 is allocated to volume set FVSET1.

- Device F03 is allocated to FVSET2.
- Devices F04 and F05 are allocated to the free pool volume set .FOOMEA and available for subsequent assignment to other user volume sets.



**Figure 1-2: Multiple Level F Volume Set Configuration**

### Creating Level F Volume Sets

Once SGI configures level F, system administrators can use the StorHouse Command Language CREATE VSET command to create a new level F volume set. The volume set can reside in the primary, backup, or archive directory. Note that administrators cannot assign a specific level F device to a specific volume set. Instead, StorHouse selects the appropriate device(s)/volume(s) for the volume set based on information specified in the CREATE VSET command.

Administrators use information from a level F free pool volume set name for values of the CREATE VSET /LIBRARY and /MEDIA modifiers.

- The /LIBRARY modifier must always equal the second, third, and fourth characters (the Fxx part) of a level F free pool volume set name.
- The /MEDIA modifier must always equal the last three characters of the free pool volume set name. These characters indicate the media and recording type of the device(s) to be allocated.

For example, if the free pool volume set name is .FOOMEA, then /LIBRARY must equal F00 (no leading period) and /MEDIA must equal MEA.

### Assigning Volumes to Level F Volume Sets

StorHouse allocates volumes to a level F volume set depending on the value of the CREATE VSET /SIZE modifier. If /SIZE=0, StorHouse allocates one or more volumes



when files are actually written to the volume set. If /SIZE is greater than zero, StorHouse allocates one or more entire volumes from a level F free pool to the volume set at create time. Here are some examples.

- If /SIZE=11GB and each volume in the free pool is approximately 10 GB, StorHouse allocates two whole volumes to the volume set.
- If /SIZE=1 GB and each free pool volume is approximately 10 GB, StorHouse allocates one whole volume to the volume set.

StorHouse allocates each level F device to one and only one volume set. System administrators can release a level F device allocated to a volume set back to a free pool when all files in the volume set have been deleted.

### Level F Volume Sets and the Performance Buffer

Files written to level F may circumvent the performance buffer depending on their location and the setting of the VTF modifier. Files written to file sets in the MAGDISK volume set will continue to bypass the performance buffer. Files written to other level F volume sets will be written to the performance buffer first and later copied (written-back) to their resident file sets unless VTF=DIRECT is specified.

### Level F Device and Volume IDs

Some StorHouse Command Language commands now support the display and/or specification of level F volume and/or device IDs. A level F device ID has the following format:

```
{storage_level}{unit_number}
```

where storage\_level is always F, and unit\_number is a hexadecimal value ranging from 00 to FF (for example, 0A). A StorHouse system can support a maximum of 256 level F devices. An example of a level F device ID is F0A.

A level F volume ID has the following format:

```
{media_type}{recording_type}{device_ID}{:}{side_indicator}
```

An example of a level F volume ID is MEAF04:A, where:

- ME indicates the media type for RAID.
- A indicates the recording type for standard density.
- F04 indicates the device ID.

- The colon symbol (:) always precedes the side indicator.
- A is always used for the side indicator.

## Commands That Do Not Affect Level F

The following StorHouse Command Language commands do not apply to level F:

- CATALOG VSET
- ERASE VOLUME
- ERASE VSET
- EXPORT
- IMPORT
- MIGRATE/BLANKS
- MIGRATE/BY\_VSET
- MOVE VOLUME
- MOVE VSET
- RECOVER VOLUME
- RETIRE VOLUME
- SET VOLUME
- UNCATALOG VOLUME
- UNCATALOG VSET
- VALIDATE VOLUME

## Centera as Level F

StorHouse Release 5.6 supports Centera as part of level F storage by defining a Centera cluster as a logical level F volume (or device). Centera clusters may be shared by non-StorHouse applications and by other StorHouse systems.

StorHouse uses media type MJ to identify Centera devices. Media type MJ identifies the vendor application program interface (API) used to access the storage device. A recording type of A indicates a Centera device operating in non-compliance mode. A recording type of B indicates a Centera device operating in Basic Compliance or Compliance Plus mode (Centera with retention enforced by the device itself). Vendor software/firmware controls all media characteristics.

## Hitachi Content Archive Platform as Level F

StorHouse Release 5.6 supports the Hitachi Content Archive Platform (HCAP) as part of level F storage. StorHouse uses media type MK to identify the vendor application



program interface to access the HCAP storage device. Recording type A is the only valid recording type. Vendor software/firmware controls all media characteristics.

## Cloud Storage as Level F

StorHouse supports Cloud storage Level F devices on Linux. The supported devices are Amazon S3, Amplidata AmpliStor, and Cleversafe dsNet. All use the recording type MLA. Cloud storage is network attached and uses a representational state transfer (REST) interface.

## Device Scan Recovery

Device scan recovery is a new recovery strategy that supports either directory or disaster recovery of level F devices. Device scan recovery scans the level F device to be recovered to obtain information about the StorHouse files, extents, volume sets, and file sets on the device. Then it catalogs those entities. System administrators implement device scan recovery with the new StorHouse Command Language CATALOG DEVICE command.

Some level F devices (for example, Centera) have built-in drive replication capabilities. If your StorHouse system utilizes such technology, you can use it to replicate data from your primary StorHouse site to a disaster recovery StorHouse site. Should a disaster occur, you can subsequently use CATALOG DEVICE as a recovery technique.

To prepare for disaster recovery of a level F with drive replication capability, the device at the primary site must replicate data to a device at the backup site. This requires a network connection of sufficient bandwidth. Because level F devices that support replication write files to the backup site almost continuously, device scan recovery restores files almost to the point of the disaster. (Device replication differs from StorHouse replication.)

Device scan recovery is independent of existing disaster recovery procedures for level L optical and tape volumes. StorHouse systems with level F and level L devices must perform disaster recovery for both. Furthermore, if checkpoint recovery is used to recover level L, it must be completed before level F device scan recovery begins.

As with checkpoint recovery, directory recovery from extraction files, and directory recovery from physical volumes, SGI must assist customers with device scan recovery.

## Enhancements to Level L Storage

StorHouse Release 5.6 adds support for COPAN Systems Revolution 200T massive arrays of idle disk (MAID) and LTO3, LTO4, and LTO5 drives and media to the StorHouse level L storage layer. The release also adds support for the Sun STK SL8500 and SL3000 libraries and media types TDC, TDD, TIC, TID, and TIZ.

### MAID Support

COPAN Systems MAID is a “power-managed” storage technology where disks spin and consume power only when required for data access. StorHouse enhances MAID reliability, performance, application support, scalability, and operation. The media and recording type for COPAN Systems MAID is PAA, where the general media type P indicates power-managed disk arrays, the form factor A indicates a MAID device that emulates tape, and the recording type A specifically indicates the Revolution 200T.

PA has the following characteristics:

- MAID device emulating a tape library and tape drives.
- High-performance.
- Zero time extent positioning.
- Erasable (volume reusable).
- Drives with optional support for compression.
- Single-sided virtual tape volumes (VTC), which are permanent and cannot be removed.
- VTCs with a shelf life that is a function of the time-powered-on and the number of power cycles.
- Volumes that can be read many times without degrading the media.
- Support for one read or one write at a time.

- Recording type: A indicates a COPAN Revolution 200T shelf with 27 VTCs (each with 750 GB) and a maximum of seven virtual tape drives (VTDs), which can be powered on concurrently.

## LTO Drive and Media Support

StorHouse supports LTO drives and media as part of level L storage. The media type is TF, and the recording types are B or C for LTO3, D or F for LTO4, E or G for LTO5, and H or I for LTO6.

Media type TF has the following characteristics:

- LTO Ultrium high-performance, magnetic tape cartridges.
- Erasable (volume reusable).
- Drives support compression, which can increase average storage capacity.
- Volumes are single-sided.
- Volumes are enclosed in plastic cartridges and can be stored on shelves.
- Volumes have a shelf life ranging from 15 to 30 years.
- Volume can be read and/or written many times and degrade with use.
- Medium supports one read or one write at a time.
- Recording type:
  - B indicates an uncompressed data cartridge (200 GB).
  - C indicates an uncompressed data cartridge (400 GB).
  - D indicates an uncompressed data cartridge (800 GB).
  - E indicates an uncompressed data cartridge (1.5 TB).
  - F indicates an uncompressed WORM data cartridge (800 GB).
  - G indicates an uncompressed WORM data cartridge (1.5 TB).
  - H indicates an uncompressed erasable data cartridge (2.5 TB)
  - I indicates an uncompressed WORM data cartridge (2.5 TB)

StorHouse Release 5.6 continues to support recording type A and Z.

## Support for Other New Media/Recording Types

StorHouse supports the following new media types:

- Media type TD and recording types C and D for the 9840 drive.
  - TDC indicates a data cartridge (uncompressed 75 GB erasable).
  - TDD indicates a cartridge (uncompressed 75 GB WORM).
- Media type TI (recording types C, D, E, F and Z) for the T10000B drive.
  - TIC indicates a data cartridge (uncompressed 1000 GB erasable).
  - TID indicates a data cartridge (uncompressed 1000 GB WORM).
  - TIE indicates a data cartridge (uncompressed 5000 GB erasable).
  - TIF indicates a data cartridge (uncompressed 5000 GB WORM).
  - TIZ indicates a cleaning cartridge (50 passes).
- Media type TH (recording types C, D, E, F, G, H, I, J) for the IBM 3592 drive and library.
  - THC indicates a data cartridge (1 GB erasable).
  - THD indicates a data cartridge (1 GB WORM).
  - THE indicates a data cartridge (640 GB erasable).
  - THF indicates a data cartridge (640 GB WORM).
  - THG indicates a data cartridge (700 GB WORM).
  - THH indicates a data cartridge (1000 GB erasable).
  - THI indicates a data cartridge (1000 GB WORM).
- Media type TH (recording types K, L, M, and N) for the IBM 3592 drive and Spectra Logic library.
  - THK indicates a data cartridge (1600 GB erasable).
  - THL indicates a data cartridge (1600 GB WORM).





- THM indicates a data cartridge (4000GB erasable).
- THN indicates a data cartridge (4000 GB WORM).
- THZ indicates a 300-pass cleaning cartridge.

## File Retention

StorHouse Release 5.6 provides new features for setting and enforcing file retention. These features ensure StorHouse archiving adequately supports industry-based compliance rules and data remains accessible throughout its required life span.

### About File Retention

Starting with SM 5.6, each file written to StorHouse has a *retention attribute*, which determines a file's retention period. A *retention period* indicates the time span that a file may not be deleted from StorHouse. A file is considered *retained* when it has a non-zero retention period and that retention period has not expired.

StorHouse calculates file retention periods by adding the number of days in the retention period to a file's last modified date (the `modified_file` date as displayed by the StorHouse Command Language `SHOW FILE` command). A file retention period ends when the current date is beyond the last modified date plus the retention period. For example, if a file with a 3-day retention period was last modified at 11 p.m. on December 12, it would expire at 11 p.m. on December 15.

StorHouse does not automatically delete files after their retention period expires. It simply allows applications or users to delete them.

A user or an application can specify one of four retention values at file create time:

- `DEFAULT` (the retention is not specified and assumes the default value).
- `FOREVER` (never allow the file to be deleted).
- `ZERO` (do not apply any retention period to the file).
- `Number of days` (retain the file for the specified number of days).

The maximum file retention period is 65,000 days, or approximately 178 years.

Copies of primary files (for example, backup, archive, replica, and relocated files) inherit the retention attribute of the primary. The exception is the performance buffer copy. It is treated as temporary cache and has no retention attribute.

StorHouse records a file's retention setting on the media where the file resides. If the retention period is modified, the system updates the media to preserve the change.

Note the following:

- To specify retention on a StorHouse 5.6 system, you can use any version of the StorHouse Host Application Programming Interface (API) with the Interactive Interface. You must use version 2.5 (for Windows/UNIX) or version 1.8 (for MVS) of the StorHouse Host API with the Callable Interface. If you use prior versions of the Host API, StorHouse assumes that the retention period is the default.
- The new Host API releases do not pass retention information from Callable Interface functions to a StorHouse system running any release prior to StorHouse 5.6. Therefore, the Callable Interface can be used with older StorHouse systems. However, interactive commands with retention parameters/modifiers will fail if executed on a StorHouse system running a release prior to 5.6.
- When an older StorHouse release is upgraded to Release 5.6, files and file sets created with previous StorHouse releases will be assigned a retention attribute of ZERO after the conversion.

## Setting Retention Periods

Retention can be set at the file and file set level. File-level retention takes precedence over file set-level retention unless the system administrator or a user with the proper authority instructs the software otherwise. (See the `/FORCE_RETENTION` modifier described in the “File Set-Level Retention” section below.

## Setting File-Level Retention

You can set file-level retention when writing a file to StorHouse by specifying:

- The /RETENTION modifier on the StorHouse Command Language CREATE FILE or PUT command.
- The retention\_interval list member on the Generic Callable Interface LSMCO or LSMOS (MODE=WRITE) function.
- The FATTR-RETENTION-INTERVAL on the mainframe Callable Interface OPEN-SEQ or CREATE-OPEN function.

You can change the file-level retention attribute for a particular file version by specifying a new retention value on the SET FILE command.

## Setting File Set-Level Retention

System administrators or users with the proper authority can configure file set-level retention by specifying the /RETENTION modifier on the CREATE FSET or SET FSET command. CREATE FSET /RETENTION applies to all files in the file set. SET FSET/RETENTION applies only to new files in the file set.

CREATE FSET and SET FSET also support a /FORCE\_RETENTION modifier. When a file is created, the /FORCE\_RETENTION modifier tells StorHouse to override the file retention attribute with the file set retention attribute. Administrators can use /FORCE\_RETENTION to supersede the retention period specified by an application through the Callable or Interactive Interface.

For example, assume the following:

- The system administrator specified /FORCE RETENTION and a file set-level retention of FOREVER on the CREATE FSET command for FSET\_A.
- An application creates File A in FSET\_A and explicitly specifies a 2-day retention period.

In this case, at file create time, the file set-level retention attribute overrides the file-level retention attribute of two days, causing StorHouse to retain File A forever.

## Determining File Retention Values When DEFAULT is Specified

When StorHouse creates a primary file, it checks several places to determine the file's retention value. As soon as StorHouse finds a retention value other than DEFAULT, it assigns that value to the file.

- First, StorHouse checks the Interactive Interface command or Callable Interface function that creates the file. If it finds a value other than DEFAULT, StorHouse assigns that value to the file unless /FORCE\_RETENTION is in effect on the file set. In this case, StorHouse overrides the file-level specification with the file set level specification.
- If the file retention value is DEFAULT, StorHouse checks the file set retention attribute for a value other than DEFAULT. If found, it assigns that retention value to the file.
- If the file set retention value is DEFAULT, StorHouse uses the value of the RETENTION\_MODE system parameter to determine file retention. If RETENTION\_MODE is set to BASIC, StorHouse sets the file-level retention to ZERO, which indicates no retention. If set to STRICT, StorHouse sets the file-level retention to FOREVER, which indicates infinite retention.

Table 1-1 provides some examples of how StorHouse determines file-level retention.

**Table 1-1: File Retention Examples**

Category	Value			
	Example 1	Example 2	Example 3	Example 4
File level retention	DEFAULT	DEFAULT	30 days	30 days
File set-level retention	DEFAULT	DEFAULT	60 days	60 days
/FORCE_RETENTION	N/A	N/A	No	Yes
RETENTION_MODE	BASIC	STRICT	N/A	N/A
Final File Retention	ZERO	FOREVER	30 days	60 days

## Uncataloging Volumes with Retained Files

The new retention feature affects the operation of the UNCATALOG command. Therefore, customers who currently use UNCATALOG and EXPORT to remove archive volumes from StorHouse may want to rethink this strategy.

StorHouse cannot uncatalog a volume that contains retained files because those files cannot be deleted until their retention period expires. When StorHouse attempts to uncatalog such a volume, it ignores the retained files, deletes and removes all nonretained files on the volume, and leaves the volume in the UNCATALOGING state. A volume in the UNCATALOGING state cannot be exported. To circumvent this restriction, SGI recommends that instead of using UNCATALOG and EXPORT to remove archive volumes from StorHouse, you simply write lock the volumes to be exported and use the MOVE VOLUME command to transfer them from level L to shelf storage.

## File Replication

StorHouse Release 5.6 supports application-independent replication of one or more eligible primary files from a source StorHouse system to a destination, or target, StorHouse system. Replication is typically used to provide redundancy for disaster

recovery or off-site storage. A file is considered *replicated* when at least one copy resides on the destination system, including on the performance buffer. Any StorHouse file type can be replicated (VRAM, sequential, or STORHOUSE).

## About Replication

A primary file is eligible for replication when it has an assigned replication class. A *replication class* is a named set of information about the destination StorHouse system. Table 1-2 describes the components that compose a replication class.

**Table 1-2: Replication Class Components**

Component	Specifies the
RPL_CLASS_NAME	Name of the replication class. A replication class name can consist of from 1-8 of the following ASCII characters: A_Z, 0-9, _ and \$. StorHouse always forces replication class name to uppercase, even when enclosed in quotes.
DISABLED_FLAG	Indicator specifying whether the replication class is disabled, A non-blank character indicates disabled,
NETWORK_DEVICE	Network device used to connect to the destination StorHouse system. This value must be set to N00 to indicate TCP/IP.
SYSNAME	Network system name used to connect to the destination StorHouse system. This value must be the same as the SM_HOSTID in the default SMCONFIG file on the destination system.
LINKNAME	Network link name used to connect to the destination StorHouse system. This value must be the same as the SM_LINKNAME in the default SMCONFIG file on the destination system.
VSET	Volume set that will contain the replicated files on the destination StorHouse system. Valid values are: <ul style="list-style-type: none"> <li>■ VSET= volume_set_name</li> <li>■ VSET=* (indicates to use the same volume set name as the</li> </ul>



	source file) If VSET is omitted, StorHouse uses the account's default volume set on the destination StorHouse.
FSET	File set name that will contain the replicated files on the destination StorHouse system. Valid values are: <ul style="list-style-type: none"><li>■ FSET= file_set_name</li><li>■ FSET=* (indicates to use the same file set name as the source file)</li></ul> If FSET is omitted, StorHouse uses the account's default file set on the destination StorHouse.

StorHouse associates a file with a replication class by:

- Assigning a replication class to the file's resident file set (SET FSET or CREATE FSET command). In this case, at create time, the file inherits its replication class from its file set.
- Explicitly specifying a default replication class for the file version (SET FILE command).

Currently, SGI is responsible for creating and maintaining replication classes. For more information about these tasks, contact your SGI customer support representative.

## Implementing Replication

System administrators implement replication with the StorHouse Command Language REPLICATE command. For convenience, this command can be scheduled to run periodically

A replication operation works as follows:

- When an application creates or modifies a replication-eligible file, StorHouse queues the same action to occur on the destination StorHouse system the next time the REPLICATE command executes on the source system and selects that file.

- When a file is deleted (and removed) from the source system, StorHouse queues the same action to occur on the destination StorHouse the next time the REPLICATE command executes on the destination system and selects any file.

StorHouse does not copy file and file set replication class attributes from one system to another. A replica inherits the replication class attribute of its target file set. In addition, the system does not write queued files to the destination system in any particular order or within a set time interval.

REPLICATE uses the same account, password, and file access group on the source and target systems. The system administrator must manually create the respective accounts and volume sets on the target location. If the required groups and file sets do not already exist on the destination system, StorHouse creates them automatically. System-created file sets are noncontiguous and use the other CREATE FSET command defaults.

## System Parameter Changes

StorHouse Release 5.6 supports three new system parameters: RETENTION\_MODE, SNMP\_TRAPS, and SQL\_DROP\_HOLD. The release also contains changes to the XFR\_COUNT, SSPNO, VRAM\_CACHE\_MAX, XFR\_BUFSIZ, XFR\_BUFSIZ\_D, XFR\_MEMADR\_D, XRF\_MEMSIZ\_D, and SQL\_BKUP\_LIMIT system parameters.

### RETENTION\_MODE System Parameter

The RETENTION\_MODE system parameter supports two system-wide *retention enforcement levels*: BASIC and STRICT. These levels determine how strictly StorHouse manages file retention. SGI Customer Support sets the value of RETENTION\_MODE for your StorHouse system at installation and can subsequently reset the parameter at your request.

The RETENTION\_MODE system parameter is defined as follows.

- RETENTION\_  
MODE
- Specifies the retention enforcement level for all files.
- Expanded Name: Retention mode
  - Type: Static parameter





- Range: BASIC, STRICT
- Default: BASIC
- User Access: SHOW

**Table 1-3: RETENTION\_MODE Value Descriptions**

Value	Description
BASIC	<p>Prevents deletion of retained file, irrespective of user privilege, before the file's retention period expires.</p> <p>Sets the default retention attribute to ZERO when the retention period is not specified at the file and the file set level.</p> <p>Enables the retention setting specific on SET FSET to be more or less restrictive than the current setting. For example, a retention setting of 30 days can be changed to 60 days or 20 days.</p>
STRICT	<p>Prevents deletion of a retained file, irrespective of user privilege, before the file's retention period expires.</p> <p>Sets the default retention attribute to FOREVER when the retention period is not specific at the file and the file set level.</p> <p>Forces StorHouse to write retained files to WORM or compliant media.</p> <p>Forces the retention setting specific on SET FSET to be more rather than less restrictive than the current setting. For example, a retention period of 30 days can be changed to 60 days but not to 20 days.</p>

The default RETENTION\_MODE value at StorHouse installation is BASIC.

## SNMP\_TRAPS System Parameter

The SNMP\_TRAPS system parameter controls the StorHouse SNMP trap facility. It works as follows.

**SNMP\_TRAPS** Controls the StorHouse SNMP trap facility. TRUE activates the facility, allowing trap generation as specified by the SNMPCONFIG file in the home directory for account “operator.” Traps can be generated for events such as call home requests and operator messages.

- Expanded Name: SNMP trap generation
- Type: Dynamic parameter
- Range: TRUE, FALSE
- Default: FALSE
- User Access: SET, SHOW

## SQL\_DROP\_HOLD System Parameter

SQL\_DROP\_HOLD works in conjunction with the DROP TABLE and PURGE TABLE commands as follows.



**SQL\_DROP\_HOLD** Specifies the minimum number of days that must elapse between StorHouse/RM DROP TABLE and PURGE TABLE operations. An error occurs if a user tries to purge a dropped user table before that time has expired.

- Expanded Name: SQL length of time to defer actual table drop actions
- Type: Dynamic parameter (affects all engines started after the change)
- Range: 0 to 65000 days
- Default: 0 days (a purge may occur immediately after a delete operation)
- User Access: SET, SHOW

## SQL\_FTPS\_ONLY

SQL\_FTPS\_ONLY controls whether StorHouse disallows insecure (unencrypted) logins or data transfers. It works as follows.

**SQL\_FTPS\_ONLY** Specifies whether StorHouse disallows insecure (unencrypted) data transfers or logins. TRUE indicates that StorHouse disallows them. FALSE (or a missing parameter) indicates that StorHouse allows them.

- Expanded Name: SQL Secure FTP Only
- Type: Static parameter
- Range: TRUE, FALSE
- Default: FALSE
- User Access: SET, SHOW

## Changes to System Parameters

This section describes changes to the XFR\_COUNT, SSPNO, VRAM\_CACHE\_MAX, XFR\_BUFSIZ, XFR\_BUFSIZ\_D, XFR\_MEMADR\_D, XRF\_MEMSIZ\_D, EDC\_TYPE, and SQL\_BKUP\_LIMIT system parameters.

- The maximum value of the XFR\_COUNT system parameter has been increased from 64 to 128.

- The SSPNO system parameter is now a dynamic parameter with a range of 0 to 100. Setting SSPNO to 0 suspends scheduling. If SSPNO is reduced to less than the number of currently running scheduled commands, the scheduler will allow the commands to complete normally rather than terminate them to satisfy a lower SSPNO.
- VRAM\_CACHE\_MAX has a new maximum value of 67108864 bytes and a new default value of 16777216 bytes.
- XFR\_BUFSIZ has a new minimum value of 0 and a maximum value of 1240. The default is 620.
- XFR\_BUFSIZ\_D has new minimum value of 0 and a maximum value of 1240. The default value is 0.
- XFR\_MEMADR\_D has a range of 0 and a default of 0.
- XFR\_MEMSIZ\_D has a range of and a default of 0.
- EDC\_TYPE has a new maximum value of 3.
- SQL\_BKUP\_LIMIT
  - New Description: Specifies the maximum number of backup files to maintain for each database. Only the last backup is needed to recover a system, but facilities like Point-In-Time recovery require access to older backups. Therefore, this limit should be set large enough to cover the timeframe these facilities are expected to support. Old backup files are automatically deleted to make room for new ones based on the SQL\_BKUP\_LIMIT value.
  - New range: 1 through 1100 (about 30 years of daily backups)
  - New default: 366 (about 7 years of weekly backups)

## User Log Changes

StorHouse Release 5.6 supports changes to five user log records:

- security\_file
- security\_cmd
- command\_exec

- device\_error
- file\_copy

## security\_file Record

StorHouse logs a security\_file record (record type 7) when:

- Access to a file is denied due to either a missing or incorrect file password or an appropriate privilege.
- There is an attempt to shorten the retention period of a retained file.
- Two new fields, status and status\_message, have been added to this record. Table 1-4 shows the new record layout.

**Table 1-4: security\_file record layout**

Field	Type	Max. Len.	Value
record_code	num	2	7
system_id	char	6	System identifier
account_id	char	12	Account id of user who submitted the access request.
user_id	num	10	User id of user's session.
log_time	time	20	Time record was logged.
start_time	time	20	Time request was received.
end_time	time	20	Time request processing was completed. (This value is always the same as start_time.)
command	char	255	Description of the activity that caused the violation. This may be a StorHouse command, Callable Interface function, or another action (for example, and internal StorHouse process).
file_sysid	num	5	System identifier specified in command on which the violation attempt was made. If no file identifier was specified or the file identifier had not been

			located before the violation attempt was recognized, this field contains a 0 (zero).
file_fno	num	10	File number specified in the command on which the violation attempt was made. If no file identifier was specified or the file identifier had not been located before the violation attempt was recognized, this field contains a 0 (zero).
filename	char	56	Name of the file for which required password was not specified.
group	char	8	Name of the group in which the specified file is located.
status	num	5	StorHouse status code that identifies error conditions.
status_message	char	132	Text message corresponding to the status code in the status field.

## security\_cmd Record

In addition to its other functions, the security\_cmd record (record type 8) reports retention violations (for example, an attempt to delete a retained file).

## command\_exec Record

The following values have been added to the list of command identifiers in the StorHouse user log command\_exec record (record type 9):

- 1711 for the CATALOG DEVICE command
- 1411 for the REPLICATE command
- 1412 to indicate [REPLICATION\_SERVER]. This identifier specifies that a file copy action occurred on a secondary StorHouse system due to a REPLICATE command on a primary StorHouse system.



## device\_error Record

The following identifiers have been added to the list of values for the dtype field in the user log device\_error (record type 14) record.

- FILESYST
- CENTERA
- SPECTRANT

## file\_copy Record

The following identifiers have been added to the list of values for the command field in the user log file\_copy (record type 22) record.

- 7 for the REPLICATE command
- 8 to indicate [REPLICATION\_SERVER]. This identifier specifies that a file copy action occurred on a secondary StorHouse system due to a REPLICATE command on the primary StorHouse system.

This record applies to the ARCHIVE, CREATE BACKUP, CREATE PRIMARY, RELOCATE, RECOVER VOLUME, RETIRE VOLUME, and REPLICATE commands.

## New Status Codes

### 0333 **XBSEGLIMIT Segment limit exceeded.**

**Explanation:** The segment limit was exceeded while writing a file. The file was aborted.

### 0334 **XBINVDEV Invalid device specification.**

**Explanation:** The device specification is invalid.

0335 **XBDEVNF Device not found.**

*Explanation:* The device was not found

0336 **XBUNLABELED Volume is not labeled.**

*Explanation:* The volume is not labeled.

0337 **XBLABELED Volume is labeled.**

*Explanation:* The volume is labeled.

0338 **XBINVUFF Invalid UFF.**

*Explanation:* The files universal file finder (UFF) is invalid.

0339 **XBEDENF External directory not found.**

*Explanation:* The requested entry was not found in the external directory for the device or volume.

0340 **XBDEEXISTS External directory entry exists.**

*Explanation:* The entry exists in the external directory for the device or volume.

0341 **XBFILENF File not found.**

*Explanation:* The file was not found.

0342 **XBFILEEXISTS File exists.**

*Explanation:* The file exists.

4565 **XREXTBUSY Extent busy.**

*Explanation:* The extent is currently busy and cannot be used for the requested operation.

4566 **XREXTCHAIN Problem detected in the file's extent chain.**

*Explanation:* A problem was detected in the file's extent chain.

5673 **XWRETEN Request rejected, operation prohibited by retention rules.**

*Explanation:* File retention rules prohibit the requested operation. Examples include attempting to delete a file whose retention period has not expired, and



attempting to shorten a file's retention period.

**4559 %RM-E-XRINACCVOL, Selected Volume is inaccessible.**

**Explanation:** The volume is in an offline library storage module in an STK SL8500 library and therefore inaccessible.

**4563 XROMIR Library condition requires manual intervention.**

**Explanation:** StorHouse generates this operator message when it detects a library condition that may be cleared by manual intervention. The error text given should be used to determine the exact condition and which actions to take. For more information, including which actions to take, refer to the *StorHouse System Operator's Guide*.

**4564 XRTRANSCOND Transient condition.**

**Explanation:** A transient condition has caused the operation to fail. Retry the operation.

**4959 XTRPLCLNF Replication class not found.**

**Explanation:** The replication class does not exist. Verify that the correct replication class name was specified.

**4960 XTRPLSYS Replication name invalid.**

**Explanation:** The replication system does not exist. Verify that the correct system name was specified.

**4961 XTRPLNCONN Replication connect failed.**

**Explanation:** Replication failed to connect to the destination StorHouse. The condition may be permanent (e.g., a configuration error) or temporary (e.g., destination StorHouse is down).

**4962 XTRPLLOGIN Replication login failed.**

**Explanation:** Replication failed to log in to the destination StorHouse. Verify that the account and password are defined on the destination StorHouse.

**4963 XTRPLNDISC Replication network disconnected.**

**Explanation:** Replication was disconnected from the destination StorHouse. This

is usually caused by an error with the network or the destination StorHouse.

**4964 XTRPCNFLCT Replication file or extent conflict.**

**Explanation:** Replication detected a conflict between a source file or its extents and an existing file on the destination StorHouse. File on the source and destination StorHouse systems can get out of sync if they are updated independently.

**4965 %TC-E-XTFDIS, File is disabled.**

**Explanation:** The file is unusable because it is software-disabled, hardware-disabled, or damaged in some other way.

**4966 XTNOREMOVE /NOREMOVE prohibits file from being removed.**

**Explanation:** The file could not be removed because /NOREMOVE was specified on the command.

## Operator Messages

This section describes new and updated operator messages.

### New Messages

The following are new operator messages.

**XROERR LxxEyy, EXCHANGE STATION TRAY IS MISSING; REPLACE, THEN REPLY 'C' – CONTINUE 'E' – ERROR**

This message appears when a tray is missing during an exchange station operation. Replace the tray used with the exchange station.

**XROINFO Fxx, VOLUME IS WRITE-PROTECTED**

The magnetic disk drive Fxx is preventing any further writing. StorHouse sets the magnetic disk drive read-only and generates the informational message: XROINFO, Fxx, DEVICE IS UP, READ\_ONLY, and automatically attempts to activate the Call Home error reporting facility. Ask your system administrator



to contact your SGI customer support representative.

**XROINFO XROINFO, LxxDyy, LIBRARY COMPONENT ERROR, VOLUME=vid**

This message appears when a library component error occurred while trying to access drive LxxDyy. Because the library storage module was taken offline, the specified drive is inaccessible even though it remains in the up mode. This message pertains to libraries such as the STK SL8500, which contains multiple levels. The library uses a transport mechanism to move a volume from a drive on one level in the library to another. If this mechanism becomes inoperable, a specific drive cannot be accessed even when it is in the up mode.

**XROINFO XROINFO, LxxDyy, DEVICE IS UP, INACCESSIBLE**

This message appears when the system operator or a user with the proper privilege tries to execute an UP DEVICE command for drive LxxDyy. Because the library storage module was taken offline, the specified drive is inaccessible even though the drive itself is not in the down mode. This message pertains to libraries such as the STK SL8500, which contains multiple levels. The library uses a transport mechanism to move a volume from a drive on one level in the library to another. If this mechanism becomes inoperable, a specific drive cannot be accessed even when it is in the up mode.

**XROINFO XROINFO, Lxx, EXCHANGE STATION IS OPEN**

This message appears for certain libraries when the exchange station is open during initialization of library device Lxx. The system shuts down the library device and generates the informational message: XROINFO, Lxx, DEVICE IS DOWN. Close the exchange station and try to bring down the library.

**XROINFO XROINFO, Lxx, EXCHANGE STATION IS FULL**

This message appears for certain libraries when the exchange station is full during initialization of library device Lxx. The system shuts down the library device and generates the informational message: XROINFO, Lxx, DEVICE IS DOWN. Empty the exchange station and try to bring the library online.

**XROMIR Lxx, EXCHANGE STATION IS OPEN; CLOSE THE EXCHANGE STATION DOOR**

This message appears when StorHouse detects a condition with a library that

requires the operator to take some action to correct. If the condition has been corrected, reply with “C” to continue/retry. If the condition cannot be corrected, respond with “D” to down the library.

If you reply with “D,” StorHouse displays the following XROINFO message.

**XROINFO Lxx, MANUAL INTERVENTION REQUEST REJECTED**

**XROINFO Fxx, INSUFFICIENT SPACE ON DEVICE**

The magnetic disk drive Fxx has unexpectedly run out of physical space on the device. StorHouse writelocks the magnetic disk drive and generates the informational message: XROINFO, Fxx, DEVICE IS UP, WRITE\_LOCKED, and automatically attempts to activate the Call Home error reporting facility.

Ask your system administrator to contact your SGI customer support representative.

## Updated Operator Messages

The following operator messages have been changed.

- XRORST RESET-LIBRARY Lxx now accepts a “D” - Down response instead of an “E” - Error response.
- XROINFO Lxx, OPERATOR REFUSED RESET REQUEST now accepts a “D” - Down response instead of an “E” - Error response.
- The WRITE\_LOCKED mode can now be displayed by the XROINFO, device, DEVICE is state [,MODE] operator message.



## Changes to the StorHouse Interface

This chapter describes the StorHouse Release 5.6 changes to Command Language commands, Generic Callable Interface functions, and mainframe Callable Interface functions.

### StorHouse Command Language

StorHouse Release 5.6 supports the following changes to StorHouse Command Language:

- Addition of the date keywords YESTERDAY, TODAY, TOMORROW, LAST, and NEXT in the absolute time specification.
- New CATALOG DEVICE, REPLICATE, SHOW CONFIGURATION, VALIDATE DEVICE, and VALIDATE FILE commands.
- Updated BACKUP, CREATE FILE, CREATE FSET, CREATE VSET, ERASE, EXTRACT DIRECTORY, IMPORT, MIGRATE, PUT, RECOVER DEVICE, RECOVER VOLUME, RELOCATE, REMOVE FILE, RETIRE VOLUME, RESTORE DIRECTORY, SET DEVICE, SCHEDULE, SET FILE, SET FSET, SET VOLUME, SET VSET, SHOW DEVICE, SHOW

FILE, SHOW FSET, SHOW VOLUME, SHOW VSET, STAGE, VALIDATE DEVICE, and VALIDATE FILE commands.

- Major enhancements to the VALIDATE VOLUME command.
- Addition of the following additional frequencies to the /SCHEDULE modifier: SUNDAY, MONDAY, TUESDAY, WEDNESDAY, THURSDAY, FRIDAY, SATURDAY, NOSUNDAY, NOMONDAY, NOTUESDAY, NOWEDNESDAY, NOTHURSDAY, NOFRIDAY, NOSATURDAY, and MINUTE.

This chapter documents new commands and the VALIDATE VOLUME command in their entirety. It describes only the new features for updated commands.

## CATALOG DEVICE

CATALOG DEVICE initiates device scan recovery for a level F device.

### Format

CATALOG DEVICE did

COMMAND FORMAT SUMMARY					
COMMAND, PARAMETER, OR MODIFIER	REQUIRED COMMAND PRIVILEGE	REQUIRED GROUP ACCESS	REQUIRED FILE ACCESS	MINIMUM ACCOUNT ACCESS	DEFAULT
CATALOG DEVICE	ALLOCATION + SYSTEM	-	-	-	-
/CONFIRM	-	-	-	-	/CONFIRM
/REPORT	-	-	-	-	-
/WAIT	-	-	-	-	-
did	-	-	-	-	(required)

### Description

CATALOG DEVICE initiates device scan recovery for a level F device. It scans the level F device to be recovered to obtain information about the StorHouse files, extents, volume sets, and file sets on the device. Then it catalogs those entities.

One CATALOG DEVICE command does device scan recovery on one device.



#### Note the following:

- The device to be cataloged must be online and uncataloged.
- During the device scan, if the command encounters any files that are already in the StorHouse catalog, it skips them.
- Multiple CATALOG DEVICE commands can be executed concurrently as long as each specifies a different device.

### Parameters

**did** Specifies the device identification code (did) of the level F device to be scanned.

- **FORMAT:** F{unit\_number}
- **DEFAULT:** None; you must specify this parameter.

### Command Modifiers

**/CONFIRM** Controls whether StorHouse asks you to confirm the command.

- **FORMAT:** /CONFIRM or /NOCONFIRM
- **DEFAULT:** /CONFIRM

When the system requests a confirmation, enter YES (also Y or YE) or NO (also N). If you press R or enter any characters other than those described as a YES response, StorHouse interprets them as NO.

**/REPORT** Controls the generation of special text responses for the completion of significant actions. /REPORT instructs StorHouse to generate a text response. /NOREPORT instructs StorHouse not to generate a text response. The report generated for this command only includes summary information (for example, the number of files added, modified, and skipped). It does not contain the names of all files that are processed.

- **FORMAT:** /REPORT or /NOREPORT
- **DEFAULT:** /NOREPORT

## ■ ■ ■ ■ Chapter 2 – Changes to the StorHouse Interface

**/WAIT** Instructs StorHouse to wait for a locked file to be unlocked before attempting to use it in the command execution. Without /WAIT, StorHouse returns an error status if it encounters a locked file.

- **FORMAT:** /WAIT
- **DEFAULT:** No /WAIT modifier. StorHouse aborts the command or processing of an item if the specified file version is not available.

### Examples

To initiate device scan recovery for a level F device with a device identification code of F01 and generate text responses, enter:

```
? CATALOG DEVICE F01 /REPORT
```



## CREATE FILE

StorHouse Release 5.6 adds a new parameter modifier, `/RETENTION`, to the `CREATE FILE` command.

### Parameter Modifiers

`/RETENTION` Specifies the retention attribute (retention period) for the file being created. (If force retention is set at the file set level, the file assumes the file set retention value.)

- **FORMAT:**

Option	Description
<code>/RETENTION=DEFAULT</code>	Sets the retention period to the default value.
<code>/RETENTION=number_of_days</code>	<p>Sets the retention period to the specified number of days. The retention period ends when the current data is beyond the file's last modification date plus the specified retention value.</p> <p>A value of 0 indicates no retention period (same as specifying ZERO).</p> <p>Example: <code>/RETENTION=3</code></p> <p>In this example, the period for a file that was last modified at 11 p.m. on December 12 expires at 11 p.m. on December 15.</p>
<code>/RETENTION=ZERO</code>	Sets no retention period, which indicates the file may be deleted.
<code>/RETENTION=FOREVER</code>	Sets an infinite retention period, which indicates the file may never be deleted.

- **DEFAULT:** If you omit `/RETENTION` or specify `/RETENTION=DEFAULT`, StorHouse determines the file's default retention attribute as follows:

## ■ ■ ■ ■ Chapter 2 – Changes to the StorHouse Interface

- If the file's resident file set has a retention attribute equal to FOREVER, ZERO, or a specified number of days, the file set retention attribute determines the default file retention attribute.
  - If the file's resident file set has a retention attribute of DEFAULT, the RETENTION\_MODE system parameter determines the default file retention attribute. If RETENTION\_MODE is set to BASIC, the default file retention is ZERO. If RETENTION\_MODE is set to STRICT, the default file retention is FOREVER.
- RESTRICTIONS: None



## CREATE FSET

StorHouse Release 5.6 adds the parameter modifiers, `/FORCE_RETENTION`, `/RETENTION`, and `/RPL_CLASS` to the CREATE FSET command.

### Parameter Modifiers

`/FORCE_RETENTION` Specifies whether the file set retention value overrides the file retention value (for files in this file set) explicitly supplied by an application at file create time. `/NOFORCE_RETENTION` indicates that the file set retention value does not override the file retention value.

- **FORMAT:**
  - `/FORCE_RETENTION`
  - `/NOFORCE_RETENTION`
- **DEFAULT:** `/NOFORCE_RETENTION`
- **RESTRICTIONS:** `/FORCE_RETENTION` is valid for primary file sets only.

**/RETENTION** Specifies the file set retention attribute (retention period). The file set retention attribute applies only to files (in the file set) that are created without a specified retention value, or whenever **/FORCE\_RETENTION** is in effect on the file set.

■ **FORMAT:**

Option	Description
<b>/RETENTION=DEFAULT</b>	Sets the retention period to the default value.
<b>/RETENTION=number_of_days</b>	<p>Sets the retention period to the specified number of days. The retention period ends when the current date is beyond the file's <code>last_modified</code> date plus the retention value.</p> <p>A value of 0 indicates no retention period (same as specifying ZERO).</p> <p>Example: <code>/RETENTION=3</code></p> <p>In this example, the period for a file that was last modified at 11 p.m. on December 12 expires at 11 p.m. on December 15.</p>
<b>/RETENTION=ZERO</b>	Sets no retention period.
<b>/RETENTION=FOREVER</b>	Sets an infinite retention period.

- **DEFAULT:** If you omit `/RETENTION`, the default retention attribute is `DEFAULT`.
- **RESTRICTIONS:** `/RETENTION` is valid for primary file sets only.



**/RPL\_CLASS** Specifies the name of the replication class for files written to this file set. A replication class is a collection of replication-related information (system, file set, volume set, and link names and network device identification) about the target StorHouse system. If the specified replication class does not exist, StorHouse generates a warning message.

- **FORMAT:** /RPL\_CLASS=replication\_class\_name  
A replication\_class\_name can consist of from 1 to 8 of the following ASCII characters: A-Z, 0-9, \_, and \$. StorHouse always forces replication class names to uppercase, even when enclosed in quotes.
- **DEFAULT:** If you omit /RPL\_CLASS, the default is no assigned replication class.
- **RESTRICTIONS:** /RPL\_CLASS is valid for primary file sets only.

### Examples

- To create the contiguous file set USERFSET in the volume set USERVSET with a /SIZE of 1000 MB, a /LIMIT of 4000 MB, and a /RETENTION of 60 days, enter:  

```
? CREATE FSET USERFSET /VSET=USERVSET /SIZE=1000M /LIMIT=4000M /RETENTION=60
```

## CREATE VSET

StorHouse Release 5.6 expands the capability of the CREATE VSET /LIBRARY modifier to allow creation of volume sets on level F.

### Parameter Modifier

**/LIBRARY** For level L, specifies the library device where the volume set will be created. StorHouse allocates volumes for the volume set from the free pool associated with the library device.

For level F, specifies the free pool indicator, which consists of the second, third, and fourth characters of a level F free pool volume set name. StorHouse selects one or more level F devices (volumes) in this free pool and allocates them to the new volume set.

- **FORMAT:** /LIBRARY=Lxx or /LIBRARY=Fxx  
where Lxx is the device identification code for the library-level device with the unit number xx (for example, L00), and Fxx is the free pool indicator (for example, F00).
- **DEFAULT:** If you omit /LIBRARY, the default is the value of the DEFAULT\_LD system parameter. If you do not specify /LIBRARY and DEFAULT\_LD does not specify a valid device, StorHouse rejects the command.

### Example

Before creating a level F volume set, administrators must know the name of an available level F free pool volume set for the desired media type. To obtain the name of a level F free pool volume set and determine which volumes are allocated to that free pool, run the SHOW DEVICE command as follows:

```
? SHOW DEVICE F* /FULL  
  
DEVICE=F00 STATUS=UP MEDIA=(MIA:RW) VSET=MAGDISK  
DIRECTORY=PRIMARY SIZE=63488000KB  
  
DEVICE=F01 STATUS=UP MEDIA=(MIA:RW) VSET=MAGDISK  
DIRECTORY=PRIMARY SIZE=63488000KB  
  
DEVICE=F02 STATUS=UP MEDIA=(MEA:RW) VSET=.FOOMEA  
DIRECTORY=none SIZE=9989582KB
```



```
DEVICE=F03 STATUS=UP MEDIA=(MEA:RW) VSET=.FO0MEA  
DIRECTORY=none SIZE=9989582KB
```

```
DEVICE=F04 STATUS=UP MEDIA=(MEA:RW) VSET=.FO0MEA  
DIRECTORY=PRIMARY SIZE=9989582KB
```

In the preceding example, there are five level F devices. Devices F00 and F01 are allocated to the MAGDISK volume set. Devices F02, F03, and F04 are assigned to the free pool volume set .FO0MEA. You know that .FO0MEA is a free pool volume set because its name begins with a period (.).

To create a level F volume set that uses volumes from free pool .FO0MEA and has an initial size requirement of 12 GB, enter:

```
? CREATE VSET MYVSET /LIBRARY=F00 /MEDIA=MEA /SIZE=12GB
```

Note the following:

- /MEDIA specifies the fifth, sixth, and seventh character of the free pool volume set name. These characters indicate the media and recording type of the volumes in free pool .FO0MEA.
- StorHouse always allocates whole volumes from the free pool to the volume set being created.
- The output of the SHOW DEVICE example indicates that each level F device in free pool .FO0MEA is slightly less than 10 GB. Because the CREATE VSET /SIZE modifier specifies a requirement of 12 GB, StorHouse must allocate two volumes from free pool .FO0MEA to satisfy the 12-GB size request.

## EXTRACT DIRECTORY

StorHouse Release 5.6 adds the capability to extract file set and volume set information for files on level F. It also:

- Adds the parameter modifier, `/RETENTION`, to `EXTRACT DIRECTORY`.
- Shows the number of non-cataloged volume sides and pending extents skipped during the extract.

### Parameter Modifier

`/RETENTION` Specifies the retention attribute (retention period) for the files created by the `EXTRACT DIRECTORY` operation.

- **FORMAT:**

Option	Description
<code>/RETENTION=DEFAULT</code>	Sets the retention period to the default value.
<code>/RETENTION=number_of_days</code>	<p>Sets the file retention period to the specified number of days. The retention period ends when the current date is beyond the file's <code>last_modified</code> date plus the retention value.</p> <p>A value of 0 indicates no retention period (same as specifying <code>ZERO</code>).</p> <p>Example: <code>/RETENTION=3</code></p> <p>In this example, the retention period is 3 days. The retention period for a file that was last modified at 11 p.m. on December 12 expires at 11 p.m. on December 15.</p>
<code>/RETENTION=ZERO</code>	Sets no retention period, which indicates the file may be deleted.
<code>/RETENTION=FOREVER</code>	Sets an infinite retention period, which indicates the file may never be deleted.





- **DEFAULT:** If you omit /RETENTION or specify /RETENTION=DEFAULT, StorHouse determines the default file retention attribute as follows:
  - If the file's resident file set has a retention attribute equal to FOREVER, ZERO, or a specified number of days, the file set retention attribute determines the default file retention attribute.
  - If the file's resident file set has a retention attribute of DEFAULT, the RETENTION\_MODE system parameter determines the default file retention attribute. If RETENTION\_MODE is set to BASIC, the default file retention is ZERO. If RETENTION\_MODE is set to STRICT, the default file retention is FOREVER.
- **RESTRICTIONS:** None.

### Example:

Enter the following command to initiate a directory extraction that:

- Saves all current information in StorHouse files in group SMDIR1 directly in volume set SMDIR1 and file set SMDIR1.
- Requests text responses.
- Retains the files created by the EXTRACT DIRECTORY operation for 180 days.

```
? EXTRACT DIRECTORY /REPORT /GROUP=SMDIR1 /VTF=DIR  
/TO_VSET=SMDIR1 /TO_FSET=SMDIR1 /RETENTION=180
```

## IMPORT

Release 5.6 adds the `/VALIDATE_DATE` modifier to set the `VALIDATE_DATE` timestamps of imported volumes.

`/VALIDATE_DATE` Sets the `VALIDATE_DATE` timestamps of the imported volumes to the current date/time or the specified date/time. For a full description of the absolute time and delta time data fields, see Appendix B, “Data Field, Parameter, and Modifier Definitions,” in the Command Language Reference Manual.

- **FORMAT:**
  - `/VALIDATE_DATE`
  - `/VALIDATE_DATE=absolute_time`
  - `/VALIDATE_DATE=delta_time`
- **DEFAULT:** If you omit `/VALIDATE_DATE`, StorHouse sets the `VALIDATE_DATE` timestamps of the imported volumes to the date/time the volumes were initialized.



## MIGRATE

StorHouse Release 5.6 changes the MIGRATE command as follows:

- MIGRATE /BY\_VSET always creates noncontiguous destination file sets.
- MIGRATE supports a file name parameter. MIGRATE filename migrates extents of the specified file(s) from the performance buffer.
- For the complete MIGRATE command description, refer to the *Command Language Reference Manual*.

### Format

MIGRATE

MIGRATE filename

MIGRATE /BLANKS=number did

MIGRATE /BY\_VSET source\_vset\_name destination\_vset\_name

COMMAND FORMAT SUMMARY					
COMMAND, PARAMETER, OR MODIFIER	REQUIRED COMMAND PRIVILEGE	REQUIRED GROUP ACCESS	REQUIRED FILE ACCESS	MINIMUM ACCOUNT ACCESS	DEFAULT
MIGRATE	SYSTEM	-	-	-	-
/ALL	-	-	-	-	-
/BLANKS=...	-	-	-	-	-
/BY_VSET	COPY + ALLOCATION	-	-	-	-
did	-	-	-	-	See text
/BYPASS_LABEL	-	-	-	-	-
/MEDIA=...	-	-	-	-	See text
filename	-	D	D	-	See text
/GROUP=...	SETGROUP	D	-	-	Current default
/NOSPECIAL	-	-	-	-	-
/PASSWORDS=...	-	-	-	-	-
/REPORT	-	-	-	-	-
/VERSION=...	-	-	-	-	/VERSION=0
/WAIT	-	-	-	-	-
source_vset_name	-	-	-	-	See text
/CONFIRM	-	-	-	-	-
/MINIMUM	-	-	-	-	(Required)
/ORDER_BY=...	-	-	-	-	/ORDER_BY=LAST
/REPORT	-	-	-	-	-
/WAIT	-	-	-	-	/WAIT
destination_vset_name	-	-	-	-	See text



## Updated Description

The MIGRATE command now migrates file extents of specific files from the performance buffer.

The MIGRATE filename and MIGRATE /BY\_VSET command returns the command prompt after it completes the migration.

## Performance Buffer Migration of Specific Files

If you specify MIGRATE with a filename, the system migrates extents of the specified file from the performance buffer. If a copy of an extent of a selected file already exists in the performance buffer and in its primary file set, StorHouse simply removes the performance buffer copy. The MIGRATE filename command does not migrate extents with a status of NEW (that is, extents without a copy in their primary file set).

## Parameters

**filename** Specifies the name of the file whose extents will be migrated from the performance buffer.

- **FORMAT:**
  - filename
  - partial\_filename\*
  - \*

The wild card is valid only if you specify it as the last or only character in the filename.

- **DEFAULT:** None.
- **RESTRICTIONS:** This parameter is valid only if you do not specify /BLANKS or /BY\_VSET.
- **ACCESS REQUIREMENTS:** Delete access to the file and group.

## Command Modifiers

- /BLANKS** ■ RESTRICTIONS: /BLANKS is mutually exclusive with the filename parameter and /BY\_VSET.
- /BY\_VSET** ■ RESTRICTIONS: /BY\_VSET is mutually exclusive with the filename parameter and /BLANKS.

## filename Parameter Modifiers

- /GROUP** Specifies a file access group name and, optionally, group passwords.
  - **FORMAT:**
    - /GROUP=groupname<:::deletepw>
    - /GROUP=partial\_groupname\*
    - /GROUP=\*
  - **DEFAULT:**
    - If you omit /GROUP, the default is your current default group and default access rights.
    - If you specify the current default group name and omit the delete password, the defaults for your group access rights apply.
    - If you specify a group name that is not the current default group and omit the delete password, the delete password defaults to null.
  - **ACCESS REQUIREMENTS:** Delete access to the group. You must specify the group delete access password unless:
    - The group is not protected by a delete password.
    - Your privilege bypasses delete access password checks.
    - Your default access to the group includes delete access.
  - **PRIVILEGE:** You must have SETGROUP privilege to specify any group except your default group.

**/NOSPECIAL** Specifies only data (non-control) extents are to be migrated.

- **FORMAT:** /NOSPECIAL
- **DEFAULT:** If you omit /NOSPECIAL, all extents of a file are migrated.

**/PASSWORDS** Specifies the file's delete password.

- **FORMAT:** /PASSWORDS=::deletepw  
You must specify a delete password unless:
  - The file is not protected by a delete access password.
  - You have a privilege that bypasses delete access password checks.
- **DEFAULT:** If you omit /PASSWORDS, the password defaults to nulls.
- **RESTRICTIONS:** /PASSWORDS is not allowed if the file or group name specification includes a wild card.

**/REPORT** Controls generation of text responses for MIGRATE filename.

/REPORT instructs StorHouse to generate a text response. /NOREPORT instructs StorHouse not to generate a text response.

- **FORMAT:**
  - /REPORT
  - /NOREPORT
- **DEFAULT:** /NOREPORT

**/VERSION** Specifies the relative version number of the file to be migrated

- **FORMAT:**
  - /VERSION=version
  - /VERSION=\*
- **DEFAULT:** /VERSION=0

**/WAIT** Instructs StorHouse to wait for a locked file to be unlocked before attempting to use it in the command execution.

## ■ ■ ■ ■ Chapter 2 – Changes to the StorHouse Interface

- **FORMAT:**
  - /WAIT
  - /NOWAIT
- **DEFAULT:** No/WAIT modifier. StorHouse aborts the command or processing of an item if the specified file version is not available.

### Example

To migrate file extents of the file USERFILE, enter:

```
? MIGRATE USERFILE
```

If a copy of an extent of USERFILE exists in the performance buffer and in its primary file set, StorHouse simply removes the performance buffer copy.



# PUT

StorHouse Release 5.6 adds the parameter modifier, `/RETENTION`, to the PUT command.

## Parameter Modifier

`/RETENTION` Specifies the retention attribute (retention period) for the file.

■ **FORMAT:**

Option	Description
<code>/RETENTION=DEFAULT</code>	Sets the retention period of the specified file version to the default value.
<code>/RETENTION=number_of_days</code>	<p>Sets the retention period of the specified file version to the specified number of days. The retention period ends when the current date is beyond the file's last_modified date plus the specified retention value.</p> <p>A value of 0 indicates no retention period (same as specifying ZERO).</p> <p>Example: <code>/RETENTION=3</code></p> <p>In this example, the retention period is 3 days. The retention period for a file that was last modified at 1 p.m. on December 12 expires at 11 p.m. on December 15.</p>
<code>/RETENTION=ZERO</code>	Sets no retention period, which indicates the specified file version may be deleted.
<code>/RETENTION=FOREVER</code>	Sets an infinite retention period, which indicates the specified file version may never be deleted.

- **DEFAULT:** If you omit `/RETENTION`, there is no change to the file's current retention value.
- **RESTRICTIONS:** You may only specify a more restrictive value. For example,

if the current file retention is 100 days, you may specify 200 days but not 50 days.

## RECOVER DEVICE

Release 5.6 adds support for the `/BYPASS_LABEL`, `/LABEL`, and `/VALIDATE_DATE` modifiers. Moreover, the command description section is changed as follows:

### Description

This command initiates the recovery of a storage device for which directory information was destroyed. The storage device must be in the UP state with recovery required. Any blank volumes are normally ejected from the library.

The command locates all volumes in the specified storage device and adds them to the directory. Following a RECOVER DEVICE command, the volumes must be cataloged before their files can be accessed. (See the CATALOG VSET description.) If you recover a volume in a library that cannot be written in that library (such as a Hitachi-1 volume in a Hitachi-2 library), StorHouse will writelock that volume. Similarly, you cannot recover an empty (free pool) volume that cannot be written in the library.

### Note the Following:

When a tape cleaning cartridge is loaded into a drive, the drive is cleaned automatically. If you do not want to unnecessarily use one pass of your tape cleaning cartridges during library recovery, remove all tape cleaning cartridges from the library before beginning recovery operations.

StorHouse allows only one RECOVER DEVICE command to execute at a time. However, StorHouse will accept and process user requests for one device while it is recovering another.

**Labeling Blank Volumes.** The RECOVER DEVICE command can also be used on an operational system to label blank volumes. The advantage of using RECOVER DEVICE over the MIGRATE /BLANKS command is that a large number of blank volumes can be loaded directly into the library, thereby bypassing the exchange station.

### Note the Following:

Labeling blank volumes with RECOVER DEVICE is recommended only for barcode media.

1. Take the library or StorHouse down.
2. Load the blank volumes to be labeled into the library.
3. Bring the library or StorHouse back up.
4. Issue a RECOVER DEVICE /LABEL command.

It is not necessary to remove any existing volumes from the library except possibly to make room for the blank volumes. The library supports normal operations while the RECOVER DEVICE command is running.

### Parameter Modifiers

**/BYPASS\_LABEL** When used with /LABEL, /BYPASS\_LABEL instructs StorHouse to label non-StorHouse volumes.

**CAUTION:** /BYPASS\_LABEL forces StorHouse to bypass its standard label checking, which is designed to prevent overwriting non-StorHouse volumes. /BYPASS\_LABEL should be used very carefully, especially with a shared library containing volumes from non-StorHouse applications.

- **FORMAT:** /BYPASS\_LABEL
- **DEFAULT:** If you omit /BYPASS\_LABEL, StorHouse will eject any non-StorHouse volumes from the library.
- **RESTRICTIONS:** /BYPASS\_LABEL only applies when /LABEL is also specified.

**/LABEL** By default, RECOVER DEVICE ejects non-StorHouse volumes including blank volumes, from the library. If /LABEL is specified, StorHouse will label blank volumes instead. If /BYPASS\_LABEL is also specified, StorHouse will also label non-StorHouse volumes instead of ejecting them.

- **FORMAT:** /LABEL
- **DEFAULT:** If you omit /LABEL, StorHouse will eject any non-StorHouse volumes, including blank volumes, from the library.

`/VALIDATE_DATE` Sets the VALIDATE\_DATE timestamps of the recovered volumes to the current date/time or the specified date/time. For a full description of the absolute time and delta time data fields, see Appendix B, “Data Field, Parameter, and Modifier Definitions,” in the *Command Language Reference Manual*.

- **FORMAT:**
  - `/VALIDATE_DATE`
  - `/VALIDATE_DATE=absolute_time`
  - `/VALIDATE_DATE=delta_time`
- **DEFAULT:** If you omit `/VALIDATE_DATE`, StorHouse sets the VALIDATE\_DATE timestamps of the recovered volumes to the date/time the volumes were initialized.

## RECOVER VOLUME

StorHouse Release 5.6 makes the following changes to the RECOVER VOLUME command:

- Adds the parameter modifier, /ERASE
- Reports the extents and bytes copied
- Allows users to execute more than one RECOVER VOLUME command at a time.

### Parameter Modifier

**/ERASE** Indicates that the command is to erase each volume instead of exporting it after moving all extents off the volume and uncataloging it. StorHouse returns an error if you specify /ERASE and the medium that you specified with the vid parameter is not erasable.

- **FORMAT:** /ERASE
- **DEFAULT:** If you do not specify /ERASE, the command exports volumes instead of erasing them.

### Example

To recover volume OEB01234567 and erase it after moving all extents off the volume and uncataloging it, enter:

```
? RECOVER VOLUME OEB01234567 /ERASE
```

## RELOCATE

StorHouse Release 5.6 adds the parameter modifiers, /VOLUME, /UNUSED, and /USED to the RELOCATE command.

**/UNUSED** If specified, only those file versions with a last access date and time value earlier than that specified will be relocated. See the “data\_field date-time” topic for more information about absolute and delta time specifications.

- **FORMAT:**
  - /UNUSED=absolute\_time
  - /UNUSED=delta\_time
- **/DEFAULT:** If you omit both /UNUSED and /USED, the system does not select files based on last access date and time.

**/USED** If specified, only those file versions with a last access date and time value the same as or later than that specified will be relocated. See the “data\_field date-time” topic for more information about absolute and delta time specifications.

- **FORMAT:**
  - /USED=absolute\_time
  - /USED=delta\_time
- **/DEFAULT:** If you omit both /UNUSED and /USED, the system does not select files based on last access date and time.

### Example

To relocate all files on the level F volume MEAF04:A to the file set FSET1 in the volume set VSET1, enter:

```
? RELOCATE * /VOLUME=MEAF04:A /TO_VSET=VSET1  
/TO_FSET=FSET1
```

Note that the volume identification code for the level F volume specifies a side indicator (A).



- /VOLUME** Specifies the volume identifier (vid) of the volume where the source file is located.
- **FORMAT:** /VOLUME={media\_type}{recording\_type}{volume\_label}:{side}
  - **DEFAULT:** If you omit /VOLUME, the volume is not used to select source files.
  - **RESTRICTIONS:**
    - Files cannot be relocated from one side of a volume to the other side of the same volume.
    - /VOLUME is mutually exclusive with /DIRECTORY and /VSET.

## REMOVE FILE

Release 5.6 adds the `/BEFORE` modifier to the `REMOVE FILE` command. If specified, `/BEFORE` directs `REMOVE FILE` to remove files only if they were deleted before the specified (absolute or relative) date/time.

`/BEFORE` Selects file versions for removal with a delete date and time value that is the same as or earlier than the value you specify.

- **FORMAT:** `/BEFORE`
  - `/BEFORE=absolute_time`
  - `/BEFORE=delta_time`

*Absolute time* specifies a calendar date and clock time. Specify absolute time as:

`<<day>-<month>-<year>><:<hour><:<minute><:<second>>>`

or

`<<date keyword><:<minute><:<second>>>`

If you specify `/BEFORE` but do not specify a day, month, or year, then the default is the current day, month, or year. If you do not specify an hour, minute, or second, then the subfield defaults to zero.

*Delta time* specifies the amount of time since or before an event. If you specify a delta time, the system subtracts it from the current date and time to determine the absolute date and time to use.

Specify delta time as follows (the leading `D` is required):

`D<days>-><hours<:minutes<:seconds>>>`

The default for any field that is not specified is zero. For a full description of the absolute time and delta time data fields, see Appendix B, “Data Field, Parameter, and Modifier Definitions.”

- **DEFAULT:** If you omit `/BEFORE`, the system does not select files for removal based on deletion date and time.



# REPLICATE

StorHouse Release 5.6 adds the REPLICATE command to the StorHouse Command Language. REPLICATE copies eligible files from one StorHouse system to another.

## Format

REPLICATE filename

COMMAND FORMAT SUMMARY					
COMMAND, PARAMETER, OR MODIFIER	REQUIRED COMMAND PRIVILEGE	REQUIRED GROUP ACCESS	REQUIRED FILE ACCESS	MINIMUM ACCOUNT ACCESS	DEFAULT
REPLICATE	SYSTEM	-	-	-	-
/CHECK					
/CONFIRM	-	-	-	-	-
/DEF_RPL_CLASS=...	-	-	-	-	-
/PREVIEW	-	-	-	-	-
/RECORD	-	-	-	-	/RECORD
/REPORT	-	-	-	-	-
/WAIT	-	-	-	-	-
filename	-	-	-	-	(required)
/FSET=...	-	-	-	-	/FSET=*
/GROUP=...	SETGROUP	-	-	-	/GROUP=*
/VERSION=...	-	-	-	-	/VERSION=*
/VSET=...	-	-	-	-	/VSET=*

## Description

REPLICATE copies one or more eligible files from the primary directory on one StorHouse system to the primary directory on a destination, or target, StorHouse system. If the primary copy is unavailable on the source system, StorHouse accesses the duplex copy to create the replica.

Replication works as follows. When an application creates or modifies a replication eligible file, StorHouse queues the same action to occur on the destination StorHouse system the next time the REPLICATE command executes on the source system and selects that file. When a file is deleted (and removed) from the source system, StorHouse queues the same action to occur on the destination StorHouse the next time the REPLICATE command executes on the source system and selects any file.

For convenience, you can schedule the REPLICATE command to run periodically.

REPLICATE uses the same account, password, and file access group on the source and target systems. The system administrator must manually create the respective accounts and volume sets on the target location. If the required groups and file sets do not already exist on the destination system, StorHouse creates them automatically. System-created file sets are noncontiguous and use the other CREATE FSET command defaults.

Files are eligible for replication when they have a pre-assigned replication class or an explicitly specified default replication. The replication class defines information about the target StorHouse system (for example, the system, file set, volume set, and network link names and the network device identifier).

**Note the following:**

- REPLICATE requires SYSTEM privilege on both the source and target systems.
- A file and its replica may have different version numbers on the source and destination systems.
- The source and destination systems do not need to run the same StorHouse release as long as both releases support replication.
- Files queued for replication are not written to the destination system in any particular order or within a set time period.
- A file is considered replicated when at least one copy resides on the destination system, including on the performance buffer.
- Replicated files retain many of their original attributes including file name, group, FID, creation date/time, and retention characteristics.

- StorHouse does not copy file and file set replication class attributes from the source system to the target location. A replica inherits the replication class attribute of its target file set.
- StorHouse can replicate any type of primary file (VRAM, sequential, or STORHOUSE).

## Parameters

**filename** Specifies the name of the file or files to be replicated.

- **FORMAT:**
  - Filename
  - Partial\_filename\*
  - \*

StorHouse file names must contain 1 to 56 printable ASCII characters. At least one character must be non-blank. Lowercase characters are distinct from uppercase characters. File names must be unique within a file access group.

The wild card is valid only if you specify it as the last or only character in the file name.

- **DEFAULT:** None, you must specify this parameter.

## Command Modifiers

**/CHECK** Tells StorHouse to ignore a file's replicated flag and check whether every specified file has already been replicated on the destination StorHouse system. Unless necessary, do not use /CHECK because it increases processing time significantly.

- **FORMAT:** /CHECK
- **DEFAULT:** If /CHECK is omitted, StorHouse checks the replicated flag of only the file queued for replication to determine whether to copy it to the target StorHouse system.

- /CONFIRM** Controls whether StorHouse asks you to confirm the replication of each individual file.
- **FORMAT:** /CONFIRM or /NOCONFIRM
  - **DEFAULT:** /NOCONFIRM
- When the system requests a confirmation, enter YES (also Y or YE) or NO (also N). If you press R or enter any characters other than those described as a YES response, StorHouse interprets them as NO.
- If the command is to be performed for more than one item, the system generally requests a confirmation of each item and allows an END (entered as E, EN, or END) response. END directs the system not to perform the command for the current item and any further items.
- /DEF\_RPL\_CLASS** Specifies the name of the default replication class for files with no pre-assigned replication class. This feature enables one-time, ad hoc replication of files that otherwise would not be copied to the target StorHouse system. If the specified replication class does not exist, StorHouse generates an error message.
- **FORMAT:** /DEF\_RPL\_CLASS=replication\_class\_name
- A replication\_class\_name must consist of 1 to 8 of the following ASCII characters: A-Z (uppercase), 0-9, \_ (underscore), and \$ (dollar sign). StorHouse always translates replication class names to uppercase characters, even if enclosed in quotes.
- **DEFAULT:** If you omit /DEF\_RPL\_CLASS, StorHouse does not copy any file without a pre-assigned replication class to the target StorHouse system.
- /PREVIEW** Indicates that REPLICATE will display the number of files and bytes that will be replicated when the command is executed. REPLICATE/PREVIEW does not display individual file names or copy files to the target StorHouse system.
- **FORMAT:** /PREVIEW
  - **DEFAULT:** If you omit /PREVIEW, the command copies selected eligible files to the target StorHouse system.
- /RECORD** Indicates whether the command will change the replicated flag in the primary directory of the source StorHouse system when it copies a file version to the



target StorHouse system. /RECORD instructs the command to mark each replicated file version as having a current replica. /NORECORD instructs the command not to change the replicated flag.

- FORMAT: /RECORD or /NORECORD
- DEFAULT: /RECORD

**/REPORT** Controls the generation of special text responses for the completion of significant actions. /NOREPORT instructs StorHouse not to generate text responses.

- FORMAT: /REPORT or /NOREPORT
- DEFAULT: /NOREPORT

**/WAIT** Instructs StorHouse to wait for a locked file to be unlocked before attempting to use it in the command execution.

- FORMAT: /WAIT or /NOWAIT
- DEFAULT: /NOWAIT

## Parameter Modifiers

**/FSET** Replicates eligible files in the specified primary file set (and volume set) to the target StorHouse system.

- FORMAT: /FSET=fset\_name or /FSET=\*
- DEFAULT: If you omit /FSET, the default is all file sets (/FSET=\*).
- RESTRICTIONS:
  - If you specify /FSET, you must also specify /VSET.
  - If you specify the performance buffer file set name, StorHouse does not select any files for replication.

**/GROUP** Replicates eligible files in the specified file access group to the target StorHouse system.

- **FORMAT:**
  - /GROUP=groupname
  - /GROUP=partial\_groupname\*
  - /GROUP=\*
- **DEFAULT:** If you omit /GROUP, the default is all groups (/GROUP=\*).
- **PRIVILEGE:** You must have SETGROUP privilege to specify any group except your default group.

**/VERSION** Specifies the relative version number of files to be replicated.

- **FORMAT:** /VERSION=version or /VERSION=\*
- **DEFAULT:** If you omit /VERSION, the default is all versions (/VERSIONS=\*).

**/VSET** Replicates eligible files in the specified primary volume set to the target StorHouse system.

- **FORMAT:** /VSET=vset\_name or /VSET=\*
- **DEFAULT:** If you omit /VSET, the default is all volume sets (/VSET=\*).

## Examples

- To replicate all eligible files and wait for locked files to be unlocked before attempting to use them in the command execution, enter:  
? REPLICATE \* /WAIT
- To check whether version 0 of the file USERFILE in the file access group ACCT already exists on the target system and then copy it there, if required, enter:  
? REPLICATE USERFILE /CHECK /GROUP=ACCT /VERSION=0
- To preview the number of files and bytes that will be replicated to the target StorHouse system from volume set MAR\_2004 and to generate a text response for each processed file, enter:  
? REPLICATE \* /VSET=MAR\_2004 /REPORT /PREVIEW

## SCHEDULE

StorHouse Release 5.6 adds the following modifiers to the SCHEDULE command:

`/DISABLED`, `/EMAIL`, `/ON_ERROR` and `/TEST_EMAIL`. It also changes the `/SCHEDULE` modifier.

### Command Modifiers

- `/DISABLED` Instructs StorHouse to set the scheduled command in the disabled state. (StorHouse does not execute scheduled commands while they are disabled.) The command remains scheduled until enabled or removed. (This modifier is primarily for use by SGI customer support personnel.)
- **FORMAT:** `/DISABLED`
  - **DEFAULT:** Not disabled
- `/EMAIL` Specifies the email address that will receive the output of the session that executes the scheduled command.
- **FORMAT:** `/EMAIL=email_address`
  - **DEFAULT:** If you omit `/EMAIL`, StorHouse will not email the session output.
- `/ON_ERROR` Specifies to email the output of the session that executes the scheduled command only if command execution failed.
- **FORMAT:** `/ON_ERROR`
  - **DEFAULT:** If you omit `/ON_ERROR` and specify an email address, StorHouse emails the session output if the command execution succeeded or failed.
  - **RESTRICTIONS:** If you specify `/ON_ERROR`, you must also specify `/EMAIL` with an email address.
- `/SCHEDULE` Specifies how often the command is to be executed. After each run, the system reschedules the command to run again according to the specified schedule.
- Note: The start time for a scheduled event is specified separately (using `/START`) or is defaulted to the current or next possible time.

■ **FORMAT: /SCHEDULE=frequency<:n>**

The frequency specifies the basic rate, such as daily or monthly, at which the event will occur. You can use the optional multiplier to set the actual frequency to an integer multiple of the basic rate.

The frequency must be one of the following:

- HOURLY (60 minutes)
- DAILY (24 hours)
- WEEKDAYS (Monday, Tuesday, Wednesday, Thursday and Friday only)
- SUNDAY, MONDAY, TUESDAY, WEDNESDAY, THURSDAY, FRIDAY, and SATURDAY (command will run on the specified day)
- NOSUNDAY, NOMONDAY, NOTUESDAY, NOWEDNESDAY, NOTHURSDAY, NOFRIDAY, and NOSATURDAY (command will run every day except the indicated day)
- SATSUN (Saturday and Sunday only)
- MONTHLY (28 to 31 days depending on actual month)
- MINUTE (command will run every minute)
- EOM (end of month–last day)

If an event is scheduled to occur MONTHLY on day *n* of a month, StorHouse reschedules the event to occur on the *n*th day of every month, unless *n* is 29, 30, or 31. If a MONTHLY event is started on the 29th, 30th, or 31st of a month, StorHouse reschedules the event to occur on the last day of any month that does not have the desired day.

You can specify the multiplier *n* only for the following frequencies:

- MINUTE, where *n* must be 1 through 9999.
- HOURLY, where *n* must be 1 through 9999
- DAILY, where *n* must be 1 through 366
- MONTHLY, where *n* must be 1 through 12.

For example, MONTHLY:3 schedules an event to occur every 3 months





(quarterly). DAILY: 7 indicates weekly, that is, once every 7 days.

DEFAULT: If you omit /SCHEDULE, the system executes the command\_line once at the time specified by /START or its default.

**/TEST\_EMAIL** Specifies to send a test email message to the email address indicated by the /EMAIL modifier before scheduling the command. StorHouse can only detect some email address problems, and when it does, the software reports the problem and does not schedule the command. However, because StorHouse cannot detect email delivery problems, it schedules the command regardless if the test email is delivered successfully. For email delivery problems, you must remove and reschedule the command when you do not receive the test email.

- **FORMAT:** /TEST\_EMAIL
- **DEFAULT:** If you omit /TEST\_EMAIL, the system will not send a test email.
- **RESTRICTIONS:** If you specify /TEST\_EMAIL, you must also specify an email address with /EMAIL.

## SET DEVICE

StorHouse Release 5.6 adds six new parameter modifiers, `/ACCESSIBLE`, `/CLEANED`, `/NEEDS_CLEANING`, `/SIZE`, `/UNWRITELOCK`, `/WRITELOCK` to the SET DEVICE command. These modifiers enable you to:

- Change a drive status from INACCESSIBLE to ACCESSIBLE.
- Change the mode of a level L drive to CLEANED.
- Change the mode of a level L drive to NEEDS\_CLEANING.
- Dynamically change the size of fixed content object-based disk and filesystem level F drives.
- Change the mode of a level F drive to UNWRITELOCKED.
- Change the mode of a level F drive to WRITELOCKED.

### Parameter Modifiers

`/ACCESSIBLE` Clears the INACCESSIBLE mode of all drives in the library device. The did parameter must specify the device identification code of the level L library that contains the inaccessible drives.

- **FORMAT:** `/ACCESSIBLE`
- **DEFAULT:** If you omit `/ACCESSIBLE`, the INACCESSIBLE mode of the drive is not affected.
- **ACCESS REQUIREMENTS:** No additional privileges are required to use this modifier.

`/CLEANED` Changes the mode of a level L drive. `/CLEANED` instructs StorHouse to clear the NEEDS\_CLEANING mode for the drive. StorHouse generally handles level L drive cleaning automatically. However, if StorHouse was unable to clean a drive, it sets the drive to NEEDS\_CLEANING mode, and stops using the drive. If the drive is then cleaned manually, you can use this command to inform StorHouse that the drive has been cleaned, and that StorHouse can resume using it.

- **FORMAT:** `/CLEANED`
- **DEFAULT:** If you omit this modifier, the NEEDS\_CLEANING mode of the drive is unaffected.

- **RESTRICTIONS:** /CLEANED and /NEEDS\_CLEANING are mutually exclusive.

**/NEEDS\_CLEANING**

Changes the mode of a level L drive. /NEEDS\_CLEANING instructs StorHouse to set the NEEDS\_CLEANING mode for the drive. StorHouse generally handles level L drive cleaning automatically. However, this command can be used to inform StorHouse that the drive needs cleaning. StorHouse will stop using the drive, and clean it if possible.

- **FORMAT:** /NEEDS\_CLEANING
- **DEFAULT:** If you omit this modifier, the NEEDS\_CLEANING mode of the drive is unaffected.
- **RESTRICTIONS:** /CLEANED and /NEEDS\_CLEANING are mutually exclusive.

**/SIZE** Specifies the size of fixed content object-based disk or filesystem level F drives.

- **FORMAT:** /SIZE=number\_of\_bytes  
 The value for number\_of\_bytes can range up to 4294967295 (do not use commas in the specification). Use K to indicate 1,000-byte units, M to indicate 1,000,000-byte units, and G to indicate 1,000,000,000-byte units. If you do not specify K, M, or G, the number defaults to 1-byte units.
- **DEFAULT:** If you omit /SIZE, the size attribute is not changed.
- **RESTRICTIONS:** /SIZE can only be used to specify the size of fixed content object-based disk and filesystem level F drives. It is not valid for other types of level F devices or for level L devices.  
  
 The value of /SIZE may not be less than the amount of space currently allocated on the device.

**/UNWRITELOCK**

Changes the mode of a level F drive. /UNWRITELOCK instructs StorHouse to clear writelocked mode for the drive.

- **FORMAT:** /UNWRITELOCK
- **DEFAULT:** If you omit this modifier, the WRITELOCKED mode of the drive is unaffected.
- **RESTRICTIONS:** /WRITELOCK and /UNWRITELOCK are mutually exclusive.

`/WRITELOCK` Changes the mode of a level F drive. `/WRITELOCK` instructs StorHouse to set the `WRITELOCKED` mode for the drive.

- **FORMAT:** `/WRITELOCK`
- **DEFAULT:** If you omit this modifier, the `WRITELOCKED` mode of the drive is unaffected.
- **RESTRICTIONS:** `/WRITELOCK` and `/UNWRITELOCK` are mutually exclusive.

### Examples

- To set the size of F00 to 500 GB, enter:  
? SET DEVICE F00 /SIZE=500G
- To change device L00D01 from `INACCESSIBLE` to `ACCESSIBLE`, enter:  
? SET DEVICE L00D01 /ACCESSIBLE

## SET FILE

StorHouse Release 5.6 adds one new command modifier, `/WAIT`, and four new parameter modifiers, `/FSET`, `/NOREPLICATED`, `/RETENTION`, and `/RPL_CLASS` to the SET FILE command.

### Command Modifier

`/WAIT` Instructs StorHouse to wait for a locked file to be unlocked before attempting to use it in the command execution.

- **FORMAT:** `/WAIT` or `/NOWAIT`
- **DEFAULT:** `/NOWAIT`**FORMAT:** `/UNWRITELOCK`

### Parameter Modifiers

`/FSET` Indicates that the command will only select files in the specified file set (and volume set).

- **FORMAT:** `/FSET=fset_name`
- **FORMAT RESTRICTIONS:** Wildcards are not allowed.
- **DEFAULT:** If you omit `/FSET`, StorHouse does not use the file set name to select files.
- **RESTRICTIONS:** If you specify `/FSET`, you must also specify `/VSET`.

`/NOREPLICATED` Clears the replicated status of a file version.

`/NOREPLICATED` is a version-dependent attribute. The `/VERSION` modifier determines which version is changed.

- **FORMAT:** `/NOREPLICATED`
- **DEFAULT:** If you omit `/NOREPLICATED`, the replicated status of the file version is not changed.
- **PRIVILEGE:** You need SYSTEM privilege to specify `/NOREPLICATE`.

**/RETENTION** Specifies the retention attribute (retention period) for the file.

RETENTION is a version-dependent attribute. The /VERSION modifier determines which version is changed.

■ **FORMAT:**

Option	Description
/RETENTION=DEFAULT	Sets the retention period of the specified file version to the default value.
/RETENTION=number_of_days	<p>Sets the retention period of the specified file version to the specified number of days. The retention period ends when the current date is beyond the file's last_modified date plus the specified retention value.</p> <p>A value of 0 indicates no retention period (same as specifying ZERO).</p> <p>Example: /RETENTION=3</p> <p>In this example, the retention period is 3 days. The retention period for a file that was last modified at 1 p.m. on December 12 expires at 11 p.m. on December 15.</p>
/RETENTION=ZERO	Sets no retention period, which indicates the specified file version may be deleted.
/RETENTION=FOREVER	Sets an infinite retention period, which indicates the specified file version may never be deleted.

- **DEFAULT:** If you omit /RETENTION, there is no change to the file's current retention value.
- **RESTRICTIONS:** You may only specify a more restrictive value. For example, if the current file retention is 100 days, you may specify 200 days but not 50 days.



`/RPL_CLASS` Specifies the name of the replication class for the specified file. A replication class is a collection of replication-related information (system, file set, volume set, and link names and network device identification) about the target StorHouse system. `/NORPL_CLASS` specifies that the file has no replication class. If the specified replication class does not exist, StorHouse generates a warning message.

`/RPL_CLASS` is a version-dependent attribute. The `/VERSION` modifier defines the version that is assigned the specified replication class.

- **FORMAT:** `/RPL_CLASS=replication_class_name` or `/NORPL_CLASS`  
A `replication_class_name` can consist of from 1 to 8 of the following ASCII characters: A-Z, 0-9, `_`, and `$`. StorHouse always forces replication class names to uppercase, even when enclosed in quotes.
- **DEFAULT:** If `/RPL_CLASS` is omitted, the default is not to change the replication class.

## Example

To specify that version -1 of USERFILE will be assigned the STANDARD replication class and a retention period of 60 days, enter:

```
? SET FILE USERFILE /RPL_CLASS=STANDARD /RETENTION=60 /VERSION=-1
```

## SET FSET

StorHouse Release 5.6 adds four new parameter modifiers, `/FORCE_RETENTION`, `/REDUCE`, `/RETENTION`, and `/RPL_CLASS`, to the SET FSET command.

### Parameter Modifiers

All parameter modifiers except `/VSET` are mutually exclusive. You can only specify one in each command.

#### `/FORCE_RETENTION`

Specifies whether the file set retention value overrides the file retention value explicitly supplied by an application at file create time. This modifier affects only new files in the file set. `/NOFORCE_RETENTION` indicates that the file set retention value does not override the file retention value.

- **FORMAT:** `/FORCE_RETENTION` or `/NOFORCE_RETENTION`
- **DEFAULT:** If you omit `/FORCE_RETENTION`, there is no change.
- **RESTRICTIONS:** `/FORCE_RETENTION` can only be used when specifying a primary file set.

#### `/REDUCE`

Indicates that StorHouse will deallocate all unnecessary general-use free storage from the file set. It also deallocates all free storage reserved for updates except for any that must be retained to meet the requirements of the update attribute. If the value of the update attribute is zero, StorHouse deallocates all unnecessary free storage.

The system returns the deallocated storage to the free pool of the associated volume set. Unlike `/RELEASE`, if the file set size becomes zero after the reduce operation, StorHouse does not remove the file set definition from the system.

- **FORMAT:** `/REDUCE`
- **DEFAULT:** If you omit `/REDUCE`, other modifiers control any changes to file set storage.
- **RESTRICTIONS:**
  - You cannot reduce the performance buffer FSET (`VSET=MAGDISK, FSET=$$BUFFER`)
  - You cannot reduce the checkpoint file set if the system parameter



CHKP\_ON is set to TRUE.

**/RETENTION** Specifies the file set retention attribute (retention period). The file set retention attribute applies only to new files (in the file set) that are created without a specified retention value, or whenever /FORCE\_RETENTION is in effect on the file set.

■ **FORMAT:**

Option	Description
/RETENTION=DEFAULT	Sets the retention period to the default value.
/RETENTION=number_of_days	<p>Sets the retention period to the specified number of days. The retention period ends when the current date is beyond the file's last_modified date plus the specified retention value.</p> <p>A value of 0 indicates no retention period (same as specifying ZERO).</p> <p>Example: /RETENTION=3</p> <p>In this example, the retention period is 3 days. The retention period for a file that was last modified at 11 p.m. on December 12 expires at 11 p.m. on December 15.</p>
/RETENTION=ZERO	Sets no retention period.
/RETENTION=FOREVER	Sets an infinite retention period.

- **DEFAULT:** If you omit /RETENTION, the default is to keep the current file set retention attribute.
- **RESTRICTIONS:**
  - The RETENTION\_MODE system parameter determines which retention settings are valid. If the RETENTION\_MODE is STRICT, you may set /RETENTION only to a more restrictive value than its current setting (for example, from a current retention of 100 days to a new retention of 200

days). If the RETENTION\_MODE is BASIC, you may set /RETENTION to any valid value (for example, from a current retention of 60 days to a retention of 30 days or 90 days).

- When you specify /RETENTION, the specified file set must be a primary file set.

**/RPL\_CLASS** Specifies the name of the replication class for new files written to this file set. A replication class is a collection of replication-related information (network device identification and system, file set, volume set, and link names) about the target StorHouse system. If the specified replication class does not exist, StorHouse generates a warning message.

- **FORMAT:** /RPL\_CLASS=replication\_class\_name or /NORPL\_CLASS  
A replication\_class\_name can consist of from 1 to 8 of the following ASCII characters: A-Z, 0-9, \_, and \$. StorHouse always forces replication class names to uppercase, even when enclosed in quotes.
- **DEFAULT:** If you omit /RPL\_CLASS, the default is no change.
- **RESTRICTIONS:** When you specify /RPL\_CLASS, the specified file set must be a primary file set.

### Examples

- To set the replication class for file set FSET1 in volume set VSET1 to STANDARD, enter:  
? SET FSET FSET1 /RPL\_CLASS=STANDARD /VSET=VSET1
- To set the retention period for files in file set FSET1 in volume set VSET1 to FOREVER, enter:  
? SET FSET FSET1 /RETENTION=FOREVER /VSET=VSET1

## SET VOLUME

Release 5.6 adds support for the /VALIDATE\_DATE modifier. This modifier requires ALLOCATION command privilege.

**/VALIDATE\_DATE** Sets the VALIDATE\_DATE timestamps of each selected volume to the current date/time, the specified date/time, or the date/time the volume was initialized. For a full description of the absolute time and delta time data fields, see Appendix B, “Data Field, Parameter, and Modifier Definitions,” in the *Command Language Reference Manual*.

- **FORMAT:**
  - /VALIDATE\_DATE
  - /VALIDATE\_DATE=absolute\_time
  - /VALIDATE\_DATE=delta\_time
  - /NOVALIDATE\_DATE
- **DEFAULT:** If you omit /VALIDATE\_DATE, the modifier has no effect.
- **RESTRICTIONS:** This modifier is mutually exclusive with all parameter modifiers except /DIRECTORY and /VSET.

## SET VSET

StorHouse Release 5.6 adds the parameter modifier, `/REDUCE`, to the `SET VSET` command. In addition, the `/LIBRARY` modifier can specify a level F device.

### Parameter Modifiers

`/LIBRARY` Specifies a new value for the volume set's library device attribute. StorHouse will allocate new volumes to the volume set from the free pool for the specified device. This free pool must contain the same media and recording type as the volume set. This modifier does not move volumes.

- **FORMAT:** `/LIBRARY=Lxx` or `/LIBRARY=Fxx`

where `Lxx` is the device identification code for the library-level device with the unit number `xx` (for example, `L00`), and `Fxx` is the second, third, and fourth characters of a free pool volume set name (for example, `F00`). This free pool will be used for subsequent allocations to the volume set.

- **DEFAULT:**

- If you omit `/LIBRARY`, the volume set's library device attribute is not changed.
- If you specify the modifier without a device specification, the system uses the value of the `DEFAULT_LD` system parameter as the default device.

- **RESTRICTIONS:**

- All parameter modifiers are mutually exclusive; you can specify only one in each command.
- The specified library device must support the media that compose the volume set.



**/REDUCE** Indicates that StorHouse is to deallocate all empty volumes that have no space allocated to file sets from the volume set. StorHouse places the deallocated volumes in the free pool for the library where each volume resides. Unlike /RELEASE, /REDUCE retains the volume set definition if the volume set size becomes zero (all volumes in the volume set were deallocated).

- **FORMAT:** /REDUCE
- **DEFAULT:** If you omit /REDUCE, other modifiers control any changes to volume set storage.
- **RESTRICTIONS:** All parameter modifiers are mutually exclusive; you can specify only one in each command.

## SHOW CONFIGURATION

The SHOW CONFIGURATION command displays configuration information.

### Format

SHOW CONFIGURATION

COMMAND FORMAT SUMMARY					
COMMAND, PARAMETER, OR MODIFIER	REQUIRED COMMAND PRIVILEGE	REQUIRED GROUP ACCESS	REQUIRED FILE ACCESS	MINIMUM ACCOUNT ACCESS	DEFAULT
SHOW CONFIGURATION	OPERATOR, SHOW, or SYSTEM	-	-	-	-
/DEVICE	-	-	-	-	-
/HARDWARE	-	-	-	-	-
/MEDIA	-	-	-	-	-
/SOFTWARE	-	-	-	-	-

### Description

SHOW CONFIGURATION displays system and StorHouse configuration information, including software, hardware, devices, and media. If no specific information is requested, all configuration information is displayed.

## Command Modifiers

**/DEVICE** Displays information about StorHouse devices.

- **FORMAT:** /DEVICE
- **DEFAULT:** If none of /DEVICE, /HARDWARE, /MEDIA, and /SOFTWARE is specified, all system information is displayed.

**/HARDWARE** Displays information about system hardware.

- **FORMAT:** /HARDWARE
- **DEFAULT:** If none of /DEVICE, /HARDWARE, /MEDIA, and /SOFTWARE is specified, all system information is displayed.

**/MEDIA** Displays information about StorHouse media.

- **FORMAT:** /MEDIA
- **DEFAULT:** If none of /DEVICE, /HARDWARE, /MEDIA, and /SOFTWARE is specified, all system information is displayed.

**/SOFTWARE** Displays information about system and StorHouse software.

- **FORMAT:** /SOFTWARE
- **DEFAULT:** If none of /DEVICE, /HARDWARE, /MEDIA, and /SOFTWARE is specified, all system information is displayed.

## Example

To display all system configuration information, enter:

```
? SHOW CONFIGURATION
```

The system displays the following information:

```
SOFTWARE=OS SYSTEM=Linux RELEASE=2.6.18-238.9.1.el5
```

```
VERSION="#1 SMP Tue Apr 12 18:10:13 EDT 2011"
```

```
SOFTWARE=StorHouse/SM RELEASE=5.6
```

```
SOFTWARE=VRAM RELEASE=1.7_KRA
SOFTWARE=StorHouse/RM RELEASE=3.4HARDWARE=SYSTEM CPUS=8
HARDWARE=CPU0 MODEL="Intel(R) Xeon(R) CPU      E5410 @ 2.33GHz"
HARDWARE=CPU1 MODEL="Intel(R) Xeon(R) CPU      E5410 @ 2.33GHz"
HARDWARE=CPU2 MODEL="Intel(R) Xeon(R) CPU      E5410 @ 2.33GHz"
HARDWARE=CPU3 MODEL="Intel(R) Xeon(R) CPU      E5410 @ 2.33GHz"
HARDWARE=CPU4 MODEL="Intel(R) Xeon(R) CPU      E5410 @ 2.33GHz"
HARDWARE=CPU5 MODEL="Intel(R) Xeon(R) CPU      E5410 @ 2.33GHz"
HARDWARE=CPU6 MODEL="Intel(R) Xeon(R) CPU      E5410 @ 2.33GHz"
HARDWARE=CPU7 MODEL="Intel(R) Xeon(R) CPU      E5410 @ 2.33GHz"
HARDWARE=MEMORY SIZE=8038580KB
DEVICE=F00 TYPE=FILESYSTEM SIZE=252299026KB
DEVICE=F01 TYPE=FILESYSTEM SIZE=252299026KB
DEVICE=F02 TYPE=FILESYSTEM SIZE=252299026KB
DEVICE=F03 TYPE=FILESYSTEM SIZE=252299026KB
DEVICE=F04 TYPE=FILESYSTEM SIZE=74633371KB
DEVICE=F05 TYPE=FILESYSTEM SIZE=74633371KB
DEVICE=F06 TYPE=FILESYSTEM SIZE=74633371KB
DEVICE=F07 TYPE=FILESYSTEM SIZE=74633371KB
DEVICE=L00 MFG=Qualstar MODEL=QUALSTAR SLOTS=155
FIRMWARE=3262
```





SERIAL="07011241"

DEVICE=L00A00

DEVICE=L00D00 MFG=IBM MODEL=IBMLTO FIRMWARE=97F0

SERIAL="1310181910"

DEVICE=L00D01 MFG=IBM MODEL=IBMLTO FIRMWARE=97F0

SERIAL="1310207418"

DEVICE=L00D02 MFG=IBM MODEL=IBMLTO FIRMWARE=97F0

SERIAL="9310014136"

DEVICE=L00E00 SLOTS=10

DEVICE=N00 PROTOCOL=TCP/IP

DEVICE=N02 PROTOCOL=DIRECT\_CONNECT

DEVICE=S00

MEDIA=MIA DESC="Unix file system disks"

MEDIA=NAA DESC="TCP network"

MEDIA=NCA DESC="SGI Direct Connect network"

MEDIA=TFD DESC="LTO4 800GB Ultrium tape cartridge"

MEDIA\_SIZE=799994281KB MEDIA\_LIFE=10000

MEDIA=TFZ DESC="LTO Ultrium 100-pass cleaning tape cartridge"

## SHOW DEVICE

StorHouse Release 5.6 adds the /FULL command modifier to the SHOW DEVICE command. It also displays a new mode of INACCESSIBLE, which pertains only to the STK SL8500 library.

### Command Modifier

**/FULL** Displays all available information for a device. For each level F device, /FULL adds the following fields to the default display: VSET=vset\_name, DIRECTORY=directory, and SIZE=size.

- **FORMAT:** /FULL
- **DEFAULT:** If you omit /FULL, the command displays the information listed in the command description.
- **PRIVILEGE:** None.

## SHOW FILE

StorHouse Release 5.6 changes the SHOW FILE command as follows:

- Adds the /NAME command modifier.
- Adds the /ARCHIVE\_EXISTS, /BACKUP\_EXISTS, /BKP\_ATTR, /BUFFERED, /EXPIRED, /MAXEXTENTS, /MAXSIZE, /MINEXTENTS, /MINSIZE, /PHYSICAL, /REPLICA\_EXISTS, /RESIDENT, /RETENTION, /RPL\_CLASS, /SAFE\_COPIES, /UNUSED, and /USED parameter modifiers.
- Expands the definition of the /DAMAGED parameter modifier.
- Adds the capability for /VOLUME to specify a level F volume ID.
- Adds the capability for /EXTENT to display the actual media and recording type for level F volumes.

You can use the /BKP\_ATTR, /BUFFERED, /PHYSICAL, /RESIDENT, /EXPIRED, /MAXEXTENTS, /MAXSIZE, /MINEXTENTS, /MINSIZE, /PHYSICAL, /REPLICA\_EXISTS, /RESIDENT, /RETENTION, /RPL\_CLASS, /SAFE\_COPIES, /UNUSED, and /USED modifiers to select files for display.

### Command Modifiers

The following information is new or has changed for StorHouse Release 5.6.

- /FULL** Adds the REPLICATED descriptor flag and the RETENTION and RPL\_CLASS fields to the default display for each file version:
- DESCRIPTOR=flags. Valid values are:
    - ARCHIVED – current revision was archived.
    - BACKED\_UP – current revision was backed up.
    - CATALOGING – being cataloged.
    - DELETED – deleted.
    - HARD\_DISABLED – hardware disabled.
    - NAMED – named file.

- NOBACKUP – do not back up file version.
- PENDING – file copy/transfer in progress.
- SOFT\_DISABLED – software disabled.
- REPLICATED – a replica of the current file version exists on another StorHouse system.
- RETENTION=retention\_attribute.
- RPL\_CLASS=rpl\_class\_name.

**/NAME** Displays the file name in the format:

“filename” /GROUP=group\_name /VERSION=version\_number

The output of a SHOW FILE/NAME can be directed to a file and subsequently edited and used in other file-related commands such as DELETE.

- **FORMAT:** /NAME
- **DEFAULT:** If you omit /NAME, the command does not display the file name in the specified format.
- **RESTRICTION:** /NAME is mutually exclusive with /BRIEF, /EXTENT, and /FULL.

### Parameter Modifiers

The following information is new or has changed for StorHouse Release 5.6.

**/ARCHIVE\_EXISTS** Selects files that either have or do not have an archive copy.

- **FORMAT:**
  - /ARCHIVE\_EXISTS
  - /NOARCHIVE\_EXISTS
- **DEFAULT:** If you omit /ARCHIVE\_EXISTS, the system does not select files based on whether there is an archive copy.
- **RESTRICTION:** Do not use /ARCHIVE\_EXISTS when displaying information for an archive file.



**/BKP\_ATTR** Selects files based on whether their BACKUP attribute is set. **/NOBKP\_ATTR** selects files that do not have their BACKUP attribute set.

- **FORMAT:**
  - **/BKP\_ATTR**
  - **/NOBKP\_ATTR**
- **DEFAULT:** If you omit **/BKP\_ATTR**, the system does not select files based on whether the BACKUP attribute is set.
- **RESTRICTION:** Do not use **/BKP\_ATTR** when displaying information for a backup file.

**/BACKUP\_EXISTS** Selects files that either have or do not have a backup copy.

- **FORMAT:**
  - **/BACKUP\_EXISTS**
  - **/NOBACKUP\_EXISTS**
- **DEFAULT:** If you omit **/BACKUP\_EXISTS**, the system does not select files based on whether there is a backup copy.
- **RESTRICTION:** Do not use **/BACKUP\_EXISTS** when displaying information for a backup file.

**/BUFFERED** Selects files for display based on whether copies of all file extents reside in the performance buffer or are queued for upward migration to the performance buffer. **/NOBUFFERED** selects files that do not have copies of all extents in the performance buffer.

- **FORMAT:**
  - **/BUFFERED**
  - **/NOBUFFERED**
- **DEFAULT:** If you omit **/BUFFERED**, the command does not select files based on whether they have copies of all extents in the performance buffer.
- **RESTRICTIONS:** **/BUFFERED** is meaningful only for files in the PRIMARY directory.

**/DAMAGED** Selects files for display based on whether they are marked in the directory as damaged. Damaged files are files marked PENDING, SOFT\_DISABLED, or HARD\_DISABLED, or that are missing one or more extents, that is, marked as truncated or partial. /DAMAGED checks for problems with the file's directory information, not for problems with the physical media where the file resides.

- **FORMAT:**
  - /DAMAGED
  - /NODAMAGED
- **DEFAULT:** If you omit /DAMAGED, the command does not select files based on whether they are damaged.

**/EXPIRED** Selects for display file versions that are expired, that is, file versions that have a retention attribute and that retention period has passed.

- **FORMAT:** /EXPIRED
- **DEFAULT:** If you omit /EXPIRED, the system does not select files based on whether or not they are expired.

**/MAXEXTENTS** Selects for display files that have at most the specified number of extents.

- **FORMAT:** /MAXEXTENTS=number\_of\_extents
- **DEFAULT:** If you omit /MAXEXTENTS, the system does not select files based on the number of extents they have.

**/MAXSIZE** Selects for display files that are no larger than the specified size.

- **FORMAT:** /MAXSIZE=number\_of\_bytes  
The value for number\_of\_bytes can range from 0 up to 4,294,967,29. (Do not use commas in the specification.) Use K to indicate 1,000-byte units, M to indicate 1,000,000-byte units, and G to indicate 1,000,000,000-byte units. If you do not specify K, M, or G, the number defaults to 1-byte units.
- **DEFAULT:** If you omit /MAXSIZE, the system does not select files based on file size.



- /MINEXTENTS** Selects for display files that have at least the specified number of extents.
- **FORMAT:** /MINEXTENTS=number\_of\_extents
  - **DEFAULT:** If you omit /MINEXTENTS, the system does not select files based on the number of extents they have.

- /MINSIZE** Selects for display files that are at least as large as the specified size.
- **FORMAT:** /MINSIZE=number\_of\_bytes  
The value for number\_of\_bytes can range from 0 up to 4,294,967,295 (do not use commas in the specification). Use K to indicate 1,000-byte units, M to indicate 1,000,000-byte units, and G to indicate 1,000,000,000-byte units. If you do not specify K, M, or G, the number defaults to 1-byte units.
  - **DEFAULT:** If you omit /MINSIZE, the system does not select files based on file size.

- /PHYSICAL** Selects files for display based on whether copies of all file extents physically reside in the performance buffer. Files that are queued for upward migration (transfer to the performance buffer) do not qualify. /PHYSICAL is a qualifier for the /BUFFERED modifier.
- **FORMAT:** /PHYSICAL
  - **DEFAULT:** If you omit /PHYSICAL, the command does not select files based on whether copies of all file extents physically reside in the performance buffer.
  - **RESTRICTIONS:** /PHYSICAL is meaningful only for files in the PRIMARY directory.  
  
/PHYSICAL must be specified in combination with /BUFFERED. Otherwise StorHouse ignores it.

- /REPLICA\_EXISTS** Selects files that have/don't have a replica copy.
- **FORMAT:**
    - /REPLICA\_EXISTS
    - /NOREPLICA\_EXISTS

- **DEFAULT:** If you omit `/REPLICA_EXISTS`, the system does not select files based on whether or not a replica copy exists.
- **RESTRICTIONS:** `/REPLICA_EXISTS` is meaningful only for files in the `PRIMARY` directory.

`/RETAINED` Selects for display file versions that are retained, that is, file versions that have a retention attribute and that retention period has not passed.

- **FORMAT:** `/RETAINED`
- **DEFAULT:** If you omit `/RETAINED`, the system does not select files based on whether or not they are retained.

`/RETENTION` Selects for display files that have/do not have a retention attribute (retention period).

- **FORMAT:**
  - `/RETENTION`
  - `/NORETENTION`
- **DEFAULT:** If you omit `/RETENTION`, the command does not select files based on whether they have a retention attribute.

`/RPL_CLASS` Selects files for display based on their replication class. A replication class is a collection of replication-related information (system, file set, volume set, and link names and network device identification) about the target StorHouse system.

- **FORMAT:**
  - `/RPL_CLASS`
  - `/RPL_CLASS=replication_class_name`
  - `/NORPL_CLASS`

A `replication_class_name` can consist of from 1 to 8 of the following ASCII characters: A-Z, 0-9, `_`, and `$`. StorHouse always forces replication class names to uppercase, even when enclosed in quotes.

`/RPL_CLASS` selects files with any replication class name.



`/RPL_CLASS=replication_class_name` selects files with the specified replication class name.

`/NORPL_CLASS` selects files with no replication class name.

- **DEFAULT:** If you omit `/RPL_CLASS`, the system does not select files based on their replication class.

`/RESIDENT` Selects files for display based on whether all extents have been written to their resident file sets. `/NORESIDENT` selects files whose extents have not all been written to their resident file sets.

- **FORMAT:**
  - `/RESIDENT`
  - `/NORESIDENT`
- **DEFAULT:** If you omit `/RESIDENT`, the command does not select files based on whether all extents have been written to their resident file sets.
- **RESTRICTIONS:** `/RESIDENT` is meaningful only for files in the PRIMARY directory.

`/SAFE_COPIES` Selects files for display with at least the specified number of safe copies in the PRIMARY, BACKUP, and ARCHIVE directories. For a file to be considered safe, it must be complete (that is, no missing extents) and usable (that is, not marked PENDING, SOFT\_DISABLED, or HARD\_DISABLED). In addition, for files in the PRIMARY directory, all extents must have been written to their resident file sets.

- **FORMAT:** `/SAFE_COPIES=number_of_copies`  
The `number_of_copies` can range from 1 through 3.
- **DEFAULT:** If you omit `/SAFE_COPIES`, the system does not select files based on the number of safe copies that exist.

**/UNUSED** Selects files for display that have not been accessed since the specified time. See the "data\_field date-time" topic for information about absolute and delta time specifications.

- **FORMAT:**
  - /UNUSED=absolute\_time
  - /UNUSED=delta\_time
- **DEFAULT:** If you omit both /UNUSED and /USED, the default is to select files with any last-access date.

**/USED** Selects files for display that have been accessed since the specified time. See the "data\_field date-time" topic for information about absolute and delta time specifications.

- **FORMAT:**
  - /USED=absolute\_time
  - /USED=delta\_time
- **DEFAULT:** If you omit both /UNUSED and /USED, the default is to select files with any last-access date.

## Example



The following example illustrates new display information for SHOW FILE.

To display all information available for version 0 of the file RPLCONFIG, including file extent information, enter:

```
? show file RPLCONFIG /FULL /EXTENT
```

```
FILE="RPLCONFIG" GROUP=STH VERSION=0 FID=1234.1125388
DATE=30-JAN-2006:16:20:56 SIZE=203 REVISION=1
DIRECT_REC=22329 LASTACCESS=30-JAN-2006:16:20:56
MODIFIED_DIR=30-JAN-2006:16:20:56 MODIFIED_FILE=30-JAN-
2006:16:20:56 BACKUP=none LIMIT=32768 PROTECTION=none VTF=NEXT ATF=3
ORGANIZATION=SEQUENTIAL FILE_SYSTEM=67 HOST_TYPE=33
FRAME_VER=1 FRAME_SIZE=31744 FRAME_HDR=20 RECORD_HDR=5
HOST_UNIT=8 MAX_LENGTH=16384 RECORD_COUNT=0 ATTRIBUTES=0
ACCESSES=1 EDC=2 STATUS=COMPLETE
DESCRIPTOR=(NAMED, NOBACKUP) VSET=FVSET1 FSET=FVSET1
RETENTION=60 RPL_CLASS=none

EXTENT_COUNT=1

EXTENT_NUMBER=1000000
EXTENT_SID=1234
EXTENT_DATE=30-JAN-2006:16:20:56
EXTENT_WRITTEN=30-JAN-2006:16:20:56
EXTENT_REVISION=1 EXTENT_SIZE=203
EXTENT_LOCATION=MEA"F04":A EXTENT_LEVEL=F
EXTENT_STATUS=(BUFFERED, LAST, NEW)
EXTENT_MF=4000000
EXTENT_RETENTION_DATE=60
EXTENTS DISPLAYED=1

Total files displayed=1
```

## SHOW FSET

StorHouse Release 5.6 adds a new state, `FORCE_RETENTION`, and the new fields, `RETENTION` and `RPL_CLASS`, to the `SHOW FSET` default display.

- `STATE=(CONTIGUOUS, AUTO_STAGE, FORCE_RETENTION)` indicates that the file set has the contiguous storage allocation attribute, the `AUTO_STAGE` attribute, the `FORCE_RETENTION` attribute, or some combination of the three. If a file set has no assigned attributes, `STATE` does not display.
- `RETENTION=(number_of_days, DEFAULT, FOREVER)` indicates the file set retention attribute. The number of days can range from 0-65000. StorHouse displays `RETENTION` only for primary file sets.
- `RPL_CLASS=(rpl_class_name, none)` indicates the name of the replication class assigned to the file set or none if the file set does not have a replication class. StorHouse displays this field only for primary file sets.

### Examples

The following examples show the `STATE`, `RETENTION`, and `RPL_CLASS` fields.

- To display basic information for file set `PRR` in volume set `PRR`, enter:

```
? SHOW FSET P /VSET=P
```

```
FSET=PRR VSET=PRR DIRECTORY=PRIMARY SIZE=10444KB  
GENERAL_FREE=10433KB UPDATE_FREE=0KB STATE=(CONTIGUOUS,  
FORCE_RETENTION) RETENTION=365 RPL_CLASS=STANDARD
```

```
Total fsets displayed=1
```

- To show all information for file set `PRR` in volume set `PRR`, enter:

```
? SHOW FSET P /VSET=P /FULL
```

```
FSET=PRR VSET=PRR DIRECTORY=PRIMARY SIZE=10444KB  
GENERAL_FREE=10433KB UPDATE_FREE=0KB  
GENERAL_ALLOCATED=11KB UPDATE_ALLOCATED=0KB  
UPDATE_PERCENT=00 LIMIT=0KB CREATED=26-MAR-2004:13:59:59  
MODIFIED=26-MAR-2004:14:05:57 STATE=(CONTIGUOUS,  
FORCE_RETENTION) RETENTION=365 RPL_CLASS=STANDARD
```



Total fsets displayed=1

## SHOW VOLUME

StorHouse Release 5.6 adds the following capabilities to SHOW VOLUME:

- Ability to display information for a level F volume only when any one of the following is specified:
  - The volume ID begins with the letter M, which indicates a level F media type.
  - /FIXED is specified.
  - /VSET is specified, and it names a level F volume set.
  - /FREE\_POOL is specified, and the free pool name begins with the letter F.
  - /LOCATION is specified, and it begins with the letter F.
- Ability to display VALIDATE\_DATE and LAST\_VALIDATED information when the /DATE and /FULL modifiers are specified. For these modifiers:
  - VALIDATE\_DATE indicates the volume side's validate date used by the VALIDATE VOLUME command. Basically, it indicates the oldest non-validated data on the volume.
  - LAST\_VALIDATED indicates when the last successful validate of the volume side started.
- Nine new parameter modifiers: /DISMOUNTED, /FIRST\_ALLOCATION, /FIXED, /LOCATION, /LAST\_ALLOCATION, /NONVALIDATED, /NOTDISMOUNTED, /UNUSED, and /VALIDATED to the command

## Parameter Modifiers

**/DISMOUNTED** Selects volumes that have been dismounted since the specified date/time. /DISMOUNTED can roughly be interpreted as “used since.” See the “data\_field date-time” topic for information about absolute and delta time specifications.

- **FORMAT:**
  - /DISMOUNTED=absolute\_time
  - /DISMOUNTED =delta\_time
- **DEFAULT:** If you omit /DISMOUNTED, the system does not select volumes based on their LAST\_DISMOUNT timestamps (unless /NOTDISMOUNTED is specified).
- **RESTRICTIONS:** If both /DISMOUNTED and /NOTDISMOUNTED are specified, the /DISMOUNTED date/time must precede the /NOTDISMOUNTED date/time.

**/FIRST\_ALLOCATION** Specifies the time period from which volumes are to be selected based on their first allocation dates. For two-sided volumes, the first allocation dates for both sides must be within the time period.

- **FORMAT:** /FIRST\_ALLOCATION=(from\_date<, to\_date>)
 

The from\_date specifies the oldest first allocation date-time, and the to\_date specifies the most recent. You must specify a from\_date. The date-times can be absolute or delta times.
- **DEFAULT:**
  - If you do not specify /FIRST\_ALLOCATION, the command does not select volumes by first allocation date.
  - If you specify /FIRST\_ALLOCATION but do not specify a to\_date, the default to\_date is the current date and time (when the command is invoked).

**/FIXED** Displays information for fixed volumes (level F).

- **FORMAT:** /FIXED
- **DEFAULT:** If you omit /FIXED, the volume ID parameter and other modifiers determine whether StorHouse displays information for fixed or removable

volumes.

- **RESTRICTIONS:** If you specify `/FIXED`, the volume identification code must specify a fixed volume. In addition, other modifiers that you specify must apply to fixed volumes.

`/LAST_`  
`ALLOCATION`

Specifies the time period when volumes are to be selected based on their last allocation dates. For two-sided volumes, the last allocation dates for both sides must be within the time period.

- **FORMAT:** `/LAST_ALLOCATION=(from_date<, to_date>)`

The `from_date` specifies the oldest last allocation date-time, and the `to_date` specifies the most recent. You must specify a `from_date`. The date-times can be absolute or delta times.

`/LOCATION`

Selects volumes in the specified location for display.

- **FORMAT:**
  - `/LOCATION=L`
  - `/LOCATION=Lxx`
  - `/LOCATION=S`
  - `/LOCATION=Sxx`
  - `/LOCATION=F`
  - `/LOCATION=Fxx`

where L is a library-level device, S is shelf storage, F is fixed storage, and xx is the unit number for the device.

- **DEFAULT:** If you omit `/LOCATION`, the command does not select volumes based on their location.



- /NOTDISMOUNTED** Selects volumes that have not been dismounted since the specified date/time. /NOTDISMOUNTED can roughly be interpreted as “not used since.” See the “data\_field date-time” topic for information about absolute and delta time specifications.
- **FORMAT:**
    - /NOTDISMOUNTED=absolute\_time
    - /NOTDISMOUNTED=delta\_time
  - **DEFAULT:** If you omit /NOTDISMOUNTED, the system does not select volumes based on their LAST\_DISMOUNT timestamps (unless /DISMOUNTED is specified).
  - **RESTRICTIONS:** If both /DISMOUNTED and /NOTDISMOUNTED are specified, the /DISMOUNTED date/time must precede the /NOTDISMOUNTED date/time.

- /NOTVALIDATED** Selects written volumes that have not been validated or that have not been validated since the specified date/time. For a full description of the absolute time and delta time data fields, see Appendix B, “Data Field, Parameter, and Modifier Definitions,” in the *Command Language Reference Manual*.
- Note the following. /NOTVALIDATED uses the LAST\_VALIDATED timestamp for the volume while the VALIDATE VOLUME command (with /VALIDATE\_DATE) selects volumes based on the VALIDATE\_DATE timestamp. Therefore, if you want to see which volumes would be selected by a VALIDATE VOLUME command, do not use SHOW VOLUME /NOTVALIDATED. Use VALIDATE VOLUME /PREVIEW instead.

- **FORMAT:**
  - /NOTVALIDATED
  - /NOTVALIDATED=absolute\_time
  - /NOTVALIDATED=delta\_time

For a full description of the absolute time and delta time data fields, see Appendix B, “Data Field, Parameter, and Modifier Definitions,” in the *Command Language Reference Manual*.

- **DEFAULT:** If you omit `/NOTVALIDATED`, the system does not select volumes based on their `LAST_VALIDATED` timestamps (unless `/VALIDATED` is specified).
- **RESTRICTIONS:** If both `/VALIDATED` and `/NOTVALIDATED` are specified, the `VALIDATED` date/time must precede the `NOTVALIDATED` date/time.

`/UNUSED` Selects volumes that have no more than a specified percentage or amount of their writable space remaining. For two-sided volumes, the unused space for each side must be within the percentage or amount. You can use this modifier to select volumes that do not have enough writable space left to hold a normal file or extent.

When using percentages, be aware of the sizes of volumes. For example, one percent of a one-gigabyte volume is 10 megabytes, which may be smaller than your normal file or extent size. On the other hand, one percent of a 10-gigabyte volume is 100 megabytes, which may be larger than your normal file or extent size.

**Note:** It is unlikely that there will be volumes with none of their writable space remaining.

- **FORMAT:**
  - `/UNUSED=number_of_bytes`
  - `/UNUSED=percentageP`

The value for `number_of_bytes` can range from 0 up to 4,294,967,295 (do not use commas in the specification). Use `K` to indicate 1,000-byte units, `M` to indicate 1,000,000-byte units, and `G` to indicate 1,000,000,000-byte units. If you do not specify `K`, `M`, or `G`, the number defaults to 1-byte units. The value for percentage can range from 0 to 100 and must be followed by the letter `P`.

- **DEFAULT:** If you omit `/UNUSED`, the command does not select volumes based on remaining writable space.

**/VALIDATED** Selects written volumes that have been validated or that have been validated since the specified date/time. (Because StorHouse only validates written volumes, the /VALIDATED display contains the most relevant list of volumes and excludes freepool volumes and cleaning cartridges.)

- **FORMAT:**
  - /VALIDATED
  - /VALIDATED=absolute\_time
  - /VALIDATED=delta\_time

For a full description of the absolute time and delta time data fields, see Appendix B, “Data Field, Parameter, and Modifier Definitions,” in the *Command Language Reference Manual*.

- **DEFAULT:** If you omit /VALIDATED, the system does not select volumes based on their LAST\_VALIDATED timestamps (unless /NOTVALIDATED is specified).
- **RESTRICTIONS:** If both /VALIDATED and /NOTVALIDATED are specified, the VALIDATED date/time must precede the /NOTVALIDATED date/time.

## Examples

- To show basic information for all level F volumes, enter:

```
? show volume * /fixed
```

```
MIA"F00":A VSET=MAGDISK
MIA"F01":A VSET=MAGDISK
MEA"F02":A VSET=.FOOMEA
MEA"F03":A VSET=.FOOMEA
MEA"F04":A VSET=BOB
MEA"F05":A VSET=FBACKUP
MEA"F06":A VSET=.FOOMEA
MEA"F07":A VSET=FVSET2
MEA"F08":A VSET=FVSET2
MEA"F09":A VSET=.FOOMEA
MEA"F0A":A VSET=.FOOMEA
MEA"F0B":A VSET=.FOOMEA
```

Total volume sides displayed=12

- To show all information for volume MIA"F00":A, enter:

? show volume MIA"F00":A /full

```
MIA"F00":A VSET=MAGDISK DIRECTORY=PRIMARY LOCATION=F00
CREATED=30-JAN-2006:11:29:18 MODIFIED=15-MAR-2006:00:15:01
FIRST=30-JAN-2006:11:30:15 LAST=15-MAR-2006:00:15:00
GENERAL_FREE=0KB
GENERAL_ALLOCATED=63488000KB
```

- To show basic information for all volumes in the free pool .FO0MEA, enter:

? show volume \* /free\_pool=.FO0MEA MEA"F02":A VSET=.FO0MEA

```
MEA"F02":A VSET=.FO0MEA
MEA"F03":A VSET=.FO0MEA
MEA"F06":A VSET=.FO0MEA
MEA"F09":A VSET=.FO0MEA
MEA"F0A":A VSET=.FO0MEA
MEA"F0B":A VSET=.FO0MEA
```

Total volume sides displayed=6

Note that you do not specify the period (.) in the free pool volume set name.

## SHOW VSET

StorHouse Release 5.6 adds a new modifier, `/FREE_POOL`, to the `SHOW VSET` command. This parameter enables you to display information about one or more free pool volume sets.

### Parameter Modifier

`/FREE_POOL` Selects free pool volume sets for display.

- **FORMAT:** `/FREE_POOL`
- **DEFAULT:** If you omit `/FREE_POOL`, the command selects user volume sets for display.
- **RESTRICTIONS:** `/FREE_POOL` is mutually exclusive with `/DIRECTORY`, `/HOLD`, `/MEMO`, and `/NOHOLD`.

### Examples

- To display basic information for all free pool volume sets, enter:

```
SHOW VSET * /FREE_POOL
```

```
VSET=.F00MEA DIRECTORY=none SIZE=69927076KB SURFACES=7
MEDIA=MEA
```

```
VSET=.F00MIA DIRECTORY=none SIZE=0KB SURFACES=0 MEDIA=MIA
```

```
VSET=.L00TDA DIRECTORY=none SIZE=343298410KB SURFACES=17
MEDIA=TDA
```

```
VSET=.L00TDZ DIRECTORY=none SIZE=0KB SURFACES=0 MEDIA=TDZ
```

```
VSET=.L01OEA DIRECTORY=none SIZE=0KB SURFACES=0 MEDIA=OEA
```

```
VSET=.L01OEB DIRECTORY=none SIZE=10095288KB SURFACES=10
MEDIA=OEB
```

```
VSET=.L01OEC DIRECTORY=none SIZE=0KB SURFACES=0 MEDIA=OEC
```

```
VSET=.L01OED DIRECTORY=none SIZE=31236317KB SURFACES=24
MEDIA=OED
```

■ ■ ■ ■ Chapter 2 – Changes to the StorHouse Interface

```
VSET=.L02TDA DIRECTORY=none SIZE=0KB SURFACES=0 MEDIA=TDA
```

```
VSET=.L02TDZ DIRECTORY=none SIZE=0KB SURFACES=0 MEDIA=TDZ
```

- To display all information for free pool volume set .L02TDZ, enter:

```
? SHOW VSET .L02TDZ /FREE_POOL /FULL
```

```
VSET=.L02TDZ DIRECTORY=none SIZE=0KB SURFACES=0 MEDIA=TDZ  
CREATED=11-MAR-2004:14:30:35 MODIFIED=22-MAR-2006:11:14:50  
LIBRARY=L02
```

## STAGE

StorHouse Release 5.6 adds the parameter modifier, `/FORCE`, to the `STAGE` command. It also supports the staging of files written to file sets on all level F volume sets except `MAGDISK`.

### Parameter Modifier

`/FORCE` Specifies that StorHouse will stage files to the performance buffer even when the staging will trigger a migration operation to make more performance buffer space available.

- **FORMAT:** `/FORCE`
- **DEFAULT:** If you omit `/FORCE`, files are not staged to the performance buffer if doing so will trigger a migration operation (in other words, if it results in less than `MIG_MIN%` available space in the performance buffer).
- **RESTRICTIONS:** If you specify `/FORCE`, you may not specify a file that resides in a file set in the `MAGDISK` volume set.

### Example

To stage the file, `STATEMENTS`, which resides in your default file access group, to the performance buffer even if the staging operation will trigger a migration, enter:

```
? STAGE STATEMENTS /FORCE
```

The command will queue transfers of all extents of the `STATEMENTS` file to the performance buffer.

## VALIDATE DEVICE

VALIDATE DEVICE verifies that StorHouse can read all file extents on a level F device.

### Format

VALIDATE DEVICE did

COMMAND FORMAT SUMMARY					
COMMAND, PARAMETER, OR MODIFIER	REQUIRED COMMAND PRIVILEGE	REQUIRED GROUP ACCESS	REQUIRED FILE ACCESS	MINIMUM ACCOUNT ACCESS	DEFAULT
VALIDATE DEVICE	SYSTEM	-	-	-	-
/CONFIRM	-	-	-	-	/NOCONFIRM
/MAX_ERRORS=...	-	-	-	-	-
/NODATA	-	-	-	-	-
/PB_PENDING	-	-	-	-	/PB_PENDING
/REPORT	-	-	-	-	/NOREPORT
did	-			-	(required)
/DIRECTORY=...	-	-	-	-	-
/VSET=...	-	-	-	-	-

### Description

VALIDATE DEVICE verifies that all file extents on a level F device can be read. The command does not validate extents belonging to deleted or locked files or extents located on freepool, offline, or non-cataloged devices.

StorHouse typically validates pending extents (performance buffer extents that have not yet been written back to their resident file sets) by reading the performance buffer copy. You can instruct StorHouse to skip pending extents by specifying the /NOPB\_PENDING modifier.

If any device cannot be validated, contact your SGI customer support representative.



## Parameter

**did** Specifies the device identification code(s) of the device(s) to be validated.

- **FORMAT:**

- Fxx
- Fx\*
- F\*

where xx is the device number, x is a partial device number, or the wildcard (\*) matches all numbers.

- **DEFAULT:** None, you must specify this parameter.
- **RESTRICTIONS:** Only level F devices can be specified.

## Command Modifiers

**/CONFIRM** Controls whether StorHouse prompts you to confirm the command.

- **FORMAT:**

- /CONFIRM
- /NOCONFIRM

- **DEFAULT:** /NOCONFIRM

When the system requests a confirmation, enter YES or NO. YES (entered as Y, YE, or YES) directs the system to perform the command. Do not place a space, tab, or other character after the YES response. If you do, StorHouse will not recognize the response as a YES and will not execute the command. If you press enter or enter any characters other than those described as a YES response, StorHouse interprets the response as NO.

**/MAX\_ERRORS** Controls the maximum number of errors StorHouse allows when it finds file extents that are not accessible on a device. If the number of errors reaches this limit, StorHouse stops processing the current device.

- **FORMAT:**

- /MAX\_ERRORS=number

- /MAX\_ERRORS=0
  - /MAX\_ERRORS
- Specifying either /NOMAX\_ERRORS or /MAX\_ERRORS=0 indicates no limit on the number of possible errors.

- DEFAULT: /MAX\_ERRORS=0

**/NODATA** Indicates that the data portion of each file extent should not be read. However, the file extents will be accessed to ensure that they are present on the media.

- FORMAT: /NODATA
- DEFAULT: No /NODATA modifier

**/PB\_PENDING** Indicates whether the command should validate pending extents (extents in the performance buffer that have not yet been written back to their resident FSET). /PB\_PENDING instructs StorHouse to validate pending extents. /NOPB\_PENDING instructs StorHouse not to validate pending extents.

- FORMAT:
  - /PB\_PENDING
  - /NOPB\_PENDING
- DEFAULT: /PB\_PENDING

## Parameter Modifiers

**/DIRECTORY** Indicates the devices in the specified directory will be selected for validation.

- FORMAT:
  - /DIRECTORY=PRIMARY
  - /DIRECTORY=BACKUP
  - /DIRECTORY=ARCHIVE
- DEFAULT: If you omit /DIRECTORY, the system does not select devices based on a specific directory.
- RESTRICTIONS: /DIRECTORY and /VSET are mutually exclusive.



**/VSET** Validates devices that make up the specified volume set.

- **FORMAT:** /VSET=vset\_name
- **FORMAT RESTRICTIONS:** No wildcards are allowed.
- **DEFAULT:** If you omit /VSET, the system does not use the volume set name to select devices. In other words, the system validates all devices unless restricted by other modifiers.
- **RESTRICTIONS:** /VSET and /DIRECTORY are mutually exclusive.

### Example

- To validate device F02, enter:

```
? VALIDATE DEVICE F02
```

The system responds with the following message:

```
Extents: Total=574 Good=574 Skipped=0 Bad=0
```

If the system identifies any unreadable extents, notify your SGI customer support representative.

- To validate all devices in the BACKUP\_F volume set, enter:

```
? VALIDATE DEVICE F* /VSET=BACKUP_F
```

## VALIDATE FILE

The VALIDATE FILE command verifies that StorHouse can read the extents of a specific file.

### Format

VALIDATE FILE filename

COMMAND FORMAT SUMMARY					
COMMAND, PARAMETER, OR MODIFIER	REQUIRED COMMAND PRIVILEGE	REQUIRED GROUP ACCESS	REQUIRED FILE ACCESS	MINIMUM ACCOUNT ACCESS	DEFAULT
VALIDATE FILE	SYSTEM	-	-	-	-
/MAX_ERRORS=...	-	-	-	-	-
/NODATA	-	-	-	-	-
/PENDING	-	-	-	-	/PENDING
/REPORT	-	-	-	-	-
filename	-	R	R	-	(required)
/DIRECTORY=...	-	-	-	-	-
/GROUP=...	SETGROUP	R	-	-	current default
/VERSION=...	-	-	-	-	/VERSION=0

### Description

VALIDATE FILE verifies that StorHouse can read all extents of a specific file. The command does not validate extents belonging to deleted files. If any file cannot be validated, notify your SGI customer support representative.

StorHouse typically validates the performance buffer copy of pending extents. A pending extent is a file extent that has not been written back to its resident file set.

VALIDATE FILE validates the extents of a file in their preferred order (i.e., the order that they would generally be written to tape).

## Parameter

**filename** Specifies the StorHouse file or file extent to be validated.

- **FORMAT:** filename
- **DEFAULT:** None; you must specify this parameter.
- **ACCESS REQUIREMENTS:** Read access to the file and group.

## Command Modifiers

**/MAX\_ERRORS** Controls the maximum number of errors StorHouse will allow when it finds inaccessible extents in a specific file. If the number of errors reaches this limit, StorHouse stops processing the file.

- **FORMAT:**
  - /MAX\_ERRORS=number
  - /MAX\_ERRORS=0
  - /NO\_MAX\_ERRORS

Specifying either /NOMAX\_ERRORS or /MAX\_ERRORS=0 indicates an unlimited number of errors.

- **DEFAULT:** /MAX\_ERRORS=0

**/NODATA** Indicates that the data portion of each file extent should not be read. However, the file extents will be accessed to ensure that they are present on the media.

- **FORMAT:** /NODATA
- **DEFAULT:** No /NODATA modifier

**/PENDING** Indicates whether the command should validate pending extents. /PENDING validates pending extents. /NOPENDING does not validate pending extents.

- **FORMAT:**
  - /PENDING
  - /NOPENDING

- DEFAULT: /PENDING

**/REPORT** Controls the generation of special text responses for the completion of significant actions. /REPORT instructs StorHouse to generate a text response. /NOREPORT instructs StorHouse not to generate a text response (other than a summary line).

- FORMAT:
  - /REPORT
  - /NOREPORT
- DEFAULT: /NOREPORT

### Parameter Modifiers

**/DIRECTORY** Specifies which directory to use for validating files.

- FORMAT:
  - /DIRECTORY=ARCHIVE
  - /DIRECTORY=BACKUP
  - /DIRECTORY=PRIMARY
- DEFAULT: /DIRECTORY=PRIMARY

**/GROUP** Specifies a file access group name and, optionally, the group's read password.

- FORMAT: /GROUP=groupname<:readpw>
- DEFAULT: If you omit /GROUP, the default is your default file access group.
- ACCESS REQUIREMENTS: Read access to the group.  
You must specify the read password unless:
  - The group is not protected by a read password.
  - Your privilege bypasses read access password checks.
  - Your default access to the group indicates read access.
- PRIVILEGE: You must have SETGROUP privilege to specify any group except



your default group.

**/VERSION** Selects the specified version of a file for validation. The versions are relative version numbers in the directory selected by **/DIRECTORY**. If you omit **/DIRECTORY**, the versions are relative version numbers in the primary directory.

- **FORMAT:** **/VERSION=version**

The value of version can range from 0 through -32767 for primary files. For files in the archive and backup directories, the range is 0 through -65535.

- **DEFAULT:** **/VERSION=0.**

### Example

To validate version 0 of the file **USERFILE** in your default file access group, enter:

```
? VALIDATE FILE USERFILE
```

The system responds with a message:

```
Extents: Total=4 Good=4 Pending=0 Bad=0
```

If the system identifies any unreadable extents, contact your SGI customer support representative.

## VALIDATE VOLUME

The VALIDATE VOLUME command verifies that all file extents on a level L or level S volume can be read.

### Format

VALIDATE VOLUME vid

COMMAND FORMAT SUMMARY					
COMMAND, PARAMETER, OR MODIFIER	REQUIRED COMMAND PRIVILEGE	REQUIRED GROUP ACCESS	REQUIRED FILE ACCESS	MINIMUM ACCOUNT ACCESS	DEFAULT
VALIDATE VOLUME	SYSTEM	-	-	-	-
/CONFIRM	-	-	-	-	/NOCONFIRM
/DEADLINE=...	-	-	-	-	No start time limit
/LIMIT=...					No limit
/MAX_ERRORS=...	-	-	-	-	-
/NODATA	-	-	-	-	-
/PENDING	-	-	-	-	/PENDING
/PREVIEW	-	-	-	-	-
/REPORT	-	-	-	-	/NOREPORT
/RESTART	-	-	-	-	/RESTART
vid	-	-	-	-	(required)
/DIRECTORY=...	-	-	-	-	-
/LOCATION=...	-	-	-	-	-
/VALIDATE_DATE	-	-	-	-	-
/VSET=...	-	-	-	-	-

### Description

VALIDATE VOLUME verifies that all file extents on a volume can be read. This command will not validate extents that belong to deleted files. If any volume cannot be validated, contact your SGI customer support representative.



Pending extents (extents in the performance buffer which have not yet been written back to their resident FSET) are usually validated by reading the performance buffer copy. However, pending extents can be skipped by using the /NOPENDING modifier, if desired.

If a validate of a volume is interrupted by reaching the deadline time or by a system shutdown, the command can save information to allow a subsequent validate of that volume to resume validation at the point it was interrupted.

It is common to schedule a periodic validate (e.g., every night or every weekend) as part of regular tape maintenance. Usually the volume id would be (at least partially) wildcarded, and other selection modifiers (e.g., /VSET) might also be used to restrict the volumes to be validated. /VALIDATE\_DATE should also be used so that only volumes that have not been validated recently would be validated.

The validate command only validates one volume at a time. If you want to validate multiple volumes at a time, you can schedule more than one validate command. (If two validate commands try to validate the same volume at the same time, the first command will invalidate the volume, and the second command will skip it.)

### Automatic Periodic Validation

VALIDATE VOLUME supports automatic periodic validation as follows. Once data is written to a volume, that volume will be validated N days later and again every N days after that. The process is automatic and customizable.

**Volume Timestamps.** To support automatic periodic validates, StorHouse maintains these timestamps for every volume.

- VALIDATE\_DATE
- LAST\_VALIDATED

VALIDATE VOLUME uses the VALIDATE\_DATE timestamp to determine when to validate a volume. StorHouse initializes VALIDATE\_DATE to the current date/time when the first file (extent) is written to the volume. Then StorHouse updates the VALIDATE\_DATE timestamp to the current date/time when the validation occurs. The VALIDATE\_DATE can also be set by user request on several commands.

The LAST\_VALIDATED timestamp indicates the last successful validate of the volume. StorHouse clears this timestamp when the volume enters the system (by

MIGRATE /BLANKS, IMPORT, or RECOVER DEVICE), and then updates it to the current date/time after every successful validate of the volume.

After the first successful validate, the two timestamps will usually match. A failed validate does not change either timestamp.

**Validate Frequency.** A volume's validate frequency indicates how often the volume should be validated expressed in units of days. In other words, the validate frequency of a volume is the desired maximum age of non-validated data on the volume.

The appropriate validate frequency of a volume is site-specific and should normally be based on media type and the importance of data on the volume. Therefore, periodic validates may not be appropriate for some volumes.

You can specify the validate frequency on the VALIDATE VOLUME command. However, for convenience, SGI can configure your system with a default validate frequency for each configured media type. If there is no specifically defined default frequency for a given media type, StorHouse uses zero for the default validate frequency.

**Performing Periodic Validates.** To perform a periodic validate, specify the /VALIDATE\_DATE modifier on the VALIDATE VOLUME command. When /VALIDATE\_DATE is specified, VALIDATE VOLUME will validate a volume based on the volume's VALIDATE\_DATE timestamp and the desired validate frequency.

You typically specify /VALIDATE\_DATE as a delta time. For example, for a validate frequency of one year, specify "/VALIDATE\_DATE=D365-0." In this example, StorHouse will validate a volume if it contains any data that has not been validated in the past year.

You can also specify /VALIDATE\_DATE as an absolute time. Again, the volume will be validated only if its VALIDATE\_DATE timestamp is earlier than or equal to that date/time.

If you omit a value for /VALIDATE\_DATE, the command uses the default validate frequency for the volume's media type. If the default value is non-zero, it is expressed as a delta time.

If the default validate frequency for the volume's media type is zero, you can specify the `VALIDATE VOLUME * /VALIDATE_DATE` command to simply skip all volumes of all media types where validation does not apply.

Without the `/VALIDATE_DATE` modifier, the `VALIDATE VOLUME` command validates all volumes selected by the specified volume id and any other modifiers, and ignores the volume's `VALIDATE_DATE` timestamp.

With or without `/VALIDATE_DATE`, if the validate process is successful, the command updates the volume's `VALIDATE_DATE` and `LAST_VALIDATED` timestamps.

**Scheduling Automatic Periodic Validates.** For automatic periodic validates, schedule one or more `VALIDATE VOLUME` commands to run on a regular basis. Typically, you will use a wildcarded volume ID on the `SCHEDULE` command. For more precise selection, you may use a partial wildcarded volume ID, and/or `/DIRECTORY`, `/LOCATION`, and/or `/VSET` modifiers. This technique is useful for indicating different values such as validate frequencies for different volumes.

- You can use the `/DEADLINE`, `/LIMIT`, and/or `/MAX_ERRORS` modifiers to limit the scope of the `VALIDATE VOLUME` command.
- You can schedule multiple concurrent `VALIDATE VOLUME` commands to provide multiple concurrent validates. (A single `VALIDATE VOLUME` command validates just one volume at a time.)
- You can use the `/EMAIL` and `/ON_ERROR` modifiers on the `SCHEDULE` command to report any problems to an administrator.

**Setting a Value for `VALIDATE_DATE`.** Normally, the system manages a volume's `VALIDATE_DATE` timestamp. However, if desired, you can set the value with the `SET VOLUME /VALIDATE_DATE` command. The `VALIDATE_DATE` timestamp cannot be set earlier than the volume's `INITIALIZED` timestamp, and it cannot be set to a future date/time.

**Note the following.**

When a volume containing files is brought into StorHouse by an `IMPORT` or `RECOVER DEVICE` command, its `VALIDATE_DATE` timestamp should ideally be restored to the timestamp value in the previous StorHouse system. However, because the new StorHouse does not have access to the old timestamp value, it defaults the

VALIDATE\_DATE timestamp to the INITIALIZED timestamp (the date/time the volume was originally migrated into StorHouse and labeled).

You can set the VALIDATE\_DATE timestamps of such volumes to another value by specifying the /VALIDATE\_DATE modifier on the IMPORT or RECOVER DEVICE commands that bring the volumes into the new StorHouse system. Alternatively, you can set the timestamps anytime thereafter using the /VALIDATE\_DATE modifier on the SET VOLUME command. If the actual value from the previous StorHouse is available, the VALIDATE\_DATE timestamp could be set to that value or another appropriate value.

## Parameters

**vid** Specifies the volume identification code (vid) of the volume that you want to validate.

■ **FORMAT:**

- {media\_type}{recording\_type}{volume\_label}:{side}
- {media\_type}{recording\_type}{volume\_label}\*
- {media\_type}{recording\_type}\*
- {media\_type}\*
- {partial\_media\_type}\*
- \*

The braces “{ }” are not part of the specification. See Appendix B, “Data Field, Parameter, and Modifier Definitions” in the *Command Language Reference Manual* for a description of the vid data field.

- **FORMAT RESTRICTION:** The volume must be a level L (library) or level S (shelf) volume.
- **DEFAULT:** None; you must specify this parameter.

## Command Modifiers

**/CONFIRM** Controls whether StorHouse asks you to confirm the command.

- **FORMAT:** /CONFIRM or /NOCONFIRM
- **DEFAULT:** /NOCONFIRM

When the system requests a confirmation, enter YES (also Y or YE) or NO (also N). If you press “Enter” or enter any characters other than those described as a YES response, StorHouse interprets them as NO.

**/DEADLINE** Specifies the date and time after the VALIDATE VOLUME command is invoked when the command can start validating extents. When the /DEADLINE expires, the command will complete validation of any additional extents or volumes.

- **FORMAT:**
  - /DEADLINE=absolute\_time
  - /DEADLINE=delta\_time

For a full description of the absolute time and delta time data fields, see Appendix B, “Data Field, Parameter, and Modifier Definitions,” in the *Command Language Reference Manual*.

- **DEFAULT:** If you omit /DEADLINE, the command is not limited by time.

**/LIMIT** Specifies the maximum number of volumes the command can select.

- **FORMAT:** /LIMIT=n  
where n is the maximum number of volumes the command can select.
- **DEFAULT:** If you omit /LIMIT, the command is not limited by number of volumes.

**/MAX\_ERRORS** Controls the maximum number of errors StorHouse allows when it finds file extents that are not accessible on a volume. If the number of errors reaches this limit, StorHouse stops processing the current volume.

- **FORMAT:**
  - /MAX\_ERRORS=number

- /MAX\_ERRORS=0

- /NOMAX\_ERRORS

- DEFAULT: /MAX\_ERRORS=0

Specifying either /NOMAX\_ERRORS or /MAX\_ERRORS=0 indicates that the number of possible errors is not limited.

**/NODATA** Indicates that the data portion of each file extent should not be read. However, the file extents will be accessed to ensure that they are present on the media.

- FORMAT: /NODATA

- DEFAULT: No /NODATA modifier

**/PENDING** Indicates whether the command should validate pending extents (performance buffer extents that have not been written back to their resident file set). /PENDING instructs the command to validate pending extents. /NOPENDING instructs the command not to validate pending extents.

- FORMAT:

- /PENDING

- /NOPENDING

- DEFAULT: /PENDING

**/PREVIEW** Indicates that the command will only list the volume IDs of the volumes that would be validated with the command. /PREVIEW does not actually validate the volumes.

- FORMAT: /PREVIEW

- /DEFAULT: If /PREVIEW is omitted, the command validates volumes.



**/REPORT** Controls the generation of special text responses for the completion of significant actions. /REPORT instructs StorHouse to generate a text response. /NOREPORT instructs StorHouse not to generate a text response.

- **FORMAT:** /REPORT or /NOREPORT
- **DEFAULT:** /NOREPORT

**/RESTART** Indicates whether the command should allow a restart of the volume being validated if interrupted by reaching the deadline time or by a system shutdown. /RESTART instructs the command to allow a restart. /NORESTART instructs the command not to allow a restart.

Allowing a restart is generally useful because any extents that had just been validated before the interruption do not need to be validated again so soon. The system will not restart the volume validation if the restart information is out-of-date.

An interrupted VALIDATE VOLUME command cannot be restarted if either the interrupted VALIDATE VOLUME command or the subsequent VALIDATE VOLUME command specifies /NORESTART.

- **FORMAT:**
  - /RESTART
  - /NORESTART
- **DEFAULT:** /RESTART

## Parameter Modifiers

**/DIRECTORY** Indicates that volumes in the specified directory will be selected for validation.

- **FORMAT:**
  - /DIRECTORY=PRIMARY
  - /DIRECTORY=BACKUP
  - /DIRECTORY=ARCHIVE
- **DEFAULT:** If you omit /DIRECTORY, the system does not select volumes based

on a specific directory.

- RESTRICTIONS: /DIRECTORY and /VSET are mutually exclusive.

**/LOCATION** Selects volumes in the specified location for validation.

- FORMAT:

- /LOCATION=L
- /LOCATION=Lxx
- /LOCATION=S
- /LOCATION=Sxx

where L is a library-level device, s is shelf storage, and xx is the unit number for the device.

- DEFAULT: If you omit /LOCATION, the command does not select volumes based on their location.

**/VALIDATE\_**  
**DATE** Indicates to only validate volumes with a VALIDATE\_DATE timestamp equal to or earlier than the specified date/time. Using a delta time here is often convenient, especially when scheduling a VALIDATE VOLUME command. See the “data\_field date-time” topic for information about absolute and delta time specifications.

If /VALIDATE\_DATE is specified without a date/time value, the system will look up the default validate frequency for the volume's media type. If zero, the volume will not be validated. Otherwise, the system will take the default validate frequency and create a timestamp that many days before the current date/time, and use that timestamp to compare to the volume's VALIDATE\_DATE timestamp.

You can check the default validate frequency for a media type with the SHOW CONFIGURATION /MEDIA command. If a default has been defined, it will appear as VALIDATE\_FREQ.

- FORMAT:

- /VALIDATE\_DATE
- /VALIDATE\_DATE=absolute\_time



- /VALIDATE\_DATE=delta\_time

For a full description of the absolute time and delta time data fields, see Appendix B, “Data Field, Parameter, and Modifier Definitions,” in the *Command Language Reference Manual*.

- DEFAULT: If you omit /VALIDATE\_DATE, the system does not validate volumes based on their VALIDATE\_DATE timestamp.

**/VSET** Validates volumes in the specified volume set.

- FORMAT: /VSET=vset\_name
- FORMAT RESTRICTIONS: Wild cards are not allowed.
- DEFAULT: If you omit /VSET, the system does not use the volume set name to select volumes. In other words, all volumes are validated unless restricted by another modifier.
- RESTRICTIONS: /DIRECTORY and /VSET are mutually exclusive

## Examples

- To validate side A of volume OED34A38DAE, enter:

```
? VALIDATE VOLUME OED34A38DAE:A
```

The system responds with a message:

```
Extents: Total=574 Good=574 Pending=0 Bad=0
```

*Pending* means that the performance buffer contains a copy of an extent that has not yet been written back to optical or tape.

If the system identifies any unreadable (bad) extents, contact your SGI customer support representative.

- To validate all volumes in the BILLING volume set, which currently reside in a library device, and to terminate the command if it is still running after eight hours, enter:

```
? VALIDATE VOLUME * /VSET=BILLING /LOCATION=L /DEADLINE=D8
```

- To validate side A of volume OED34A38DAE and limit the number of possible errors to 5, enter:

```
? VALIDATE VOLUME OED34A38DAE:A /MAX_ERRORS=5
```

## Absolute Time Specification

The absolute time specification now supports date keywords.

**Absolute Time** Specifies a calendar date (or date keyword) and clock time as follows:

```
<<day>-<month>-<year>><:<hour><:<minute><:<second>>>
```

or

```
<<date keyword><:<minute><:<second>>>
```

The date subfields can have the following values:

- day – 1 through 31
- month – JAN, FEB, MAR, APR, MAY, JUN, JUL, AUG, SEP, OCT, NOV, DEC
- year – 1984 through 9999
- hour – 0 through 23
- minute – 0 through 59
- second – 0 through 59

Midnight is hour 0; noon is hour 12. Days begin at midnight; years begin at midnight on January 1. StorHouse stores absolute time fields as the number of seconds since midnight, January 1, 1970. If a date/time field has been set to its initial value, but no event has occurred to change it, the system normally displays it as midnight, January 1, 1970.

If you do not specify a day, month or year, the current day, month or year, respectively, is the default. If you do not specify an hour, minute or second, the subfield defaults to zero.



The date keywords can have the following values:

- YESTERDAY
- TODAY
- TOMORROW
- LAST
- NEXT

For example, NEXT:20 indicates the next 8:00 P.M., which could be either today or tomorrow, depending on when you run the command.

## Miscellaneous Changes

StorHouse Release 5.6 provides the following other changes to the Command Language:

- On the BACKUP command, /DRIVES does not apply to level F drives.
- Users can execute more than one RETIRE VOLUME command at a time.
- RESTORE DIRECTORY restores file set and volume set information for files on level F.
- VALIDATE FILE and VALIDATE VOLUME check for pending extents.

## Generic Callable Interface

This section provides the changes to the LSMCO and LSMOS Generic Callable Interface Functions.

### LSMCO - Create Open

StorHouse Release 5.6 adds the `retention_interval` member to the `LSMS_CATTR` structure.

#### Function Prototype Definition

```
extern int LSMCO ( struct LSMS_TOKEN *c_token,
                  int message_flag,
                  struct LSMS_TOKEN *o_token,
                  char *file_name,
                  char *file_password,
                  char *group_name,
                  char *group_password,
                  char *model_file_name,
                  struct LSMS_FLOC *file_location,
                  struct LSMS_CATTR *file_attrib
                  ) ;
```

#### Data Structures

```
struct LSMS_FLOC
{
    char volumeset_name[ 9 ];
    char fileset_name[ 9 ];
};

struct LSMS_CATTR
{
    long list_size;
    long block_size;
    long checkpt;
    long file_size;
    long data_xlate_flag;
    long atf;
    long cache;
    long edc;
    long limit;
```

```

    long vtf;
    long retention_interval;

};

```

## Argument Description

The `retention_interval` attribute of the `LSMS_CATTR` structure is the retention attribute for the file. Valid values are:

- Number of days specified as a non-zero, positive integer (for example, 60).
- `LSMV_RETEN_FOREVER`, which indicates infinite retention.
- `LSMV_RETEN_ZERO`, which indicates the file has no retention period and can be deleted.
- `LSMV_DEFAULT`, which indicates the retention period is not specified at the file level and assumes the default value.
  - If the file's resident file set has a retention attribute equal to `FOREVER`, `ZERO`, or a specified number of days, the file set retention attribute determines the file retention attribute.
  - If the file's resident file set has a retention attribute of `DEFAULT`, the `RETENTION_MODE` system parameter determines the file retention attribute. If `RETENTION_MODE` is set to `BASIC`, the file retention is `ZERO`. If `RETENTION_MODE` is set to `STRICT`, the file retention is `FOREVER`.

**Note:** If force retention is set at the file set level, the file assumes the file set retention value.

## LSMOS – Open Sequential

StorHouse Release 5.6 adds the `retention_interval` member to the file attribute list.

### Function Prototype Definition

```

extern int LSMOS ( struct LSMS_TOKEN *c_token,
                  int message_flag,
                  struct LSMS_TOKEN *o_token,
                  char *mode,
                  char *file_name,
                  long version,

```

```

        struct LSMS_FPW *file_passwords,
        char *group_name,
        struct LSMS_FPW *group_passwords,
        struct LSMS_FLOC *file_location,
        struct LSMS_ATTR *file_attr,
        struct LSMS_OPTS *file_options
    );

```

## Data Structures

```

struct LSMS_FPW
{
    char  read_password[ 9 ];
    char  write_password[ 9 ];
    char  delete_password[ 9 ];
};

struct LSMS_FLOC
{
    char  volumeset_name[ 9 ];
    char  filesset_name[ 9 ];
};

struct LSMS_ATTR
{
    long  list_size;
    long  file_size;
    long  max_record_len;
    long  transport_flag;
    long  data_xlate_flag;
    long  fixed_record_fl;
    long  cc_ansi_flag;
    long  cc_mach_flag;
    long  block_size;
    long  retention_interval;
};

struct LSMS_OPTS
{
    long  list_size;
    long  lock;
    long  wait;
    long  atf;
};

```



```
    long   edc;  
    long   limit;  
    long   new; /* This will be newx if compiled with C++. */  
    long   unlock;  
    long   vtf;  
};
```

### Argument Description

The `retention_interval` member is the retention period for the file. Valid values are:

- Number of days specified as a non-zero, positive integer (for example, 60).
- `LSMV_RETEN_FOREVER`, which indicates infinite retention.
- `LSMV_RETEN_ZERO`, which indicates the file has no retention period and can be deleted.
- `LSMV_RETEN_DEFAULT`, which indicates the retention period is not specified at the file level and assumes the default value.
- If the file's resident file set has a retention attribute equal to `FOREVER`, `ZERO`, or a specified number of days, the file set retention attribute determines the file retention attribute.
- If the file's resident file set has a retention attribute of `DEFAULT`, the `RETENTION_MODE` system parameter determines the file retention attribute. If `RETENTION_MODE` is set to `BASIC`, the file retention is `ZERO`. If `RETENTION_MODE` is set to `STRICT`, the file retention is `FOREVER`.

**Note:** If force retention is set at the file set level, the file assumes the file set retention value.

## Mainframe Callable Interface

This section contains changes to the mainframe Callable Interface CREATE-OPEN and OPEN-SEQ functions.

### CREATE-OPEN

StorHouse Release 5.6 adds the FATTR-RETENTION-INTERVAL element to the FILE-ATTRIB array.

#### Statement Format for COBOL

##### TSO/Batch/IMS Environment

```
CALL 'LSMCALL' USING CREATE-OPEN, C-TOKEN, R-CODE,  
MESSAGE-FLAG, O-TOKEN, FILE-NAME  
FILE-PASSWORD, GROUP-NAME,  
GROUP-PASSWORD, MODEL-FILE-NAME, FILE-  
LOCATION, FILE-ATTRIB.
```

##### CICS Environment

```
CALL 'LSMCICS' USING DFHEIBLK, COMMAREA,  
CREATE-OPEN, C-TOKEN, R-CODE,  
MESSAGE-FLAG, O-TOKEN, FILE-NAME,  
FILE-PASSWORD, GROUP-NAME,  
GROUP-PASSWORD, MODEL-FILE-NAME,  
FILE-LOCATION, FILE-ATTRIB.
```





## Working Storage Section for COBOL Program

```
01 CREATE-OPEN PIC X(16) VALUE 'CREATE-OPEN'.
01 C-TOKEN PIC S9(8) COMP SYNC.
01 R-CODE PIC S9(8) COMP SYNC.
01 MESSAGE-FLAG PIC S9(8) COMP SYNC.
01 O-TOKEN PIC S9(8) COMP SYNC.
01 FILE-NAME PIC X(56).
01 FILE-PASSWORD PIC X(8).
01 GROUP-NAME PIC X(8).
01 GROUP-PASSWORD PIC X(8).
01 MODEL-FILE-NAME PIC X(56).
01 FILE-LOCATION.
    05 VOLUMESET-NAME PIC X(8)
    05 FILESET-NAME PIC X(8).
01 FILE-ATTRIB.
    05 FATTR-LIST-SIZE PIC S9(8) COMP SYNC VALUE 10.
    05 FATTR-BLOCK-SIZE PIC S9(8) COMP SYNC.
    05 FATTR-CHECKPOINT PIC S9(8) COMP SYNC.
    05 FATTR-FILE-SIZE PIC S9(8) COMP SYNC.
    05 FATTR-DATA-XLATE PIC S9(8) COMP SYNC.
    05 FATTR-ATF PIC S9(8) COMP SYNC.
    05 FATTR-CACHE PIC S9(8) COMP SYNC.
    05 FATTR-EDC PIC S9(8) COMP SYNC.
    05 FATTR-LIMIT PIC S9(8) COMP SYNC.
    05 FATTR-VTF PIC S9(8) COMP SYNC.
    05 FATTR-RETENTION-INTERVAL PIC S9(8) COMP SYNC.
```

## Parameter Overview

The FATTR-RETENTION-INTERVAL is the retention period for the file. Valid values are:

- Non-zero positive integer (for example, 60), which indicates the number of days to retain the file.
- -2, which indicates to retain the file forever.
- -1, which indicates zero days, or no retention.
- 0, which indicates the retention period is not specified at the file level and assumes the default value. StorHouse determines the default value as follows:

- If the file’s resident file set has a retention attribute equal to FOREVER, ZERO, or a specified number of days, the file set retention attribute determines the file retention attribute.
- If the file’s resident file set has a retention attribute of 0, the RETENTION\_MODE system parameter determines the file retention attribute. If RETENTION\_MODE is set to BASIC, the file retention is ZERO. If RETENTION\_MODE is set to STRICT, the file retention is FOREVER.

**Note:** If force retention is set at the file set level, the file assumes the file set retention value.

## OPEN-SEQ

StorHouse Release 5.6 adds the FATTR-RETENTION-INTERVAL to the FILE-ATTRIB list.

### Statement Format for COBOL

#### TSO/Batch/IMS Environment

```
CALL 'LSMCALL' USING OPEN-SEQ, C-TOKEN, R-CODE, MESSAGE-FLAG,
                     O-TOKEN, MODE, FILE-NAME, VERSION,
                     FILE-PASSWORDS, GROUP-NAME,
                     GROUP-PASSWORDS, FILE-LOCATION,
                     FILE-ATTRIB, FILE-OPTIONS.
```

#### CICS Environment

```
CALL 'LSMCICS' USING DFHEIBLK, COMMAREA,
                     OPEN-SEQ, C-TOKEN, R-CODE, MESSAGE-FLAG,
                     O-TOKEN, MODE, FILE-NAME, VERSION,
                     FILE-PASSWORDS, GROUP-NAME,
                     GROUP-PASSWORDS, FILE-LOCATION,
                     FILE-ATTRIB, FILE-OPTIONS.
```



## Working Storage Section for COBOL Program

```
01 OPEN-SEQ                PIC X(16)    VALUE 'OPEN-SEQ' .
01 C-TOKEN                 PIC S9(8)    COMP SYNC .
01 R-CODE                  PIC S9(8)    COMP SYNC .
01 MESSAGE-FLAG           PIC S9(8)    COMP SYNC .
01 O-TOKEN                 PIC S9(8)    COMP SYNC .
01 MODE                    PIC X(6) .
01 FILE-NAME               PIC X(56) .
01 VERSION                 PIC S9(8)    COMP SYNC .
01 FILE-PASSWORDS .
   05 FILE-READ-PASSWORD   PIC X(8)    VALUE SPACES .
   05 FILE-WRITE-PASSWORD  PIC X(8)    VALUE SPACES .
   05 FILE-DELETE-PASSWORD PIC X(8)    VALUE SPACES .
01 GROUP-NAME              PIC X(8) .
01 GROUP-PASSWORDS .
   05 GROUP-READ-PASSWORD  PIC X(8)    VALUE SPACES .
   05 GROUP-WRITE-PASSWORD PIC X(8)    VALUE SPACES .
   05 GROUP-DELETE-PASSWORD PIC X(8)    VALUE SPACES .
01 FILE-LOCATION .
   05 VOLUMESET-NAME       PIC X(8) .
   05 FILESET-NAME         PIC X(8) .
01 FILE-ATTRIB .
   05 FATTR-LIST-SIZE      PIC S9(8)    COMP SYNC VALUE 9 .
   05 FATTR-FILE-SIZE     PIC S9(8)    COMP SYNC .
   05 FATTR-MAX-RECORD-LEN PIC S9(8)    COMP SYNC .
   05 FATTR-TRANSPORT-FLAG PIC S9(8)    COMP SYNC .
   05 FATTR-DATA-XLATE-FLAG PIC S9(8)    COMP SYNC .
   05 FATTR-FIXED-RECORD-FL PIC S9(8)    COMP SYNC .
   05 FATTR-CC-ANSI-FLAG  PIC S9(8)    COMP SYNC .
   05 FATTR-CC-MACH-FLAG  PIC S9(8)    COMP SYNC .
   05 FATTR-BLOCK-SIZE    PIC S9(8)    COMP SYNC .
   05 FATTR-RETENTION-INTERVAL PIC S9(8)    COMP SYNC .
01 FILE-OPTIONS .
   05 FOPTS-LIST-SIZE     PIC S9(8)    COMP SYNC VALUE 8 .
   05 FOPTS-LOCK         PIC S9(8)    COMP SYNC .
   05 FOPTS-WAIT         PIC S9(8)    COMP SYNC .
   05 FOPTS-ATF          PIC S9(8)    COMP SYNC .
   05 FOPTS-EDC          PIC S9(8)    COMP SYNC .
   05 FOPTS-LIMIT        PIC S9(8)    COMP SYNC .
   05 FOPTS-NEW          PIC S9(8)    COMP SYNC .
   05 FOPTS-UNLOCK       PIC S9(8)    COMP SYNC .
   05 FOPTS-VTF          PIC S9(8)    COMP SYNC .
```

## Parameter Overview

FATTR-RETENTION-INTERVAL is the retention period for the file. Valid values are:

- Non-zero positive integer (for example, 60), which indicates the number of days to retain the file.
- -2, which indicates to retain the file forever.
- -1, which indicates zero days, or no retention.
- 0, which indicates the retention period is not specified at the file level and assumes the default value. StorHouse determines the default value as follows:
  - If the file's resident file set has a retention attribute equal to -2, -1, or a specified number of days, the file set retention attribute determines the file retention attribute.
  - If the file's resident file set has a retention attribute of 0, the RETENTION\_MODE system parameter determines the file retention attribute. If RETENTION\_MODE is set to BASIC, the file retention is ZERO. If RETENTION\_MODE is set to STRICT, the file retention is FOREVER.

**Note:** If force retention is set at the file set level, the file assumes the file set retention value.